GX-Force
Unit Test Specification
(with Results)

Document No.GX-Force_VR005

Approval	Review	Preparation
Name: Engineering Div.2 Mutou	Name: Engineering Div.2 Hirao	Name: Engineering Div.2 Ikarashi
Date 2021/7/30	Date 2021/7/30	Date 2021/7/7

No.	Date	Version	Revised content	Remarks
Ex.	20XX/XX/X	RevX.X	Create New	
1	2021/7/7	Rev1.0	Create New	
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Unit Test Specification

Test details and method

・Perform white-box testing of program unit.
Implementation: Use the C++test development support tool.
Desired coverage: C1 Level *Branch coverage: Execute all branches in the code at least once.
Schedule: 2021/6/4 to 2021/7/7

Modules targeted for unit test

File name	Function name	Last updated unit version	Owner	Last test date	Test owner	Judgment date	Judgment owner	Result	Remarks
ad_24bit.c	Ad24bit_Ad_Ave_Op	req[13-6-2-6]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_bit24AD_Conversion	req[13-6-2-7]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_Get_NC_Count	req[13-6-1-4]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_Int_Ad_Start	req[13-6-1-5]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_Int_Adc	req[13-6-2-8]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_Nc_Ad_CalcAverage	req[13-6-2-14]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_Set_AD	req[13-6-2-9]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	Ad24bit_Set_InitRAM	req[13-6-1-8]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	m_bit24AD_Acquisition	req[13-6-2-10]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	m_bit24AD_Channel_Check	req[13-6-2-11]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	m_bit24AD_Chk_ChangePGA	req[13-6-1-6]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	m_bit24AD_Chk_minus	req[13-6-2-12]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	m_bit24AD_Set_PGARegister	req[13-6-1-7]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ad_24bit.c	m_bit24AD_To_buffer	req[13-6-2-13]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Calc_AD_Main	req[3-2-6-7]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Fixed_PGA1_Flg	req[3-2-6-11]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Get_Chk_Status	req[3-2-6-1]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Get_Difference_AD	req[3-2-6-2]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Interval_Connect_Chk	req[3-2-6-3]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Not_Use_Flg	req[3-2-6-4]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Set_AD_Connect_Chk	req[3-2-6-5]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Set_InitRAM	req[3-2-6-10]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	ECSENSOR_Start_ConnectChk	req[3-2-6-6]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	m_Set_AD_Buf_Clr	req[3-2-6-8]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
ec_sensor.c	m_Set_EcChk_Port	req[3-2-6-9]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
interval_timer.c	IntervalTimer_Flag_Acquisition_250msec	req[13-4-1-2]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
interval_timer.c	IntervalTimer_Flag_Set_250msec	req[13-4-1-3]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
interval_timer.c	IntervalTimer_Int_Timer_Op	req[13-4-1-4]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
interval_timer.c	IntervalTimer_Set_InitRAM	req[13-4-1-5]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
lvd.c	LVD_Cyclic_250msecProcess_isFault	req[13-13-1-10]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
lvd.c	LVD_Set_InitRAM	req[13-13-1-9]	Ikarashi	2021/6/7	Taki	2021/6/7	Ikarashi	OK	
main.c (SENSOR)	Main_250msecCyclic_24hCheckCount	req[13-3-2-7]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
main.c (SENSOR)	Main_Get_FailStatus	req[3-2-4-4]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
main.c (SENSOR)	Main_Get_Gas_Setting	req[5-8-2-2]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
main.c (SENSOR)	Main_Process	req[13-4-2-9]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
main.c (SENSOR)	Main_Set_Gas_Setting	req[5-8-2-3]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	M_Set_Element_Port	req[1-1-7-1]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Element_Count	req[1-1-6-1]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Get_Long_Energy	req[11-2-2-5]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Get_Protection	req[1-1-6-2]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Reading_Element_Flg	req[1-1-6-3]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Set_InitRAM	req[13-11-1-1]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Set_Long_Energy	req[11-2-2-6]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Set_Protection	req[1-1-6-4]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	m_Set_NcProtectionFlg	req[1-1-7-2]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Set_ProtectionPermissionFlg	req[1-1-6-5]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
nc_sensor.c	NcSensor_Set_Warmup	req[1-1-1-9]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	f_Check_CopyRam	req[13-3-2-13]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	f_Check_MainRam	req[13-3-2-15]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	f_Cyclic_CheckProcess	req[13-3-2-10]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	f_Cyclic_CheckProcessMainPhase	req[13-3-2-12]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	f_Cyclic_CheckProcessSubPhase	req[13-3-2-11]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	f_Init_MainRam	req[13-3-2-14]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	FuncRamChk_Cyclic_250msecProcess	req[13-3-2-9]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	FuncRamChk_Start_Check	req[13-3-2-8]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
ram_chk.c	FuncRamChk_Start_InitRamChk	req[3-2-1-26]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
rom_chk.c	m_rom_sum_calc	req[3-2-1-27]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
rom_chk.c	ROMCHK_Get_SUM_NUM	req[3-2-1-28]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
rom_chk.c	ROMCHK_Get_SUMChkflg	req[3-2-1-29]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
rom_chk.c	ROMCHK_Process	req[3-2-1-30]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
rom_chk.c	ROMCHK_Set_SUMChkflg_Start	req[3-2-1-31]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
tool.c (SENSOR)	Tool_Asc_Hex	req[13-11-1-19]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
tool.c (SENSOR)	Tool_Bin_Asc	req[13-11-1-20]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
tool.c (SENSOR)	Tool_Comp_Pnt	req[13-11-1-21]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
tool.c (SENSOR)	Tool_Copy_Pnt	req[13-11-1-22]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
tool.c (SENSOR)	Tool_Hex_Asc	req[13-11-1-27]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_DU	req[13-7-3-27]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_DUR	req[13-7-2-35]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_DUW	req[13-7-2-36]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_HA	req[13-7-3-28]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_HAR	req[13-7-2-37]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_HAW	req[13-7-2-38]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_RC	req[13-7-3-29]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_RCR	req[13-7-2-39]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_VP	req[13-7-3-30]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	

uart_sau_command.c	m_SAU_Code_VPR	req[13-7-2-40]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_VS	req[13-7-3-31]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_VSC	req[13-7-2-41]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_VSR	req[13-7-2-42]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	m_SAU_Code_VSW	req[13-7-2-43]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	UartSAUCommand_Command_Op	req[13-7-2-44]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	UartSAUCommand_Get_Cmd_NUM	req[13-7-2-45]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	UartSAUCommand_Get_Comnd_Data	req[13-7-2-46]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_command.c	UartSAUCommand_Get_Comnd_Pointer	req[13-7-2-47]	Ikarashi	2021/6/8	Taki	2021/6/8	Ikarashi	OK	
uart_sau_main.c	m_Comand_Chk	req[13-7-3-32]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	m_Comu_Op_Sub	req[13-7-2-48]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	m_Rxd_Sum_Chk	req[13-7-3-37]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	m_Txd_Busy	req[13-7-2-49]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	m_Txd_Start	req[13-7-2-50]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	m_Txd_Sum_Calc	req[13-7-2-51]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UARTSAUMAIN_Comu_Op	req[13-7-3-33]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UartSAUMain_Err_Op	req[13-7-2-52]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UARTSAUMAIN_Get_Rxd_Data	req[13-7-3-34]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UartSAUMain_Head_Set	req[13-7-2-53]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UartSAUMain_Rxd_Buf_Pointre	req[13-7-3-35]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UARTSAUMAIN_Set_InitRAM	req[13-11-1-35]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UARTSAUMAIN_Set_Rxd_Next_Buf	req[13-7-3-36]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UartSAUMain_tx_nbyte	req[13-7-2-54]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
uart_sau_main.c	UartSAUMain_Txd_Buf_Pointre	req[13-7-2-55]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_camp.c	R_CAMP0_Start	req[13-6-1-20]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_camp.c	R_CAMP1_Start	req[13-6-1-21]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_camp.c	R_CAMP_Create	req[13-6-1-19]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_cgc.c	R_CGC_Create	req[13-12-1-2]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_cgc_user.c	R_CGC_Get_ResetSource	req[13-12-1-3]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_it.c	R_IT_Create	req[13-4-1-24]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_it.c	R_IT_Start	req[13-4-1-25]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_it.c	R_IT_Stop	req[13-4-1-26]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_it_user.c	r_it_interrupt	req[13-4-1-27]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_lvi.c	R_LVI_Create	req[13-13-1-7]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_lvi_user.c	r_lvi_interrupt	req[13-13-1-8]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_main.c	main(SENSOR)	req[1-3-12-1-7]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_main.c	R_MAIN_UserInit	req[1-3-12-1-9]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad.c	R_PGA_DSAD_CAMP_OffsetTrimming	req[13-6-1-9]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad.c	R_PGA_DSAD_Create	req[13-6-1-10]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad.c	R_PGA_DSAD_Get_AverageResult	req[13-6-1-11]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad.c	R_PGA_DSAD_Get_Result	req[13-6-1-12]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad.c	R_PGA_DSAD_Start	req[13-6-1-13]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad.c	R_PGA_DSAD_Stop	req[13-6-1-14]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_pga_dsad_user.c	r_pga_dsad_interrupt_conversion	req[13-6-1-15]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_port.c	R_PORT_Create	req[13-1-1-3]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau.c	R_SAU0_Create	req[13-7-1-1]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau.c	R_UART0_Create	req[13-7-1-2]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau.c	R_UART0_Receive	req[13-7-3-1]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau.c	R_UART0_Send	req[13-7-2-1]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau.c	R_UART0_Start	req[13-7-1-3]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau.c	R_UART0_Stop	req[13-7-1-4]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_callback_error	req[13-7-2-2]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_callback_receiveend	req[13-7-3-2]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_callback_sendend	req[13-7-2-3]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_interrupt_error	req[13-7-2-4]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_interrupt_receive	req[13-7-3-4]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_interrupt_send	req[13-7-2-5]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_sau_user.c	r_uart0_callback_softwareoverrun	req[13-7-3-3]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_systeminit.c	hdwinit	req[13-12-1-11]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_systeminit.c	R_Systeminit	req[13-12-1-12]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_wdt.c	R_WDT_Create	req[13-10-1-1]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_wdt.c	R_WDT_Restart	req[13-10-2-1]	Ikarashi	2021/6/9	Taki	2021/6/9	Ikarashi	OK	
r_cg_wdt_user.c	r_wdt_interrupt	req[13-10-1-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	AD12BIT_Init_Setting	req[13-6-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	Ad12bit_Ad_Ave_Op	req[13-6-2-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	Ad12bit_Calc_Voltage	req[13-6-2-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	Ad12bit_Check_InitalProcessFlag	req[13-6-2-4]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	Ad12bit_Get_Ad_Data	req[13-6-2-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	Ad12bit_Int_Ad_Start	req[13-6-1-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	Ad12bit_Int_Adc	req[13-6-1-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ad_12bit.c	m_bit12AD_To_Average_Buffer	req[13-6-2-5]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
alarm_event.c	AlarmEvent_Judg_AlarmEvent	req[2-1-5-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
alarm_event.c	AlarmEvent_Set_AlarmReset	req[2-1-4-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
alarm_event.c	AlarmEvent_Set_InitAlarmFlg	req[2-1-4-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
alarm_event.c	m_Check_GasAlarm	req[2-1-5-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
alarm_event.c	m_Set_GasAlarmFlg	req[2-1-5-4]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	Battery_Chk_Battery	req[10-4-3-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	Battery_Get_BatteryAD	req[10-4-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	Battery_Get_ComBatteryAD	req[10-4-1-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	Battery_Get_ComBatteryIconAD	req[10-4-3-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	Battery_Get_PowerOffFlg	req[6-11-2-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	Battery_Set_BatteryMark	req[10-4-4-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	f_Check_LowTempMoveFlag	req[10-4-3-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
Battery.c	m_Calc_BatteryAD	req[10-4-1-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	

buzzer.c	BUZZER_Chk_TestFlg	req[4-2-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Get_BuzzerOnOff	req[14-8-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Get_MotorOnOff	req[14-9-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Hz_Set_Op	req[14-8-3-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Init_Set	req[14-8-1-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Interrupt_A_CompareMatch	req[13-5-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Main_Loop_Op	req[14-8-3-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Set_AlarmSilence	req[2-6-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Set_Boot_Flg	req[14-8-3-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Set_FailBuzzer	req[3-1-5-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Set_PowerOFF	req[14-8-1-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Set_TestBuzzer	req[4-2-1-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Short_Command	req[14-8-3-4]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	BUZZER_Test_Command	req[4-2-1-3]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	f_Cyclic_250msecProcess	req[14-8-5-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_Check_AlarmCondition	req[2-1-5-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_Io_Buzzer_Set	req[14-8-1-5]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_Io_Led_Set	req[14-6-1-1]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_Io_Vibration_Set	req[14-9-1-2]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_cyc_exec	req[14-8-3-5]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_get_param	req[14-8-3-6]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_init_get_param	req[14-8-1-4]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
buzzer.c	m_judge_request	req[14-8-3-7]	Ikarashi	2021/6/10	Taki	2021/6/10	Ikarashi	OK	
ComCommand.c	ComCommand_Init_Process	req[7-1-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Check_UsbComNonPartner	req[7-1-1-9]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Chk_ReceiveSumFault	req[7-1-1-7]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Error_ReceiveData	req[7-1-1-6]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Get_DeviceStatus	req[7-1-1-10]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Get_DeviceTransmitEnable	req[7-1-1-11]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Get_ReceiveData	req[7-1-1-5]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Get_UsbComPartner	req[7-1-1-8]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Process_AnalysisData	req[7-1-1-4]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Start_Device	req[7-1-1-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	ComCommand_Stop_Device	req[7-1-1-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	f_Set_SendData	req[7-1-1-12]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
ComCommand.c	f_Sum_Calc	req[7-1-1-13]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bit.c	DA12bit_Init_Setting	req[13-14-1-5]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bit.c	DA12bit_Set_OutputNum	req[13-14-1-8]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bit.c	DA12bit_Start_Output	req[13-14-1-6]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bit.c	DA12bit_Stop_Output	req[13-14-1-7]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bit.c	f_Calc_RegisterNum	req[13-14-1-9]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bitSelfCheck.c	DA12bitSelfCheck_Init_Setting	req[13-14-1-10]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bitSelfCheck.c	DA12bitSelfCheck_Set_CheckStart	req[13-14-1-11]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
DA12bitSelfCheck.c	DA12bitSelfCheck_Set_CheckWaitTimer	req[13-14-1-12]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	DataFlash_flash_write	req[14-2-2-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	Flash_10msec_interrupt	req[14-2-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	flash_Block_Erase	req[14-2-2-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	flash_read	req[14-2-1-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	flash_status_read	req[14-2-1-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	m_Chg_Address_LNG_CHR	req[14-2-2-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_flash.c	m_flash_write_enable	req[14-2-2-4]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_AlarmReset_Load	req[11-9-1-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_AlarmReset_Write	req[11-9-2-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_FRAM_Diag	req[14-1-3-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_FRAM_Factory_load	req[6-10-2-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_AlarmResetAddress	req[14-1-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_ChkDataAddress	req[14-1-1-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_DataLoggerAddress	req[14-1-1-8]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_Dumpload	req[14-1-1-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_FramDataAddress	req[14-1-1-4]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_LunchBreakAddress	req[14-1-1-5]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_StIDAddress	req[14-1-1-6]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_Get_UserIDAddress	req[14-1-1-7]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_LunchBreak_Load	req[6-4-4-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_LunchBreak_Write	req[6-4-2-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	DataFram_write_factory_fram	req[6-10-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Get_DefaultAnsFlg	req[6-6-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Get_DefaultEndFlg	req[6-6-1-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Get_flg	req[14-1-2-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_ID_load	req[14-1-1-9]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_id_write_op	req[14-1-2-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Init_DataAddress	req[14-1-1-10]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_load	req[14-1-2-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_ChkData_Default	req[6-6-1-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_Default	req[6-6-1-4]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_DefaultStartFlg	req[6-6-1-5]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_FailFlg	req[3-2-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_flg	req[14-1-2-4]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_ID_Default	req[6-6-2-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_Set_MainDefault	req[6-6-1-6]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	FRAM_write_op	req[14-2-2-5]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_alarm_reset_sum_calc	req[11-9-2-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_FRAM_AlarmReset_Save	req[11-9-2-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_FRAM_AlarmReset_Side_Load	req[11-9-1-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_factory_save	req[14-1-2-5]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	

data_fram.c	m_FRAM_Factory_Side_Load	req[14-1-1-11]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_factory_sum_calc	req[14-1-3-2]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_id_save	req[14-1-2-6]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_id_save_all	req[14-1-2-7]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_FRAM_LunchBreak_Save	req[6-4-2-1]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_FRAM_LunchBreak_Side_Load	req[14-1-1-12]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_lunchbreak_sum_calc	req[14-1-3-3]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_read	req[14-1-1-13]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_save	req[14-1-2-8]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_FRAM_Side_Load	req[14-1-2-9]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_sum_calc	req[14-1-3-4]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_fram_write	req[14-1-2-10]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_write_and_check_fram	req[14-1-2-11]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_write_id_data_fram	req[14-1-2-12]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram.c	m_write_setting_fram	req[14-1-2-13]	Ikarashi	2021/6/11	Taki	2021/6/11	Ikarashi	OK	
data_fram_factory.c	DataFramFactry_Alarmpoint_load	req[6-10-2-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_fram_factory.c	DataFramFactry_All_load	req[6-10-2-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_fram_factory.c	DataFramFactry_Chk_AlarmReset_load	req[6-10-2-4]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_fram_factory.c	DataFramFactry_Chk_AlarmResetOnOffFlg	req[11-9-1-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_fram_factory.c	DataFramFactry_write_op	req[6-10-1-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_air_adjustment	req[5-1-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_bump	req[8-1-6-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_calibration	req[8-1-6-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_clear	req[8-2-2-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_clearPower	req[8-2-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_confirm	req[8-1-2-6]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_default	req[6-6-2-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_Init_EepAddress	req[8-2-1-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_peak_clear	req[1-8-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_power_off	req[8-1-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_power_on	req[8-1-1-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_Set_Default	req[8-1-1-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_setting_check	req[8-1-7-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_snap	req[8-1-8-13]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_snap_end_data	req[8-1-8-15]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_snap_read	req[8-1-8-14]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_start	req[8-1-2-7]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_stop	req[8-1-2-8]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	DataLogger_write_test	req[8-1-9-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_comp_pnt	req[8-1-2-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_copy_nbyte	req[8-1-2-9]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_flash_sector_erase	req[8-1-2-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_flash_status	req[8-1-2-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_initialize	req[6-6-2-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_isLeapYear	req[8-1-2-4]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm	req[8-1-4-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_event	req[8-1-4-2]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_event_check	req[8-1-4-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_event_data	req[8-1-4-4]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_event_fit	req[8-1-4-5]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_event_write	req[8-1-4-6]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_head	req[8-1-4-7]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_head_write	req[8-1-4-8]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_a_check	req[8-1-4-9]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_b_check	req[8-1-4-10]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_c_check	req[8-1-4-11]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_copy	req[8-1-4-12]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_data	req[8-1-4-13]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_pad_write	req[8-1-4-14]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_alarm_trend_write	req[8-1-4-15]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_bump_data	req[8-1-6-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_calibration_data	req[8-1-6-4]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_calibration_fit	req[8-1-6-5]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_calibration_write	req[8-1-6-6]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_change_header	req[8-1-2-10]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_chmap	req[13-1-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_clear_alarm_work	req[8-2-1-3]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_clear_wait	req[8-1-2-43]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_concentration	req[8-1-2-11]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_fatal	req[8-1-1-4]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_flash_write	req[8-1-2-12]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_full	req[8-4-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_full_check	req[8-1-2-13]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_gas	req[8-1-2-14]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_gas_invalid	req[8-1-2-15]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_gas_off	req[8-1-2-16]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_gas_result	req[8-1-2-5]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_get_snap_write_pnt	req[8-1-8-18]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_header	req[8-1-2-17]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_header_data	req[8-1-2-18]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_header_plus	req[8-1-2-44]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_interval_check	req[8-1-2-19]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_interval_event	req[8-1-2-20]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_interval_event_check	req[8-7-1-1]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	
data_logger.c	m_lg_interval_event_data	req[8-1-2-21]	Ikarashi	2021/6/14	Taki	2021/6/14	Ikarashi	OK	

data_logger.c	m_lg_interval_next	req[8-1-2-22]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_interval_pad_write	req[8-1-2-23]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_interval_trend	req[8-1-2-24]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_interval_trend_data	req[8-1-2-25]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_interval_trend_temp	req[8-1-2-26]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_interval_write	req[8-1-2-27]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_now_time	req[8-1-2-28]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_power_event_data	req[8-1-1-5]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_power_event_fit	req[8-1-1-6]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_power_event_write	req[8-1-1-7]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_ppm2code	req[8-1-2-29]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_remain	req[8-4-1-2]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_round	req[8-1-2-30]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_set_error	req[8-1-2-31]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_setting_data	req[8-1-7-2]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_setting_fit	req[8-1-2-32]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_setting_memory	req[8-1-7-3]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_setting_write	req[8-1-7-4]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_snap_data	req[8-1-8-16]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_snap_write	req[8-1-8-17]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_stop_interval	req[8-1-2-33]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trailer	req[8-1-2-34]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trailer_data	req[8-1-2-35]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trend_check	req[8-1-2-36]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trend_keep_going	req[8-1-4-16]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trouble_event	req[8-1-5-1]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trouble_event_check	req[8-1-5-2]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trouble_event_data	req[8-1-5-3]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trouble_event_fit	req[8-1-5-4]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_trouble_event_write	req[8-1-5-5]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_unset_error	req[8-1-2-37]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_wait	req[8-4-1-3]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_ui2uc	req[8-1-2-38]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_day	req[8-1-2-39]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_DiffTime	req[8-1-2-40]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_info_write	req[8-1-1-8]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_time	req[8-1-2-41]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_logger.c	m_lg_time_value	req[8-1-2-42]	lkarashi	2021/6/14	Taki	2021/6/14	lkarashi	OK	
data_struct_const.c	Data_Detector_Init_EepAddress	req[13-3-1-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	Data_Detector_Set_DataAddress	req[13-3-1-2]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	Data_Detector_Set_Default	req[6-6-1-9]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_def_DenstData	req[13-11-1-2]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_DefaultStId	req[6-6-1-7]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_DefaultUserId	req[6-6-1-8]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_Set_DenstDataAddress	req[13-3-1-3]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_Set_FramGasSpec	req[13-3-1-4]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_Set_GasSpec	req[13-3-1-5]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_Set_NCGasSpec	req[11-5-2-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	DataStructConst_Set_StatusDataAddress	req[13-3-1-6]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_Alarm1hON	req[5-8-1-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_AlarmTwaON	req[13-11-1-3]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_CombGasON	req[5-8-1-2]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_Gas_Comb	req[5-8-2-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_GasData_Add	req[13-3-1-7]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_H2MeasurementFlag	req[13-11-1-39]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_IDDispON	req[6-5-1-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasCalc_Get_ToxicGasON	req[5-8-1-3]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasTableData_Init_EepAddress	req[13-3-1-8]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
data_struct_const.c	GasTableData_Set_Default	req[6-6-1-10]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Chk_AlarmSettingEndNum	req[2-2-1-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Chk_AlarmSettingResetNum	req[2-2-1-2]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Disp_AlarmPointChange	req[2-2-1-3]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Disp_AlarmSettingMenu	req[2-2-1-4]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Disp1_FactoryDefault	req[11-9-1-4]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Disp2_FactoryDefault	req[11-9-1-5]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_End_AlarmPointChange	req[2-2-1-5]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Init_AlarmPointChange	req[2-2-1-6]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Init_AlarmSettingMenu	req[2-2-1-7]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Select_AlarmPointChange	req[2-2-1-8]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Select_AlarmSettingMenu	req[2-2-1-9]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Select_PointTypeChange	req[2-2-1-10]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_alarm_setting_display.c	FlowAlarmSettingDisplay_Set_FactoryDefault	req[11-9-1-6]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Disp_BeepSelect	req[6-3-2-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Disp_BeepTime	req[6-3-2-2]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_End_BeepSelect	req[6-3-2-3]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_End_BeepTime	req[6-3-2-4]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Init_BeepMenu	req[6-3-2-5]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Init_BeepSelect	req[6-3-2-6]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Init_BeepTime	req[6-3-2-7]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Select_BeepMenu	req[6-3-2-8]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Select_BeepSelect	req[6-3-2-9]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Select_BeepTime	req[6-3-2-10]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_beep_display.c	FlowBeepDisplay_Set_BeepMenu	req[6-3-2-11]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Disp_BumpExprd	req[5-6-2-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Disp_BumpInterval	req[5-6-6-1]	lkarashi	2021/6/15	Taki	2021/6/15	lkarashi	OK	

flow_bump_limit_display.c	FlowBumpLimitDisplay_Disp_BumpReminder	req[5-6-5-1]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_End_BumpExprd	req[5-6-2-2]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_End_BumpInterval	req[5-6-6-2]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_End_BumpReminder	req[5-6-5-2]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Init_BumpExprd	req[5-6-2-3]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Init_BumpInterval	req[5-6-6-3]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Init_BumpLimitMenu	req[5-6-2-4]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Init_BumpReminder	req[5-6-5-3]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Select_BumpExprd	req[5-6-2-5]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Select_BumpInterval	req[5-6-6-4]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Select_BumpLimitMenu	req[5-6-2-6]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Select_BumpReminder	req[5-6-5-4]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_limit_display.c	FlowBumpLimitDisplay_Set_BumpLimitMenu	req[5-6-2-7]	Ikarashi	2021/6/15	Taki	2021/6/15	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Disp_BumpCalOnOff	req[4-1-3-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Disp_BumpCalTime	req[4-1-4-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Disp_BumpLimit	req[4-1-4-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Disp_BumpTime	req[4-1-4-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_End_BumpCalOnOff	req[4-1-3-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_End_BumpCalTime	req[4-1-4-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_End_BumpLimit	req[4-1-4-5]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_End_BumpTime	req[4-1-4-6]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Init_BumpCalOnOff	req[4-1-3-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Init_BumpCalTime	req[4-1-4-7]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Init_BumpLimit	req[4-1-4-8]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Init_BumpParameterMenu	req[4-1-4-9]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Init_BumpTime	req[4-1-4-10]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Select_BumpCalOnOff	req[4-1-3-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Select_BumpCalTime	req[4-1-4-11]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Select_BumpLimit	req[4-1-4-12]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Select_BumpParameterMenu	req[4-1-4-13]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Select_BumpTime	req[4-1-4-14]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_parameter_display.c	FlowBumpParameterDisplay_Set_BumpParameterMenu	req[4-1-4-15]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_10msec_interrupt	req[13-4-1-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpAlarm	req[4-1-1-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpCalNecessity	req[4-1-2-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpCalOnOff	req[4-1-2-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpCalStartInitial	req[4-1-2-11]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpCalTimeEnd	req[4-1-2-5]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpGasSelect	req[4-1-1-5]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpSuccess	req[4-1-2-10]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpTestResult	req[4-1-1-6]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_BumpTestTimeEnd	req[4-1-1-7]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Chk_HCModeWarmFlg	req[4-1-2-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Com_BumpTest	req[4-1-2-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Disp_BumpGasSelect	req[4-1-1-8]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Disp_BumpResult	req[4-1-1-9]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Disp_BumpTest	req[4-1-1-10]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Disp_SetBumpCal	req[4-1-2-6]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Execution_BumpCal	req[4-1-2-7]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Get_BumpCalTime	req[4-1-2-8]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Get_BumpGasSelect	req[4-1-1-11]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Get_BumpResultDenst	req[4-1-1-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Get_BumpSDMSelear	req[4-1-1-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Get_BumpTestTime	req[4-1-1-12]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Init_BumpTest	req[4-1-1-13]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_SelectUp_BumpGasSelect	req[4-1-1-15]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Select_BumpGasSelect	req[4-1-1-14]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Select_BumpResult	req[4-1-1-16]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Set_BumpCalData	req[4-1-2-9]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Set_BumpErrorOff	req[4-1-5-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	FlowBumpTestDisplay_Set_BumpSDMSelear	req[4-1-1-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	m_Chk_BumpPermission	req[4-1-1-17]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	m_judg_BumpTest	req[4-1-1-18]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_bump_test_display.c	m_Set_BumpResult	req[4-1-1-19]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Disp_CalExprd	req[5-5-1-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Disp_CalInterval	req[5-5-6-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Disp_CalReminder	req[5-5-5-1]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_End_CalExprd	req[5-5-1-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_End_CalInterval	req[5-5-6-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_End_CalReminder	req[5-5-5-2]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Init_CalExprd	req[5-5-1-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Init_CalInterval	req[5-5-6-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Init_CalLimitMenu	req[5-5-1-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Init_CalReminder	req[5-5-5-3]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Select_CalExprd	req[5-5-1-5]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Select_CalInterval	req[5-5-6-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Select_CalLimitMenu	req[5-5-1-6]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Select_CalReminder	req[5-5-5-4]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_cal_limit_display.c	FlowCalLimitDisplay_Set_CalLimitMenu	req[5-5-1-7]	Ikarashi	2021/6/16	Taki	2021/6/16	Ikarashi	OK	
flow_com_display.c	FlowComDisplay_Disp_SDMMode	req[9-7-2-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_com_display.c	FlowComDisplay_Set_ComConnect	req[9-7-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_AlarmPointData	req[10-2-5-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_BumpData	req[11-8-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_CalData	req[11-7-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_HCGasSetting	req[11-5-2-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	

flow_disp_display.c	FlowDispDisplay_Disp_HCLongSetting	req[11-2-2-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_PeakClearHold	req[1-8-1-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_PeakClearRelease	req[1-8-1-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_PumpOff	req[14-12-1-11]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_SnapData	req[8-1-8-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_StationIdSelect	req[6-5-2-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Disp_UserIdSelect	req[6-5-3-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_End_HCGasSetting	req[11-5-2-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_End_HCLongSetting	req[11-2-2-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_End_PumpOff	req[14-12-1-10]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_End_StationIdSelect	req[6-5-2-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_End_UserIdSelect	req[6-5-3-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Get_LightOnPermission	req[14-7-1-9]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_AlarmPointData	req[10-2-5-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_BumpData	req[11-8-1-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_CalData	req[11-7-1-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_DispMenu	req[9-2-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_HCGasSetting	req[11-5-2-5]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_HCLongSetting	req[11-2-2-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_PumpOff	req[14-12-1-9]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_SnapData	req[8-1-8-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_StationIdSelect	req[6-5-2-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Init_UserIdSelect	req[6-5-3-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_AlarmPointData	req[10-2-5-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_BumpData	req[11-8-1-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_CalData	req[11-7-1-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_DispMenu	req[9-2-1-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_HCGasSetting	req[11-5-2-6]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_HCLongSetting	req[11-2-2-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_SnapDetail	req[8-1-8-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_SnapRecData	req[8-1-8-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_SnapUpDownMode	req[8-1-8-5]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_StationIdSelect	req[6-5-2-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Select_UserIdSelect	req[6-5-3-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Set_AlarmPointData	req[4-2-1-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Set_DispMenu	req[9-2-1-3]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	FlowDispDisplay_Set_LightOnOffSetting	req[14-7-1-8]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Chk_DispHCGasSel	req[11-5-2-2]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Chk_DispMenuItem	req[9-2-1-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_1h	req[1-15-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_Light	req[14-7-1-10]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_Peak	req[1-7-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_SnapConcentration	req[8-1-8-7]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_SnapDateTime	req[8-1-8-6]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_SnapNoData	req[8-1-8-8]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_Stel	req[1-11-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Disp_Twa	req[1-13-1-1]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_disp_display.c	m_Select_CalBumpData	req[11-7-1-4]	Ikarashi	2021/6/17	Taki	2021/6/17	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Chk_GasCombChannelSelect2chH2Num	req[5-8-1-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Chk_GasCombChannelSelectEndNum	req[5-8-1-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Chk_RomSumDisp	req[10-5-3-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp_GasCombChannelSelect	req[5-8-1-6]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp_GasCombGasSelect	req[5-8-1-7]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp_RomSumDisp	req[10-5-2-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp_RunningFactoryDefault	req[6-10-1-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp1_FactoryDefault	req[6-10-1-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp2_FactoryDefault	req[6-10-1-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_End_FlowAdjSetting	req[14-13-1-37]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_End_FlowAdjSettingAndCheck	req[14-13-1-38]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_End_FlowCheck	req[14-13-1-42]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_End_GasCombGasSelect	req[5-8-1-8]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp_FlowAdjSetting	req[14-13-1-39]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Disp_FlowCheck	req[14-13-1-43]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_ADDisp	req[10-7-1-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_FactMenu	req[9-6-1-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_FlowAdjSetting	req[14-13-1-36]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_FlowCheck	req[14-13-1-41]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_GasCombChannelSelect	req[5-8-3-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_GasCombGasSelect	req[5-8-3-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Init_RomSumDisp	req[10-5-1-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Select_ADDispNum	req[10-7-1-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Select_FactMenu	req[9-6-1-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Select_FlowAdjSetting	req[14-13-1-40]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Select_GasCombChannelSelect	req[5-8-3-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Select_GasCombGasSelect	req[5-8-3-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Set_ADDisp	req[10-7-1-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Set_FactMenu	req[9-6-1-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowFactDisplay_Set_FactoryDefault	req[6-10-1-6]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowGasCalDisplay_Disp_DefaultConfirmation	req[6-6-1-11]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowGasCalDisplay_Disp_DefaultNG	req[6-6-1-12]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowGasCalDisplay_Disp_DefaultOK	req[6-6-1-13]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_fact_display.c	FlowGasCalDisplay_Disp_SetDefault	req[6-6-1-14]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Chk_AutoCalMenuGasOn	req[5-4-1-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Chk_AutoCalMenuZIPC	req[5-4-1-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Chk_AutoCalSelectMenu	req[5-4-1-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	

flow_gascal_display.c	FlowGasCalDisplay_Chk_HCModeWarmFlg	req[11-5-2-7]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AirCalDenst	req[5-1-1-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AirCalPass	req[5-1-1-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AirFail	req[5-1-2-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AirHold	req[5-1-1-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AirRelease	req[5-1-1-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AutoCal_ChgDenst	req[5-4-2-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AutoCal_SelectGas	req[5-4-2-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AutoCalBlinkDenst	req[5-4-1-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AutoCalDenst	req[5-4-1-6]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AutoCalPass	req[5-4-1-7]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_AutoCalRunning	req[5-4-1-8]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_EndAutoCalDenst	req[5-4-1-9]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_FailAutoCal	req[5-4-4-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_GroupSelectGas	req[5-4-3-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_GroupSelectGroup	req[5-4-3-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_HCModeWarmFlg	req[11-5-1-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Disp_SpanMargin	req[11-3-2-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_End_AutoCal_ChgDenst	req[5-4-2-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_End_GroupSelectGroup	req[5-4-3-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Execution_AutoCal	req[5-4-1-10]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Get_SpanMarginOnOff	req[11-3-3-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_AutoCal_ChgDenst	req[5-4-2-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_AutoCal_SelectGas	req[5-4-2-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_AutoCalBlinkDenst	req[5-4-1-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_AutoCalMenu	req[5-4-1-11]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_GasCalMenu	req[9-3-1-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_GroupSelectGas	req[5-4-3-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_GroupSelectGroup	req[5-4-3-6]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Init_LimitGasCalMenu	req[9-3-1-2]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_SelectUp_AutoCalMenu	req[5-4-1-13]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Select_AutoCal_ChgDenst	req[5-4-2-6]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Select_AutoCal_SelectGas	req[5-4-2-7]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Select_AutoCalMenu	req[5-4-1-12]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Select_GasCalMenu	req[9-3-1-3]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Select_GroupSelectGas	req[5-4-3-7]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Select_GroupSelectGroup	req[5-4-3-8]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	FlowGasCalDisplay_Set_GasCalMenu	req[9-3-1-4]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	m_Chk_AutoCalGasOn	req[5-4-3-1]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gascal_display.c	m_Chk_GasCalMenuItem	req[9-3-1-5]	Ikarashi	2021/6/18	Taki	2021/6/18	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Disp_ChangeLELValue	req[11-6-2-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Disp_MarginDispOnOff	req[11-3-3-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Disp_ResetAlarmPoint	req[11-9-2-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Disp_StealthMotorOnOff	req[11-4-2-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Disp_StealthOnOff	req[11-4-2-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_End_ChangeLELValue	req[11-6-2-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_End_MarginDispOnOff	req[11-3-3-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_End_ResetAlarmPoint	req[11-9-2-5]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_End_StealthOnOff	req[11-4-2-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Init_ChangeLELValue	req[11-6-2-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Init_GasSelMenu	req[9-5-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Init_MarginDispOnOff	req[11-3-3-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Init_StealthMotorOnOff	req[11-4-2-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Init_StealthOnOff	req[11-4-2-5]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Select_ChangeLELValue	req[11-6-2-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Select_GasSelMenu	req[9-5-1-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Select_MarginDispOnOff	req[11-3-3-5]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Select_StealthMotorOnOff	req[11-4-2-6]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Select_StealthOnOff	req[11-4-2-7]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_gassel_display.c	FlowGasSelDisplay_Set_GasSelMenu	req[9-5-1-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Calc_InitDispTime	req[9-8-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Chk_HCModeHHCflg	req[11-5-2-8]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_BumpLimitEffe	req[5-6-4-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_BumpLimitOk	req[5-6-4-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_BumpLimitOper	req[5-6-4-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_BumpLimitRequ	req[5-6-4-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_CalLimitEffe	req[5-5-4-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_CalLimitOk	req[5-5-4-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_CalLimitOper	req[5-5-4-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_CalLimitRequ	req[5-5-4-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_MainteNoticeEffe	req[5-7-4-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_MainteNoticeOk	req[5-7-4-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_MainteNoticeOper	req[5-7-4-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_MainteNoticeRequ	req[5-7-4-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_PleaseWait	req[6-11-2-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_ResumeConfirmation	req[6-4-3-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_ResumeFail	req[6-4-3-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_ResumeOFF	req[6-4-3-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Disp_ResumeON	req[6-4-3-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_1hAlarm	req[10-2-4-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_1stAlarm	req[10-2-4-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_2ndAlarm	req[10-2-4-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_3rdAlarm	req[10-2-4-4]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_AutoZeroConfirm	req[5-3-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_Battery	req[10-4-2-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	

flow_initial_display.c	FlowInitialDisplay_DispSet_DateAndTime	req[10-3-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_FS	req[10-2-2-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_GasName	req[10-2-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_SensorInitFail	req[3-1-2-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_StationId	req[10-6-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_STELAlarm	req[10-2-4-5]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_TWAAlarm	req[10-2-4-6]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_DispSet_UserId	req[10-6-2-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Get_InitDispTime	req[9-8-1-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Set_AllOffDisp	req[6-11-2-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Set_AllOnDisp	req[9-8-1-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_initial_display.c	FlowInitialDisplay_Set_HCModeHHCflgGasName	req[11-5-2-9]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_main.c	FlowMain_Disp_FailDisp	req[3-1-2-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_main.c	FlowMain_Get_ModeStatus	req[9-9-1-1]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_main.c	FlowMain_Set_ModeStatus	req[9-9-1-2]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_main.c	FlowMain_Select_MenuNum	req[9-9-1-3]	Ikarashi	2021/6/21	Taki	2021/6/21	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Chk_ChangeSensorDateEndNum	req[12-2-1-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_DigitDown_MaintePassSelect	req[6-8-4-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_DigitUp_MaintePassSelect	req[6-8-4-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_AlarmPatternOnOff	req[2-3-1-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_AlarmSilenceOnOff	req[2-6-1-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_AutoZeroOnOff	req[5-3-2-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_ChangeSenDateSet	req[12-2-1-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_ChangeSensorDate	req[12-2-1-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_DemandZeroOnOff	req[5-2-2-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_DispPumpOff	req[14-12-1-14]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_DispSuppressOnOff	req[5-8-5-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_DispZeroFllwrOnOff	req[5-8-4-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_GasTest	req[12-1-1-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_IdDispOnOff	req[6-5-1-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_MaintePassOnOff	req[6-8-4-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_MaintePassSelect	req[6-8-4-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp_RunningFactoryDefault	req[6-10-2-5]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp1_FactoryDefault	req[6-10-2-6]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Disp2_FactoryDefault	req[6-10-2-7]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_AlarmPatternOnOff	req[2-3-1-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_AutoZeroOnOff	req[5-3-2-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_ChangeSenDateSet	req[12-2-2-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_DemandZeroOnOff	req[5-2-2-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_DispPumpOff	req[14-12-1-13]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_DispSuppressOnOff	req[5-8-5-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_DispZeroFllwrOnOff	req[5-8-4-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_IdDispOnOff	req[6-5-1-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_MaintePassOnOff	req[6-8-4-5]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_End_MaintePassSelect	req[6-8-4-6]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_AlarmPatternOnOff	req[2-3-1-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_AlarmSilenceOnOff	req[2-6-1-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_AutoZeroOnOff	req[5-3-2-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_ChangeSensorDate	req[12-2-2-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_DemandZeroOnOff	req[5-2-2-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_DispPumpOff	req[14-12-1-12]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_DispSuppressOnOff	req[5-8-5-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_DispZeroFllwrOnOff	req[5-8-4-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_IdDispOnOff	req[6-5-1-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_MainteMenu	req[9-4-1-1]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_MaintePassOnOff	req[6-8-4-7]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Init_MaintePassSelect	req[6-8-4-8]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_AlarmPatternOnOff	req[2-3-1-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_AlarmSilenceOnOff	req[2-6-1-5]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_AutoZeroOnOff	req[5-3-2-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_ChangeSensorDate	req[12-2-2-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_DemandZeroOnOff	req[5-2-2-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_DispPumpOff	req[14-12-1-15]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_DispSuppressOnOff	req[5-8-5-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_DispZeroFllwrOnOff	req[5-8-4-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_IdDispOnOff	req[6-5-1-5]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_MainteMenu	req[9-4-1-2]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_MaintePassOnOff	req[6-8-4-9]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Select_MaintePassSelect	req[6-8-4-10]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Set_FactoryDefault	req[6-10-2-8]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	FlowMainteDisplay_Set_MainteMenu	req[9-4-1-3]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_display.c	m_Chk_MainteMenuItem	req[9-4-1-4]	Ikarashi	2021/6/22	Taki	2021/6/22	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Disp_CalExprd	req[5-7-2-1]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Disp_MainteNoticeDay	req[5-7-6-1]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Disp_MainteNoticeMenu	req[5-7-2-2]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Disp_MainteNoticeOnOff	req[5-7-5-1]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Disp_MainteNoticeReset	req[5-7-6-2]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_End_CalExprd	req[5-7-2-3]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_End_MainteNoticeDay	req[5-7-6-3]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_End_MainteNoticeOnOff	req[5-7-5-2]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_End_MainteNoticeReset	req[5-7-6-4]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Init_CalExprd	req[5-7-2-4]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Init_MainteNoticeDay	req[5-7-6-5]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Init_MainteNoticeMenu	req[5-7-2-5]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Init_MainteNoticeOnOff	req[5-7-5-3]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	

flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Select_CalExprd	req[5-7-2-6]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Select_MainteNoticeDay	req[5-7-6-6]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Select_MainteNoticeMenu	req[5-7-2-7]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_mainte_notice_display.c	FlowMainteNoticeDisplay_Select_MainteNoticeOnOff	req[5-7-5-4]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_meas_display.c	FlowMeasDisplay_Set_MeasDisp	req[9-1-2-1]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Chg_PassNum	req[6-8-1-1]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Chg_UpDown	req[6-8-1-2]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Chk_End	req[6-8-1-3]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Chk_Pass	req[6-8-1-4]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Disp_PassErr	req[6-8-1-5]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Get_PassType	req[6-8-1-6]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Get_Return	req[6-8-1-7]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Input_Pass	req[6-8-1-8]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Set_Enter	req[6-8-1-9]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_pass_display.c	FlowPassDisplay_Set_PassInit	req[6-8-1-10]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Set_TurnOff	req[6-11-2-5]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Set_UpData	req[7-1-2-1]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_10msec_Interrupt	req[6-11-2-11]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Chk_PurgeTimeEnd	req[6-11-2-10]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Get_PumpOffFlag	req[14-12-1-17]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Get_PurgeFlag	req[6-11-2-8]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Init_Process	req[6-11-2-4]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Set_PumpOffFlag	req[14-12-1-16]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Set_Purge	req[6-11-2-6]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Set_PurgeFlag	req[6-11-2-7]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_proces_display.c	FlowProcesDisplay_Set_PurgeTime	req[6-11-2-9]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_snaplog_display.c	FlowSnaplogDisplay_Disp_SnapDisp	req[8-1-8-10]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_snaplog_display.c	FlowSnaplogDisplay_Set_SnapLog	req[8-1-8-11]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_snaplog_display.c	FlowSnaplogDisplay_Disp_SnapSaved	req[8-1-8-12]	Ikarashi	2021/6/23	Taki	2021/6/23	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Chk_UserPassOnOff	req[6-8-2-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Chk_ZeroSuppressEsc	req[5-8-5-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_DigitDown_UserPassSelect	req[6-8-2-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_DigitUp_UserPassSelect	req[6-8-2-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_BackLightTime	req[6-1-2-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_DispModeSettingItemDisplay	req[6-7-1-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_KeyTone	req[6-2-2-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_LunchBreak	req[6-4-1-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_UserPassOnOff	req[6-8-2-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_UserPassSelect	req[6-8-2-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_ZeroFollowerGas	req[5-8-4-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_ZeroFollowerSet	req[5-8-4-6]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_ZeroSuppressGas	req[5-8-5-6]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Disp_ZeroSuppressSet	req[5-8-5-7]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_BackLightTime	req[6-1-2-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_DateSetteng	req[10-3-4-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_DispModeSettingItemDisplay	req[6-7-1-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_KeyTone	req[6-2-2-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_LunchBreak	req[6-4-1-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_UserPassOnOff	req[6-8-2-6]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_UserPassSelect	req[6-8-2-7]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_ZeroFollowerSet	req[5-8-4-11]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_End_ZeroSuppressSet	req[5-8-5-12]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_BackLightTime	req[6-1-2-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_BumpLimitUserMenu	req[9-3-1-8]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_CalLimitUserMenu	req[9-3-1-7]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_DateSetteng	req[10-3-4-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_DispModeSettingItemDisplay	req[6-7-1-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_KeyTone	req[6-2-2-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_LunchBreak	req[6-4-1-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_UserMenu	req[9-3-1-6]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_UserPassOnOff	req[6-8-2-8]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_UserPassSelect	req[6-8-2-9]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_ZeroFollowerGas	req[5-8-4-7]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_ZeroFollowerSet	req[5-8-4-8]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_ZeroSuppressGas	req[5-8-5-8]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Init_ZeroSuppressSet	req[5-8-5-9]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_BackLightTime	req[6-1-2-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_DateSettengItem	req[10-3-4-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_DateSettingNum	req[10-3-4-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_DispModeSettingItemDisplay	req[6-7-1-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_KeyTone	req[6-2-2-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_LunchBreak	req[6-4-1-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_UserMenu	req[9-3-1-9]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_UserPassOnOff	req[6-8-2-10]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_UserPassSelect	req[6-8-2-11]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_ZeroFollowerGas	req[5-8-4-9]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_ZeroFollowerSet	req[5-8-4-10]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_ZeroSuppressGas	req[5-8-5-10]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Select_ZeroSuppressSet	req[5-8-5-11]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Set_DateSetting	req[10-3-4-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Set_EndDisp	req[9-3-1-10]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	FlowUserDisplay_Set_UserMenu	req[9-3-1-11]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
flow_user_display.c	m_Chk_UserMenuItem	req[9-3-1-12]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	FuncBumpLimit_Chk_BumpLimitDay	req[5-6-3-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	FuncBumpLimit_Chk_BumpLimitExpired	req[5-6-3-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	

func_bump_limit.c	FuncBumpLimit_Get_BumpLimitExpiredBehavior	req[5-6-3-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	FuncBumpLimit_Get_BumpLimitOnOff	req[5-6-1-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	FuncBumpLimit_Get_ChkBumpLimitExpired	req[5-6-3-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	FuncBumpLimit_Set_BumpEndDenst	req[5-6-6-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	FuncBumpLimit_Set_BumpCalFlagSuccess	req[5-6-3-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_bump_limit.c	m_Chk_BumpLimitFlag	req[5-6-3-6]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Calc_Cal_BumpLimitDay	req[5-5-3-1]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Chk_CalLimitDay	req[5-5-3-2]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Chk_CalLimitExpired	req[5-5-3-3]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Get_CalLimitExpiredBehavior	req[5-5-3-4]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Get_CalLimitOnOff	req[5-5-1-8]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Get_ChkCalLimitExpired	req[5-5-3-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Set_CalEndDenst	req[5-5-6-5]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_cal_limit.c	FuncCalLimit_Set_CalStartDenst	req[5-5-6-6]	Ikarashi	2021/6/24	Taki	2021/6/24	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Chk_ConfirmEffectiveValue	req[3-2-1-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Chk_SensorTrouble	req[3-2-5-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Chk_ThermistorTrouble	req[3-2-4-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Int_1sec	req[3-2-1-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_BakUpBattFailFlg	req[3-2-2-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_BattFailFlg	req[3-2-7-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_ChkStart	req[3-2-1-4]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_CircuitVoltageFailFlg	req[3-2-3-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_ClockFailFlg	req[3-2-2-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_FlowFailFlg	req[3-2-9-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_FramFailFlg	req[3-2-1-5]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_PumpFailFlg	req[3-2-10-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_RamFailFlg	req[3-2-1-6]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_SensorCalFlg	req[3-2-5-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_SensorSubstrateFailFlg	req[3-2-4-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_SensorTroubleFlg	req[3-2-5-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	FuncChkMain_Set_ThermistorFailFlg	req[3-2-4-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_Range1stAlarm	req[3-2-1-7]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_Range2ndAlarm	req[3-2-1-8]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_Range3rdAlarm	req[3-2-1-9]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeAlarm	req[3-2-1-10]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeCalc	req[3-2-1-11]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeDigitCalc	req[3-2-1-12]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeFloat	req[3-2-1-13]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeFSCalc	req[3-2-1-14]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeNum	req[3-2-1-15]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeSpanCalc	req[3-2-1-16]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeStelAlarm	req[3-2-1-17]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeTwaAlarm	req[3-2-1-18]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Chk_RangeZeroCalc	req[3-2-1-19]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_chk_main.c	m_Set_FailCode	req[8-3-1-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	FuncConfirmationBeep_1sec_interrupt	req[6-3-1-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	FuncConfirmationBeep_Clr_ActiveFlg	req[6-3-2-12]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	FuncConfirmationBeep_Update_AlarmActiveFlg	req[6-3-1-4]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	FuncConfirmationBeep_Update_BumpCalActiveFlg	req[6-3-1-5]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	FuncConfirmationBeep_Update_DownloadActiveFlg	req[6-3-1-6]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	FuncConfirmationBeep_Update_LimitActiveFlg	req[6-3-1-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	m_Set_BeepEvent	req[6-3-1-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_confirmation_beep.c	m_Set_NciActiveFlg	req[6-3-1-7]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor.c	ECSENSOR_Calc_AD_Main	req[3-2-6-7]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	FuncECSensorChk_10msec_Interrupt	req[3-2-6-13]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	FuncECSensorChk_1sec_Interrupt	req[3-2-6-14]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	FuncECSensorChk_Set_StartChk	req[3-2-6-15]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	m_Chk_AD	req[3-2-6-16]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	m_Set_ECSensorChk	req[3-2-6-17]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	m_Set_Err	req[3-2-6-18]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_ec_sensor_chk.c	m_Set_StartInitialChk	req[3-2-6-12]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_lunch_break.c	FuncLunchBreak_Get_LunchBreakFramData	req[6-4-4-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_lunch_break.c	FuncLunchBreak_Get_LunchBreakOnOff	req[6-4-1-5]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_lunch_break.c	FuncLunchBreak_Get_StatusFlg	req[6-4-4-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_lunch_break.c	FuncLunchBreak_Set_LunchBreakData	req[6-4-4-4]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_lunch_break.c	FuncLunchBreak_Set_LunchBreakFramData	req[6-4-2-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_lunch_break.c	FuncLunchBreak_Set_OnOffFlg	req[6-4-3-5]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Chk_MainteLimitDay	req[5-7-3-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Chk_MainteNoticeLimitExpired	req[5-7-3-4]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Get_MainteLimitExpiredBehavior	req[5-7-3-2]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Get_MainteNoticeLimitExpired	req[5-7-3-5]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Get_MainteNoticeOnOff	req[5-7-1-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Set_MainteNoticeReset	req[5-7-6-7]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Set_MainteNoticeStart	req[5-7-6-8]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	FuncMainteNotice_Set_MainteNoticeUpDate	req[5-7-6-9]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_mainte_notice.c	m_Calc_MainteLimitDay	req[5-7-3-3]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_margin_chk.c	FuncMarginChk_Calc_SpanMargine	req[11-3-1-1]	Ikarashi	2021/6/25	Taki	2021/6/25	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Func_main	req[14-16-1-6]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Get_AD_data	req[14-16-1-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Get_WarmFlg	req[14-16-1-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Interval_10msec	req[14-16-1-7]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Set_AD	req[14-16-1-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Set_Init	req[14-16-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	O2WarmUp_Set_InitRAM	req[14-16-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_o2_warm_up.c	m_O2WarmUp_Port_Term	req[14-16-1-8]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	

func_ram_chk.c	f_Check_CopyRam	req[13-3-2-6]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	f_Check_MainRam	req[13-3-2-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	f_Cyclic_CheckProcess	req[13-3-2-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	f_Init_MainRam	req[13-3-2-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	FuncRamChk_Cyclic_100msecProcess	req[13-3-2-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	FuncRamChk_Set_RamFailFlg	req[3-2-1-20]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	FuncRamChk_Start_Check	req[13-3-2-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_ram_chk.c	FuncRamChk_Start_InitRamChk	req[3-2-1-21]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_rom_chk.c	FuncRomChk_Chk_EndRomChk	req[3-2-1-22]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_rom_chk.c	FuncRomChk_Get_SumNum	req[13-2-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_rom_chk.c	FuncRomChk_process_RomChk	req[3-2-1-23]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_rom_chk.c	FuncRomChk_Start_InitRomChk	req[3-2-1-24]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_rom_chk.c	FuncRomChk_Start_RomChk	req[3-2-1-25]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_rom_chk.c	m_Calc_RomSum	req[13-2-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	FuncSDM_Get_CombData	req[9-7-2-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	FuncSDM_Get_ExCommandFlg	req[9-7-2-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	FuncSDM_Set_DispData	req[9-7-2-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	FuncSDM_Set_ExCommandData	req[9-7-2-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	FuncSDM_Set_ExCommandFlg	req[9-7-2-6]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	m_Disp_MainteData	req[9-7-2-7]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_sdm.c	m_Disp_MeasData	req[9-7-2-8]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_temp_warning.c	FuncTempWarning_10msec_interrupt	req[11-1-2-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_temp_warning.c	FuncTempWarning_Chk_OutOfRange	req[11-1-2-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_temp_warning.c	FuncTempWarning_Get_TempBuzzerFlg	req[11-1-2-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_temp_warning.c	FuncTempWarning_Get_OutOfRangeFlg	req[11-1-2-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
func_temp_warning.c	FuncTempWarning_Reset_OutOfRangeBuzzer	req[11-1-2-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	GasCalc_10msec_interrupt	req[1-1-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	GasCalc_Calc_Gas_Concentration	req[1-1-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	GasCalc_Clear_GasFailFlg	req[3-1-5-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	GasCalc_Clear_HCPeak	req[1-6-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	GasCalc_Clear_Peak	req[1-6-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	f_Shift_TwaBuf	req[1-9-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_alarm1h_op	req[1-14-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_Calc_AverageStelTwa	req[1-9-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_Calc_Denst	req[1-1-1-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_Calc_DispDenstData	req[1-1-4-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_Calc_Peak	req[1-6-1-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_stel_op	req[1-10-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc.c	m_twa_op	req[1-12-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	GasCalcEC_Back_Linearize_Denst	req[1-3-2-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	GasCalcEC_Calc_Tracking	req[1-4-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	GasCalcEC_Get_ECChkStartFlg	req[3-2-6-19]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	GasCalcEC_Linearize_Denst	req[1-3-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	GasCalcEC_Reset_Tracking	req[1-4-1-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	GasCalcEC_Set_ECChkStartFlg	req[3-2-6-20]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	m_Get_PlusTrackingDenst	req[1-4-1-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec.c	m_Get_PlusTrackingFlg	req[1-4-1-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	GasCalcECDenst_AirCal_Denst	req[5-1-1-6]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	GasCalcECDenst_AutoCal_Denst	req[5-4-1-14]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	GasCalcECDenst_Calc_Denst	req[1-2-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	GasCalcECDenst_Set_SpanMargine	req[11-3-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	m_Calc_ECSpanTempComp	req[1-2-2-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	m_Calc_ECZeroTempComp	req[1-2-2-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	m_Get_SensorAD	req[1-2-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	m_Get_SpanAD	req[1-2-1-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	m_Get_SuppressNum	req[1-5-1-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_denst.c	m_Get_ExportControlsNum	req[1-5-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	GasCalcECH2CancelCO_Calc_Denst	req[1-2-1-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	GasCalcECH2CancelCO_Set_AirCal	req[5-1-1-7]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	GasCalcECH2CancelCO_Set_AutoCal	req[5-4-1-15]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	GasCalcECH2CancelCO_Set_SpanMargine	req[11-3-1-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Calc_InverseMatrix	req[1-2-1-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Calc_TempComp	req[1-2-2-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Chk_PresenceOfInverseMatrix	req[1-2-1-6]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Correction_DenstValComp	req[1-2-1-7]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Correction_ZeroComp	req[1-2-2-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Get_AveAD	req[13-6-2-16]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_h2_cancel_co.c	m_Linearaise_H2CancelCO	req[1-3-1-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	GasCalcECO2_10msec_interrupt	req[1-2-5-1]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	GasCalcECO2_AirCal_O2Denst	req[5-1-1-8]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	GasCalcECO2_AutoCal_O2Denst	req[5-4-1-16]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	GasCalcECO2_Calc_O2Denst	req[1-2-1-8]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	GasCalcECO2_Set_SpanMargine	req[11-3-1-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	m_Calc_O2TempComp	req[1-2-2-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	m_Chk_PressureDenst	req[1-2-5-2]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	m_ForcedStop_Pressure	req[1-2-5-3]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	m_Peak_Pressure	req[1-2-5-4]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	m_Start_Pressure	req[1-2-5-5]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_ec_o2.c	m_Stop_Pressure	req[1-2-5-6]	Ikarashi	2021/6/28	Taki	2021/6/28	Ikarashi	OK	
gas_calc_nc.c	GasCalcNC_10msec_interrupt	req[1-1-1-4]	Ikarashi	2021/6/29	Taki	2021/6/29	Ikarashi	OK	
gas_calc_nc.c	GasCalcNC_AirCal_HCDenst	req[5-1-1-9]	Ikarashi	2021/6/29	Taki	2021/6/29	Ikarashi	OK	
gas_calc_nc.c	GasCalcNC_AutoCal_HCDenst	req[5-4-1-17]	Ikarashi	2021/6/29	Taki	2021/6/29	Ikarashi	OK	
gas_calc_nc.c	GasCalcNC_Calc_NCDenst	req[1-1-1-5]	Ikarashi	2021/6/29	Taki	2021/6/29	Ikarashi	OK	
gas_calc_nc.c	GasCalcNC_Chk_OffNcPower	req[14-15-1-1]	Ikarashi	2021/6/29	Taki	2021/6/29	Ikarashi	OK	
gas_calc_nc.c	GasCalcNC_Chk_SensorFail	req[3-2-5-5]	Ikarashi	2021/6/29	Taki	2021/6/29	Ikarashi	OK	

gas_calc_nc.c	GasCalcNC_Get_LongEnergyFlg	req[11-2-1-1]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	GasCalcNC_Get_NCProtectionCount	req[1-1-7-3]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	GasCalcNC_Get_NCProtectionFlg	req[1-1-7-4]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	GasCalcNC_Get_NCWarmupFlg	req[1-1-1-8]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	GasCalcNC_Set_NCProtectionFlgOff	req[1-1-7-5]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	GasCalcNC_Set_SpanMargine	req[11-3-1-5]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	GasCalcNC_Start_ChkFlg	req[3-2-5-4]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Calc_NC_DiffMode	req[1-4-1-1]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Calc_AbsoluteHumidity	req[1-1-3-1]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Calc_NC_OutPutMode	req[1-1-1-6]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Change_LinearRaiseBackOp	req[1-3-2-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Change_LinearRaiseOp	req[1-3-1-3]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Get_NC_100msTempComp	req[1-1-2-3]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Get_NC_SpanHumComp	req[1-1-3-3]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Get_NC_SpanTempComp	req[1-1-2-1]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Get_NC_ZeroHumComp	req[1-1-3-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Get_NC_ZeroTempComp	req[1-1-2-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Set_NC_Denst	req[1-1-1-7]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_nc.c	m_Set_NC_OffMode	req[1-1-7-6]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_tbl.c	GasCalcTbl_Get_RomNum	req[10-5-1-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_calc_tbl.c	GasCalcTbl_Get_SumNum	req[10-5-2-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_air.c	GasFuncAir_Calc_1sec_AirCal	req[5-1-1-11]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_air.c	GasFuncAir_Calc_Get_AirCalFlg	req[5-1-1-12]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_air.c	GasFuncAir_Calc_Set_AirCalFlg	req[5-1-1-13]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_air.c	GasFuncAir_Disp_Get_AirCalFlg	req[5-1-1-14]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_air.c	GasFuncAir_Disp_Set_AirCalFlg	req[5-1-1-15]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_air.c	GasFuncAir_Set_AirCalErrorOff	req[5-1-1-10]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Calc_1sec_AutoCal	req[5-4-1-24]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Calc_Get_AutoCalFlg	req[5-4-1-25]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Calc_Set_AutoCalFlg	req[5-4-1-26]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Calc_Set_BumpCalFlg	req[4-1-2-12]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Calc_Set_ComAutoCalFlg	req[5-4-1-18]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Get_AutoCalGroup	req[5-4-1-19]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Get_AutoCalSDMSelear	req[5-4-1-20]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Set_AutoCalErrorOff	req[5-4-1-21]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Set_AutoCalGroup	req[5-4-1-22]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Set_AutoCalSDMSelear	req[5-4-1-23]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	GasFuncAutoCal_Set_ComAutoCalDenst	req[5-4-2-8]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
gas_func_autocal.c	m_Set_AutoCalDenst	req[5-4-2-9]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Flag_Acquisition_100msec	req[13-4-1-6]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Flag_Acquisition_250msec	req[13-4-1-7]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Flag_Set_100msec	req[13-4-1-8]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Flag_Set_250msec	req[13-4-1-9]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Int_CMT_Start	req[13-4-1-10]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Int_CMT_Stop	req[13-4-1-11]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
interval_timer.c	IntervalTimer_Int_Timer_CMT0_Op	req[13-4-1-12]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Chk_DispPowerOff	req[13-4-1-13]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Chk_KeyPowerOff	req[13-4-1-14]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Get_ForcedPowerOff	req[6-11-2-12]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Get_KeyEvent	req[14-10-2-1]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Get_KeyPort	req[14-10-1-1]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Get_KeyPowerOff	req[6-11-2-15]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Get_ReleaseKeyFlg	req[14-10-2-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Get_TimeNum	req[14-10-2-3]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Make_Event	req[14-10-2-4]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_no_op_time_set	req[14-10-2-5]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Reset_KeyTime	req[6-11-2-16]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Scan_KeyPort	req[14-10-1-2]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_Set_ForcedPowerOff	req[6-11-2-17]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	KeyEvent_time_set	req[14-10-2-6]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	m_Get_KeyIo	req[14-10-1-3]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
key_event.c	m_key_set	req[14-10-2-7]	lkarashi	2021/6/29	Taki	2021/6/29	lkarashi	OK	
lcd_main.c	LCDMAIN_10msec_interrupt	req[14-5-1-7]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Clr_DispData	req[6-1-1-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_BackLight	req[6-1-1-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_BackLightTimer	req[6-1-1-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_BattIconDisp	req[14-5-2-23]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_BumpChkIconDisp	req[14-5-2-28]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_Com_Discon_count	req[6-1-1-1]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_set_data	req[14-5-2-22]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_HartIconDisp	req[14-5-2-24]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_PumpIconDisp	req[14-5-2-29]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_LongIconDisp	req[14-5-2-27]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_ModelconDisp	req[14-5-2-26]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_OffBackLight	req[6-1-1-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_OnSdmBackLight	req[6-1-1-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_Set_TimeIconDisp	req[14-5-2-25]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_start	req[14-5-1-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_main.c	LCDMAIN_stop	req[14-5-1-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_all_display_ram	req[14-5-2-1]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_clr_display_ram	req[14-5-2-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_int_10msec	req[14-5-1-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_Set_BackLight	req[14-6-1-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_set_data	req[14-5-2-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_Set_DispDataUpDATA	req[14-5-2-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	

lcd_normal_drv.c	LCDNORMAL_Set_Display_point	req[14-5-1-1]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_start	req[14-5-1-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	LCDNORMAL_stop	req[14-5-1-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_drv.c	m_write_normal_disp_data	req[14-5-2-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_font.c	LCDNormalFont_seg_conv7	req[14-5-2-7]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_font.c	LCDNormalFont_seg_conv14	req[14-5-2-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_7segGasNameDisp	req[14-5-2-8]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_AlarmCommentDisp	req[14-5-2-9]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_BattIconDisp	req[14-5-2-10]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_BumpChkIconDisp	req[14-5-2-20]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_CharacterDisp	req[14-5-2-11]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_DenstDisp	req[14-5-2-12]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_GasNameDisp	req[14-5-2-13]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_HartIconDisp	req[14-5-2-14]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_LongIconDisp	req[14-5-2-15]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_MeasurementDisp	req[14-5-2-16]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_ModeIconDisp	req[14-5-2-17]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_NoALARMIconDisp	req[14-5-2-18]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_TimeIconDisp	req[14-5-2-19]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lcd_normal_show.c	LCDNormalShow_Set_PumpIconDisp	req[14-5-2-21]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_Control_Led_Port	req[14-6-1-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_10msec_Interrupt	req[14-7-1-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_Get_LightOnOffStatus	req[14-7-1-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_Set_InitRAM	req[14-7-1-1]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_Set_LightOff	req[14-7-1-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_Set_LightOnTime	req[14-7-1-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	LED_Test_Command	req[14-7-1-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
led.c	m_Set_LedLightPort	req[14-7-1-7]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Chk_1secLoggerRun	req[8-1-2-50]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Chk_LoggerDataClear	req[8-2-1-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Chk_LoggerPowerClear	req[8-1-1-9]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Clear_LoggerDataClear	req[8-2-1-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Clear_LoggerPowerClear	req[8-1-1-10]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Clear_PeakClear	req[8-1-2-45]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Get_FastBumpDenst	req[8-1-6-11]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Get_LoggerRun	req[8-1-2-51]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Init_Logger	req[8-2-1-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Periodic_1secConfirm	req[8-1-2-46]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_AirEvent	req[8-1-6-7]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_BUMPEvent	req[8-1-6-8]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_FastBumpDenst	req[8-1-6-10]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_LoggerDefault	req[6-6-1-15]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_LunchFlg	req[8-1-2-47]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_PowerOff	req[8-1-1-11]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_PowerOn	req[8-1-1-12]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_SettingCheck	req[8-1-7-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_SpanEvent	req[8-1-6-9]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_SnapShot	req[8-1-8-19]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_StartLoggerInterval	req[8-1-2-48]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_StopLoggerInterval	req[8-1-2-49]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Set_TestData	req[8-1-9-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Start_LoggerDataClear	req[8-2-1-7]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
logger_GXForce.c	LoggerGXForce_Start_LoggerPowerClear	req[8-1-1-13]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lvd.c	Lvd_Cyclic_250msecProcess	req[13-13-1-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lvd.c	Lvd_Init_Setting	req[13-13-1-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
lvd.c	Lvd_Stop_Setting	req[13-13-1-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_1000msec_Task_A	req[13-4-2-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_1000msec_Task_B	req[13-4-2-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_1000msec_Task_C	req[13-4-2-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_1000msec_Task_D	req[13-4-2-5]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_250msec_TaskA	req[13-4-2-6]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_250msec_TaskB	req[13-4-2-7]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	m_Every_Time_Task	req[13-4-2-8]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Get_PowerOffFlg	req[6-11-2-21]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Get_RomNum	req[13-2-1-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Get_SumNum	req[13-2-1-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	Main_Process	req[13-4-2-1]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Set_ForcedPowerOffFlg	req[6-11-2-20]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	Main_Set_Power_OFF	req[6-11-2-18]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	Main_Set_Power_ON	req[6-11-1-1]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Set_PowerOffFlg	req[6-11-2-19]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Set_UpdateFlg	req[7-1-2-2]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Get_UpdateModeFlg	req[7-1-2-4]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
main.c (MAIN)	MAIN_Set_UpdateModeFlg	req[7-1-2-3]	lkarashi	2021/6/30	Taki	2021/6/30	lkarashi	OK	
Port.c	Port_Get_UsbPort_isOn	req[13-1-1-5]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Port.c	Port_Init_Setting	req[13-1-1-4]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Pressure.c	Pressure_Change_Device0OutPut	req[14-13-1-33]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Pressure.c	Pressure_Get_FlowValue	req[14-13-1-34]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Pressure.c	Pressure_Init_Process	req[14-13-1-30]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Pressure.c	Pressure_Start_Device	req[14-13-1-31]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Pressure.c	Pressure_Stop_Device	req[14-13-1-32]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
Pressure.c	f_Check_DacOutPutPer	req[14-13-1-35]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	
pump.c	Pump_Control_PumpMove	req[14-12-1-2]	lkarashi	2021/7/1	Taki	2021/7/1	lkarashi	OK	

pump.c	Pump_Get_Phase_isNormalOrEnd	req[14-12-1-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
pump.c	Pump_Init_Process	req[14-12-1-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
pump.c	f_Drive_Set	req[14-12-1-8]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
pump.c	f_LoHi_Set	req[14-12-1-7]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
pump.c	f_Phase_PumpMove	req[14-12-1-4]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
pump.c	f_Power_Set	req[14-12-1-6]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
pump.c	f_Set_PumpOperation	req[14-12-1-5]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Check_FlowTest	req[14-13-1-22]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Check_PumpTest	req[14-13-1-21]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Check_PurgeActive	req[14-13-1-17]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Control_DriveMode	req[14-13-1-16]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Cyclic_250msec	req[14-13-1-15]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Default_Device	req[14-13-1-4]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_BoostCheck_isRetry	req[14-13-1-18]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_DropOutPut	req[14-13-1-8]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_FlowOutPut	req[14-13-1-9]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_LoHiFlag_isHi	req[14-13-1-12]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_MotionFlag_isMove	req[14-13-1-14]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_PumpFailPressValue	req[14-13-1-20]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_PumpOffPressValue	req[14-13-1-19]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Get_SettingData	req[14-13-1-5]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Init_Process	req[14-13-1-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Set_FlowOutPut	req[14-13-1-10]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Set_LoHiRequest_isHi	req[14-13-1-11]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Set_MotionRequest_isMove	req[14-13-1-13]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Set_SettingData	req[14-13-1-6]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Set_SettingDefaultData	req[14-13-1-7]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Start_Device	req[14-13-1-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	PumpFlow_Stop_Device	req[14-13-1-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Check_FlowFail	req[14-13-1-25]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Check_FlowFailInPsSensor	req[14-13-1-26]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Check_PumpFail	req[14-13-1-24]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Check_PurgeLimitCheck_isNeed	req[14-13-1-29]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Get_OffFlowValue	req[14-13-1-23]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Set_FlowFail	req[14-13-1-28]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
PumpFlow.c	f_Set_PumpFail	req[14-13-1-27]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_cgc.c	R_CGC_Create	req[13-12-1-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_cmt.c	R_CMT0_Create	req[13-4-1-13]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_cmt.c	R_CMT0_Start	req[13-4-1-14]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_cmt.c	R_CMT0_Stop	req[13-4-1-15]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_cmt_user.c	r_cmt_cmi0_interrupt	req[13-4-1-16]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_hardware_setup.c	HardwareSetup	req[13-12-1-4]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_hardware_setup.c	R_Systeminit	req[13-12-1-5]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_access_exception	req[13-4-1-17]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_brk_exception	req[13-4-1-18]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_floatingpoint_exception	req[13-4-1-19]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_nmi_exception	req[13-4-1-20]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_privileged_exception	req[13-4-1-21]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_reserved_exception	req[13-4-1-22]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_intprg.c	r_undefined_exception	req[13-4-1-23]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_lvd.c	R_LVD1_Create	req[13-13-1-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_lvd.c	R_LVD1_Start	req[13-13-1-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_lvd.c	R_LVD1_Stop	req[13-13-1-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_main.c	main(MAIN)	req[13-12-1-6]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_main.c	R_MAIN_UserInit	req[13-12-1-8]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_mtu2.c	R_MTU2_C2_Start	req[13-5-1-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_mtu2.c	R_MTU2_C2_Stop	req[13-5-1-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_mtu2.c	R_MTU2_Create	req[13-5-1-4]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_mtu2_user.c	r_mtu2_tgia2_interrupt	req[13-5-1-5]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_port.c	R_PORT_Create	req[13-1-1-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_r12da.c	R_R12DA_Create	req[13-14-1-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_r12da.c	R_R12DA0_Set_ConversionValue	req[13-14-1-4]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_r12da.c	R_R12DA0_Start	req[13-14-1-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_r12da.c	R_R12DA0_Stop	req[13-14-1-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_resetprg.c	PowerON_Reset_PC	req[13-12-1-10]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi.c	R_RSPI0_Create	req[13-8-1-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi.c	R_RSPI0_Send_Receive	req[13-8-2-1]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi.c	R_RSPI0_Send_Receive_Copy	req[13-8-2-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi.c	R_RSPI0_Start	req[13-8-1-2]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi.c	R_RSPI0_Stop	req[13-8-1-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_callback_error	req[13-8-2-3]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_callback_receiveend	req[13-8-2-4]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_callback_transmitend	req[13-8-2-5]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_error_interrupt	req[13-8-2-6]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_idle_interrupt	req[13-8-2-7]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_receive_interrupt	req[13-8-2-8]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_rspi_user.c	r_rspi0_transmit_interrupt	req[13-8-2-9]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_s12ad.c	R_S12AD_Create	req[13-6-1-16]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_s12ad.c	R_S12AD_Get_ValueResult	req[13-6-2-18]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_s12ad.c	R_S12AD_Start	req[13-6-1-17]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_s12ad.c	R_S12AD_Stop	req[13-6-1-18]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_s12ad_user.c	r_s12ad_interrupt	req[13-6-2-17]	Ikarashi	2021/7/1	Taki	2021/7/1	Ikarashi	OK	
r_cg_sci.c	R_SCI5_Create	req[13-7-1-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI5_Serial_Receive	req[13-7-3-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI5_Serial_Send	req[13-7-2-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	

r_cg_sci.c	R_SCI5_Start	req[13-7-1-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI5_Stop	req[13-7-1-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_Create	req[13-9-1-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_IIC_Master_Receive	req[13-9-3-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_IIC_Master_Send	req[13-9-2-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_IIC_StartCondition	req[13-9-2-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_IIC_StopCondition	req[13-9-2-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_Start	req[13-9-1-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI6_Stop	req[13-9-1-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI9_Create	req[13-7-1-8]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI9_Serial_Receive	req[13-7-3-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI9_Serial_Send	req[13-7-2-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI9_Start	req[13-7-1-9]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci.c	R_SCI9_Stop	req[13-7-1-10]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_callback_receiveend	req[13-7-3-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_callback_receiveerror	req[13-7-3-8]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_callback_transmitend	req[13-7-2-8]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_receive_interrupt	req[13-7-3-9]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_receiveerror_interrupt	req[13-7-3-10]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_transmit_interrupt	req[13-7-2-9]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci5_transmitend_interrupt	req[13-7-2-10]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci6_callback_receiveend	req[13-9-3-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci6_callback_transmitend	req[13-9-2-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci6_receive_interrupt	req[13-9-3-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci6_transmit_interrupt	req[13-9-2-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci6_transmitend_interrupt	req[13-9-2-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_callback_receiveend	req[13-7-3-11]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_callback_receiveerror	req[13-7-3-12]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_callback_transmitend	req[13-7-2-11]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_receive_interrupt	req[13-7-3-13]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_receiveerror_interrupt	req[13-7-3-14]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_transmit_interrupt	req[13-7-2-12]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_sci_user.c	r_sci9_transmitend_interrupt	req[13-7-2-13]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_cg_wdt.c	R_WDT_Restart	req[13-10-2-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_spi.c	m_Wait_Spi_Send_Time	req[13-8-2-10]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_spi.c	RSPI_count_down	req[13-8-1-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_spi.c	RSPI_Int_Start	req[13-8-1-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_spi.c	RSPI_Int_Stop	req[13-8-1-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_spi.c	RSPI_Send_Receive	req[13-8-3-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
r_spi.c	RSPI_write_chk_off	req[13-8-1-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
ref_volt.c	RefVolt_Cyclic_250msec	req[3-2-3-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
ref_volt.c	RefVolt_Init_HcvCheckDisableFlg	req[3-2-3-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
ref_volt.c	RefVolt_Set_HcvCheckDisableFlg	req[3-2-3-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
ref_volt.c	f_Set_RefVoltFail	req[3-2-3-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
ref_volt.c	f_Check_RefVoltFail	req[3-2-3-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_10msec_interrupt	req[13-7-1-11]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Calc_H2CancelCOAdAverage	req[14-16-1-9]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Calc_H2CancelCOAdSUB	req[14-16-1-10]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Calc_Temp	req[14-11-1-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Chk_TempError	req[3-2-4-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Command_250_Loop	req[13-7-2-14]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Command_Control	req[13-7-3-15]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Command_ReSet	req[13-7-1-12]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Get_Data_point	req[14-3-1-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Get_MainData_point	req[14-3-1-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Set_Control_Status	req[13-7-3-16]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Set_Gas_Status	req[5-8-2-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Set_ROM_NUM	req[13-2-1-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Set_ROM_SUM	req[13-2-1-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Set_ROM_VER	req[13-2-1-8]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_control.c	RL78I1EControl_Start_ChkFlg	req[3-2-4-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_10msec_interrupt	req[13-2-1-9]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_Chk_4Count	req[13-2-1-10]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_Get_PhaseFlg	req[13-2-1-11]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_Set_PhaseEnd	req[13-2-1-12]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_Set_PhaseOFF	req[13-2-1-13]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_Set_PhaseWait	req[13-2-1-14]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rl78i1e_sum_chk.c	RL78I1ESumChk_Set_Start	req[13-2-1-15]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_data_check	req[3-2-2-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Get_Oscillation_Stop	req[14-3-1-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Get_RegisterData	req[14-3-1-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Read_DateAndTime	req[14-3-3-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Rtc_Writ_VerifyCheck	req[14-3-2-1]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Set_RegisterData	req[14-3-1-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Writ_CircuitSetting	req[14-3-1-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	m_Writ_DateAndTime	req[14-3-2-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	RTC_DateAndTime_Config	req[14-3-2-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	RTC_DateAndTime_Set_OnOffFlg	req[14-3-1-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	RTC_Get_Data	req[14-3-3-2]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	RTC_Read_DateAndTime	req[14-3-3-3]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc.c	RTC_Start_Up	req[14-3-1-8]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc_time.c	m_Chg_Time	req[14-3-2-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc_time.c	m_Chg_Time_Value	req[14-3-2-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc_time.c	RTCTime_Chg_Day	req[14-3-2-6]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc_time.c	RTCTime_Chk_Day	req[14-3-2-7]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	

rtc_time.c	RTCTime_Chk_LeapYear	req[3-2-2-4]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
rtc_time.c	RTCTime_Chk_Time_Error	req[3-2-2-5]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	SCI5UartMain_Comu_Op	req[7-1-1-120]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Get_Rxd_Data	req[13-7-3-38]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Get_TransmitEnable	req[13-7-2-56]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Init_Setting	req[13-7-1-15]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	m_Txd_Busy	req[13-7-2-59]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Int_Sci5_Start	req[13-7-1-18]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Start_Driver	req[13-7-1-16]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Int_Sci5_Stop	req[13-7-1-19]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Interrupt_Sci5Error	req[13-7-3-39]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Interrupt_Sci5SendEnd	req[13-7-2-58]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Set_Rxd_Next_Buf	req[13-7-1-17]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci5_uart.c	UartSci5_Send_Data	req[13-7-2-57]	Ikarashi	2021/7/2	Taki	2021/7/2	Ikarashi	OK	
sci6_iic.c	m_set_send_receive_delay	req[13-9-3-4]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	m_set_send_receive_flg	req[13-9-3-5]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	m_set_send_receive_init_flg	req[13-9-3-6]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_count_delay_time	req[13-9-3-7]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_Int_Sci6_Receive	req[13-9-3-8]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_Int_Sci6_Send	req[13-9-2-7]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_Int_Sci6_Send_Init_LCD	req[13-9-2-8]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_Int_Sci6_Send_LCD	req[13-9-2-9]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_Int_Sci6_Start	req[13-9-1-4]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_Int_Sci6_Stop	req[13-9-1-5]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci6_iic.c	SCI6IIC_set_send_receive_end_flg	req[13-9-2-10]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	m_SCI9_Code_DU	req[13-7-3-17]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	m_SCI9_Code_HA	req[13-7-3-18]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	m_SCI9_Code_RC	req[13-7-3-19]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	m_SCI9_Code_VP	req[13-7-3-20]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	m_SCI9_Code_VS	req[13-7-3-21]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	m_Set_Start_Command	req[13-7-2-15]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Command_Op	req[13-7-2-16]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Get_Cmd_NUM	req[13-7-2-17]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Get_Comnd_Data	req[13-7-2-18]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Get_Comnd_Pointer	req[13-7-2-19]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_DUW	req[13-7-2-20]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_HAW	req[13-7-2-21]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_RCR	req[13-7-2-22]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_VPR	req[13-7-2-23]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_VSC	req[13-7-2-24]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_VSR	req[13-7-2-25]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_command.c	SCI9UartCommand_Send_VSW	req[13-7-2-26]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	m_Comand_Chk	req[13-7-3-22]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	m_Comu_Op_Sub	req[13-7-2-27]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	m_Rxd_Sum_Chk	req[13-7-3-27]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	m_Txd_Busy	req[13-7-2-28]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	m_Txd_Start	req[13-7-2-29]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	m_Txd_Sum_Calc	req[13-7-2-30]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Comu_Op	req[13-7-3-24]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Err_Op	req[13-7-2-31]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Get_Rxd_Data	req[13-7-3-25]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Head_Set	req[13-7-2-32]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Int_Sci9_Start	req[13-7-1-13]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UARTMain_Int_Sci9_Stop	req[13-7-1-14]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Set_Rxd_Next_Buf	req[13-7-3-26]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_tx_nbyte	req[13-7-2-33]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
sci9_uart_main.c	SCI9UartMain_Txd_Buf_Pointre	req[13-7-2-34]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdBaudRateSet	req[7-1-2-75]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdBlockBlankCheck	req[7-1-2-76]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdBlockErase	req[7-1-2-77]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdCheckSum	req[7-1-2-78]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdProgramming_step1	req[7-1-2-79]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdProgramming_step2	req[7-1-2-80]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdProgramming_step3	req[7-1-2-81]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_cmdReset	req[7-1-2-82]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_csum_ui16	req[7-1-2-83]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_csum_ui8	req[7-1-2-84]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_eraseUntiLastBlock	req[7-1-2-85]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_isRxACK	req[7-1-2-86]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_reqRx	req[7-1-2-87]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_reqRx_DataFram	req[7-1-2-88]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_reqTx	req[7-1-2-89]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_reqTx_CmdFrame	req[7-1-2-90]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_reqTx_DataFrame	req[7-1-2-91]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_restartComDrv	req[7-1-2-92]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_sendComMethodData	req[7-1-2-93]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_updateWaitTime	req[7-1-2-94]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_wait	req[7-1-2-95]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	m_waitForRxComplete	req[7-1-2-96]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	RL78FProg_execTargetPowerOff	req[7-1-2-97]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	RL78FProg_rewriteBlock	req[7-1-2-98]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	RL78FProg_setStartBlock	req[7-1-2-99]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_RL78FProg.c	RL78FProg_startRewritingProg	req[7-1-2-100]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_prog.c	SelfSubProg_Get_10msecChgFlg	req[7-1-2-69]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_prog.c	SelfSubProg_Get_ModeOnFlg	req[7-1-2-74]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	

self_sub_prog.c	SelfSubProg_Int_10msec	req[7-1-2-70]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_prog.c	SelfSubProg_Main_RewriteProcessing	req[7-1-2-71]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_prog.c	SelfSubProg_Set_ModeOnFlg	req[7-1-2-73]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
self_sub_prog.c	SelfSubProg_Set_RewiteStart	req[7-1-2-72]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Asc_BinL	req[13-11-1-17]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Asc_Dec	req[13-11-1-18]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Asc_Hex	req[7-1-2-30]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Bin_Asc	req[13-11-1-11]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_CHR_Chg_OnOff	req[13-11-1-12]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Comp_Pnt	req[13-11-1-13]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Copy_Pnt	req[13-11-1-14]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Copy_Reverse_Pnt	req[13-11-1-23]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Copy_SHORT_CHR	req[13-11-1-24]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Copy_SHORT_Pnt	req[13-11-1-25]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Disp_OnOff_Menu	req[13-11-1-15]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Dump_Hex_Asc	req[13-11-1-26]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Hex_Asc	req[7-1-2-29]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_Hex_ShortAsc	req[13-11-1-28]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_OneCount	req[13-11-1-29]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_set_Dec_ChrAscii	req[13-11-1-30]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_set_Dec_ChrSymbolAscii	req[13-11-1-31]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_set_Dec_DenstAscii	req[13-11-1-32]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_set_Dec_DenstShortAscii	req[13-11-1-33]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
tool.c (MAIN)	Tool_set_Dec_ShortAscii	req[13-11-1-34]	Ikarashi	2021/7/5	Taki	2021/7/5	Ikarashi	OK	
usb_com.c	USBCom_Cmd_Proc	req[7-1-1-14]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_AsciiDenst_proc	req[13-11-1-4]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_Calc_PCSum	req[7-1-1-15]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_Chk_AsciiOnOffDec	req[13-11-1-5]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_Chk_DecOnOffAscii	req[13-11-1-6]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_Cmp_SUM	req[7-1-1-17]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_DenstAscii_proc	req[13-11-1-7]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_Get_UnitPnt	req[13-11-1-8]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_90_proc	req[7-1-1-107]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_93_proc	req[7-1-1-104]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_94_proc	req[7-1-1-105]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_A7_proc	req[7-1-1-21]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_AF_proc	req[7-1-1-52]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_AJ_proc	req[7-1-1-96]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_AM_proc	req[7-1-1-51]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_AP_proc	req[7-1-1-50]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_B0_proc	req[7-1-1-53]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_B7_proc	req[7-1-1-24]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_B8_proc	req[7-1-1-23]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_B9_proc	req[7-1-1-22]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BJ_proc	req[7-1-1-62]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BK_proc	req[7-1-1-61]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BL_proc	req[7-1-1-63]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BO_proc	req[7-1-1-30]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BP_proc	req[7-1-1-108]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BS_proc	req[7-1-1-40]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_BU_proc	req[7-1-1-65]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_CP_proc	req[7-1-1-110]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_D2_proc	req[7-1-1-99]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_D3_proc	req[7-1-1-100]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_DH_proc	req[7-1-1-111]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_DP_proc	req[7-1-1-113]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_DW_proc	req[7-1-1-88]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_EF_proc	req[7-1-1-89]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_EP_proc	req[7-1-1-90]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_EX_proc	req[7-1-1-101]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_FB_proc	req[7-1-1-95]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_G0_proc	req[7-1-1-94]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_GJ_proc	req[7-1-1-18]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_GT_proc	req[7-1-1-45]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_GY_proc	req[7-1-1-46]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_H1_proc	req[7-1-1-29]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_H3_proc	req[7-1-1-31]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_HA_proc	req[7-1-1-114]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_HC_proc	req[7-1-1-80]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_HF_proc	req[7-1-1-109]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_HN_proc	req[7-1-1-37]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_IN_proc	req[7-1-1-36]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_JZ_proc	req[7-1-1-68]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_KW_proc	req[7-1-1-118]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_LB_proc	req[7-1-1-97]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_LI_proc	req[7-1-1-19]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_LJ_proc	req[7-1-1-47]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_LK_proc	req[7-1-1-48]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_LP_proc	req[7-1-1-20]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_LQ_proc	req[7-1-1-27]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_MO_proc	req[7-1-1-75]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_MP_proc	req[7-1-1-76]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_MR_proc	req[7-1-1-64]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_MX_proc	req[7-1-1-55]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	
usb_com.c	m_cmd_MY_proc	req[7-1-1-56]	Ikarashi	2021/7/6	Taki	2021/7/6	Ikarashi	OK	

usb_com.c	m_cmd_MZ_proc	req[7-1-1-57]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_OI_proc	req[7-1-1-72]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_PG_proc	req[7-1-1-77]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_PN_proc	req[7-1-1-49]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_PS_proc	req[7-1-1-32]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_RA_proc	req[7-1-1-116]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_RC_proc	req[7-1-1-115]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_RL_proc	req[7-1-1-119]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_RM_proc	req[7-1-1-117]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_RO_proc	req[7-1-1-98]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_ResponsTerm	req[7-1-1-16]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_S2_proc	req[7-1-1-34]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_S4_proc	req[7-1-1-26]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SA_proc	req[7-1-1-25]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SC_proc	req[7-1-1-58]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SD_proc	req[7-1-1-59]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SG_proc	req[7-1-1-71]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SJ_proc	req[7-1-1-70]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SM_proc	req[7-1-1-60]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SO_proc	req[7-1-1-28]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_SZ_proc	req[7-1-1-33]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_U8_proc	req[7-1-1-69]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_UJ_proc	req[7-1-1-41]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_US_proc	req[7-1-1-43]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_UX_proc	req[7-1-1-87]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_UZ_proc	req[7-1-1-44]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_VL_proc	req[7-1-1-83]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_VP_proc	req[7-1-1-81]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_VQ_proc	req[7-1-1-82]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_VS_proc	req[7-1-1-84]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_VT_proc	req[7-1-1-85]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_VV_proc	req[7-1-1-86]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_W0_proc	req[7-1-1-106]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_W8_proc	req[7-1-1-103]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_W9_proc	req[7-1-1-35]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_WU_proc	req[7-1-1-102]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_X8_proc	req[7-1-1-73]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XA_proc	req[7-1-1-74]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XB_proc	req[7-1-1-54]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XC_proc	req[7-1-1-92]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XH_proc	req[7-1-1-91]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XI_proc	req[7-1-1-66]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XL_proc	req[7-1-1-67]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XP_proc	req[7-1-1-78]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XS_proc	req[7-1-1-39]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XT_proc	req[7-1-1-42]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XW_proc	req[7-1-1-79]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XY_proc	req[7-1-1-93]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	m_cmd_XZ_proc	req[7-1-1-38]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
usb_com.c	str_set_uw_hex	req[13-11-1-9]	lkarashi	2021/7/6	Taki	2021/7/6	lkarashi	OK	
vibration.c	VIBRATION_Control_VIB_Port	req[14-9-1-3]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	m_Get_SensorFailFlg	req[3-2-5-6]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Chk_AirFail	req[5-1-2-2]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Chk_AutoCal_EscNum	req[5-4-2-10]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Chk_DemandZero_onoff	req[5-2-2-5]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Chk_SensorAllInitFailFlg	req[3-2-5-7]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Chk_SpanCalFail	req[5-4-4-2]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_AlarmFlg	req[2-1-5-6]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_AlarmManDownFlg	req[2-1-5-5]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_AlarmPattern	req[2-3-1-5]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_AutoZeroFlg	req[5-3-1-2]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_DispFlicker2	req[13-11-1-41]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_DispFlicker3	req[13-11-1-42]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_DispFlicker4	req[13-11-1-43]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_FailFlg	req[3-2-1-32]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_GasTestFlg	req[4-2-1-5]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_NotManDownFlg	req[2-1-5-7]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_SensorInitFailFlg	req[3-2-5-8]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_ZipcModeNum	req[9-9-1-4]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Get_ZipcNumAddress	req[13-11-1-36]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Int_1sec	req[13-11-1-40]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_mainte_pass_onoff	req[6-8-3-1]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_pass_protect_onoff	req[6-9-1-1]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Reset_InitPossibleResetFailFlg	req[3-1-5-3]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Reset_PossibleResetFailFlg	req[3-1-5-4]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Select_ZipcNumUpDown	req[13-11-1-37]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_Set_GasTestFlg	req[4-2-1-6]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	
zipc_num_set.c	ZIPCNumSet_user_pass_onoff	req[6-8-2-12]	lkarashi	2021/7/7	Taki	2021/7/7	lkarashi	OK	

Unit Test Specification(with Results)

Project name	GX-Force
File name	ad_24bit.c

Ad24bit_Ad_Ave_Op

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<2; luc_i++)		○	○	○	○
	for(luc_j=0; luc_j<M_EC_AVERAGE_NUMBER; luc_j++)		○	○	○	○
	if(++ucs_m_ad_ec_count[luc_i] >= 3U)		-	○	-	○
	else		○	-	○	-
	for(luc_i=0; luc_i<M_TEMP_AVERAGE_NUMBER; luc_i++)		○	○	○	○
	if(ucs_m_nc_migrationflg == M_ON)		○	○	-	○
	if(++ucs_m_nc_update_count > 9U)		○	○	-	-
	else		○	○	-	○
	else		-	-	○	○
	Symbol					
Input	ucs_m_nc_migrationflg	0,1	1	1	0	1
	dbs_m_ec_ad_value[0][0]	0	0	0	0	0
	ucs_m_ad_ec_count[0]	0~4	1	2	0	4
	ucs_m_ad_ec_count[1]	0~2	1	2	0	2
	dbs_m_nc_ad_ave[0][0]	0	0	0	0	0
	dbs_m_nc_ad_ave[1][0]	0	0	0	0	0
	dbs_m_nc_ad_ave[2][0]	0	0	0	0	0
	dbs_m_nc_ad_ave[3][0]	0	0	0	0	0
	dbs_m_nc_ad_ave[4][0]	0	0	0	0	0
	dbs_m_nc_ad_ave[5][0]	0	0	0	0	0
	ucs_m_nc_update_count	0,10	9	10	0	0
	dbs_m_temp_ad_value[0]	0	0	0	0	0
	Ad24bit_bit24AD_Conversion	0	0	0	0	0
Output	ucs_m_nc_migrationflg	0	0	0	0	0
	dbs_m_ad_ec_buf[0][0]	0	0	0	0	0
	ucs_m_ad_ec_count[0]	0~2	2	0	1	0
	ucs_m_ad_ec_count[1]	0~2	2	0	1	0
	fls_m_nc_data[0]	0	0	0	0	0
	fls_m_nc_data[1]	0	0	0	0	0
	fls_m_nc_data[2]	0	0	0	0	0
	fls_m_nc_data[3]	0	0	0	0	0
	fls_m_nc_data[4]	0	0	0	0	0
	fls_m_nc_data[5]	0	0	0	0	0
	ucs_m_nc_update_count	0,1	0	0	0	1
	fls_m_ec_data[0]	0	0	0	0	0
	fls_m_temp_data	0	0	0	0	0
Judgment			OK	OK	OK	OK

Ad24bit_bit24AD_Conversion

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pdb_data	-8388608～8388608	-8388608	8388607	8388608
Output	lfl_ad	-800~800	-800	799.9999	800
Judgment			OK	OK	OK

Ad24bit_Get_NC_Count

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_nc_update_count	1	1
Judgment			OK

Ad24bit_Int_Ad_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(DSADST == 0)		○	-
	else		-	○
	Symbol	Range		
Input	DSADST	0,1	0	1
Output	None	-	-	-
Judgment			OK	OK

Ad24bit_Int_Adc

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_ch <= 0x04)		○	○	-
	if(ECSENSOR_Not_Use_Flg(luc_ch) == M_OFF)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	m_bit24AD_Channel_Check	3,4	3	3	4
	m_bit24AD_Acquisition	0	0	0	0
	m_bit24AD_Chk_minus	0	0	0	0
	ECSENSOR_Not_Use_Flg	0,1	0	1	0
Output	None	-	-	-	-
Judgment			OK	OK	OK

Ad24bit_Nc_Ad_CalcAverage

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0U; luc_i<M_NC_AVERAGE_NUMBER; luc_i++)		○	○	○	○
	for(luc_i=0U; luc_i<M_NC_AVERAGE_NUMBER; luc_i++)		○	○	○	○
	if(ldb_ad_nc_min > dbs_m_nc_ad_backup[puc_element_flg][luc_j])		-	-	-	○
	else		○	○	○	-
	if((puc_element_flg >= 5U) && (ucs_m_nc_migrationflg == M_OFF))		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	puc_element_flg	0~5	5	5	4	0
	Ad24bit_bit24AD_Conversion()	0	0	0	0	0
	dbs_m_nc_ad_ave[0]	1	1	1	1	1
	dbs_m_nc_ad_backup[0]	1	1	1	1	1
	dbs_m_nc_ad_backup[0][0]	0	0	0	0	0
	dbs_m_nc_ad_backup[0][1]	0	0	0	0	0
	dbs_m_nc_ad_backup[0][2]	0	0	0	0	0
	dbs_m_nc_ad_backup[0][3]	0	0	0	0	0
	dbs_m_nc_ad_backup[0][4]	0	0	0	0	0
	dbs_m_nc_ad_value[0]	0	0	0	0	0
	ucs_m_nc_migrationflg	0,1	0	1	1	0
Output	ucs_m_nc_migrationflg	0,1	1	1	0	0
Judgment			OK	OK	OK	OK

Ad24bit_Set_AD

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(puc_ch)		○	○	○	○	○
	case 0		○	-	-	-	-
	case 1		-	○	-	-	-
	case 2		-	-	○	-	-
	case 3		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	puc_ch	0~4	0	1	2	3	4
	fls_m_nc_data[puc_timing]	0	0	0	0	0	0
	fls_m_ec_data[0]	0	0	0	0	0	0
	fls_m_ec_data[1]	0	0	0	0	0	0
	fls_m_temp_data	0	0	0	0	0	0
Output	lfl_data	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

Ad24bit_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i=0; luc_i<7; luc_i++)		○
	for(luc_j=0; luc_j<M_NC_AVERAGE_NUMBER; luc_j++)		○
	for(luc_i=0; luc_i<2; luc_i++)		○
	for(luc_j=0; luc_j<M_EC_AVERAGE_NUMBER; luc_j++)		○
	for(luc_i=0; luc_i<M_TEMP_AVERAGE_NUMBER; luc_i++)		○
	for(luc_i=0; luc_i<6; luc_i++)		○
	for(luc_j=0; luc_j<2; luc_j++)		○
	Symbol	Range	
Input	None		
Output	dbs_m_ec_ad_value[0][0]	0	0
	dbs_m_nc_ad_ave[0][0]	0	0
	dbs_m_nc_ad_backup[0][0]	0	0
	dbs_m_nc_ad_value[0][0]	0	0
	dbs_m_temp_ad_value[0]	0	0
	fls_m_ec_data[0]	0	0
	fls_m_nc_data[0]	0	0
	fls_m_temp_data	0	0
	ucs_m_nc_migrationflg	0	0
	ucs_m_nc_update_count	0	0
	ucs_m_pga[0]	2	2
	ucs_m_pga[1]	0	0
	ucs_m_pga[2]	0	0
	ucs_m_pga[3]	0	0
Judgment			OK

m_bit24AD_Acquisition

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		○
	Symbol	Range	
Input	None	-	-
Output	lul_ad_data	0	0
Judgment			OK

m_bit24AD_Channel_Check

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(DSADCRC & 0xe0)		○	○	○	○	○
	case 0x20		○	-	-	-	-
	case 0x40		-	○	-	-	-
	case 0x60		-	-	○	-	-
	case 0xa0		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	DSADCRC	0x00~0xA0	0x20	0x40	0x60	0xA0	0x00
Output	luc_ch	0x00~0xFF	0x00	0x01	0x02	0x03	0xFF
Judgment			OK	OK	OK	OK	OK

m_bit24AD_Chk_ChangePGA

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(puc_ch != M_TEMP_AD_CH)		○	-	○	○	○	○	○	○
	if(ECSENSOR_Fixed_PGA1_Flg(puc_ch) == M_ON)		-	-	○	-	-	-	-	-
	else		○	-	-	○	○	○	○	○
	if(luc_num != M_PGA1_NUM)		○	-	-	○	○	○	-	-
	if(ulc_m_chg_ad_max < pul_ad)		-	-	-	-	○	○	-	-
	else		○	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	○	○
	if(luc_num != M_PGA16_NUM)		○	-	-	○	○	○	○	○
	if(ulc_m_chg_ad_min> pul_ad)		-	-	-	○	-	-	○	-
	else		○	-	-	-	○	○	-	○
	else		-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-
	Symbol	Range								
Input	puc_ch	0~4	4	3	4	0	0	0	0	0
	pul_ad	3145727~6710887	6710887	6710887	6710887	6710887	6710887	6710886	3145727	3145729
	ucs_m_pga[0]	0,1	1	1	1	1	0	1	0	0
	ECSENSOR_Fixed_PGA1_Flg	1,2	2	2	1	2	2	2	2	2
Output	ucs_m_pga[puc_ch]	0,1	1	1	1	0	0	1	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

m_bit24AD_Chk_minus

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(((*pulp_ad_data) & M_PGA_SIGN_FLG) != 0)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	*pulp_ad_data	0~16777216	8388608	0	16777215	16777216
Output	lsl_data	8388608~16777216	-8388608	0	-1	16777216
Judgment			OK	OK	OK	OK

m_bit24AD_Set_PGARegister

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		○	○	○
	Symbol	Range			
Input	ucs_m_pga[0]	2~5	2	4	5
	ucs_m_pga[1]	0~5	0	4	5
	ucs_m_pga[2]	0~5	0	4	5
	ucc_m_pga_reg_table[0]	0x40~0x48	0x40	0x47	0x48
	ucc_m_pga_reg_table[2]	0x42~0x48	0x42	0x47	0x48
Output	PGA0CTL0	1~71	66	71	1
	PGA1CTL0	1~71	64	71	1
	PGA2CTL0	1~71	64	71	1
	PGA4CTL0	64	64	64	64
Judgment			OK	OK	OK

m_bit24AD_To_buffer

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(luc_count[0] >= M_NC_AVERAGE_NUMBER)		○	○	○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-	-	-
	if(luc_count[1] >= M_EC_AVERAGE_NUMBER)		○	○	○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-	-	-
	if(luc_count[2] >= M_EC_AVERAGE_NUMBER)		○	○	○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-	-	-
	if(luc_count[3] >= M_EC_AVERAGE_NUMBER)		○	○	○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-	-	-
	switch(puc_ch)		○	○	○	○	○	○	○	○	○
	case 0x00		○	-	-	-	-	-	-	○	-
	case 0x01		-	○	-	-	-	-	-	-	-
	case 0x02		-	-	○	-	-	-	-	-	-
	case 0x03		-	-	-	○	-	-	-	-	-
	default		-	-	-	-	○	○	○	-	○
	Symbol	Range									
Input	puc_ch	0~7	0	1	2	3	4	5	6	0	7
	pdb_ad_data	-8388608~8388608	4096	4096	4096	4096	4096	4096	-8388608	8388607	8388608
	luc_count[0]	0x00~0xFF	0x00	0x1E	0x1E	0x1E	0x1E	0x1E	0x00	0x00	0xFF
	luc_count[1]	0x00~0xFF	0x00	0x1E	0x1E	0x1E	0x1E	0x1E	0x00	0x00	0xFF
	luc_count[2]	0x00~0xFF	0x00	0x1E	0x1E	0x1E	0x1E	0x1E	0x00	0x00	0xFF
	luc_count[3]	0x00~0xFF	0x00	0x1E	0x1E	0x1E	0x1E	0x1E	0x00	0x00	0xFF
Output	NcSensor_Reading_Element_Flg	0	0	0	0	0	0	0	0	0	0
	dbs_m_ec_ad_value[0]	0,4096	0	4096	4096	4096	4096	4096	4096	4096	4096
	dbs_m_nc_ad_value[0]	0	0	0	0	0	0	0	0	0	0
	dbs_m_temp_ad_value	0,4096	0	0	0	4096	4096	4096	4096	4096	4096
	luc_count[0]	0,1	1	0	0	0	0	0	0	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	ec_sensor.c

ECSENSOR_Calc_AD_Main

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(uc_m_sub_ec_chk_flg[1] == M_ON)		○	-
	else		-	○
	Symbol	Range		
Input	Ad24bit_bit24AD_Conversion	0	0	0
	uc_m_sub_ec_chk_flg[1]	0,1	0x01	0x00
Output	fl_m_ec_chk_difference[2]	0	0	0
	fl_m_ec_chk_difference[3]	0	0	0
	uc_m_sub_ec_chk_flg[1]	0	0	0
Judgment			OK	OK

ECSENSOR_Fixed_PGA1_Flg

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(puc_ch == 0)		○	-	-	-	-	-
	else		-	○	○	○	○	○
	if((uc_m_ec_chk_flg[1] != M_CHK_STAGE_OFF) && (ucsc_not_use_flg[puc_ch] == M_ON))		-	○	○	○	○	-
	else		-	-	-	-	-	○
	Symbol	Range						
Input	puc_ch	0~4	0	1	1	2	2	4
	uc_m_ec_chk_flg[1]	0~2	0	0	1	0	1	2
	ucsc_not_use_flg[0]	0	0	0	0	0	0	0
	ucsc_not_use_flg[1]	1	1	1	1	1	1	1
	ucsc_not_use_flg[2]	1	1	1	1	1	1	1
	ucsc_not_use_flg[3]	1	1	1	1	1	1	1
	ucsc_not_use_flg[4]	0	0	0	0	0	0	0
Output	luc_flg	0x00,0x01	0x00	0x01	0x01	0x00	0x01	0x00
Judgment			OK	OK	OK	OK	OK	OK

ECSENSOR_Get_Chk_Status

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_m_ec_chk_flg[0] != M_OFF)		○	-	-	○	○
	else		-	○	○	-	-
	if(uc_m_ec_chk_flg[1] != M_OFF)		○	-	○	○	○
	else		-	○	-	-	-
	Symbol	Range					
Input	uc_m_ec_chk_flg[0]	0~15	1	0	0	14	15
	uc_m_ec_chk_flg[1]	0~15	0	0	1	14	15
Output	luc_ans	0x00~0x03	0x01	0x00	0x02	0x03	0x03
Judgment			OK	OK	OK	OK	OK

ECSENSOR_Get_Difference_AD

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_ch	0~4	0	3	4
Output	fl_m_ec_chk_difference[puc_ch]	0	0	0	0
Judgment			OK	OK	OK

ECSENSOR_Interval_Connect_Chk

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<2; luc_i++)		○	○	○	○	○
	if(uc_m_ec_chk_flg[luc_i] != M_CHK_STAGE_OFF)		○	○	-	-	○
	if(uc_m_ec_chk_flg[luc_i] == M_CHK_STAGE_END)		○	-	-	-	-
	else		-	○	-	-	○
	if(uc_m_ec_chk_flg[luc_i] == M_CHK_STAGE_PULSE)		-	○	-	-	-
	else		○	-	-	-	○
	if(us_m_not_use_data_time[luc_i] != 0)		○	-	-	○	○
	else		-	○	○	-	-
	Symbol	Range					
Input	uc_m_ec_chk_flg[0]	0~15	14	2	0	0	15
	us_m_not_use_data_time[0]	0~501	1	0	0	500	501
Output	uc_m_sub_ec_chk_flg[0]	1	1	1	1	1	1
	uc_m_ec_chk_flg[0]	0~16	0	3	0	0	16
	us_m_not_use_data_time[0]	0~500	0	0	0	499	500
Judgment			OK	OK	OK	OK	OK

ECSENSOR_Not_Use_Flg

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(puc_ch == 0)		○	-	-	-	-	○	-
	else		-	○	○	○	○	-	○
	if(us_m_not_use_data_time[1] != 0 && ucsc_not_use_flg[puc_ch] == M_ON)		-	○	○	○	○	-	○
	else		-	-	-	-	-	-	-
	Symbol	Range							
Input	puc_ch	0x00~0xFF	0x00	0x01	0x01	0x02	0x02	0x00	0xFF
	us_m_not_use_data_time[1]	0~501	0	0	1	0	1	500	501
Output	luc_flg	0x00,0x01	0x00	0x00	0x01	0x00	0x01	0x00	0x00
Judgment			OK	OK	OK	OK	OK	OK	OK

ECSENSOR_Set_AD_Connect_Chk

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(uc_m_ec_chk_flg[1] != M_CHK_STAGE_OFF)		○	○	-	○	○	○
	else		-	-	○	-	-	-
	Symbol	Range						
Input	puc_ch	0~2	1	0	0	0	0	2
	pdb_data	-8388608~8388608	4096	4096	4096	-8388608	8388607	8388608
	uc_m_ec_chk_flg[1]	0x00~0xFF	0x00	0x01	0x00	0x0E	0x0F	0xFF
Output	db_m_ec_chk_data[0]	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

ECSENSOR_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i=0; luc_i<2; luc_i++)		○
	for(luc_i=0; luc_i<M_DATA_CH; luc_i++)		○
	for(luc_j=0; luc_j<M_CHK_STAGE_END; luc_j++)		○
	Symbol	Range	
Input	None	-	
Output	uc_m_ec_chk_flg[luc_i]	0	0
	us_m_not_use_data_time[luc_i]	0	0
	uc_m_sub_ec_chk_flg[luc_i]	0	0
	db_m_ec_chk_data[luc_i][luc_j]	0	0
	fl_m_ec_chk_difference[luc_i]	0	0
Judgment			OK

ECSENSOR_Start_ConnectChk

	Responsible	Last update
Coding	lkarashi	2021/6/7
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_m_ec_chk_flg[puc_flg] == M_CHK_STAGE_OFF)		○	○	-	○	-
	if(puc_flg == 0)		○	-	-	-	-
	else		-	○	-	○	-
	else		-	-	○	-	○
	Symbol	Range					
Input	uc_m_ec_chk_flg[puc_flg]	0x00~0x0F	0x00	0x00	0x01	0x0E	0x0F
	puc_flg	0x00~0x02	0x00	0x01	0x01	0x02	0x00
Output	us_m_not_use_data_time[puc_flg]	200	200	200	200	200	200
	uc_m_ec_chk_flg[puc_flg]	0x00~0x0F	0x01	0x00	0x01	0x0E	0x0F
Judgment			OK	OK	OK	OK	OK

m_Set_AD_Buf_Clr

	Responsible	Last update
Coding	lkarashi	2021/6/7
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i=0; luc_i<M_CHK_STAGE_END; luc_i++)		○
	Symbol	Range	
Input	none		
Output	db_m_ec_chk_data[1][luc_i]	0	0
	db_m_ec_chk_data[2][luc_i]	0	0
Judgment			OK

m_Set_EcChk_Port

	Responsible	Last update
Coding	lkarashi	2021/6/7
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(puc_flg == 0)		○	○	-	-	-
	else		-	-	○	○	○
	if(puc_onoff == M_ON)		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	puc_flg	0~2	0	0	1	1	2
	puc_onoff	0~2	1	0	1	0	2
Output	P1	8,9	8	8	9	8	8
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	interval_timer.c

IntervalTimer_Flag_Acquisition_250msec

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_250msec_flag_time	0	0
Judgment			OK

IntervalTimer_Flag_Set_250msec

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~2	0	1	2
Output	uc_m_250msec_flag_time	0~2	0	1	2
Judgment			OK	OK	OK

IntervalTimer_Int_Timer_Op

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(uc_t250m != 0)		-	○	○	○
	if(!uc_t250m)		-	○	-	-
	else		-	-	○	○
	else		○	-	-	-
	Symbol	Range				
Input	uc_t250m	0~25	0	1	25	26
Output	uc_t250m	24,25	25	25	24	25
	uc_m_250msec_flag_time	0,1	0	1	1	1
Judgment			OK	OK	OK	OK

IntervalTimer_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_250msec_flag_time	0	0
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	lvd.c

LVD_Cyclic_250msecProcess_isFault

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(LVIF == 0U)		○	-	○
	else		-	○	-
	if(fuc_lvd_count >= F_LVD_COUNT_MAX)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	fuc_lvd_count	0~21	0	20	21
	LVIF	0,1	0	1	0
Output	fuc_lvd_count	0,20	0	20	0
	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

LVD_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	fuc_lvd_count	0	0
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	main.c

Main_250msecCyclic_24hCheckCount

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luls_24h_count >= M_24H_COUNT_MAX_NUM)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	luls_24h_count	0~345601	0	345600	345601
Output	luls_24h_count	0,1	1	0	0
Judgment			OK	OK	OK

Main_Get_FailStatus

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_status	0	0
Judgment			OK

Main_Get_Gas_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~4	0	3	4
Output	uc_gas_onoff[puc_num]	0	0	0	0
Judgment			OK	OK	OK

Main_Process

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	Yes
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(FuncRamChk_Start_InitRamChk() == M_NG)		○	-	-	-	-
	else		-	○	○	○	○
	while(1U)		○	○	○	○	○
	if(IntervalTimer_Flag_Acquisition_250msec() == TRUE)		○	○	○	-	-
	if(luc_buf == M_NG)		○	-	-	-	-
	else if(luc_buf == M_OK)		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	FuncRamChk_Start_InitRamChk()	0~2	1	0	0	0	2
	ROMCHK_Process()	0~3	1	0	2	0	3
	IntervalTimer_Flag_Acquisition_250msec()	0^2	1	1	1	0	2
	FuncRamChk_Cyclic_250msecProcess()	0,3	3	0	0	0	0
	LVD_Cyclic_250msecProcess_isFault()	0,1	1	0	0	0	0
Output	uc_m_status	0x00,0x18	0x18	0x00	0x00	0x00	0x00
Judgment			OK	OK	OK	OK	OK

Main_Set_Gas_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_num	0~4	0	3	4
	puc_onoff	0~2	0	1	2
Output	uc_gas_onoff	0	0	0	0
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	nc_sensor.c

M_Set_Element_Port

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(luc_num)		○	○	○	○
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		○	-	-	○
	Symbol	Range				
Input	luc_num	0~3	0	1	2	3
Output	P1	8~12	8	10	12	8
Judgment			OK	OK	OK	OK

NcSensor_Get_Long_Energy

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_long_energy_flg	0	0
Judgment			OK

NcSensor_Get_Protection

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_nc_protection_flg	0	0
Judgment			OK

NcSensor_Reading_Element_Flg

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_nc_element_flg	0x05	0x05
Judgment			OK

NcSensor_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_nc_element_flg	0	0
	uc_m_long_energy_flg	0	0
	uc_m_nc_protection_main_flg	0	0
	uc_m_nc_protection_sub_flg	0	0
	uc_m_nc_warmup_main_flg	1	1
	uc_m_nc_protection_permission_flg	0	0
Judgment			OK

NcSensor_Set_Long_Energy

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onoff	0~2	1	0	2
Output	uc_m_long_energy_flg	0,1	1	0	0
Judgment			OK	OK	OK

NcSensor_Set_Protection

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onoff	0~2	1	0	2
Output	uc_m_nc_protection_flg	0,1	1	0	0
Judgment			OK	OK	OK

NcSensor_Set_ProtectionPermissionFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onoff	0~2	1	0	2
Output	uc_m_nc_protection_permission_flg	0,1	1	0	0
Judgment			OK	OK	OK

NcSensor_Set_Warmup

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onoff	0~2	1	0	2
Output	uc_m_nc_warmup_main_flg	0,1	1	0	0
Judgment			OK	OK	OK

NcSensor_Element_Count

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	15
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15
Root	if(luss_count != 0)	○	○	○	○	-	-	-	-	-	-	-	-	-	○	○
	else	-	-	-	-	○	○	○	○	○	○	○	○	○	-	-
	switch(uc_m_nc_element_flg)	-	-	-	-	○	○	○	○	○	○	○	○	○	-	-
	case 0	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case 1	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case 2	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case 3	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case 4	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case 5	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-
	if(uc_m_long_energy_flg == M_ON)	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case 6	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	default	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if((uc_m_nc_element_flg >= 1U) && (uc_m_nc_element_flg <= 6U))	-	-	-	-	○	○	○	○	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-	○	○	○	○
	if((uc_m_nc_element_flg == 6U) && (uc_m_long_energy_flg == M_OFF))	○	○	○	○	-	-	-	-	-	○	-	-	-	-	-
	if(luss_count > M_NC_ELEMENT_COUNT_OFF)	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	○	○	○	○	○	-	○	○	○	○	○
	if((uc_m_nc_protection_main_flg == M_ON) || (uc_m_nc_protection_sub_flg == M_ON))	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	if((uc_m_nc_protection_main_flg == M_ON) && (uc_m_nc_protection_sub_flg == M_ON))	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	○	○	○	○	○	○	○	○	○	○	○	○
	if(Main_Get_Gas_Setting(0) == M_OFF)	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	○	○	○	○	○	○	○	○
	switch(uc_m_nc_element_flg)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2	-	-	-	-	○	○	-	-	-	-	-	○	○	-	-
	case 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 5	-	-	-	-	-	-	○	○	○	-	-	-	-	○	○
	case 6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default	○	○	○	○	-	-	-	-	-	○	○	-	-	-	-
	Symbol	Range														
Input	luss_count	0~1301	400	400	400	400	0	0	0	0	0	0	0	0	1300	1301
	uc_m_nc_element_flg	0~7	6	6	6	6	0	1	2	3	4	5	5	6	7	4	5
	uc_m_long_energy_flg	0~2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2
	uc_m_nc_protection_main_flg	0~2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_nc_protection_sub_flg	0~2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	2
	Main_Get_Gas_Setting	0,1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
Output	uc_m_nc_element_flg	0~6	6	6	6	6	1	2	3	4	5	6	6	0	0	4	5
	uc_m_nc_protection_sub_flg	0~2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2
	luss_count	0~1300	0	0	0	0	40	50	10	40	50	300	1300	10	10	1299	1300
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_NcProtectionFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if(((puc_element_flg >= 1U) && (puc_element_flg <= 2U)) || ((puc_element_flg >= 4U) && (puc_element_flg <= 5U)))	○	○	○	○	-	-	-	○	○	○
	if((ldb_diff_volt > db_nc_over_limit_max[puc_element_flg]) || (ldb_diff_volt < db_nc_over_limit_min[puc_element_flg]))	○	○	○	○	-	-	-	○	○	○
	if((uc_m_nc_protection_main_flg == M_OFF) && (uc_m_nc_warmup_main_flg == M_OFF) && (uc_m_nc_protection_permission_flg == M_ON))	○	-	○	-	-	-	-	-	-	-
	else	-	○	-	○	–	–	–	-	-	-
	else	-	-	-	-	-	-	-	○	○	○
	else	-	-	-	-	○	○	○	-	-	-
	Symbol	Range									
Input	puc_element_flg	0~6	1	2	5	4	3	6	0	1	1	1
	Ad24bit_Get_Nc_DiffVolt()	0	0	0	0	0	0	0	0	0	0	0
	uc_m_nc_protection_main_flg	0,1	0	0	0	0	0	0	0	0	0	1
	uc_m_nc_protection_permission_flg	0,1	1	0	1	0	1	1	1	2	1	1
	uc_m_nc_warmup_main_flg	0,1	0	0	0	0	0	0	0	0	1	0
	dc_nc_over_limit_max[0]	0,1	0	0	0	0	1	1	1	1	1	1
	dc_nc_over_limit_max[1]	0,1	0	0	0	0	1	1	1	1	1	1
	dc_nc_over_limit_max[2]	0,1	0	0	0	0	1	1	1	1	1	1
	dc_nc_over_limit_max[3]	0,1	0	0	0	0	1	1	1	1	1	1
	dc_nc_over_limit_max[4]	0,1	0	0	0	0	1	1	1	1	1	1
	dc_nc_over_limit_max[5]	0,1	0	0	0	0	1	1	1	1	1	1
	dc_nc_over_limit_min[0]	0,1	1	1	1	1	0	0	0	0	0	0
	dc_nc_over_limit_min[1]	0,1	1	1	1	1	0	0	0	0	0	0
	dc_nc_over_limit_min[2]	0,1	1	1	1	1	0	0	0	0	0	0
	dc_nc_over_limit_min[3]	0,1	1	1	1	1	0	0	0	0	0	0
	dc_nc_over_limit_min[4]	0,1	1	1	1	1	0	0	0	0	0	0
	dc_nc_over_limit_min[5]	0,1	1	1	1	1	0	0	0	0	0	0
Output	uc_m_nc_protection_flg	1	1	1	1	1	1	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	ram_chk.c

f_Check_CopyRam

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	if(lus_num == 0U)		-
	else		○
	Symbol	Range	
Input	None	-	-
Output	len_ans	0	0
Judgment			OK

f_Check_MainRam

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_loop=0U; lus_loop<F_CHECK_RAM_BUF_SIZE; lus_loop++)		○	○	○
	if((fuc_num[0] != F_55_BIT) || (fuc_num[1] != F_AA_BIT))		-	-	-
	break		-	-	-
	else		○	○	○
	if(fus_check_main_address >= F_MAIN_RAM_END_ADDRESS)		-	○	○
	if(len_ans == FEN_RAMCHECK_ANS_SUB)		-	○	○
	else		-	-	-
	else		○	-	-
	Symbol	Range			
Input	fus_check_main_address	0xE780~0xFDE1	0xE780	0xFDE0	0xFDE1
	*fucp_check_point	0	0	0	0
	fuc_num[0]	0	0	0	0
	fuc_num[1]	0	0	0	0
	fuc_sub_buf	0	0	0	0
Output	fuc_sub_buf	0	0	0	0
	*fucp_check_point	0	0	0	0
	fuc_num[0]	85	85	85	85
	fuc_num[1]	170	170	170	170
	len_ans	0,2	2	0	0
Judgment			OK	OK	OK

f_Cyclic_CheckProcessMainPhase

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(len_chk == FEN_ROMCHECK_ANS_SUB)		○	-	-
	else if(len_chk == FEN_ROMCHECK_ANS_PASS)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	pen_phase	3,4	3	4	3
	f_Check_MainRam()	0~3	2	0	3
Output	len_ans	3~5	3	4	5
Judgment			OK	OK	OK

f_Cyclic_CheckProcessSubPhase

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(len_chk == FEN_ROMCHECK_ANS_PASS)		○	-
	else		-	○
	Symbol	Range		
Input	f_Check_CopyRam()	0,2	0	2
Output	len_ans	3,5	3	5
Judgment			OK	OK

f_Init_MainRam

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fus_check_main_address	0xE780	0xE780
Judgment			OK

FuncRamChk_Cyclic_250msecProcess

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	f_Cyclic_CheckProcess()	0~3	0	3
Output	len_result	0~3	0	3
Judgment			OK	OK

FuncRamChk_Start_Check

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(fen_phase_flag == FEN_RAMCHECK_PHASE_OFF)		○	-
	else		-	○
	Symbol	Range		
Input	fen_phase_flag	0,5	0	5
Output	fen_phase_flag	1,5	1	5
	fuc_err_count	0	0	0
Judgment			OK	OK

FuncRamChk_Start_InitRamChk

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(lul_num=0U; lul_num<F_INIT_LOOP_MAX; lul_num++)		○	○	○	○
	if((len_result == EN_RAMCHECK_RESULT_FAIL) ‖ (len_result == EN_RAMCHECK_RESULT_PASS))		○	○	-	-
	break		○	○	-	-
	else		-	-	○	○
	if(len_result == EN_RAMCHECK_RESULT_FAIL)		○	○	○	○
	else		-	-	-	-
	Symbol	Range				
Input	f_Cyclic_CheckProcess()	0~3	0	1	2	3
Output	RPERDIS	1	1	1	1	1
	luc_ans	0,1	0	0	0	1
Judgment			OK	OK	OK	OK

f_Cyclic_CheckProcess

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	switch(fen_phase_flag)		○	○	○	○	○	○	○	○
	case FEN_ROMCHECK_PHASE_OFF:		○	-	-	-	-	-	-	-
	case FEN_RAMCHECK_PHASE_START:		-	○	-	-	-	-	-	-
	case FEN_RAMCHECK_PHASE_SUB:		-	-	○	-	-	-	-	-
	case FEN_RAMCHECK_PHASE_MAIN:		-	-	-	○	-	-	-	-
	case FEN_RAMCHECK_PHASE_PASS:		-	-	-	-	○	-	-	-
	case FEN_RAMCHECK_PHASE_FAIL:		-	-	-	-	-	○	○	-
	if(fuc_err_count >= F_ERR_RETRY_MAX)		-	-	-	-	-	-	○	-
	else		-	-	-	-	-	○	-	-
	default:		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	fen_phase_flag	0~6	0	1	2	3	4	5	5	6
	f_Cyclic_CheckProcessSubPhase()	3,5	3	3	5	5	5	5	5	5
	f_Cyclic_CheckProcessMainPhase()	4,5	4	4	5	5	5	5	5	5
	fuc_err_count	0~3	0	0	0	0	0	3	0	0
Output	fuc_err_count	0~4	0	0	0	0	0	4	1	0
	fen_phase_flag	0~5	0	2	5	5	0	0	2	0
	len_ans	0~3	0	1	1	1	2	3	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	rom_chk.c

m_rom_sum_calc

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	while(lucp_now_address <= (U_CHR*)pus_end)		○	○	○	-
	else		-	-	-	○
	Symbol	Range				
Input	pus_start	0x0000~0x6001	0x0000	0x6000	0x6001	0x0001
	pus_end	0x1FFF~0x7FFF	0x1FFF	0x7FFF	0x8000	0x0000
	pus_base	0x0000	0x0000	0x0000	0x0000	0x0000
Output	pus_base	0x0000~0xFE00	0x12CE	0xFE00	0xFE00	0x0000
Judgment			OK	OK	OK	OK

ROMCHK_Get_SUM_NUM

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	uss_m_rom_sum_data	10	10
Output	uss_m_rom_sum_data	10	10
Judgment			OK

ROMCHK_Get_SUMChkflg

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	ucs_m_sum_chk	0x01	0x01
Output	ucs_m_sum_chk	0x01	0x01
Judgment			OK

ROMCHK_Process

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(ucs_m_sum_chk)		○	○	○	○	○	○
	case 1		○	-	-	-	-	-
	case 2		-	○	-	-	-	-
	case 3		-	-	○	-	-	-
	case 4		-	-	-	○	○	-
	if(uss_m_rom_sum_data != us_m_sum_adr[1][0])		-	-	-	○	-	-
	else		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	ucs_m_sum_chk	0~4	1	2	3	4	4	0
	uss_m_rom_sum_data	0	0	0	0	0	0	0
	ussc_m_sum_area[0][0]	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
	ussc_m_sum_area[0][1]	0x0000,0x1fff	0x1fff	0x1fff	0x1fff	0x1fff	0x0000	0x1fff
	ussc_m_sum_area[1][0]	0x0000,0x2000	0x2000	0x2000	0x2000	0x2000	0x0000	0x2000
	ussc_m_sum_area[1][1]	0x0000,0x3fff	0x3fff	0x3fff	0x3fff	0x3fff	0x0000	0x3fff
	ussc_m_sum_area[2][0]	0x0000,0x4000	0x4000	0x4000	0x4000	0x4000	0x0000	0x4000
	ussc_m_sum_area[2][1]	0x0000,0x5fff	0x5fff	0x5fff	0x5fff	0x5fff	0x0000	0x5fff
	ussc_m_sum_area[3][0]	0x0000,0x6000	0x6000	0x6000	0x6000	0x6000	0x0000	0x6000
	ussc_m_sum_area[3][1]	0x7fff	0x7fff	0x7fff	0x7fff	0x7fff	0x0000	0x7fff
	us_m_sum_adr[1][0]	0x0000,0xd793	0xd793	0xd793	0xd793	0xd793	0x0000	0xd793
Output	uss_m_rom_sum_data	0	0	0	0	0	0	0
	ucs_m_sum_chk	0~4	2	3	4	0	0	0
	luc_ans	0~2	2	2	2	1	0	2
Judgment			OK	OK	OK	OK	OK	OK

ROMCHK_Set_SUMChkflg_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	None	-
Output	ucs_m_sum_chk	1	1
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	tool.c

Tool_Asc_Hex

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(i = 0; i < puc_num; i++)		○	○	-	○
	for(luc_loop = 0; luc_loop < 2; luc_loop++)		○	○	-	○
	if((*pucp_pnt1 >= '0') && (*pucp_pnt1 <= '9'))		○	-	-	-
	if(luc_loop == 0)		○	-	-	-
	else		-	-	-	-
	else		-	○	-	○
	if((*pucp_pnt1 >= 'A') && (*pucp_pnt1 <= 'F'))		-	○	-	-
	if(luc_loop == 0)		-	○	-	-
	else		-	-	-	-
	else		○	-	-	○
	if(puc_num == 2)		○	-	-	-
	else		-	○	○	○
	if(puc_num == 4)		-	○	-	-
	else		○	-	○	○
	Symbol	Range				
Input	*pucp_pnt1	0x00~0xFF	0x30	0x41	0x00	0xFF
	*pucp_pnt2	0x00~0xFF	0x00	0x00	0xFF	0x00
	puc_num	0~5	2	4	0	5
Output	*pucp_pnt2	0~255	0	16	255	0
Judgment			OK	OK	OK	OK

Tool_Bin_Asc

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	-		-	-
	Symbol	Range		
Input	pus_num	0x0000,0xFFFF	0x0000	0xFFFF
	ppuc_asc	0x00,0xFF	0x00	0xFF
Output	ppuc_asc[0]	48,54	48	54
	ppuc_asc[1]	48,53	48	53
	ppuc_asc[2]	48,53	48	53
	ppuc_asc[3]	48,51	48	51
	ppuc_asc[4]	48,53	48	53
Judgment			OK	OK

Tool_Comp_Pnt

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(lul_num=0; lul_num<pul_size; lul_num++)		○	○	○	○
	if(*ppuc_add1 != *ppuc_add2)		○	○	-	○
	break		-	○	-	-
	else		-	-	-	-
	Symbol	Range				
Input	ppuc_add1	0x00,0xFF	0x00	0x00	0x00	0xFF
	ppuc_add2	0x00,0xFF	0x00	0x01	0x01	0xFF
	pul_size	0~11	10	10	0	11
Output	luc_ACK	0,1	1	1	0	1
	ppuc_add1	0~255	0	1	1	255
	ppuc_add2	0,255	0	0	0	255
Judgment			OK	OK	OK	OK

Tool_Copy_Pnt

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(lul_loop = 0; lul_loop < pul_num ; ++lul_loop)		○	○	○
	Symbol	Range			
Input	ppuc_pnt1	0x00,0xFF	0x00	0xFF	0xFF
	ppuc_pnt2	0x00,0xFF	0x00	0xFF	0xFF
	pul_num	0~257	0	256	257
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_Hex_Asc

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	for(luc_i = 0; luc_i < puc_num; luc_i++)		○	○	○	○	○	○	-	-	○	○
	for(luc_loop = 0; luc_loop < 2; luc_loop++)		○	○	○	○	○	○	-	-	○	○
	switch(*pusp_pnt2)		○	○	○	○	○	-	-	-	○	○
	case 10		○	-	-	-	-	-	-	-	-	-
	case 11		-	○	-	-	-	-	-	-	-	-
	case 12		-	-	-	-	○	-	-	-	-	-
	case 13		-	-	○	-	-	-	-	-	-	-
	case 14		-	-	-	-	-	○	-	-	-	-
	case 15		-	-	-	○	-	-	○	-	○	○
	default		○	○	○	-	○	○	○	-	○	○
	Symbol	Range										
Input	pucp_pnt1	0x00−0xFF	0xA0	0xB0	0xD0	0xFF	0xC0	0xE0	0xF0	0xF0	0xF0	0xF0
	pucp_pnt2	0x0000,0xFFFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFFFF
	puc_num	0x00−0xFF	0x01	0x01	0x01	0x01	0x01	0x01	0x01	0x00	0x02	0xFF
Output	pucp_pnt2[0]	0−70	65	66	68	70	67	69	70	0	48	48
	pucp_pnt2[1]	48,70	48	48	48	70	48	48	48	48	48	48
	luc_conv	0−254	2	2	2	2	2	2	2	0	4	254
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	uart_sau_command.c

m_SAU_Code_DU

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'W')		○	-	-	-
	else if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x57	0x52	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_SAU_Code_DUR

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○
	Symbol	Range		
Input	pucp_rxd	0x00,0xFF	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0xFF
Output	pucp_txd[10]	44,48	48	44
	pucp_txd[11]	44,48	48	44
Judgment			OK	OK

m_SAU_Code_DUW

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((luc_com_data[luc_i] != '1') && (luc_com_data[luc_i] != '0'))		○	○	○	○
	else		-	-	-	-
	if(luc_flg == M_OK)		-	-	-	-
	for(luc_i=0; luc_i<4; luc_i++)		-	-	-	-
	else		○	○	○	○
	Symbol	Range				
Input	*(pucp_rxd + M_PC_DATA_POS + (luc_i * 2))	0x00~0xFF	0x00	0x30	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0xFF
Output	*pucp_rxd	0	0	0	0	0
	*pucp_txd	0,255	0	0	0	255
Judgment			OK	OK	OK	OK

m_SAU_Code_HA

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'W')		○	-	-	-
	else if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x57	0x52	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_SAU_Code_HAR

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(luc_i=0; luc_i<13; luc_i++)		○	○
	Symbol	Range		
Input	pucp_rxd	0x00,0xFF	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0xFF
Output	pucp_txd[18]	44	44	44
	pucp_txd[127]	48	48	48
	pucp_txd[128]	44	44	44
	pucp_txd[129]	48	48	48
	pucp_txd[130]	44	44	44
	pucp_txd[131]	48	48	48
	pucp_txd[132]	44	44	44
	pucp_txd[133]	48	48	48
	pucp_txd[134]	44	44	44
	pucp_txd[135]	48	48	48
	pucp_txd[136]	44	44	44
Judgment			OK	OK

m_SAU_Code_HAW

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((luc_com_data[0] != '1') && (luc_com_data[0] != '0'))		○	-	-	-	-	○
	else		-	○	○	○	○	-
	if((luc_com_data[1] != '2') && (luc_com_data[1] != '1') && (luc_com_data[1] != '0'))		○	-	-	-	-	○
	else		-	○	○	○	○	-
	if((luc_com_data[2] != '3') && (luc_com_data[2] != '2') && (luc_com_data[2] != '1') && (luc_com_data[2] != '0'))		○	-	-	-	-	○
	else		-	○	○	○	○	-
	if((luc_com_data[3] != '1') && (luc_com_data[3] != '0'))		○	-	-	-	-	○
	else		-	○	○	○	○	-
	if(luc_flg == M_OK)		-	○	○	○	○	-
	if(luc_com_data[1] == '1')		-	-	○	-	-	-
	else if(luc_com_data[1] == '2')		-	-	-	○	○	-
	else		-	○	-	-	-	-
	if(luc_com_data[2] == '1')		-	-	○	-	-	-
	else if(luc_com_data[2] == '2')		-	-	-	○	-	-
	else if(luc_com_data[2] == '3')		-	-	-	-	○	-
	else		-	○	-	-	-	-
	if(luc_com_data[3] == '1')		-	-	○	○	○	-
	else		-	○	-	-	-	-
	else		○	-	-	-	-	○
	Symbol	Range						
Input	*(pucp_rxd + M_PC_DATA_POS + 0)	0x30~0xFF	0x32	0x30	0x31	0x31	0x31	0xFF
	*(pucp_rxd + M_PC_DATA_POS + 2)	0x30~0xFF	0x33	0x30	0x31	0x32	0x32	0xFF
	*(pucp_rxd + M_PC_DATA_POS + 4)	0x30~0xFF	0x34	0x30	0x31	0x32	0x33	0xFF
	*(pucp_rxd + M_PC_DATA_POS + 16)	0x30~0xFF	0x30	0x30	0x31	0x31	0x31	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

m_SAU_Code_RC

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	-	-
	else		-	○	○
	Symbol	Range			
Input	(*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x52	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0xFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_SAU_Code_RCR

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pucp_rxd	0x00,0xFF	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0xFF
Output	None	-	-	-
Judgment			OK	OK

m_SAU_Code_VP

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	-	-
	else		-	○	○
	Symbol	Range			
Input	*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')	0x00~0xFF	0x52	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0xFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_SAU_Code_VPR

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pucp_rxd	0x00,.0xFF	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0xFF
Output	pucp_txd[15]	44	44	44
Judgment			OK	OK

m_SAU_Code_VS

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	-	-	-	-
	else if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'W')		-	○	-	-	-
	else if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'C')		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')	0x00~0xFF	0x52	0x57	0x43	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_SAU_Code_VSC

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ROMCHK_Get_SUMChkflg() != M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ROMCHK_Get_SUMChkflg()	0~2	0	1	2
	pucp_rxd	0x00,0xFF	0x00	0xFF	0x00
	pucp_txd	0x00,0xFF	0x00	0xFF	0x00
Output	pucp_txd[10]	48,49	48	49	49
Judgment			OK	OK	OK

m_SAU_Code_VSR

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pucp_rxd	0x00,0xFF	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0xFF
Output	None	-	-	-
Judgment			OK	OK

m_SAU_Code_VSW

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ROMCHK_Get_SUMChkflg() == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ROMCHK_Get_SUMChkflg()	0~2	0	1	2
	pucp_rxd	0x00,0xFF	0x00	0xFF	0xFF
	pucp_txd	0x00,0xFF	0x00	0xFF	0xFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

UartSAUCommand_Command_Op

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(puc_com_mode)		○	○	○	○	○	○
	case Cmd_RC		○	-	-	-	-	-
	case Cmd_VP		-	○	-	-	-	-
	case Cmd_VS		-	-	○	-	-	-
	case Cmd_DU		-	-	-	○	-	-
	case Cmd_HA		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	puc_com_mode	1~9	1	2	3	4	5	9
	pucp_rxd	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

UartSAUCommand_Get_Cmd_NUM

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	Cmd_NUM	7	7
Judgment			OK

UartSAUCommand_Get_Comnd_Data

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_cmd_num	0~9	0	8	9
Output	uc_comnd_tbl[puc_cmd_num].uc_cmd_code	1~147	1	147	40
Judgment			OK	OK	OK

UartSAUCommand_Get_Comnd_Pointer

	Responsible	Last update
Coding	Ikarashi	2021/6/8
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_cmd_num	0~9	0	8	9
Output	&uc_comnd_tbl[puc_cmd_num].uc_cmd_chr[0]	82~215	82	204	215
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	uart_sau_main.c

m_Comand_Chk

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(m_Rxd_Sum_Chk() == M_OK)		-	-	-	○	○
	while(luc_cmd_cnt <= UartSAUCommand_Get_Cmd_NUM())		-	-	-	○	○
	if(Tool_Comp_Pnt((U_CHR *)&uc_m_rxd_buf[M_PC_COMMAND_POS], (U_CHR *)UartSAUCommand_Get_Comnd_Pointer(luc_cmd_cnt), 2) == 0)		-	-	-	○	-
	else		-	-	-	-	○
	else		-	-	-	-	-
	else		○	○	○	-	-
	Symbol	Range					
Input	m_Rxd_Sum_Chk()	0~2	1	1	2	0	0
	luc_cmd_cnt	0~9	0	8	9	0	0
	Tool_Comp_Pnt((U_CHR*)&uc_m_rxd_buf[M_PC_COMMAND_POS],(U_CHR*)UartSAUCommand_Get_Comnd_Pointer(luc_cmd_cnt), 2)	0~2	0	1	2	0	1
Output	luc_com_mode	0,255	255	255	255	0	255
Judgment			OK	OK	OK	OK	OK

m_Comu_Op_Sub

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(luc_com_mode != 0xff)		○	-
	else		-	○
	Symbol	Range		
Input	m_Comand_Chk	0x00~0xFF	0x00	0xFF
Output	None	-	-	-
Judgment			OK	OK

m_Rxd_Sum_Chk

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(lus_i = 0; lus_i < sizeof(uc_m_rxd_buf); lus_i++)		○	○	○	○
	if(*lucp_pnt == M_C_STX)		○	-	-	-
	else		-	○	○	○
	if(luc_stx_flg == M_ON)		○	-	-	-
	else		-	○	○	○
	if(*lucp_pnt == M_C_ETX)		-	○	-	-
	else		○	-	○	○
	if(!(luc_sum_1_dat == luc_sum_ck_dat))		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	&uc_m_rxd_buf[0]	0x00~0xFF	0x02	0x03	0x00	0xFF
Output	luc_ans	1	1	1	1	1
Judgment			OK	OK	OK	OK

m_Txd_Busy

	Responsible	Last update
Coding	lkarashi	2021/6/9
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	us_m_rxd_cnt	0	0
	uc_rxd_chk_flg	0	0
Output	None	-	-
Judgment			OK

m_Txd_Start

	Responsible	Last update
Coding	lkarashi	2021/6/9
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pus_txd_size	0x0000,0xFFFF	0x0000	0xFFFF
Output	None	-	-	-
Judgment			OK	OK

m_Txd_Sum_Calc

	Responsible	Last update
Coding	lkarashi	2021/6/9
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(lus_i = 0; lus_i < (lus_txd_count); lus_i++)		○	○
	Symbol	Range		
Input	lus_txd_count	0x0000,0xFFFF	0x0000	0xFFFF
	uc_m_txd_buf[0]	0x00,0xFF	0x00	0xFF
Output	luc_sum_1_dat	0,76	0	76
Judgment			OK	OK

UARTSAUMAIN_Comu_Op

	Responsible	Last update
Coding	lkarashi	2021/6/9
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_rxd_chk_flg == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_rxd_chk_flg	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

UartSAUMain_Err_Op

	Responsible	Last update
Coding	lkarashi	2021/6/9
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_txd_buf[M_PC_DATA_POS]	0x15	0x15
Judgment			OK

UARTSAUMAIN_Get_Rxd_Data

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(luc_rxd_one_data == M_C_STX)		-	-	-	○	-	-	-
	else		○	○	○	-	○	○	○
	if(suc_stx_received_flg == M_ON)		○	○	-	○	-	-	○
	else		-	-	○	-	○	○	-
	if(us_m_rxd_cnt >= lus_buf_limit)		○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-
	if((luc_rxd_one_data == M_C_CR) || (luc_rxd_one_data == M_C_EOT) || (luc_rxd_one_data == M_C_ETB))		-	-	-	-	○	○	○
	else		○	○	○	○	-	-	-
	Symbol	Range							
Input	uc_m_one_char_receive[0]	0x00~0xFF	0x00	0xFF	0x00	0x02	0x04	0x17	0x0D
	us_m_rxd_cnt	0~300	300	299	0	0	299	0	299
	suc_stx_received_flg	0,1	1	1	0	0	0	0	1
Output	uc_m_rxd_buf[us_m_rxd_cnt]	0x00,0x02	0x00	0x00	0x00	0x02	0x02	0x02	0x02
	us_m_rxd_cnt	0~299	299	299	0	1	0	0	299
	uc_m_terminator	0x00~0x17	0x00	0x00	0x00	0x00	0x04	0x17	0x0D
Judgment			OK	OK	OK	OK	OK	OK	OK

UartSAUMain_Head_Set

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(luc_i = 0; luc_i < M_PC_DATA_POS; ++luc_i)		○	-
	Symbol	Range		
Input	uc_m_rxd_buf[luc_i]	0x00,0xFF	0x00	0xFF
Output	uc_m_txd_buf[luc_i]	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

UartSAUMain_Rxd_Buf_Pointre

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	-		-
	Symbol	Range	
Input	None	-	-
Output	*uc_m_rxd_buf	255	255
Judgment			OK

UARTSAUMAIN_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(lus_i=0; lus_i<M_RXD_MAX_SIZE; lus_i++)		○
	for(lus_i=0; lus_i<M_TXD_MAX_SIZE; lus_i++)		○
	for(lus_i=0; lus_i<2; lus_i++)		○
	Symbol	Range	
Input	None	-	-
Output	uc_m_rxd_buf[lus_i]	0	0
	uc_m_txd_buf[lus_i]	0	0
	uc_m_one_char_receive[lus_i]	0	0
	us_m_rxd_cnt	0	0
	uc_m_terminator	0	0
	uc_rxd_chk_flg	0	0
Judgment			OK

UARTSAUMAIN_Set_Rxd_Next_Buf

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

UartSAUMain_tx_nbyte

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pus_num	0x0000,0xFFFF	0x0000	0xFFFF
	uc_m_terminator	0x00,0xFF	0x00	0xFF
Output	uc_m_txd_buf[13 + pus_num]	255	255	255
Judgment			OK	OK

UartSAUMain_Txd_Buf_Pointre

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	*uc_m_txd_buf	255	255
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_camp.c

R_CAMP0_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_CAMP1_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_CAMP_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_cgc.c

R_CGC_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_cgc_user.c

R_CGC_Get_ResetSource

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_it.c

R_IT_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_IT_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_IT_Stop

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_it_user.c

r_it_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_lvi.c

R_LVI_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_lvi_user.c

r_lvi_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_main.c

main

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	Yes
Verification pass	1
Judgment	OK

			case1
Root	while(1U)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_MAIN_UserInit

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_pga_dsad.c

R_PGA_DSAD_CAMP_OffsetTrimming

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_PGA_DSAD_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	while(0U == AFESTAT)		○
	while(0U == PGASTAT)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_PGA_DSAD_Get_AverageResult

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	bufferH	0x0000,0xFFFFF	0x0000	0xFFFFF
	bufferL	0x0000,0xFFFFF	0x0000	0xFFFFF
	DSADMV0	0x00,0xFF	0x00	0xFF
	DSADMV1	0x00,0xFF	0x00	0xFF
Output	*bufferH	0x00,0xFF	0x00	0xFF
	*bufferL	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

R_PGA_DSAD_Get_Result

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	-		-	-
	Symbol	Range		
Input	bufferH	0x0000,0xFFFFF	0x0000	0xFFFFF
	bufferL	0x0000,0xFFFFF	0x0000	0xFFFFF
	DSADCR0	0x00,0xFF	0x00	0xFF
	DSADCR1	0x00,0xFF	0x00	0xFF
Output	*bufferH	0x00,0xFF	0x00	0xFF
	*bufferL	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

R_PGA_DSAD_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_PGA_DSAD_Stop

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_pga_dsad_user.c

r_pga_dsad_interrupt_conversion

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_port.c

R_PORT_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_sau.c

R_SAU0_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_UART0_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_UART0_Receive

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(rx_num < 1U)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	rx_buf	0x00,0xFF	0x00	0x00	0xFF
	rx_num	0~2	0	1	2
Output	g_uart0_rx_count	0	0	0	0
	g_uart0_rx_length	0~2	0	1	2
	gp_uart0_rx_address	0x00,0xFF	0x00	0x00	0xFF
	status	0x00,0x81	0x81	0x00	0x00
Judgment			OK	OK	OK

R_UART0_Send

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(tx_num < 1U)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	tx_num	0~123	0	122	123
	tx_buf	0x00,0xFF	0x00	0x00	0xFF
Output	gp_uart0_tx_address	0x00	0x00	0x00	0x00
	g_uart0_tx_count	0~122	0	121	122
	status	0,129	129	0	0
Judgment			OK	OK	OK

R_UART0_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_UART0_Stop

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_sau_user.c

r_uart0_callback_error

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	err_type	0	0
Output	None	-	-
Judgment			OK

r_uart0_callback_receiveend

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_uart0_callback_sendend

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_uart0_interrupt_error

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	*gp_uart0_rx_address	0	0
Judgment			OK

r_uart0_interrupt_receive

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(g_uart0_rx_length > g_uart0_rx_count)		○	○	-
	if(g_uart0_rx_length == g_uart0_rx_count)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	g_uart0_rx_length	0x0000~0xFFFF	0x0001	0xFFFF	0x0000
	g_uart0_rx_count	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
Output	gp_uart0_rx_address	1, 65535	1	1	65535
	g_uart0_rx_count	0	0	0	0
Judgment			OK	OK	OK

r_uart0_interrupt_send

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(g_uart0_tx_count > 0U)		○	-
	else		-	○
	Symbol	Range		
Input	g_uart0_tx_count	0x0000,0xFFFF	0xFFFF	0x0000
Output	gp_uart0_tx_address	-	0	0
	g_uart0_tx_count	0,65534	65534	0
Judgment			OK	OK

r_uart0_callback_softwareoverrun

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1
Root	None		○
	Symbol	Range	
Input	rx_data	0x0000	0x0000
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_systeminit.c

hdwinit

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_Systeminit

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_wdt.c

R_WDT_Create

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_WDT_Restart

	Responsible	Last update
Coding	Ikarashi	2021/6/9
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_wdt_user.c

r_wdt_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	ad_12bit.c

AD12BIT_Init_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_ch=F_INIT_VAL; luc_ch<(U8_CHR)M_12BITAD_QUANTITY_NUMBER; luc_ch++)		○
	for(luc_ave=F_INIT_VAL; luc_ave<M_12BITAD_AVERAGE_NUMBER; luc_ave++)		○
	Symbol	Range	
Input	None	-	
Output	fen_inital_flag	0	0
	fl_m_12bit_ad_data[0]	0	0
	uc_m_12bit_ad_average_flg	0	0
	uc_m_12bit_ad_average[0][0]	0	0
Judgment			OK

Ad12bit_Ad_Ave_Op

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<M_12BITAD_QUANTITY_NUMBER; luc_i++)		○	○	○
	for(luc_j=0; luc_j<M_12BITAD_AVERAGE_NUMBER; luc_j++)		○	○	○
	Symbol	Range			
Input	us_m_12bit_ad_average[luc_i][luc_j]	0~2801	0	2800	2801
	m_bit12AD_Conversion(lsl_ad_average_data[luc_i])	0~1915	0	1914.0625	1915
Output	uc_m_12bit_ad_average_flg	0	0	0	0
	fl_m_12bit_ad_data[luc_i]	0~1915	0	1914.063	1915
Judgment			OK	OK	OK

Ad12bit_Calc_Voltage

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	psl_data	0	0
Output	lfl_ad	0	0
Judgment			OK

Ad12bit_Check_InitalProcessFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(fen_inital_flag != FEN_AD12BIT_INITIAL_PROCESS_COMPLETE)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	fen_inital_flag	0~3	0	2	3
Output	lbl_ans	0,1	1	0	1
Judgment			OK	OK	OK

Ad12bit_Get_Ad_Data

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_ch	0~3	0	2	3
Output	fl_m_12bit_ad_data[puc_ch]	0,1915	1915	0	0
Judgment			OK	OK	OK

Ad12bit_Int_Ad_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case1	case3
Root	if(S12AD.ADCSR.BIT.ADST == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	S12AD.ADCSR.BIT.ADST	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

Ad12bit_Int_Adc

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_12bit_ad_average_flg == OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_m_12bit_ad_average_flg	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_bit12AD_To_Average_Buffer

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<M_12BITAD_QUANTITY_NUMBER; luc_i++)		○	○	○	○	○
	if ((luc_i==AD12BIT_AN002)&&(O2WarmUp_Get_WarmFlg()==1))		-	-	-	-	-
	else		○	○	○	○	○
	if(ECSENSOR_Not_Use_Flg(luc_i) == 0U)		○	○	-	○	○
	if(luc_average_num[luc_i] >= M_12BITAD_AVERAGE_NUMBER)		○	○	-	-	○
	if(fen_inital_flag == FEN_AD12BIT_INITIAL_PROCESS_GET_AD)		○	○	-	-	-
	else		-	-	○	-	-
	Symbol	Range					
Input	*pusp_buffer	0	0	0	0	0	0
	ECSENSOR_Not_Use_Flg()	0,1	0	0	1	0	0
	O2WarmUp_Get_WarmFlg()	1,2	1	2	1	1	1
	fen_inital_flag	0,1	0	0	0	0	1
	luc_average_num[0]	18,19	19	19	19	18	19
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	alarm_event.c

AlarmEvent_Judg_AlarmEvent

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

		case1	case2	case3	case4	case5
Root	if(stp_data->uc_alarm_on == M_ON)	○	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	○	-	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))	○	○	-	-	-
	if((luc_i == DataStructConst_CO_CAN_H2_H2_CH) && ((stp_GasInfo_addr+luc_i)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM))	○	-	-	-	-
	else	-	○	-	-	-
	else	-	-	○	-	-
	else	-	-	-	○	○
	Symbol	Range				
Input	stp_data->uc_alarm_on	0~2	1	1	1	0	2
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	3	29	0	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x00	0xFF	0x00	0xFF
	(stp_denst+luc_i)->uc_over_flg	0~2	1	0	0	0	2
Output	(stp_denst+luc_i)->uc_state_alm	0x00	0x00	0x00	0x00	0x00	0x00
	(stp_denst+luc_i)->uc_alarm_lat	0x00	0x00	0x00	0x00	0x00	0x00
	ucs_m_reset_flg	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

AlarmEvent_Set_AlarmReset

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

		case1
Root	None	-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_reset_flg	1	1
Judgment			OK

AlarmEvent_Set_InitAlarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

		case1
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○
	Symbol	Range	
Input	None	-	-
Output	(stp_denst+luc_i)->uc_alarm_lat	0x00	0x00
	(stp_denst+luc_i)->uc_state_alm	0x00	0x00
Judgment			OK

m_Check_GasAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(ZIPCNumSet_Get_GasTestFlg() == M_ON)		○	-	-	-	-	-	-	-	-
	if((puc_num == DataStructConst_IR_GAS_CH) && (stp_GasInfo_addr->uc_gas_comb == DataStructConst_IR_GAS_NUM))		○	-	-	-	-	-	-	○	○
	else		-	-	-	-	-	-	-	○	○
	else		-	○	○	○	○	○	○	-	-
	if(stp_denst->ul_denst != M_OFF_VAL)		○	○	○	○	○	○	-	-	○
	if(stp_denst->uc_denst_minus_flg == M_OFF)		○	○	○	○	○	○	-	-	○
	if(puc_num == 1)		-	-	-	○	○	-	-	-	-
	if(stp_GasInfo_addr->ul_1st_alarm != M_OFF_VAL)		-	-	-	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-
	if(stp_GasInfo_addr->ul_2nd_alarm != M_OFF_VAL)		-	-	-	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-
	else		○	○	○	-	-	-	-	-	○
	if(stp_GasInfo_addr->ul_1st_alarm != M_OFF_VAL)		○	-	-	-	-	-	-	-	○
	else		-	○	○	-	-	-	-	-	-
	if(stp_GasInfo_addr->ul_2nd_alarm != M_OFF_VAL)		○	-	-	-	-	-	-	-	○
	else		-	○	○	-	-	-	-	-	-
	if(stp_GasInfo_addr->ul_3rd_alarm != M_OFF_VAL)		○	-	-	○	-	-	-	-	○
	else		-	○	○	-	○	-	-	-	-
	else		-	-	-	-	-	○	-	-	-
	if(puc_num == 1)		-	-	-	-	-	○		-	-
	else		-	-	-	-	-	-	○	-	-
	if(stp_denst->ul_stel != M_OFF_VAL)		○	○	-	-	-	-	-	-	○
	if(stp_GasInfo_addr->ul_stel_alarm != M_OFF_VAL)		○	-	-	-	-	-	-	-	○
	else		-	○	-	-	-	-	-	-	-
	else		-	-	○	○	○	○	○	-	-
	if(stp_denst->ul_twa != M_OFF_VAL)		○	○	○	○	○	-	-	-	○
	if(stp_GasInfo_addr->ul_twa_alarm != M_OFF_VAL)		○	○	○	○	-	-	-	-	○
	if(puc_num == DataStructConst_CO_CH)		-	○	○	-	-	-	-	-	-
	if(stp_data->uc_1h_alarm_on == M_ON)		-	○	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-
	else		○	-	-	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	○	○	-	-
	else		-	-	-	-	-	-	-	○	○
	if(stp_GasInfo_addr->ul_over_alarm != M_OFF_VAL)		○	-	-	-	-	-	-	○	○
	else		-	○	○	○	○	○	○	-	-
	if(stp_GasInfo_addr->ul_upper_limit != M_OFF_VAL)		○	○	-	-	-	-	-	○	○
	else		-	-	○	○	○	○	○	-	-
	Symbol	Range									
Input	puc_num	0~5	4	3	3	1	1	1	3	0	5
	stp_GasInfo_addr->ul_1st_alarm	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0x00000000	0xDDDDDDDD	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_GasInfo_addr->ul_2nd_alarm	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0x00000000	0xDDDDDDDD	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_GasInfo_addr->ul_3rd_alarm	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0x00000000	0xDDDDDDDD	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_GasInfo_addr->ul_stel_alarm	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_GasInfo_addr->ul_twa_alarm	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xDDDDDDDD	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	ZIPCNumSet_Get_GasTestFlg()	0~2	1	0	0	0	0	0	2	1	1
	stp_GasInfo_addr->uc_gas_comb	0~30	1	0	0	0	0	0	30	29	29
	stp_denst->ul_denst	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0xDDDDDDDD	0xDDDDDDDD
	stp_denst->uc_denst_minus_flg	0~2	0	0	0	0	0	1	1	2	2
	stp_denst->ul_stel	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xFFFFFFFF	0xFFFFFFFF
	stp_denst->ul_twa	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0xFFFFFFFF	0xFFFFFFFF
	stp_data->uc_1h_alarm_on	0~2	1	1	0	1	0	0	0	0	2
	stp_GasInfo_addr->ul_over_alarm	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xFFFFFFFF	0xFFFFFFFF
	stp_denst->uc_over_minus_flg	0~2	1	1	1	1	1	1	0	2	2
	stp_denst->uc_over_flg	0~2	1	1	1	1	1	1	0	2	2
	stp_GasInfo_addr->ul_upper_limit	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xFFFFFFFF	0xFFFFFFFF
Output	stp_denst->uc_state_alm	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_GasAlarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(stp_data->uc_alarm_pattern == M_OFF || ucs_m_reset_flg == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_num	0~5	4	0	5
	stp_denst->uc_state_alm	0x00~0xFF	0x00	0xFF	0xFF
	stp_data->uc_alarm_pattern	0~2	0	1	2
	ucs_m_reset_flg	0~2	1	0	2
Output	stp_denst->uc_alarm_lat	0x00,0xFF	0x00	0xFF	0x00
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	Battery.c

Battery_Get_BatteryAD

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fls_batt_ad	5743	5743
Judgment			OK

Battery_Get_ComBatteryAD

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fls_com_batt_ad	5743	5743
Judgment			OK

Battery_Get_ComBatteryIconAD

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_batt_status	6	6
Judgment			OK

Battery_Get_PowerOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_batt_power_off	1	1
Judgment			OK

Battery_Set_BatteryMark

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch(uc_batt_status)		○	○	○	○	○	○	○
	case BATTERY_BATT_STATUS_FULL		○	-	-	-	-	-	-
	case BATTERY_BATT_STATUS_MANY		-	○	-	-	-	-	-
	case BATTERY_BATT_STATUS_NORMAL		-	-	○	-	-	-	-
	case BATTERY_BATT_STATUS_WARNING		-	-	-	○	-	-	-
	case BATTERY_BATT_STATUS_ALARM		-	-	-	-	○	-	-
	default		-	-	-	-	-	○	○
	Symbol	Range							
Input	uc_batt_status	0~6	5	4	3	2	1	0	6
Output	None	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK

f_Check_LowTempMoveFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3
Root	if(FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER)		○	-	○
	if(stp_sub_ad->fl_temp_data < F_LOW_TEMP_VAL)		○	-	-
	else		-	-	○
	else		-	○	-
	Symbol	Range			
Input	RL78I1EControl_Get_Data_point()	-11,-10	-11	-11	-10
	FlowMain_Get_ModeStatus()	0,1	0	1	0
Output	lbl_init_mode_flag	0,1	1	0	0
Judgment			OK	OK	OK

m_Calc_BatteryAD

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Ad12bit_Get_Ad_Data(0)	0~1915	1914.0625	0	1915
Output	ucs_batt_type	0	0	0	0
Judgment			OK	OK	OK

Battery_Chk_Battery

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
verification pass	15

Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16
Root	if(fbl_init_move_flag == false)	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	else	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(ucs_init_count == 0)	○	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(fls_com_batt_ad > fls_batt_ad)	○	○	-	○	-	-	-	-	-	-	○	○	○	○	○	○
	else	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uss_return_count >= M_BATT_RETURN_COUNT_MAX)	○	○	○	○	-	○	○	○	-	-	-	-	-	-	-	-
	if((uc_batt_status != BATTERY_BATT_STATUS_ALARM) && (uc_batt_status != BATTERY_BATT_STATUS_SHUT_DOWN))	○	○	○	○	-	-	-	○	-	-	-	-	-	-	-	-
	if((flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_1] <= fls_batt_ad)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_2] <= fls_batt_ad)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_3] <= fls_batt_ad)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((uc_batt_status == BATTERY_BATT_STATUS_FULL) && (flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_3] > fls_batt_ad))	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else if((uc_batt_status == BATTERY_BATT_STATUS_MANY) && (flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_2] > fls_batt_ad))	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else if((uc_batt_status == BATTERY_BATT_STATUS_NORMAL) && (flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_1] > fls_batt_ad))	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else if((uc_batt_status == BATTERY_BATT_STATUS_WARNING) && (flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_0] > fls_batt_ad))	○	○	○	○	○	-	-	-	-	-	-	-	○	-	-	-
	else if((uc_batt_status == BATTERY_BATT_STATUS_ALARM) && (flc_level_down_limit_val[uc_low_temp_flag][FEN_BATT_LEVEL_5] > fls_batt_ad))	-	-	-	-	-	○	-	-	-	-	-	-	-	○	-	-
	else if(uc_batt_status == BATTERY_BATT_STATUS_SHUT_DOWN)	-	-	-	-	-	-	○	-	-	-	-	-	-	-	○	-
	else	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	○
	Symbol	Range															
Input	f_Check_LowTempMoveFlag()	0,1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	ucs_init_count	0,1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	fls_batt_ad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	fls_com_batt_ad	0,1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
	uss_return_count	238,239	239	239	239	239	238	239	239	238	238	238	238	238	238	238	238
	uc_batt_status	0~255	2	2	2	2	2	1	0	255	6	5	4	3	2	1	0	255
Output	fls_com_batt_ad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_batt_status	0~255	1	1	1	1	1	0	0	255	6	4	3	2	1	0	0	255
	ucs_batt_power_off	0,1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	luc_ans	0,1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	buzzer.c

BUZZER_Chk_TestFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(us_m_test_sdm_flg == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	us_m_test_sdm_flg	0~2	0	1	2
	us_m_test_type_flg	0~2	0	1	2
Output	luc_ans	0~2	0	1	2
Judgment			OK	OK	OK

BUZZER_Get_BuzzerOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_buzz_on_flg	0	0
Judgment			OK

BUZZER_Get_MotorOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_motor_on_flg	0	0
Judgment			OK

BUZZER_Hz_Set_Op

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_m_FLG_ALARM_CYC_REQ_chk == M_ON)		○	-	-	-	-
	else		-	○	○	○	○
	if(us_m_test_sdm_flg)		○	-	-	○	○
	else		-	○	○	-	-
	if(uc_m_FLG_ALARM_CYC_REQ_flg == M_ON)		○	○	○	-	-
	if((us_m_test_sdm_flg == 0U) || (us_m_test_type_flg != 0U))		-	○	-	-	-
	else		○	-	○	-	-
	if(is_tbl_end)		○	○	-	-	-
	if(tinfo->short_bz_req!=ALARM_PTN_OFF)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	uc_m_FLG_ALARM_CYC_REQ_chk	0~2	1	0	0	2	2
	us_m_test_sdm_flg	0~2001	2000	0	0	2001	2001
	uc_m_FLG_ALARM_CYC_REQ_flg	0~2	1	1	1	2	0
	us_m_test_type_flg	0~3	0	3	0	0	2
	tinfo->short_bz_req	0~17	0	16	0	17	17
Output	alarm_ptn	17	17	17	17	17	17
	uc_alm_ptn	0	0	0	0	0	0
	tinfo->short_bz_req	0~17	0	16	0	17	17
	uc_m_FLG_ALARM_CYC_REQ_chk	0~2	1	0	0	2	2
	us_m_test_sdm_flg	0~2000	1999	0	0	2000	2000
Judgment			OK	OK	OK	OK	OK

BUZZER_Init_Set

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		○
	Symbol	Range	
Input	None		
Output	alarm_ptn	0	0
	uc_m_FLG_ALARM_CYC_REQ_chk	0	0
	uc_m_FLG_ALARM_CYC_REQ_flg	0	0
Judgment			OK

BUZZER_Interrupt_A_CompareMatch

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_buzz_on_flg == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	us_tgr_a_buf	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	us_tgr_b_buf	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	uc_buzz_on_flg	0~2	0	1	2
Output	MTU2.TGRA	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	MTU2.TGRB	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
Judgment			OK	OK	OK

BUZZER_Set_AlarmSilence

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_onoff	0~2	1	0	2
Output	uc_silence_active_flag	0~2	1	0	2
Judgment			OK	OK	OK

BUZZER_Set_Boot_Flg

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	1	0	2
Output	uc_m_boot_status	0~2	1	0	2
Judgment			OK	OK	OK

BUZZER_Set_FailBuzzer

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_onoff == M_OFF)		○	○	-	-
	if(uc_fail_bz == M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	if(uc_fail_bz == M_OFF)		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	puc_onoff	0~2	0	0	1	2
	uc_fail_bz	0~2	1	0	0	2
Output	uc_fail_bz	0~2	0	0	1	2
Judgment			OK	OK	OK	OK

BUZZER_Set_PowerOFF

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_turn_off_flg	1	1
Judgment			OK

BUZZER_Set_TestBuzzer

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(puc_onoff == M_OFF)		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○
	switch(puc_type)		-	○	○	○	○	○	○	○	○
	case 0		-	○	-	-	-	-	-	-	-
	case 1		-	-	○	-	-	-	-	-	-
	case 2		-	-	-	○	-	-	-	-	-
	case 3		-	-	-	-	○	-	-	-	-
	case 4		-	-	-	-	-	○	-	-	-
	case 5		-	-	-	-	-	-	○	-	-
	case 6		-	-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	-	○
	Symbol	Range									
Input	puc_onoff	0~2	0	1	1	1	1	1	1	1	2
	puc_type	0~7	0	0	1	2	3	4	5	6	7
Output	us_m_test_type_flg	0~3	0	2	1	2	2	1	1	3	3
	us_m_test_sdm_flg	0,2000	0	2000	2000	2000	2000	2000	2000	2000	2000
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

BUZZER_Short_Command

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if(puc_type == 9)		○	-	-	-	-	-	-	-	-	-
	else if(puc_type == 7)		-	○	-	-	-	-	-	-	-	-
	else if(puc_type == 6)		-	-	○	-	-	-	-	-	-	-
	else if(puc_type == 5)		-	-	-	○	-	-	-	-	-	-
	else if(puc_type == 4)		-	-	-	-	○	-	-	-	-	-
	else if(puc_type == 3)		-	-	-	-	-	○	-	-	-	-
	else if(puc_type == 2)		-	-	-	-	-	-	○	-	-	-
	else if(puc_type == 1)		-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	○	○
	if(puc_type == 1 || puc_type == 2)		-	-	-	-	-	-	○	○	-	-
	if(stp_data->uc_key_tone_onoff == M_OFF)		-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	○	-	-
	else		○	○	○	○	○	○	-	-	○	○
	Symbol	Range										
Input	puc_type	0~10	9	7	6	5	4	3	2	1	0	10
	stp_data->uc_key_tone_onoff	0,1	1	1	1	1	1	1	0	1	0	2
Output	uc_m_short_bz_flg	0~7	1	7	6	5	4	3	0	1	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

BUZZER_Test_Command

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_buz_onoff == M_OFF)		○	-	-
	else		-	○	○
	if(puc_vib_onoff == M_OFF)		○	-	-
	else		-	○	○
	if(puc_led_onoff == M_OFF)		○	-	-
	else		-	○	○
	if((puc_buz_onoff | puc_vib_onoff | puc_led_onoff) != M_OFF		-	○	○
	else		○	-	-
	Symbol	Range			
Input	puc_buz_onoff	0~2	0	1	2
	puc_vib_onoff	0~2	0	1	2
	puc_led_onoff	0~2	0	1	2
Output	us_m_test_sdm_flg	0,6000	0	6000	6000
Judgment			OK	OK	OK

f_Cyclic_250msecProcess

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3	case4
Root	if((FlowMain_Get_ModeStatus() != FlowMain_MEASURE_MODE_NUMBER) && (FlowMain_Get_ModeStatus() != FlowMain_DISP_MODE_NUMBER))		○	-	○	○
	else		-	○	-	-
	if(GasFuncAir_Disp_Get_AirCalFlg() == M_ON)		○	-	○	-
	else		-	○	-	○
	if(Battery_Get_ComBatteryIconAD() <= BATTERY_BATT_STATUS_WARNING)		○	-	-	-
	else		-	○	○	○
	if((luc_mode_chk == M_OFF) && (luc_batt_chk == M_OFF))		-	○	-	-
	else		○	-	○	○
	if(luc_4sec_count >=M_4SEC_NUM)		○	-	○	○
	if(luc_batt_chk == M_ON)		○	-	-	-
	else		-	-	○	○
	else		-	○	-	-
	シンボル	範囲				
Input	luc_4sec_count	0~16	15	0	15	16
	FlowProcesDisplay_Get_PumpOffFlag()	0~2	0	0	1	2
	FlowProcesDisplay_Get_PurgeFlag()	0~2	0	0	1	2
	FlowMain_Get_ModeStatus()	0~8	0	1	7	8
	GasFuncAir_Disp_Get_AirCalFlg()	0~2	1	0	1	2
	Battery_Get_ComBatteryIconAD()	2~6	2	5	5	6
Output	luc_4sec_count	0,1	0	1	0	0
	uc_m_batt_buz_flg	1	1	1	1	1
	uc_m_mode_led_flg	0,1	0	0	1	1
Judgment			OK	OK	OK	OK

m_Check_AlarmCondition

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○
	Symbol	Range		
Input	(stp_denst + luc_i)->uc_alarm_lat	0x00,0xFF	0xFF	0x00
Output	luc_alarm_total	0x00,0xFF	0xFF	0x00
Judgment			OK	OK

m_lo_Buzzer_Set

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(puc_onoff == M_ON)		○	○	○	-	-
	if((3000 <= pus_count) && (pus_count <= 7373))		○	○	-	-	-
	if(puc_hilo == BUZZER_LV_LO)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	puc_onoff	0~2	1	1	1	0	2
	pus_count	2999~7374	7373	7373	2999	2999	7374
	puc_hilo	0~101	0	100	0	0	101
Output	uc_buzz_on_flg	0,1	1	1	1	0	0
	us_tgr_a_buf	7373	7373	7373	7373	7373	7373
	us_tgr_b_buf	3686	3686	3686	3686	3686	3686
Judgment			OK	OK	OK	OK	OK

m_lo_Led_Set

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_set & M_LED_P1)		○	-	-	-
	else		-	○	○	○
	if(puc_set & M_LED_P2)		-	○	-	-
	else		○	-	○	○
	if(puc_set & M_LED_P3)		-	-	○	-
	else		○	○	-	○
	Symbol	Range				
Input	puc_set	0x01~0xFF	0x01	0x02	0x04	0xFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_lo_Vibration_Set

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	puc_onoff	0,1	0	1	2
Output	uc_motor_on_flg	0,1	0	1	1
Judgment			OK	OK	OK

m_cyc_exec

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((*is_tbl_end) && (tmp_para.alm_re_excute == (U_CHR)ALARM_RE_EXECUTE_ENA))		○	-	-	-	-	-
	else		-	○	○	○	○	-
	if(*is_tbl_end)		○	-	-	-	-	-
	else		-	○	○	○	○	-
	if(stp_data->uc_stealth_on == M_ON)		-	○	-	-	-	-
	else		-	-	○	○	○	-
	if(ZIPCNumSet_Get_GasTestFlg() == M_ON)		-	-	○	○	○	-
	if(arm_ctl_tbl[alarm_ptn].bz_freq_typ != BUZZER_FREQ_TYP_STATIC)		-	-	○	-	-	-
	else if((arm_ctl_tbl[p_alarm_ptn].uc_silence == ALARM_SILENCE_ENA) && (uc_silence_active_flag == M_ON))		-	-	-	○	-	-
	else		-	-	-	-	○	-
	if((stp_data->uc_stealth_on == M_ON) && (stp_data->uc_stealth_motor_on == M_OFF))		○	-	-	-	-	-
	else		-	○	○	○	○	-
	Symbol	Range						
Input	is_tbl_end	0,1	1	0	0	0	0	2
	stp_data->uc_stealth_on	0,1	0	1	0	0	0	2
	ZIPCNumSet_Get_GasTestFlg()	0,1	0	0	1	0	0	2
	alarm_ptn	0~16	0	0	0	1	0	17
	stp_data->uc_stealth_motor_on	0,1	0	1	1	1	1	2
	uc_silence_active_flag	0x0000~0xFFFF	0x0000	0x0000	0x0000	0x0003	0x0000	0xFFFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

m_init_get_param

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None		-
Output	pst_tinfo->bz_tbl_alm_idx	0	0
	pst_tinfo->is_req_power_off	0	0
Judgment			OK

BUZZER_Main_Loop_Op

	Responsible	Last update
Coding	Ikeiishi	2021/6/10
Test	Taki	
Judgment	Ikeiishi	

Infinite loop	None
Verification pass	22
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20	case21	case22	case23	case24
Root	if(uc_m_boot_status == 1)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(KeyEvent_Get_KeyPowerOff() == M_ON || uc_turn_off_flg == M_ON)		-	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(us_m_test_sdm_flg != 0)		-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(us_m_test_type_flg == 3)		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(us_m_test_type_flg == 1)		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(us_m_test_type_flg == 2)		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_fail_bz != M_OFF)		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else if(GasFuncAr_Disp_Get_ArCalFlg() == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	else if(uc_mandown_bz != M_OFF)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else if(stp_status->uc_man_down & (DataStrcutConst_MAN_AL1 | DataStrcutConst_FAN_AL1))		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-
	if(stp_status->uc_man_down & (DataStrcutConst_FAN_AL1 | DataStrcutConst_MAN_AL2))		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else if(stp_status->uc_man_down & DataStrcutConst_MAN_AL1)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER || ZPCNumGet_Get_GasTestFlg() == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_alarm_chk_bit)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_1ST)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_2ND)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_3RD)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_MIN)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_OVR)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_ _1HI)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_TWA)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(uc_alarm_chk_bit & ALARMEVENT_ALM_STL)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	if(uc_vb_lp_flg == ALARM_PTN_OFF) && (FlowProcessDisplay_Get_PurgeFlag() == false))		-	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_m_short_bz_flg == 1)		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_short_bz_flg == 2)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_short_bz_flg == 3)		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_short_bz_flg == 4)		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_short_bz_flg == 5)		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_short_bz_flg == 6)		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_short_bz_flg == 7)		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(uc_vb_lp_flg == ALARM_PTN_OFF)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-
	if((KeyEvent_Get_KeyPowerOff() != M_ON) && (uc_turn_off_flg != M_ON)) || (FlowProcessDisplay_Get_PurgeFlag() == true))		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-
	if(uc_m_batt_buz_flg != M_OFF)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	else if(uc_m_mode_led_flg != M_OFF)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	○	○	○	○	○	○	○
	else		○	-	-	-	-	-	-	-	-	○	○	○	○	○	○	-	-	○	○	○	○	○	○	○
	if(uc_alm_pin > ALARM_PTN_BOOT) || (uc_alm_pin == ALARM_PTN_OFF))		-	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	○	○	○	○	○	○	○
	else		○	-	-	-	-	-	-	-	-	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
	if(uc_vb_lp_flg > ALARM_PTN_BOOT)			-	-	-	-	-	-	-	-	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-
	else if(((uc_alm_pin == ALARM_PTN_MODE_LED) || (uc_alm_pin == ALARM_PTN_BAT1_BUZ)) && (uc_vb_lp_flg != ALARM_PTN_OFF))			-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		Symbol	Range																							
Input	uc_m_boot_status		0-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	KeyEvent_Get_KeyPowerOff()		0-1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GasFuncAr_Disp_Get_ArCalFlg()		0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
	FlowMain_Get_ModeStatus()		0-7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	m_Check_AlarmCondition()		0x00-0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0x01	0x02	0x00
	ZPCNumGet_Get_GasTestFlg()		0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	FlowProcessDisplay_Get_PurgeFlag()		0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
	uc_turn_off_flg		0-1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	us_m_test_sdm_flg		0~2000	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	2000
	us_m_test_type_flg		0~3	0	0	0	0	0	0	0	0	0	3	1	2	0	0	0	0	0	0	0	0	0	0	0
	uc_fail_bz		0-1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	uc_mandown_bz		0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	stp_status->uc_man_down		0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x1F	0x01	0x10	0x00	0x00	0x00
	uc_vb_lp_flg		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_m_short_bz_flg		0~8	0	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_m_batt_buz_flg		0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	uc_m_mode_led_flg		0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	uc_alm_pin		0~10	10	0	0	0	0	0	0	0	0	9	9	9	9	9	7	7	7	0	0	0	0	0	39
Output	uc_vb_lp_flg		0~18	10	1	2	3	4	5	6	9	0	18	13	14	0	11	8	7	0	17	16	15	0	13	14	0
	uc_m_boot_status		0,2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_alm_pin		0~18	10	1	2	3	4	5	6	9	0	18	13	14	9	11	8	7	7	17	16	15	0	13	14	0
	uc_m_FLG_ALARM_CYC_REQ_chk		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	uc_m_short_bz_flg		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Judgment				OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_get_param

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification time	21
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17
Root	if(alarm_ptn == ALARM_PTN_W)		-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	if(!is_tbl_end)		-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(pst_tmfo->uc_w_pateln_flg == 0U)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	○	-	-	-	○	○	○	○	○	○	○	○	○	○	○	○
	for(bus_devidr=(U_SHORT)0; bus_devidr<(U_SHORT)ALARM_DEV_MAX; bus_devidr++)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(*pTmp_alm_time == 0U)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(*pTmp_alm_time == 0U)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(*pTmp_alm_time != (U_CHR)TERMINATER_TIC_VALUE)		-	-	○	○	-	-	○	○	○	○	○	○	○	○	○	○	○
	if(*pTmp_alm_time == 0U)		-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(*pTmp_alm_time == 0U)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	○	○	○	○	○	○	○	○	○	○	○
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	for(bus_devidr=(U_SHORT)0; bus_devidr<(U_SHORT)ALARM_DEV_MAX; bus_devidr++)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(pst_tmfo->cur_tar_tic[bus_devid] != (U_CHR)TERMINATER_TIC_VALUE)		○	○	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	switch(anm_ctl_tbl[alarm_ptn].bz_freq_typ)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case BUZZER_FREQ_TYP_1		-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-
	if(bz_tbl_alm2[pst_tmfo->bz_tbl_alm_idx] == (U_SHORT)TERMINATER_BZ_TBL)		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case BUZZER_FREQ_TYP_2		-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
	if(bz_tbl_alm2[pst_tmfo->bz_tbl_alm_idx] == (U_SHORT)TERMINATER_BZ_TBL)		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case BUZZER_FREQ_TYP_3		-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-
	if(bz_tbl_alm3[pst_tmfo->bz_tbl_alm_idx] == (U_SHORT)TERMINATER_BZ_TBL)		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	case BUZZER_FREQ_TYP_4		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-
	if(bz_tbl_alm4[pst_tmfo->bz_tbl_alm_idx] == (U_SHORT)TERMINATER_BZ_TBL)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	case BUZZER_FREQ_TYP_STATIC		○	○	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	default		-	-	○	-	○	-	-	-	-	-	-	-	-	-	-	-	○
	if(bz_tbl_alm0[pst_tmfo->bz_tbl_alm_idx] == (U_SHORT)TERMINATER_BZ_TBL)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	else		-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	Symbol	Range																	
Input	p_alarm_ptn	0~18	0	0	18	18	18	0	5	5	14	14	17	17	11	14	13	13	17
	is_tbl_end	0~2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	2
	pst_tmfo->uc_w_pateln_flg	0~2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2
	pst_tmfo->cur_tar_tic[bus_devid]	0~255	0	0	1	1	255	2	2	2	2	2	2	2	2	2	2	2	3
	pst_tmfo->bz_tbl_alm_idx	-1~255	0	0	-1	255	255	255	0	255	0	50	0	50	0	49	0	255	-2
Output	tmfo->uc_w_pateln_flg	0~2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2
	is_tbl_end	0,1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	pst_tmfo->bz_tbl_alm_idx	0~51	0	0	0	0	0	0	0	0	1	0	1	51	0	50	1	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_judge_request

	Responsible	Last update
Coding	Ikarashi	2021/6/10
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification time	22
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20	case21	case22
Root	switch(uc_alm_ptn)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	case 1		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 3		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 4		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 5		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 6		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 7		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 8		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 9		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case 10		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case 11		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case 12		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case 13		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case 14		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case 15		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case 16		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case 17		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	case 18		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	○
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if(tmp_alm_ptn == ALARM_PTN_OFF)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-
	if(pst_tmfo->short_bz_req != ALARM_PTN_OFF)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○
	if(*alarm_ptn != tmp_alm_ptn)		-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	○	○
	if(tmp_alm_ptn == ALARM_PTN_W)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range																						
Input	uc_alm_ptn	0~19	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0	18
	p_alarm_ptn[0]	0~19	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	18	19	16	16
	short_bz_req	0~19	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	18	19	0	0
Output	uc_alm_ptn	0~16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	13	14	15	16	17
	pst_tmfo->short_bz_req	0~19	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	18	19	0	0
	p_alarm_ptn[0]	0~19	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0	18
	uc_w_pateln_flg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	ComCommand.c

ComCommand_Init_Process

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(lus_i=F_INIT_VAL; lus_i<F_RECEIVE_BUFFER_SIZE; lus_i++)		○
	for(lus_i=F_INIT_VAL; lus_i<F_SEND_BUFFER_SIZE; lus_i++)		○
	Symbol	Range	
Input	None	-	-
Output	fus_receive_count	0	0
	fnc_receive_end_char	4	4
	fbl_receive_end_flg	0	0
	fbl_Active_flg	0	0
Judgment			OK

ComCommand_Check_UsbComNonPartner

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luss_non_partner_count != F_INIT_VAL)		○	-	○
	if(Port_Get_UsbPort_isOn() == false)		○	○	-
	if(luss_non_partner_count == F_INIT_VAL)		-	-	○
	else		○	○	-
	Symbol	Range			
Input	Port_Get_UsbPort_isOn()	0,1	0	0	1
	luss_non_partner_count	0,1	1	0	1
Output	luss_non_partner_count	0,720	720	720	0
Judgment			OK	OK	OK

ComCommand_Chk_ReceiveSumFault

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(lus_i=F_INIT_VAL; lus_i<puc_buf_size; lus_i++)		○	-	○	○	○
	if(*lncp_r_buf_p == COMDATA_STX)		○	-	○	-	-
	if(lbl_com_start == true)		○	-	○	-	-
	if(*lncp_r_buf_p != COMDATA_ETX)		○	-	○	-	○
	else		-	-	-	○	-
	if(luc_calc_sum == luc_rxd_sum)		-	○	-	○	○
	else		○	-	○	-	-
	Symbol	Range					
Input	pnc_buf[0]	2~4	2	2	2	3	4
	puc_buf_size	0~2	1	0	2	1	1
Output	lbl_ans_fault	0,1	1	0	1	0	0
Judgment			OK	OK	OK	OK	OK

ComCommand_Error_ReceiveData

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(lus_i=F_INIT_VAL; lus_i<F_RECEIVE_BUFFER_SIZE; lus_i++)		○
	Symbol	Range	
Input	None	-	-
Output	fnc_receive_data[0]	0	0
	fus_receive_count	0	0
Judgment			OK

ComCommand_Get_DeviceStatus

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case1	case1
Root	None		-	-	-
	Symbol	Range			
Input	fbl_Active_flg	0~2	0	1	2
Output	fbl_Active_flg	0~2	0	1	2
Judgment			OK	OK	OK

ComCommand_Get_DeviceTransmitEnable

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_reg != F_INIT_VAL)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	UartSci5_Get_TransmitEnable()	0~2	1	0	2
Output	lbl_ans	0,1	1	0	1
Judgment			OK	OK	OK

ComCommand_Get_ReceiveData

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(fbl_Active_flg == true)		○	-	○	○	○	○	○	○
	if(fus_receive_count > F_RECEIVE_COUNTER_MAX_SIZE)		○	-	-	-	-	-	-	-
	if((lnc_data == COMDATA_EOT) || (lnc_data == COMDATA_ETB) || (lnc_data == COMDATA_CR))		○	-	○	-	○	-	○	-
	Symbol	Range								
Input	puc_data	4~24	4	4	4	5	23	24	13	14
	fbl_Active_flg	1,2	1	2	1	1	1	1	1	1
	fus_receive_count	638,639	639	639	638	638	638	638	638	638
Output	fus_receive_count	639	639	639	639	639	639	639	639	639
	fbl_receive_end_flg	1	1	1	1	1	1	1	1	1
	fnc_receive_data	0	0	0	0	0	0	0	0	0
	fnc_receive_end_char	4~23	4	4	4	4	23	23	13	13
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

ComCommand_Get_UsbComPartner

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(Port_Get_UsbPort_isOn() == false)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	Port_Get_UsbPort_isOn()	0~2	0	1	2
Output	lbl_ans	0,1	1	0	0
Judgment			OK	OK	OK

ComCommand_Process_AnalysisData

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(fbl_Active_flg == true)		○	-	○	○
	if(fbl_receive_end_flg == true)		○	-	-	○
	for(luc_i=F_INIT_VAL; luc_i<COMDATA_DATA_POS; luc_i++)		○	-	-	○
	if(lbl_chk_fault == false)		○	-	-	-
	else		-	-	-	○
	Symbol	Range				
Input	ComCommand_Chk_ReceiveSumFault()	0,1	0	0	0	1
	USBCom_Cmd_Proc()	0	0	0	0	0
	fbl_Active_flg	1,2	1	2	1	1
	fbl_receive_end_flg	1,2	1	1	2	1
	fnc_receive_data[0]	0	0	0	0	0
Output	fbl_receive_end_flg	0~2	0	1	2	0
	fnc_receive_data[10]	0,21	0	0	0	21
	fnc_send_data[0]	0	0	0	0	0
	fus_receive_count	0	0	0	0	0
Judgment			OK	OK	OK	OK

ComCommand_Start_Device

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_boot_status	1	1
Judgment			OK

ComCommand_Stop_Device

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_boot_status	0	0
Judgment			OK

f_Set_SendData

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	*pnc_buff	0	0
	pus_size	0	0
	f_Sum_Calc()	0	0
	Tool_Hex_Asc()	0	0
	R_SCI5_Serial_Send()	0	0
Output	None	-	-
Judgment			OK

f_Sum_Calc

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_i=F_INIT_VAL; lus_i<pus_size; lus_i++)		-	-	-
	Symbol	Range			
Input	*pnc_buff	1	1	1	1
	pnc_buf[0]	0	0	0	0
	pus_size	0~2	1	0	2
Output	luc_ans	0,1	0	0	1
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	DA12bit.c

DA12bit_Init_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fuss_percentage	0	0
Judgment			OK

DA12bit_Set_OutputNum

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case4
Root	if(pus_num < DA12BIT_PERCENTAGE_MAX_SIZE)		○	○	-
	else		-	-	○
	Symbol	Range			
Input	pus_num	0~1000	0	999	1000
	f_Calc_RegisterNum()	0,1	0	0	1
	fuss_percentage	0,1	0	0	1
Output	fuss_percentage	0~1000	0	999	1000
Judgment			OK	OK	OK

DA12bit_Start_Output

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DA12bit_Stop_Output

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

f_Calc_RegisterNum

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pus_percentage	0,255	0	255
Output	lus_ans	0,1044	0	1044
Judgment			OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	DA12bitSelfCheck.c

DA12bitSelfCheck_Init_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fbl_active_flag	0	0
	fuc_check_wait	2	2
Judgment			OK

DA12bitSelfCheck_Set_CheckStart

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pbl_flag	0~2	0	1	2
Output	fbl_active_flag	0~2	0	1	2
Judgment			OK	OK	OK

DA12bitSelfCheck_Set_CheckWaitTimer

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fuc_check_wait	2	2
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	data_flash.c

DataFlash_flash_write

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if((pul_flash_address + (U_LNG)pus_write_byte) -1 <= 0x000FFFFF)		○	○	○	○	○	○	○	○	○
	for(lus_i=0; lus_i<(M_FLASH_RECEIVE_MAX); lus_i++)		○	○	○	-	○	○	○	○	○
	if (puc_err_write == 0x00)		○	-	○	-	○	○	○	-	○
	if(luc_ack == M_FLASH_OK)		○	-	-	-	○	○	○	-	○
	for(lus_i=0; lus_i<pus_write_byte; lus_i++)		○	-	-	-	○	○	○	-	○
	if(uc_m_flash_receive_buf[lus_i + M_DATA_POS] != 0xFF)		○	-	-	-	○	○	○	-	○
	else		-	-	○	-	-	-	-	-	-
	for(lus_i=0; lus_i<(M_FLASH_RECEIVE_MAX); lus_i++)		○	○	○	-	○	○	○	○	○
	if(luc_ack == M_FLASH_OK)		-	○	-	-	-	-	-	○	-
	if(luc_ack == M_FLASH_OK)		-	○	-	-	-	-	-	○	-
	if (puc_err_write == 0x00)		-	○	-	-	-	-	-	-	-
	if(luc_ack == M_FLASH_OK)		-	○	-	-	-	-	-	○	-
	while(ucc_m_wait_count)		-	○	-	-	-	-	-	○	-
	if((luc_ack == M_FLASH_OK) && (luc_status == 0x00)		-	○	-	-	-	-	-	○	-
	if(ucc_m_wait_count == 0)		-	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-
	Symbol	Range									
Input	*pucp_sram_addres	0	0	0	0	0	0	0	0	0	0
	pul_flash_addres	0	0	0	0	0	0	0	0	0	0
	pus_write_byte	0~2	1	1	1	0	2	1	1	1	1
	puc_err_write	0,1	0	1	0	0	0	0	0	1	0
	m_flash_write_enable()	0,1	0	0	0	0	0	0	1	0	0
	flash_status_read(&luc_status, 0)	0,1	0	0	0	0	0	0	0	0	1
	uc_m_flash_receive_buf[lus_i + M_DATA_POS]	0,1	0	0	1	0	0	0	0	0	0
	ucc_m_wait_count	0	0	0	0	0	0	0	0	0	0
Output	uc_m_flash_send_buf[lus_i]	0	0	0	0	0	0	0	0	0	0
	uc_m_flash_receive_buf[lus_i]	0,2	0	2	0	0	0	0	0	2	0
	ucc_m_wait_count	0,2	0	2	0	0	0	0	0	2	0
	m_cmd_ResponsTerm()	0,1	1	0	1	1	1	1	1	0	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

Flash_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(ucc_m_wait_count)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	ucc_m_wait_count	0~3	0	2	3
Output	None	0~2	0	1	2
Judgment			OK	OK	OK

flash_Block_Erase

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((puc_type < 3) &&(((puc_type == 0) && (pul_flash_address <= 0x000FF000) && ((pul_flash_address % 0x001000) == 0)) ||((puc_type == 1) && (pul_flash_address <= 0x000F8000) && ((pul_flash_address % 0x008000) == 0)) || ((puc_type == 2) && (pul_flash_address <= 0x000F0000) && ((pul_flash_address % 0x010000) == 0))))		○	○	-	-
	for(lus_i=0; lus_i<5; lus_i++)		○	○	-	-
	if(luc_ack == M_FLASH_OK)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	pul_flash_address	0x00000000~0xFFFFFFFF	0x00001000	0x00010000	0x00000000	0xFFFFFFFF
	puc_type	0~3	0	2	1	3
	m_flash_write_enable()	0~2	0	1	1	2
Output	uc_m_flash_send_buf[lus_i]	32~216	32	216	82	82
	luc_ack	0,1	0	1	1	1
Judgment			OK	OK	OK	OK

flash_read

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((((pul_flash_address + (U_LNG)pus_read_byte) - 1) <= 0x000FFFFF) && (pus_read_byte <= M_FLASH_SEND_DATA))		○	○	-
	for(lus_i=0; lus_i<M_FLASH_RECEIVE_MAX; lus_i++)		○	○	-
	if(luc_ack == M_FLASH_OK)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	pul_flash_address	0x00000000~0xFFFFFFFF	0x00000001	0x00000000	0xFFFFFFFF
	pus_read_byte	0x0000~0xFFFF	0x0000	0x0001	0xFFFF
	RSPI_Send_Receive(uc_m_flash_send_buf, (pus_read_byte + M_DATA_POS), uc_m_flash_receive_buf, M_FLASH_CH)	0,1	0	1	2
Output	uc_m_flash_send_buf[lus_i]	3	3	3	3
	uc_m_flash_receive_buf[lus_i]	0	0	0	0
	luc_ack	0,1	0	1	1
Judgment			OK	OK	OK

flash_status_read

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_type < 2)		○	○	-
	for(lus_i=0; lus_i<2; lus_i++)		○	○	-
	if(luc_ack == M_FLASH_OK)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_type	0~2	0	1	2
	RSPI_Send_Receive(uc_m_flash_status_send_buf, 2, uc_m_flash_status_receive_buf, M_FLASH_CH)	0~2	0	0	2
Output	uc_m_flash_status_send_buf[lus_i]	5,53	5	53	53
	uc_m_flash_status_receive_buf[lus_i]	0	0	0	0
	luc_ack	0,1	0	0	1
Judgment			OK	OK	OK

m_Chg_Address_LNG_CHR

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pul_16_address	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	-	-	-	-
Judgment			OK	OK

m_flash_write_enable

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	RSPI_Send_Receive(luc_send_buf, 1, luc_receive_buf, M_FLASH_CH)	0,1	0	1	2
Output	luc_ack	0~2	0	1	2
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	data_fram.c

DataFram_AlarmReset_Load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8
Root	if(fram_ar_data[0].uc_alarm_reset_wcount[0] != fram_ar_data[0].uc_alarm_reset_wcount[1])	-	-	○	○	-	○	○	○
	else	○	○	-	-	○	-	-	-
	if(fram_ar_data[1].uc_alarm_reset_wcount[0] != fram_ar_data[1].uc_alarm_reset_wcount[1])	-	-	○	-	○	○	○	○
	else	○	○	-	○	-	-	-	-
	if(luc_ret == 0)	○	○	○	○	-	-	-	-
	if(luc_ans == 0)	○	○	-	-	-	-	-	-
	if(fram_ar_data[0].uc_alarm_reset_wcount[0] >= fram_ar_data[1].uc_alarm_reset_wcount[0])	○	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-
	if((luc_sort[0] == 0xff)&&(luc_sort[1] == 0x00))	-	○	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-
	else if((luc_ans & 0x0f) && (luc_ans & 0xf0))	-	-	○	-	-	-	-	-
	else if(luc_ans & 0x0f)	-	-	-	○	-	-	-	-
	else	-	-	-	-	○	-	-	-
	else if(luc_ack[0] == 0)	-	-	-	-	-	○	-	-
	else if(luc_ack[1] == 0)	-	-	-	-	-	-	○	-
	else	-	-	-	-	-	-	-	○
	if(luc_side_select == 0xff)	-	-	○	-	-	-	-	○
	else	○	○	-	○	○	-	-	-
	for(lus_i = 0; lus_i < sizeof(FRAM_AR_DATA); lus_i++)	○	○	-	○	○	-	-	-
	Symbol	Range							
Input	m_FRAM_AlarmReset_Side_Load (&fram_ar_data[0], 0)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF
	m_FRAM_AlarmReset_Side_Load (&fram_ar_data[1], 1)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF
	fram_ar_data[0].uc_alarm_reset_wcount[0]	0x00,0xFF	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00
	fram_ar_data[0].uc_alarm_reset_wcount[1]	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF
	fram_ar_data[1].uc_alarm_reset_wcount[0]	0x00,0xFF	0x00	0xFF	0xFF	0x00	0xFF	0x00	0x00	0x00
	fram_ar_data[1].uc_alarm_reset_wcount[1]	0x00,0xFF	0x00	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF
Output	luc_ret	0x00~0xFF	0x00	0x00	0xFE	0x00	0x00	0x0F	0xFF	0xFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

DataFram_AlarmReset_Write

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	FRAM_AR_DATA.uc_alarm_reset_wcount[0]	0x00,0xFF	0x00	0xFF	0xFF
	m_fram_alarm_reset_sum_calc(&FRAM_AR_DATA)	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	m_FRAM_AlarmReset_Save(0)	0~2	0	1	2
	m_FRAM_AlarmReset_Save(1)	0~2	0	1	2
Output	FRAM_AR_DATA.uc_alarm_reset_wcount[0]	0,1	1	0	0
	FRAM_AR_DATA.uc_alarm_reset_wcount[1]	0	0	0	0
	FRAM_AR_DATA.uc_alarm_reset_wcount[2]	85	85	85	85
	FRAM_AR_DATA.uc_alarm_reset_wcount[3]	170	170	170	170
	FRAM_AR_DATA.us_alarm_reset_sum	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	luc_ack[0]	0~2	0	1	2
Judgment			OK	OK	OK

DataFram_FRAM_Diag

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if(FRAM_DATA.us_fram_sum == lus_fram_sum)	○	-	-
	else	-	○	○
	Symbol	Range		
Input	m_fram_sum_calc()	0x0000,0xFFFF	0xFFFF	0x0000	0xFFFF
	FRAM_DATA.us_fram_sum	0x0000,0xFFFF	0xFFFF	0xFFFF	0x0000
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

DataFram_FRAM_Factory_load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8
Root	if(fram_factory[0].uc_fram_wcount[0] != fram_factory[0].uc_fram_wcount[1])	-	-	○	○	-	○	○	○
	else	○	○	-	-	○	-	-	-
	if(fram_factory[1].uc_fram_wcount[0] != fram_factory[1].uc_fram_wcount[1])	-	-	○	-	○	○	○	○
	else	○	○	-	○	-	-	-	-
	if(luc_ret == 0)	○	○	○	○	-	-	-	-
	if(luc_ans == 0)	○	○	-	-	-	-	-	-
	if(fram_factory[0].uc_fram_wcount[0] >= fram_factory[1].uc_fram_wcount[0])	○	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-
	if((luc_sort[0] == 0xff)&&(luc_sort[1] == 0x00))	-	○	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-
	else if((luc_ans & 0x0f) && (luc_ans & 0xf0))	-	-	○	-	-	-	-	-
	else if(luc_ans & 0x0f)	-	-	-	○	-	-	-	-
	else	-	-	-	-	○	-	-	-
	else if(luc_ack[0] == 0)	-	-	-	-	-	○	-	-
	else if(luc_ack[1] == 0)	-	-	-	-	-	-	○	-
	else	-	-	-	-	-	-	-	○
	if(luc_side_select == 0xff)	-	-	○	-	-	-	-	○
	else	○	○	-	○	○	-	-	-
	for(lus_i = 0; lus_i < pus_byte; lus_i++)	○	○	-	○	○	-	-	-
	Symbol	Range							
Input	pul_factory_start_byte	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	pus_byte	0x0000,0xFFFF	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0xFFFF
	m_FRAM_Factory_Side_Load(&fram_factory[0], 0)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF
	m_FRAM_Factory_Side_Load(&fram_factory[1], 1)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF
	fram_factory[0].uc_fram_wcount[0]	0x00,0xFF	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00
	fram_factory[0].uc_fram_wcount[1]	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF
	fram_factory[1].uc_fram_wcount[0]	0x00,0xFF	0x00	0xFF	0xFF	0x00	0xFF	0x00	0x00	0x00
	fram_factory[1].uc_fram_wcount[1]	0x00,0xFF	0x00	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF
Output	luc_ret	0x00,0xFF	0x00	0x00	0xFE	0x00	0x00	0x0F	0xFF	0xFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

DataFram_Get_AlarmResetAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_AR_DATA	-	Indefinite value
Judgment			OK

DataFram_Get_ChkDataAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_DATA.CHK_DATA	-	Indefinite value
Judgment			OK

DataFram_Get_DataLoggerAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_DATA.DATA_LOGGER	-	Indefinite value
Judgment			OK

DataFram_Get_Dumpload

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pus_num <= M_DUMP_SIZE_NUM)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	pus_num	0x0000~0xFFFF	0x0000	0xFFFF	0xFFFF
	m_fram_read((pus_num*(M_FRAM_STACK_SIZE*2)), pucp_set_data, M_FRAM_STACK_SIZE)	0,1	0	1	2
	m_fram_read((pus_num*(M_FRAM_STACK_SIZE*2))+ M_FRAM_STACK_SIZE, pucp_set_data, M_FRAM_STACK_SIZE)	01,	0	1	2
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

DataFram_Get_FramDataAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_DATA	-	Indefinite value
Judgment			OK

DataFram_Get_LunchBreakAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_LB_DATA	-	Indefinite value
Judgment			OK

DataFram_Get_StIDAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_ST_ID_DATA	-	Indefinite value
Judgment			OK

DataFram_Get_UserIDAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&FRAM_USER_ID_DATA	-	Indefinite value
Judgment			OK

DataFram_LunchBreak_Load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8
Root	if(fram_lb_data[0].uc_lb_wcount[0] != fram_lb_data[0].uc_lb_wcount[1])	-	-	○	○	-	○	○	○
	else	○	○	-	-	○	-	-	-
	if(fram_lb_data[1].uc_lb_wcount[0] != fram_lb_data[1].uc_lb_wcount[1])	-	-	○	-	○	○	○	○
	else	○	○	-	○	-	-	-	-
	if(luc_ret == 0)	○	○	○	○	-	-	-	-
	if(luc_ans == 0)	○	○	-	-	-	-	-	-
	if(fram_lb_data[0].uc_lb_wcount[0] >= fram_lb_data[1].uc_lb_wcount[0])	○	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-
	if((luc_sort[0] == 0xff)&&(luc_sort[1] == 0x00))	-	○	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-
	else if((luc_ans & 0x0f) && (luc_ans & 0xf0))	-	-	○	-	-	-	-	-
	else if(luc_ans & 0x0f)	-	-	-	○	-	-	-	-
	else	-	-	-	-	○	-	-	-
	else if(luc_ack[0] == 0)	-	-	-	-	-	○	-	-
	else if(luc_ack[1] == 0)	-	-	-	-	-	-	○	-
	else	-	-	-	-	-	-	-	○
	if(uc_m_side_select_lunchbreak == 0xff)	-	-	○	-	-	-	-	○
	else	○	○	-	○	○	-	-	-
	for(lus_i = 0; lus_i < sizeof(FRAM_LB_DATA); lus_i++)	○	○	-	○	○	-	-	-
	Symbol	Range							
Input	m_FRAM_LunchBreak_Side_Load(&fram_lb_data[0], 0)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF
	m_FRAM_LunchBreak_Side_Load(&fram_lb_data[1], 1)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF
	fram_lb_data[0].uc_lb_wcount[0]	0x00,0xFF	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00
	fram_lb_data[0].uc_lb_wcount[1]	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF
	fram_lb_data[1].uc_lb_wcount[0]	0x00,0xFF	0x00	0xFF	0xFF	0x00	0xFF	0x00	0x00	0x00
	fram_lb_data[1].uc_lb_wcount[1]	0x00,0xFF	0x00	0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF
Output	uc_m_side_select_lunchbreak	0x00~0xFF	0x00	0x00	0xFF	0x01	0x00	0x00	0x01	0xFF
	luc_ret	0x00~0xFF	0x00	0x00	0xFE	0x00	0x00	0x0F	0xFF	0xFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

DataFram_LunchBreak_Write

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if(uc_m_side_select_lunchbreak == 0x00)	○	-	-
	else if(uc_m_side_select_lunchbreak == 0x01)	-	○	-
	else	-	-	○
	Symbol	Range		
Input	FRAM_LB_DATA.uc_lb_wcount[0]	0x00,0xFF	0x00	0x00	0xFF
	uc_m_side_select_lunchbreak	0x00~0xFF	0x01	0x00	0xFF
	m_fram_lunchbreak_sum_calc(&FRAM_LB_DATA)	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	m_FRAM_LunchBreak_Save(0)	0,1	1	1	0
	m_FRAM_LunchBreak_Save(1)	0,1	1	1	0
Output	FRAM_LB_DATA.uc_lb_wcount[0]	0,1	1	1	0
	FRAM_LB_DATA.us_lunch_break_sum	0x0000	0x0000	0x0000	0x0000
	uc_m_side_select_lunchbreak	0,1	0	1	0
	luc_ack[0]	0	1	1	0
Judgment			OK	OK	OK

DataFram_write_factory_fram

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	m_fram_sum_calc()	0x0000~0xFFFF	0x0000	0xFFFF	0xFFFF
	m_fram_factory_save(0)	0~2	0	1	2
	m_fram_factory_save(1)	0~2	0	1	2
Output	FRAM_DATA.us_fram_sum	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	luc_ack[0]	0~2	0	1	2
Judgment			OK	OK	OK

FRAM_Get_DefaultAnsFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

		case1
Root	None	-
	Symbol	Range	
Input	None	-	-
Output	uc_default_chk_ans	0	0
Judgment			OK

FRAM_Get_DefaultEndFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(default_flg == DataFram_DEFAULT_OFF)		○	-	-
	else if(default_flg == DataFram_DEFAULT_ERROR)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	default_flg	0~8	0	7	8
Output	luc_ret	0,1	0	0	1
Judgment			OK	OK	OK

FRAM_Get_flg

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_fram_write_flg	0	0
Judgment			OK

FRAM_ID_load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lul_i = 0; lul_i < (sizeof(FRAM_USER_ID_DATA) / M_FRAM_STACK_SIZE); lul_i++)		○	○	○
	if((sizeof(FRAM_USER_ID_DATA) % M_FRAM_STACK_SIZE) != 0)		○	-	-
	else		-	○	○
	for(lul_i = 0; lul_i < (sizeof(FRAM_ST_ID_DATA) / M_FRAM_STACK_SIZE); lul_i++)		○	-	○
	if((sizeof(FRAM_ST_ID_DATA) % M_FRAM_STACK_SIZE) != 0)			○	-
	else		-	-	○
	Symbol	Range			
Input	m_fram_read((((U_LNG)&FRAM_USER_ID_VOL.uc_ur_id) + (lul_i * M_FRAM_STACK_SIZE)), ((U_CHR*)&FRAM_USER_ID_DATA.uc_ur_id + (lul_i * M_FRAM_STACK_SIZE)), (sizeof(FRAM_USER_ID_DATA) % M_FRAM_STACK_SIZE))	0~2	1	0	2
	m_fram_read((((U_LNG)&FRAM_ST_ID_VOL.uc_st_id) + (lul_i * M_FRAM_STACK_SIZE)), ((U_CHR*)&FRAM_ST_ID_DATA.uc_st_id + (lul_i * M_FRAM_STACK_SIZE)), (sizeof(FRAM_ST_ID_DATA) % M_FRAM_STACK_SIZE))	0~2	1	0	2
Output	luc_ack	0~2	1	0	2
Judgment			OK	OK	OK

FRAM_id_write_op

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_status->us_main_state & (DataStructConst_FAIL_FRM | DataStructConst_FAIL_ROM | DataStructConst_FAIL_RAM)) == 0)		○	-	-
	else		-	○	○
	if(luc_status == M_NG)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	puc_select_id	0,1	0	0	1
	puc_idx	0~129	0	128	129
	m_write_id_data_fram(puc_select_id, puc_idx)	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FRAM_Init_DataAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FRAM_load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(fram[0].uc_fram_wcount[0] != fram[0].uc_fram_wcount[1])		-	-	○	○	-	○	○	○
	else		○	○	-	-	○	-	-	-
	if(fram[1].uc_fram_wcount[0] != fram[1].uc_fram_wcount[1])		-	-	○	-	○	○	○	○
	else		○	○	-	○	-	-	-	-
	if(luc_ret == 0)		○	○	○	○	-	-	-	-
	if(luc_ans == 0)		○	○	-	-	-	-	-	-
	if(fram[0].uc_fram_wcount[0] >= fram[1].uc_fram_wcount[0])		○	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-
	if((luc_sort[0] == 0xff)&&(luc_sort[1] == 0x00))		-	○	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-
	else if((luc_ans & 0x0f) && (luc_ans & 0xf0))		-	-	○	-	-	-	-	-
	else if(luc_ans & 0x0f)		-	-	-	○	-	-	-	-
	else		-	-	-	-	○	-	-	-
	else if(luc_ack[0] == 0)		-	-	-	-	-	○	-	-
	else if(luc_ack[1] == 0)		-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	○
	if(uc_m_side_select == 0xff)		-	-	○	-	-	-	-	○
	else		○	○	-	○	○	-	-	-
	for(lus_i = 0; lus_i < sizeof(FRAM_DATA); lus_i++)		○	○	-	○	○	-	-	-
	Symbol	Range								
Input	m_FRAM_Side_Load((U_CHR *)&fram[0].us_fram_st_size, 0)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF
	m_FRAM_Side_Load((U_CHR *)&fram[1].us_fram_st_size, 1)	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF
	fram[0].uc_fram_wcount[0]	0x00,0xFF	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00
	fram[0].uc_fram_wcount[1]	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF
	fram[1].uc_fram_wcount[0]	0x00,0xFF	0x00	0xFF	0xFF	0x00	0xFF	0x00	0x00	0x00
	fram[1].uc_fram_wcount[1]	0x00,0xFF	0x00	0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF
Output	uc_m_side_select	0x00~0xFF	0x00	0x00	0xFF	0x01	0x00	0x00	0x01	0xFF
	luc_ret	0x00~0xFF	0x00	0x00	0xFE	0x00	0x00	0x0F	0xFF	0xFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FRAM_Set_ChkData_Default

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FRAM_Set_Default

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->us_fram_st_size	0x0000	0x0000
	stp_data->uc_fram_prg_ver[0]	48	48
	stp_data->uc_fram_prg_ver[1]	48	48
	stp_data->uc_fram_prg_ver[2]	48	48
	stp_data->uc_fram_prg_ver[3]	48	48
	stp_data->uc_fram_prg_ver[4]	48	48
	stp_data->uc_fram_dumy1	0x00	0x00
	stp_data->uc_fram_wcount[0]	0x00	0x00
	stp_data->uc_fram_wcount[1]	0x00	0x00
	stp_data->us_fram_sum	0x0000	0x0000
Judgment			OK

FRAM_Set_DefaultStartFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	default_flg	1	1
Judgment			OK

FRAM_Set_FailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FRAM_Set_flg

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_fram_write_flg	1	1
Judgment			OK

FRAM_Set_ID_Default

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FRAM_Set_MainDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	switch(default_flg)		○	○	○	○	○	○	-	-
	case DataFram_DEFAULT_OFF		○	-	-	-	-	-	-	-
	case DataFram_DEFAULT_FRAM_DATA		-	○	-	-	-	-	-	-
	case DataFram_DEFAULT_LOGGER		-	-	○	○	-	-	-	-
	if(LoggerGXForce_Set_LoggerDefault() == M_OFF)		-	-	○	-	-	-	-	-
	else		-	-	-	○	-	-	-	-
	case DataFram_DEFAULT_ID_DATA		-	-	-	-	○	-	-	-
	case DataFram_DEFAULT_END		-	-	-	-	-	○	-	-
	case DataFram_DEFAULT_ERROR		-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	○
	if(default_flg != DataFram_DEFAULT_OFF)		-	○	○	○	○	○	○	○
	if(luc_status == M_NG)		-	-	-	○	-	-	-	-
	else		-	○	○	-	○	○	○	○
	else		○	-	-	-	-	-	-	-
	Symbol	Range								
Input	default_flg	0~8	0	1	4	4	5	6	7	8
	LoggerGXForce_Set_LoggerDefault()	0~2	0	0	0	1	0	0	0	2
	m_write_setting_fram()	0~2	0	0	1	0	0	0	0	2
Output	default_flg	0~8	0	4	5	4	6	0	0	8
	uc_m_side_select	0x00~0xFF	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	uc_m_fram_write_flg	0	0	0	0	0	0	0	0	0
	uc_default_chk_ans	0,1	0	0	0	0	0	0	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FRAM_write_op

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(uc_m_fram_write_flg == 1)		○	○	-	-
	if((stp_status->us_main_state & (DataStructConst_FAIL_FRM | DataStructConst_FAIL_ROM | DataStructConst_FAIL_RAM)) == 0)		○	-	-	-
	else		-	○	-	-
	if(luc_status == M_NG)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	uc_m_fram_write_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF
	m_write_setting_fram()	0~2	1	0	0	2
Output	uc_m_fram_write_flg	0,2	0	0	0	2
Judgment			OK	OK	OK	OK

m_fram_alarm_reset_sum_calc

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(lus_i = 0 ; lus_i < stp_data->us_alarm_reset_st_size ; lus_i++)		-	○
	Symbol	Range		
Input	stp_data->us_alarm_reset_st_size	0x0000~0xFFFF	0x0000	0xFFFF
Output	lus_ans	0x0000	0x0000	0x0000
Judgment			OK	OK

m_FRAM_AlarmReset_Save

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_num = 0; lus_num < sizeof(FRAM_AR_DATA); lus_num++)		○	○	○
	if(pus_side == 0)		○	-	-
	else		-	○	○
	for(lus_turn=0; lus_turn<lus_fram_freq; lus_turn++)		○	-	-
	if(lus_fram_rem != 0)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pus_side	0~2	0	1	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_ar_pnt, M_FRAM_STACK_SIZE)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, M_FRAM_STACK_SIZE)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_ar_pnt, lus_fram_rem)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, lus_fram_rem)	0~2	1	0	2
Output	luc_status	0~2	1	0	2
Judgment			OK	OK	OK

m_FRAM_AlarmReset_Side_Load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(lul_i = 0; lul_i < (sizeof(FRAM_AR_DATA) / M_FRAM_STACK_SIZE); lul_i++)		○	○	○	○	○	○
	if(puc_side == 0)		○	-	○	○	-	○
	else		-	○	-	-	○	-
	for(lul_j = 0; lul_j < M_FRAM_STACK_SIZE; lul_j++)		○	○	○	○	○	○
	if((*lucp_ram_pnt + luc_data[lul_j]) != 0xff)		○	○	○	○	○	○
	else		○	-	-	-	-	-
	if((sizeof(FRAM_AR_DATA) % M_FRAM_STACK_SIZE) != 0)		○	-	○	○	○	-
	if(puc_side == 0)		-	-	○	○	-	-
	else		-	-	-	-	○	-
	for(lul_j = 0; lul_j < (sizeof(FRAM_AR_DATA) % M_FRAM_STACK_SIZE); lul_j++)		○	-	○	○	○	-
	if((*lucp_ram_pnt + luc_data[lul_j]) != 0xff)		○	-	○	○	○	-
	else		-	-	-	-	○	-
	else		-	○	-	-	-	○
	if(stp_data->us_alarm_reset_st_size == sizeof(FRAM_AR_DATA.ALARM_POINT))		-	-	○	○	-	-
	if(lus_sum != stp_data->us_alarm_reset_sum)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		○	○	-	-	○	○
	Symbol	Range						
Input	puc_side	0~2	0	1	0	0	2	0
	stp_data->us_alarm_reset_st_size	0x0000~0xFFFF	0x0033	0x0000	0x00A0	0x00A0	0x0033	0xFFFF
	m_fram_alarm_reset_sum_calc(stp_data)	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF	0x0000	0x0000
	stp_data->us_alarm_reset_sum	0x0000,0xFFFF	0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF	0x0000
Output	luc_ack	80	80	80	80	80	80	80
Judgment			OK	OK	OK	OK	OK	OK

m_fram_factory_save

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_num = 0; lus_num < sizeof(FRAM_DATA); lus_num++)		○	○	○
	if(pus_side == 0)		○	-	-
	else		-	○	○
	for(lus_turn=0; lus_turn<lus_fram_freq; lus_turn++)		○	-	-
	if(lus_fram_rem != 0)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pus_side	0,2	0	1	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_pnt, M_FRAM_STACK_SIZE)	0,2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, M_FRAM_STACK_SIZE)	0,2	1	0	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_pnt, lus_fram_rem)	0,2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, lus_fram_rem)	0,2	1	0	2
Output	luc_status	0~2	1	0	2
Judgment			OK	OK	OK

m_FRAM_Factory_Side_Load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(lul_i = 0; lul_i < (sizeof(FRAM_DATA) / M_FRAM_STACK_SIZE); lul_i++)		○	○	○	○	○	○
	if(puc_side == 0)		○	-	○	○	-	○
	else		-	○	-	-	○	-
	for(lul_j = 0; lul_j < M_FRAM_STACK_SIZE; lul_j++)		○	○	○	○	○	○
	if((*lucp_ram_pnt + luc_data[lul_j]) != 0xff)		○	○	○	○	○	○
	else		○	-	-	-	-	-
	if((sizeof(FRAM_DATA) % M_FRAM_STACK_SIZE) != 0)		○	-	○	○	○	-
	if(puc_side == 0)		-	-	○	○	-	-
	else		-	-	-	-	○	-
	for(lul_j = 0; lul_j < (sizeof(FRAM_DATA) % M_FRAM_STACK_SIZE); lul_j++)		○	-	○	○	○	-
	if((*lucp_ram_pnt + luc_data[lul_j]) != 0xff)		○	-	○	○	○	-
	else		-	-	-	-	○	-
	else		-	○	-	-	-	○
	if(stp_data->us_fram_st_size == sizeof(FRAM_DATA.CHK_DATA))		-	-	○	○	-	-
	if(lus_sum != stp_data->us_fram_sum)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		○	○	-	-	○	○
	Symbol	Range						
Input	puc_side	0~2	0	1	0	0	2	0
	stp_data->us_fram_st_size	0x0000~0xFFFF	0x0033	0x0000	0x0678	0x0678	0x0033	0xFFFF
	m_fram_factory_sum_calc(stp_data)	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF	0x0000	0x0000
	stp_data->us_fram_sum	0x0000,0xFFFF	0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF	0x0000
Output	luc_ack	80	80	80	80	80	80	80
Judgment			OK	OK	OK	OK	OK	OK

m_fram_factory_sum_calc

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(lus_i = 0 ; lus_i < stp_data->us_fram_st_size ; lus_i++)		-	○
	Symbol	Range		
Input	stp_data->us_fram_st_size	0x0000,0xFFFF	0x0000	0xFFFF
Output	lus_ans	0x0000	0x0000	0x0000
Judgment			OK	OK

m_fram_id_save

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_select_id == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_select_id	0~2	0	1	2
	puc_idx	0x00,0xFF	0x00	0x00	0xFF
	m_write_and_check_fram(lul_set_add, lucp_fram_id_pnt, luc_write_byte)	0~2	0	1	2
Output	luc_status	0~2	0	1	2
Judgment			OK	OK	OK

m_fram_id_save_all

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_turn=0; lus_turn<lus_fram_freq; lus_turn++)		○	○	○
	if(lus_fram_rem != 0)		○	-	○
	else		-	○	-
	for(lus_turn=0; lus_turn<lus_fram_freq; lus_turn++)		○	○	○
	if(lus_fram_rem != 0)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	m_write_and_check_fram(lul_set_add, lucp_fram_pnt, M_FRAM_STACK_SIZE)	0,1	0	1	2
	m_write_and_check_fram(lul_set_add, lucp_fram_pnt, lus_fram_rem)	0,1	0	1	2
Output	luc_status	0~2	0	1	2
Judgment			OK	OK	OK

m_FRAM_LunchBreak_Save

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_num = 0; lus_num < sizeof(FRAM_LB_DATA); lus_num++)		○	○	○
	if(pus_side == 0)		○	-	-
	else		-	○	○
	for(lus_turn=0; lus_turn<lus_fram_freq; lus_turn++)		○	-	-
	if(lus_fram_rem != 0)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pus_side	0~2	0	1	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_lb_pnt, M_FRAM_STACK_SIZE)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, M_FRAM_STACK_SIZE)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_lb_pnt, lus_fram_rem)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, lus_fram_rem)	0~2	1	0	2
Output	luc_status	0~2	1	0	2
Judgment			OK	OK	OK

m_FRAM_LunchBreak_Side_Load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(lul_i = 0; lul_i < (sizeof(FRAM_LB_DATA) / M_FRAM_STACK_SIZE); lul_i++)		○	○	○	○	○	○
	if(puc_side == 0)		○	-	○	○	-	○
	else		-	○	-	-	○	-
	for(lul_j = 0; lul_j < M_FRAM_STACK_SIZE; lul_j++)		○	○	○	○	○	○
	if((*lucp_ram_pnt + luc_data[lul_j]) != 0xff)		○	○	○	○	○	○
	else		○	-	-	-	-	-
	if((sizeof(FRAM_LB_DATA) % M_FRAM_STACK_SIZE) != 0)		○	-	○	○	○	-
	if(puc_side == 0)		-	-	○	○	-	-
	else		-	-	-	-	○	-
	for(lul_j = 0; lul_j < (sizeof(FRAM_LB_DATA) % M_FRAM_STACK_SIZE); lul_j++)		○	-	○	○	○	-
	if((*lucp_ram_pnt + luc_data[lul_j]) != 0xff)		○	-	○	○	○	-
	else		-	-	-	-	○	-
	else		-	○	-	-	-	○
	if(stp_data->us_lunch_break_st_size == sizeof(FRAM_LB_DATA.st_param))		-	-	○	○	-	-
	if(lus_sum != stp_data->us_lunch_break_sum)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		○	○	-	-	○	○
	Symbol	Range						
Input	puc_side	0~2	0	1	0	0	2	0
	stp_data->us_lunch_break_st_size	0x0000~0xFFFF	0x0033	0x0000	0x00C8	0x00C8	0x0033	0xFFFF
	m_fram_lunchbreak_sum_calc(stp_data)	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF	0x0000	0x0000
	stp_data->us_lunch_break_sum	0x0000,0xFFFF	0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF	0x0000
Output	luc_ack	80	80	80	80	80	80	80
Judgment			OK	OK	OK	OK	OK	OK

m_fram_lunchbreak_sum_calc

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(lus_i = 0 ; lus_i < stp_data->us_lunch_break_st_size ; lus_i++)		-	○
	Symbol	Range		
Input	stp_data->us_lunch_break_st_size	0x0000,0xFFFF	0x0000	0xFFFF
Output	lus_ans	0x00	0x00	0x00
Judgment			OK	OK

m_fram_read

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pus_byte <= M_FRAM_STACK_SIZE)		○	○	-
	for(luc_i=0; luc_i<(M_FRAM_STACK_SIZE+3); luc_i++)		○	○	-
	if(luc_chk != M_OK)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	pus_byte	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	RSPI_Send_Receive(luc_fram_send_data, (pus_byte + 3), (U_CHR *)&luc_fram_ receive_data[0], M_FRAM_CH)	0~2	1	0	2
Output	luc_chk	1	1	1	1
Judgment			OK	OK	OK

m_fram_save

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_num = 0; lus_num < sizeof(FRAM_DATA); lus_num++)		○	○	○
	if(pus_side == 0)		○	-	-
	else		-	○	○
	for(lus_turn=0; lus_turn<lus_fram_freq; lus_turn++)		○	-	-
	if(lus_fram_rem != 0)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pus_side	0~2	0	1	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_pnt, M_FRAM_STACK_SIZE)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, M_FRAM_STACK_SIZE)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add0, lucp_fram_pnt, lus_fram_rem)	0~2	1	0	2
	m_write_and_check_fram(lul_set_add1, lucp_rev_pnt, lus_fram_rem)	0~2	1	0	2
Output	luc_status	0~2	1	0	2
Judgment			OK	OK	OK

m_FRAM_Side_Load

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(lul_i = 0; lul_i < (sizeof(FRAM_DATA) / M_FRAM_STACK_SIZE); lul_i++)		○	○	○	○	○	○
	if(puc_side == 0)		○	-	○	○	-	○
	else		-	○	-	-	○	-
	for(lul_j = 0; lul_j < M_FRAM_STACK_SIZE; lul_j++)		○	○	○	○	○	○
	if((*lucp_frm_pnt + luc_data[lul_j]) != 0xff)		○	○	○	○	○	○
	else		○	-	-	-	-	-
	if((sizeof(FRAM_DATA) % M_FRAM_STACK_SIZE) != 0)		○	-	○	○	○	-
	if(puc_side == 0)		-	-	○	○	-	-
	else		-	-	-	-	○	-
	for(lul_j = 0; lul_j < (sizeof(FRAM_DATA) % M_FRAM_STACK_SIZE); lul_j++)		○	-	○	○	○	-
	if((*lucp_frm_pnt + luc_data[lul_j]) != 0xff)		○	-	○	○	○	-
	else		-	-	-	-	○	-
	else		-	○	-	-	-	○
	if(FRAM_DATA.us_fram_st_size == sizeof(FRAM_DATA.CHK_DATA))		-	-	○	○	-	-
	if(lus_sum != FRAM_DATA.us_fram_sum)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		○	○	-	-	○	○
	for(lul_i = 0; lul_i < sizeof(FRAM_DATA); lul_i++)		○	○	○	○	○	○
	Symbol	Range						
Input	puc_side	0~2	0	1	0	0	2	0
	FRAM_DATA.us_fram_st_size	0x0000~0xFFFF	0x0033	0x0000	0x0678	0x0678	0x0033	0xFFFF
	m_fram_sum_calc()	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF	0x0000	0x0000
	FRAM_DATA.us_fram_sum	0x0000,0xFFFF	0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF	0x0000
Output	luc_ack	80	80	80	80	80	80	80
Judgment			OK	OK	OK	OK	OK	OK

m_fram_sum_calc

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(lus_i = 0 ; lus_i < FRAM_DATA.us_fram_st_size ; lus_i++)		○	○
	Symbol	Range		
Input	FRAM_DATA.us_fram_st_size	0x00,0xFF	0x00	0xFF
Output	lus_ans	-	0	Indefinite value
Judgment			OK	OK

m_fram_write

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<(M_FRAM_STACK_SIZE+3); luc_i++)		○	○	○
	if(luc_chk != M_OK)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	puc_byte	0x00,0xFF	0x00	0x00	0xFF
	RSPI_Send_Receive(luc_fram_send_data, ((U_SHORT)puc_byte + 3), luc_fram_receive_data, M_FRAM_CH)	0~2	1	0	2
Output	luc_chk	0,1	1	0	1
Judgment			OK	OK	OK

m_write_and_check_fram

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	while((luc_write_counter < M_WRITE_RETRY_COUNT) && (luc_read_counter < M_READ_RETRY_COUNT) && (luc_verify_counter < M_VERIFY_RETRY_COUNT))		○	○	○	○	○
	if(luc_write_success == 0U){		○	○	-	-	-
	if(luc_ret != M_OK)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	if(luc_err_flg == M_OFF)		-	○	○	○	-
	if(luc_ret != M_OK)		-	○	-	-	-
	else		-	-	-	-	-
	else		○	-	○	○	○
	if(luc_err_flg == M_OFF)		-	-	○	○	-
	if(Tool_Comp_Pnt(ppuc_write_buff, luc_read_buff, (U_LNG)pus_size) == 0)		-	-	○	-	-
	else		-	-	-	○	-
	else		○	○	-	-	○
	if((luc_write_counter >= M_WRITE_RETRY_COUNT) || (luc_read_counter >= M_READ_RETRY_COUNT) || (luc_verify_counter >= M_VERIFY_RETRY_COUNT))		○	○	○	-	-
	else		-	-	-	-	-
	Symbol	Range					
Input	pul_addr	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	*ppuc_write_buff	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF
	pus_size	0x0000,0xFFFF	0x0000	0x0000	0x0000	0x0000	0xFFFF
	m_fram_write((U_SHORT)pul_addr, ppuc_write_buff, pus_size)	0~2	1	0	1	0	2
	m_fram_read((U_SHORT)pul_addr, luc_read_buff, pus_size)	0~2	1	1	0	0	2
	Tool_Comp_Pnt(ppuc_write_buff, luc_read_buff, (U_LNG)pus_size)	0~2	1	1	0	1	2
Output	luc_ret	1	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK

m_write_id_data_fram

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_select_id	0~2	0	1	2
	puc_idx	0x00,0xFF	0x00	0x00	0xFF
	m_fram_id_save(puc_select_id, puc_idx)	0~2	0	1	2
Output	luc_ack	0~2	0	1	2
Judgment			OK	OK	OK

m_write_setting_fram

	Responsible	Last update
Coding	Ikarashi	2021/6/11
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(uc_m_side_select == 0x00)		○	-	-
	else if(uc_m_side_select == 0x01)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	FRAM_DATA.uc_fram_wcount[0]	0x00,0xFF	0x00	0x00	0xFF
	m_fram_sum_calc()	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	uc_m_side_select	0x00~0xFF	0x01	0x00	0xFF
Output	FRAM_DATA.us_fram_st_size	1400	1400	1400	1400
	FRAM_DATA.uc_fram_wcount[0]	0,1	1	1	0
	FRAM_DATA.uc_fram_wcount[1]	0	0	0	0
	FRAM_DATA.uc_fram_wcount[2]	85	85	85	85
	FRAM_DATA.uc_fram_wcount[3]	170	170	170	170
	FRAM_DATA.us_fram_sum	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	uc_m_side_select	0,1	0	1	0
	luc_ack[0]	0	0	0	0
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	data_fram_factory.c

DataFramFactry_Alarmpoint_load

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○
	Symbol	Range			
Input	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_1st_alarm	0~1001	0	1000	1001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_2nd_alarm	0~25001	0	25000	25001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_3rd_alarm	0~50001	0	50000	50001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_stel_alarm	0~5001	0	5000	5001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_twa_alarm	0~2501	0	2500	2501
Output	(stp_GasInfo_addr+luc_i)->ul_1st_alarm	0	0	0	0
	(stp_GasInfo_addr+luc_i)->ul_2nd_alarm	0	0	0	0
	(stp_GasInfo_addr+luc_i)->ul_3rd_alarm	0	0	0	0
	(stp_GasInfo_addr+luc_i)->ul_stel_alarm	0	0	0	0
	(stp_GasInfo_addr+luc_i)->ul_twa_alarm	0	0	0	0
	luc_status	0	0	0	0
Judgment			OK	OK	OK

DataFramFactry_All_load

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	DataFram_FRAM_Factory_load (luc_fram_load_addr, lul_factory_start_byte, (U_SHORT)sizeof(struct st_sum_chk_rom))	0x00,0xFF	0x00	0xFF
Output	luc_status	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

DataFramFactry_Chk_AlarmReset_load

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	for(luc_i = 0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->ul_full_scale != stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_full_scale)		-	-	-	-	-	-	○	-	-
	if((stp_GasInfo_addr+luc_i)->ul_digit != stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_digit)		-	-	-	-	-	-	-	○	-
	if((stp_GasInfo_addr+luc_i)->uc_uni_dec != stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].uc_uni_dec)		-	-	-	○	-	-	-	-	-
	if(stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_1st_alarm != GASTABLEDATA_OFF_VAL)		-	○	○	-	○	○	-	-	○
	if((stp_GasInfo_addr+luc_i)->ul_gas_alarm_max < stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_1st_alarm || (stp_GasInfo_addr+luc_i)->ul_gas_alarm_min > stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_1st_alarm)		-	-	○	-	-	-	-	-	-
	if(stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_2nd_alarm != GASTABLEDATA_OFF_VAL)		○	○	-	-	○	○	-	-	○
	if((stp_GasInfo_addr+luc_i)->ul_gas_alarm_max < stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_2nd_alarm || (stp_GasInfo_addr+luc_i)->ul_gas_alarm_min > stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_2nd_alarm)		○	-	-	-	-	-	-	-	-
	if(stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_3rd_alarm != GASTABLEDATA_OFF_VAL)		-	○	-	-	○	○	-	-	○
	if((stp_GasInfo_addr+luc_i)->ul_gas_alarm_max < stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_3rd_alarm || (stp_GasInfo_addr+luc_i)->ul_gas_alarm_min > stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_3rd_alarm)		-	-	-	-	-	○	-	-	-
	if(stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_stel_alarm != GASTABLEDATA_OFF_VAL)		-	○	-	-	○	-	-	-	○
	if((stp_GasInfo_addr+luc_i)->ul_stel_alarm_max < stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_stel_alarm || (stp_GasInfo_addr+luc_i)->ul_stel_alarm_min > stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_stel_alarm)		-	○	-	-	-	-	-	-	-
	if(stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_twa_alarm != GASTABLEDATA_OFF_VAL)		-	-	-	-	○	-	-	-	○
	if((stp_GasInfo_addr+luc_i)->ul_twa_alarm_max < stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_twa_alarm || (stp_GasInfo_addr+luc_i)->ul_twa_alarm_min > stp_AlarmReset_addr->ALARM_POINT.RESET_ALARM[luc_i].ul_twa_alarm)		-	-	-	-	○	-	-	-	-
	if(luc_ans == M_OFF)		○	○	○	○	○	○	○	○	-
	else		-	-	-	-	-	-	-	-	○
	Symbol	Range									
Input	ALARM_POINT.RESET_ALARM[0].ul_full_scale	0	0	0	0	0	0	0	0	0	0
	ALARM_POINT.RESET_ALARM[0].ul_digit	0	0	0	0	0	0	0	0	0	0
	ALARM_POINT.RESET_ALARM[0].uc_uni_dec	0	0	0	0	0	0	0	0	0	0
	ALARM_POINT.RESET_ALARM[0].ul_1st_alarm	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD
	ALARM_POINT.RESET_ALARM[0].ul_2nd_alarm	0xDDDDDDDD~0xDDDDDDDF	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDF	0xDDDDDDDF	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDF	0xDDDDDDDF	0xDDDDDDDD
	ALARM_POINT.RESET_ALARM[0].ul_3rd_alarm	0xDDDDDDDD~0xDDDDDDDF	0xDDDDDDDF	0xDDDDDDDE	0xDDDDDDDF	0xDDDDDDDF	0xDDDDDDDE	0xDDDDDDDF	0xDDDDDDDF	0xDDDDDDDF	0xDDDDDDDD
	ALARM_POINT.RESET_ALARM[0].ul_stel_alarm	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD
	ALARM_POINT.RESET_ALARM[0].ul_twa_alarm	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD
	ul_full_scale	0,1	0	0	0	0	0	0	1	0	0
	ul_digit	0,1	0	0	0	0	0	0	1	1	0
	uc_uni_dec	0,1	0	0	0	1	0	0	1	1	0
	ul_gas_alarm_min	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ul_gas_alarm_max	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
	ul_stel_alarm_min	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ul_stel_alarm_max	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ul_twa_alarm_min	0xDDDDDDDE,0xDDDDDDDF	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ultwa_alarm_max	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
Output	uc_m_alarm_reset_onoff_flg	0,1	0	0	0	0	0	0	0	0	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

DataFramFactry_Chk_AlarmResetOnOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_alarm_reset_onoff_flg	1	1
Judgment			OK

DataFramFactry_write_op

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if((stp_status->us_main_state & (DataStructConst_FAIL_FRM | DataStructConst_FAIL_ROM)) == 0)		○	○	-
	if(luc_status == M_NG)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	DataFram_write_factory_fram()	0~2	1	0	2
Output	luc_status	0~2	1	0	2
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	data_logger.c

DataLogger_air_adjustment

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)		○	○	○	○	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)		○	○	○	-	-	-
	for(lus_i=0; lus_i<M_MAX_PORT; lus_i++)		○	○	○	-	-	-
	if((luc_port & puc_portCode) == luc_port)		○	○	-	-	-	-
	for(lus_k=0; lus_k<M_MAX_CHANNEL; lus_k++)		○	○	-	-	-	-
	if(st_m_ch_datas[lus_k].uc_port == lus_i)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		-	-	-	-	○	○
	Symbol	Range						
Input	puc_portCode	0x00~0xFF	0x01	0x01	0x01	0x00	0x80	0xFF
	st_datalogger_control.uc_state	0x00~0xFF	0x01	0x01	0x00	0x00	0xFF	0xFF
	st_m_ch_datas[lus_k].uc_port	0x00,0xFF	0x00	0xFF	0xFF	0x00	0xFF	0xFF
Output	uc_m_keep_auto_zero	0,1	0	0	1	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

DataLogger_bump

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(puc_portCode != 0)		○	○	○	○	-	○
	do		○	○	○	○	-	○
	do		○	○	○	○	-	○
	for(lus_i=lus_is; lus_i<M_MAX_PORT; lus_i++)		○	○	○	○	-	○
	if((luc_port & puc_portCode) == luc_port)		○	○	○	○	-	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	-	○	○	-	○
	if(st_m_ch_datas[luc_ch].uc_port == lus_i)		-	○	○	○	-	○
	if((luc_port & puc_portPass) == luc_port)		○	-	-	-	-	○
	else		-	○	○	○	-	-
	if(puc_fast_flg == 1)		-	○	-	-	-	-
	else		-	-	○	○	-	-
	if(uc_m_max_same_calibration < uc_m_same_calibration_time)		○	-	○	○	-	○
	if (luc_cnt>0)		○	○	○	○	-	○
	for(lus_i=0; lus_i<M_PAGE_SIZE-luc_cnt; lus_i++)		○	○	○	○	-	○
	if(uc_m_same_calibration_time > 1)		○	○	○	-	-	○
	while(lus_is < M_MAX_PORT)		○	○	○	○	-	○
	while(uc_m_calibration_fail == 1)		-	-	-	-	-	-
	if(uc_m_calibration_fail > 1)		-	○	-	○	-	○
	Symbol	Range						
Input	puc_portCode	0~6	1	1	5	5	0	6
	puc_portPass	0~6	1	0	0	0	5	6
	puc_fast_flg	0~2	0	1	0	0	0	2
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0xFF	0xFF	0xFF	0x00	0xFF
	uc_m_max_same_calibration	0x00,0xFF	0x00	0xFF	0x00	0xFF	0x00	0xFF
	uc_m_calibration_fail	0~3	0	2	0	2	0	3
Output	uc_m_buffer[luc_cnt+lus_i]	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

DataLogger_calibration

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(puc_portCode != 0)		○	-	○	○	○	○	○	○
	do		○	-	○	○	○	○	○	○
	do		○	-	○	○	○	○	○	○
	for(lus_i=lus_is; lus_i<M_MAX_PORT; lus_i++)		○	-	○	○	○	○	○	○
	if((luc_port & puc_portCode) == luc_port)		○	-	○	○	○	○	○	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	-	○	○	○	○	○	○
	if(st_m_ch_datas[luc_ch].uc_port == lus_i)		○	-	○	○	○	○	○	○
	if(m_comp_pnt(st_m_gas_datas[lus_i].uc_cal_data, (stp_m_sensor+lus_i)->uc_cal_date, 6) == M_TRUE)		○	-	○	○	-	○	-	-
	else		-	-	-	-	○	-	○	○
	if(uc_m_max_same_calibration < uc_m_same_calibration_time)		○	-	○	○	○	○	○	○
	if (luc_cnt>0)		○		○	○	○	○	○	○
	for(lus_i=0; lus_i<M_PAGE_SIZE-luc_cnt; lus_i++)		○	-	○	○	○	○	○	○
	if(uc_m_same_calibration_time > 1)		○	-	○	○	○	○	○	○
	while(lus_is < M_MAX_PORT)		○	-	○	○	○	○	○	○
	while(uc_m_calibration_fail == 1)		-	-	-	-	-	-	-	-
	if(uc_m_calibration_fail > 1)		-	-	-	-	-	○	○	-
	Symbol	Range								
Input	puc_portCode	0~2	1	0	2	1	1	1	1	1
	ul_denst	0	0	0	0	0	0	0	0	0
	uc_cal_date	1	1	1	1	1	1	1	1	1
	ul_cal_before	0	0	0	0	0	0	0	0	0
	ul_cal_after	0	0	0	0	0	0	0	0	0
	m_comp_pnt()	0~2	1	1	1	1	2	1	2	2
	m_lg_calibration_data()	0	0	0	0	0	0	0	0	0
	m_lg_header_data()	0	0	0	0	0	0	0	0	0
	uc_port	0,1	0	0	0	1	1	0	1	1
	uc_cal_data	1	1	1	1	1	1	1	1	1
	uc_m_buffer	255	255	255	255	255	255	255	255	255
	uc_m_calibration_fail	0x00,0xFF	0x00	0xFF	0x00	0x00	0xFF	0x00	0x00	0xFF
	uc_m_max_same_calibration	0~2	1	0	0	0	0	0	0	2
	uc_m_same_calibration_time	0x00,0xFF	0x00	0xFF	0x00	0x00	0xFF	0x00	0x00	0xFF
	uc_calibration_cnt	0~3	0	0	2	2	0	2	2	3
Output	uc_m_buffer	255	255	255	255	255	255	255	255	255
	uc_m_max_same_calibration	0,33	33	0	33	33	33	33	33	33
	uc_m_same_calibration_time	0,1	1	0	1	1	1	1	1	1
	uc_calibration_cnt	0,1	0	0	0	0	0	1	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

DataLogger_clear

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)		○	○	○	○	-
	if(luc_step == 0)		○	-	-	-	-
	if((luc_ack == 0x00) && (luc_sts == 0x00))		-	-	-	-	-
	if (luc_step>0		-	-	-	-	-
	if(luc_step == 8)		-	-	-	-	-
	if(luc_step == 0)		-	-	-	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)		-	-	-	-	-
	Symbol	Range					
Input	st_datalogger_control.uc_state	0x00~0xFF	0x00	0x01	0x00	0x00	0xFF
	luc_step	0~15	0	1	1	14	15
	flash_status_read(&luc_sts, 0)	0~2	0	0	0	1	2
Output	st_datalogger_control.uc_state	0x00~0xFF	0x04	0x01	0x00	0x00	0xFF
	ul_m_flash_erase_address	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	stp_DataLogger_addr->us_logger_count	0	0	0	0	0	0
	stp_DataLogger_addr->us_interval_page_pnt	0	0	0	0	0	0
	stp_DataLogger_addr->uc_interval_page_address	0	0	0	0	0	0
	stp_DataLogger_addr->uc_alarm_head_page_pnt	0	0	0	0	0	0
	stp_DataLogger_addr->uc_calibration_cnt	0,4	4	0	0	0	0
	stp_DataLogger_addr->uc_alarm_event_cnt	0	0	0	0	0	0
	stp_DataLogger_addr->uc_trouble_cnt	0	0	0	0	0	0
	stp_DataLogger_addr->uc_over_write	0	0	0	0	0	0
	stp_DataLogger_addr->uc_snap_rec_pnt	0	0	0	0	0	0
	st_datalogger_control.uc_lunch_break	0	0	0	0	0	0
	st_datalogger_control.uc_cmd_sts	1~15	9	1	1	14	15
	luc_step	1~15	9	1	1	14	15
Judgment			OK	OK	OK	OK	OK

DataLogger_clearPower

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)		○	○	○	○	-
	if(luc_step == 0)		○	-	-	-	-
	if((luc_ack == 0x00) && (luc_sts == 0x00))		-	-	-	-	-
	if(luc_step == 2)		-	-	-	-	-
	else if(luc_step == 1)		-	-	-	-	-
	if(luc_step == 0)		-	-	-	-	-
	Symbol	Range					
Input	st_datalogger_control.uc_state	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF
	luc_step	0~15	0	1	14	14	15
	flash_status_read(&luc_sts, 0)	0~2	0	0	0	1	2
Output	st_datalogger_control.uc_state	0x00~0xFF	0x04	0x00	0x00	0x00	0xFF
	luc_step	1~15	2	1	14	14	15
	ul_m_flash_erase_address	0x00	0x00	0x00	0x00	0x00	0x00
	stp_DataLogger_addr->us_power_count	0	0	0	0	0	0
	stp_DataLogger_addr->uc_power_cnt	0	0	0	0	0	0
	stp_DataLogger_addr->uc_setting_cnt	0	0	0	0	0	0
	st_datalogger_control.uc_cmd_sts	0~15	2	1	14	14	15
	uc_m_power_cnt	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

DataLogger_default

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(luc_step == 0)		○	-	-	-
	if((luc_ack == 0x00) && (luc_sts == 0x00))		-	-	-	-
	if(luc_step>1)		-	-	-	-
	else if (luc_step == 1)		-	-	-	-
	Symbol	Range				
Input	luc_step	0~15	0	1	14	15
	flash_status_read(&luc_sts, 0)	0~2	0	0	1	2
Output	st_datalogger_control.uc_state	0x04	0x04	0x04	0x04	0x04
	ul_m_flash_erase_address	0x00	0x00	0x00	0x00	0x00
	luc_step	1~15	11	1	14	15
	st_datalogger_control.uc_cmd_sts	1~15	11	1	14	15
Judgment			OK	OK	OK	OK

DataLogger_Init_EepAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	*stp_data	0x0000,0xFFFF	0x0000	0xFFFF
Output	stp_DataLogger_addr	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

DataLogger_peak_clear

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DataLogger_power_on

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if((st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_INIT) == DATALOGGER_ER_NO_INIT)		○	-	-
	do		○	○	○
	while(uc_m_power_fail == 1)		-	-	-
	if(uc_m_power_fail > 1)		○	-	○
	Symbol	Range			
Input	uc_m_power_fail	0~3	2	0	3
	st_datalogger_control.uc_error_state	0x00,0xFF	0xFF	0x00	0x00
Output	ul_m_pass_count	0	0	0	0
Judgment			OK	OK	OK

DataLogger_Set_Default

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1
Root	for(lul_i=0; lul_i<sizeof(DataLogger_Default); lul_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DataLogger_setting_check

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

		case1	case2	case3	case4
Root	for(luc_i=0; luc_i<M_MAX_PORT; luc_i++)	○	○	○	○
	if(st_m_last_gas_comb[luc_i].uc_gas_comb != (stp_m_sensor+luc_i)->uc_gas_comb)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].uc_sub_comb != (stp_m_sensor+luc_i)->uc_sub_comb)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].uc_uni_dec != (stp_m_sensor+luc_i)->uc_uni_dec)	○	○	○	-
	else	-	-	-	○
	if(m_comp_pnt(st_m_last_gas_comb[luc_i].uc_name[1], (stp_m_sensor+luc_i)->uc_name[1], M_GAS_NAME_SIZE) != M_TRUE)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].ul_1st_alarm != (stp_m_sensor+luc_i)->ul_1st_alarm)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].ul_2nd_alarm != (stp_m_sensor+luc_i)->ul_2nd_alarm)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].ul_3rd_alarm != (stp_m_sensor+luc_i)->ul_3rd_alarm)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].ul_stel_alarm !=(stp_m_sensor+luc_i)->ul_stel_alarm)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].ul_twa_alarm != (stp_m_sensor+luc_i)->ul_twa_alarm)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].ul_cal_denst != (stp_m_sensor+luc_i)->ul_cal_denst)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].uc_cal_group != (stp_m_sensor+luc_i)->uc_cal_group)	○	○	○	-
	else	-	-	-	○
	if(m_comp_pnt(st_m_last_gas_comb[luc_i].uc_notice_date, (stp_m_sensor+luc_i)->uc_notice_date, 3) != M_TRUE)	○	○	○	-
	else	-	-	-	○
	if(m_comp_pnt(st_m_last_gas_comb[luc_i].uc_chg_sen_date, (stp_m_sensor+luc_i)->uc_chg_sen_date, 3) != M_TRUE)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].uc_suppress_onoff != (stp_m_sensor+luc_i)->uc_suppress_onoff)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_gas_comb[luc_i].uc_follower_onoff != (stp_m_sensor+luc_i)->uc_follower_onoff)	○	○	○	-
	else	-	-	-	○
	if(m_comp_pnt(st_m_last_setup.uc_sep_no, stp_m_setting-> uc_sep_no, HDDEF_SEP_NO_SIZE) != M_TRUE)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_exp_flg != stp_m_setting->uc_exp_flg)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.us_back_light_time != stp_m_setting->us_back_light_time)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_auto_back_light != stp_m_setting->uc_auto_back_light)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_date_format != stp_m_setting->uc_date_format)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_lcd_invert != stp_m_setting->uc_lcd_invert)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_lcd_back_ground != stp_m_setting->uc_lcd_back_ground)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_lcd_contrast != stp_m_setting->uc_lcd_contrast)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_long_energy_flg != stp_m_setting->uc_long_energy_flg)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_lel_sel_flg != stp_m_setting->uc_lel_sel_flg)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_margin_disp_onoff != stp_m_setting->uc_margin_disp_onoff)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_ir_unit != stp_m_setting->uc_ir_unit)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_ir_air_onoff != stp_m_setting->uc_ir_air_onoff)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_up_cal_flg != stp_m_setting->uc_up_cal_flg)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_alarm_pattern != stp_m_setting->uc_alarm_pattern)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_1h_alarm_on != stp_m_setting->uc_1h_alarm_on)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_alarm_on != stp_m_setting->uc_alarm_on)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_stealth_on != stp_m_setting->uc_stealth_on)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_stealth_motor_on != stp_m_setting->uc_stealth_motor_on)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_buzz_volume != stp_m_setting->uc_buzz_volume)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_beep_select != stp_m_setting->uc_beep_select)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.uc_beep_time != stp_m_setting->uc_beep_time)	○	○	○	-
	else	-	-	-	○
	if (st_m_last_setup.uc_disp_pump_off != stp_m_setting->uc_disp_pump_off)	○	○	○	-
	else	-	-	-	○
	if (st_m_last_setup.uc_alarm_silence != stp_m_setting->uc_alarm_silence)	○	○	○	-
	else	-	-	-	○
	if (st_m_last_setup.uc_leak_buzzer_onoff != stp_m_setting->uc_leak_buzzer_onoff)	○	○	○	-
	else	-	-	-	○
	if (st_m_last_setup.uc_leak_fs_dp != stp_m_setting->uc_leak_fs_dp)	○	○	○	-
	else	-	-	-	○
	if (st_m_last_setup.uc_leak_mode_flg != stp_m_setting->uc_leak_mode_flg)	○	○	○	-
	else	-	-	-	○
	if(st_m_last_setup.us_cal_lmt_day != stp_m_setting->us_cal_lmt_day)	○	○	○	-
	else	-	-	-	○

if(st_m_last_setup.uc_cal_lmt_set != stp_m_setting->uc_cal_lmt_set)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_cal_lmt_chk != stp_m_setting->uc_cal_lmt_chk)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_cal_lmt_gas != stp_m_setting->uc_cal_lmt_gas)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.us_bump_lmt_day != stp_m_setting->us_bump_lmt_day)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_lmt_set != stp_m_setting->uc_bump_lmt_set)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_lmt_chk != stp_m_setting->uc_bump_lmt_chk)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_lmt_gas != stp_m_setting->uc_bump_lmt_gas)	○	○	○	-
else	-	-	-	○
if(m_comp_pnt(st_m_last_setup.uc_chg_batt_date, stp_m_setting->uc_chg_batt_date, 3) != M_TRUE)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_mainte_notice_onoff != stp_m_setting->uc_mainte_notice_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_mainte_lmt_chk != stp_m_setting->uc_mainte_lmt_chk)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_mainte_notice_gas != stp_m_setting->uc_mainte_notice_gas)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.us_mainte_notice_day != stp_m_setting->us_mainte_notice_day)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_time != stp_m_setting->uc_bump_time)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_limit != stp_m_setting->uc_bump_limit)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_cal_time != stp_m_setting->uc_bump_cal_time)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_bump_cal_flg != stp_m_setting->uc_bump_cal_flg)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_ble_onoff != stp_m_setting->uc_ble_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_id_disp_onoff != stp_m_setting->uc_id_disp_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_disp_zero_fllwr != stp_m_setting->uc_disp_zero_fllwr)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_disp_suppress_val != stp_m_setting->uc_disp_suppress_val)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_lunch_onoff != stp_m_setting->uc_lunch_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_key_tone_onoff != stp_m_setting->uc_key_tone_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_disp_setting_item_onoff != stp_m_setting->uc_disp_setting_item_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_auto_zero_onoff != stp_m_setting->uc_auto_zero_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_demand_zero_onoff != stp_m_setting->uc_demand_zero_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_user_pass_onoff != stp_m_setting->uc_user_pass_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_mainte_pass_onoff != stp_m_setting->uc_mainte_pass_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.us_user_pass_num != stp_m_setting->us_user_pass_num)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.us_mainte_pass_num != stp_m_setting->us_mainte_pass_num)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_protect_pword_onoff != stp_m_setting->uc_protect_pword_onoff)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.us_interval_time != stp_m_setting->us_interval_time)	○	○	○	-
else	-	-	-	○
if(st_m_last_setup.uc_over_write != stp_m_setting->uc_over_write)	○	○	○	-
else	-	-	-	○
if(luc_hit>0 || stp_DataLogger_addr->uc_setting_cnt == 0)	○	○	○	-
do	○	○	○	-
if(luc_cnt>0)	○	○	○	-
for(lus_i=0; lus_i<M_PAGE_SIZE-luc_cnt; lus_i++)	○	○	○	-
else	-	-	-	-
while(uc_m_setting_fail == 1)	-	-	-	-
if(uc_m_setting_fail > 1)	-	○	-	-
else	-	-	○	-
else	-	-	-	○

	Symbol	Range				
Input	st_m_last_gas_comb[luc_i].uc_gas_comb	0,29	0	29	29	0
	(stp_m_sensor+luc_i)->uc_gas_comb	0,29	29	0	0	0
	st_m_last_gas_comb[luc_i].uc_sub_comb	0,29	0	29	29	0
	(stp_m_sensor+luc_i)->uc_sub_comb	0,29	29	0	0	0
	st_m_last_gas_comb[luc_i].uc_uni_dec	0x00,0x22	0x00	0x22	0x22	0
	(stp_m_sensor+luc_i)->uc_uni_dec	0x00,0x22	0x22	0x00	0x00	0
	m_comp_pnt()	0~2	0	1	2	0
	st_m_last_gas_comb[luc_i].ul_1st_alarm	0,1000	0	1000	1000	0
	(stp_m_sensor+luc_i)->ul_1st_alarm	0,1000	1000	0	0	0
	st_m_last_gas_comb[luc_i].ul_2nd_alarm	0,25000	0	25000	25000	0
	(stp_m_sensor+luc_i)->ul_2nd_alarm	0,25000	25000	0	0	0
	st_m_last_gas_comb[luc_i].ul_3rd_alarm	0,50000	0	50000	50000	0
	(stp_m_sensor+luc_i)->ul_3rd_alarm	0,50000	50000	0	0	0
	st_m_last_gas_comb[luc_i].ul_stel_alarm	0,5000	0	5000	5000	0
	(stp_m_sensor+luc_i)->ul_stel_alarm	0,5000	5000	0	0	0
	st_m_last_gas_comb[luc_i].ul_twa_alarm	0,2500	0	2500	2500	0
	(stp_m_sensor+luc_i)->ul_twa_alarm	0,2500	2500	0	0	0
	st_m_last_gas_comb[luc_i].ul_cal_denst	0,50000	0	50000	50000	0
	(stp_m_sensor+luc_i)->ul_cal_denst	0,50000	50000	0	0	0
	st_m_last_gas_comb[luc_i].uc_cal_group	0x00,0xFF	0x00	0xFF	0xFF	0x00
	(stp_m_sensor+luc_i)->uc_cal_group	0x00,0xFF	0xFF	0x00	0x00	0x00
	st_m_last_gas_comb[luc_i].uc_suppress_onoff	0,1	0	1	1	0
	(stp_m_sensor+luc_i)->uc_suppress_onoff	0,1	1	0	0	0
	st_m_last_gas_comb[luc_i].uc_follower_onoff	0,1	0	1	1	0
	(stp_m_sensor+luc_i)->uc_follower_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_exp_flg	0,3	0	3	3	0
	stp_m_setting->uc_exp_flg	0,3	3	0	0	0
	st_m_last_setup.us_back_light_time	0,255	0	255	255	0
	stp_m_setting->us_back_light_time	0,255	255	0	0	0
	st_m_last_setup.uc_auto_back_light	0,1	0	1	1	0
	stp_m_setting->uc_auto_back_light	0,1	1	0	0	0
	st_m_last_setup.uc_date_format	0,2	0	2	2	0
	stp_m_setting->uc_date_format	0,2	2	0	0	0
	st_m_last_setup.uc_lcd_invert	0,2	0	2	2	0
	stp_m_setting->uc_lcd_invert	0,2	2	0	0	0
	st_m_last_setup.uc_lcd_back_ground	0,1	0	1	1	0
	stp_m_setting->uc_lcd_back_ground	0,1	1	0	0	0
	st_m_last_setup.uc_lcd_contrast	0~50	1	50	50	0
	stp_m_setting->uc_lcd_contrast	0~50	50	1	1	0
	st_m_last_setup.uc_long_energy_flg	0,1	0	1	1	0
	stp_m_setting->uc_long_energy_flg	0,1	1	0	0	0
	st_m_last_setup.uc_lel_sel_flg	0,2	0	2	2	0
	stp_m_setting->uc_lel_sel_flg	0,2	2	0	0	0
	st_m_last_setup.uc_margin_disp_onoff	0,1	0	1	1	0
	stp_m_setting->uc_margin_disp_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_ir_unit	0x00,0xFF	0x00	0xFF	0xFF	0x00
	stp_m_setting->uc_ir_unit	0x00,0xFF	0xFF	0x00	0x00	0x00
	st_m_last_setup.uc_ir_air_onoff	0,1	0	1	1	0
	stp_m_setting->uc_ir_air_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_up_cal_flg	0x00~0x08	0x01	0x08	0x08	0x00
	stp_m_setting->uc_up_cal_flg	0x00~0x08	0x08	0x01	0x01	0x00
	st_m_last_setup.uc_alarm_pattern	0,1	0	1	1	0
	stp_m_setting->uc_alarm_pattern	0,1	1	0	0	0
	st_m_last_setup.uc_1h_alarm_on	0,1	0	1	1	0
	stp_m_setting->uc_1h_alarm_on	0,1	1	0	0	0
	st_m_last_setup.uc_alarm_on	0,1	0	1	1	0
	stp_m_setting->uc_alarm_on	0,1	1	0	0	0
	st_m_last_setup.uc_stealth_on	0,1	0	1	1	0
	stp_m_setting->uc_stealth_on	0,1	1	0	0	0
	st_m_last_setup.uc_stealth_motor_on	0,1	0	1	1	0
	stp_m_setting->uc_stealth_motor_on	0,1	1	0	0	0
	st_m_last_setup.uc_buzz_volume	0,1	0	1	1	0
	stp_m_setting->uc_buzz_volume	0,1	1	0	0	0
	st_m_last_setup.uc_beep_select	0~4	0	4	4	0
	stp_m_setting->uc_beep_select	0~4	4	0	0	0
	st_m_last_setup.uc_beep_time	0,99	0	99	99	0
	stp_m_setting->uc_beep_time	0,99	99	0	0	0
	st_m_last_setup.uc_disp_pump_off	0,1	0	1	1	0
	stp_m_setting->uc_disp_pump_off	0,1	1	0	0	0
	st_m_last_setup.uc_alarm_silence	0,1	0	1	1	0
	stp_m_setting->uc_alarm_silence	0,1	1	0	0	0
	st_m_last_setup.uc_leak_buzzer_onoff	0,1	0	1	1	0
	stp_m_setting->uc_leak_buzzer_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_leak_fs_dp	0,1	0	1	1	0

	stp_m_setting->uc_leak_fs_dp	0,1	1	0	0	0
	st_m_last_setup.uc_leak_mode_flg	0,1	0	1	1	0
	stp_m_setting->uc_leak_mode_flg	0,1	1	0	0	0
	st_m_last_setup.us_cal_lmt_day	0~1000	1	1000	1000	0
	stp_m_setting->us_cal_lmt_day	0~1000	1000	1	1	0
	st_m_last_setup.uc_cal_lmt_set	0,1	0	1	1	0
	stp_m_setting->uc_cal_lmt_set	0,1	1	0	0	0
	st_m_last_setup.uc_cal_lmt_chk	0,2	0	2	2	0
	stp_m_setting->uc_cal_lmt_chk	0,2	2	0	0	0
	st_m_last_setup.uc_cal_lmt_gas	0,1	0	1	1	0
	stp_m_setting->uc_cal_lmt_gas	0,1	1	0	0	0
	st_m_last_setup.us_bump_lmt_day	0~1000	1	1000	1000	0
	stp_m_setting->us_bump_lmt_day	0~1000	1000	1	1	0
	st_m_last_setup.uc_bump_lmt_set	0,1	0	1	1	0
	stp_m_setting->uc_bump_lmt_set	0,1	1	0	0	0
	st_m_last_setup.uc_bump_lmt_chk	0,2	0	2	2	0
	stp_m_setting->uc_bump_lmt_chk	0,2	2	0	0	0
	st_m_last_setup.uc_bump_lmt_gas	0,1	0	1	1	0
	stp_m_setting->uc_bump_lmt_gas	0,1	1	0	0	0
	st_m_last_setup.uc_mainte_notice_onoff	0,1	0	1	1	0
	stp_m_setting->uc_mainte_notice_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_mainte_lmt_chk	0,2	0	2	2	0
	stp_m_setting->uc_mainte_lmt_chk	0,2	2	0	0	0
	st_m_last_setup.uc_mainte_notice_gas	0,1	0	1	1	0
	stp_m_setting->uc_mainte_notice_gas	0,1	1	0	0	0
	st_m_last_setup.us_mainte_notice_day	0~1000	1	1000	1000	0
	stp_m_setting->us_mainte_notice_day	0~1000	1000	1	1	0
	st_m_last_setup.uc_bump_time	0~90	30	90	90	0
	stp_m_setting->uc_bump_time	0~90	90	30	30	0
	st_m_last_setup.uc_bump_limit	0~50	10	50	50	0
	stp_m_setting->uc_bump_limit	0~50	50	10	10	0
	st_m_last_setup.uc_bump_cal_time	0~120	90	120	120	0
	stp_m_setting->uc_bump_cal_time	0~120	120	90	90	0
	st_m_last_setup.uc_bump_cal_flg	0,1	0	1	1	0
	stp_m_setting->uc_bump_cal_flg	0,1	1	0	0	0
	st_m_last_setup.uc_ble_onoff	0,1	0	1	1	0
	stp_m_setting->uc_ble_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_id_disp_onoff	0,1	0	1	1	0
	stp_m_setting->uc_id_disp_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_disp_zero_fllwr	0,1	0	1	1	0
	stp_m_setting->uc_disp_zero_fllwr	0,1	1	0	0	0
	st_m_last_setup.uc_disp_suppress_val	0,1	0	1	1	0
	stp_m_setting->uc_disp_suppress_val	0,1	1	0	0	0
	st_m_last_setup.uc_lunch_onoff	0,1	0	1	1	0
	stp_m_setting->uc_lunch_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_key_tone_onoff	0,1	0	1	1	0
	stp_m_setting->uc_key_tone_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_disp_setting_item_onoff	0,1	0	1	1	0
	stp_m_setting->uc_disp_setting_item_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_auto_zero_onoff	0,1	0	1	1	0
	stp_m_setting->uc_auto_zero_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_demand_zero_onoff	0,1	0	1	1	0
	stp_m_setting->uc_demand_zero_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_user_pass_onoff	0,1	0	1	1	0
	stp_m_setting->uc_user_pass_onoff	0,1	1	0	0	0
	st_m_last_setup.uc_mainte_pass_onoff	0,1	0	1	1	0
	stp_m_setting->uc_mainte_pass_onoff	0,1	1	0	0	0
	st_m_last_setup.us_user_pass_num	0,9999	0	9999	9999	0
	stp_m_setting->us_user_pass_num	0,9999	9999	0	0	0
	st_m_last_setup.us_mainte_pass_num	0,9999	0	9999	9999	0
	stp_m_setting->us_mainte_pass_num	0,9999	9999	0	0	0
	st_m_last_setup.uc_protect_pword_onoff	0,1	0	1	1	0
	stp_m_setting->uc_protect_pword_onoff	0,1	1	0	0	0
	st_m_last_setup.us_interval_time	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF	0x0000
	stp_m_setting->us_interval_time	0x0000,0xFFFF	0xFFFF	0x0000	0x0000	0x0000
	st_m_last_setup.uc_over_write	0,1	0	1	1	0
	stp_m_setting->uc_over_write	0,1	1	0	0	2
	stp_DataLogger_addr->uc_setting_cnt	0x00,0xFF	0x00	0xFF	0xFF	0
	m_lg_setting_data(uc_m_buffer)	0x00,0xFF	0xFF	0x00	0x00	0x00
	uc_m_setting_fail	0~2	1	2	0	0
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

DataLogger_snap

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if ((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)		○	-	○	○	○	○	○	○
	do		○	-	○	○	○	○	○	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	-	○	○	○	○	○	○
	if (stp_m_ad->fl_disp_temp < 0)		○	-	-	-	-	○	○	-
	else		-	-	○	○	○	-	-	-
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	-	○	○	○	○	○	○
	if (luc_port != M_NO_ASSIGN)		○	-	○	○	○	○	○	○
	if ((stp_m_denst+luc_port)->uc_denst_minus_flg) luc_minus_bit |= (1 << luc_port)		○	-	○	○	○	○	○	○
	for(lui_i=0; lui_i<M_PAGE_SIZE-lui_cnt; lui_i++)		○	-	○	○	○	○	○	○
	while(uc_m_snap_fail == 1)		-	-	-	-	-	-	-	-
	if (uc_m_snap_fail > 1)		-	-	-	-	-	-	○	○
	if (stp_DataLogger_addr->uc_snap_rec_pnt == 0)		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	uc_denst_minus_flg	0,1	1	1	1	1	0	1	1	1
	m_lg_header_data()	0	0	0	0	0	0	0	0	0
	m_lg_header_plus()	0	0	0	0	0	0	0	0	0
	m_lg_snap_data()	0	0	0	0	0	0	0	0	0
	m_ui2uc()	0	0	0	0	0	0	0	0	0
	fl_disp_temp	-1,0	-1	-1	0	0	0	-1	-1	-1
	uc_port	254	254	254	255	254	254	254	254	254
	uc_snap_rec_pnt	0,255	255	255	255	255	255	255	0	255
	uc_m_buffer	0	0	0	0	0	0	0	0	0
	uc_m_snap_fail	0,2	0	0	0	0	0	0	2	2
	st_datalogger_control.uc_state	0,128	0	128	0	0	0	0	0	0
Output	uc_snap_rec_pnt	0~255	255	255	255	255	255	255	1	0
	uc_snap_ext_flg	0,1	0	0	0	0	0	0	0	1
	us_snap_write_pnt	0~3	0	0	0	0	0	0	2	3
	uc_m_buffer	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

DataLogger_snap_end_data

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if (stp_DataLogger_addr->uc_snap_ext_flg > 0)		○	-	-
	else		-	○	○
	if (lus_pnt >= M_LG_SNAP_REC_MAX)		-	○	-
	else		-	-	○
	if((luc_buf[0] == 0xEEU) && (luc_buf[1] == 0xEEU))		-	-	-
	else		○	○	○
	Symbol	Range			
Input	flash_read()	0	0	0	0
	m_lg_get_snap_write_pnt()	255,256	256	256	255
	uc_snap_wxt_flg	0,1	1	0	0
Output	luc_ans	0,1	1	0	0
Judgment			OK	OK	OK

DataLogger_snap_read

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if (lus_pnt >= M_LG_SNAP_REC_MAX		○	-
	else		-	○
	Symbol	Range		
Input	puc_pnt	0	0	0
	flash_read	0	0	0
	m_lg_get_snap_write_pnt()	255,256	256	255
	uc_m_buffer	0	0	0
Output	uc_m_buffer	0	0	0
Judgment			OK	OK

DataLogger_start

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

		case1	case2	case3	case4	case5	case6	case7
Root	if (((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL) && ((st_datalogger_control.uc_state & DATALOGGER_RUN) != DATALOGGER_RUN))	○	○	○	○	○	○	-
	if(st_datalogger_control.uc_measure_mode != 0x01)	○	○	-	-	○	-	-
	if((st_datalogger_control.uc_lunch_break != M_ON) && (st_datalogger_control.uc_lunch_break != M_OFF))	○	-	-	-	○	-	-
	else	-	○	-	-	-	-	-
	else	-	-	○	○	-	○	-
	if(us_m_lg_int > 50000)	○	-	-	-	-	-	-
	else	-	○	○	○	○	○	-
	for(lus_i=0; lus_i<M_MAX_CHANNEL; lus_i++)	○	○	○	○	○	○	-
	if(ul_m_logger_count != 0)	○	-	-	-	-	-	-
	else	-	○	○	○	○	○	-
	for(lus_i=0; lus_i<M_MAX_PORT; lus_i++)	○	○	○	○	○	○	-
	if((((stp_m_denst+lus_i)->uc_state_trb & DataStructConst_S_FAIL_SEN) != DataStructConst_S_FAIL_SEN) && ((stp_m_denst+lus_i)->ul_denst != GASTABLEDATA_OFF_VAL))	○	○	○	○	○	○	-
	if((stp_m_denst+lus_i)->uc_denst_minus_flg)	○	-	-	-	-	-	-
	else	○	○	○	○	○	○	-
	else	-	-	○	○	○	○	-
	for(lus_i=0; lus_i<M_MAX_CHANNEL; lus_i++)	○	○	○	○	○	○	-
	if(st_m_ch_datas[lus_i].uc_port != M_NO_ASSIGN)	○	○	○	○	○	○	-
	else	-	○	○	○	○	○	-
	if(stp_m_setting->us_interval_time <= 60)	○	-	-	-	-	-	-
	else	-	○	○	○	○	○	-
	if(stp_DataLogger_addr->us_interval_page_pnt == (M_INTERVAL_PAGES - 1))	○	○	-	-	-	-	-
	if(stp_m_setting->uc_over_write == M_OFF)	○	-	-	-	-	-	-
	else	-	○	-	-	-	-	-
	else	-	○	○	○	○	○	-
	if((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)	○	○	○	-	○	-	-
	if(uc_m_channels != 0)	○	○	○	-	○	-	-
	if(st_datalogger_control.uc_measure_mode != 0x01)	○	○	-	-	○	-	-
	else	-	-	○	-	-	-	-
	else	-	-	○	○	-	-	-
	if(((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL) && ((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN))	○	○	-	-	○	-	-
	if((st_datalogger_control.uc_measure_mode == 0x00) || (st_datalogger_control.uc_measure_mode == 0x03))	○	-	-	-	-	-	-
	else if(st_datalogger_control.uc_measure_mode == 0x02)	-	○	-	-	-	-	-
	else	-	-	-	-	○	-	-
	if((stp_m_setting->uc_auto_zero_onoff == M_ON) && (uc_m_keep_auto_zero != 0))	○	○	-	-	-	-	-
	else	-	-	-	-	○	-	-
	else	-	-	-	-	-	-	-
	else	-	-	-	-	-	○	-
	else	-	-	-	-	-	-	○

	Symbol	Range							
Input	st_datalogger_control.uc_measure_mode	0x00~0xFF	0x00	0x02	0x01	0x01	0x02	0xFF	0xFF
	st_datalogger_control.uc_lunch_break	0~4	3	0	0	0	4	4	4
	stp_DataLogger_addr->us_logger_count	0x0000~0xFFFF	0xFFFE	0x0000	0x0000	0x0000	0x0000	0x0000	0xFFFF
	us_m_lg_int	0x0000,0xFFFF	0xFFFF	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
	ul_m_logger_count	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	(stp_m_denst+lus_i)->uc_state_trb	0x00,0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF
	(stp_m_denst+lus_i)->ul_denst	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0xDDDDDDDD	0x00000000	0x00000000	0x00000000
	(stp_m_denst+lus_i)->uc_denst_minus_flg	0~2	1	0	0	0	2	2	2
	st_m_ch_datas[lus_i].uc_port	0x00,0xFF	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	ul_m_last_interval_checktime	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	stp_m_setting->us_interval_time	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	stp_DataLogger_addr->us_interval_page_pnt	0x011F,0xFFFF	0x011F	0x011F	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	stp_m_setting->uc_over_write	0~2	0	1	1	1	2	2	2
	st_datalogger_control.uc_state	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	uc_m_channels	0~6	5	5	5	0	6	6	6
	stp_m_setting->uc_auto_zero_onoff	0~2	1	1	0	0	2	2	2
	uc_m_keep_auto_zero	0~6	5	5	0	0	6	6	6
Output	us_m_lg_int	0,1	0	1	1	1	1	1	0
	uc_m_last_alm_state[lus_i]	0	0	0	0	0	0	0	0
	uc_m_last_state_trb[lus_i]	0	0	0	0	0	0	0	0
	us_m_last_main_state	0	0	0	0	0	0	0	0
	ul_m_logger_count	0	0	0	0	0	0	0	0
	stp_DataLogger_addr->uc_interval_page_address	0	0	0	0	0	0	0	0
	st_m_alarm[0].uc_trend_page_address	0	0	0	0	0	0	0	0
	st_m_alarm[1].uc_trend_page_address	0	0	0	0	0	0	0	0
	uc_m_alarm_cnt	0	0	0	0	0	0	0	0
	uc_m_FuncTempWarning_Get_OutOfRangeFlg	0	0	0	0	0	0	0	0
	ul_m_tick_total[lus_i]	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	ul_m_tick_count[lus_i]	0,1	1	1	0	0	0	0	0
	st_m_gas_datas[lus_i].uc_invalid	0	0	0	0	0	0	0	0
	st_m_gas_datas[lus_i].uc_onoff	1	1	1	1	1	1	1	1
	ul_m_alarm_peak_datas[lus_i]	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0xDDDDDDDD	0x00000000	0x00000000	0x00000000
	uc_m_alarm_peak_minus[lus_i]	0~2	1	0	0	0	2	2	2
	fl_m_alarm_peak_temp_data	0	0	0	0	0	0	0	0
	fl_m_tick_temp_total	0	0	0	0	0	0	0	0
	ul_m_tick_temp_count	1	1	1	1	1	1	1	1
	ul_m_last_interval_checktime	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	ul_m_last_alarm_checktime	0	0	0	0	0	0	0	0
	st_m_ch_datas[lus_i].ul_write_time	0	0	0	0	0	0	0	0
	st_m_ch_datas[lus_i].ul_concentration	0	0	0	0	0	0	0	0
	st_m_ch_datas[lus_i].uc_count	0	0	0	0	0	0	0	0
	sl_temp_data	0	0	0	0	0	0	0	0
	uc_temp_count	0	0	0	0	0	0	0	0
	ul_temp_write_time	0	0	0	0	0	0	0	0
	us_m_auto_sec	9,003,600	900	3600	3600	3600	3600	3600	3600
	uc_m_interval_own_sec	61,79	79	79	61	61	61	61	61
	st_datalogger_control.uc_state	0x00~0xFF	0x01	0x01	0x00	0x00	0x01	0x80	0xFF
	uc_m_interval_flag	0	0	0	0	0	0	0	0
	uc_m_same_alarm_event_time	0	0	0	0	0	0	0	0
	uc_m_same_trouble_time	0	0	0	0	0	0	0	0
	uc_m_same_calibration_time	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

DataLogger_stop

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((((st_datalogger_control.uc_state & DATALOGGER_TOSTOP) == DATALOGGER_TOSTOP) || ((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)) && ((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL))		○	○	○	○	-	-
	if(luc_step == 0)		○	○	-	-	-	-
	for(lus_i=0; lus_i<uc_m_alarm_cnt; lus_i++)		○	○	-	-	-	-
	if(st_m_alarm[lus_i].uc_trend_page_address !=0)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	○	-	-
	if(luc_step == 1)		○	○	○	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	-	-	-	-
	if(luc_sts == 0x00)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	-	-
	if(luc_step == 2)		○	-	○	-	-	-
	while(uc_m_alarm_cnt > 0)		○	-	-	-	-	-
	else		-	-	-	-	-	-
	if(luc_step == 3)		○	-	○	○	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL)		○	-	-	-	-	-
	if(luc_sts == 0)		○	-	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	-	-
	if((luc_step == 4) || (luc_sts >= 3))		○	-	-	○	-	-
	else		-	-	○	-	-	-
	else		-	○	-	-	-	-
	if(luc_step == 4)		○	-	-	○	-	-
	else		-	○	○	-	-	-
	else		-	-	-	-	○	○
	Symbol	Range						
Input	luc_step	0~15	0	0	1	3	14	15
	st_datalogger_control.uc_state	0x00~0xFF	0x01	0x01	0x01	0x03	0xFF	0x00
	st_m_alarm[lus_i].uc_trend_page_address	0x00~0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x00
	ul_m_flash_erase_address	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00100000	0x00100000	0x00100000	0xFFFFFFFF
	m_flash_status()	0~2	0	1	1	1	1	2
	uc_m_alarm_cnt	0,2	2	2	0	0	0	3
	m_lg_stop_interval()	0~15	0	0	1	14	14	15
Output	st_datalogger_control.uc_state	0x00~0xFF	0x00	0x01	0x01	0x02	0xFF	0x00
	st_datalogger_control.uc_cmd_sts	0~15	0	1	3	0	14	15
	ul_m_logger_count	0	0	0	0	0	0	0
	luc_step	0~15	0	1	3	0	14	15
Judgment			OK	OK	OK	OK	OK	OK

DataLogger_write_test

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=luc_cnt; lui_i<M_PAGE_SIZE; lui_i++)		○
	for(lui_i=0; lui_i<M_PAGE_SIZE; lui_i++)		○
	Symbol	Range	
Input	None	-	-
Output	uc_m_buffer[0]	0x00	0x00
Judgment			OK

m_comp_pnt

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(luc_i=0U;luc_i<pus_num;luc_i++)		○	○	○
	if(pucp_pnt1[luc_i] == pucp_pnt2[luc_i])		○	-	-
	else		-	○	○
	Symbol	Range			
Input	pucp_pnt1[luc_i]	0x00,0xFF	0x00	0xFF	0x00
	pucp_pnt2[luc_i]	0x00,0xFF	0x00	0x00	0xFF
	pus_num	0x0000,0xFFFF	0xFFFF	0x0000	0xFFFF
Output	luc_ret	0,1	0	1	0
Judgment			OK	OK	OK

m_copy_nbyte

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	*ppuc_to	0x00,0xFF	0x00	0xFF
	*ppuc_from	0x00,0xFF	0x00	0xFF
	pnbyte	0x0000,0xFFFF	0x0000	0xFFFF
Output	None	-	-	-
Judgment			OK	OK

m_flash_status

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((luc_ack == 0x00) && (luc_sts == 0x00))		-	-	-	-
	if(uc_m_erase_cnt == 0)		-	-	-	-
	else		-	-	-	-
	else		○	○	○	○
	Symbol	Range				
Input	flash_status_read(&luc_sts, 0)	0~2	0	0	1	2
	uc_m_erase_cnt	0~4	0	3	0	4
Output	ul_m_flash_erase_address	0	0	0	0	0
	ul_m_erase_address	0	0	0	0	0
	luc_ret	1	1	1	1	1
Judgment			OK	OK	OK	OK

m_isLeapYear

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((y % 4) == 0x00)		○	○	○	-	-	○
	if((y % 100) == 0x00)		○	○	-	-	-	○
	if((y % 400) == 0x00)		○	-	-	-	-	-
	else		-	○	-	-	-	○
	else		-	-	○	-	-	-
	else		-	-	-	○	○	-
	Symbol	Range						
Input	y	1~2100	2000	1900	2096	2099	1	2100
Output	plus_day	0,1	1	0	1	1	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_lg_alarm

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(stp_DataLogger_addr->uc_alarm_head_page_pnt == M_LG_ALARM_HEAD_MAX)		○	-	○	○
	if(ul_m_flash_erase_address == M_FLASH_SIZE)		○	○	-	-
	else		-	-	○	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○
	if(st_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)		○	○	-	-
	for(lui_i=0; lui_i<M_MAX_PORT; lui_i++)		○	○	○	○
	Symbol	Range				
Input	m_lg_time_value()	0	0	0	0	0
	m_ui2uc()	0	0	0	0	0
	uc_datetime	0	0	0	0	0
	uc_code	0	0	0	0	0
	uc_port	254,255	254	254	254	255
	uc_alarm_head_page_pnt	9,10	9	10	9	9
	fl_disp_temp	0	0	0	0	0
	ul_denst	0	0	0	0	0
	uc_denst_minus_flg	0	0	0	0	0
	uc_m_alarm_cnt	0	0	0	0	0
	uc_m_now_time	0	0	0	0	0
	ul_m_flash_erase_address	1048576, 1048577	1048576	1048576	1048577	1048577
	us_m_alarm_work_pnt	0	0	0	0	0
Output	uc_trend_page_address	0	0	0	0	0
	uc_head_page_cnt	9,10	9	10	9	9
	us_trend_cnt	361	361	361	361	361
	us_copy_pnt	0	0	0	0	0
	uc_copy_page	0	0	0	0	0
	us_work_pnt	0	0	0	0	0
	uc_fail	0	0	0	0	0
	ul_last_datetime	4294967291	4294967291	4294967291	4294967291	4294967291
	ul_peak_datas	0	0	0	0	0
	uc_peak_minus	0	0	0	0	0
	fl_temp_data	0	0	0	0	0
	uc_alarm_head_page_pnt	0,11	0	11	0	0
	uc_m_alarm_cnt	1	1	1	1	1
	ul_m_flash_erase_address	32768~1048577	65536	32768	1048577	1048577
	ul_m_flash_pending_erase_address	0,65536	0	0	65536	65536
Judgment			OK	OK	OK	OK

m_lg_alarm_event

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	do		○	○	○
	if(luc_cnt>0)		-	-	-
	for(lus_i=0; lus_i<M_PAGE_SIZE-luc_cnt; lus_i++)		-	-	-
	else		○	○	○
	while(uc_m_alarm_event_fail == 1)		-	-	-
	if(uc_m_alarm_event_fail > 1)		○	○	-
	Symbol	Range			
Input	puc_event	0x00,0xFF	0x00	0x00	0xFF
	puc_ch	0~6	0	5	6
	m_lg_alarm_event_data()	0	0	0	0
	m_lg_header_data()	0	0	0	0
	uc_m_event_fail	0,2	2	2	0
	uc_m_buffer	0x00~0xFF	0xFF	0x00	0x00
	uc_m_alarm_event_cnt	0,2	0	0	2
Output	uc_m_buffer	0,255	255	0	0
	uc_alarm_event_cnt	1,2	1	1	2
	uc_m_max_same_alarm_event	127	127	127	127
	uc_m_same_alarm_event_time	1	1	1	1
Judgment			OK	OK	OK

m_lg_alarm_event_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if (uc_m_same_alarm_event_time == 1)		○	-
	else		-	○
	Symbol	Range		
Input	*ppuc_data	0	0	0
	puc_event	0	0	0
	puc_ch	0	0	0
	m_ui2uc()	0	0	0
	uc_m_same_alarm_event_time	1,2	1	2
Output	luc_cnt	2,4	4	2
Judgment			OK	OK

m_lg_alarm_event_fit

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_DataLogger_addr->uc_alarm_event_cnt >= puc_cnt)		-	○	○
	for(lui_i=luc_cnt; lui_i>0; lui_i--)		-	○	○
	if(luc_write_res != 0x00)		-	○	○
	else		-	○	○
	if(lui_i > 0)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	DataFlash_flash_write()	0,1	1	1	0
	puc_cnt	0,1	1	1	0
	flash_read()	0	0	0	0
	uc_alarm_event_cnt	0,10	0	10	10
	uc_m_alarm_event_fail	0	0	0	0
	uc_m_alarm_event_sec_pnt	0	0	0	0
	uc_m_buffer	0	0	0	0
	ul_m_flash_erase_address	0	0	0	0
Output	uc_alarm_event_cnt	0,10	0	10	10
	uc_m_alarm_event_fail	0,1	0	1	0
	uc_m_alarm_event_sec_pnt	0,1	0	0	1
	ul_m_flash_erase_address	0,360448	0	0	360448
Judgment			OK	OK	OK

m_lg_alarm_event_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_DataLogger_addr->uc_alarm_event_cnt == M_LG_ALARM_EVENT_MAX)		○	-	-	-	-
	else		-	○	○	○	○
	if(uc_m_same_alarm_event_time == 1)		-	○	-	-	-
	else		-	-	○	○	○
	if(luc_write_res != 0x00)		-	-	-	○	-
	else		-	○	○	-	○
	if(uc_m_same_alarm_event_time == 1)		-	○	-	-	-
	if(uc_m_alarm_event_fail == 1)		-	○	○	-	-
	else		-	-	-	-	○
	Symbol	Range					
Input	ppuc_data	0	0	0	0	0	0
	pus_cnt	0	0	0	0	0	0
	DataFlash_flash_write()	0,1	0	0	0	1	0
	uc_alarm_event_cnt	128,129	128	129	129	129	129
	uc_m_alarm_event_fail	1,2	1	1	1	1	2
	uc_m_alarm_event_sec_pnt	0	0	0	0	0	0
	uc_m_last_alarm_event_writetime	0	0	0	0	0	0
	uc_m_max_same_alarm_event	0	0	0	0	0	0
	uc_m_now_time	0	0	0	0	0	0
	uc_m_same_alarm_event_time	1,2	1	1	2	2	2
	ul_m_alarm_event_last_address	0	0	0	0	0	0
	ul_m_flash_erase_address	0	0	0	0	0	0
Output	uc_alarm_event_cnt	128~130	128	130	129	129	129
	uc_m_alarm_event_fail	0~2	1	0	0	2	0
	uc_m_same_alarm_event_time	0~2	1	1	2	0	2
	ul_m_alarm_event_last_address	0,360704	0	360704	0	0	0
	ul_m_flash_erase_address	0,360448	0	360448	360448	0	0
Judgment			OK	OK	OK	OK	OK

m_lg_alarm_head

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(luc_step == 0x00)		○	-	○	○	-
	if(luc_step == 1)		○	○	○	○	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	○	-	○	-
	if(luc_sts == 0x00)		○	○	-	-	-
	else		-	-	○	-	-
	Symbol	Range					
Input	luc_step	0~2	0	1	0	0	2
	ul_m_flash_erase_address	1048576, 1048577	1048577	1048577	1048576	1048577	1048577
	m_flash_status()	0,1	0	0	0	1	0
Output	luc_step	0~2	0	0	0	1	2
Judgment			OK	OK	OK	OK	OK

m_lg_alarm_head_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(st_m_alarm[0].uc_fail == 0)		○	-	○
	if(luc_write_res != 0x00)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	*ppuc_data	0	0	0	0
	pus_cnt	0	0	0	0
	DataFlash_flash_write()	0,1	1	1	0
	uc_head_page_pnt	0	0	0	0
	uc_fail	0,1	0	1	0
	uc_m_buffer	0	0	0	0
Output	st_m_alarm[0].uc_fail	0	0	0	0
Judgment			OK	OK	OK

m_lg_alarm_trend_a_check

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(lui_alm=0; lui_alm<uc_m_alarm_cnt; lui_alm++)		○	○	○	○	○	○
	if((m_lg_time_value(uc_m_now_time) - st_m_alarm[lui_alm].ul_last_datetime) >= M_ALARM_INTERVAL)		○	○	○	○	○	-
	for(lui_i=0; lui_i<M_MAX_PORT; lui_i++)		○	○	○	○	○	-
	else		-	○	○	○	○	○
	for(lui_i=0; lui_i<M_MAX_PORT; lui_i++)		-	○	○	○	○	○
	if(((stp_m_denst+lui_i)->ul_denst != GASTABLEDATA_NA_VAL) && ((stp_m_denst+lui_i)->ul_denst != GASTABLEDATA_OFF_VAL))		-	○	-	-	○	-
	if((st_m_alarm[lui_alm].ul_peak_datas[lui_i] == GASTABLEDATA_NA_VAL) || (st_m_alarm[lui_alm].ul_peak_datas[lui_i] == GASTABLEDATA_OFF_VAL))		-	○	-	-	-	-
	else if(lui_i != M_UNIT_OX)		-	-	○	○	○	-
	if(st_m_alarm[lui_alm].ul_peak_datas[lui_i] < (stp_m_denst+lui_i)->ul_denst)		-	-	○	-	-	-
	else		-	-	○	○	○	-
	else if (lui_i == M_UNIT_OX)		-	-	○	○	○	-
	if(st_m_alarm[lui_alm].ul_peak_datas[lui_i] > (stp_m_denst+lui_i)->ul_denst)		-	-	-	○	○	-
	else		-	-	○	○	-	-
	else		-	-	-	-	-	-
	else		-	-	-	-	○	○
	if(st_m_alarm[lui_alm].fl_temp_data < stp_m_ad->fl_disp_temp)		-	○	○	-	-	-
	else		-	-	-	○	○	○
	if((uc_m_alarm_cnt > 0) && (st_m_alarm[0].us_trend_cnt > M_LG_ALARM_RECORD))		○	○	○	○	○	-
	else		-	-	-	-	-	○
	Symbol	Range						
Input	m_lg_time_value(uc_m_now_time)	0~3126988800	3126988799	0	0	0	0	3126988800
	st_m_alarm[lui_alm].ul_last_datetime	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	(stp_m_denst+lui_i)->ul_denst	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000001	0x00000000	0x00000000	0xFFFFFFFF
	(stp_m_denst+lui_i)->uc_denst_minus_flg	0~2	2	1	1	1	1	0
	st_m_alarm[lui_alm].ul_peak_datas[lui_i]	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000001	0x00000001	0x00000000
	st_m_alarm[lui_alm].fl_temp_data	-55~86	86	-55	-55	85	85	85
	stp_m_ad->fl_disp_temp	-55~86	86	85	85	-55	-55	-55
	uc_m_alarm_cnt	0~3	3	2	2	2	2	0
	st_m_alarm[0].us_trend_cnt	0x0000,0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000
Output	st_m_alarm[lui_alm].ul_peak_datas[lui_i]	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000001	0x00000001	0x00000001	0x00000000
	st_m_alarm[lui_alm].uc_peak_minus[lui_i]	1,2	2	1	1	1	1	1
	st_m_alarm[lui_alm].ul_last_datetime	0x00000000~0xFFFFFFFF	0x00000005	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	st_m_alarm[lui_alm].fl_temp_data	85,86	86	85	85	85	85	85
Judgment			OK	OK	OK	OK	OK	OK

m_lg_alarm_trend_b_check

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

		case1	case2	case3	case4	case5	case6
Root	if((m_lg_time_value(uc_m_now_time) - ul_m_last_alarm_checktime) >= M_ALARM_INTERVAL)	○	-	-	-	-	○
	for(lus_i=0; lus_i<M_MAX_PORT; lus_i++)	○	-	-	-	-	○
	else	-	○	○	○	○	-
	for(lus_i=0; lus_i<M_MAX_PORT; lus_i++)	-	○	○	○	○	-
	if(((stp_m_denst+lus_i)->ul_denst != GASTABLEDATA_NA_VAL) && ((stp_m_denst+lus_i)->ul_denst != GASTABLEDATA_OFF_VAL))	-	○	○	○	○	-
	if((ul_m_alarm_peak_datas[lus_i] == GASTABLEDATA_NA_VAL) || (ul_m_alarm_peak_datas[lus_i] == GASTABLEDATA_OFF_VAL))	-	○	-	-	○	-
	else if((lus_i != M_UNIT_OX) && (ul_m_alarm_peak_datas[lus_i] < (stp_m_denst+lus_i)->ul_denst))	-	-	○	-	-	-
	else if((lus_i == M_UNIT_OX) && (ul_m_alarm_peak_datas[lus_i] > (stp_m_denst+lus_i)->ul_denst))	-	-	-	○	-	-
	else	-	○	-	-	-	-
	else	-	-	-	-	○	-
	if(fl_m_alarm_peak_temp_data < stp_m_ad->fl_disp_temp)	-	-	○	○	○	-
	else	-	○	-	-	-	-
	Symbol	Range					
Input	m_lg_time_value(uc_m_now_time)	0~3126988800	3126988799	0	0	0	0	3126988800
	ul_m_last_alarm_checktime	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	(stp_m_denst+lus_i)->ul_denst	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFE	0x00000001	0x00000000	0xDDDDDDDD	0xFFFFFFFE
	(stp_m_denst+lus_i)->uc_denst_minus_flg	0~2	2	0	0	0	0	1
	stp_m_ad->fl_disp_temp	-55~86	86	-55	85	85	85	-55
	ul_m_alarm_peak_datas[lus_i]	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0x00000000	0x00000001	0xFFFFFFFF	0xFFFFFFFF
	fl_m_alarm_peak_temp_data	-55~86	86	85	-55	-55	-55	85
Output	ul_m_alarm_peak_datas[lus_i]	0x00000001~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFE	0x00000001	0x00000001	0xFFFFFFFF	0xFFFFFFFE
	uc_m_alarm_peak_minus[lus_i]	0~2	2	0	0	0	0	1
	fl_m_alarm_peak_temp_data	-55~86	86	85	85	85	85	-55
	ul_m_last_alarm_checktime	0x00000004~0xFFFFFFFF	0x00000005	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000004
Judgment			OK	OK	OK	OK	OK	OK

m_lg_alarm_trend_c_check

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

		case1	case2	case3
Root	if(uc_m_alarm_cnt > 1)	○	-	○
	for(luc_i=0; luc_i<M_MAX_PORT; luc_i++)	○	-	○
	Symbol	Range			
Input	uc_m_alarm_cnt	0~3	2	0	3
	st_m_alarm[0].uc_trend_page_address	0x00,0xFF	0x00	0x00	0xFF
	st_m_alarm[0].uc_head_page_pnt	0x00,0xFF	0x00	0x00	0xFF
	st_m_alarm[0].us_trend_cnt	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	st_m_alarm[0].us_copy_cnt	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	st_m_alarm[0].uc_copy_page	0x00,0xFF	0x00	0x00	0xFF
	st_m_alarm[0].us_work_pnt	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	st_m_alarm[1].uc_code	0x00,0xFF	0x00	0x00	0xFF
	st_m_alarm[0].uc_fail	0x00,0xFF	0x00	0x00	0xFF
	st_m_alarm[0].ul_last_datetime	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	st_m_alarm[0].uc_datetime	0x00,0xFF	0x00	0x00	0xFF
	st_m_alarm[0].fl_temp_data	-55~86	-55	85	86
Output	st_m_alarm[0].uc_trend_page_address	0	0	0	0
	st_m_alarm[0].uc_head_page_pnt	0	0	0	0
	st_m_alarm[0].us_trend_cnt	0	0	0	0
	st_m_alarm[0].us_copy_cnt	0	0	0	0
	st_m_alarm[0].uc_copy_page	0	0	0	0
	st_m_alarm[0].us_work_pnt	0	0	0	0
	st_m_alarm[0].uc_code	0	0	0	0
	st_m_alarm[0].uc_fail	0	0	0	0
	st_m_alarm[0].ul_last_datetime	0	0	0	0
	st_m_alarm[0].fl_temp_data	0,85	0	85	0
	uc_m_alarm_cnt	1~255	1	255	2
Judgment			OK	OK	OK

m_lg_alarm_trend_copy

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lui_pad >= TREND_RECORD)		○	○	○	○
	else		-	-	-	-
	for(lui_i=lui_pad; lui_i>0; lui_i--)		○	○	○	○
	if((lui_i % M_ALARM_GAS_BYTE) == 0)		○	○	○	○
	if(st_m_alarm[puc_idx].us_work_pnt == 0)		○	○	○	○
	else		-	-	-	-
	if(luc_dat == 0)		○	○	○	○
	if(lui_cnt != 0)		○	○	○	○
	for(lui_i=0; lui_i<(U_SHORT)M_PAGE_SIZE - lui_pad / M_ALARM_GAS_BYTE * RECORD_SIZE; lui_i++)		-	-	-	-
	if(st_m_alarm[puc_idx].uc_fail == 0)		○	○	○	○
	if(luc_write_res != 0x00)		○	○	○	○
	else		-	-	-	-
	Symbol	Range				
Input	puc_idx	0	0	0	0	0
	DataFlash_flash_write()	0,1	1	1	1	0
	uc_head_page_pnt	0	0	0	0	0
	us_copy_cnt	0	0	0	0	0
	uc_copy_page	0	0	0	0	0
	us_work_pnt	0,1	0	1	0	0
	uc_fail	0,1	0	0	1	0
	uc_m_alarm_work	0	0	0	0	0
	uc_m_buffer	0	0	0	0	0
Output	us_copy_cnt	16	16	16	16	16
	uc_copy_page	1	1	1	1	1
	us_work_pnt	4602, 4603	4602	4603	4602	4602
	uc_fail	0,1	1	1	1	0
	uc_m_buffer	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_lg_alarm_trend_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○	○	○
	if(st_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)		○	○	○	○	○	-
	else		-	-	-	-	-	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○	○	○
	if(luc_port != M_NO_ASSIGN)		○	○	○	○	○	-
	if(st_m_gas_datas[luc_port].uc_invalid == M_TRUE)		○	-	-	-	-	-
	else		-	○	○	○	○	-
	if(puc_idx != M_NO_ASSIGN)		-	○	○	-	-	-
	if((st_m_alarm[puc_idx].ul_peak_datas[luc_port] == GASTABLEDATA_NA_VAL) || (st_m_alarm[puc_idx].ul_peak_datas[luc_port] == GASTABLEDATA_OFF_VAL))		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	○	-
	if((ul_m_alarm_peak_datas[luc_port] == GASTABLEDATA_NA_VAL) || (ul_m_alarm_peak_datas[luc_port] == GASTABLEDATA_OFF_VAL))		-	-	-	○	○	-
	else		-	-	-	-	○	-
	else		-	-	-	-	-	○
	if(puc_idx != M_NO_ASSIGN)		○	○	○	-	-	-
	if(st_m_alarm[puc_idx].fl_temp_data < 0)		○	-	-	-	-	-
	else		-	○	○	-	-	-
	else		-	-	-	○	○	○
	if(fl_m_alarm_peak_temp_data < 0)		-	-	-	○	-	-
	else		-	-	-	-	○	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○	○	○
	if(luc_port != M_NO_ASSIGN)		○	○	○	○	○	-
	if(puc_idx != M_NO_ASSIGN)		○	○	○	-	-	-
	if(st_m_alarm[puc_idx].uc_peak_minus[luc_port])		○	-	-	-	-	-
	else		-	○	○	-	-	-
	else		-	-	-	○	○	-
	if(uc_m_alarm_peak_minus[luc_port])		-	-	-	○	-	-
	else		-	-	-	-	○	-
	else		-	-	-	-	-	○
	for(luc_i=luc_cnt; luc_i<M_LG_ALARM_DATA_SZ; luc_i++)		○	○	○	○	○	-
	Symbol	Range						
Input	puc_idx	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF
	*ppuc_data	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
	st_m_gas_datas[luc_port].uc_invalid	0~2	1	0	0	0	0	2
	st_m_alarm[puc_idx].ul_peak_datas[luc_port]	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	ul_m_alarm_peak_datas[luc_port]	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	st_m_alarm[puc_idx].fl_temp_data	-55~86	-55	85	85	0	0	86
	fl_m_alarm_peak_temp_data	-55~86	0	0	0	-55	85	86
	st_m_alarm[puc_idx].uc_peak_minus[luc_port]	0~2	1	0	0	0	0	2
	uc_m_alarm_peak_minus[luc_port]	0~2	0	0	0	1	0	2
Output	luc_cnt	16	16	16	16	16	16	16
Judgment			OK	OK	OK	OK	OK	OK

m_lg_alarm_trend_pad_write

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(lui_i=0; lui_i<M_PAGE_SIZE - st_m_alarm[puc_idx].uc_trend_page_address; lui_i++)		○	○	○
	if(st_m_alarm[puc_idx].uc_fail == 0)		○	-	○
	if(luc_write_res != 0x00)		○	-	-
	else		-	○	-
	Symbol	Range			
Input	puc_idx	0	0	0	0
	DataFlash_flash_write()	0	0	0	0
	m_lg_flash_write()	0,1	1	1	0
	uc_trend_page_address	0	0	0	0
	uc_head_page_pnt	0	0	0	0
	us_trend_cnt	0	0	0	0
	uc_fail	0,1	0	1	0
Output	uc_trend_page_address	0	0	0	0
	uc_fail	0,1	1	1	0
	uc_m_buffer	255	255	255	255
Judgment			OK	OK	OK

m_lg_alarm_trend_write

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(st_m_alarm[puc_idx].uc_fail == 0)		○	-	○
	if(luc_write_res != 0x00)		○	-	-
	else		-	○	-
	Symbol	Range			
Input	*ppuc_data	0	0	0	0
	puc_cnt	0	0	0	0
	puc_idx	0	0	0	0
	DataFlash_flash_write()	0	0	0	0
	m_lg_flash_write()	0,1	1	1	0
	uc_trend_page_address	0	0	0	0
	uc_head_page_pnt	0	0	0	0
	us_trend_cnt	0	0	0	0
	uc_fail	0,1	0	1	0
	uc_m_buffer	0	0	0	0
Output	uc_trend_page_address	0	0	0	0
	us_trend_cnt	1	1	1	1
	uc_fail	0,1	1	1	0
Judgment			OK	OK	OK

m_lg_bump_data

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(uc_m_same_calibration_time == 1)		○	-
	Symbol	Range		
Input	*ppuc_data	0	0	0
	puc_ch	0	0	0
	pul_bump_denst	0	0	0
	pul_test_denst	0	0	0
	puc_pass	0	0	0
	m_lg_concentration()	0	0	0
	m_ui2uc()	0	0	0
	uc_port	0	0	0
	uc_denst_minus_flg	0	1	2
	uc_m_same_calibration_time	1,2	1	2
Output	luc_cnt	7,9	7	9
	*ppuc_data	0	0	0
Judgment			OK	OK

m_lg_calibration_data

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_pass == M_TRUE)		○	-	-
	else		-	○	○
	if(uc_m_same_calibration_time == 1)		○	○	-
	Symbol	Range			
Input	*ppuc_data	0	0	0	0
	puc_ch	0	0	0	0
	pul_before	0	0	0	0
	pul_after	0	0	0	0
	puc_pass	1,2	1	2	1
	m_lg_concentration()	0	0	0	0
	m_ui2uc()	0	0	0	0
	uc_port	0	0	0	0
	uc_denst_minus_flg	0	0	0	0
	uc_m_same_calibration_time	1,2	1	1	2
Output	luc_cnt	7~9	9	7	7
Judgment			OK	OK	OK

m_lg_calibration_fit

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_DataLogger_addr->uc_calibration_cnt >= puc_cnt)		○	○	○	-
	for(lui_i=luc_cnt; lui_i>0; lui_i--)		○	○	○	-
	if(luc_write_res != 0x00)		○	-	-	-
	if(lui_i > 0)		○	-	-	-
	else		-	○	○	-
	Symbol	Range				
Input	puc_cnt	0,1	0	1	1	0
	DataFlash_flash_write()	0,1	1	1	0	0
	flash_read()	0	0	0	0	0
	uc_calibration_cnt	0,5	5	5	0	5
	uc_m_buffer	0	0	0	0	0
	uc_m_calibration_fail	0	0	0	0	0
	uc_m_calibration_sec_pnt	0	0	0	0	0
	ul_m_flash_erase_address	0	0	0	0	0
Output	uc_calibration_cnt	0,5	5	5	0	5
	uc_m_calibration_fail	0,1	1	1	0	0
	uc_m_calibration_sec_pnt	0,1	0	0	0	1
	ul_m_flash_erase_address	0,589824	589824	589824	0	0
Judgment			OK	OK	OK	OK

m_lg_calibration_write

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_DataLogger_addr->uc_calibration_cnt == M_LG_CALIBRATION_MAX)		○	-	-	-	-
	else		-	○	○	○	○
	if(uc_m_same_calibration_time == 1)		-	○	-	-	-
	else		-	-	○	○	○
	if(luc_write_res != 0x00)		-	-	-	○	-
	else		-	○	○	-	○
	if(uc_m_same_calibration_time == 1)		-	○	-	-	-
	if(uc_m_calibration_fail == 1)		-	○	○	-	-
	Symbol	Range					
Input	*ppuc_data	0	0	0	0	0	0
	pus_cnt	0	0	0	0	0	0
	DataFlash_flash_write()	0,1	0	0	0	1	0
	uc_calibration_cnt	254,255	255	254	254	254	254
	uc_m_calibration_fail	1	1	1	1	1	2
	uc_m_calibration_sec_pnt	0	0	0	0	0	0
	uc_m_last_calibration_writetime	0	0	0	0	0	0
	uc_m_max_same_calibration	0	0	0	0	0	0
	uc_m_now_time	0	0	0	0	0	0
	uc_m_same_calibration_time	1,2	1	1	2	2	2
	ul_m_calibration_last_address	0	0	0	0	0	0
	ul_m_flash_erase_address	0	0	0	0	0	0
Output	uc_calibration_cnt	254,255	255	255	254	254	254
	uc_m_calibration_fail	0~2	1	0	0	2	0
	uc_m_same_calibration_time	212	1	1	2	2	2
	ul_m_calibration_last_address	0,523776	0	523776	0	0	0
	ul_m_flash_erase_address	0,589824	0	589824	589824	589824	0
Judgment			OK	OK	OK	OK	OK

m_lg_change_header

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_buf	0x00,0xFF	0x00	0xFF
	puc_offset	0x00,0xFF	0x00	0xFF
	uc_m_buffer[4]	0x00,0xFF	0x00	0xFF
	uc_m_buffer[37]	0x00,0xFF	0x00	0xFF
	uc_m_buffer[60]	0x00,0xFF	0x00	0xFF
	uc_m_buffer[86]	0x00,0xFF	0x00	0xFF
	uc_m_buffer[112]	0x00,0xFF	0x00	0xFF
	uc_m_buffer[138]	0x00,0xFF	0x00	0xFF
	uc_m_buffer[164]	0x00,0xFF	0x00	0xFF
Output	uc_m_buffer[4]	0x00,0xFE	0x00	0xFE
	uc_m_buffer[37]	0x00,0xFE	0x00	0xFE
	uc_m_buffer[60]	0x00,0xFE	0x00	0xFE
	uc_m_buffer[86]	0x00,0xFE	0x00	0xFE
	uc_m_buffer[112]	0x00,0xFE	0x00	0xFE
	uc_m_buffer[138]	0x00,0xFE	0x00	0xFE
	uc_m_buffer[164]	0x00,0xFE	0x00	0xFE
Judgment			OK	OK

m_lg_chmap

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(lus_ch=0; lus_ch<M_MAX_CHANNEL; lus_ch++)		○	○	○
	for(lus_i = 0; lus_i<M_MAX_PORT; lus_i++)		○	-	-
	if((stp_m_sensor+lus_i)->uc_gas_comb != 0)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	(stp_m_sensor+lus_i)->uc_gas_comb	0~30	29	0	30
Output	st_m_ch_datas[lus_ch].uc_port	0x00,0xFF	0x00	0xFF	0x00
Judgment			OK	OK	OK

m_lg_clear_alarm_work

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	for(lui_i=0; lui_i< M_LG_ALARM_WORK_SZ; lui_i++)		○
	Symbol	Range	
Input	None	-	-
Output	uc_m_alarm_work[lui_i]	0xFF	0xFF
Judgment			OK

m_lg_clear_wait

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	st_datalogger_control.uc_state	0x00	0x00
	uc_m_wait_sts	0x00	0x00
Judgment			OK

m_lg_concentration

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((pul_concentration == GASTABLEDATA_OFF_VAL) || (pul_concentration == GASTABLEDATA_NA_VAL))		○	○	-	-	-	-
	if(pul_concentration == GASTABLEDATA_OFF_VAL)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	○	-	-
	if(puc_minus == M_OFF)		-	-	○	○	-	-
	if((stp_m_sensor+puc_port)->ul_full_scale < pul_concentration)		-	-	○	-	-	-
	else		-	-	-	-	○	○
	if(((stp_m_sensor+puc_port)->ul_upper_limit / 10) < pul_concentration)		-	-	-	-	○	-
	Symbol	Range						
Input	pul_concentration	0x00000000~0xFFFFFFFF	0xDDDDDDDD	0xFFFFFFFF	0x000186A0	0x00000000	0x000186A0	0x00000000
	puc_port	0~6	0	1	2	3	5	6
	puc_minus	0~2	2	2	0	0	1	1
	m_lg_ppm2code	0	0	0	0	0	0	0
	ul_full_scale	0~100001	100001	100000	0	100000	0	100000
	ul_digit	0~501	0	0	500	100	0	501
	ul_upper_limit	0~100001	100001	0	0	100000	0	100000
Output	lus_ret	0~65278	56797	65278	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_lg_fatal

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	st_datalogger_control.uc_state	0xBF	0xBF
Judgment			OK

m_lg_flash_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(pul_flash_address >= ul_interval_trend_addr)		○	-
	Symbol	Range		
Input	*sram_address	0x00,0xFF	0x00	0xFF
	pul_flash_address	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000
	write_byte	0x0000,0xFFFF	0x0000	0xFFFF
	puc_idx	0x00,0xFF	0x00	0xFF
	DataFlash_flash_write()	0	0	0
	page_address	0x00,0xFF	0x00	0xFF
Output	luc_ret	0	0	0
Judgment			OK	OK

m_lg_full

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	st_datalogger_control.uc_state	0xBF	0xBF
Judgment			OK

m_lg_full_check

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(lui_i=0; lui_i<3; lui_i++)		○	○	○	○
	if(lus_pnt == (M_INTERVAL_PAGES - 1))		○	-	○	-
	if(stp_m_setting->uc_over_write == M_OFF)		○	-	-	-
	else		-	-	○	-
	else		-	○	○	○
	if((lus_pnt + 992) == lus_pst)		-	-	○	-
	Symbol	Range				
Input	own_sec	0x00~0xFF	0x00	0x3E	0x3E	0xFF
	us_interval_page_pnt	0x0000~0xFFFF	0x011F	0x0000	0x011D	0xFFFF
	uc_over_write	0~2	0	0	1	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_lg_gas

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((stp_m_sensor+puc_port)->ul_1st_alarm != GASTABLEDATA_OFF_VAL)		○	○	-	○
	else		-	-	○	-
	if((stp_m_sensor+puc_port)->ul_2nd_alarm != GASTABLEDATA_OFF_VAL)		○	○	-	○
	else		-	-	○	-
	if((stp_m_sensor+puc_port)->ul_3rd_alarm != GASTABLEDATA_OFF_VAL)		○	○	-	○
	else		-	-	○	-
	if((stp_m_sensor+puc_port)->ul_stel_alarm != GASTABLEDATA_OFF_VAL)		○	○	-	○
	else		-	-	○	-
	if((stp_m_sensor+puc_port)->ul_twa_alarm != GASTABLEDATA_OFF_VAL)		○	○	-	○
	else		-	-	○	-
	if(puc_mode == 0x42)		○	-	-	-
	else		-	○	○	○
	if(stp_m_setting->uc_alarm_on != M_ON)		-	-	○	○
	else		○	○	-	-
	if(((puc_port != 2) && (puc_port != 3) && (puc_port != 4)) || (stp_m_setting->uc_alarm_on != M_ON))		-	-	○	○
	else		○	○	-	-
	if(stp_m_setting->uc_alarm_on != M_ON)		-	-	○	○
	else		○	○	-	-
	Symbol	Range				
Input	puc_mode	0x00~0xFF	0x42	0x00	0x00	0xFF
	puc_port	0~6	0	4	5	6
	ppuc_buf	0x00~0xFF	0x00	0x00	0xFF	0xFF
	(stp_m_sensor+puc_port)->ul_1st_alarm	0~0xDDDDDDDD	0	1000	0xDDDDDDDD	1001
	(stp_m_sensor+puc_port)->ul_2nd_alarm	0~0xDDDDDDDD	0	25000	0xDDDDDDDD	25001
	(stp_m_sensor+puc_port)->ul_3rd_alarm	0~0xDDDDDDDD	0	50000	0xDDDDDDDD	50001
	(stp_m_sensor+puc_port)->ul_stel_alarm	0~0xDDDDDDDD	0	5000	0xDDDDDDDD	5001
	(stp_m_sensor+puc_port)->ul_twa_alarm	0~0xDDDDDDDD	0	2500	0xDDDDDDDD	2501
	stp_m_setting->uc_alarm_on	0~2	1	1	0	2
Output	ppuc_buf	-	Indefinite value	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK	OK

m_lg_gas_invalid

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	for(lui_i=1; lui_i<19 ;lui_i++)		○	○
	Symbol	Range		
Input	puc_mode	0x00,0xFF	0x00	0xFF
	puc_port	0x00,0xFF	0x00	0xFF
	*ppuc_buf	0x00,0xFF	0x00	0xFF
Output	ppuc_buf	-	Indefinite value	Indefinite value
Judgment			OK	OK

m_lg_gas_off

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	for(lui_i=1; lui_i<5 ;lui_i++)		○	○
	for(lui_i=5; lui_i<11 ;lui_i++)		○	○
	for(lui_i=11; lui_i<13 ;lui_i++)		○	○
	for(lui_i=13; lui_i<19 ;lui_i++)		○	○
	Symbol	Range		
Input	puc_mode	0x00,0xFF	0x00	0xFF
	puc_port	0x00,0xFF	0x00	0xFF
	*ppuc_buf	0x00,0xFF	0x00	0xFF
Output	ppuc_buf	-	Indefinite value	Indefinite value
Judgment			OK	OK

m_lg_gas_result

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(ul_m_logger_count != 0)		○	-	-
	else		-	○	○
	if(puc_port == M_UNIT_OX)		-	○	-
	else		-	-	○
	for(lui_i=11; lui_i<19 ;lui_i++)		-	-	○
	Symbol	Range			
Input	puc_mode	0x00,0xFF	0x00	0x00	0xFF
	puc_port	0x00~0xFF	0x00	0x01	0xFF
	*ppuc_buf	0x00,0xFF	0x00	0x00	0xFF
	ul_m_logger_count	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	ppuc_buf	-	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK

m_lg_get_snap_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if ((luc_ov == 0) || ((luc_ov == 1) && (stp_DataLogger_addr->us_snap_write_pnt == 0x100)))		○	○	-	○	-
	else		-	-	○	-	○
	if (puc_pnt >= stp_DataLogger_addr->uc_snap_rec_pnt)		-	-	○	-	-
	else		-	-	-	-	○
	Symbol	Range					
Input	puc_pnt	0	0	0	0	0	0
	uc_snap_rec_pnt	0,1	0	0	0	1	1
	uc_snap_ext_flg	0~2	0	1	2	1	2
	us_snap_write_pnt	256,257	256	256	257	256	257
Output	lus_x	0~256	0	0	256	0	240
Judgment			OK	OK	OK	OK	OK

m_lg_header

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(puc_type == M_LG_ALARM_TREND)		○	-	-	-
	for(lui_i=luc_cnt; lui_i< M_PAGE_SIZE ; lui_i++)		○	-	-	-
	else		-	○	○	○
	do		-	○	○	○
	while((uc_m_interval_fail == 1) || (uc_m_interval_fail == 2))		-	-	-	-
	Symbol	Range				
Input	puc_type	0~7	2	6	0	7
	uc_m_interval_fail	0,3	3	3	3	0
Output	uc_m_buffer[lui_i]	0xFF	0xFF	0xFF	0xFF	0xFF
Judgment			OK	OK	OK	OK

m_lg_header_data

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch (puc_type)		○	○	○	○	○	○	○
	case M_LG_ALARM_TREND		○	-	-	-	-	-	-
	case M_LG_ALARM_EVENT		-	○	-	-	-	-	-
	case M_LG_TROUBLE_EVENT		-	-	○	-	-	-	-
	case M_LG_INTERVAL		-	-	-	○	-	-	-
	case M_LG_CALIBRATION		-	-	-	-	○	-	-
	case M_LG_BUMP		-	-	-	-	-	○	-
	default		-	-	-	-	-	-	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○	○	○	○
	if(st_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)		○	○	○	○	○	○	-
	else		-	-	-	-	-	-	○
	Symbol	Range							
Input	puc_type	0~7	2	3	4	0	5	6	7
	*ppuc_data	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
Output	luc_cnt	153,181	181	181	181	181	181	181	153
Judgment			OK	OK	OK	OK	OK	OK	OK

m_lg_header_plus

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○
	if(st_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)		○	-
	else		-	○
	Symbol	Range		
Input	*data	0x00,0xFF	0x00	0xFF
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0xFF
Output	luc_cnt	6,8	8	6
Judgment			OK	OK

m_lg_interval_check

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(lul_now_sec > ul_m_last_interval_checktime)		○	-	-	-	-
	if(lul_now_sec >= stp_m_setting->us_interval_time)		○	-	-	-	-
	for(lus_i=0; lus_i<M_MAX_PORT; lus_i++)		○	-	○	○	○
	if((((stp_m_denst+lus_i)->uc_state_trb & DataStructConst_S_FAIL_SEN) != DataStructConst_S_FAIL_SEN) && ((stp_m_denst+lus_i)->ul_denst != GASTABLEDATA_OFF_VAL))		○	-	○	○	○
	if((stp_m_denst+lus_i)->uc_denst_minus_flg)		○	-	-	-	-
	else		○	-	○	○	○
	else		-	-	○	○	-
	else		-	-	-	-	-
	for(lus_i=0; lus_i<M_MAX_PORT; lus_i++)		-	-	-	-	-
	if((((stp_m_denst+lus_i)->uc_state_trb & DataStructConst_S_FAIL_SEN) != DataStructConst_S_FAIL_SEN) && ((stp_m_denst+lus_i)->ul_denst != GASTABLEDATA_OFF_VAL))		-	-	-	-	-
	if((stp_m_denst+lus_i)->uc_denst_minus_flg)		-	-	-	-	-
	else		-	-	-	-	-
	Symbol	Range					
Input	m_lg_time_value	0,1	1	0	1	1	1
	fl_m_tick_temp_total	0	0	0	0	0	0
	fl_disp_temp	0	0	0	0	0	0
	ul_denst	3722304899, 3722304990	3722304990	3722304990	3722304990	3722304989	3722304990
	uc_denst_minus_flg	0,1	1	1	1	1	0
	uc_state_trb	4,8	8	8	4	8	8
	us_interval_time	1	1	1	1	1	1
	uc_m_now_time	0	0	0	0	0	0
	ul_m_last_interval_checktime	0	0	0	0	0	0
	ul_m_tick_temp_count	0	0	0	0	0	0
	ul_m_tick_count[0]	0	0	0	0	0	0
	ul_m_tick_total[0]	0	0	0	0	0	0
Output	fl_m_tick_temp_total	0	0	0	0	0	0
	ul_m_last_interval_checktime	0,1	1	0	1	1	1
	ul_m_tick_temp_count	0,1	1	0	1	1	1
	ul_m_tick_count[0]	0,1	1	0	0	0	1
	ul_m_tick_total[0]	0,3722304990	0	0	0	0	3722304990
Judgment			OK	OK	OK	OK	OK

m_lg_interval_event

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL) && ((st_datalogger_control.uc_state & DATALOGGER_WAIT) != DATALOGGER_WAIT))		○	-	○	-	○	○	○	○
	for(lui_i=0; lui_i<M_MAX_CHANNEL; lui_i++)		○	-	○	-	○	○	○	○
	if(st_m_ch_datas[lui_i].uc_count != 0)		○	-	○	-	○	○	○	○
	if(lui_i<M_MAX_CHANNEL)		○	-	○	-	-	-	-	-
	do		○	-	○	-	○	○	○	○
	while((uc_m_interval_fail == 1) || (uc_m_interval_fail == 2))		-	-	-	-	-	-	-	-
	Symbol	Range								
Input	puc_event	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	puc_ch	0	0	0	0	0	0	0	0	0
	m_lg_interval_event_data()	0	0	0	0	0	0	0	0	0
	ul_temp_write_time	0	0	0	0	0	0	0	0	0
	m_lg_time_value()	0	0	0	0	0	0	0	0	0
	uc_count	0,1	1	1	1	1	0	0	0	0
	uc_m_buffer	1	1	1	1	1	1	1	1	1
	uc_m_interval_fail	0~4	3	3	3	3	3	0	3	4
	uc_m_now_time	1	1	1	1	1	1	1	1	1
	ul_m_last_interval_checktime	0	0	0	0	0	0	0	0	0
	uc_state	0~128	4	2	128	64	4	4	4	4
Output	None	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

m_lg_interval_event_check

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(m_comp_pnt(stp_m_setting->uc_st_id, uc_m_last_station, 16) != M_TRUE)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	m_comp_pnt(stp_m_setting->uc_st_id, uc_m_last_station, 16)	0,1	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_lg_interval_event_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_interval_flag == M_FALSE)		○	-	-
	else		-	○	○
	if(puc_event == M_LG_EV_STATION)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	*ppuc_data	0x00,0xFF	0x00	0xFF	0x00
	puc_event	0x00~0xFF	0x3C	0x00	0xFF
	puc_ch	0~6	0	5	6
	pui_sec	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	uc_m_interval_flag	0~2	0	1	2
Output	luc_cnt	4~22	22	4	4
Judgment			OK	OK	OK

m_lg_interval_next

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_DataLogger_addr->us_interval_page_pnt == (M_INTERVAL_PAGES - 1))		○	○	-	-	-
	if(stp_m_setting->uc_over_write == M_OFF)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	if((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL)		○	○	○	○	-
	if(stp_DataLogger_addr->us_interval_page_pnt == (M_INTERVAL_PAGES - 1))		○	-	○	-	-
	if(stp_m_setting->uc_over_write != M_OFF)		-	-	○	-	-
	else		○	-	-	-	-
	else		-	○	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	stp_DataLogger_addr->us_interval_page_pnt	0x0000~0xFFFF	0x011F	0x011F	0x011E	0x0000	0xFFFF
	stp_m_setting->uc_over_write	0~2	0	1	1	0	2
	st_datalogger_control.uc_state	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF
Output	stp_DataLogger_addr->us_interval_page_pnt	0~287	287	0	287	1	0
	stp_DataLogger_addr->uc_over_write	0,1	0	1	0	0	0
Judgment			OK	OK	OK	OK	OK

m_lg_interval_pad_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(pus_cnt > lui_remain)		○	○	○	-
	if((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL)		○	○	-	-
	if(ul_m_flash_pending_erase_address != M_FLASH_SIZE)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	*ppuc_data	0x00,0xFF	0x00	0x00	0x00	0xFF
	pus_cnt	0x0000,0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000
	stp_DataLogger_addr->uc_interval_page_address	0x00,0xFF	0xFF	0xFF	0xFF	0xFF
	st_datalogger_control.uc_state	0x00,0xFF	0x00	0x00	0xFF	0xFF
	ul_m_flash_pending_erase_address	0x00000000~0xFFFFFFFF	0x00000000	0x00100000	0xFFFFFFFF	0xFFFFFFFF
Output	ul_m_flash_erase_address	0	0	0	0	0
	ul_m_flash_pending_erase_address	0x00100000,0xFFFFFFFF	0x00100000	0x00100000	0xFFFFFFFF	0xFFFFFFFF
	stp_DataLogger_addr->uc_interval_page_address	0x00,0xFF	0x00	0x00	0xFF	0xFF
Judgment			OK	OK	OK	OK

m_lg_interval_trend

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_ch == 5)		-	○	-
	else		○	-	○
	do		○	○	○
	while((uc_m_interval_fail == 1) || (uc_m_interval_fail == 2))		-	-	-
	Symbol	Range			
Input	puc_ch	0~6	0	5	6
	uc_m_interval_fail	0~4	0	4	3
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_lg_interval_trend_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_interval_flag == M_FALSE)		○	-	-
	else		-	○	-
	if((st_m_gas_datas[luc_port].uc_invalid == M_TRUE) || (st_m_ch_datas[puc_ch].ul_concentration == GASTABLEDATA_OFF_VAL))		○	-	-
	else if(st_m_gas_datas[luc_port].uc_onoff == M_OFF)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	*ppuc_data	0x00,0xFF	0x00	0x00	0xFF
	puc_ch	0~6	0	5	6
	uc_m_interval_flag	0~2	0	1	2
	st_m_gas_datas[luc_port].uc_invalid	0~3	1	0	3
	st_m_ch_datas[puc_ch].ul_concentration	0x00000000~0xFFFFFFFF	0xDDDDDDDD	0x00000000	0xFFFFFFFF
	st_m_gas_datas[luc_port].uc_onoff	0~2	1	0	2
Output	luc_cnt	4,6	6	4	4
Judgment			OK	OK	OK

m_lg_interval_trend_temp

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(sl_temp_data < 0)		-	○	-
	else		○	-	○
	if(uc_m_interval_flag == M_FALSE)		○	-	-
	else		-	○	○
	if(sl_temp_data == GASTABLEDATA_OFF_VAL)		-	-	○
	else		○	○	-
	Symbol	Range			
Input	*ppuc_data	0x00,0xFF	0x00	0x00	0xFF
	sl_temp_data	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xDDDDDDDD
	uc_m_interval_flag	0~2	0	1	2
Output	luc_cnt	4,6	6	4	4
Judgment			OK	OK	OK

m_lg_interval_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if ((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL)		○	○	○	-	-	-
	if ((U_SHORT)luc_pnt + pus_cnt == (U_SHORT)M_PAGE_SIZE)		○	-	-	-	-	-
	else		-	○	○	-	-	-
	if (luc_write_res != 0x00)		○	○	-	-	-	-
	if (uc_m_interval_fail > 2)		○	-	-	-	-	-
	else		-	-	○	-	-	-
	if (uc_m_interval_fail == 0)		-	-	-	○	○	○
	if ((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)		-	-	-	○	○	○
	if (puc_ch < M_MAX_CHANNEL)		-	-	-	○	-	-
	else if (puc_ch == M_MAX_CHANNEL)		-	-	-	-	○	○
	else if (uc_m_interval_fail > 0)		○	○	-	-	-	-
	if (ul_m_flash_pending_erase_address != M_FLASH_SIZE)		○	-	○	-	-	-
	Symbol	Range						
Input	ppuc_data	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF
	pus_cnt	0x0001~0x0101	0x0001	0x0101	0x0101	0xFF	0xFF	0xFF
	puc_ch	0,4	0	0	0	0	4	4
	st_datalogger_control.uc_state	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF
	stp_DataLogger_addr->uc_interval_page_address	0x00,0xFF	0xFF	0xFF	0x00	0xFF	0xFF	0xFF
	m_lg_flash_write()	0~2	1	1	0	0	0	2
	uc_m_interval_fail	0~2	2	1	0	0	0	0
	us_ineterval_page_pnt	0	0	0	0	0	0	0
	ul_m_flash_erase_address	0	0	0	0	0	0	0
	ul_m_last_interval_checktime	0	0	0	0	0	0	0
	ul_m_flash_pending_erase_address	0x00000000~0xFFFFFFFF	0x00000000	0x00100000	0xFFFFFFFF	0x00100000	0x00100000	0x00100000
Output	stp_DataLogger_addr->uc_interval_page_address	0x00~0xFF	0x00	0x00	0x0100	0xFF	0xFF	0xFF
	uc_m_interval_fail	0~3	3	2	0	0	0	0
	ul_temp_write_time	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	ul_m_flash_erase_address	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	ul_write_time	0	0	0	0	0	0	0
	uc_m_interval_flg	0,1	0	0	0	1	1	1
	ul_m_flash_pending_erase_address	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000
Judgment			OK	OK	OK	OK	OK	OK

m_lg_now_time

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_lg_power_event_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	*ppuc_data	0x00~0xFF	0x00	0xFF
	puc_power	0x00~0xFF	0x00	0xFF
Output	luc_cnt	32	32	32
	uc_m_power_cnt	2	2	2
Judgment			OK	OK

m_lg_power_event_fit

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_DataLogger_addr->uc_power_cnt >= puc_cnt)		○	○	-
	for(lui_i=luc_cnt; lui_i>0; lui_i--)		○	○	-
	if(luc_write_res != 0x00)		○	-	-
	else		-	○	-
	if(lui_i > 0)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_cnt	0x00,0xFF	0x00	0xFF	0xFF
	stp_DataLogger_addr->uc_power_cnt	0x00,0xFF	0xFF	0xFF	0x00
	DataFlash_flash_write(uc_m_buffer,lul_address, M_PAGE_SIZE, FALSE)	0~2	1	0	2
Output	ul_m_flash_erase_address	32768	32768	32768	32768
	stp_DataLogger_addr->uc_power_cnt	0x00~0xFF	0xFF	0x64	0x00
	uc_m_power_sec_pnt	0,1	0	1	1
Judgment			OK	OK	OK

m_lg_power_event_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_DataLogger_addr->uc_power_cnt == M_LG_POWER_MAX)		○	-	-	-
	else		-	○	○	○
	if(luc_write_res != 0x00)		-	○	-	-
	else		-	-	○	○
	if(uc_m_power_fail == 1)		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	*ppuc_data	0x00~0xFF	0x00	0x00	0xFF	0xFF
	pus_cnt	0x0000~0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF
	stp_DataLogger_addr->uc_power_cnt	0x00~0xFF	0x80	0x00	0xFF	0xFF
	DataFlash_flash_write(ppuc_data, lul_address, pui_cnt, FALSE)	0,1	2	1	0	0
	uc_m_power_fail	0~2	3	2	1	0
Output	uc_m_power_fail	0~3	3	3	0	0
Judgment			OK	OK	OK	OK

m_lg_ppm2code

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case2
Root	if(pfl_concentration > 50000)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pfl_concentration	0~50001	50000	0	50001
Output	lui_ret	0~53248	50000	0	53248
Judgment			OK	OK	OK

m_lg_remain

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_DataLogger_addr->us_interval_page_pnt < (M_INTERVAL_PAGES-1))		○	-	-	-	-
	else if(stp_m_setting->uc_over_write == M_OFF)		-	○	-	-	-
	else		-	-	○	○	○
	if((st_datalogger_control.uc_state & DATALOGGER_OVERWRITE) != DATALOGGER_OVERWRITE)		-	-	○	○	-
	if(uc_m_interval_own_sec != (U_CHR)(ul_interval_trend_addr / M_PAGE_SIZE / 16))		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	stp_DataLogger_addr->us_interval_page_pnt	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	stp_m_setting->uc_over_write	0~2	1	0	1	1	2
	st_datalogger_control.uc_state	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF
	uc_m_interval_own_sec	0x00~0xFF	0x00	0x00	0x00	0x3E	0xFF
Output	st_datalogger_control.ul_dlog_time	0,60	60	0	60	0	60
Judgment			OK	OK	OK	OK	OK

m_lg_round

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_m_denst+puc_port)->ul_digit != 0)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	pul_dat	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	puc_port	0~6	0	5	6
	(stp_m_denst+puc_port)->ul_digit	0~501	0	500	501
Output	pul_dat	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
Judgment			OK	OK	OK

m_lg_set_error

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(!(st_datalogger_control.uc_error_record & (DATALOGGER_ER_FLASH_ERASE | DATALOGGER_ER_FLASH_WRITE)))		○	○	○	○	-
	if(((stp_m_status->us_main_state & luc_er) != luc_er) && ((us_m_last_main_state & luc_er) != luc_er) && ((puc_err == DATALOGGER_ER_FLASH_ERASE) || (puc_err == DATALOGGER_ER_FLASH_WRITE)))		○	○	○	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_CLEAR) != DATALOGGER_CLEAR)		○	○	-	-	-
	if((puc_err == DATALOGGER_ER_FLASH_WRITE) || ((stp_DataLogger_addr->uc_over_write == 1) || (puc_kbn != M_KBN_INTERVAL)))		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	puc_err	0x00~0xFF	0x08	0x04	0x04	0x00	0xFF
	puc_kbn	1~9	1	1	1	9	10
	st_datalogger_control.uc_error_record	0x00~0xFF	0x00	0x00	0x00	0x00	0xFF
	stp_m_status->us_main_state	0x0000~0xFFFF	0x0000	0x0000	0x00	0x40	0xFF
	us_m_last_main_state	0x0000~0xFFFF	0x0000	0x0000	0x00	0x40	0xFF
	st_datalogger_control.uc_state	0x00~0xFF	0x00	0x00	0x04	0x00	0xFF
	stp_DataLogger_addr->uc_over_write	0,1	1	0	0	0	2
Output	uc_m_error_kbn	1,9	1	1	1	9	9
	us_m_last_main_state	0x00~0xFF	0x00	0x00	0x40	0x40	0xFF
	stp_m_status->us_main_state	0x00~0xFF	0x00	0x00	0x40	0x40	0xFF
	st_datalogger_control.uc_error_state	0x00~0xFF	0x08	0x0C	0x0C	0x0C	0xFF
	st_datalogger_control.uc_error_record	0x00~0xFF	0x08	0x04	0x04	0x00	0xFF
Judgment			OK	OK	OK	OK	OK

m_lg_setting_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	for(luc_i=0; luc_i<M_MAX_PORT; luc_i++)		○	○
	Symbol	Range		
Input	*ppuc_data	0x00,0xFF	0x00	0xFF
Output	luc_cnt	222	222	222
Judgment			OK	OK

m_lg_setting_fit

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(stp_DataLogger_addr->uc_setting_cnt >= puc_cnt)		○	○	-
	for(lui_i=luc_cnt; lui_i>0; lui_i--)		○	○	-
	if(luc_write_res != 0x00)		○	-	-
	else		-	○	-
	if(lui_i > 0)		-	○	-
	else		○	-	-
	else		-	-	○
	Symbol	Range			
Input	puc_cnt	0x00,0xFF	0x00	0xFF	0xFF
	stp_DataLogger_addr->uc_setting_cnt	0x00,0xFF	0xFF	0xFF	0x00
	DataFlash_flash_write(uc_m_buffer, lul_address, M_PAGE_SIZE, FALSE)	0~2	1	0	2
Output	ul_m_flash_erase_address	0x88000	0x88000	0x88000	0x88000
	stp_DataLogger_addr->uc_setting_cnt	0x00~0xFF	0xFF	0x64	0x00
	uc_m_setting_sec_pnt	0,1	0	1	1
Judgment			OK	OK	OK

m_lg_setting_memory

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	for(luc_i=0; luc_i<M_MAX_PORT; luc_i++)		○
	Symbol	Range	
Input	uc_gas_comb	0	0
	uc_sub_comb	0	0
	uc_uni_dec	0	0
	ul_1st_alarm	0	0
	ul_2nd_alarm	0	0
	ul_3rd_alarm	0	0
	ul_stel_alarm	0	0
	ul_twa_alarm	0	0
	ul_cal_denst	0	0
	uc_cal_group	0	0
	uc_suppress_onoff	0	0
	uc_follower_onoff	0	0
	uc_notice_date	0	0
	uc_chg_sen_date	0	0
	uc_sep_no	0	0
	uc_exp_flg	0	0
	us_back_light_time	0	0
	uc_auto_back_light	0	0
	uc_date_format	0	0
	uc_lcd_invert	0	0
	uc_lcd_back_ground	0	0
	uc_lcd_contrast	0	0
	uc_disp_suppress_val	0	0
	uc_disp_zero_fllwr	0	0
	uc_long_energy_flg	0	0
	uc_margin_disp_onoff	0	0
	uc_ir_unit	0	0
	uc_ir_air_onoff	0	0
	uc_up_cal_flg	0	0
	uc_alarm_pattern	0	0
	uc_1h_alarm_on	0	0
	uc_alarm_on	0	0
	uc_stealth_on	0	0
	uc_stealth_motor_on	0	0
	uc_buzz_volume	0	0
	uc_beep_select	0	0
	uc_beep_time	0	0
	uc_disp_pump_off	0	0
	uc_alarm_silence	0	0
	uc_leak_buzzer_onoff	0	0
	uc_leak_fs_dp	0	0
	uc_leak_mode_flg	0	0
	us_cal_lmt_day	0	0
	uc_cal_lmt_set	0	0
	uc_cal_lmt_chk	0	0
	uc_cal_lmt_gas	0	0
	us_bamp_lmt_day	0	0
	uc_bump_lmt_set	0	0
	uc_bump_lmt_chk	0	0
	uc_bump_lmt_gas	0	0
	uc_chg_batt_date	0	0
	uc_mainte_notice_onoff	0	0
	uc_mainte_lmt_chk	0	0
	uc_mainte_notice_gas	0	0
	us_mainte_notice_day	0	0
	uc_bump_time	0	0
	uc_bump_limit	0	0
	uc_bump_cal_time	0	0
	uc_bump_cal_flg	0	0
	uc_ble_onoff	0	0
	uc_id_disp_onoff	0	0
	uc_lunch_onoff	0	0
	uc_key_tone_onoff	0	0
	uc_disp_setting_item_onoff	0	0
	uc_auto_zero_onoff	0	0
	uc_demand_zero_onoff	0	0
	uc_user_pass_onoff	0	0
	uc_mainte_pass_onoff	0	0
	us_user_pass_num	0	0
	us_mainte_pass_num	0	0
	uc_protect_pword_onoff	0	0
	us_interval_time	0	0
	uc_over_write	0	0

Output	uc_gas_comb	0	0
	uc_sub_comb	0	0
	uc_uni_dec	0	0
	ul_1st_alarm	0	0
	ul_2nd_alarm	0	0
	ul_3rd_alarm	0	0
	ul_stel_alarm	0	0
	ul_twa_alarm	0	0
	ul_cal_denst	0	0
	uc_cal_group	0	0
	uc_suppress_onoff	0	0
	uc_follower_onoff	0	0
	uc_notice_date	0	0
	uc_chg_sen_date	0	0
	uc_sep_no	0	0
	uc_exp_flg	0	0
	us_back_light_time	0	0
	uc_auto_back_light	0	0
	uc_date_format	0	0
	uc_lcd_invert	0	0
	uc_lcd_back_ground	0	0
	uc_lcd_contrast	0	0
	uc_disp_suppress_val	0	0
	uc_disp_zero_fllwr	0	0
	uc_long_energy_flg	0	0
	uc_margin_disp_onoff	0	0
	uc_ir_unit	0	0
	uc_ir_air_onoff	0	0
	uc_up_cal_flg	0	0
	uc_alarm_pattern	0	0
	uc_1h_alarm_on	0	0
	uc_alarm_on	0	0
	uc_stealth_on	0	0
	uc_stealth_motor_on	0	0
	uc_buzz_volume	0	0
	uc_beep_select	0	0
	uc_beep_time	0	0
	uc_disp_pump_off	0	0
	uc_alarm_silence	0	0
	uc_leak_buzzer_onoff	0	0
	uc_leak_fs_dp	0	0
	uc_leak_mode_flg	0	0
	us_cal_lmt_day	0	0
	uc_cal_lmt_set	0	0
	uc_cal_lmt_chk	0	0
	uc_cal_lmt_gas	0	0
	us_bamp_lmt_day	0	0
	uc_bump_lmt_set	0	0
	uc_bump_lmt_chk	0	0
	uc_bump_lmt_gas	0	0
	uc_chg_batt_date	0	0
	uc_mainte_notice_onoff	0	0
	uc_mainte_lmt_chk	0	0
	uc_mainte_notice_gas	0	0
	us_mainte_notice_day	0	0
	uc_bump_time	0	0
	uc_bump_limit	0	0
	uc_bump_cal_time	0	0
	uc_bump_cal_flg	0	0
	uc_ble_onoff	0	0
	uc_id_disp_onoff	0	0
	uc_lunch_onoff	0	0
	uc_key_tone_onoff	0	0
	uc_disp_setting_item_onoff	0	0
	uc_auto_zero_onoff	0	0
	uc_demand_zero_onoff	0	0
	uc_user_pass_onoff	0	0
	uc_mainte_pass_onoff	0	0
	us_user_pass_num	0	0
	us_mainte_pass_num	0	0
	uc_protect_pword_onoff	0	0
	us_interval_time	0	0
	uc_over_write	0	0
Judgment			OK

m_lg_setting_write

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_DataLogger_addr->uc_setting_cnt == M_LG_SETTING_MAX)		○	-	-	-
	else		-	○	○	○
	if(luc_write_res != 0x00)		-	○	-	-
	else		-	-	○	○
	if(uc_m_setting_fail == 1)		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	*ppuc_data	0x00,0xFF	0x00	0x00	0x00	0xFF
	pus_cnt	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF
	stp_DataLogger_addr->uc_setting_cnt	0x00~0xFF	0x80	0x00	0x00	0xFF
	DataFlash_flash_write(ppuc_data, lul_address, M_PAGE_SIZE, FALSE)	0~2	2	1	0	0
	uc_m_setting_fail	0~2	2	0	1	0
Output	ul_m_flash_erase_address	0x000000~0x00088000	0x00000000	0x00000000	0x88000	0x88000
	uc_m_setting_fail	0~2	2	1	0	0
Judgment			OK	OK	OK	OK

m_lg_snap_data

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if (puc_ch == 0)		○	-	○	○	○	○	○	○	○
	if (luc_port != M_NO_ASSIGN)		○	○	-	○	○	○	○	○	○
	if ((st_m_gas_datas[luc_port].uc_invalid == M_TRUE) || ((stp_m_denst+luc_port)->ul_denst == GASTABLEDATA_OFF_VAL))		-	-	-	-	-	-	-	-	-
	else if ((st_m_gas_datas[luc_port].uc_onoff == M_OFF) && ((st_datalogger_control.uc_measure_mode == 0x00) || (st_datalogger_control.uc_measure_mode == 0x03)))		○	○	-	○	○	○	-	○	-
	else		-	-	-	-	-	-	○	-	○
	else		-	-	○	-	-	-	-	-	-
	Symbol	Range									
Input	*ppuc_data	0	0	0	0	0	0	0	0	0	0
	puc_ch	0,1	0	1	0	0	0	0	0	0	0
	m_lg_concentraton()	0	0	0	0	0	0	0	0	0	0
	m_ui2uc()	0	0	0	0	0	0	0	0	0	0
	uc_port	254,255	254	254	255	254	254	254	254	254	254
	uc_onoff	0,1	0	0	0	0	0	1	0	0	0
	uc_invalid	1,2	2	2	2	1	1	2	2	2	2
	ul_denst	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	uc_denst_minus_flg	0	0	0	0	0	0	0	0	0	0
	uc_alarm_lat	0	0	0	0	0	0	0	0	0	0
	st_datalogger_control.uc_measure_mode	0~4	0	0	0	0	0	0	1	3	4
Output	lnc_cnt	3,5	5	3	5	5	5	5	5	5	5
	*ppuc_data	0	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

m_lg_snap_write

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if (lus_pnt >= M_LG_SNAP_REC_MAX)		○	-	○	○
	else		-	○	-	-
	if (luc_write_res != 0x00)		-	-	○	-
	else		○	○	-	○
	if (stp_DataLogger_addr->uc_snap_rec_pnt == 0)		○	○	-	-
	Symbol	Range				
Input	*pucp_data	0	0	0	0	0
	pui_cnt	0	0	0	0	0
	DataFlash_flash_write()	0,1	0	0	1	0
	m_ui2uc()	0	0	0	0	0
	uc_snap_rec_pnt	0,255	255	255	255	0
	uc_snap_exit_flg	0	0	0	0	0
	us_snap_write_pnt	256	256	256	256	256
	uc_m_now_time	0	0	0	0	0
	uc_m_snap_fail	0	0	0	0	0
Output	uc_snap_rec_pnt	0~255	0	0	255	1
	uc_snap_ext_flg	0,1	1	1	0	0
	us_snap_write_pnt	256,257	257	256	256	257
	uc_m_buffer	0,255	0	0	255	255
	uc_m_snap_fail	0,1	0	0	1	0
Judgment			OK	OK	OK	OK

m_lg_stop_interval

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(luc_step == 0)		○	-	-	-	-	-	-
	else		-	○	○	○	○	○	○
	if(luc_step == 1)		○	○	○	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	○		-	-	-	-
	if(luc_sts == 0x00)		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	else		-	-	-	○	○	○	○
	if(luc_step == 2)		○	-	○	-	-	-	-
	if((st_datalogger_control.uc_measure_mode == 0x00) || (st_datalogger_control.uc_measure_mode == 0x03))		○	-	-	-	-	-	-
	else		-	-	○	-	-	-	-
	else		-	○	-	○	○	○	○
	if(luc_step == 3)		○	-	○	○	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	-	○	-	-	-
	if(luc_sts == 0x00)		○	-	-	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	○	-	-	-	-
	else		-	○	-	-	○	○	○
	if(luc_step == 4)		○	-	○	-	-	-	-
	else		-	○	-	○	○	○	○
	if(luc_step == 5)		○	-	○	-	○	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	-	-	○	-	-
	if(luc_sts == 0x00)		○	-	-	-	-	-	-
	else		-	-	-	-	○	-	-
	else		-	-	○	-	-	-	-
	else		-	○	-	○	-	○	○
	Symbol	Range							
Input	luc_step	0~15	0	1	1	3	5	14	15
	ul_m_flash_erase_address	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00100000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	m_flash_status()	0~2	0	1	0	1	1	1	2
	st_datalogger_control.uc_measure_mode	0x00~0xFF	0x00	0x00	0xFF	0xFF	0xFF	0x00	0xFF
Output	luc_step	0~15	0	1	0	3	5	14	15
Judgment			OK	OK	OK	OK	OK	OK	OK

m_lg_trailer

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	do		○	○	○
	while((uc_m_interval_fail == 1) || (uc_m_interval_fail == 2))		-	○	-
	Symbol	Range			
Input	puc_type	0~7	0	6	7
	uc_m_interval_fail	0~4	0	4	3
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_lg_trailer_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(puc_type == M_LG_BARHOLE)		○	-	-	-
	else		-	○	○	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○
	if(st_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)		○	-	-	-
	if(st_m_gas_datas[luc_i].uc_invalid == M_TRUE)		○	-	-	-
	else if(st_m_gas_datas[luc_i].uc_onoff == M_OFF)		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	*ppuc_data	0x00~0xFF	0x00	0x00	0x00	0xFF
	puc_type	0~7	1	0	6	7
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0x00	0x00	0xFF
	st_m_gas_datas[luc_i].uc_invalid	0~2	1	0	0	2
	st_m_gas_datas[luc_i].uc_onoff	0~2	1	0	1	2
Output	luc_cnt	71,91	91	91	91	71
Judgment			OK	OK	OK	OK

m_lg_trend_check

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

		case1	case2	case3	case4	case5	case6
Root	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)	○	○	○	○	○	○
	if(st_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)	○	○	○	○	○	-
	if(ul_m_tick_count[lui_i] > 0)	○	-	-	○	○	-
	else	-	○	○	-	-	-
	if((puc_force == M_FALSE) && ((st_m_gas_datas[lui_i].uc_invalid == M_TRUE) ‖ (st_m_gas_datas[lui_i].uc_onoff == M_OFF) ‖ (st_m_ch_datas[luc_ch].uc_count == 0) ‖ ((lul_dat == st_m_ch_datas[luc_ch].ul_concentration))))	-	○	○	-	-	-
	if((lul_now_sec >= us_m_auto_sec) ‖ (st_m_ch_datas[luc_ch].uc_count == M_AUTO_LIMIT_COUNT))	-	○	-	-	-	-
	else	-	-	○	-	-	-
	else	○	-	-	○	○	-
	if(st_m_ch_datas[luc_ch].uc_count != 0)	○	-	-	○	-	-
	else	-	-	-	-	○	-
	if((puc_force == M_FALSE) && (lul_dat != st_m_ch_datas[luc_ch].ul_concentration))	-	-	-	-	-	-
	else	○	-	-	○	○	-
	else	-	-	-	-	-	○
	if(ul_m_tick_temp_count > 0)	○	-	-	○	○	-
	else	-	○	○	-	-	○
	if((puc_force == M_FALSE) && ((uc_temp_count == 0) ‖ ((lsl_data == sl_temp_data))))	-	○	○	-	-	-
	if((lul_now_sec >= us_m_auto_sec) ‖ (uc_temp_count == M_AUTO_LIMIT_COUNT))	-	○	-	-	-	-
	else	-	-	○	-	-	-
	else	○	-	-	○	○	○
	if(uc_temp_count != 0)	○	-	-	○	-	-
	else	-	-	-	-	○	○
	if((puc_force == M_FALSE) && (lsl_data != sl_temp_data))	-	-	-	-	-	○
	else	○	-	-	○	○	-
	Symbol	Range					
Input	puc_force	0~2	1	0	0	0	1	2
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
	ul_m_tick_count[lui_i]	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	st_m_gas_datas[lui_i].uc_invalid	0~2	1	1	1	0	0	2
	st_m_gas_datas[lui_i].uc_onoff	0~2	1	0	0	1	1	2
	st_m_ch_datas[luc_ch].uc_count	0~180	180	179	0	180	0	181
	m_lg_time_value(uc_m_now_time)	0~3126988800	0	3126988799	0	0	0	3126988800
	st_m_ch_datas[luc_ch].ul_write_time	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	us_m_auto_sec	0x0000,0xFFFF	0xFFFF	0x000	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	st_m_ch_datas[luc_ch].ul_concentration	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0x00000000
	ul_m_tick_temp_count	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	uc_temp_count	0~180	180	179	0	180	0	181
	sl_temp_data	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0xDDDDDDDD	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	ul_temp_write_time	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000000
Output	st_m_ch_datas[luc_ch].ul_concentration	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0xDDDDDDDD	0x00000000	0xDDDDDDDD	0x00000000
	sl_temp_data	-572662307~0	-1	-572662307	-572662307	0	-1	0
	uc_temp_count	0~181	0	0	1	0	0	181
Judgment			OK	OK	OK	OK	OK	OK

m_lg_trend_keep_going

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if(us_m_alarm_work_pnt == M_LG_ALARM_WORK_SZ)	-	○	-
	Symbol	Range			
Input	us_m_alarm_work_pnt	0x0000~0xFFFF	0x0000	0x12BD	0xFFFF
Output	us_m_alarm_work_pnt	0~13	13	0	12
Judgment			OK	OK	OK

m_lg_trouble_event

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

		case1	case2	case3	case4
Root	do	○	○	○	○
	if(luc_cnt>0)	○	○	○	○
	for(lus_i=0; lus_i<M_PAGE_SIZE-luc_cnt; lus_i++)	○	-	-	○
	while(uc_m_trouble_event_fail == 1)	-	-	-	-
	if(uc_m_trouble_event_fail > 1)	-	-	○	○
	Symbol	Range			
Input	m_lg_header_data(uc_m_buffer, M_LG_TROUBLE_EVENT)	0x00~0xFF	0xFF	0x00	0x00	0xFF
	puc_event	0x00~0xFF	0x00	0x00	0x00	0xFF
	puc_ch	0~5	0	5	5	6
	uc_m_trouble_event_fail	0~2	0	0	2	3
Output	uc_m_same_trouble_time	1	1	1	1	1
	uc_m_max_same_trouble	0x7F	0x7F	0x7F	0x7F	0x7F
	uc_m_buffer[luc_cnt+lus_i]	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_lg_trouble_event_data

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(uc_m_same_trouble_time == 1)		○	-	○	○	○	○
	for(luc_i=0; luc_i<3; luc_i++)		○	○	○	○	○	○
	switch (puc_ch)		○	○	○	○	○	○
	case M_UNIT_OX		○	○	-	-	-	-
	case M_UNIT_H2S		-	-	○	-	-	-
	case M_UNIT_CO		-	-	-	○	-	-
	case M_UNIT_EC		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	puc_ch	1~5	1	1	2	3	4	5
	Ad12bit_Get_Ad_Data()	0	0	0	0	0	0	0
	m_ui2uc()	0	0	0	0	0	0	0
	PumpFlow_Get_PumpFailPressValue()	0	0	0	0	0	0	0
	PumpFlow_Get_PumpOffPressValue()	0	0	0	0	0	0	0
	fl_nc_ad_data[0][0][2]	0	0	0	0	0	0	0
	fl_nc_ad_data[0][1][2]	0	0	0	0	0	0	0
	fl_ec_ad_data[0]	0	0	0	0	0	0	0
	fl_ec_ad_data[1]	0	0	0	0	0	0	0
	fl_ec_ad_data[2]	0	0	0	0	0	0	0
	fl_ec_chk_data[0]	0	0	0	0	0	0	0
	fl_ec_chk_data[1]	0	0	0	0	0	0	0
	fl_ec_chk_data[2]	0	0	0	0	0	0	0
	fl_temp_data	0	0	0	0	0	0	0
	uc_state_error_code[0]	1~6	1	2	4	5	5	6
	uc_m_same_trouble_time	1,2	1	2	1	1	1	1
Output	luc_cnt	55~63	59	57	63	63	55	55
	*ppuc_data	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_lg_trouble_event_fit

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(stp_DataLogger_addr->uc_trouble_cnt >= puc_cnt)		○	○	-
	for(lui_i=luc_cnt; lui_i>0; lui_i--)		○	○	-
	if(luc_write_res != 0x00)		○	-	-
	else		-	○	-
	if(lui_i > 0)else		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_cnt	0x00,0xFF	0x00	0xFF	0xFF
	stp_DataLogger_addr->uc_trouble_cnt	0x00,0xFF	0xFF	0xFF	0x00
	uc_m_trouble_sec_pnt	0,2	0	0	2
	DataFlash_flash_write(uc_m_buffer, lul_address, M_PAGE_SIZE, FALSE)	0~2	1	0	2
Output	ul_m_flash_erase_address	0x00068000	0x00068000	0x00068000	0x00068000
	stp_DataLogger_addr->uc_trouble_cnt	0x00~0xFF	0xFF	0x64	0x00
	uc_m_trouble_sec_pnt	0~2	0	1	2
Judgment			OK	OK	OK

m_lg_trouble_event_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(stp_DataLogger_addr->uc_trouble_cnt == M_LG_TROUBLE_EVENT_MAX)		○	-	-	-
	else		-	○	○	-
	if(uc_m_same_trouble_time == 1)		-	○	○	-
	else		-	-	-	-
	if(luc_write_res != 0x00)		-	○	-	-
	else		-	-	○	-
	if(uc_m_same_trouble_time == 1)		-	-	○	-
	else		-	-	-	○
	if(uc_m_trouble_event_fail == 1)		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	*ppuc_data	0x00,0xFF	0x00	0x00	0xFF	0xFF
	pui_cnt	0x00,0xFF	0x00	0x00	0xFF	0xFF
	stp_DataLogger_addr->uc_trouble_cnt	0x00,0xFF	0x80	0x00	0xFF	0xFF
	uc_m_same_trouble_time	0x00,0x01	0x00	0x01	0x01	0x00
	uc_m_trouble_event_fail	0~3	3	0	1	2
	DataFlash_flash_write(ppuc_data, lul_address, M_PAGE_SIZE, FALSE)	0~2	2	1	0	0
	uc_m_max_same_trouble	0x00,0xFF	0x00	0x00	0x0FF	0xFF
Output	uc_m_same_trouble_time	0,1	0	0	1	0
	ul_m_trouble_last_address	0x00000000~0x000700FE	0x00000000	0x00000000	0x0006FFFF	0x000700FE
	ul_m_flash_erase_address	0x00000000,0x00068000	0x00000000	0x00000000	0x00068000	0x00068000
	uc_m_trouble_event_fail	0~3	3	1	0	0
Judgment			OK	OK	OK	OK

m_lg_unset_error

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_err	0x00,0xFF	0x00	0xFF
Output	st_datalogger_control.uc_error_state	0x00,0xFF	0xFF	0x00
Judgment			OK	OK

m_lg_wait

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_sts	0x00,0xFF	0x00	0xFF
Output	uc_m_wait_sts	0x00,0xFF	0x00	0xFF
	st_datalogger_control.uc_state	0xFF	0xFF	0xFF
Judgment			OK	OK

m_ui2uc

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_order != M_BIG)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pus_val	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	*ppuc_val	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	puc_order	0~2	1	0	2
Output	ppuc_val	-	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK

m_lg_day

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((lsc_bc0 < 0) || (lsc_bc0 > 99))		○	-	○
	else		-	○	-
	if((lsc_bc1 <= 0) || (lsc_bc1 > 12))		○	-	○
	else		-	○	-
	if((lsc_bc2 <= 0) || (lsc_bc2 > 31))		○	-	○
	else		-	○	-
	for(i=1; i<= (lsc_bc1 - 1) ;i++)		○	-	-
	switch (i)		○	-	-
	case 1		○	-	-
	case 3		○	-	-
	case 5		○	-	-
	case 7		○	-	-
	case 8		○	-	-
	case 10		○	-	-
	case 12		○	-	-
	case 2		○	-	-
	default		○	-	-
	if(lsc_bc0>0)		○	-	-
	for(i=1; i<= (lsc_bc0 - 1) ;i++)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_bc[0]	0~100	99	100	0
	puc_bc[1]	1~13	12	13	1
	puc_bc[2]	1~32	31	32	1
Output	lul_day	0,36134	36134	0	0
Judgment			OK	OK	OK

m_lg_DiffTime

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(((lsl_day * 86400 + lsl_sec) < lsl_max) && ((lsl_day * 86400 + lsl_sec) > lsl_min))		○	-	-
	else if((lsl_day * 86400 + lsl_sec) > lsl_max)		-	-	○
	else		-	○	-
	Symbol	Range			
Input	*pucp_ac	0x00,0xFF	0x00	0x00	0xFF
	*pucp_bc	0x00,0xFF	0xFF	0xFF	0x00
Output	lsl_ret	0	0	0	0
Judgment			OK	OK	OK

m_lg_info_write

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_DataLogger_addr->uc_power_cnt == M_LG_POWER_MAX)		○	-	-
	else		-	○	○
	if(stp_DataLogger_addr->uc_power_cnt != 0)		-	-	○
	else		-	○	-
	if(uc_m_power_cnt > (luc_in_page_max-1))		-	○	-
	else		-	-	○
	for(lui_i=0; lui_i<M_DEBUG_REC_SZ-puc_cnt; lui_i++)		-	-	○
	Symbol	Range			
Input	*ppuc_data	0x00,0xFF	0x00	0x00	0xFF
	puc_cnt	0x00,0xFF	0xFF	0x00	0x00
	stp_DataLogger_addr->uc_power_cnt	0x00~0xFF	0x80	0x00	0xFF
	uc_m_power_cnt	0x00,0xFF	0xFF	0xFF	0x00
Output	uc_m_power_cnt	0x01,0xFF	0xFF	0xFF	0x01
Judgment			OK	OK	OK

m_lg_time

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_bc[5]	0~60	0	59	60
	puc_bc[4]	0~60	0	59	60
	puc_bc[3]	0~24	0	23	24
Output	lul_sec + lul_mins + lul_hours	0~90060	0	86399	90060
Judgment			OK	OK	OK

m_lg_time_value

	Responsible	Last update
Coding	lkarashi	2021/6/14
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_bc	0~86400	0	86399	86400
Output	m_lg_day(puc_bc) * (U_LNG)86400 + m_lg_time(puc_bc)	0	0	0	0
Judgment			OK	OK	OK

DataLogger_confirm

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Execution time	21
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20	case21
Root	if(stp_m_setting->us_interval_time == 0x00)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	else	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_POWER) != DATALOGGER_ER_NO_POWER)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-
	if(uc_m_main_mode != FlowMain_Get_ModeStatus())	○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-
	if(st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_INIT) != DATALOGGER_ER_NO_INIT)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_WAIT) == DATALOGGER_WAIT)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	○	-	-
	switch (uc_m_wait_sts)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	○	-	-
	case DATALOGGER_READY	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_RUN	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	if((st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_POWER) == DATALOGGER_ER_NO_POWER)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default	-	-	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	if((st_datalogger_control.uc_control == DATALOGGER_CTRL_RERUN)	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_sts == 0)	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if((((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL) && (((st_datalogger_control.uc_state & DATALOGGER_CLEAR) != DATALOGGER_CLEAR) && (((st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_POWER)) != DATALOGGER_ER_NO_POWER)) && (((st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_INIT) != DATALOGGER_ER_NO_INIT))	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-
	if(((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN) || (((st_datalogger_control.uc_state & DATALOGGER_TOSTOP) == DATALOGGER_TOSTOP))	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
	if(st_m_flash_erase_address == M_FLASH_SIZE)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_WAIT) == DATALOGGER_WAIT)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
	switch (uc_m_wait_sts)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
	case DATALOGGER_READY	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_RUN	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_FULL	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((m_tp_stop_interval) == 0)	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_CLEAR	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_OVERWRITE	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_CLOSE	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_FATAL	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(uc_m_error_kbn != M_KBN_TROUBLE_EVENT)	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if((((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN) && (uc_m_error_kbn != M_KBN_INTERVAL))	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_CLOSE) != DATALOGGER_CLOSE)	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(st_datalogger_control.uc_error_record & (DATALOGGER_ER_FLASH_ERASE | DATALOGGER_ER_FLASH_WRITE))	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case DATALOGGER_STOP	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-
	if(uc_sts == 0)	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-
	if(st_datalogger_control.uc_error_record & (DATALOGGER_ER_FLASH_ERASE | DATALOGGER_ER_FLASH_WRITE))	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	default	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-
	if ((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
	if(st_m_flash_erase_address == M_FLASH_SIZE)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL) && (((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN))	○	○	○	○	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
	if(st_m_flash_erase_address == M_FLASH_SIZE)	○	○	○	○	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	if(((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL) && (((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN))	○	○	○	○	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
	if(st_m_flash_erase_address == M_FLASH_SIZE)	○	○	○	○	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	if(((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL) && (((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN))	○	○	○	○	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
	if(st_m_flash_erase_address == M_FLASH_SIZE)	○	○	○	○	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	if (st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL																					
	if((ul_m_flash_erase_address == M_FLASH_SIZE) && (ul_m_flash_pending_erase_address != M_FLASH_SIZE))	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((((st_datalogger_control.uc_state & DATALOGGER_FULL) != DATALOGGER_FULL) && (((st_datalogger_control.uc_state & DATALOGGER_WAIT) != DATALOGGER_WAIT) && (((st_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN))	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
	else	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-
	if(ul_m_flash_erase_address == M_FLASH_SIZE)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-

	Symbol	Range	case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20	case21
Input	stp_m_setting->us_interval_time	0x0000,0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000
	st_datalogger_control.uc_error_state	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x40	0xFF	0xFF
	uc_m_main_mode	0,7	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
	FlowMain_Get_ModeStatus()	0,7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
	st_datalogger_control.uc_state	0x00~0xFF	0x61	0x61	0x61	0x61	0x61	0x61	0x61	0x61	0x70	0x71	0x61	0x61	0x61	0x61	0x21	0x21	0x00	0x00	0x40	0xFF	0xFF
	uc_m_wait_sts	0x00~0xFF	0x00	0x01	0x02	0x02	0x04	0x08	0x10	0x80	0x80	0x80	0xFF	0xFF	0xFF	0x20	0xFF	0xFF	0xFF	0xFF	0x01	0x01	0xFF
	st_datalogger_control.uc_control	0x00~0xFF	0x01	0x01	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	DataLogger_stop()	0~15	0	14	14	14	14	14	14	14	14	14	0	0	14	14	14	14	14	14	14	14	15
	ul_m_flash_erase_address	0x00000000~0xFFFFFFFF	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0xFFFFFFFF	0x00100000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	m_tp_stop_interval()	0~15	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
	uc_m_error_kbn	0~10	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	9	9	10
	st_datalogger_control.uc_error_record	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	ul_m_flash_pending_erase_address	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0xFFFFFFFF
	ul_m_alarm[us_l].ul_cs_copy_cnt	0x0000,0xFFFF	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	uc_m_alarm_cnt	0~3	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	3
Output	uc_m_main_mode	0~8	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
	ul_m_pass_count	0~18	0	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	st_datalogger_control.uc_control	0x00~0xFF	0x00	0x01	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	uc_m_last_main_state	0x00,0x40	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40
	stp_m_status->us_main_state	0x00,0x40	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40	0x40
	ul_m_flash_pending_erase_address	0x00100000,0xFFFFFFFF	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0xFFFFFFFF	0x00100000	0xFFFFFFFF	0xFFFFFFFF	0x00100000	0xFFFFFFFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

DataLogger_power_off

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Execution cycle	17
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17
Root	if((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL) && ((st_datalogger_control.uc_error_state & DATALOGGER_ER_NO_POWER) != DATALOGGER_ER_NO_POWER))		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(uc_step <= 1)		○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((st_datalogger_control.uc_measure_mode == 0x00) || (st_datalogger_control.uc_measure_mode == 0x03))		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_sts == 0)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(uc_step == 2)		○	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_sts == 0x00)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(uc_step == 3)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((uc_pnt != uc_m_alarm_event_sec_pnt) && (ul_m_flash_erase_address == M_FLASH_SIZE))		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(uc_step == 4)		○	-	○	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	○	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(uc_sts == 0x00)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_alarm_event_fail < M_LG_POWER_OFF_FAIL_LIMIT)		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(uc_step == 5)		○	-	○	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	if((uc_pnt != uc_m_trouble_sec_pnt) && (ul_m_flash_erase_address == M_FLASH_SIZE))		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(uc_step == 6)		○	-	○	-	-	-	○	○	○	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	○	-	-	-	○	○	○	-	-	-	-	-	-	-	-
	if(uc_sts == 0x00)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_trouble_event_fail < M_LG_POWER_OFF_FAIL_LIMIT)		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	-	-	-	○	○	○	○	○	○	○	-
	if(uc_step == 7)		○	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	if((uc_pnt != uc_m_calibration_sec_pnt) && (ul_m_flash_erase_address == M_FLASH_SIZE))		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	-	○	○	○	○	○	○	○	-
	if(uc_step == 8)		○	-	○	-	-	-	-	-	○	○	○	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	○	-	-	-	-	-	○	○	○	-	-	-	-	-	-
	if(uc_sts == 0x00)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_calibration_fail < M_LG_POWER_OFF_FAIL_LIMIT)		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	-	-	-	○	○	○	○	○	-
	if(uc_step == 9)		○	-	○	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	if((uc_pnt != uc_m_power_sec_pnt) && (ul_m_flash_erase_address == M_FLASH_SIZE))		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	-	○	○	○	○	○	○	-
	if(uc_step == 10)		○	-	○	-	-	-	-	-	-	○	-	○	○	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	○	-	-	-	-	-	-	○	-	○	-	-	-	-	-
	if(uc_sts == 0x00)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_power_fail < M_LG_POWER_OFF_FAIL_LIMIT)		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	-	○	-	-	○	○	○	-
	if(uc_step == 11)		○	-	○	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	if((uc_pnt != uc_m_setting_sec_pnt) && (ul_m_flash_erase_address == M_FLASH_SIZE))		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	-	○	○	○	-
	if(uc_step == 12)		○	-	○	-	-	-	-	-	-	-	-	-	○	○	○	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	○	-	-	-	-	-	-	-	-	-	○	○	-	-	-
	if(uc_sts == 0x00)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(uc_m_setting_fail < M_LG_POWER_OFF_FAIL_LIMIT)		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	-	○	-
	if(uc_step == 13)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	do		○	-	○	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	while(uc_m_power_fail == 1)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_m_power_fail > 1)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	-	○	-
	if(uc_step == 14)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	if(uc_sts == 0x00)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	if(uc_step == 0)		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

	Symbol	Range																	
Input	st_datalogger_control.uc_state	0x00.0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	st_datalogger_control.uc_error_state	0x00.0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	luc_step	0~15	0	0	0	2	4	4	4	6	6	8	8	10	10	12	12	14	15
	st_datalogger_control.uc_measure_mode	0x00.0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	DataLogger_stop()	0~15	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
	ul_m_flash_erase_address	0x00000000~0xFFFFFFFF	0x00100000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00100000	0x00000000	0x00100000	0x00100000	0x00000000	0x00000000	0x00100000	0x00000000	0x00100000	0x00000000	0xFFFFFFFF
	m_flash_status()	0~2	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	2
	uc_m_alarm_event_sec_pnt	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_alarm_event_fail	0~3	0	0	2	2	1	2	2	1	1	1	2	2	1	2	1	2	3
	uc_m_trouble_sec_pnt	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_trouble_event_fail	0~3	0	0	2	2	1	2	2	1	1	1	2	2	1	2	1	2	3
	uc_m_calibration_sec_pnt	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_calibration_fail	0~3	0	0	2	2	1	2	2	1	1	1	2	2	1	2	1	2	3
	uc_m_power_sec_pnt	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_power_fail	0~3	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	3
	uc_m_setting_sec_pnt	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_setting_fail	0~3	0	0	2	2	1	2	2	1	1	1	2	2	1	2	1	2	3
Output	st_datalogger_control.uc_state	0x00~0xFF	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x10	0x00	0x10	0xFF
	ul_m_flash_erase_address	0x00000000~0xFFFFFFFF	0x00050000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00060000	0x00000000	0x00070000	0x00000000	0x00000000	0x00000000	0x00080000	0x00000000	0x00100000	0x00000000	0xFFFFFFFF
	st_datalogger_control.uc_cmd_sts	0~15																	
	luc_step	0~15	0	1	0	2	4	4	6	6	8	8	10	8	10	12	12	0	14	15
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_initialize

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification policy	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if ((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)		○	○	○	○	○	○	○	○	-
	if (uc_m_lg_int > 50000)		○	○	○	○	○	-	-	-	-
	if (stp_DataLogger_addr->uc_interval_page_address != 0)		○	○	○	-	-	-	-	-	-
	if (uc_m_channels == 0)		-	-	-	-	-	-	-	-	-
	for(lux_i=0; lux_i<M_MAX_PORT; lux_i++)		○	○	○	○	○	○	○	○	-
	if ((uc_m_buffer[0] == M_BLANK) && (uc_m_buffer[1] == M_BLANK)&& (uc_m_buffer[M_LG_ALARM_EVENT_SZ-2] == M_BLANK) && (uc_m_buffer[M_LG_ALARM_EVENT_SZ-1] == M_BLANK))		○	○	○	○	○	-	-	-	-
	else		-	-	-	-	-	○	○	○	-
	if ((uc_m_buffer[0] == M_BLANK) && (uc_m_buffer[1] == M_BLANK) && (uc_m_buffer[M_LG_TROUBLE_EVENT_SZ-2] == M_BLANK) && (uc_m_buffer[M_LG_TROUBLE_EVENT_SZ-1] == M_BLANK))		○	○	○	○	○	-	-	-	-
	else		-	-	-	-	-	○	○	○	-
	if ((uc_m_buffer[0] == M_BLANK) && (uc_m_buffer[1] == M_BLANK) && (uc_m_buffer[M_LG_CALIBRATION_SZ-2] == M_BLANK) && (uc_m_buffer[M_LG_CALIBRATION_SZ-1] == M_BLANK))		○	○	○	○	○	-	-	-	-
	else		-	-	-	-	-	○	○	○	-
	if ((uc_m_buffer[0] == M_BLANK) && (uc_m_buffer[1] == M_BLANK) && (uc_m_buffer[M_LG_POWER_SZ-2] == M_BLANK) && (uc_m_buffer[M_LG_POWER_SZ-1] == M_BLANK))		○	○	○	○	○	-	-	-	-
	else		-	-	-	-	-	○	○	○	-
	if ((uc_m_buffer[0] == M_BLANK) && (uc_m_buffer[1] == M_BLANK) && (uc_m_buffer[M_LG_SETTING_SZ-2] == M_BLANK) && (uc_m_buffer[M_LG_SETTING_SZ-1] == M_BLANK))		○	○	○	○	○	-	-	-	-
	else		-	-	-	-	-	○	○	○	-
	if (stp_m_setting->uc_over_write == M_OFF)		-	-	-	○	○	-	-	○	-
	else if (stp_DataLogger_addr->uc_over_write != 0)		○	○	○	-	-	○	-	-	-
	Symbol	Range									
Input	uc_over_write	0,1	1	1	1	1	1	1	1	0	0
	flash_read()	0	0	0	0	0	0	0	0	0	0
	FlowMain_Get_ModeStatus()	0	0	0	0	0	0	0	0	0	0
	uc_cal_date	0	0	0	0	0	0	0	0	0	0
	uc_cal_data	0	0	0	0	0	0	0	0	0	0
	stp_DataLogger_addr[0].us_power_count	49999, 50000	50000	50000	49999	49999	49999	49999	49999	49999	49999
	stp_DataLogger_addr[0].uc_interval_page_address	0,1	1	1	1	0	0	0	0	0	0
	stp_DataLogger_addr[0].uc_over_write	0,1	1	1	1	1	1	1	0	1	1
	uc_m_buffer[0]	254,255	255	255	255	255	255	254	255	255	255
	uc_m_buffer[1]	254,255	255	255	255	255	255	255	254	255	255
	uc_m_buffer[254]	254,255	255	255	255	255	255	255	254	255	255
	uc_m_buffer[255]	254,255	255	255	255	255	255	255	254	255	255
	uc_m_channels	0,1	0	0	0	0	1	1	1	1	1
	uc_m_last_alarm_event_writetime	0	0	0	0	0	0	0	0	0	0
	uc_m_last_calibration_writetime	0	0	0	0	0	0	0	0	0	0
	uc_m_last_trouble_writetime	0	0	0	0	0	0	0	0	0	0
	uc_m_now_time	0	0	0	0	0	0	0	0	0	0
	us_m_lg_int	0	0	0	0	0	0	0	0	0	0
	st_datalogger_control.uc_state	0,255	0	0	0	0	0	0	0	0	255
Output	us_power_count	0~50000	0	0	50000	50000	50000	50000	50000	50000	49999
	uc_interval_page_address	0	0	0	0	0	0	0	0	0	0
	uc_over_write	0,1	1	1	1	1	1	0	0	1	1
	us_m_lg_int	0,50000	0	0	50000	50000	50000	50000	50000	50000	0
	uc_m_alarm_event_sec_pnt	0,1	0	0	0	0	0	1	1	1	1
	uc_m_calibration_sec_pnt	0,1	0	0	0	0	0	1	1	1	1
	uc_m_erase_cnt	0	0	0	0	0	0	0	0	0	0
	uc_m_main_mode	0	0	0	0	0	0	0	0	0	0
	uc_m_power_sec_pnt	0,1	0	0	0	0	0	1	1	1	1
	uc_m_same_alarm_event_time	0	0	0	0	0	0	0	0	0	0
	uc_m_calibration_event_time	0	0	0	0	0	0	0	0	0	0
	uc_m_same_trouble_time	0	0	0	0	0	0	0	0	0	0
	uc_m_setting_sec_pnt	0,1	0	0	0	0	0	1	1	1	1
	uc_m_trouble_sec_pnt	0,1	0	0	0	0	0	1	1	1	1
	ul_m_flash_erase_address	1048576	1048576	1048576	1048576	1048576	1048576	1048576	1048576	1048576	1048576
	ul_m_flash_pending_erase_address	1048576	1048576	1048576	1048576	1048576	1048576	1048576	1048576	1048576	1048576
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

m_lg_alarm_event_check

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification policy	15
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15
Root	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(st_m_ch_data[luc_ch].uc_port != M_NO_ASSIGN)		○	○	○	○	○	-	-	-	-	-	-	-	-	-	-
	do		○	○	○	○	○	-	-	-	-	-	-	-	-	-	-
	if(((stp_m_dsmstr+luc_i)->uc_alarm_lat & luc_ebit) == luc_ebit) && ((uc_m_last_alm_state[uc_ch] & luc_ebit) != luc_ebit))		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((uc_ebit != ALARMEVENT_ALM_MIN)		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((luc_i != M_UNIT_H2_COH2) || (((stp_m_sensor+luc_i)->uc_sub_comb) != M_GASCOMB_H2_COH2))		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(((stp_m_dsmstr+luc_i)->uc_alarm_lat & luc_ebit) != luc_ebit) && ((uc_m_last_alm_state[uc_ch] & luc_ebit) == luc_ebit))		-	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	while((luc_e < M_ALM_STATES) && ((st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL))		○	○	○	○	○	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		○	○	-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	○	-	-	-	-	-	-	-	-	-	-
	if(((stp_m_dsmstr+luc_i)->uc_alarm_lat == M_OFF) && (uc_m_last_alm_state[luc_ch] != M_OFF))		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	○	○	○	○	○	○	○	○	○
	if(st_datalogger_control.uc_state & DATALOGGER_FATAL) == DATALOGGER_FATAL)		-	-	-	-		○	○	○	○	○	○	○	○	○	○
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address == M_FLASH_SIZE) && (st_datalogger_control.uc_state & DATALOGGER_WAIT) != DATALOGGER_WAIT))		-	-	-	-	○	○	○	○	○	○	○	○	○	○	○
	if(ul_m_flash_erase_address == M_FLASH_SIZE)		-	-	-	-	○	-	○	○	○	○	○	-	-	-	-
	if(luc_e != uc_m_FuncTempWarning_Get_OutOfRangeFlg)		-	-	-	-	○	-	-	○	○	○	○	-	-	-	-
	if(luc_e == 0)		-	-	-	-	○	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	if((luc_e == 1) && (uc_m_FuncTempWarning_Get_OutOfRangeFlg == 0))		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	○	-	-	-	-
	if(luc_e == 2)		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	○	○	○	-	-	-	-	-	-	-	-	-	-	○
	if(uc_hit == 1)		-	○	-	-	-	-	-	○	○	○	○	○	○	○	-
	if(uc_m_alarm_cnt < 2)		-	○	-	-	-	-	-	○	○	○	-	-	-	-	-
	if(uc_m_alarm_cnt == 1)		-	○	-	-	-	-	-	○	○	-	-	-	-	-	-
	if((m_lg_DiffTime(ust_m_alarm[uc_m_alarm_cnt-1].uc_datetime, uc_m_now_time) >= M_ALARM_RAISE_MIN)		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	else		○	-	○	○	○	○	-	-	-	-	-	-	-	-	○
	Symbol	Range															
Input	st_m_ch_data[luc_ch].uc_port	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x00
	(stp_m_dsmstr+luc_i)->uc_alarm_lat	0x00,0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	uc_m_last_alm_state[luc_ch]	0x00~0xFF	0x00	0x00	0xFF	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x00
	(stp_m_sensor+luc_i)->uc_sub_comb	0~30	3	0	0	0	0	0	0	1	0	0	0	0	29	30	0
	ul_m_flash_erase_address	0x00000000~0xFFFFFFFF	0x00100000	0x00000000	0x00100000	0x00000000	0x00100000	0x00000000	0x00000000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0xFFFFFFFF	0x00000000
	st_datalogger_control.uc_state	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x80	0x80	0x80	0x80	0x80	0x80	0x80	0x00
	FuncTempWarning_Get_OutOfRangeFlg()	0~3	0	0	0	0	0	0	1	2	3	2	1	0	3	0	
	uc_m_FuncTempWarning_Get_OutOfRangeFlg	0~3	0	0	1	0	2	0	0	0	0	0	2	0	3	0	
	uc_m_alarm_cnt	0~3	1	1	0	0	0	2	0	1	0	2	2	2	2	3	1
	m_lg_DiffTime()	-2147483648~2147483647	2147483647	-2147483648	0	0	0	-2147483648	-2147483648	0	0	0	0	0	0	0xFFFFFFFE	900
Output	uc_m_last_alm_state[luc_ch]	0x00~0xFF	0xFF	0x00	0x00	0xFF	0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x00
	uc_m_FuncTempWarning_Get_OutOfRangeFlg	0~3	0	0	0	0	0	0	0	1	2	3	2	1	0	3	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_lg_trouble_event_check

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	11
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11
Root	if((ul_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN) || ((ul_datalogger_control.uc_measure_mode != 0x00) && (ul_datalogger_control.uc_measure_mode != 0x03))	○	○	○	○	○	○	○	-	○	-	○
	for(luc_ch=0; luc_ch<M_MAX_CHANNEL; luc_ch++)	○	○	○	○	○	○	○	-	○	-	
	if(ul_m_ch_datas[luc_ch].uc_port != M_NO_ASSIGN)	○	○	○	○	-	-	-	-	-	-	-
	do	○	○	○	○	-	-	-	-	-	-	-
	if(((stp_m_dens!=luc_0->uc_state_trb & luc_ebit) == luc_ebit) && ((uc_m_last_state_trb[luc_ch] & luc_ebit) != luc_ebit))	○	○	○	-	-	-	-	-	-	-	-
	if(luc_ebit == DataStructConst_S_FAIL_SEN)	○	○	-	-	-	-	-	-	-	-	-
	else	○	○	○	-	-	-	-	-	-	-	-
	if(((luc_ebit & DataStructConst_S_FAIL_INS) == 0x00) || (ul_datalogger_control.uc_measure_mode == 255))	○	○	○	-	-	-	-	-	-	-	-
	else	○	○	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	○	-	-	-	-	-	-	-	-
	else	○	○	-	-	-	-	-	-	-	-	-
	if((ul_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN) && ((ul_datalogger_control.uc_measure_mode == 0x00) || (ul_datalogger_control.uc_measure_mode == 0x03))	○	-	-	-	-	-	-	-	-	-	-
	if(((luc_ebit & DataStructConst_S_FAIL_INS) == 0x00)	○	-	-	-	-	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-	-	-	-
	if(((luc_ebit & (DataStructConst_S_FAIL_SEN | DataStructConst_S_FAIL_UPD | DataStructConst_S_FAIL_INS))	○	-	-	-	-	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	○	-	-	-	-	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-	-	-	-
	else	-	-	-	○	-	-	-	-	-	-	-
	while((luc_e < M_TRB_STATES) && ((ul_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL))	○	○	○	○	-	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	-	○	-	○
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	○	○	○	○	-	-	-	-	○	-	○
	else	-	-	-	-	○	○	○	-	-	-	-
	if((ul_m_logger_count > 0) && (luc_hit == 1))	○	○	-	-	-	-	-	-	-	-	-
	else	-	-	○	○	○	○	○	-	○	-	○
	else	-	-	-	-	-	-	-	○	-	○	-
	if((ul_m_flash_erase_address == M_FLASH_SIZE) && ((ul_datalogger_control.uc_state & DATALOGGER_WAIT) != DATALOGGER_WAIT))	-	-	-	-	○	○	○	○	-	○	-
	do	-	-	-	-	○	○	○	○	-	○	-
	if(((stp_m_status->us_main_state & lus_body_ebit) == lus_body_ebit) && ((us_m_last_main_state & lus_body_ebit) != lus_body_ebit))	-	-	-	-	○	○	○	○	-	○	-
	if(luc_ecode != M_LG_EV_FAIL_FLOW)	-	-	-	-	○	○	○	○	-	○	-
	else	-	-	-	-	○	○	○	-	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	○	○	○	-	-	-	-
	else	-	-	-	-	○	○	○	○	-	○	-
	if((ul_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)	-	-	-	-	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	-	○	-	○	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	-	-	-	○	-	○	-
	else	-	-	-	-	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-
	while(luc_ecode < M_LG_TROUBLE_BODY5)	-	-	-	-	○	○	○	○	-	○	-
	do	-	-	-	-	○	○	○	○	-	○	-
	if(((stp_m_status->us_main_state & lus_body_ebit) == lus_body_ebit) && ((us_m_last_main_state & lus_body_ebit) != lus_body_ebit))	-	-	-	-	○	○	○	○	-	○	-
	if (luc_ecode_temp != M_LG_EV_FAIL_BLE)	-	-	-	-	○	○	○	○	-	○	-
	else	-	-	-	-	○	○	○	-	-	-	-
	if(luc_ecode_temp == M_LG_EV_FAIL_ACC)	-	-	-	-	○	○	○	○	-	○	-
	else	-	-	-	-	○	○	○	-	-	-	-
	if (ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	○	○	○	○	-	-	-
	else	-	-	-	-	○	○	○	○	-	○	-
	if ((ul_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)	-	-	-	-	○	○	○	○	-	-	-
	else	-	-	-	-	-	-	-	○	-	○	-
	if (ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	-	-	-	○	-	○	-
	else	-	-	-	-	○	○	○	○	-	-	-
	while (luc_ecode < M_LG_TROUBLE_BODY52)	-	-	-	-	○	○	○	○	-	○	-
	do	-	-	-	-	○	○	○	○	-	○	-
	if (((stp_m_status->us_main_state & lus_body_ebit) == lus_body_ebit) && ((us_m_last_main_state & lus_body_ebit) != lus_body_ebit))	-	-	-	-	○	○	○	○	-	○	-
	if (luc_ecode != M_LG_EV_FAIL_PUMP)	-	-	-	-					-	○	-
	else	-	-	-	-							
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	○	○	○	-	-	○	-
	else	-	-	-	-	○	○	○	○	-	-	-
	if((ul_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)	-	-	-	-	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	-	○	-	-	-
	if(ul_m_flash_erase_address != M_FLASH_SIZE)	-	-	-	-	-	-	-	○	-	-	-
	else	-	-	-	-	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-
	while (luc_ecode < (M_LG_TROUBLE_BODY53 - 1U))	-	-	-	-	-	-	-	-	-	-	○
	if((ul_datalogger_control.uc_state & DATALOGGER_RUN) == DATALOGGER_RUN)	-	-	-	-	○	○	○	-	-	-	-
	if(((stp_m_status->us_main_state & DataStructConst_FAIL_FLW) != DataStructConst_FAIL_FLW) && ((us_m_last_main_state & DataStructConst_FAIL_FLW) == DataStructConst_FAIL_FLW))	-	-	-	-	○	○	-	-	-	-	-
	if(ul_m_flash_erase_address == M_FLASH_SIZE)	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	○	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	-	-	-	-
	if(((stp_m_status->us_main_state & DataStructConst_FAIL_PMP) != DataStructConst_FAIL_PMP) && ((us_m_last_main_state & DataStructConst_FAIL_PMP) == DataStructConst_FAIL_PMP))	-	-	-	-	○	○	-	-	-	-	-
	if(ul_m_flash_erase_address == M_FLASH_SIZE)	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	○	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	-	-	-	-
	if(((stp_m_status->us_main_state & DataStructConst_FAIL_BLE) != DataStructConst_FAIL_BLE) && ((us_m_last_main_state & DataStructConst_FAIL_BLE) == DataStructConst_FAIL_BLE))	-	-	-	-	○	○	-	-	-	-	-
	if(ul_m_flash_erase_address == M_FLASH_SIZE)	-	-	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	○	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	-	-	-	-
	if(luc_hit == 1)	-	-	-	-	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	○	-	-	-	-
	else	-	-	-	-	-	-	-	○	-	○	○
	else	○	○	○	○	-	-	-	-	○	-	-
	if((ul_m_flash_erase_address == M_FLASH_SIZE) && ((ul_datalogger_control.uc_state & DATALOGGER_WAIT) != DATALOGGER_WAIT))	-	-	-	-	-	○	-	-	-	-	-
	else	○	○	○	○	○	-	○	-	○	-	○

	Symbol	Range										
Input	ul_datalogger_control.uc_state	0x00-0xFF	0x01	0x01	0x01	0x00	0x01	0x01	0x01	0x00	0x01	0x00	0xFF
	ul_datalogger_control.uc_measure_mode	0x00-0xFF	0x00	0x01	0x00	0xFF	0xFF	0xFF	0xFF	0x00	0xFF	0x00	0x00
	st_m_ch_datas[luc_ch].uc_port	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	(stp_m_dens!=luc_0->uc_state_trb	0x00,0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	uc_m_last_state_trb[luc_ch]	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	ul_m_flash_erase_address	0x00000000-0xFFFFFFFF	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x00000000	0x00100000	0xFFFFFFFF
	ul_m_logger_count	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	stp_m_status->us_main_state	0x0000,0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000
	us_m_main_state	0x0000,0xFFFF	0x0000	0x0000	0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0x0000	0x0000	0xFFFF
Output	uc_m_last_state_trb[luc_ch]	0x00,0xFF	0xFF	0xFF	0xFF	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	st_m_gas_datas[luc_i].uc_invalid	1	1	1	1	1	1	1	1	1	1	1	1
	us_m_last_main_state	0x0000,0xFFFF	0x0000	0x0000	0x0000	0x0000	0x0003	0xFFFF	0x0000	0xFFFF	0x0000	0xFFFF	0xFFFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_flash_sector_erase

	Responsible	Last update
Coding	Ikarashi	2021/6/14
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	13
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	if ((pul_flash_address >= ul_interval_trend_addr) && (pul_flash_address <= (ul_interval_trend_addr + (U_LNG)(M_INTERVAL_SECTORS-1) * (U_LNG)M_PAGE_SIZE)))	-	-	-	-	-	○	-	-	○	-	-	-	-
	if ((pul_flash_address % 0x010000) == 0x00)	-	-	-	-	-	-	-	-	○	-	-	-	-
	while (m_flash_status() != 0)	-	-	-	-	-	-	-	-	-	-	-	-	-
	erase	-	-	-	-	-	○	-	-	-	-	-	-	-
	else if ((pul_flash_address >= ul_snap_addr) && (pul_flash_address <= (ul_snap_addr + (U_LNG)M_SNAP_REC_SECTORS-1) * (U_LNG)M_PAGE_SIZE)))	-	-	-	-	-	-	○	○	-	-	-	-	-
	if ((pul_flash_address % 0x010000) == 0x00)	-	-	-	-	-	-	-	○	-	-	-	-	-
	while (m_flash_status() != 0)	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	○	-	-	-	-	-	-
	else if ((pul_flash_address >= ul_snap_addr_ext) && (pul_flash_address <= (ul_snap_addr_ext + (U_LNG)M_LG_SNAP_REC_EXT-1) * (U_LNG)M_PAGE_SIZE))	-	-	-	-	○	-	-	-	-	-	-	-	○
	if ((pul_flash_address % 0x010000) == 0x00)	-	-	-	-	-	-	-	-	-	-	-	-	○
	while (m_flash_status() != 0)	-	-	-	-	○	-	-	-	-	-	-	-	-
	else	-	-	-	○	-	-	-	-	-	-	-	-	-
	else if ((pul_flash_address >= ul_alarm_trend_addr[0]) && (pul_flash_address <= ul_alarm_trend_addr[M_ALARM_TREND_SECTORS-1]))	-	-	○	-	-	-	-	-	-	-	-	-	-
	else if ((pul_flash_address == ul_power_addr[0]) || (pul_flash_address == ul_power_addr[1]))	○	-	-	-	-	-	-	-	-	-	-	-	-
	else if ((pul_flash_address == ul_alarm_event_addr[0]) || (pul_flash_address == ul_alarm_event_addr[1]))	-	-	-	-	-	-	-	-	-	-	○	-	-
	else if ((pul_flash_address == ul_trouble_addr[0]) || (pul_flash_address == ul_trouble_addr[1]))	-	-	-	-	-	-	-	-	-	○	-	-	-
	else if ((pul_flash_address == ul_calibration_addr[0]) || (pul_flash_address == ul_calibration_addr[1]))	-	○	-	-	-	-	-	-	-	-	-	-	-
	else if ((pul_flash_address == ul_setting_addr[0]) || (pul_flash_address == ul_setting_addr[1]))	-	-	-	-	-	-	-	-	-	-	-	○	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	○

	Symbol	Range												
Input	pul_flash_address	0-855360	0	1	458752	176128	659200	327424	253696	188416	253952	425984	327680	524288	655360
	flash_Block_Erase()	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	m_flash_status()	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_m_erase_cnt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_m_erase_type	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ul_m_flash_erase_address	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Output	uc_m_erase_cnt	0,2	0	0	2	0	0	0	0	0	0	0	0	0	0
	uc_m_erase_type	0-2	1	0	2	0	0	0	0	0	1	1	1	0
	ul_m_flash_erase_address	0-855360	0	0	458752	176128	1048576	1048576	1048576	188416	253952	425984	327680	524288	655360
	ul_m_erase_address	0,176128	0	0	0	176128	176128	176128	176128	176128	176128	176128	176128	176128	176828
	ul_m_erase_block_size	0,4096	0	0	0	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	data_struct_const.c

Data_Detector_Init_EepAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Data_Detector_Set_DataAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Data_Detector_Set_Default

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(lul_i=0; lul_i<sizeof(Detector_Defaurt); lul_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DataStructConst_def_DenstData

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○
	for(luc_j=0; luc_j<M_DATE_SIZE; luc_j++)		○
	for(luc_j=0; luc_j<15; luc_j++)		○
	Symbol	Range	
Input	None	-	-
Output	DENST_DATA[luc_i].ul_m_denst	0	0
	DENST_DATA[luc_i].ul_denst	0	0
	DENST_DATA[luc_i].uc_m_over_flg	0	0
	DENST_DATA[luc_i].uc_over_flg	0	0
	DENST_DATA[luc_i].uc_m_denst_minus_flg	0	0
	DENST_DATA[luc_i].uc_denst_minus_flg	0	0
	DENST_DATA[luc_i].fl_span_fix	0	0
	DENST_DATA[luc_i].fl_span_max[0]	0	0
	DENST_DATA[luc_i].fl_span_max[1]	0	0
	DENST_DATA[luc_i].fl_span_min[0]	0	0
	DENST_DATA[luc_i].fl_span_min[1]	0	0
	DENST_DATA[luc_i].fl_air_max	0	0
	DENST_DATA[luc_i].fl_air_min	0	0
	DENST_DATA[luc_i].fl_zero[0]	0	0
	DENST_DATA[luc_i].fl_zero[1]	0	0
	DENST_DATA[luc_i].uc_uni_dec	0	0
	DENST_DATA[luc_i].uc_state_alm	0	0
	DENST_DATA[luc_i].uc_state_trb	0	0
	DENST_DATA[luc_i].uc_alarm_lat	0	0
	DENST_DATA[luc_i].ul_pmin	0	0
	DENST_DATA[luc_i].ul_pmax	0	0
	DENST_DATA[luc_i].uc_pmin_tim[luc_j]	0	0
	DENST_DATA[luc_i].uc_pmax_tim[luc_j]	0	0
	DENST_DATA[luc_i].ul_60_ave	0	0
	DENST_DATA[luc_i].ul_60_total	0	0
	DENST_DATA[luc_i].ul_60_cnt	0	0
	DENST_DATA[luc_i].ul_ave	0	0
	DENST_DATA[luc_i].ul_ave_60_cnt	0	0
	DENST_DATA[luc_i].ul_ave_total_cnt	0	0
	DENST_DATA[luc_i].ul_stel_point	0	0
	DENST_DATA[luc_i].ul_stel_buf[luc_j]	0	0
	DENST_DATA[luc_i].ul_stel	0	0
	DENST_DATA[luc_i].ul_twa	0	0
Judgment			OK

DataStructConst_DefaultStId

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i = 0U; luc_i < HDDEF_STATION_ID_NUM; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DataStructConst_DefaultUserId

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i = 0U; luc_i < HDDEF_USER_ID_NUM; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DataStructConst_Set_DenstDataAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

DataStructConst_Set_FramGasSpec

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3
Root	if(puc_ch == 0 || puc_ch == 1)		○	-	-
	else if(puc_ch == 2 || puc_ch == 3)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_ch	0~4	0	2	4
	puc_num	0~29	0	1	29
Output	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_gas_comb	0~29	0	1	29
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_sub_comb	0~29	0	1	29
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_full_scale	0~2000	100	2000	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_digit	0,1	1	1	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_uni_dec	0,33	0	33	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_gas_direction	0,1	1	0	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_1st_alarm	0,10	0	10	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_2nd_alarm	0,100	0	100	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_3rd_alarm	0,100	0	100	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_over_alarm	0,2000	0	2000	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_upper_limit	0,1000	0	1000	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_stel_alarm	0,50	0	50	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_twa_alarm	0,10	0	10	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_gas_alarm_max	0~2000	60	2000	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_gas_alarm_min	0,10	0	10	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_stel_alarm_max	0~3722304989	3722304989	2000	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_stel_alarm_min	0~3722304989	3722304989	10	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_cal_denst	0,250	0	250	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_cal_denst_min	0,10	0	10	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].ul_cal_denst_max	0,2000	0	2000	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_zero_temp_data	0,25	0	25	25
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_zero_buf[0]	0	0	0	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_zero_buf[1]	0	0	0	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_span_temp_data	0,25	0	25	25
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_span_buf[0]	0~167.18	4.2	167.18	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_span_buf[1]	0~167.18	4.2	167.18	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_lin_number[0]	0	0	0	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_lin_number[1]	0,1	0	0	0
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_meas_group[0]	0,1	0	1	1
	stp_GasCalc_addr->GAS_DATA[puc_ch].uc_meas_group[1]	0,1	0	1	1
	stp_GasCalc_addr->GAS_DATA[puc_ch].us_hc_rate[0]	0,100	0	100	100
	stp_GasCalc_addr->GAS_DATA[puc_ch].us_hc_rate[1]	0,100	0	100	100
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_lel_ratio[0]	0~5000	0	50000	10000
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_lel_ratio[1]	0~5000	0	50000	10000
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_lel_max_ratio[0]	0~5000	0	50000	10000
	stp_GasCalc_addr->GAS_DATA[puc_ch].fl_lel_max_ratio[1]	0~5000	0	50000	10000
Judgment			OK	OK	OK

DataStructConst_Set_GasSpec

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if(luc_i == 1)		○	○	○
	else		○	○	○
	for(luc_j=0; luc_j<15; luc_j++)		○	○	○
	Symbol	Range			
Input	stp_GasCalc_addr->GAS_DATA[luc_i].uc_gas_comb	0~30	0	29	30
	stp_GasCalc_addr->GAS_DATA[luc_i].ul_digit	0~501	0	500	501
	stp_GasCalc_addr->GAS_DATA[luc_i].fl_zero_buf[0]	-68.9~76	-68.9	75	76
	stp_GasCalc_addr->GAS_DATA[luc_i].fl_zero_buf[1]	-68.9~76	-68.9	75	76
	stp_GasCalc_addr->GAS_DATA[luc_i].uc_uni_dec	0x00~0x23	0x00	0x22	0x23
Output	ENST_DATA[luc_i].ul_digit	0~501	0	500	501
	DENST_DATA[luc_i].fl_span_fix	0,1	0	1	0
	DENST_DATA[luc_i].fl_span_max[0]	0~8	0	8	0
	DENST_DATA[luc_i].fl_span_max[1]	0~8	0	8	0
	DENST_DATA[luc_i].fl_span_min[0]	0~1.6	0	1.6	0
	DENST_DATA[luc_i].fl_span_min[1]	0~1.6	0	1.6	0
	DENST_DATA[luc_i].fl_air_max	0,85	0	85	0
	DENST_DATA[luc_i].fl_air_min	0,65	0	65	0
	DENST_DATA[luc_i].fl_zero[0]	-68.9~76	-68.9	75	76
	DENST_DATA[luc_i].fl_zero[1]	-68.9~76	-68.9	75	76
	DENST_DATA[luc_i].uc_uni_dec	0~35	0	34	35
	DENST_DATA[luc_i].uc_state_alm	0	0	0	0
	DENST_DATA[luc_i].uc_state_trb	0	0	0	0
	DENST_DATA[luc_i].uc_alarm_lat	0	0	0	0
	DENST_DATA[luc_i].ul_pmin	0	0	0	0
	DENST_DATA[luc_i].ul_pmax	0	0	0	0
	DENST_DATA[luc_i].ul_60_ave	0	0	0	0
	DENST_DATA[luc_i].ul_60_total	0	0	0	0
	DENST_DATA[luc_i].ul_60_cnt	0	0	0	0
	DENST_DATA[luc_i].ul_ave	0	0	0	0
	DENST_DATA[luc_i].ul_ave_60_cnt	0	0	0	0
	DENST_DATA[luc_i].ul_ave_total_cnt	0	0	0	0
	DENST_DATA[luc_i].ul_stel_point	0	0	0	0
	DENST_DATA[luc_i].ul_stel_buf[luc_j]	0	0	0	0
	DENST_DATA[luc_i].ul_stel	0	0	0	0
	DENST_DATA[luc_i].ul_twa	0	0	0	0
Judgment			OK	OK	OK

DataStructConst_Set_NCGasSpec

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~30	0	29	30
Output	stp_GasCalc_addr->GAS_DATA[0].uc_lin_number[1]	0	0	0	0
	stp_GasCalc_addr->GAS_DATA[0].uc_meas_group[1]	0,1	0	1	0
	stp_GasCalc_addr->GAS_DATA[0].us_hc_rate[1]	0~7680	0	7424	7680
	stp_GasCalc_addr->GAS_DATA[0].fl_lel_ratio[1]	0~1.076197E-41	0	1.04E-41	1.08E-41
	stp_GasCalc_addr->GAS_DATA[0].fl_lel_max_ratio[1]	0~1.076197E-41	0	1.04E-41	1.08E-41
Judgment			OK	OK	OK

DataStructConst_Set_StatusDataAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasCalc_Get_Alarm1hON

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_Detector_addr->uc_1h_alarm_on == M_ON)		○	○	-	-
	if(stp_GasCalc_addr->GAS_DATA[3].uc_gas_comb != 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_Detector_addr->uc_1h_alarm_on	0~2	1	1	0	2
	stp_GasCalc_addr->GAS_DATA[3].uc_gas_comb	0~30	29	0	29	30
Output	luc_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

GasCalc_Get_AlarmTwaON

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=2; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if(stp_GasCalc_addr->GAS_DATA[luc_i].uc_gas_comb != 0x00)		○	○	-	○
	if(luc_i != DataStructConst_CO_CH || stp_Detector_addr->uc_1h_alarm_on != M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	Symbol	Range				
Input	stp_GasCalc_addr->GAS_DATA[luc_i].uc_gas_comb	0~30	29	29	0	30
	stp_Detector_addr->uc_1h_alarm_on	0~2	0	1	0	2
Output	luc_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

GasCalc_Get_CombGasON

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_GasCalc_addr->GAS_DATA[0].uc_gas_comb != 0x00)		○	○	-	○
	if(puc_flg == M_ON && stp_GasCalc_addr->GAS_DATA[0].uc_gas_comb <= 2)		○	-	-	-
	else		-	○	-	○
	if(puc_flg == M_OFF)		-	○	-	-
	else		○	-	-	-
	else		-	-	○	-
	Symbol	Range				
Input	puc_flg	0~2	1	0	0	2
	stp_GasCalc_addr->GAS_DATA[0].uc_gas_comb	0~30	1	29	0	30
Output	luc_ans	0,1	1	1	0	0
Judgment			OK	OK	OK	OK

GasCalc_Get_Gas_Comb

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_num > 4)		-	-	○
	else		○	○	-
	Symbol	Range			
Input	puc_num	0~5	0	4	5
Output	stp_GasCalc_addr->GAS_DATA[puc_num].uc_gas_comb	0,30	30	0	0
Judgment			OK	OK	OK

GasCalc_Get_GasData_Add

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasCalc_Get_H2MeasurementFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_ch=0U; luc_ch<HDDEF_GAS_MAX; luc_ch++)		○	○	○	○	○
	if(stp_GasCalc_addr->GAS_DATA[luc_ch].uc_gas_comb != 0x00U)		○	○	○	○	○
	if((luc_ch != DataStructConst_CO_CAN_H2_H2_CH) || (stp_GasCalc_addr->GAS_DATA[luc_ch].uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		○	-	○	○	○
	if(Tool_Comp_Pnt((const U8_CHR *)&(stp_GasCalc_addr->GAS_DATA[luc_ch].uc_disp_name[1][0]), (const U8_CHR *)lncc_h2_name, 3U) == 0U)		○	-	○	○	-
	else		-	-	-	-	○
	else		-	-	-	-	-
	else		-	○	-	-	-
	Symbol	Range					
Input	stp_GasCalc_addr[0].GAS_DATA[0].uc_gas_comb	0~4	1	0	3	4	1
	Tool_Comp_Pnt()	0,1	0	0	0	0	1
Output	lbl_ans	0,1	1	0	1	1	0
Judgment			OK	OK	OK	OK	OK

GasCalc_Get_IDDispON

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_Detector_addr->uc_id_disp_onoff	0	0
Judgment			OK

GasCalc_Get_ToxicGasON

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=2; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if(stp_GasCalc_addr->GAS_DATA[luc_i].uc_gas_comb != 0x00)		○	○	-	-
	if(stp_GasCalc_addr->GAS_DATA[luc_i].ul_stel_alarm != M_OFF_VAL || stp_GasCalc_addr->GAS_DATA[luc_i].ul_twa_alarm != M_OFF_VAL)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_GasCalc_addr->GAS_DATA[luc_i].uc_gas_comb	0~30	29	29	0	30
	stp_GasCalc_addr->GAS_DATA[luc_i].ul_stel_alarm	0~0xDDDDDDDD	0	0xDDDDDDDD	5000	5001
	stp_GasCalc_addr->GAS_DATA[luc_i].ul_twa_alarm	0~0xDDDDDDDD	0	0xDDDDDDDD	2500	2501
Output	luc_ans	0,1	0	0	0	0
Judgment			OK	OK	OK	OK

GasTableData_Init_EepAddress

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasTableData_Set_Default

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(lul_i=0; lul_i<sizeof(Gas_Calc_Defaurt); lul_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_alarm_setting_display.c

FlowAlarmSettingDisplay_Chk_AlarmSettingEndNum

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX+1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000-0xFFFFFFFF	0x00000000	0x00000005	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowAlarmSettingDisplay_Chk_AlarmSettingResetNum

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000-0xFFFFFFFF	0x00000000	0x00000004	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowAlarmSettingDisplay_Disp_AlarmPointChange

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch(stp_zipu_num->ul_sub_item_select_num)		○	○	○	○	○	○	-
	case 0		○	-	-	-	-	-	-
	case 1		-	○	-	-	-	-	-
	case 2		-	-	○	-	-	-	-
	case 3		-	-	-	○	-	-	-
	case 4		-	-	-	-	○	○	-
	if(stp_zipu_num->ul_item_select_num == DataStructConst_CO_CH && stp_data->uc_1h_alarm_on == M_ON)		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	-
	default		-	-	-	-	-	-	○
	Symbol	Range							
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000-0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000004	0xFFFFFFFF
	stp_zipu_num->ul_item_select_num	0x00000000-0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000003	0x00000000	0xFFFFFFFF
	stp_data->uc_1h_alarm_on	0-2	0	0	0	0	1	0	2
Output	None	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK

FlowAlarmSettingDisplay_Disp_AlarmSettingMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num < HDDEF_GAS_MAX)		○	-	-
	else if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	3-5	3	4	5
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowAlarmSettingDisplay_Disp1_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0-3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowAlarmSettingDisplay_Disp2_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0-3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowAlarmSettingDisplay_End_AlarmPointChange

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	uls_m_getting_data[0]	0-1001	0	1000	1001
	uls_m_getting_data[1]	0-25001	0	25000	25001
	uls_m_getting_data[2]	0-50001	0	50000	50001
	uls_m_getting_data[3]	0-5001	0	5000	5001
	uls_m_getting_data[4]	0-2501	0	2500	2501
Output	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_1st_alarm	0	0	0	0
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_2nd_alarm	0	0	0	0
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_3rd_alarm	0	0	0	0
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_stel_alarm	0	0	0	0
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_twa_alarm	0	0	0	0
Judgment			OK	OK	OK

FlowAlarmSettingDisplay_Init_AlarmPointChange

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_1st_alarm	0~1001	0	1000	1001
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_2nd_alarm	0~25001	0	25000	25001
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_3rd_alarm	0~50001	0	50000	50001
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_stel_alarm	0~5001	0	5000	5001
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_twa_alarm	0~2501	0	2500	2501
Output	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
	uls_m_getting_data[0]	0~1001	0	1000	1001
	uls_m_getting_data[1]	0~25001	0	25000	25001
	uls_m_getting_data[2]	0~50001	0	50000	50001
	uls_m_getting_data[3]	0~5001	0	5000	5001
	uls_m_getting_data[4]	0~2501	0	2500	2501
Judgment			OK	OK	OK

FlowAlarmSettingDisplay_Init_AlarmSettingMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((luc_i != DataStructConst_CO_CAN_H2_H2_CH) ‖ ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	○	○
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	30
Output	stp_zipu_num->ul_item_select_num	0	0	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowAlarmSettingDisplay_Select_AlarmPointChange

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	15
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15
Root	if(stp_zipu_num->ul_item_select_num == 1)		○	○	○	○	○	○	○	○	○	-	-	-	-	-	-
	switch(stp_zipu_num->ul_sub_item_select_num)		○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
	case 0:		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(stp_GasInfo_addr->ul_gas_alarm_min != M_OFF_VAL && stp_GasInfo_addr->ul_gas_alarm_max != M_OFF_VAL)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(stp_GasInfo_addr->ul_gas_alarm_min != M_OFF_VAL && stp_GasInfo_addr->ul_gas_alarm_max != M_OFF_VAL)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	○	○	-	-	-	-	-	-	-	-	-
	if(stp_GasInfo_addr->ul_gas_alarm_min != M_OFF_VAL && stp_GasInfo_addr->ul_gas_alarm_max != M_OFF_VAL)		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case 3:		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case 4:		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	○	○	○	○	○
	switch(stp_zipu_num->ul_sub_item_select_num)		-	-	-	-	-	-	-	-	-	○	○	○	○	○	○
	case 0:		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	case 3:		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	case 4:		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	if(pul_num == 100)		-	-	-	-	-	-	-	○	○	-	-	-	-	-	-
	if((lul_max - lul_min) / M_MAX_CHG_DIGIT > (stp_GasInfo_addr->ul_digit * 10))		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	Symbol	Range															
Input	pul_num	0x00000000-0xFFFFFFFF	0	0	0	0	0	0	0	100	100	0	0	0	0	0	0
	ul_gas_alarm_max	0~100001	100000	0xDDDDDDDD	100000	0xDDDDDDDD	100000	0xDDDDDDDD	0	100000	100000	100000	100000	100000	100000	100000	100001
	ul_gas_alarm_min	0~501	500	0xDDDDDDDD	500	0xDDDDDDDD	500	0xDDDDDDDD	0	500	500	500	500	500	500	500	501
	ul_stel_alarm_max	0~100001	100000	100000	100000	100000	100000	100000	0	100000	100000	100000	100000	100000	100000	100000	100001
	ul_stel_alarm_min	0~501	500	500	500	500	500	500	0	500	500	500	500	500	500	500	501
	ul_twa_alarm_max	0~100001	100000	100000	100000	100000	100000	100000	0	100000	100000	100000	100000	100000	100000	100000	100001
	ul_twa_alarm_min	0~501	500	500	500	500	500	500	0	500	500	500	500	500	500	500	501
	ul_item_select_num	0x00000000-0xFFFFFFFF	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	ul_sub_item_select_num	0x00000000-0xFFFFFFFF	0x00000000	0x00000000	0x00000001	0x00000001	0x00000002	0x00000002	0x00000003	0x00000004	0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF
	ul_digit	1~501	1	1	1	1	1	1	1	1	500	1	1	1	1	1	501
Output	None	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowAlarmSettingDisplay_Select_AlarmSettingMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	○	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		○	-	-	-	-	-	-
	else		-	○	○	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == (HDDEF_GAS_MAX+1))		-	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	for(luc_i; luc_i < HDDEF_GAS_MAX; luc_i++)		-	○	○	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00 && (luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		-	○	○	-	-	-	-
	else		-	○	○	-	-	-	-
	if(luc_i == HDDEF_GAS_MAX && DataFramFactry_Chk_AlarmResetOnOffFlg() == M_OFF)		-	-	○	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	-	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == 0)		-	-	-	○	-	-	-
	else		-	-	-	-	○	○	○
	if(stp_zipu_num->ul_item_select_num == (HDDEF_GAS_MAX+1))		-	-	-	-	○	○	-
	if(luc_i == HDDEF_GAS_MAX && DataFramFactry_Chk_AlarmResetOnOffFlg() == M_OFF)		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	-
	else		-	-	-	-	-	-	○
	if(luc_i != HDDEF_GAS_MAX)		-	-	-	-	○	-	○
	for(luc_i; luc_i > 0; luc_i--)		-	-	-	-	○	-	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00 && (luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	-	-	-	-	○
	else		-	-	-	-	○	-	○
	if(luc_i == 0 && (stp_GasInfo_addr+luc_i)->uc_gas_comb == 0x00)		-	-	-	-	○	-	-
	else		-	-	-	-	-	-	○
	else		-	-	-	-	○	○	-
	Symbol	Range							
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	0	1	1	1	2
	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000004	0x00000005	0xFFFFFFFF	0x00000000	0x00000005	0x00000005	0x00000002
	stp_GasInfo_addr+luc_i)->uc_gas_comb	0,29	0	29	0	0	0	0	29
	DataFramFactry_Chk_AlarmResetOnOffFlg()	0~2	0	1	0	0	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~5	5	0	1	5	1	4	1
Judgment			OK	OK	OK	OK	OK	OK	OK

FlowAlarmSettingDisplay_Select_PointTypeChange

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(puc_type == M_OFF)		○	○	○	○	-	-	-
	if(stp_zipu_num->ul_item_select_num < 2)		○	○	-	-	-	-	-
	if(stp_zipu_num->ul_sub_item_select_num < 2)		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	○	○	-	-	-
	if(stp_zipu_num->ul_sub_item_select_num < 4)		-	-	○	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	-	○	○	○
	if(stp_zipu_num->ul_sub_item_select_num == 0)		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	○
	Symbol	Range							
Input	puc_type	0~2	0	0	0	0	1	1	2
	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
Output	luc_ans	0~2	0	1	0	1	2	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

FlowAlarmSettingDisplay_Set_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_beep_display.c

FlowBeepDisplay_Disp_BeepSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○	○	○	○	○
	case 0		○	-	-	-	-	-	-	-
	case 1		-	○	-	-	-	-	-	-
	case 2		-	-	○	-	-	-	-	-
	case 3		-	-	-	○	-	-	-	-
	case 4		-	-	-	-	○	-	-	-
	case 5		-	-	-	-	-	○	-	-
	case 6		-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005	0x00000006	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_blink	0xFF	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowBeepDisplay_Disp_BeepTime

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_item_select_num == 0)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_mes_dot[0]	0x20	0	0
	stp_nl_disp->uc_lcd_mes_dot_blink[0]	0x20	0	0
	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0	0
Judgment			OK	OK

FlowBeepDisplay_End_BeepSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_data->uc_beep_select != stp_zipu_num->ul_item_select_num)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_data->uc_beep_select	0~6	0	6	7
Output	stp_data->uc_beep_select	0x00~0xFF	0x00	0xFF	0xFF
Judgment			OK	OK	OK

FlowBeepDisplay_End_BeepTime

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_beep_time	0x00~0xFF	0x00	0x00
Judgment			OK	OK

FlowBeepDisplay_Init_BeepMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
	stp_zipu_num->ul_item_select_num	0	0
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowBeepDisplay_Init_BeepSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_beep_select	0~4	0	4	5
Output	stp_zipu_num->ul_item_select_num	0~5	0	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBeepDisplay_Init_BeepTime

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_beep_time	0~99	0	99	100
Output	stp_zipu_num->ul_item_select_num	0~99	0	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBeepDisplay_Select_BeepMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_sub_mode_select_num, 1, (U_LNG)U_P_END_NUM, (U_LNG)U_P_BEEP_ONOFF_NUM, stp_zipu_num->uc_sub_mode_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0
Judgment			OK	OK

FlowBeepDisplay_Select_BeepSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)4, (U_LNG)0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowBeepDisplay_Select_BeepTime

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_num	0x00~0xFF	0x00	0x01	0xFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)M_BEEP_TIME_MAX, (U_LNG)M_BEEP_TIME_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)M_BEEP_TIME_MAX, (U_LNG)M_BEEP_TIME_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000	0x00000000	0x00000000	0x00000000
Judgment			OK	OK	OK

FlowBeepDisplay_Set_BeepMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(stp_zipu_num->ul_sub_mode_select_num)		○	○	○	○
	case U_P_BEEP_ONOFF_NUM		○	-	-	-
	case U_P_BEEP_TIME_NUM		-	○	-	-
	case U_P_END_NUM		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0xFFFFFFFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_bump_limit_display.c

FlowBumpLimitDisplay_Disp_BumpExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot_blink	0	0	0	0	0
	stp_nl_disp->uc_lcd_com_blink	0	0	0	0	0
Judgment			OK	OK	OK	OK

FlowBumpLimitDisplay_Disp_BumpInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF
Judgment			OK

FlowBumpLimitDisplay_Disp_BumpReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x10	0x10
Judgment			OK

FlowBumpLimitDisplay_End_BumpExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_bump_lmt_chk	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowBumpLimitDisplay_End_BumpInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->us_bump_lmt_day	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

FlowBumpLimitDisplay_End_BumpReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_bump_lmt_set	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowBumpLimitDisplay_Init_BumpExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_lmt_chk	0~3	0	2	3
Output	stp_zipu_num->ul_item_select_num	0~3	0	2	3
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBumpLimitDisplay_Init_BumpInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_bump_lmt_day	1~1001	1	1000	1001
Output	stp_zipu_num->ul_item_select_num	1~1001	1	1000	1001
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBumpLimitDisplay_Init_BumpLimitMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
	stp_zipu_num->ul_item_select_num	0	0
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowBumpLimitDisplay_Init_BumpReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_lmt_set	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowBumpLimitDisplay_Select_BumpExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)2, (U_LNG)0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowBumpLimitDisplay_Select_BumpInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_num	0x00~0xFF	0x00	0x01	0xFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)M_BUMP_INTVL_MAX, (U_LNG)M_BUMP_INTVL_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)M_BUMP_INTVL_MAX, (U_LNG)M_BUMP_INTVL_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK

FlowBumpLimitDisplay_Select_BumpLimitMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_zipu_num->ul_sub_mode_select_num == U_B_CHK_GAS_NUM)		○	○	-	-
	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_sub_mode_select_num, 1, (U_LNG)U_B_END_NUM, (U_LNG)U_B_CONDITION_NUM, stp_zipu_num->uc_sub_mode_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000004	0x00000004	0x00000000	0xFFFFFFFF
	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000004	0x00000004	0x00000000	0xFFFFFFFF
	stp_zipu_num->uc_sub_mode_select_updown	0~2	0	1	0	2
Output	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000003	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK	OK

FlowBumpLimitDisplay_Select_BumpReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root			-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff(stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0	0	0	0
Judgment			OK	OK	OK

FlowBumpLimitDisplay_Set_BumpLimitMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/15
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(stp_zipu_num->ul_sub_mode_select_num)		○	○	○	○	○	○
	case U_B_CONDITION_NUM		○	-	-	-	-	-
	case U_B_REMAINDER_NUM		-	○	-	-	-	-
	case U_B_INTERVAL_NUM		-	-	○	-	-	-
	case U_B_EXPRD_NUM		-	-	-	○	-	-
	case U_B_END_NUM		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000005	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Judgment			OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_bump_parameter_display.c

FlowBumpParameterDisplay_Disp_BumpCalOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowBumpParameterDisplay_Disp_BumpCalTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF
Judgment			OK

FlowBumpParameterDisplay_Disp_BumpLimit

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF
Judgment			OK

FlowBumpParameterDisplay_Disp_BumpTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF
Judgment			OK

FlowBumpParameterDisplay_End_BumpCalOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_bump_cal_flg	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowBumpParameterDisplay_End_BumpCalTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_bump_cal_time	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowBumpParameterDisplay_End_BumpLimit

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_bump_limit	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowBumpParameterDisplay_End_BumpTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_bump_time	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowBumpParameterDisplay_Init_BumpCalOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_cal_flg	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowBumpParameterDisplay_Init_BumpCalTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_cal_time	90~121	90	120	121
Output	stp_zipu_num->ul_item_select_num	90~121	90	120	121
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBumpParameterDisplay_Init_BumpLimit

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_limit	10~51	10	50	51
Output	stp_zipu_num->ul_item_select_num	10~51	10	50	51
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBumpParameterDisplay_Init_BumpParameterMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub2_mode_select_num	0	0
	stp_zipu_num->uc_sub2_mode_select_updown	0	0
Judgment			OK

FlowBumpParameterDisplay_Init_BumpTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_time	30~91	30	90	91
Output	stp_zipu_num->ul_item_select_num	30~91	30	90	91
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBumpParameterDisplay_Select_BumpCalOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0,1	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowBumpParameterDisplay_Select_BumpCalTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == ucc_m_bump_c_time[0])		○	-	-
	else		-	○	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x0000005A	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	90,120	120	90	90
Judgment			OK	OK	OK

FlowBumpParameterDisplay_Select_BumpLimit

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	○	○	○	-	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[0])		○	-	-	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[1])		-	○	-	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[2])		-	-	○	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[3])		-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[0])		-	-	-	-	-	○	-	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[1])		-	-	-	-	-	-	○	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[2])		-	-	-	-	-	-	-	○	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_limit[3])		-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	○
	Symbol	Range										
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x0000000A	0x00000014	0x0000001E	0x00000028	0xFFFFFFFF	0x0000000A	0x00000014	0x0000001E	0x00000028	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	10~50	20	30	40	50	10	50	10	20	30	40
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowBumpParameterDisplay_Select_BumpParameterMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_sub2_mode_select_num, 1, (U_LNG)U_B_S_END_NUM, (U_LNG)U_B_S_BUMP_TIME_NUM, stp_zipu_num->uc_sub2_mode_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_sub2_mode_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowBumpParameterDisplay_Select_BumpTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	○	○	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_time[0])		○	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_time[1])		-	○	-	-	-	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_time[2])		-	-	○	-	-	-	-	-
	else		-	-	-	○	-	-	-	-
	else		-	-	-	-	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_time[0])		-	-	-	-	○	-	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_time[1])		-	-	-	-	-	○	-	-
	else if(stp_zipu_num->ul_item_select_num == ucc_m_bump_b_time[2])		-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	0	0	1	1	1	2
	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x0000001E	0x0000002D	0x0000003C	0xFFFFFFFF	0x0000001E	0x0000002D	0x0000003C	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	45~90	45	60	90	30	90	30	45	60
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowBumpParameterDisplay_Set_BumpParameterMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(stp_zipu_num->ul_sub2_mode_select_num)		○	○	○	○	○	○
	case U_B_S_BUMP_TIME_NUM		○	-	-	-	-	-
	case U_B_S_BUMP_CHK_NUM		-	○	-	-	-	-
	case U_B_S_BCAL_TIME_NUM		-	-	○	-	-	-
	case U_B_S_BCAL_ONOFF_NUM		-	-	-	○	-	-
	case U_B_S_END_NUM		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	stp_zipu_num->ul_sub2_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_bump_test_display.c

FlowBumpTestDisplay_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uss_bump_test_time)		-	○	○
	else		○	-	-
	if(uss_bump_cal_time)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	uss_bump_test_time	0~9001	0	9000	9001
	uss_bump_cal_time	0~9001	0	9000	9001
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowBumpTestDisplay_Chk_BumpAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num) == M_ON)		○	○	○	○	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_BMP)		○	-	-	-	-	-
	else		-	○	○	○	-	-
	if(stp_data->uc_bump_cal_flg == M_ON)		-	○	○	-	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_SPN)		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		-	-	-	-	○	○
	Symbol	Range						
Input	m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num)	0~2	1	1	1	1	0	2
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x80	0x02	0x00	0x00	0x00	0xFF
	stp_data->uc_bump_cal_flg	0,1	1	1	1	0	0	0
Output	luc_ans	0,1	1	1	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

FlowBumpTestDisplay_Chk_BumpCalNecessity

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	-	-
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num) == M_ON)		○	○	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_BMP)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num)	0~2	1	1	0	2
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0xFF	0x00	0x00	0xFF
Output	luc_ans	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

FlowBumpTestDisplay_Chk_BumpCalOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_data->uc_bump_cal_flg == M_ON)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_data->uc_bump_cal_flg	0~2	0	1	2
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowBumpTestDisplay_Chk_BumpCalStartInitial

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->ul_cal_denst != M_OFF_VAL)		○	○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_cal_group != DataStructConst_CAL_GROUP_A)		○	-	-	-	-	-	-
	if(luc_ans == M_OK)		-	○	○	○	○	○	○
	if(puc_type == 0)		-	○	○	○	-	○	-
	if(stp_data->uc_bump_end_initial != M_ON)		-	○	○	○	-	-	-
	else		-	-	-	-	○	-	○
	if(stp_data->uc_cal_end_initial != M_ON)		-	-	-	-	○	-	-
	Symbol	Range							
Input	puc_type	0,1	0	0	0	0	1	0	1
	uc_bump_end_initial	1,2	2	2	2	2	2	1	2
	uc_cal_end_initial	1,2	2	2	2	2	2	2	1
	uc_gas_comb	0,1	1	0	1	1	1	1	1
	ul_cal_denst	0x00000000~0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	uc_cal_group	0,1	1	1	1	0	0	0	0
Output	luc_ans	0,1	0	1	1	1	1	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK

FlowBumpTestDisplay_Chk_BumpCalTimeEnd

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uss_bump_cal_time != 0)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	uss_bump_cal_time	0~9001	0	9000	9001
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

FlowBumpTestDisplay_Chk_BumpGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	FlowGasCalDisplay_Chk_AutoCalMenuGasOn (M_ON, stp_zipu_num->ul_sub_mode_select_num)	0~2	0	1	2
Output	luc_ans	0~2	0	1	2
Judgment			OK	OK	OK

FlowBumpTestDisplay_Chk_BumpSuccess

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num) == M_ON)		○	○	○	○
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_BMP)		○	-	○	○
	if(stp_data->uc_bump_cal_flg == M_ON)		○	-	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_SPN)		○	-	-	-
	else		-	-	-	○
	Symbol	Range				
Input	uc_bump_cal_flg	1,2	1	1	1	2
	uc_state_trb	0,255	255	255	0	255
	m_Chk_BumpPermission()	1,2	1	2	1	1
	ul_sub_mode_select_num	0	0	0	0	0
Output	luc_ans	0,1	1	0	0	1
Judgment			OK	OK	OK	OK

FlowBumpTestDisplay_Chk_BumpTestResult

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num))		○	○	-	-
	if(luc_ans == M_OK)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num))	0~2	1	1	0	2
	m_judg_BumpTest(luc_i)	0~2	0	1	1	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowBumpTestDisplay_Chk_BumpTestTimeEnd

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uss_bump_test_time != 0)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	uss_bump_test_time	0~9001	0	9000	9001
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

FlowBumpTestDisplay_Chk_HCModeWarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(m_Chk_BumpPermission(0, stp_zipu_num->ul_sub_mode_select_num) == M_ON)		○	○	○	○	○	-	-
	if(stp_data->uc_bump_cal_flg == M_ON)		○	○	○	○	-	-	-
	if(((stp_denst+0)->uc_state_trb & DataStructConst_S_FAIL_SPN) == 0x00)		○	○	○	-	-	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (stp_GasInfo_addr->uc_gas_comb <= 0x02))		○	○	-	-	-	-	-
	if(stp_data->uc_nc_mode_flg == GAS_CALC_NC_MODE_STINT)		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	○
	Symbol	Range							
Input	m_Chk_BumpPermission(0, stp_zipu_num->ul_sub_mode_select_num)	0~2	1	1	1	1	1	0	2
	stp_data->uc_bump_cal_flg	0~2	1	1	1	1	0	0	2
	(stp_denst+0)->uc_state_trb	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF
	stp_GasInfo_addr->uc_gas_comb	0~2	1	1	0	0	0	0	2
	stp_data->uc_nc_mode_flg	0~2	1	0	0	0	0	0	2
Output	luc_ans	0,1	1	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

FlowBumpTestDisplay_Com_BumpTest

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	○	-	○
	if((stp_GasInfo_addr+luc_i)->ul_cal_denst != M_OFF_VAL)		○	○	○	-	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-	-
	if(luc_ans == M_OK)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	-	○
	else		-	-	-	-	○	-
	Symbol	Range						
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	29	29	0	30
	(stp_GasInfo_addr+luc_i)->ul_cal_denst	0~0xDDDDDDDD	0	0	50000	0xDDDDDDDD	50000	50001
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0x00	0xFF	0xFF	0xFF	0xFF
	m_judg_BumpTest(luc_i)	0~2	0	1	0	0	0	2
Output	luc_result	0~12	0	7	0	0	0	12
Judgment			OK	OK	OK	OK	OK	OK

FlowBumpTestDisplay_Disp_BumpGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->ul_sub_mode_select_num == DataStructConst_CAL_GROUP_END)		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○
	for(luc_i=0; luc_i<4; luc_i++)		-	○	○	○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	○	-	-	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_cal_group == stp_zipu_num->ul_sub_mode_select_num)		-	○	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-
	else		-	-	-	○	○	○	○	○	○
	switch(ZIPCNumSet_Get_DispFlicker2())		-	○	○	○	○	○	○	○	○
	case 0		-	○	-	-	-	-	-	-	-
	case 1		-	-	○	○	○	○	○	○	-
	switch(stp_zipu_num->ul_sub_mode_select_num)		-	-	○	○	○	○	○	○	○
	case DataStructConst_CAL_GROUP_A		-	-	○	-	-	-	-	-	-
	case DataStructConst_CAL_GROUP_B		-	-	-	○	-	-	-	-	-
	case DataStructConst_CAL_GROUP_C		-	-	-	-	○	-	-	-	-
	case DataStructConst_CAL_GROUP_D		-	-	-	-	-	○	-	-	-
	case DataStructConst_CAL_GROUP_E		-	-	-	-	-	-	○	-	-
	default		-	-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	-	○
	Symbol	Range									
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000000	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF	0xFFFFFFFF
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	30	29	29	0	0	0	0	0	0
	(stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0xFF	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0x00	0x00
	ZIPCNumSet_Get_DispFlicker2()	0~2	2	0	1	1	1	1	1	1	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowBumpTestDisplay_Disp_BumpResult

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○	○	○	○	○	○	○
	switch(luc_i)		○	○	○	○	○	○	○	○	○	○
	case 0		○	○	○	○	○	○	○	○	○	○
	case 1		○	○	○	○	○	○	○	○	○	○
	case 2		○	○	○	○	○	○	○	○	○	○
	default		○	○	○	○	○	○	○	○	○	○
	if(stp_zipu_num->ul_sub2_mode_select_num == 0)		○	○	○	○	○	-	-	-	-	-
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num))		○	○	○	○	-	-	-	-	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_BMP)		○	○	○	-	-	-	-	-	-	-
	if(stp_data->uc_bump_cal_flg == M_ON)		○	○	-	-	-	-	-	-	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_SPN)		○	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-
	else if(stp_zipu_num->ul_sub2_mode_select_num == 1)		-	-	-	-	-	○	○	-	-	-
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num))		-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	○	○	○
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num))		-	-	-	-	-	-	-	○	○	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_BMP)		-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	○
	Symbol	Range										
Input	stp_zipu_num->ul_sub2_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000001	0x00000001	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num)	0,1	2	1	1	1	0	1	0	1	1	0
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0xFF	0x80	0xFF	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF
	stp_data->uc_bump_cal_flg	0~2	1	1	0	0	0	0	0	0	0	2
Output	stp_nl_disp->uc_lcd_mes[luc_y][luc_x+1]	70,80	70	70	80	80	80	80	80	80	80	80
	stp_nl_disp->uc_lcd_mes[luc_y][luc_x+3]	70,80	70	80	80	80	80	80	80	80	80	80
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowBumpTestDisplay_Disp_BumpTest

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if(m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num))		○	○	-	-
	if(puc_num == 0 || ((stp_denst+luc_i)-> uc_state_trb & DataStructConst_S_FAIL_BMP))		○	-	-	-
	else		○	○	-	-
	else		-	-	○	-
	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○	○
	case 0		○	○	-	-
	if(puc_num == 0)		○	-	-	-
	else		-	○	-	-
	case 1		-	-	○	-
	default		-	-	-	○
	if(puc_num == 0)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	puc_num	0~2	0	1	1	2
	m_Chk_BumpPermission(luc_i, stp_zipu_num->ul_sub_mode_select_num)	0~2	1	1	0	2
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0xFF	0x00	0x00	0xFF
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	0	1	2
Output	stp_nl_disp->uc_lcd_time[0]	0x20	0x20	0x20	0x20	0x20
	stp_nl_disp->uc_lcd_time[1]	0x20	0x20	0x20	0x20	0x20
	stp_nl_disp->uc_lcd_time[2]	0x20	0x20	0x20	0x20	0x20
	stp_nl_disp->uc_lcd_time[3]	0x20	0x20	0x20	0x20	0x20
	stp_nl_disp->uc_lcd_time_blink	0xFF	0xFF	0xFF	0xFF	0xFF
	stp_nl_disp->uc_lcd_time_dot	0x00	0x00	0x00	0x00	0x00
Judgment			OK	OK	OK	OK

FlowBumpTestDisplay_Disp_SetBumpCal

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowBumpTestDisplay_Execution_BumpCal

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowBumpTestDisplay_Get_BumpCalTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_cal_time	90~121	90	120	121
	stp_data->uc_bump_time	30~91	30	90	91
Output	uss_bump_cal_time	59546, 62536	59536	62536	62536
Judgment			OK	OK	OK

FlowBumpTestDisplay_Get_BumpGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
Judgment			OK

FlowBumpTestDisplay_Get_BumpResultDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~5	0	4	5
Output	uls_bump_result_denst[0][puc_num]	0	0	0	0
Judgment			OK	OK	OK

FlowBumpTestDisplay_Get_BumpSDMSelear

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowBumpTestDisplay_Get_BumpTestTime

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_bump_time	30~91	30	90	91
Output	uss_bump_test_time	3000~9100	3000	9000	9100
Judgment			OK	OK	OK

FlowBumpTestDisplay_Init_BumpTest

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<DataStructConst_CAL_GROUP_END; luc_i++)		○	○	○
	for(luc_j=0; luc_j<HDDEF_GAS_MAX; luc_j++)		○	○	○
	if((stp_GasInfo_addr+luc_j)->uc_gas_comb != 0x00)		○	○	-
	if(luc_i == (stp_GasInfo_addr+luc_j)->uc_cal_group)		○	-	-
	else		-	○	-
	else		-	-	○
	if(luc_ans == M_ON)		○	-	-
	else		-	○	○
	for(luc_i = 0U; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_j)->uc_gas_comb	0~30	29	30	0
	(stp_GasInfo_addr+luc_j)->uc_cal_group	0x00,0xFF	0x00	0xFF	0xFF
Output	stp_zipu_num->ul_sub_mode_select_num	0,5	0	5	5
	stp_zipu_num->uc_sub_mode_select_updown	0	0	0	0
	stp_zipu_num->ul_sub2_mode_select_num	0	0	0	0
	stp_zipu_num->uc_sub2_mode_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowBumpTestDisplay_SelectUp_BumpGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_zipu_num->ul_sub_mode_select_num > DataStructConst_CAL_GROUP_END)		○	-	-	-	-
	else		-	○	○	○	○
	for(luc_i; luc_i < DataStructConst_CAL_GROUP_END; luc_i++){		-	○	○	○	○
	if(FlowGasCalDisplay_Chk_AutoCalMenuGasOn(M_OFF, luc_i) == M_OK)		-	-	○	-	○
	Symbol	Range					
Input	ul_sub_mode_select_num	3~6	6	5	3	4	5
	FlowGasCalDisplay_Chk_AutoCalMenuGasOn()	0,1	0	0	0	0	1
Output	ul_sub_mode_select_num	4~6	6	6	4	5	5
Judgment			OK	OK	OK	OK	OK

FlowBumpTestDisplay_Select_BumpGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	○	-	-	-
	if(stp_zipu_num->ul_sub_mode_select_num == DataStructConst_CAL_GROUP_END)		○	-	-	-	-
	else		-	○	-	-	-
	for(luc_i; luc_i < DataStructConst_CAL_GROUP_END; luc_i++)		○	-	-	-	-
	if(FlowGasCalDisplay_Chk_AutoCalMenuGasOn(M_OFF, luc_i) == M_OK)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	if(stp_zipu_num->ul_sub_mode_select_num == 0)		-	-	○	-	-
	else		-	-	-	○	○
	for(luc_i; luc_i > 0; luc_i--)		-	-	-	○	○
	if(FlowGasCalDisplay_Chk_AutoCalMenuGasOn(M_OFF, luc_i) == M_OK)		-	-	-	○	-
	else		-	-	-	-	○
	if((luc_i == 0) && (FlowGasCalDisplay_Chk_AutoCalMenuGasOn (M_OFF, luc_i) == M_NG))		-	-	-	-	○
	else		-	-	-	○	-
	Symbol	Range					
Input	stp_zipu_num->uc_sub_mode_select_updown	0~2	0	0	1	1	2
	stp_zipu_num->ul_sub_mode_select_num	0x00000000,0xFFFFFFFF	0x00000005	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	FlowGasCalDisplay_Chk_AutoCalMenuGasOn (M_OFF, luc_i)	0~2	0	1	2	0	1
Output	stp_zipu_num->ul_sub_mode_select_num	5~254	6	5	6	254	6
Judgment			OK	OK	OK	OK	OK

FlowBumpTestDisplay_Select_BumpResult

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(FlowBumpTestDisplay_Chk_BumpAlarm() == M_ON)		○	○	○	-	-
	if(stp_data->uc_bump_cal_flg == M_ON)		○	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	if(luc_flg == M_ON)		○	○	-	-	-
	if(stp_zipu_num->ul_sub2_mode_select_num > 2)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	if(stp_zipu_num->ul_sub2_mode_select_num > 1)		-	-	○	-	○
	else		-	-	-	○	-
	Symbol	Range					
Input	FlowBumpTestDisplay_Chk_BumpAlarm()	0~2	1	1	1	0	2
	stp_data->uc_bump_cal_flg	0~2	1	1	0	0	2
	stp_zipu_num->ul_sub2_mode_select_num	0x00000000~0xFFFFFFFF	0x00000002	0x00000000	0x00000002	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_sub2_mode_select_num	0,1	0	1	0	1	0
Judgment			OK	OK	OK	OK	OK

FlowBumpTestDisplay_Set_BumpCalData

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~5	0	4	5
	stp_denst->ul_m_denst	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_denst->uc_m_denst_minus_flg	0~2	0	1	2
	stp_denst->uc_m_over_flg	0~2	0	1	2
	stp_denst->uc_m_over_minus_flg	0^2	0	1	2
Output	uls_bump_result_denst[1][puc_num]	0	0	0	0
	uls_bump_result_minus_flg[1][puc_num]	0	0	0	0
	uls_bump_result_over_flg[1][puc_num]	0	0	0	0
	uls_bump_result_over_minus_flg[1][puc_num]	0	0	0	0
Judgment			OK	OK	OK

FlowBumpTestDisplay_Set_BumpErrorOff

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i = 0U; luc_i < HDDEF_GAS_MAX; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowBumpTestDisplay_Set_BumpSDMSelear

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Chk_BumpPermission

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if((stp_GasInfo_addr+puc_num)->uc_gas_comb != 0x00)		○	○	○	○	-
	if((stp_GasInfo_addr+puc_num)->ul_cal_denst != M_OFF_VAL)		○	○	○	-	-
	if(!((stp_denst+puc_num)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-
	if((stp_GasInfo_addr+puc_num)->uc_cal_group == stp_zipu_num->ul_sub_mode_select_num)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	puc_num	0~5	0	0	0	4	5
	puc_flg	0,2	0	0	0	0	2
	(stp_GasInfo_addr+puc_num)->uc_gas_comb	0~30	30	29	29	29	0
	(stp_GasInfo_addr+puc_num)->ul_cal_denst	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xDDDDDDDD	0x00000000
	(stp_denst+puc_num)->uc_state_trb	0x00~0xFFFF	0x00	0x00	0xFF	0x00	0xFFFF
	(stp_GasInfo_addr+puc_num)->uc_cal_group	0x00~0xFFFF	0x00	0x00	0xFF	0x00	0xFFFF
	stp_zipu_num->ul_sub_mode_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	luc_ans	0,1	1	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_judg_BumpTest

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	12
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12
Root	if(puc_num == 1)		○	○	○	○	○	○	-	-	-	-	-	-
	if((lfl_cal_denst + lfl_lim_data) >= stp_denst->ul_m_denst)		○	○	○	○	○	-	-	-	-	-	-	-
	if((lfl_cal_denst >= lfl_lim_data) && (stp_denst->uc_m_denst_minus_flg == 0))		○	○	-	-	-	-	-	-	-	-	-	-
	if(((lfl_cal_denst - lfl_lim_data) <= stp_denst->ul_m_denst) && (stp_denst->uc_m_denst_minus_flg == 0))		○	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	○	○	-	-	-	-	-	-	-
	if(stp_denst->uc_m_denst_minus_flg == 0)		-	-	○	-	-	-	-	-	-	-	-	-
	else if(((lfl_lim_data - lfl_cal_denst) >= stp_denst->ul_m_denst) && (stp_denst->uc_m_denst_minus_flg == 1))		-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	○	○	○	○	○	○
	if((lfl_cal_denst + lfl_lim_data) >= stp_denst->ul_m_denst)		-	-	-	-	-	-	○	○	○	○	○	-
	if((lfl_cal_denst >= lfl_lim_data) && (stp_denst->uc_m_denst_minus_flg == 0))		-	-	-	-	-	-	○	○	-	-	-	-
	if(((lfl_cal_denst - lfl_lim_data) <= stp_denst->ul_m_denst) && (stp_denst->uc_m_denst_minus_flg == 0))		-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	○	○	-
	if(stp_denst->uc_m_denst_minus_flg == 0)		-	-	-	-	-	-	-	-	○	-	-	-
	else if(((lfl_lim_data - lfl_cal_denst) >= stp_denst->ul_m_denst) && (stp_denst->uc_m_denst_minus_flg == 1))		-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range												
Input	puc_num	0~5	1	1	1	1	1	1	0	0	0	0	4	5
	stp_GasInfo_addr->ul_cal_denst	0x00000000~0x000000C7	0x000000C7	0x000000C7	0x00000000	0x00000000	0x00000000	0x00000000	0x0000000A	0x0000000A	0x0000000A	0x0000000A	0x0000000A	0x00000000
	stp_data->uc_bump_limit	0x0A~0xFF	0x0A	0x0A	0x32	0x32	0x32	0x0A	0x32	0x32	0xFF	0xFF	0xFF	0xFF
	stp_denst->ul_m_denst	0x00000000~0xFFFFFFFF	0x000000C7	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0x00000008	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	stp_denst->uc_m_denst_minus_flg	0~2	0	0	0	1	1	0	0	0	0	1	2	2
Output	luc_ans	0,1	1	1	1	1	1	1	0	1	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_BumpResult

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_result == M_OK)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_num	0~5	4	0	5
	puc_result	0~2	0	1	2
	stp_denst->ul_m_denst	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	stp_denst->uc_m_denst_minus_flg	0~2	0	1	2
	stp_denst->uc_m_over_flg	0~2	0	1	2
	stp_denst->uc_m_over_minus_flg	0~2	0	1	2
Output	uls_bump_result_denst[0][puc_num]	0	0	0	0
	uls_bump_result_minus_flg[0][puc_num]	0,1	0	1	1
	uls_bump_result_over_flg[0][puc_num]	0,1	0	1	1
	uls_bump_result_over_minus_flg[0][puc_num]	0,1	0	1	1
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_cal_limit_display.c

FlowCalLimitDisplay_Disp_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○
	case FuncCalLimit_CAL_CONFIRM		○	-	-	-
	case FuncCalLimit_CAL_NOTUSE		-	○	-	-
	case FuncCalLimit_CAL_TOEFFECT		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_blink	127	127	127	127	127
	stp_nl_disp->uc_lcd_com_dot_blink	127,255	127	127	255	127
Judgment			OK	OK	OK	OK

FlowCalLimitDisplay_Disp_CalInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF
Judgment			OK

FlowCalLimitDisplay_Disp_CalReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowCalLimitDisplay_End_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_cal_lmt_chk	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowCalLimitDisplay_End_CalInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->us_cal_lmt_day	0x0000,0xFFFF	0x00000	0xFFFF
Judgment			OK	OK

FlowCalLimitDisplay_End_CalReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_cal_lmt_set	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowCalLimitDisplay_Init_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_cal_lmt_chk	0~3	0	2	3
Output	stp_zipu_num->ul_item_select_num	0~3	0	2	3
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowCalLimitDisplay_Init_CalInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_cal_lmt_day	1~1001	1	1000	1001
Output	stp_zipu_num->ul_item_select_num	1~1001	1	1000	1001
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowCalLimitDisplay_Init_CalLimitMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
Judgment			OK

FlowCalLimitDisplay_Init_CalReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_cal_lmt_set	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowCalLimitDisplay_Select_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)FuncCalLimit_CAL_TOEFFECT, (U_LNG)FuncCalLimit_CAL_CONFIRM, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowCalLimitDisplay_Select_CalInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_num	0x00~0xFF	0x00	0x01	0xFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)M_CAL_INTVL_MAX, (U_LNG)M_CAL_INTVL_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)M_CAL_INTVL_MAX, (U_LNG)M_CAL_INTVL_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK

FlowCalLimitDisplay_Select_CalLimitMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_zipu_num->ul_sub_mode_select_num == U_C_CHK_GAS_NUM)		○	○	-	-
	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_sub_mode_select_num, 1, (U_LNG)U_C_END_NUM, (U_LNG)U_C_REMAINDER_NUM, stp_zipu_num->uc_sub_mode_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000003	0x00000003	0x00000000	0xFFFFFFFF
	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000003	0x00000003	0x00000000	0xFFFFFFFF
	stp_zipu_num->uc_sub_mode_select_updown	0~2	0	1	1	2
Output	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000004	0x00000002	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK	OK

FlowCalLimitDisplay_Select_CalReminder

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0,1	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowCalLimitDisplay_Set_CalLimitMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/16
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(stp_zipu_num->ul_sub_mode_select_num)		○	○	○	○	○
	case U_C_REMAINDER_NUM		○	-	-	-	-
	case U_C_INTERVAL_NUM		-	○	-	-	-
	case U_C_EXPRD_NUM		-	-	○	-	-
	case U_C_END_NUM		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000004	0xFFFFFFFF
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_com_display.c

FlowComDisplay_Disp_SDMMode

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowComDisplay_Set_ComConnect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_disp_display.c

FlowDispDisplay_Disp_AlarmPointData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8
Root	switch(stp_zipu_num->ul_sub_mode_select_num)	○	○	○	○	○	○	○	○
	case 0	○	-	-	-	-	-	-	-
	case 1	-	○	-	-	-	-	-	-
	case 2	-	-	○	-	-	-	-	-
	case 3	-	-	-	○	-	-	-	-
	case 4	-	-	-	-	○	-	-	-
	case 5	-	-	-	-	-	○	-	-
	case 6	-	-	-	-	-	-	○	-
	default	-	-	-	-	-	-	-	○
	Symbol	Range							
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005	0x00000006	0xFFFFFFFF
Output	None	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowDispDisplay_Disp_BumpData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot[0]	0x10	0x10
	stp_nl_disp->uc_lcd_mes_dot[0]	0x04	0x04
Judgment			OK

FlowDispDisplay_Disp_CalData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot[0]	0x10	0x10
	stp_nl_disp->uc_lcd_mes_dot[0]	0x04	0x04
Judgment			OK

FlowDispDisplay_Disp_HCGasSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_blink	0x7F	0x7F
Judgment			OK

FlowDispDisplay_Disp_HCLongSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x08	0x08
Judgment			OK

FlowDispDisplay_Disp_PeakClearHold

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowDispDisplay_Disp_PeakClearRelease

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowDispDisplay_Disp_PumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○
	case 0		○	-	-
	case 1		-	○	-
	default		-	-	○
	Symbol	Range			
Input	uc_lcd_mes[0][0]	0	0	0	0
	uc_lcd_com[1]	0	0	0	0
	ZIPCNumSet\Get_DispFlicker2()	0~2	0	1	2
Output	uc_lcd_com_dot	0~16	8	16	0
Judgment			OK	OK	OK

FlowDispDisplay_Disp_SnapData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_zipu_num->uc_chr_temporary_item[0] == M_OFF)		○	○	○	○
	else		-	-	-	-
	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ul_sub_mode_select_num	0	0	0	0	0
	ul_item_select_num	0~2	0	0	1	2
	uc_chr_temporary_item[0]	0,1	0	1	1	1
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowDispDisplay_Disp_StationIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_blink	0x7F	0x7F
Judgment			OK

FlowDispDisplay_Disp_UserIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_blink	0x7F	0x7F
Judgment			OK

FlowDispDisplay_End_HCGasSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	(stp_GasInfo_addr+0)->uc_sub_comb	0x00, 0xFF	0x00	0xFF
Judgment			OK	OK

FlowDispDisplay_End_HCLongSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_long_energy_flg	0x00, 0xFF	0x00	0xFF
Judgment			OK	OK

FlowDispDisplay_End_PumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(FlowProcesDisplay_Get_PumpOffFlag() == true)		○	-
	for(luc_i=0U; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-
	for(luc_j=0U; luc_j<15U; luc_j++)		○	-
	for(luc_j=0U; luc_j<9U; luc_j++)		○	-
	else		-	○
	Symbol	Range		
Input	FlowProcessDisplay_Get_PumpOffFlag()	1,2	1	2
Output	ul_60_total	0	0	0
	ul_ave	0	0	0
	ul_ave_total_cnt	0	0	0
	ul_stel_point	0	0	0
	ul_stel_buf	0	0	0
	ul_stel	0	0	0
	ul_twa	0	0	0
	ul_60min_buf	0	0	0
	uc_60min_count	0	0	0
Judgment			OK	OK

FlowDispDisplay_End_StationIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_st_idx	0x00, 0xFF	0x00	0xFF
Judgment			OK	OK

FlowDispDisplay_End_UserIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_ur_idx	0x00, 0xFF	0x00	0xFF
Judgment			OK	OK

FlowDispDisplay_Get_LightOnPermission

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_data->uc_stealth_on == M_ON)		○	-
	else		-	○
	Symbol	Range		
Input	uc_stealth_on	1,2	1	2
Output	lbl_ans	0,1	0	1
Judgment			OK	OK

FlowDispDisplay_Init_AlarmPointData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
Judgment			OK

FlowDispDisplay_Init_BumpData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
Judgment			OK

FlowDispDisplay_Init_CalData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
Judgment			OK

FlowDispDisplay_Init_DispMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
Judgment			OK

FlowDispDisplay_Init_HCGasSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+0)->uc_sub_comb	0~29	0	29	30
Output	stp_zipu_num->ul_sub_mode_select_num	0~30	0	29	30
	stp_zipu_num->uc_sub_mode_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowDispDisplay_Init_HCLongSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_long_energy_flg	0~2	0	1	2
Output	stp_zipu_num->ul_sub_mode_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowDispDisplay_Init_PumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowDispDisplay_Init_SnapData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_data_logger_data->uc_snap_rec_pnt == 0U)		○	-	○	○	-
	else		-	○	-	-	○
	if(luc_snap_end_data_flg == M_ON)		○	○	-	-	-
	else		-	-	○	○	○
	if((stp_data_logger_data->uc_snap_rec_pnt == 0U) && (luc_snap_end_data_flg == M_OFF))		-	-	-	○	-
	else		○	○	○	-	○
	Symbol	Range					
Input	uc_snap_rec_pnt	0,1	0	1	0	0	1
	DataLogger_snap_end_data()	0~2	1	1	2	0	0
Output	ul_sub_mode_select_num	0,255	255	0	255	255	0
	uc_sub_mode_select_updown	0	0	0	0	0	0
	ul_item_select_num	0	0	0	0	0	0
	ul_item_select_updown	0	0	0	0	0	0
	uc_chr_temporary_item	0	0	0	0	0	0
	ul_long_temporary_item	0~4294967295	255	255	4294967295	4294967295	0
Judgment			OK	OK	OK	OK	OK

FlowDispDisplay_Init_StationIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_st_idx	0~127	0	127	128
Output	stp_zipu_num->ul_item_select_num	0~128	0	127	128
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowDispDisplay_Init_UserIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_ur_idx	0~128	0	127	128
Output	stp_zipu_num->ul_item_select_num	0~128	0	127	128
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowDispDisplay_Select_AlarmPointData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(GasCalc_Get_ToxicGasON() == M_ON)		○	○	-	-
	if(GasCalc_Get_Alarm1hON() == M_ON)		○	-	-	-
	else		-	○	-	-
	if((stp_zipu_num->ul_sub_mode_select_num == 5) && (GasCalc_Get_AlarmTwaON() == M_OFF))		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	GasCalc_Get_ToxicGasON()	0~2	1	1	0	2
	GasCalc_Get_Alarm1hON()	0~2	1	0	0	2
	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_sub_mode_select_num, 1, (U_LNG)luc_max, (U_LNG)0, stp_zipu_num->uc_sub_mode_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000005	0x00000000	0x00000000	0xFFFFFFFF
	GasCalc_Get_AlarmTwaON()	0~2	0	1	1	2
Output	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK	OK

FlowDispDisplay_Select_BumpData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowDispDisplay_Select_CalData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowDispDisplay_Select_DispMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(luc_i; luc_i<D_MAX_NUM; luc_i++)		○	○	○
	if(uc_disp_mode_item[luc_i] == M_ON)		○	-	-
	else		-	○	-
	if(luc_i == D_MAX_NUM)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	19	0xFFFFFFFF
	uc_disp_mode_item[luc_i]	0,1	1	0	2
Output	luc_end	0,1	0	1	0
Judgment			OK	OK	OK

FlowDispDisplay_Select_HCGasSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	○	○	○	-	-	-	-	-
	for(luc_i; luc_i<HDDEF_HC_TABLE_MAX; luc_i++)		○	○	○	-	-	-	-	-	-
	if(m_Chk_DispHCGasSel(luc_i) == M_ON)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	if(luc_i >= HDDEF_HC_TABLE_MAX)		-	○	○	○	-	-	-	-	-
	for(luc_i=0; luc_i<HDDEF_HC_TABLE_MAX; luc_i++)		-	○	-	○	-	-	-	-	-
	if(m_Chk_DispHCGasSel(luc_i) == M_ON)		-	-	-	○	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_sub_mode_select_num != 0)		-	-	-	-	○	○	○	-	○
	else		-	-	-	-	-	-	-	○	-
	for(luc_i; luc_i>0; luc_i--)		-	-	-	-	○	○	○	○	○
	if(m_Chk_DispHCGasSel(luc_i) == M_ON)		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	○
	if(luc_i == 0)		-	-	-	-	○	○	-	○	○
	for(luc_i=(HDDEF_HC_TABLE_MAX-1); luc_i>0; luc_i--)		-	-	-	-	○	○	-	○	○
	if(m_Chk_DispHCGasSel(luc_i) == M_ON)		-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	○	-	○	○
	else		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	stp_zipu_num->uc_sub_mode_select_updown	0~2	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x00000000	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	m_Chk_DispHCGasSel(luc_i)	0~2	1	0	1	1	1	0	1	0	2
Output	stp_zipu_num->ul_sub_mode_select_num	0~254	1	29	0	1	28	0	254	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowDispDisplay_Select_HCLongSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_sub_mode_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_sub_mode_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowDispDisplay_Select_SnapDetail

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->uc_chr_temporary_item[0] == M_OFF)		○	-	-
	else		-	○	○
	if(stp_zipu_num->ul_item_select_num == 0U)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	ul_item_select_num	0,1	0	0	1
	uc_chr_temporary_item[0]	0,1	0	1	1
Output	ul_item_select_num	0,1	0	1	0
Judgment			OK	OK	OK

FlowDispDisplay_Select_SnapRecData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->uc_chr_temporary_item[0] == M_OFF)		○	-	-
	else		-	○	○
	if(puc_num == 1U)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_num	1,2	1	1	2
	Tool_OneCount()	0	0	0	0
	ul_sub_mode_select_num	0	0	0	0
	ul_sub_mode_select_updown	0	0	0	0
	uc_chr_temporary_item[0]	0,1	0	1	1
	ul_long_temporary_item[0]	0	0	0	0
Output	ul_sub_mode_select_num	0	0	0	0
Judgment			OK	OK	OK

FlowDispDisplay_Select_SnapUpDownMode

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->uc_chr_temporary_item[0] == M_OFF)		○	-
	else		-	○
	Symbol	Range		
Input	uc_chr_temporary_item[0]	0,1	0	1
Output	None	-	-	-
Judgment			OK	OK

FlowDispDisplay_Select_StationIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_num	0x00~0xFF	0x00	0x01	0xFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)HDDEF_STATION_ID_NUM-1, 1, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000, 0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)HDDEF_STATION_ID_NUM-1, 0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000, 0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK

FlowDispDisplay_Select_UserIdSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_num	0x00~0xFF	0x00	0x01	0xFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)HDDEF_USER_ID_NUM-1, 0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)HDDEF_USER_ID_NUM-1, 0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK

FlowDispDisplay_Set_AlarmPointData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onoff	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowDispDisplay_Set_LightOnOffSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Chk_DispHCGasSel

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(Tool_Comp_Pnt(GasTableDefData[0][puc_num].uc_disp_name, (U_CHR *)&("****"), 3) != 0)		○	○	○	○	○	-	○
	if((GasTableDefData[0][puc_num].uc_uni_dec & 0xf0) != 0x00)		○	○	○	○	-	-	-
	if(stp_GasInfo_addr->uc_gas_comb != 0x02 || (HCTableDefData[puc_num].uc_meas_group & 0x01) == 0x00)		○	○	○	-	-	-	-
	if(stp_data->uc_nc_mode_flg == GAS_CALC_NC_MODE_FULL || (HCTableDefData[puc_num].uc_meas_group & 0x10) == 0x00)		○	○	-	-	-	-	-
	if((stp_data->uc_exp_flg != DSC_TGT_JG) || (HCTableDefData[puc_num].uc_jg_support == M_ON))		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	-	○	-	○
	else		-	-	-	-	-	○	-
	Symbol	Range							
Input	puc_num	0~30	15	15	4	1	29	0	30
	Tool_Comp_Pnt(GasTableDefData[0][puc_num].uc_disp_name, (U_CHR *)&("****"), 3)	0~2	1	1	1	1	1	0	2
	stp_GasInfo_addr->uc_gas_comb	0~30	0	0	0	2	2	29	30
	stp_data->uc_nc_mode_flg	0~2	0	0	1	1	1	1	2
	stp_data->uc_exp_flg	0~4	0	3	3	3	3	3	4
Output	luc_ans	0,1	1	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

m_Chk_DispMenuItem

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(GasCalc_Get_ToxicGasON() == M_ON)		○	○	-	-
	if(GasCalc_Get_AlarmTwaON() == M_ON)		○	-	-	-
	else		-	○	-	-
	if(GasCalc_Get_Alarm1hON() == M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	if(FuncCalLimit_Get_CalLimitOnOff() == M_ON)		○	-	-	-
	else		-	○	○	○
	if(stp_data->uc_bump_lmt_set == M_ON)		○	-	-	-
	else		-	○	○	○
	if(stp_data->uc_disp_setting_item_onoff == M_ON)		○	○	-	-
	if(GasCalc_Get_CombGasON(M_ON) == M_ON)		○	-	-	-
	else		-	○	-	-
	if(GasCalc_Get_CombGasON(M_OFF) == M_ON)		○	-	-	-
	else		-	○	-	-
	if(stp_data->uc_id_disp_onoff == M_ON)		○	-	-	-
	else		-	○	-	-
	if(stp_data->uc_ble_onoff == M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	GasCalc_Get_ToxicGasON()	0,1	1	1	0	2
	GasCalc_Get_AlarmTwaON()	0,1	1	0	0	2
	GasCalc_Get_Alarm1hON()	0,1	1	0	0	2
	FuncCalLimit_Get_CalLimitOnOff()	0,1	1	0	0	2
	stp_data->uc_bump_lmt_set	0,1	1	0	0	2
	stp_data->uc_disp_setting_item_onoff	0,1	1	1	0	2
	GasCalc_Get_CombGasON(M_ON)	0,1	1	0	0	2
	GasCalc_Get_CombGasON(M_OFF)	0,1	1	0	0	2
	stp_data->uc_id_disp_onoff	0,1	1	0	0	2
	stp_data->uc_ble_onoff	0,1	1	0	0	2
Output	uc_disp_mode_item[D_STEL_NUM]	0,1	1	1	0	0
	uc_disp_mode_item[D_TWA_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_1H_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_CAL_DATA_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_BUMP_DATA_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_GASSEL_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_HC_LONG_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_US_ID_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_ST_ID_NUM]	0,1	1	0	0	0
	uc_disp_mode_item[D_BLE_NUM]	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

m_Disp_1h

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((stp_GasInfo_addr+3)->uc_gas_comb != 0x00)		○		-	○
	if(!((stp_denst+3)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	-	-	-
	else		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	(stp_GasInfo_addr+3)->uc_gas_comb	0~30	29	29	0	30
	(stp_denst+3)->uc_state_trb	0x00,0xFF	0x00	0xFF	0xFF	0xFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_Disp_Light

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(LED_Get_LightOnOffStatus() != M_OFF)		○	-
	else		-	○
	Symbol	Range		
Input	uc_lcd_mes[0][0]	0	0	0
	uc_lcd_com[1]	0	0	0
	LED_Get_LightOnOffStatus()	0,1	1	0
Output	None	-	-	-
Judgment			OK	OK

m_Disp_Peak

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-	○
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-
	if(luc_i == 1)		○	○	-	-	-
	else		○	○	-	-	-
	if(lul_peak_flg)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	○
	else		-	-	-	-	-
	else		-	-	-	○	-
	Symbol	Range					
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	29	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0x00	0xFF	0xFF	0xFF
	(stp_denst+luc_i)->uc_pmin_flg	0~2	1	0	0	0	2
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_Disp_SnapConcentration

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	for(luc_i=0U; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○	○	○
	switch(*(lucp_gas_data + (luc_i * 28U)))		○	○	○	○	○	○	○	○
	case 0x32:		○	-	-	-	-	-	-	-
	case 0x33:		-	○	-	-	-	-	-	-
	case 0x34:		-	-	○	-	-	-	-	-
	case 0x35:		-	-	-	○	-	-	-	-
	default:		-	-	-	-	○	○	○	○
	if(*(lucp_gas_data + (luc_i * 28U)) == 0x39U)		-	-	-	-	-	○	-	○
	if(*(lucp_gas_data + (luc_i * 28U) + 7U) >= 0x02U)		-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○
	else		○	○	○	○	○	-	○	-
	if(luc_mode_disp == 0U)		○	○	○	○	○	-	○	○
	for(luc_i=0U; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	-	○	○
	if(luc_gas_disp[luc_i] != M_ON)		○	○	○	○	○	-	○	○
	else		○	○	○	○	-	-	-	-
	if(lus_concentration == M_OFF_CODE)		-	-	-	-	-	-	-	-
	else		○	○	○	○	-	-	-	-
	if(luc_minus_flg != 0U)		-	-	-	-	-	-	-	-
	else		○	○	○	○	-	-	-	-
	else		-	-	-	-	-	○	-	-
	Symbol	Range								
Input	puc_rec_pnt	0	0	0	0	0	0	0	0	0
	DataLogger_snap_read[60]	50~58	50	51	52	53	54	57	58	57
	DataLogger_snap_read[67]	1,2	2	2	2	2	2	2	2	1
	DataLogger_snap_read[75]	0	0	0	0	0	0	0	0	0
	DataLogger_snap_read[72]	0	0	0	0	0	0	0	0	0
	DataLogger_snap_read[71]	0	0	0	0	0	0	0	0	0
	uc_lcd_com[1]	0	0	0	0	0	0	0	0	0
Output	None	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

m_Disp_SnapDateTime

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	for(luc_i=0U; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○
	if(*(lucp_date_data + (luc_i * 28U)) == 0x39U)		○	-
	else		-	○
	Symbol	Range		
Input	puc_rec_pnt	0	0	0
	DataLogger_snap_read()	57,58	57	58
	uc_lcd_mes[0][1]	0	0	0
	uc_lcd_mes[0][4]	0	0	0
	uc_lcd_mes[0][6]	0	0	0
	uc_lcd_mes[1][4]	0	0	0
	uc_lcd_mes[1][6]	0	0	0
	uc_lcd_com[4]	0	0	0
Output	uc_lcd_mes_dot	4	4	4
Judgment			OK	OK

m_Disp_SnapNoData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	uc_lcd_com[1]	0	0
Output	None	-	-
Judgment			OK

m_Disp_Stel

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=2; luc_i<4; luc_i++)		○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	○	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-
	if((stp_GasInfo_addr+luc_i)->ul_stel_alarm != M_OFF_VAL)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	30	3	0
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x00	0xFF	0xFF	0xFF
	(stp_GasInfo_addr+luc_i)->ul_stel_alarm	0~0xDDDDDDDD	5000	0xDDDDDDDD	0	5001	0
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_Disp_Twa

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	for(luc_i=2; luc_i<4; luc_i++)		○	○	○	○	○	○	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	○	○	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	○	○	-	-	-
	if(!((luc_i == DataStructConst_CO_CH) && (stp_data->uc_1h_alarm_on == M_ON)))		○	○	○	-	-	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->ul_twa_alarm != M_OFF_VAL)		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	-
	Symbol	Range							
Input	stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	29	29	30	3	0
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF
	stp_data->uc_1h_alarm_on	0~2	1	1	1	0	0	0	2
	(stp_GasInfo_addr+luc_i)->ul_twa_alarm	0~0xDDDDDDDD	2500	0xDDDDDDDD	0	0	0	2501	0
Output	None	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK

m_Select_CalBumpData

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	-	○	○	○	-	-	-	-
	for(luc_i; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	○	○	○	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	-	-	○	-	-	-	-	-
	if(luc_i >= HDDEF_GAS_MAX)		○	-	○	○	○	-	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	○	○	○	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	-	○	○	○	-	-	-	-
	else		-	○	-	-	-	○	-	○	○
	if(stp_zipu_num->ul_sub_mode_select_num != 0)		-	○	-	-	-	○	-	○	○
	for(luc_i; luc_i>0; luc_i--)		-	○	-	-	-	○	-	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	○	-	-
	if((luc_i == 0) && ((stp_GasInfo_addr+luc_i)->uc_gas_comb == 0x00))		-	○	-	-	-	○	-	○	○
	for(luc_i=(HDDEF_GAS_MAX-1); luc_i>0; luc_i--)		-	-	-	-	-	-	-	-	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	-	-	-	-	-	-	-	○
	Symbol	Range									
Input	uc_gas_comb	0,1	1	1	1	0	0	1	1	0	0
	ul_sub_mode_select_num	0~3	2	2	3	2	3	1	0	2	1
	uc_sub_mode_select_updown	0,1	0	1	0	0	0	1	1	1	1
Output	ul_sub_mode_select_num	0~3	0	1	0	1	1	0	3	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowDispDisplay_Set_DispMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/17
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	50

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20	case21	case22
Root	switch(stp_dpu_num->ul_mode_select_num)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case D_LIGHT_NUM:		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_PEAK_NUM:		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_STEL_NUM:		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_TWA_NUM:		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_1H_NUM:		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_GASSEL_NUM:		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-
	case 0:		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	case D_HC_LONG_NUM:		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○
	if(stp_data->uc_long_energy_flg == M_ON)		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○
	case 0:		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_PUMP_OFF_NUM:		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_US_ID_NUM:		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_ST_ID_NUM:		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_CAL_DATA_NUM:		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_BUMP_DATA_NUM:		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_SNAP_DATA_NUM:		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_DATE_NUM:		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	if(stp_sub_ad->fl_disp_temp < 0)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case D_AL_P_NUM:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case D_BLE_NUM:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	if(stp_data->uc_ble_connection_flg == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	switch(ZIPCNumSet_Get_DispFlicker3())		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case 0:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case 1:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-

	Symbol	Range																						
Input	uc_ur_id[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_st_id[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_long_energy_flg	1,2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
	uc_ble_connection_flg	1,2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	uc_lcd_icon[3]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes[0][0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes[0][1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes[0][4]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes[0][6]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes[1][4]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes[1][6]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_com[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_com[5]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	fl_disp_temp	-1, 0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	ZIPCNumSet_Get_DispFlicker3()	0~3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	0	1
	ul_mode_select_num	0~17	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	17	18	5	5	5	6	6
Output	uc_lcd_mes_dot	0,4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
	uc_lcd_com_dot	0~16	0	0	0	0	0	0	8	8	0	0	16	16	16	0	0	0	0	0	0	0	8	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

case23	case24	case25	case26	case27	case28	case29	case30	case31	case32	case33	case34	case35	case36	case37	case38	case39	case40	case41	case42	case43	case44	case45	case46	case47	case48	case49	case50
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1
2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	0	1	2	3	0	1	2	3
6	6	7	7	7	8	8	8	9	9	9	10	10	10	11	11	11	12	12	12	13	14	14	14	17	17	17	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_fact_display.c

FlowFactDisplay_Chk_GasCombChannelSelect2chH2Num

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_zipu_num->ul_item_select_num == DataStructConst_CO_CAN_H2_H2_CH)		○	○	-	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_CO_CH)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000002	0x00000002	0x00000000	0xFFFFFFFF
	(stp_GasInfo_addr+DataStructConst_ CO_CAN_H2_CO_CH)->uc_gas_comb	0~30	3	0	29	30
Output	luc_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

FlowFactDisplay_Chk_GasCombChannelSelectEndNum

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000004	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowFactDisplay_Chk_RomSumDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((!(stp_zipu_num->ul_item_select_num & 0x00000001)) && (FuncRomChk_Chk_EndRomChk() == M_OFF))		○	-	-
	else		-	○	○
	if((!(stp_zipu_num->ul_item_select_num & 0x00000002)) && (RL78I1ESumChk_Get_PhaseFlg() == M_OFF))		○	-	-
	else		-	○	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	FuncRomChk_Chk_EndRomChk()	0~2	0	1	2
	RL78I1ESumChk_Get_PhaseFlg()	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0x00000003, 0xFFFFFFFF	0x00000003	0xFFFFFFFF	0xFFFFFFFF
Judgment			OK	OK	OK

FlowFactDisplay_Disp_GasCombChannelSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num < HDDEF_GAS_MAX)		○	○	-
	if((stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_gas_comb != 0x00)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000, 0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	(stp_GasInfo_addr+(stp_zipu_num-> ul_item_select_num))->uc_gas_comb	0~30	29	0	30
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowFactDisplay_Disp_GasCombGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_blink	0x7F	0x7F
Judgment			OK

FlowFactDisplay_Disp_RomSumDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(stp_data->uc_ble_onoff == M_ON)		-	-	-	-	○	○
	else		○	○	○	○	-	-
	switch(luc_num)		○	○	○	○	-	-
	case 0		○	-	-	-	-	-
	if(stp_zipu_num->ul_item_select_num & 0x00000001)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	case 1		-	-	○	○	-	-
	if(stp_zipu_num->ul_item_select_num & 0x00000002)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	case 2		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	stp_data->uc_ble_onoff	0~2	2	0	0	0	1	1
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	0	1	1	1	2
	stp_zipu_num->ul_item_select_num	0x00000000, 0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

FlowFactDisplay_Disp_RunningFactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowFactDisplay_Disp1_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~2	0	1	2	3
Output	uc_lcd_com_dot	0~32	8	16	32	0
Judgment			OK	OK	OK	OK

FlowFactDisplay_Disp2_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~2	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowFactDisplay_End_FlowAdjSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
	None	-	-
Judgment			OK

FlowFactDisplay_End_FlowAdjSettingAndCheck

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
	None	-	-
Judgment			OK

FlowFactDisplay_End_FlowCheck

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	ul_item_select_num	0,65535	0	65535
Output	us_flow_output	0,65535	0	65535
Judgment			OK	OK

FlowFactDisplay_End_GasCombGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if((stp_zipu_num->ul_item_select_num == DataStructConst_CO_CAN_H2_CO_CH) && (stp_zipu_num->ul_sub_item_select_num == DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	-	-	○	-	-	-	-
	else		○	○	○	○	-	○	○	○	○
	if((stp_zipu_num->ul_item_select_num == DataStructConst_CO_CAN_H2_CO_CH) && (stp_zipu_num->ul_sub_item_select_num != DataStructConst_CO_CAN_H2_GAS_NUM))		○	○	-	-	-	-	-	-	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_H2_CH)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		-	-	○	○	○	○	○	○	○
	if((stp_zipu_num->ul_item_select_num == DataStructConst_CO_CAN_H2_H2_CH) && (stp_zipu_num->ul_sub_item_select_num == DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	-	-	-	-	-	○	-
	else		○	○	○	○	○	○	○	-	○
	if((stp_zipu_num->ul_item_select_num == DataStructConst_CO_CAN_H2_H2_CH) && (stp_zipu_num->ul_sub_item_select_num != DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	○	○	-	-	-	-	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_CO_CH)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	○	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-
	else		○	○	-	-	○	○	○	○	○
	if((stp_zipu_num->ul_item_select_num == DataStructConst_H2S_CH) && (stp_zipu_num->ul_sub_item_select_num == DataStructConst_SINGLE_H2S_GAS_NUM))		-	-	○	○	-	-	-	-	-
	if((stp_GasInfo_addr+DataStructConst_CO_CH)->uc_gas_comb != 0x00)		-	-	○	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-
	else		○	○	-	-	○	○	○	○	○
	if((stp_zipu_num->ul_item_select_num == DataStructConst_CO_CH) && (stp_zipu_num->ul_sub_item_select_num != 0x00))		○	○	-	-	○	-	-	-	-
	if((stp_GasInfo_addr+DataStructConst_H2S_CH)->uc_gas_comb == DataStructConst_SINGLE_H2S_GAS_NUM)		-	○	-	-	-	-	-	-	-
	else		○	-	-	-	○	-	-	-	-
	else		-	-	○	○	-	○	○	○	○
	if(stp_zipu_num->ul_item_select_num < 4)		○	○	○	○	○	○	-	○	-
	else		-	-	-	-	-	-	○	-	○
	if(stp_zipu_num->ul_item_select_num == 4)		-	-	-	-	-	-	○	-	-
	else		○	○	○	○	○	○	-	○	○
	Symbol	Range									
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000003	0x00000003	0x00000002	0x00000002	0x00000003	0x00000000	0x00000004	0x00000002	0xFFFFFFFF
	stp_zipu_num->ul_sub_item_select_num	0x00000000~0xFFFFFFFF	0x00000002	0x00000002	0x00000002	0x00000002	0x00000003	0x00000000	0x00000004	0x00000003	0xFFFFFFFF
	(stp_GasInfo_addr+2)->uc_gas_comb	0~30	3	2	0	0	0	0	29	0	30
	(stp_GasInfo_addr+3)->uc_gas_comb	0~30	0	0	3	0	0	0	29	0	30
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowFactDisplay_Disp_FlowAdjSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(lfl_press_ad_val < 0.0F)		○	-
	else		-	○
	Symbol	Range		
Input	Ad_12bit_Get_Ad_Data()	-1,0	-1	0
	uc_lcd_mes[0][0]	0	0	0
	uc_lcd_mes[1][0]	0	0	0
	uc_lcd_com[1]	0	0	0
	ul_item_select_num	0	0	0
Output	uc_lcd_mes_blink	0	0	0
	uc_lcd_mes_dot	8	8	8
	uc_lcd_mes_dot_blink	0	0	0
	uc_lcd_com_dot	8	8	8
Judgment			OK	OK

FlowFactDisplay_Disp_FlowCheck

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(lfl_press_ad_val < 0.0F)		○	-	○
	else		-	○	-
	if(lfl_pump_off_val < 0.0F)		○	-	-
	else		-	○	○
	if(lbl_check_result == true)		○	-	-
	else		-	○	○
	if(lbl_check_result == true)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	Ad_12bit_Get_Ad_Data()	-1,0	-1	0	-1
	PumpFlow_Check_FlowTest()	0,1	0	1	1
	PumpFlow_Check_PumpTest()	0,1	0	0	1
	PumpFlow_Get_PumpOffPressValue()	-1,0	0	0	-1
Output	uc_lcd_mes_dot	8	8	8	8
	uc_lcd_com_dot	0,8	0	0	8
Judgment			OK	OK	OK

FlowFactDisplay_Init_ADDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_item_select_num	0	0
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowFactDisplay_Init_FactMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_mode_select_num	0	0
	stp_zipu_num->uc_mode_select_updown	0	0
Judgment			OK

FlowFactDisplay_Init_FlowAdjSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	us_flow_output	0	0
Output	ul_item_select_num	0	0
	uc_item_select_updown	0	0
Judgment			OK

FlowFactDisplay_Init_FlowCheck

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
	None	-	-
Judgment			OK

FlowFactDisplay_Init_GasCombChannelSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_item_select_num	0	0
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowFactDisplay_Init_GasCombGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_gas_comb	0~29	0	29	30
Output	stp_zipu_num->ul_sub_item_select_num	0~30	0	29	30
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowFactDisplay_Init_RomSumDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_item_select_num	0	0
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowFactDisplay_Select_ADDispNum

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)9, (U_LNG)0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFF
Judgment			OK	OK

FlowFactDisplay_Select_FactMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_mode_select_updown == M_OFF)	○	○	○	○	-	-	-	-	-
	for(luc_i; luc_i<F_MAX_NUM; luc_i++)	○	○	○	-	-	-	-	-	-
	if(uc_fact_mode_item[luc_i] == M_ON)	○	-	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-	-
	if(luc_i >= F_MAX_NUM)	-	○	○	○	-	-	-	-	-
	for(luc_i=0; luc_i<F_MAX_NUM; luc_i++)	-	○	-	○	-	-	-	-	-
	if(uc_fact_mode_item[luc_i] == M_ON)	-	-	-	○	-	-	-	-	-
	else	-	○	-	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_mode_select_num != 0)	-	-	-	-	○	○	○	-	○
	for(luc_i; luc_i>0; luc_i--)	-	-	-	-	-	-	-	○	-
	if(uc_fact_mode_item[luc_i] == M_ON)	-	-	-	-	○	○	○	○	○
	else	-	-	-	-	-	-	○	-	-
	else	-	-	-	-	-	-	-	○	○
	if((luc_i == 0) && (uc_fact_mode_item[0] != M_ON)	-	-	-	-	○	○	-	○	○
	for(luc_i=F_END_NUM; luc_i>0; luc_i--)	-	-	-	-	○	○	-	○	○
	if(uc_fact_mode_item[luc_i] == M_ON)	-	-	-	-	-	○	-	-	-
	else	-	-	-	-	○	-	-	○	○
	else	-	-	-	-	-	-	○	-	-
	Symbol	Range								
Input	stp_zipu_num->uc_mode_select_updown	0~2	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x00000000	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	uc_fact_mode_item[luc_i]	0~2	1	0	1	1	1	0	1	0	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowFactDisplay_Select_FlowAdjSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

		case1	case2
Root	if(puc_num == 1U)	○	-
	else	-	○
	Symbol	Range	
Input	puc_num	1,2	1	2
	Tool_OneCount()	0	0	0
	ul_item_select_num	0	0	0
	uc_item_select_updown	0	0	0
Output	ul_item_select_num	0	0	0
Judgment			OK	OK

FlowFactDisplay_Select_GasCombChannelSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

		case1	case2	case3	case4	case5
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)	○	○	-	-	
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX-1)	○	-	-	-	-
	else	-	○	-	-	-
	else	-	-	○	○	○
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX-1)	-	-	○	-	-
	else	-	-	-	○	○
	Symbol	Range				
Input	Tool_OneCount(stp_zipu_num->ul_item_select_num, 1, HDDEF_GAS_MAX, 0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000004	0x00000000	0x00000004	0xFFFFFFFF	0xFFFFFFFF
	stp_zipu_num->uc_item_select_updown	0,1	0	0	1	1	2
Output	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000000	0x00000003	0xFFFFFFFF	0xFFFFFFFF
Judgment			OK	OK	OK	OK	OK

FlowFactDisplay_Select_GasCombGasSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_zipu_num->uc_sub_item_select_updown == M_OFF)		○	○	-	-	-
	for(luc_i=(stp_zipu_num->ul_sub_item_select_num + 1); luc_i <HDDEF_GAS_TABLE_MAX; luc_i++)		○	○	-	-	-
	if(Tool_Comp_Pnt(&(GasTableDefData[stp_zipu_num->ul_item_select_num][luc_i]. uc_name[0]), (U_CHR *)"**********", 10) != 0)		○	-	-	-	-
	else		-	○	-	-	-
	if(luc_i == HDDEF_GAS_TABLE_MAX)		-	○	-	-	-
	else		○	-	-	-	-
	else		-	-	○	○	○
	if(stp_zipu_num->ul_sub_item_select_num == 0)		-	-	○	-	-
	else		-	-	-	○	○
	for(luc_i; luc_i > 0; luc_i--)		-	-	○	○	○
	if(Tool_Comp_Pnt(&(GasTableDefData[stp_zipu_num->ul_item_select_num][luc_i]. uc_name[0]), (U_CHR *)"**********", 10) != 0)		-	-	○	-	○
	else		-	-	-	○	-
	Symbol	Range					
Input	stp_zipu_num->uc_sub_item_select_updown	0~2	0	0	1	1	2
	Tool_Comp_Pnt(&(GasTableDefData [stp_zipu_num->ul_item_select_num][luc_i]. uc_name[0]), (U_CHR *)"**********", 10)	0~2	1	0	1	0	2
	stp_zipu_num->ul_sub_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

FlowFactDisplay_Set_FactMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	switch(stp_zipu_num->ul_mode_select_num)		○	-	-	-	-	-	-	-
	case F_GAS_COMB_NUM		○	-	-	-	-	-	-	-
	case F_DATE_SET_NUM		-	○	-	-	-	-	-	-
	case F_ROMSUM_NUM		-	-	○	-	-	-	-	-
	case F_A_D_DISP_NUM		-	-	-	○	-	-	-	-
	case F_SHIP_RECORD_NUM		-	-	-	-	○	-	-	-
	case F_DEFAULT_NUM		-	-	-	-	-	○	-	-
	case F_END_NUM		-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000003	0x00000004	0x00000006	0x00000007	0x00000008	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0~64	16	0	0	0	8	8	64	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowFactDisplay_Set_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_DefaultConfirmation

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_DefaultNG

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x40	0x40
Judgment			OK

FlowGasCalDisplay_Disp_DefaultOK

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x40	0x40
Judgment			OK

FlowGasCalDisplay_Disp_SetDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x40	0x40
Judgment			OK

FlowFactDisplay_Set_ADDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgement	Ikarashi	

nblock-loop	None
NaN-butten-pass	42
Judgement	OK

				case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17
Root			switch(stop_disp_num->ut_item_select_num)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
			case 0:	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 1:	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 2:	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 3:	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN000) < 0)	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN003) < 0)	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 4:	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN004) < 0)	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 5:	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN005) < 0)	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(Ad12bit_Get_Ad_Data(AD1260T_AN006) < 0)	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 6:	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0][0][0] < 0)	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(stp_status->if(_absolute_humidity[0] < 0)	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 7:	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0][0][1] < 0)	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 8:	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0][0][2] < 0)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
			case 9:	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0][1][0] < 0)	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(stp_status->if(_absolute_humidity[1] < 0)	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 10:	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0][1][1] < 0)	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 11:	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0][1][2] < 0)	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 12:	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[0] < 0)	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 13:	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
			if(stp_sub_ad->if(_inc_ad_data[1] < 0)	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 14:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
			if(stp_sub_ad->if(_inc_ad_data[2] < 0)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			case 15:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
			if(stp_sub_ad->if(_temp_ad_data < 0)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			if(stp_sub_ad->if(_temp_data < 0)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
			else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			default:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
			if(stp_disp_num!=ut_item_select_num >= 0)||(stp_disp_num!=ut_item_select_num <= 14))	-	-	-	-	-	-	○	○	○	○	○	○	-	-	-	-	-
			else	○	○	○	○	○	○	-	-	-	-	-	-	○	○	○	○	○
	Symbol	Range																		
Input	Ad12bit_Get_Ad_Data()	0~0		-1	-1	-1	-1	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1
	if_inc_ad_data[0][0][0]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
	if_inc_ad_data[0][0][1]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1
	if_inc_ad_data[0][0][2]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1
	if_inc_ad_data[0][1][0]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
	if_inc_ad_data[0][1][1]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1
	if_inc_ad_data[0][1][2]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	if_inc_ad_data[2]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	if_inc_ad_data[1]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	if_inc_ad_data[3]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	if_temp_ad_data	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	if_temp_data	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	ut_item_select_num	0~66		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	if_absolute_humidity[0]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	if_absolute_humidity[1]	-1~0		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	stn_bcd_max[0][0]	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	stn_bcd_max[1][0]	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	stn_bcd_min[0]	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	stn_bcd_min[1][0]	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Test_Hex_Ave[0]	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Test_Hex_Ave[1]	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Output	stn_bcd_max[0][0]	0~65		65	65	65	65	65	65	65	65	0	65	65	65	65	65	65	65	0
	stn_bcd_min_dot	0,8		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	0
	stn_bcd_com_dot	64,66		64	64	64	64	64	64	66	66	66	66	66	66	64	64	64	64	64
Judgment				OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

case18	case19	case20	case21	case22	case23	case24	case25	case26	case27	case28	case29	case30	case31	case32	case33	case34	case35	case36	case37	case38	case39	case40	case41	case42
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	○	-	-	○
-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	○
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	○	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	○	○	○	○	○	○	○	○	-	-	-	-	-	○	-	○	-	○	○
○	○	○	○	○	○	-	-	-	-	-	-	-	-	○	○	○	○	○	-	○	-	○	-	-
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	-1	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	-1	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1
0	1	2	3	4	5	6	6	7	8	9	9	10	11	12	13	14	15	15	6	5	11	12	8	11
-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	-1	0
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	45	0	0	0	45	0	0	0	0	0	0	0	0	45	45	45
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
64	64	64	64	64	64	66	66	66	66	66	66	66	66	64	64	64	64	64	66	64	66	64	66	66
OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_gascal_display.c

FlowGasCalDisplay_Chk_AutoCalMenuGasOn

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((stp_GasInfo_addr+luc_i)->uc_cal_group == puc_num))		○	○	-
	if(puc_flg == M_OFF || (stp_GasInfo_addr+luc_i)->ul_cal_denst != M_OFF_VAL)		○	-	-
	else		-	○	-
	else		-	-	○
	else		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	0	1	2
	puc_num	0x00,0xFF	0x00	0xFF	0x00
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	0
	(stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0x00	0xFF	0xFF
	(stp_GasInfo_addr+luc_i)->ul_cal_denst	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowGasCalDisplay_Chk_AutoCalMenuZIPC

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	FlowGasCalDisplay_Chk_AutoCalMenuGasOn (M_ON, stp_zipu_num->ul_sub2_mode_select_num)	0,1	0	1	2
Output	luc_ans	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasCalDisplay_Chk_AutoCalSelectMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	ul_sub2_mode_select_num	0~65535	0	65535
Output	ul_sub2_mode_select_num	0~255	0	255
Judgment			OK	OK

FlowGasCalDisplay_Chk_HCModeWarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(m_Chk_AutoCalGasOn(0, M_ON, GasFuncAutoCal_Get_AutoCalGroup()) == M_OK)		○	○	○	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (stp_GasInfo_addr->uc_gas_comb <= 0x02))		○	○	-	-	-
	if(stp_data->uc_nc_mode_flg == GAS_CALC_NC_MODE_STINT)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	Symbol	Range					
Input	m_Chk_AutoCalGasOn(0, M_ON, GasFuncAutoCal_Get_AutoCalGroup())	0~2	0	0	0	1	2
	stp_GasInfo_addr->uc_gas_comb	0~30	1	1	0	29	30
	stp_data->uc_nc_mode_flg	0~2	1	0	0	0	2
Output	luc_ans	0,1	1	0	0	0	0
Judgment			OK	OK	OK	OK	OK

FlowGasCalDisplay_Disp_AirCalDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_AirCalPass

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_AirFail

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_ZRO)		○	○	-	-
	switch(luc_i)		○	○	-	-
	case 0		○	○	-	-
	case 1		○	○	-	-
	case 2		○	○	-	-
	case 3		○	○	-	-
	default		○	○	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	30	0
	(stp_denst + luc_i)->uc_state_trb	0x00~0xFF	0xFF	0xFF	0xFF	0
	FlowMain_Get_ModeStatus()	0~7	1	0	7	8
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowGasCalDisplay_Disp_AirHold

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_AirRelease

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_AutoCal_ChgDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_AutoCal_SelectGas

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_item_select_num < HDDEF_GAS_MAX)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	None	-	-	-
Judgment			OK	OK

FlowGasCalDisplay_Disp_AutoCalBlinkDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○
	if(m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup()) == M_OK)		○	-	-
	else		-	○	○
	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○
	case 0		○	-	-
	case 1		-	○	-
	default		-	-	○
	Symbol	Range			
Input	m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup())	0~2	0	1	2
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowGasCalDisplay_Disp_AutoCalDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	11
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11
Root	if(stp_zipu_num->ul_sub2_mode_select_num == M_AUTO_GROUP_MAX+2)		○	-	-	-	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_sub2_mode_select_num == M_AUTO_GROUP_MAX+1)		-	○	-	-	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_sub2_mode_select_num == M_AUTO_GROUP_MAX)		-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	○	○	○	○	○	○	○	○
	for(luc_i=0; luc_i<5; luc_i++)		-	-	-	○	○	○	○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	-	-	○	○	-	-	-	-	-	-
	if(stp_zipu_num->ul_sub2_mode_select_num == (stp_GasInfo_addr+luc_i)->uc_cal_group)		-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	○	○	○	○	○	○
	switch(ZIPCNumSet_Get_DispFlicker2())		-	-	-	○	○	○	○	○	○	○	○
	case 0		-	-	-	○	-	-	-	-	-	-	-
	case 1		-	-	-	-	○	○	○	○	○	○	-
	switch(stp_zipu_num->ul_sub2_mode_select_num)		-	-	-	-	○	○	○	○	○	○	-
	case DataStructConst_CAL_GROUP_A		-	-	-	-	○	-	-	-	-	-	-
	case DataStructConst_CAL_GROUP_B		-	-	-	-	-	○	-	-	-	-	-
	case DataStructConst_CAL_GROUP_C		-	-	-	-	-	-	○	-	-	-	-
	case DataStructConst_CAL_GROUP_D		-	-	-	-	-	-	-	○	-	-	-
	case DataStructConst_CAL_GROUP_E		-	-	-	-	-	-	-	-	○	-	-
	default		-	-	-	-	-	-	-	-	-	○	-
	default		○	○	○	-	-	-	-	-	-	-	○
	Symbol	Range											
Input	stp_zipu_num->ul_sub2_mode_select_num	0x00000000~0xFFFFFFFF	0x00000007	0x00000006	0x00000005	0x00000000	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF	0xFFFFFFFF
	(stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	30	0	0	29	29	0	0	0	0	0	0
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	0	0	0	1	1	1	1	1	1	2
Output	None	-	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowGasCalDisplay_Disp_AutoCalPass

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_AutoCalRunning

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_EndAutoCalDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○
	if(m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup()) == M_OK)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup())	0,1	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowGasCalDisplay_Disp_FailAutoCal

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if(m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup()) == M_OK)		○	○	-	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_SPN)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup())	0,1	0	0	1	2
	(stp_denst + luc_i)->uc_state_trb	0x00~0xFF	0xFF	0x00	0xFF	0xFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowGasCalDisplay_Disp_GroupSelectGas

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_item_select_num < HDDEF_GAS_MAX)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	None	-	-	-
Judgment			OK	OK

FlowGasCalDisplay_Disp_GroupSelectGroup

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(stp_zipu_num->ul_sub_item_select_num)		○	○	○	○	○	○
	case DataStructConst_CAL_GROUP_A		○	-	-	-	-	-
	case DataStructConst_CAL_GROUP_B		-	○	-	-	-	-
	case DataStructConst_CAL_GROUP_C		-	-	○	-	-	-
	case DataStructConst_CAL_GROUP_D		-	-	-	○	-	-
	case DataStructConst_CAL_GROUP_E		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

FlowGasCalDisplay_Disp_HCModeWarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Disp_SpanMargin

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if(m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get _AutoCalGroup()) == M_OK)		○	○	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr + luc_i)-> uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup())	0~2	0	0	1	2
	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	0	3	29	30
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowGasCalDisplay_End_AutoCal_ChgDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	(stp_GasInfo_addr+(stp_zipu_num-> ul_item_select_num))->ul_cal_denst	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_zipu_num->uc_sub_item_select_updown	0	0	0
Judgment			OK	OK

FlowGasCalDisplay_End_GroupSelectGroup

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	(stp_GasInfo_addr+(stp_zipu_num-> ul_item_select_num))->uc_cal_group	0x00,0xFF	0x00	0xFF
	stp_zipu_num->uc_sub_item_select_updown	0	0	0
Judgment			OK	OK

FlowGasCalDisplay_Execution_AutoCal

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Get_SpanMarginOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_data->uc_margin_disp_onoff == M_ON)		○	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	-	-
	if(m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_ Get_AutoCalGroup()) == M_OK)		○	○	-	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr + luc_i)-> uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	(stp_data->uc_margin_disp_onoff	0,1	1	1	1	0	2
	m_Chk_AutoCalGasOn(luc_i, M_ON, GasFuncAutoCal_Get_AutoCalGroup())	0,1	0	0	1	0	2
	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~29	0	3	29	0	30
Output	luc_ans	0,1	1	1	1	0	0
Judgment			OK	OK	OK	OK	OK

FlowGasCalDisplay_Init_AutoCal_ChgDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	(stp_GasInfo_addr+(stp_zipu_num-> ui_item_select_num))->ul_cal_denst	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_zipu_num->uc_sub_item_select_updown	0	0	0
Judgment			OK	OK

FlowGasCalDisplay_Init_AutoCal_SelectGas

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	30
Output	stp_zipu_num->ul_item_select_num	0,1	1	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowGasCalDisplay_Init_AutoCalBlinkDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasCalDisplay_Init_AutoCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<M_AUTO_GROUP_MAX; luc_i++)		○	○	○
	for(luc_j=0; luc_j<HDDEF_GAS_MAX; luc_j++)		○	○	○
	if((stp_GasInfo_addr+luc_j)->uc_gas_comb != 0x00)		○	○	-
	if(luc_i == (stp_GasInfo_addr+luc_j)->uc_cal_group)		○	-	-
	else		-	○	-
	else		-	-	○
	if(luc_ans == M_ON)		○	-	-
	else		-	○	-
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_j)->uc_gas_comb	0~29	29	30	0
	(stp_GasInfo_addr+luc_j)->uc_cal_group	0x00~0xFF	0x00	0xFF	0xFF
Output	stp_zipu_num->ul_sub2_mode_select_num	0~5	0	0	0
	stp_zipu_num->uc_sub2_mode_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowGasCalDisplay_Init_GasCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
Judgment			OK

FlowGasCalDisplay_Init_GroupSelectGas

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	30
Output	stp_zipu_num->ul_item_select_num	0,1	1	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowGasCalDisplay_Init_GroupSelectGroup

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_cal_group	0x00,0xFF	0x00	0xFF
Output	stp_zipu_num->ul_sub_item_select_num	0x00,0xFF	0x00	0xFF
	stp_zipu_num->uc_sub_item_select_updown	0	0	0
Judgment			OK	OK

FlowGasCalDisplay_Init_LimitGasCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ul_sub_mode_select_num	2	2
	ul_sub_mode_select_updown	0	0
Judgment			OK

FlowGasCalDisplay_SelectUp_AutoCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_i > (M_AUTO_GROUP_MAX + 3))		○	-	○
	for(luc_i; luc_i < M_AUTO_GROUP_MAX; luc_i++)		○	○	○
	if(FlowGasCalDisplay_Chk_AutoCalMenuGasOn(M_OFF, luc_i) == M_OK)		○	-	○
	Symbol	Range			
Input	FlowGasCalDisplay_Chk_AutoCalMenuGasOn()	0,1	0	0	1
	ul_sub2_mode_select_num	7,8	8	7	8
Output	ul_sub2_mode_select_num	0~8	0	8	5
Judgment			OK	OK	OK

FlowGasCalDisplay_Select_AutoCal_ChgDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(lul_digit == 100)		○	○	-	-	-
	if((lul_max - lul_min) / M_MAX_CHG_DIGIT > (lul_gas_digit * 10))		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	Symbol	Range					
Input	lul_digit	1~501	100	100	1	500	501
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_cal_denst_max	0~100001	100000	0	0	0	100001
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_cal_denst_min	0~501	500	0	0	0	501
	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->ul_digit	0~501	1	500	1	1	501
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

FlowGasCalDisplay_Select_AutoCal_SelectGas

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	○	-	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		○	-	-	-	-	-	-	-
	else		-	○	○	-	-	-	-	-
	for(luc_i; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	-	-	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	-	-	-	-	-	-
	else		○	○	-	-	-	-	-	-
	else		-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == 0)		-	-	-	-	-	○	-	-
	else		-	-	-	○	○	-	○	○
	for(luc_i=(stp_zipu_num->ul_item_select_num-1); luc_i > 0; luc_i--)		-	-	-	○	○	-	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	-	-	-	○	-	○	-
	else		-	-	-	○	-	-	-	○
	if((luc_i == 0) && ((stp_GasInfo_addr+0)->uc_gas_comb == 0x00))		-	-	-	○	-	-	-	○
	else		-	-	-	-	○	-	○	-
	Symbol	Range								
Input	stp_zipu_num->uc_item_select_updown	0,1	0	0	0	1	1	1	2	1
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~29	29	0	29	0	29	0	30	29
	(stp_GasInfo_addr+0)->uc_gas_comb	0~29	29	0	29	0	29	0	30	0
Output	None	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowGasCalDisplay_Select_AutoCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if(stp_zipu_num->uc_sub2_mode_select_updown == M_OFF)		○	○	○	○	-	-	-	-	-	-
	if(stp_zipu_num->ul_sub2_mode_select_num == M_AUTO_GROUP_MAX)		○	-	-	-	-	-	-	-	-	-
	else if(stp_zipu_num->ul_sub2_mode_select_num == M_AUTO_GROUP_MAX+1)		-	○	-	-	-	-	-	-	-	-
	else		-	-	○	○	-	-	-	-	-	-
	if(stp_zipu_num->ul_sub2_mode_select_num == (M_AUTO_GROUP_MAX+2))		-	-	○	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-
	for(luc_i; luc_i < M_AUTO_GROUP_MAX; luc_i++)		-	-	○	○	-	-	-	-	-	-
	if(FlowGasCalDisplay_Chk_AutoCalMenuGasOn(M_OFF, luc_i) == M_OK)		-	-	○	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	○	○	○
	if(stp_zipu_num->ul_sub2_mode_select_num == 0)		-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	○	○	○	○	-
	if(stp_zipu_num->ul_sub2_mode_select_num == (M_AUTO_GROUP_MAX+2))		-	-	-	-	-	○	-	-	-	-
	else if(stp_zipu_num->ul_sub2_mode_select_num == (M_AUTO_GROUP_MAX+1))		-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	○	○	○
	for(luc_i; luc_i > 0; luc_i--)		-	-	-	-	-	-	-	○	○	○
	if(FlowGasCalDisplay_Chk_AutoCalMenuGasOn(M_OFF, luc_i) == M_OK)		-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	○	○
	if((luc_i == 0) && (FlowGasCalDisplay_Chk_ AutoCalMenuGasOn(M_OFF, luc_i) == M_NG))		-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	○	-	○
	Symbol	Range										
Input	stp_zipu_num->uc_sub2_mode_select_updown	0~2	0	0	0	0	1	1	1	1	1	2
	stp_zipu_num->ul_sub2_mode_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000006	0x00000007	0xFFFFFFFF	0x00000000	0x00000007	0x00000006	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	FlowGasCalDisplay_Chk_ AutoCalMenuGasOn(M_OFF, luc_i)	0~2	0	0	0	1	1	0	1	0	1	2
Output	None	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowGasCalDisplay_Select_GasCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	○	○	○	-	-	-	-	-
	for(luc_i; luc_i<U_G_MAX_NUM; luc_i++)		○	○	○	-	-	-	-	-	-
	if(uc_gascal_mode_item[luc_i] == M_ON)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	if(luc_i >= U_G_MAX_NUM)		-	○	○	○	-	-	-	-	-
	for(luc_i=0; luc_i<U_G_MAX_NUM; luc_i++)		-	○	-	○	-	-	-	-	-
	if(uc_gascal_mode_item[luc_i] == M_ON)		-	-	-	○	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_sub_mode_select_num != 0)		-	-	-	-	○	○	○	-	○
	for(luc_i; luc_i>0; luc_i--)		-	-	-	-	-	-	-	○	-
	if(uc_gascal_mode_item[luc_i] == M_ON)		-	-	-	-	○	○	○	○	○
	else		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	○
	if((luc_i == 0) && (uc_gascal_mode_item[0] != M_ON))		-	-	-	-	○	○	-	○	○
	for(luc_i=U_G_END_NUM; luc_i>0; luc_i--)		-	-	-	-	○	○	-	○	○
	if(uc_gascal_mode_item[luc_i] == M_ON)		-	-	-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-	○	○
	else		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	stp_zipu_num->uc_sub_mode_select_updown	0,1	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x0000001E	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	uc_gascal_mode_item[luc_i]	0,1	1	0	1	1	1	0	1	0	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowGasCalDisplay_Select_GroupSelectGas

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	○	-	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		○	-	-	-	-	-	-	-
	else		-	○	○	-	-	-	-	-
	for(luc_i; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	-	-	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	-	-	-	-	-	-
	else		○	○	-	-	-	-	-	-
	else		-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == 0)		-	-	-	-	-	○	-	-
	else		-	-	-	○	○	-	○	○
	for(luc_i=(stp_zipu_num->ul_item_select_num-1); luc_i > 0; luc_i--)		-	-	-	○	○	-	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	-	-	-	○	-	○	-
	else		-	-	-	○	-	-	-	○
	if((luc_i == 0) && ((stp_GasInfo_addr+0)->uc_gas_comb == 0x00))		-	-	-	○	-	-	-	○
	else		-	-	-	-	○	-	○	-
	Symbol	Range								
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	0	1	1	1	2	1
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	0	29	0	29	0	30	29
	(stp_GasInfo_addr+0)->uc_gas_comb	0~30	29	0	29	0	29	0	30	0
Output	None	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowGasCalDisplay_Select_GroupSelectGroup

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount(stp_zipu_num->ul_sub_item_ select_num, 1, M_AUTO_GROUP_MAX-1, 0, stp_zipu_num->uc_sub_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowGasCalDisplay_Set_GasCalMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(stp_zipu_num->ul_sub_mode_select_num)		○	○	○	○	○
	case U_G_AIR_NUM		○	-	-	-	-
	case U_G_N2CAL_NUM		-	○	-	-	-
	case U_G_AUTOCAL_NUM		-	-	○	-	-
	case U_G_END_NUM		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000004	0xFFFFFFFF
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_Chk_AutoCalGasOn

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case2
Root	if(((stp_GasInfo_addr+puc_gas)->uc_gas_comb != 0x00) && ((stp_GasInfo_addr+puc_gas)->uc_cal_group == puc_num))		○	○	-
	if(puc_flg == M_OFF || (stp_GasInfo_addr+puc_gas)->ul_cal_denst != M_OFF_VAL)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_gas	0~30	29	30	0
	puc_flg	0~2	0	1	2
	puc_num	0x00,0xFF	0x00	0xFF	0x00
	(stp_GasInfo_addr+puc_gas)->uc_gas_comb	0~30	29	30	0
	(stp_GasInfo_addr+puc_gas)->uc_cal_group	0x00,0xFF	0x00	0xFF	0x00
	(stp_GasInfo_addr+puc_gas)->ul_cal_denst	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xFFFFFFFF
Output	luc_ans	0,1	1	1	1
Judgment			OK	OK	OK

m_Chk_GasCalMenuItem

	Responsible	Last update
Coding	Ikarashi	2021/6/18
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	if((stp_GasInfo_addr+4)->uc_gas_comb != DataStructConst_IR_GAS_NUM)		○
	else		-
	Symbol	Range	
Input	None	-	-
Output	uc_gascal_mode_item[U_G_N2CAL_NUM]	0	0
Judgment		-	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_gassel_display.c

FlowGasSelDisplay_Disp_ChangeLELValue

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_blink	0x7F	0x7F	0x7F	0x7F	0x7F
Judgment			OK	OK	OK	OK

FlowGasSelDisplay_Disp_MarginDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasSelDisplay_Disp_ResetAlarmPoint

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowGasSelDisplay_Disp_StealthMotorOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasSelDisplay_Disp_StealthOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowGasSelDisplay_End_ChangeLELValue

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	stp_GasInfo_addr->uc_gas_comb	0~30	0	29	30
Output	stp_data->uc_lel_sel_flg	0x00,0xFF	0x00	0xFF	0xFF
	stp_GasInfo_addr->fl_lel_ratio[0]	0,5	0	0	5
Judgment			OK	OK	OK

FlowGasSelDisplay_End_MarginDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_margin_disp_onoff	0x00~0xFF	0x00	0xFF
Judgment			OK	OK

FlowGasSelDisplay_End_ResetAlarmPoint

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○
	if(luc_ret == M_NG)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	DataFram_AlarmReset_Write()	0~2	1	0	2
	(stp_GasInfo_addr+luc_i)->ul_full_scale	0~100001	0	100000	100001
	(stp_GasInfo_addr+luc_i)->ul_digit	1~501	1	500	501
	(stp_GasInfo_addr+luc_i)->uc_uni_dec	0x00~0x23	0x00	0x22	0x23
	(stp_GasInfo_addr+luc_i)->ul_1st_alarm	0~1001	0	1000	1001
	(stp_GasInfo_addr+luc_i)->ul_2nd_alarm	0~25001	0	25000	25001
	(stp_GasInfo_addr+luc_i)->ul_3rd_alarm	0~50001	0	50000	50001
	(stp_GasInfo_addr+luc_i)->ul_stel_alarm	0~5001	0	5000	5001
	(stp_GasInfo_addr+luc_i)->ul_twa_alarm	0~2501	0	2500	2501
Output	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_full_scale	0~100001	0	100000	100001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_digit	0~501	1	500	501
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].uc_uni_dec	0x00~0x23	0x00	0x22	0x23
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_1st_alarm	0~1001	0	1000	1001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_2nd_alarm	0~25001	0	25000	25001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_3rd_alarm	0~50001	0	50000	50001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_stel_alarm	0~5001	0	5000	5001
	stp_AlarmReset_addr->ALARM_POINT. RESET_ALARM[luc_i].ul_twa_alarm	0~2501	0	2500	2501
	luc_ret	0~2	1	0	2
Judgment			OK	OK	OK

FlowGasSelDisplay_End_StealthOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_stealth_on	0x00,0xFF	0x00	0xFF
	stp_data->uc_stealth_motor_on	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowGasSelDisplay_Init_ChangeLELValue

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_lel_sel_flg	0~3	0	2	3
Output	stp_zipu_num->ul_item_select_num	0~3	0	2	3
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowGasSelDisplay_Init_GasSelMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_mode_select_num	0	0
	stp_zipu_num->uc_mode_select_updown	0	0
Judgment			OK

FlowGasSelDisplay_Init_MarginDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_margin_disp_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasSelDisplay_Init_StealthMotorOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_stealth_motor_on	0~2	0	1	2
Output	stp_zipu_num->ul_sub_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasSelDisplay_Init_StealthOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_stealth_on	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasSelDisplay_Select_ChangeLELValue

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, 2, 0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowGasSelDisplay_Select_GasSelMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_mode_select_updown == M_OFF)		○	○	○	○	-	-	-	-	-
	for(luc_i; luc_i<G_MAX_NUM; luc_i++)		○	○	○	-	-	-	-	-	-
	if(uc_gassel_mode_item[luc_i] == M_ON)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	if(luc_i >= G_MAX_NUM)		-	○	○	○	-	-	-	-	-
	for(luc_i=0; luc_i<G_MAX_NUM; luc_i++)		-	○	-	○	-	-	-	-	-
	if(uc_gassel_mode_item[luc_i] == M_ON)		-	-	-	○	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_mode_select_num != 0)		-	-	-	-	○	○	○	-	○
	for(luc_i; luc_i>0; luc_i--)		-	-	-	-	-	-	-	○	-
	if(uc_gassel_mode_item[luc_i] == M_ON)		-	-	-	-	○	○	○	○	○
	else		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	○
	if((luc_i == 0) && (uc_gassel_mode_item[0] != M_ON))		-	-	-	-	○	○	-	○	○
	for(luc_i=G_END_NUM; luc_i>0; luc_i--)		-	-	-	-	○	○	-	○	○
	if(uc_gassel_mode_item[luc_i] == M_ON)		-	-	-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-	○	○
	else		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	stp_zipu_num->uc_mode_select_updown	0~2	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x0000001E	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	uc_gassel_mode_item[luc_i]	0~2	1	0	1	1	1	0	1	0	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowGasSelDisplay_Select_MarginDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0,1	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasSelDisplay_Select_StealthMotorOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_sub_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_sub_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasSelDisplay_Select_StealthOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowGasSelDisplay_Set_GasSelMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch(stp_zipu_num->ul_mode_select_num)		○	○	○	○	○	○	○
	case G_GAS_COMB_NUM		○	-	-	-	-	-	-
	case G_SET_RESET_AL_P_NUM		-	○	-	-	-	-	-
	case G_SPARE_ENERGY_NUM		-	-	○	-	-	-	-
	case G_STEALTH_NUM		-	-	-	○	-	-	-
	case G_LEL_CHG_NUM		-	-	-	-	○	-	-
	case G_END_NUM		-	-	-	-	-	○	-
	default		-	-	-	-	-	-	○
	Symbol	Range							
Input	stp_zipu_num->ul_mode_select_num	0~6	0	1	2	3	4	5	6
Output	None	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_initial_display.c

FlowInitialDisplay_Calc_InitDispTime

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(GasCalc_Get_ToxicGasON() == M_ON)		○	○	○	○	-	-
	if(GasCalc_Get_Alarm1hON() == M_ON)		○	○	-	-	-	-
	if(GasCalc_Get_AlarmTwaON() == M_ON)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	○	-	-
	if(GasCalc_Get_AlarmTwaON() == M_ON)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		-	-	-	-	○	○
	if(FuncMainteNotice_Get_MainteNoticeOnOff() == M_ON)		○	○	-	-	-	-
	if(FuncMainteNotice_Get_MainteNoticeLimitExpired() == M_OFF)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	○	○	○
	if(FuncCalLimit_Get_CalLimitOnOff() == M_ON)		○	○	-	-	-	-
	if(FuncCalLimit_Get_ChkCalLimitExpired() == M_OFF)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	○	○	○
	if(stp_data->uc_bump_lmt_set == M_ON)		○	○	-	-	-	-
	if(FuncBumpLimit_Get_ChkBumpLimitExpired() == M_OFF)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	○	○	○
	if(stp_data->uc_id_disp_onoff == M_ON)		○	-	-	-	-	-
	else		-	○	○	○	○	○
	Symbol	Range						
Input	GasCalc_Get_ToxicGasON()	0~2	1	1	1	1	0	2
	GasCalc_Get_Alarm1hON()	0~2	1	1	0	0	0	2
	GasCalc_Get_AlarmTwaON()	0~2	1	0	1	0	0	2
	FuncMainteNotice_Get_MainteNoticeOnOff()	0~2	1	1	0	0	0	2
	FuncMainteNotice_Get_MainteNoticeLimitExpired()	0~2	0	1	0	0	0	2
	FuncCalLimit_Get_CalLimitOnOff()	0~2	1	1	0	0	0	2
	FuncCalLimit_Get_ChkCalLimitExpired()	0~2	0	1	0	0	0	2
	stp_data->uc_bump_lmt_set	0~2	1	1	0	0	0	2
	FuncBumpLimit_Get_ChkBumpLimitExpired()	0~2	0	1	0	0	0	2
	stp_data->uc_id_disp_onoff	0~2	1	0	0	0	0	2
Output	uss_init_disp_time[FlowInitialDisplay_DATE_NUMBER]	400	400	400	400	400	400	400
	uss_init_disp_time[FlowInitialDisplay_BATT_NUMBER]	300,400	300	400	400	400	400	400
	uss_init_disp_time[FlowInitialDisplay_NAME_NUMBER]	250,350	250	350	350	350	350	350
	uss_init_disp_time[FlowInitialDisplay_FS_NUMBER]	200~1200	200	600	400	800	1200	1200
	uss_init_disp_time[FlowInitialDisplay_TWA_NUMBER]	0~400	300	0	400	0	0	0
	uss_init_disp_time[FlowInitialDisplay_STEL_NUMBER]	0~400	300	300	400	400	0	0
	uss_init_disp_time[FlowInitialDisplay_1H_NUMBER]	0,300	300	300	0	0	0	0
	uss_init_disp_time[FlowInitialDisplay_1ST_NUMBER]	200,400	200	400	400	400	400	400
	uss_init_disp_time[FlowInitialDisplay_2ND_NUMBER]	200,400	200	400	400	400	400	400
	uss_init_disp_time[FlowInitialDisplay_3RD_NUMBER]	200,400	200	400	400	400	400	400
	uss_init_disp_time[FlowInitialDisplay_MAINTE_NUMBER]	0,300	300	300	0	0	0	0
	uss_init_disp_time[FlowInitialDisplay_CAL_NUMBER]	0,300	300	300	0	0	0	0
	uss_init_disp_time[FlowInitialDisplay_U_ID_NUMBER]	0,150	150	0	0	0	0	0
	uss_init_disp_time[FlowInitialDisplay_S_ID_NUMBER]	0,150	150	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

FlowInitialDisplay_Chk_HCModeHHCflg

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_data->uc_nc_mode_flg == GAS_CALC_NC_MODE_STINT)		○	○	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (stp_GasInfo_addr->uc_gas_comb <= 0x02))		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_data->uc_nc_mode_flg	0~2	1	1	0	2
	stp_GasInfo_addr->uc_gas_comb	0~30	1	0	29	30
Output	luc_ans	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

FlowInitialDisplay_Disp_BumpLimitEffe

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_day >= FuncBumpLimit_INIT_BUMP_LIMIT_DAY)		○	-	○
	else		-	○	-
	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○
	case 0		○	-	-
	case 1		-	○	-
	default		-	-	○
	Symbol	Range			
Input	FuncBumpLimit_Chk_BumpLimitDay()	0~1001	1000	0	1001
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_BumpLimitOk

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_day >= FuncBumpLimit_INIT_BUMP_LIMIT_DAY)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	FuncBumpLimit_Chk_BumpLimitDay()	0~1001	1000	0	1001
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_BumpLimitOper

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○
	case 0		○	-	-
	case 1		-	○	-
	default		-	-	○
	Symbol	Range			
Input	ZIPCNumSet_Get_DispFlicker2()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_BumpLimitRequ

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_Disp_CalLimitEffe

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_day >= FuncCalLimit_INIT_CAL_LIMIT_DAY)		○	-	○
	else		-	○	-
	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○
	case 0		○	-	-
	case 1		-	○	-
	default		-	-	○
	Symbol	Range			
Input	FuncCalLimit_Chk_CalLimitDay()	0~1001	1000	0	1001
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_CalLimitOk

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_day >= FuncCalLimit_INIT_CAL_LIMIT_DAY)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	FuncCalLimit_Chk_CalLimitDay()	0~1001	1000	0	1001
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_CalLimitOper

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(ZIPCNumSet_Get_DispFlicker2())		○	○	○
	case 0		○	-	-
	case 1		-	○	-
	default		-	-	○
	Symbol	Range			
Input	ZIPCNumSet_Get_DispFlicker2()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_CalLimitRequ

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_Disp_MainteNoticeEffe

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_day >= FuncMainteNotice_INIT_MAINTE_NOTICE_DAY)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	FuncMainteNotice_Chk_MainteLimitDay()	0~1001	1000	0	1001
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_MainteNoticeOk

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_day >= FuncMainteNotice_INIT_MAINTE_NOTICE_DAY)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	FuncMainteNotice_Chk_MainteLimitDay()	0~1001	1000	0	1001
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_Disp_MainteNoticeOper

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Disp_MainteNoticeRequ

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Disp_PleaseWait

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Disp_ResumeConfirmation

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_Disp_ResumeFail

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Disp_ResumeOFF

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Disp_ResumeON

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_DispSet_1hAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_GasInfo_addr+3)->uc_gas_comb != 0x00)		○	○	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	(stp_GasInfo_addr+3)->uc_gas_comb	0~30	29	30	0
	stp_data->uc_alarm_on	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowInitialDisplay_DispSet_1stAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	3	0
	stp_data->uc_alarm_on	0~2	0	1	0	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_2ndAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	3	0
	stp_data->uc_alarm_on	0~2	0	1	0	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_3rdAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	3	0
	stp_data->uc_alarm_on	0~2	0	1	0	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_AutoZeroConfirm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_Battery

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_data->uc_alarm_pattern)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_data->uc_alarm_pattern	0,~2	0	1	2
Output	stp_nl_disp->uc_lcd_mes_dot[1]	2	2	2	2
Judgment			OK	OK	OK

FlowInitialDisplay_DispSet_DateAndTime

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_dot[1]	0x05	0x05
Judgment			OK

FlowInitialDisplay_DispSet_FS

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	-	○
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	○
	else		-	○	-	-
	else		-	-	○	-
	if(stp_data->uc_lel_sel_flg == 1)		○	-	-	-
	else if(stp_data->uc_lel_sel_flg == 2)		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	3	0	30
	tp_data->uc_lel_sel_flg	0~3	1	2	0	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_GasName

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	3	0
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_SensorInitFail

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(stp_status->us_main_state & DataStructConst_FAIL_SUB)		○	-	-	-	○
	else		-	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	○	○	○	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_INS)		-	○	○	-	-
	if((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		-	○	-	-	-
	switch(luc_i)		-	○	-	-	-
	case 0		-	○	-	-	-
	case 1		-	○	-	-	-
	case 2		-	○	-	-	-
	case 3		-	○	-	-	-
	default		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	Symbol	Range					
Input	stp_status->us_main_state	0x0000~0xFFFF	0x0200	0x0000	0x0000	0x0000	0xFFFF
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x10	0x10	0x00	0xFF
	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	29	0	3	0	30
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0~255	0	255	255	0	0
	stp_nl_disp->uc_lcd_mes_blink[1]	0~255	0	255	240	0	0
Judgment			OK	OK	OK	OK	OK

FlowInitialDisplay_DispSet_StationId

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_DispSet_STELAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	3	0
	stp_data->uc_alarm_on	0~2	0	1	0	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowInitialDisplay_DispSet_TWAAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	○	○	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	○	-	-
	if(!((luc_i == DataStructConst_CO_CH) && (stp_data->uc_1h_alarm_on == M_ON)))		○	○	-	-	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	30	3	0
	stp_data->uc_1h_alarm_on	0~2	1	1	0	0	2
	stp_data->uc_alarm_on	0~2	0	1	1	0	2
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

FlowInitialDisplay_DispSet_UserId

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Get_InitDispTime

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case3	case4
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~16	0	15	16
Output	uss_init_disp_time[puc_num]	0	0	0	0
Judgment			OK	OK	OK

FlowInitialDisplay_Set_AllOffDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Set_AllOnDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowInitialDisplay_Set_HCModeHHCflgGasName

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((HCTableDefData[stp_GasInfo_addr->uc_sub_comb].uc_meas_group & 0x10) == 0x10)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	stp_GasInfo_addr->uc_gas_comb	0~30	4	0	29	30
	stp_GasInfo_addr->uc_sub_comb	0~30	4	0	29	30
Output	stp_GasInfo_addr->uc_sub_comb	0~30	4	0	29	30
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_main.c

FlowMain_Disp_FailDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification paths	18
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18
Root	if(++lucs_count >= 4U)		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(++lucs_flg >= 7U)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(stp_status->us_main_state & DataStructConst_SYS_FAIL_PRIORITY_0)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	else if(stp_status->us_main_state & DataStructConst_SYS_FAIL_PRIORITY_1)		-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(stp_status->us_main_state & DataStructConst_SYS_FAIL_PRIORITY_2)		-	-	-	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	else if(stp_status->us_main_state & DataStructConst_SYS_FAIL_PRIORITY_3)		-	-	-	-	-	-	-	○	○	○	○	○	○	-	-	-	-	-
	if((stp_denst+0)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+1)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+2)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+3)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+4)->uc_state_trb & DataStructConst_S_FAIL_SEN)		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	○	○	○	○	-	-	-	-	-
	else if((stp_denst+0)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+1)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+2)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+3)->uc_state_trb & DataStructConst_S_FAIL_SEN | (stp_denst+4)->uc_state_trb & DataStructConst_S_FAIL_SEN)		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	else if(stp_status->us_main_state & DataStructConst_SYS_FAIL_PRIORITY_4)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(stp_status->us_main_state & DataStructConst_SYS_FAIL_PRIORITY_5)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	for(luc_i=lucs_flg; luc_i<7U; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(lul_disp_num[luc_i] != 0)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	else		-	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(luc_i >= 7U)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	for(luc_i=0U; luc_i<7U; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(lul_disp_num[luc_i] != 0U)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	else		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	switch(lul_disp_num[luc_i])		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case DataStructConst_FAIL_BTA		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	case DataStructConst_FAIL_BTL		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	case DataStructConst_FAIL_RAM		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_ROM		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_FRM		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_FLH		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_CLK		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_BKB		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_SUB		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_VLT		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_TEP		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case DataStructConst_FAIL_BLE		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case DataStructConst_FAIL_FLW		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case DataStructConst_FAIL_PMP		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case M_SEN_ERR_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	for(luc_i=0; luc_i<4; luc_i++)		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_SEN)		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	if((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	switch(luc_i)		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	case 0		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	case 1		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	case 2		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	case 3		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	default		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Symbol	Range																		
Input	lucs_count	0~30	3	3	2	0	0	0	0	0	0	0	0	0	0	0	0	3	0	30
	lucs_flg	0~30	6	5	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	30
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	30
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0x0008	0x0010	0x0020	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x1000	0x0004	0x4000	0x0001	0x0001	0x0001	0x0001	0xFFFF
	FlowMain_Get_ModeStatus()	0~8	7	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	8
	(stp_denst+0)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0x00	0x00	0x00	0x00	0x00	0x04	0x04	0x04	0x00	0xFF
	(stp_denst+1)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0x04	0x04	0x04	0xFF
	(stp_denst+2)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0x04	0x04	0x04	0xFF
	(stp_denst+3)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0x04	0x04	0x04	0xFF
	(stp_denst+4)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x04	0x04	0x04	0x04	0xFF
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0~255	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	15	0
	stp_nl_disp->uc_lcd_mes_blink[1]	0~255	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	240	255	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowMain_Get_ModeStatus

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_flow_mode_flg	0	0
Judgment			OK

FlowMain_Set_ModeStatus

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_mode_flg	0~8	0	7	8
Output	uc_flow_mode_flg	0~8	0	7	8
Judgment			OK	OK	OK

FlowMain_Select_MenuNum

	Responsible	Last update
Coding	Ikarashi	2021/6/21
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(puc_updown == M_OFF)		○	○	○	○	-	-	-	-
	for(luc_i; luc_i<puc_max; luc_i++)		○	○	○	○	-	-	-	-
	if(*(pucp_item + luc_i) == M_ON)		○	-	○	-	-	-	-	-
	if(luc_i >= puc_max)		-	-	○	○	-	-	-	-
	for(luc_i=0; luc_i<puc_max; luc_i++)		-	-	○	○	-	-	-	-
	if(*(pucp_item + luc_i) == M_ON)		-	-	○	○	-	-	-	-
	else		-	-	-	-	○	○	○	○
	if(puc_num != 0)		-	-	-	-	-	○	-	○
	for(luc_i; luc_i>0; luc_i--)		-	-	-	-	-	○	-	○
	if(*(pucp_item + luc_i) == M_ON)		-	-	-	-	-	○	-	○
	else		-	-	-	-	○	-	○	-
	if((luc_i == 0) && (*(pucp_item + 0) != M_ON))		-	-	-	-	-	-	○	○
	for(luc_i=puc_end; luc_i>0; luc_i--)		-	-	-	-	-	-	○	○
	if(*(pucp_item + luc_i) == M_ON)		-	-	-	-	-	-	-	-
	Symbol	Range								
Input	puc_max	5	5	5	5	5	5	5	5	5
	puc_end	0	0	0	0	0	0	0	0	0
	puc_num	0,5	0	5	0	5	0	5	0	5
	puc_updown	0,1	0	0	0	0	1	1	1	1
	pucp_item	0,1	1	1	0	0	1	1	0	0
Output	luc_i	0~5	1	0	5	5	0	4	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_mainte_display.c

FlowMainteDisplay_Chk_ChangeSensorDateEndNum

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == (HDDEF_GAS_MAX+1))		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000006	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowMainteDisplay_DigitDown_MaintePassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_sub_item_select_num == 0)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	luc_ans	0,1	1	0
Judgment			OK	OK

FlowMainteDisplay_DigitUp_MaintePassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_sub_item_select_num == 3)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000003	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowMainteDisplay_Disp_AlarmPatternOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_AlarmSilenceOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_AutoZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_ChangeSenDateSet

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○	○
	case 0		○	○	-	-	-
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		○	-	-	-	-
	else		-	○	-	-	-
	case 1		-	-	○	-	-
	case 2		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	0	1	2	3
	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000004	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot[0]	0~32	0	0	16	32	0
	stp_nl_disp->uc_lcd_mes_dot[0]	4	4	4	4	4	4
Judgment			OK	OK	OK	OK	OK

FlowMainteDisplay_Disp_ChangeSensorDate

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX+1)		○	-	-	-
	else		-	○	○	○
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000005	0x00000004	0x00000000	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_mes_dot[0]	4	4	4	4	4
Judgment			OK	OK	OK	OK

FlowMainteDisplay_Disp_DemandZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_DispPumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x08	0x08
Judgment			OK

FlowMainteDisplay_Disp_DispSuppressOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x08	0x08
Judgment			OK

FlowMainteDisplay_Disp_DispZeroFllwrOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x08	0x08
Judgment			OK

FlowMainteDisplay_Disp_GasTest

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(ZIPCNumSet_Get_DispFlicker2() == 1)		○	-	-	-
	else		-	○	○	○
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	○	-	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	-	○
	else		-	-	○	-
	if(luc_flg == 0)		-	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker2()	0~2	1	0	0	2
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	0	30
	(stp_denst+luc_i)->uc_alarm_lat	0x00,0xFF	0x00	0xFF	0xFF	0x00
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowMainteDisplay_Disp_IdDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_MaintePassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_MaintePassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteDisplay_Disp_RunningFactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_com_dot	0x40	0x40
Judgment			OK

FlowMainteDisplay_Disp1_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	stp_nl_disp->uc_lcd_com_dot	0~63	64	16	32	0
Judgment			OK	OK	OK	OK

FlowMainteDisplay_Disp2_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	stp_nl_disp->uc_lcd_com_dot	0~32	0	16	32	0
Judgment			OK	OK	OK	OK

FlowMainteDisplay_End_AlarmPatternOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_alarm_pattern	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_AutoZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_auto_zero_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_ChangeSenDateSet

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000004	0xFFFFFFFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

FlowMainteDisplay_End_DemandZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_demand_zero_onoff	0x00~0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_DispPumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_disp_pump_off	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_DispSuppressOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_disp_suppress_val	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_DispZeroFllwrOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_disp_zero_fllwr	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_IdDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_id_disp_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_MaintePassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_mainte_pass_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_End_MaintePassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->uc_chr_temporary_item[0]	0x00,0xFF	0x00	0xFF
	stp_zipu_num->uc_chr_temporary_item[1]	0x00,0xFF	0x00	0xFF
	stp_zipu_num->uc_chr_temporary_item[2]	0x00,0xFF	0x00	0xFF
	stp_zipu_num->uc_chr_temporary_item[3]	0x00,0xFF	0x00	0xFF
Output	stp_data->us_mainte_pass_num	0,21161	0	21161
Judgment			OK	OK

FlowMainteDisplay_Init_AlarmPatternOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_alarm_pattern	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_AlarmSilenceOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_alarm_silence	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_AutoZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_auto_zero_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_ChangeSensorDate

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i < (HDDEF_GAS_MAX+1); luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	30
Output	stp_zipu_num->ul_item_select_num	0,1	1	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowMainteDisplay_Init_DemandZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_demand_zero_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_DispPumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_disp_pump_off	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_DispSuppressOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_disp_suppress_val	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_DispZeroFllwrOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_disp_zero_fllwr	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_IdDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_id_disp_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_MainteMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_mode_select_num	0	0
	stp_zipu_num->uc_mode_select_updown	0	0
Judgment			OK

FlowMainteDisplay_Init_MaintePassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_mainte_pass_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Init_MaintePassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_mainte_pass_num	0~9999	0	9999	10000
Output	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_chr_temporary_item[0]	0~9	0	9	0
	stp_zipu_num->uc_chr_temporary_item[1]	0~9	0	9	0
	stp_zipu_num->uc_chr_temporary_item[2]	0~9	0	9	0
	stp_zipu_num->uc_chr_temporary_item[3]	0~9	0	9	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowMainteDisplay_Select_AlarmPatternOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_AlarmSilenceOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_AutoZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_ChangeSensorDate

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)	○	○	○	-	-	-	-	-
	if(stp_zipu_num->ul_item_select_num >= (HDDEF_GAS_MAX+1))	○	-	○	-	-	-	-	-
	else	-	○	-	-	-	-	-	-
	for(luc_i=stp_zipu_num->ul_item_select_num; luc_i < (HDDEF_GAS_MAX+1); luc_i++)	○	○	○	-	-	-	-	-
	if(luc_i == HDDEF_GAS_MAX)	○	-	-	-	-	-	-	-
	else	-	-	○	-	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)	-	-	○	-	-	-	-	-
	else	○	-	○	-	-	-	-	-
	else	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == 0)	-	-	-	○	-	-	-	-
	else	-	-	-	-	○	○	○	○
	for(luc_i=(stp_zipu_num->ul_item_select_num-1); luc_i > 0; luc_i--)	-	-	-	-	○	○	○	○
	if(luc_i == HDDEF_GAS_MAX)	-	-	-	-	-	○	-	-
	else	-	-	-	-		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)	-	-	-	-	-	-	○	○
	else	-	-	-	-	○	○	-	-
	if((luc_i == 0) && ((stp_GasInfo_addr+0)->uc_gas_comb == 0x00))	-	-	-	-	○	-	-	-
	else	-	-	-	-	-	○	○	○
	Symbol	Range							
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_item_select_num	0~8	6	4	6	0	3	8	2	2
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	29	0	0	29	30	30
	(stp_GasInfo_addr+0)->uc_gas_comb	0~30	0	0	0	0	0	30	29	0
Output	None	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

FlowMainteDisplay_Select_DemandZeroOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_DispPumpOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_DispSuppressOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_DispZeroFllwrOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_IdDispOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_MainteMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_mode_select_updown == M_OFF)		○	○	○	○	-	-	-	-	-
	for(luc_i; luc_i<M_MAX_NUM; luc_i++)		○	○	○	-	-	-	-	-	-
	if(uc_mainte_mode_item[luc_i] == M_ON)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	if(luc_i >= M_MAX_NUM)		-	○	○	○	-	-	-	-	-
	for(luc_i=0; luc_i<M_MAX_NUM; luc_i++)		-	○	-	○	-	-	-	-	-
	if(uc_mainte_mode_item[luc_i] == M_ON)		-	-	-	○	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_mode_select_num != 0)		-	-	-	-	○	○	○	-	○
	for(luc_i; luc_i>0; luc_i--)		-	-	-	-	-	-	-	○	-
	if(uc_mainte_mode_item[luc_i] == M_ON)		-	-	-	-	○	○	○	○	○
	else		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	○
	if((luc_i == 0) && (uc_mainte_mode_item[0] != M_ON))		-	-	-	-	○	○	-	○	○
	for(luc_i=M_END_NUM; luc_i>0; luc_i--)		-	-	-	-	○	○	-	○	○
	if(uc_mainte_mode_item[luc_i] == M_ON)		-	-	-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-	○	○
	else		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	stp_zipu_num->uc_mode_select_updown	0~2	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x0000001E	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	uc_mainte_mode_item[luc_i]	0~2	1	0	1	1	1	0	1	0	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowMainteDisplay_Select_MaintePassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteDisplay_Select_MaintePassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num-> uc_chr_temporary_item[stp_zipu_num-> ul_sub_item_select_num], 1, 9, 0, stp_zipu_num-> uc_sub_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->uc_chr_temporary_item [stp_zipu_num->ul_sub_item_select_num]	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteDisplay_Set_FactoryDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Chk_MainteMenuItem

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((stp_data->uc_exp_flg == DSC_TGT_JAPAN) || (stp_data->uc_exp_flg == DSC_TGT_JG))		○	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_data->uc_exp_flg	0~5	0	4	1	5
Output	uc_mainte_mode_item[M_M_NOTICE_NUM]	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

FlowMainteDisplay_Set_MainteMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/22
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Combination pairs	19
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19
Root	switch(stp_zipu_num->ui_mode_select_num)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case M_GASCAL_NUM		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case M_GAS_TEST_NUM		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case M_CHG_SENEOR_NUM		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case M_BUMPTEST_NUM		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case M_AL_ACTION_NUM		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case M_D_ZERO_NUM		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case M_A_ZERO_NUM		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case M_ID_DISP_NUM		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case M_M_NOTICE_NUM		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case M_SUPPRESS_NUM		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case M_Z_FLLWR_NUM		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case M_SUP_DISP_NUM		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case M_Z_FLL_DISP_NUM		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case M_DATE_SET_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case M_PASS_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	case M_ROMSUM_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	case M_RESTORE_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	case M_END_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range																			
Input	stp_zipu_num->ui_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005	0x00000006	0x00000007	0x00000008	0x00000009	0x0000000A	0x0000000B	0x0000000C	0x0000000D	0x0000000E	0x0000000F	0x00000012	0x00000013	0xFFFFFFFF
Output	stp_ril disp->uc_lcd_com_dot	0x00~0x40	0x00	0x10	0x10	0x00	0x00	0x10	0x40	0x40	0x00	0x40	0x08	0x08	0x08	0x08	0x08	0x00	0x00	0x00	0x00
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_mainte_notice_display.c

FlowMainteNoticeDisplay_Disp_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_blink	0x7F	0x7F	0x7F	0x7F	0x7F
Judgment			OK	OK	OK	OK

FlowMainteNoticeDisplay_Disp_MainteNoticeDay

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF
Judgment			OK

FlowMainteNoticeDisplay_Disp_MainteNoticeMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(stp_zipu_num->ul_sub_mode_select_num)		○	○	○	○	○	○
	case M_M_NOTICE_RESET_NUM		○	-	-	-	-	-
	case M_M_NOTICE_DISP_NUM		-	○	-	-	-	-
	case M_M_NOTICE_DAYS_NUM		-	-	○	-	-	-
	case M_M_NOTICE_EXPRD_NUM		-	-	-	○	-	-
	case M_M_END_NUM		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000005	0xFFFFFFFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

FlowMainteNoticeDisplay_Disp_MainteNoticeOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteNoticeDisplay_Disp_MainteNoticeReset

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ZIPCNumSet_Get_DispFlicker3())		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ZIPCNumSet_Get_DispFlicker3()	0~3	0	1	2	3
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowMainteNoticeDisplay_End_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_mainte_lmt_chk	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteNoticeDisplay_End_MainteNoticeDay

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->us_mainte_notice_day	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

FlowMainteNoticeDisplay_End_MainteNoticeOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_mainte_notice_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowMainteNoticeDisplay_End_MainteNoticeReset

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowMainteNoticeDisplay_Init_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_mainte_lmt_chk	0~3	0	2	3
Output	stp_zipu_num->ul_item_select_num	0~3	0	2	3
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowMainteNoticeDisplay_Init_MainteNoticeDay

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_mainte_notice_day	1~1001	1	1000	1001
Output	stp_zipu_num->ul_item_select_num	1~1001	1	1000	1001
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowMainteNoticeDisplay_Init_MainteNoticeMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_sub_mode_select_num	0	0
	stp_zipu_num->uc_sub_mode_select_updown	0	0
Judgment			OK

FlowMainteNoticeDisplay_Init_MainteNoticeOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_mainte_notice_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowMainteNoticeDisplay_Select_CalExprd

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)2, (U_LNG)0, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Judgment			OK	OK

FlowMainteNoticeDisplay_Select_MainteNoticeDay

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)M_NOTICE_DAYS_MAX, (U_LNG)M_NOTICE_DAYS_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)M_NOTICE_DAYS_MAX, (U_LNG)M_NOTICE_DAYS_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	puc_num	0x00~0xFF	0x00	0x01	0xFF
Output	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK

FlowMainteNoticeDisplay_Select_MainteNoticeMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_zipu_num->ul_sub_mode_select_num == M_M_NOTICE_CHK_GAS_NUM)		○	○	-	-
	if(stp_zipu_num->uc_sub_mode_select_updown == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_sub_mode_select_num, 1, (U_LNG)M_M_END_NUM, (U_LNG)M_M_NOTICE_RESET_NUM, stp_zipu_num->uc_sub_mode_select_updown, M_OFF)	0x00000000~0xFFFFFFFF	0x00000004	0x00000004	0x00000000	0xFFFFFFFF
	stp_zipu_num->uc_sub_mode_select_updown	0~2	0	1	1	2
	stp_zipu_num->ul_sub_mode_select_num	0x00000000~0xFFFFFFFF	0x00000004	0x00000004	0x00000000	0xFFFFFFFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FlowMainteNoticeDisplay_Select_MainteNoticeOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_meas_display.c

FlowMeasDisplay_Set_MeasDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○
	if(luc_flg != 0x00)		○	-	-	-	-	○
	else		-	○	○	○	○	-
	if((stp_GasInfo_addr+0)->uc_gas_comb != (stp_GasInfo_addr+0)->uc_sub_comb)		-	-	-	-	-	-
	else		-	-	○	○	○	-
	if(FuncTempWarning_Get_OutOfRangeFlg() != M_OFF)		○	○	○	○	○	-
	if(luc_flickr == M_ON)		○	○	-	-	-	-
	else		-	-	○	○	○	-
	switch(luc_flickr)		○	○	○	○	○	-
	case 0		○	○	-	-	-	-
	if(FuncTempWarning_Get_OutOfRangeFlg() == 2)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	case 1		-	-	○	○	-	-
	for(luc_i=0; luc_i<8; luc_i++)		-	-	○	○	-	-
	if(stp_sub_ad->fl_disp_temp < 0)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	default		-	-	-	-	○	-
	else		-	-	-	-	-	○
	Symbol	Range						
Input	FuncTempWarning_Get_OutOfRangeFlg()	0~3	2	1	1	1	3	0
	(stp_denst+luc_i)->uc_alarm_lat	0x00,0xFF	0xFF	0x00	0x00	0x00	0x00	0xFF
	(stp_GasInfo_addr+0)->uc_gas_comb	0~30	0	29	29	29	29	30
	(stp_GasInfo_addr+0)->uc_sub_comb	0~30	0	0	29	29	29	30
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	0	1	1	2	2
	ZIPCNumSet_Get_DispFlicker3()	0~3	0	0	2	2	3	3
	stp_sub_ad->fl_disp_temp	-55~86	0	0	-55	85	0	86
Output	stp_nl_disp->uc_lcd_icon[3]	0,8	0	0	8	8	8	8
	stp_nl_disp->uc_lcd_com[7]	0,67	0	0	67	67	67	67
Judgment			OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_pass_display.c

FlowPassDisplay_Chg_PassNum

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(st_pass.uc_pass_up)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	Tool_OneCount((U_LNG)st_pass.uc_pass_num [st_pass.uc_pass_digit-1], 1, 9, 0, M_ON, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)st_pass.uc_pass_num [st_pass.uc_pass_digit-1], 1, 9, 0, M_OFF, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	st_pass.uc_pass_up	0~2	0	1	2
Output	st_pass.uc_pass_num[st_pass.uc_pass_digit-1]	0x00,0xFF	0x00	0x00	0xFF
Judgment			OK	OK	OK

FlowPassDisplay_Chg_UpDown

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(st_pass.uc_pass_up == M_ON)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	st_pass.uc_pass_up	0~2	0	1	2
Output	st_pass.uc_pass_up	0~2	1	0	1
Judgment			OK	OK	OK

FlowPassDisplay_Chk_End

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(st_pass.uc_pass_digit == M_PASS_END_NUM)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	st_pass.uc_pass_digit	1~6	1	5	6
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowPassDisplay_Disp_PassErr

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowPassDisplay_Get_PassType

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	st_pass.uc_pass_type	0	0
Judgment			OK

FlowPassDisplay_Get_Return

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	st_pass.uc_pass_return	0	0
Judgment			OK

FlowPassDisplay_Input_Pass

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	st_pass.uc_pass_num[0]	0~10	0	9	10
	st_pass.uc_pass_num[1]	0~10	0	9	10
	st_pass.uc_pass_num[2]	0~10	0	9	10
	st_pass.uc_pass_num[3]	0~10	0	9	10
	st_pass.uc_pass_digit	1~6	1	5	6
Output	stp_nl_disp->uc_lcd_mes[0][2]	48~58	48	57	58
	stp_nl_disp->uc_lcd_mes[0][3]	48~58	48	57	58
	stp_nl_disp->uc_lcd_mes[0][4]	48~58	48	57	58
	stp_nl_disp->uc_lcd_mes[0][5]	48~58	48	57	58
	stp_nl_disp->uc_lcd_mes_blink[0]	1~32	32	2	1
Judgment			OK	OK	OK

FlowPassDisplay_Set_Enter

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(puc_num == 0)		○	○	-	-	-
	if(st_pass.uc_pass_digit > M_PASS_RETURN_NUM)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	if(st_pass.uc_pass_digit < M_PASS_END_NUM)		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	puc_num	0~2	0	0	1	1	2
	st_pass.uc_pass_digit	1~6	5	1	1	5	6
Output	luc_ans	0,1	0	1	0	1	1
Judgment			OK	OK	OK	OK	OK

FlowPassDisplay_Set_PassInit

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_type > FlowPassDisplay_PASS_FACT_Type)		-	○	-	-
	else		○	-	○	○
	Symbol	Range				
Input	puc_type	0~8	0	2	7	8
Output	st_pass.uc_pass_num[0]	0	0	0	0	0
	st_pass.uc_pass_num[1]	0	0	0	0	0
	st_pass.uc_pass_num[2]	0	0	0	0	0
	st_pass.uc_pass_num[3]	0	0	0	0	0
	st_pass.uc_pass_up	0	0	0	0	0
	st_pass.uc_pass_digit	1	1	1	1	1
	st_pass.uc_pass_type	0~8	0	2	7	8
	st_pass.uc_pass_return	0,1	0	0	1	1
Judgment			OK	OK	OK	OK

FlowPassDisplay_Chk_Pass

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification paths	18
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18
Root	swilch(st_pass.uc_pass_type)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case FlowPassDisplay_PASS_USER_Type	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(lus_pass == M_USER_PASSWORD || lus_pass == stp_data->us_user_pass_num)	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case FlowPassDisplay_PASS_MAINTE_Type	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(lus_pass == M_MAINTE_PASSWORD || lus_pass == stp_data->us_mainte_pass_num)	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case FlowPassDisplay_PASS_FACT_Type	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(lus_pass == M_GAS_SEL_PASSWORD)	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(lus_pass == M_FACT_PASSWORD)	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case FlowPassDisplay_PASS_INI_LIMIT_Type	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-
	if(lus_pass == M_USER_PASSWORD || lus_pass == stp_data->us_user_pass_num)	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case FlowPassDisplay_PASS_AUTO_AIR_Type	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-
	if(lus_pass == M_USER_PASSWORD || lus_pass == stp_data->us_user_pass_num)	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case FlowPassDisplay_PASS_DEMAND_AIR_Type	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
	if(lus_pass == M_USER_PASSWORD || lus_pass == stp_data->us_user_pass_num)	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case FlowPassDisplay_PASS_ALARM_RESET_Type	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-
	if(lus_pass == M_USER_PASSWORD || lus_pass == stp_data->us_user_pass_num)	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	case FlowPassDisplay_PASS_POWER_OFF_Type	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-
	if(lus_pass == M_USER_PASSWORD || lus_pass == stp_data->us_user_pass_num)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	else	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	default	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range																	
Input	st_pass.uc_pass_type	0~8	0	0	1	1	2	2	2	3	3	4	4	5	5	6	6	7	7	8
	st_pass.uc_pass_num[0]	0~10	0	0	8	0	2	1	0	0	0	0	0	0	0	0	0	0	9	10
	st_pass.uc_pass_num[1]	0~10	4	0	1	0	0	9	0	4	0	4	0	4	0	4	0	4	0	10
	st_pass.uc_pass_num[2]	0~10	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0	0	0	10
	st_pass.uc_pass_num[3]	0~10	5	0	2	0	4	4	0	5	0	5	0	5	0	5	0	5	9	10
	stp_data->us_user_pass_num	0~10000	405	9999	0	0	2014	1994	9999	405	9999	405	9999	405	9999	405	9999	405	9999	10000
	stp_data->us_mainte_pass_num	0~10000	0	0	8102	9999	0	0	0	0	0	0	0	0	0	0	0	0	0	10000
Output	luc_num	0~9	1	0	2	0	3	4	0	5	0	6	0	7	0	8	0	9	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_proces_display.c

FlowProcesDisplay_Set_TurnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowProcesDisplay_Set_UpData

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowProcesDisplay_10msec_Interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	uss_purge_count_time	0~2	0	1	2
Output	uss_purge_count_time	0,1	0	0	1
Judgment			OK	OK	OK

FlowProcesDisplay_Chk_PurgeTimeEnd

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	uss_purge_count_time	0~2	0	1	2
Output	uss_purge_count_time	0,1	1	0	0
Judgment			OK	OK	OK

FlowProcesDisplay_Get_PumpOffFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_pump_off_flag	0~2	0	1	2
Output	fbl_pump_off_flag	0~2	0	1	2
Judgment			OK	OK	OK

FlowProcesDisplay_Get_PurgeFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_pump_off_flag	0~2	0	1	2
Output	fbl_pump_off_flag	0~2	0	1	2
Judgment			OK	OK	OK

FlowProcesDisplay_Init_Process

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fbl_pump_off_flag	0	0
	fbl_purge_flag	0	0
	uss_purge_count_time	0	0
Judgment			OK

FlowProcesDisplay_Set_PumpOffFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_pump_off_flag	0~2	0	1	2
Output	fbl_pump_off_flag	0~2	0	1	2
Judgment			OK	OK	OK

FlowProcesDisplay_Set_Purge

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowProcesDisplay_Set_PurgeFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_pump_off_flag	0~2	0	1	2
Output	fbl_pump_off_flag	0~2	0	1	2
Judgment			OK	OK	OK

FlowProcesDisplay_Set_PurgeTime

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_purge_count_time	3000	3000
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_snaplog_display.c

FlowSnaplogDisplay_Disp_SnapDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(ZIPCNumSet_Get_DispFlicker4())		○	○	○	○	○
	case 0:		○	-	-	-	-
	case 1:		-	○	-	-	-
	case 2:		-	-	○	-	-
	case 3:		-	-	-	○	-
	default:			-	-	-	○
	Symbol	Range					
Input	ZIPCNumSet_Get_DispFlicker4()	0~4	0	1	2	3	4
	uc_snap_rec_pnt	0	0	0	0	0	0
	uc_lcd_mes[0][0]	0	0	0	0	0	0
	uc_lcd_mes[1][5]	0	0	0	0	0	0
	uc_lcd_mes[0][1]	0	0	0	0	0	0
	uc_lcd_mes[0][4]	0	0	0	0	0	0
	uc_lcd_mes[0][6]	0	0	0	0	0	0
	uc_lcd_mes[1][4]	0	0	0	0	0	0
	uc_lcd_mes[1][6]	0	0	0	0	0	0
	uc_lcd_com[1]	0	0	0	0	0	0
	uc_st_id[0]	0	0	0	0	0	0
Output	uc_lcd_mes_dot	0,4	0	0	4	0	0
	uc_lcd_com_dot	0~32	16	0	16	32	0
Judgment			OK	OK	OK	OK	OK

FlowSnaplogDisplay_Set_SnapLog

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowSnaplogDisplay_Disp_SnapSaved

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	uc_lcd_com[1]	0~255	0	255
Output	None	-	-	-
Judgment			OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	flow_user_display.c

FlowUserDisplay_Chk_UserPassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_item_select_num	0	0
Judgment			OK

FlowUserDisplay_Chk_ZeroSuppressEsc

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000004	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowUserDisplay_DigitDown_UserPassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_sub_item_select_num == 0)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	luc_ans	0,1	1	0
Judgment			OK	OK

FlowUserDisplay_DigitUp_UserPassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_zipu_num->ul_sub_item_select_num == 3)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000003	0xFFFFFFFF
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

FlowUserDisplay_Disp_BackLightTime

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_item_select_num == 0)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_mes_blink[0]	0xFF	0xFF	0xFF
Judgment			OK	OK

FlowUserDisplay_Disp_DispModeSettingItemDisplay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowUserDisplay_Disp_KeyTone

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowUserDisplay_Disp_LunchBreak

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowUserDisplay_Disp_UserPassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowUserDisplay_Disp_UserPassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_zipu_num->uc_chr_temporary_item[0]	0~10	0	9	10
	stp_zipu_num->uc_chr_temporary_item[1]	0~10	0	9	10
	stp_zipu_num->uc_chr_temporary_item[2]	0~10	0	9	10
	stp_zipu_num->uc_chr_temporary_item[3]	0~10	0	9	10
	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_mes[0][2]	47~58	47	57	58
	stp_nl_disp->uc_lcd_mes[0][3]	47~58	47	57	58
	stp_nl_disp->uc_lcd_mes[0][4]	47~58	47	57	58
	stp_nl_disp->uc_lcd_mes[0][5]	47~58	47	57	58
	stp_nl_disp->uc_lcd_mes_blink[0]	0,32	32	32	0
Judgment			OK	OK	OK

FlowUserDisplay_Disp_ZeroFollowerGas

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_item_select_num < HDDEF_GAS_MAX)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0x10	0x08	0x00
Judgment			OK	OK

FlowUserDisplay_Disp_ZeroFollowerSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(stp_zipu_num->ul_sub_item_select_num)		○	○	○
	case M_OFF		○	-	-
	case M_ON		-	○	-
	default		-	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0x08	0x08	0x08	0x08
Judgment			OK	OK	OK

FlowUserDisplay_Disp_ZeroSuppressGas

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(stp_zipu_num->ul_item_select_num < HDDEF_GAS_MAX)		○	-
	else		-	○
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0x00,0x08	0x08	0x00
Judgment			OK	OK

FlowUserDisplay_Disp_ZeroSuppressSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(stp_zipu_num->ul_sub_item_select_num)		○	○	○
	case M_OFF		○	-	-
	case M_ON		-	○	-
	default		-	-	○
	Symbol	Range			
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0x08	0x08	0x08	0x08
Judgment			OK	OK	OK

FlowUserDisplay_End_BackLightTime

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->us_back_light_time	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

FlowUserDisplay_End_DateSetteng

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FlowUserDisplay_End_DispModeSettingItemDisplay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_disp_setting_item_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_End_KeyTone

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_key_tone_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_End_LunchBreak

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_lunch_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_End_UserPassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_data->uc_user_pass_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_End_UserPassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_zipu_num->uc_chr_temporary_item[0]	0~10	0	9	10
	stp_zipu_num->uc_chr_temporary_item[1]	0~10	0	9	10
	stp_zipu_num->uc_chr_temporary_item[2]	0~10	0	9	10
	stp_zipu_num->uc_chr_temporary_item[3]	0~10	0	9	10
Output	stp_data->us_user_pass_num	0~11110	0	9999	11110
Judgment			OK	OK	OK

FlowUserDisplay_End_ZeroFollowerSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_follower_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_End_ZeroSuppressSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_zipu_num->ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_suppress_onoff	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_Init_BackLightTime

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_back_light_time	0~256	0	255	256
Output	stp_zipu_num->ul_item_select_num	0~256	0	255	256
	stp_zipu_num->uc_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowUserDisplay_Init_BumpLimitUserMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_item_select_num	0	0
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowUserDisplay_Init_CalLimitUserMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_item_select_num	1	1
	stp_zipu_num->uc_item_select_updown	0	0
Judgment			OK

FlowUserDisplay_Init_DateSetteng

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(luc_num[0] > 99)		-	-	○
	else		○	○	-
	if(luc_num[1] <= 0 || luc_num[1] > 12)		-	-	○
	else		○	○	-
	if(luc_num[2] <= 0 || luc_num[2] > RTCTime_Chk_Day((U_SHORT)(luc_num[0])+2000, luc_num[1]))		-	-	○
	else		○	○	-
	if(luc_num[3] > 23)		-	-	○
	else		○	○	-
	if(luc_num[4] > 59)		-	-	○
	else		○	○	-
	Symbol	Range			
Input	luc_num[0]	0~100	0	99	100
	luc_num[1]	1~13	1	12	13
	luc_num[2]	1~32	1	31	32
	luc_num[3]	0~24	0	23	24
	luc_num[4]	0~60	0	59	60
	RTCTime_Chk_Day((U_SHORT)(luc_num[0])+2000, luc_num[1]))	29~32	29	32	31
Output	stp_zipu_num->ul_item_select_num	0	0	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
	stp_zipu_num->uc_chr_temporary_item[0]	0,99	0	99	0
	stp_zipu_num->uc_chr_temporary_item[1]	1,12	1	12	1
	stp_zipu_num->uc_chr_temporary_item[2]	1,31	1	31	1
	stp_zipu_num->uc_chr_temporary_item[3]	0,23	0	23	0
	stp_zipu_num->uc_chr_temporary_item[4]	0,59	0	59	0
	stp_zipu_num->uc_chr_temporary_item[5]	0	0	0	0
Judgment			OK	OK	OK

FlowUserDisplay_Init_DispModeSettingItemDisplay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_disp_setting_item_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Init_KeyTone

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_key_tone_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Init_LunchBreak

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_lunch_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Init_UserMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_zipu_num->ul_mode_select_num	0	0
	stp_zipu_num->uc_mode_select_updown	0	0
Judgment			OK

FlowUserDisplay_Init_UserPassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->uc_user_pass_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Init_UserPassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_user_pass_num	0~10000	0	9999	10000
Output	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_chr_temporary_item[0]	0,9	0	9	0
	stp_zipu_num->uc_chr_temporary_item[1]	0,9	0	9	0
	stp_zipu_num->uc_chr_temporary_item[2]	0,9	0	9	0
	stp_zipu_num->uc_chr_temporary_item[3]	0,9	0	9	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowUserDisplay_Init_ZeroFollowerGas

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (luc_i != DataStructConst_O2_GAS_CH) && ((luc_i != DataStructConst_CO_CAN_H2_H2_CH) ‖ ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	○	○
	else		○	○	○
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	30
Output	stp_zipu_num->ul_item_select_num	0,4	4	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
Judgment			OK	OK	OK

FlowUserDisplay_Init_ZeroFollowerSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_follower_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_sub_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Init_ZeroSuppressGas

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (luc_i != DataStructConst_O2_GAS_CH) && ((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	○	○
	else		○	○	○
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	30
Output	stp_zipu_num->ul_item_select_num	0,4	4	0	0
	stp_zipu_num->uc_item_select_updown	0	0	0	0
	stp_zipu_num->ul_sub_item_select_num	0	0	0	0
	stp_zipu_num->uc_sub_item_select_updown	0	0	0	0
Judgment			OK	OK	OK

FlowUserDisplay_Init_ZeroSuppressSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+(stp_zipu_num->ul_item_select_num))->uc_suppress_onoff	0~2	0	1	2
Output	stp_zipu_num->ul_sub_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Select_BackLightTime

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_num == 1)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 1, (U_LNG)M_BACK_LIGHT_TIME_MAX, (U_LNG)M_BACK_LIGHT_TIME_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num->ul_item_select_num, 10, (U_LNG)M_BACK_LIGHT_TIME_MAX, (U_LNG)M_BACK_LIGHT_TIME_MIN, stp_zipu_num->uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	puc_num	0x00~0xFF	0x00	0x01	0xFF
Output	stp_zipu_num->ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
Judgment			OK	OK	OK

FlowUserDisplay_Select_DateSettengItem

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(puc_type == 0)		○	○	○	-	-	-
	if(stp_zipu_num->ul_item_select_num == 2)		○	○	-	-	-	-
	if(stp_zipu_num->uc_chr_temporary_item[2] > RTCTime_Chk_Day((U_SHORT)(stp_zipu_num-> uc_chr_temporary_item[0])+2000, stp_zipu_num->uc_chr_temporary_item[1]))		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	if(stp_zipu_num->ul_item_select_num == 5)		-	-	○	-	-	-
	else		○	○	-	-	-	-
	else		-	-	-	○	○	○
	if(stp_zipu_num->ul_item_select_num > 0)		-	-	-	○	-	○
	else		-	-	-	-	○	-
	Symbol	Range						
Input	puc_type	0~2	0	0	0	1	1	2
	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000001	0x00000001	0x00000004	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_zipu_num->uc_chr_temporary_item[2]	28~32	31	28	32	28	28	32
	RTCTime_Chk_Day((U_SHORT)(stp_zipu_num-> uc_chr_temporary_item[0])+2000, stp_zipu_num-> uc_chr_temporary_item[1]))	28~32	28	31	32	28	28	32
Output	stp_zipu_num->uc_chr_temporary_item[2]	28~32	28	28	32	28	28	32
	luc_ans	0~2	0	0	1	0	2	0
Judgment			OK	OK	OK	OK	OK	OK

FlowUserDisplay_Select_DateSettingNum

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○	○	○
	case 0		○	-	-	-	-	-
	case 1		-	○	-	-	-	-
	case 2		-	-	○	-	-	-
	case 3		-	-	-	○	-	-
	case 4		-	-	-	-	○	-
	default		-	-	-	-	-	○
	if(puc_num == 10)		-	-	-	-	○	-
	else		○	○	○	○	-	○
	Symbol	Range						
Input	puc_num	0x00~0xFF	0x00	0x00	0x00	0x00	0x0A	0xFF
	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num-> uc_chr_temporary_item[stp_zipu_num-> ul_item_select_num], 10, (U_LNG)luc_max, (U_LNG)luc_min, stp_zipu_num-> uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	Tool_OneCount((U_LNG)stp_zipu_num-> uc_chr_temporary_item[stp_zipu_num-> ul_item_select_num], 1, (U_LNG)luc_max, (U_LNG)luc_min, stp_zipu_num-> uc_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->uc_chr_temporary_item [stp_zipu_num->ul_item_select_num]	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
Judgment			OK	OK	OK	OK	OK	OK

FlowUserDisplay_Select_DispModeSettingItemDisplay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Select_KeyTone

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Select_LunchBreak

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Select_UserPassOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Select_UserPassSelect

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	Tool_OneCount((U_LNG)stp_zipu_num->uc_chr_temporary_item[stp_zipu_num->ul_sub_item_select_num], 1, 9, 0, stp_zipu_num->uc_sub_item_select_updown, M_OFF)	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	stp_zipu_num->uc_chr_temporary_item[stp_zipu_num->ul_sub_item_select_num]	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

FlowUserDisplay_Select_ZeroFollowerGas

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	for(luc_i; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	-	-	-	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (luc_i != DataStructConst_O2_GAS_CH) && ((luc_i != 2) || ((stp_GasInfo_addr+2)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	○	-	-	-	-
	else		○	-	-	-	-	-
	else		-	-	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == 0)		-	-	○	-	-	○
	else		-	-	-	○	○	-
	for(luc_i=(stp_zipu_num->ul_item_select_num-1); luc_i > 0; luc_i--)		-	-	-	○	○	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (luc_i != DataStructConst_O2_GAS_CH) && ((luc_i != 2) || ((stp_GasInfo_addr+2)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	-	-	-	○	-
	else		-	-	-	○	-	-
	if((luc_i == 0) && ((stp_GasInfo_addr+0)->uc_gas_comb == 0x00))		-	-	-	○	-	-
	else		-	-	-	-	○	-
	Symbol	Range						
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	1	1	1	2
	stp_zipu_num->ul_item_select_num	0,4	4	0	0	4	4	0
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	0	0	29	30
	(stp_GasInfo_addr+0)->uc_gas_comb	0~30	0	0	29	0	29	30
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

FlowUserDisplay_Select_ZeroFollowerSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_sub_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_sub_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Select_ZeroSuppressGas

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(stp_zipu_num->uc_item_select_updown == M_OFF)		○	○	-	-	-	-
	if(stp_zipu_num->ul_item_select_num == HDDEF_GAS_MAX)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	for(luc_i; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	-	-	-	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+2)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	○	-	-	-	-
	else		○	-	-	-	-	-
	else		-	-	○	○	○	○
	if(stp_zipu_num->ul_item_select_num == 0)		-	-	○	-	-	○
	else		-	-	-	○	○	-
	for(luc_i=(stp_zipu_num->ul_item_select_num-1); luc_i > 0; luc_i--)		-	-	-	○	○	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+2)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)))		-	-	-	-	○	-
	else		-	-	-	○	-	-
	if((luc_i == 0) && ((stp_GasInfo_addr+0)->uc_gas_comb == 0x00))		-	-	-	○	-	-
	else		-	-	-	-	○	-
	Symbol	Range						
Input	stp_zipu_num->uc_item_select_updown	0~2	0	0	1	1	1	2
	stp_zipu_num->ul_item_select_num	0~5	4	0	0	4	4	5
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	0	0	29	30
	(stp_GasInfo_addr+0)->uc_gas_comb	0~30	0	0	29	0	29	30
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

FlowUserDisplay_Select_ZeroSuppressSet

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Tool_CHR_Chg_OnOff (stp_zipu_num->ul_sub_item_select_num)	0~2	0	1	2
Output	stp_zipu_num->ul_sub_item_select_num	0~2	0	1	2
Judgment			OK	OK	OK

FlowUserDisplay_Set_DateSetting

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(stp_zipu_num->ul_item_select_num)		○	○	○	○	○	○
	case 0		○	-	-	-	-	-
	case 1		-	○	-	-	-	-
	case 2		-	-	○	-	-	-
	case 3		-	-	-	○	-	-
	case 4		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	stp_zipu_num->ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000004	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_mes[0][3]	45	45	45	45	45	45	45
	stp_nl_disp->uc_lcd_mes_dot[0]	4	4	4	4	4	4	4
	stp_nl_disp->uc_lcd_mes_dot[1]	5	5	5	5	5	5	5
Judgment			OK	OK	OK	OK	OK	OK

FlowUserDisplay_Set_EndDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Chk_UserMenuItem

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if((stp_data->uc_exp_flg == DSC_TGT_JAPAN)||(stp_data->uc_exp_flg == DSC_TGT_JG)		○	○	-
	else		-	-	○
	if(stp_data->uc_disp_zero_fllwr == M_OFF)		○	-	-
	else		-	○	○
	if(stp_data->uc_disp_suppress_val == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	stp_data->uc_exp_flg	0~4	0	3	4
	stp_data->uc_disp_zero_fllwr	0~2	0	1	2
	stp_data->uc_disp_suppress_val	0~2	0	1	2
Output	uc_user_mode_item[U_CAL_LIM_NUM]	0,1	0	0	1
	uc_user_mode_item[U_Z_FLLWR_NUM]	0,1	0	1	1
	uc_user_mode_item[U_SUPPRESS_NUM]	0,1	0	1	1
Judgment			OK	OK	OK

FlowUserDisplay_Select_UserMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(stp_zipu_num->uc_mode_select_updown == M_OFF)		○	○	○	○	-	-	-	-	-
	for(luc_i; luc_i<U_MAX_NUM; luc_i++)		○	○	○	-	-	-	-	-	-
	if(uc_user_mode_item[luc_i] == M_ON)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	if(luc_i >= U_MAX_NUM)		-	○	○	○	-	-	-	-	-
	for(luc_i=0; luc_i<U_MAX_NUM; luc_i++)		-	○	-	○	-	-	-	-	-
	if(uc_user_mode_item[luc_i] == M_ON)		-	-	-	○	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	○	○
	if(stp_zipu_num->ul_mode_select_num != 0)		-	-	-	-	○	○	○	-	○
	for(luc_i; luc_i>0; luc_i--)		-	-	-	-	-	-	-	○	-
	if(uc_user_mode_item[luc_i] == M_ON)		-	-	-	-	○	○	○	○	○
	else		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	○
	if((luc_i == 0) && (uc_user_mode_item[0] != M_ON))		-	-	-	-	○	○	-	○	○
	for(luc_i=U_END_NUM; luc_i>0; luc_i--)		-	-	-	-	○	○	-	○	○
	if(uc_user_mode_item[luc_i] == M_ON)		-	-	-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-	○	○
	else		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	stp_zipu_num->uc_mode_select_updown	0~2	0	0	0	0	1	1	1	1	2
	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x0000001E	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	uc_user_mode_item[luc_i]	0~2	1	0	1	1	1	0	1	0	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

FlowUserDisplay_Set_UserMenu

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	17

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17
Root	switch(stp_zipu_num->ul_mode_select_num)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case U_BUMPTEST_NUM		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case U_GASCAL_NUM		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case U_CAL_LIM_NUM		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case U_BUMP_LIM_NUM		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	case U_ALARM_P_NUM		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case U_LUNCH_NUM		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case U_BEEP_NUM		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	case U_LCD_TIME_NUM		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case U_KEY_TONE_NUM		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	case U_DISP_ON_NUM		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	case U_SUPPRESS_NUM		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case U_Z_FLLWR_NUM		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case U_DATE_SET_NUM		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
	case U_PASS_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
	case U_ROMSUM_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
	case U_END_NUM		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range																	
Input	stp_zipu_num->ul_mode_select_num	0x00000000~0xFFFFFFFF	0x00000000	0x00000001	0x00000002	0x00000003	0x00000005	0x00000006	0x00000007	0x00000009	0x0000000A	0x0000000B	0x0000000C	0x0000000D	0x00000010	0x00000013	0x00000014	0x00000016	0xFFFFFFFF
Output	stp_nl_disp->uc_lcd_com_dot	0x00~0x08	0x00	0x00	0x00	0x08	0x00	0x00	0x00	0x00	0x10	0x08	0x08	0x08	0x00	0x00	0x00	0x00	0x00
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_bump_limit.c

FuncBumpLimit_Chk_BumpLimitDay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○
	if(((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00) && (!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	○	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-	-
	if(lul_day == 0)		○	-	-	-	-
	else		-	○	-	-	-
	if(lus_ans > lul_day)			-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	29	29	3	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF
	FuncCalLimit_Calc_Cal_BumpLimitDay (&((stp_GasInfo_addr + luc_i)->uc_bump_date[0]), stp_data->us_bump_lmt_day)	0~1001	0	1	0	1000	1001
Output	lus_ans	0~1000	0	1	0	1000	1000
Judgment			OK	OK	OK	OK	OK

FuncBumpLimit_Chk_BumpLimitExpired

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(FuncBumpLimit_Chk_BumpLimitDay() == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	FuncBumpLimit_Chk_BumpLimitDay()	0~233	0	232	233
Output	ucs_bump_limit_flg	0,1	1	0	0
Judgment			OK	OK	OK

FuncBumpLimit_Get_BumpLimitExpiredBehavior

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_bump_lmt_chk	0	0
Judgment			OK

FuncBumpLimit_Get_BumpLimitOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_data->uc_bump_lmt_set == M_ON)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_data->uc_bump_lmt_set	0~2	0	1	2
Output	luc_nas	0,1	0	1	0
Judgment			OK	OK	OK

FuncBumpLimit_Get_ChkBumpLimitExpired

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_bump_limit_flg	0	0
Judgment			OK

FuncBumpLimit_Set_BumpEndDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_denst->uc_m_denst_minus_flg)		○	-	-	-
	else		-	○	○	○
	if(stp_denst->uc_m_over_flg)		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	stp_denst->uc_m_denst_minus_flg	0~2	1	0	0	2
	stp_denst->uc_m_over_flg	0~2	1	1	0	2
	stp_GasInfo_addr->ul_full_scale	0~100001	0	0	100000	100001
	stp_denst->ul_digit	1~501	1	1	500	501
	stp_denst->ul_m_denst	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	puc_gas_num	0~5	0	0	4	5
	pul_cal_denst	0~50001	0	0	50000	50001
Output	stp_GasInfo_addr->ul_bump_val	0,1	0	1	1	1
Judgment			OK	OK	OK	OK

FuncBumpLimit_Set_BumpCalFlagSuccess

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_ch	0~2	0	1	2
Output	ucs_bump_0d_cal_flg[puc_ch]	0,1	1	1	1
Judgment			OK	OK	OK

m_Chk_BumpLimitFlag

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○	○	○	○
	if(((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00) && (!((stp_denst+luc_i)-> uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	○	○	○	○	○	○	○
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)-> uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	○	○	○	-	○	○	○
	if(lul_day == 0)		○	○	○	○	○	-	○	-	-
	if(ucs_bump_0d_cal_flg[luc_i] == M_ON)		○	-	-	○	○	-	-	-	-
	else		○	○	○	○	○	-	○	-	-
	else if(lus_ans_day > lul_day)		-	-	-	-	-	○	-	○	-
	if(lus_ans_day == 0)		○	○	○	○	○	-	○	-	-
	else		-	-	-	-	-	○	-	○	○
	Symbol	Range									
Input	us_bump_lmt_day	0	0	0	0	0	0	0	0	0	0
	uc_state_trb	0,255	0	0	255	0	0	0	0	0	0
	FuncCalLimit_Calc_Cal_BumpLimitDay()	0~1000	0	0	0	0	0	1	0	999	1000
	uc_gas_comb	0~4	1	0	1	3	4	1	1	1	1
	uc_bump_date[0]	0	0	0	0	0	0	0	0	0	0
	ucs_bump_0d_cal_flg[0]	1,2	1	1	1	1	1	1	2	1	1
Output	luc_ans	0,1	0	0	0	0	0	1	0	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_cal_limit.c

FuncCalLimit_Calc_Cal_BumpLimitDay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_lmt_serial >= lul_nuw_serial)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	puc_deadline_date	0~100	0	99	100
	pus_limit_day	0~1001	0	1000	1001
	RTCTime_Chg_Day(luc_date[0])	0~36526	36525	0	36526
	RTCTime_Chg_Day(luc_date[1])	0~36526	0	36525	36526
Output	lul_ans_day	0~1001	0	1000	1001
Judgment			OK	OK	OK

FuncCalLimit_Chk_CalLimitDay

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○
	if(((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00) && (!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	○	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-	-
	if(lul_day == 0)		○	-	-	-	-
	else		-	○	-	-	-
	if(lus_ans > lul_day)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	29	29	3	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF
	FuncCalLimit_Calc_Cal_BumpLimitDay (&((stp_GasInfo_addr + luc_i)->uc_cal_date[0]), stp_data->us_cal_lmt_day)	0~1001	0	1	0	1000	1001
Output	lus_ans	0~1000	0	1	0	1000	1000
Judgment			OK	OK	OK	OK	OK

FuncCalLimit_Chk_CalLimitExpired

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(FuncCalLimit_Chk_CalLimitDay() == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	FuncCalLimit_Chk_CalLimitDay()	0~1001	0	1000	1001
Output	ucs_cal_limit_flg	0,1	1	0	0
Judgment			OK	OK	OK

FuncCalLimit_Get_CalLimitExpiredBehavior

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_cal_lmt_chk	0	0
Judgment			OK

FuncCalLimit_Get_CalLimitOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((stp_data->uc_cal_lmt_set == M_ON) && !((stp_data->uc_exp_flg == DSC_TGT_JAPAN)||(stp_data->uc_exp_flg == DSC_TGT_JG)))		-	○	-	-
	else		○	-	○	○
	Symbol	Range				
Input	stp_data->uc_cal_lmt_set	0~2	0	1	1	2
	stp_data->uc_exp_flg	0~4	0	1	3	4
Output	luc_nas	0,1	0	1	0	0
Judgment			OK	OK	OK	OK

FuncCalLimit_Get_ChkCalLimitExpired

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_cal_limit_flg	0	0
Judgment			OK

FuncCalLimit_Set_CalEndDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(stp_denst->uc_m_denst_minus_flg)		○	-	-	-
	else		-	○	○	○
	if(stp_denst->uc_m_over_flg)		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	(stp_denst->uc_m_denst_minus_flg	0~2	1	0	0	2
	stp_denst->uc_m_over_flg	0~2	1	1	0	2
	tp_GasInfo_addr->ul_full_scale	0~100001	0	0	100000	100001
	stp_denst->ul_digit	0~501	1	1	500	501
	stp_denst->ul_m_denst	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	puc_gas_num	0~5	0	0	4	5
Output	stp_GasInfo_addr->ul_cal_after	0,1	0	1	0	0
Judgment			OK	OK	OK	OK

FuncCalLimit_Set_CalStartDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/24
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(stp_denst->uc_m_denst_minus_flg)		○	-	-	-
	else		-	○	○	○
	if(stp_denst->uc_m_over_flg)		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	(stp_denst->uc_m_denst_minus_flg	0~2	1	0	0	2
	stp_denst->uc_m_over_flg	0~2	1	1	0	2
	stp_GasInfo_addr->ul_full_scale	0~100001	0	0	100000	100001
	stp_denst->ul_digit	1~501	1	1	500	501
	stp_denst->ul_m_denst	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	puc_gas_num	0~5	0	0	4	5
Output	stp_GasInfo_addr->ul_cal_before	0,1	0	1	1	1
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_chk_main.c

FuncChkMain_Chk_ConfirmEffectiveValue

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

		case1	case2	case3	case4
Root	if(FlowMain_Get_ModeStatus() != FlowMain_IRDA_MODE_NUMBER)	○	○	○	-
	if(stp_status->us_main_state & DataStructConst_FAIL_FRM)	○	-	-	-
	else	-	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	-	○	○	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)	-	○	-	-
	else	-	-	○	-
	if(luc_chk == M_NG)	-	○	-	-
	else	-	-	○	-
	else	-	-	-	○
	Symbol	Range			
Input	stp_status->us_main_state	0x0000~0xFFFF	0x0020	0x0000	0x0000	0xFFFF
	FlowMain_Get_ModeStatus()	0~8	0	0	8	7
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	30	0	0
Output	luc_flag	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

FuncChkMain_Chk_SensorTrouble

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if((stp_GasInfo_addr + 0)->uc_gas_comb != 0x00)	○	○	-
	if(GasCalcNC_Chk_SensorFail() == M_NG)	○	-	-
	else	-	○	-
	else	-	-	○
	Symbol	Range		
Input	(stp_GasInfo_addr+0)->uc_gas_comb	0~30	29	30	0
	GasCalcNC_Chk_SensorFail()	0~2	1	2	0
Output	None		-	-	-
Judgment			OK	OK	OK

FuncChkMain_Chk_ThermistorTrouble

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if(RL78I1EControl_Chk_TempError() == M_NG)	○	-	-
	else	-	○	○
	Symbol	Range		
Input	RL78I1EControl_Chk_TempError()	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

FuncChkMain_Int_1sec

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if(+ul_m_count >= M_24H_COUNT_MAX)	○	-	○
	else	-	○	-
	Symbol	Range		
Input	ul_m_count	0~86401	86400	0	86401
Output	ul_m_count	0,1	0	1	0
	uc_m_chk_on_flg	1	1	1	1
Judgment			OK	OK	OK

FuncChkMain_Set_BakUpBattFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

		case1	case2	case3	case4
Root	if(puc_flg == M_ON)	○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_BKB) == 0x00)	○	-	-	-
	else	-	○	-	-
	else	-	-	○	○
	Symbol	Range			
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0100	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0100	0xFFFF	0x0000	0xFEFF
Judgment			OK	OK	OK	OK

FuncChkMain_Set_BattFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(puc_flg == M_ON)		○	○	○	○	-	-
	if(Battery_Get_BatteryType() == BATTERY_BATT_TYPE_LI)		○	○	-	-	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_BTL) == 0x00)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_BTA) == 0x00)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		-	-	-	-	○	○
	Symbol	Range						
Input	puc_flg	0~2	1	1	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0000	0xFFFF	0x0002	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0002	0xFFFF	0x0002	0xFFFF	0x0000	0xFFFC
Judgment			OK	OK	OK	OK	OK	OK

FuncChkMain_Set_ChkStart

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_m_chk_on_flg == M_ON)		○	○	○	-	-
	if(luc_fram_write_flg == M_OFF)		○	○	-	-	-
	if(luc_result != M_OK)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	uc_m_chk_on_flg	0~2	1	1	1	0	2
	FRAM_Get_flg()	0~2	0	0	1	0	2
	DataFram_FRAM_Diag()	0~2	1	0	0	0	2
Output	uc_m_chk_on_flg	0,2	0	0	0	0	2
Judgment			OK	OK	OK	OK	OK

FuncChkMain_Set_CircuitVoltageFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_CLK) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0080	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0400	0xFFFF	0x0080	0xFBFF
Judgment			OK	OK	OK	OK

FuncChkMain_Set_ClockFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_CLK) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0080	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0080	0xFFFF	0x0000	0xFF7F
Judgment			OK	OK	OK	OK

FuncChkMain_Set_FlowFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_FRM) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0020	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0004	0xFFFF	0x0020	0xFFFB
Judgment			OK	OK	OK	OK

FuncChkMain_Set_FramFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_FRM) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0020	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0020	0xFFFF	0x0000	0xFFDF
Judgment			OK	OK	OK	OK

FuncChkMain_Set_PumpFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_FRM) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0008	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x4000	0xFFFF	0x0008	0xBFFF
Judgment			OK	OK	OK	OK

FuncChkMain_Set_RamFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_FRM) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0008	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0008	0xFFFF	0x0000	0xFFF7
Judgment			OK	OK	OK	OK

FuncChkMain_Set_SensorCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	13
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	if(puc_fail_num == 0)		○	○	○	-	-	-	-	-	-	-	-	-	-
	if(puc_flg == M_ON)		○	○	-	-	-	-	-	-	-	-	-	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (!(stp_denst->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-
	else if(puc_fail_num == 1)		-	-	-	○	○	○	-	-	-	-	-	-	-
	if(puc_flg == M_ON)		-	-	-	○	○	-	-	-	-	-	-	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (!(stp_denst->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		-	-	-	○	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-
	else if(puc_fail_num == 2)		-	-	-	-	-	-	○	○	○	-	-	-	-
	if(puc_flg == M_ON)		-	-	-	-	-	-	○	○	-	-	-	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (!(stp_denst->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-
	else if(puc_fail_num == 3)		-	-	-	-	-	-	-	-	-	○	○	○	-
	if(puc_flg == M_ON)		-	-	-	-	-	-	-	-	-	○	○	-	-
	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (!(stp_denst->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		-	-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range													
Input	puc_flg	0~2	1	1	0	1	1	0	1	1	0	1	1	0	2
	puc_gas	0~5	0	0	0	0	0	0	0	0	0	0	0	4	5
	puc_fail_num	0~4	0	0	0	1	1	1	2	2	2	3	3	3	4
	stp_GasInfo_addr->uc_gas_comb	0~30	1	0	0	1	0	0	1	0	0	1	0	0	30
	stp_denst->uc_state_trb	0x00~0xFF	0x00	0x14	0x00	0x00	0x14	0x00	0x00	0x14	0x00	0x00	0x14	0x20	0xFF
Output	stp_denst->uc_state_trb	0x00~0xFF	0x01	0x14	0x00	0x02	0x14	0x00	0x80	0x14	0x00	0x20	0x14	0x20	0xFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

FuncChkMain_Set_SensorSubstrateFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_SUB) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0xFFFF	0x0200	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0200	0xFFFF	0x0000	0xFDFF
Judgment			OK	OK	OK	OK

FuncChkMain_Set_SensorTroubleFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if((stp_GasInfo_addr->uc_gas_comb != 0x00) && (!(stp_denst->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_gas	0~5	0	0	4	5
	pus_code	99~141	100	130	141	99
	stp_GasInfo_addr->uc_gas_comb	0,29	29	29	0	30
	stp_denst->uc_state_trb	0x00,0xFF	0x00	0x00	0xFF	0xFF
	FlowMain_Get_ModeStatus()	0~8	1	0	0	8
Output	stp_denst->uc_state_trb	0x04~0xFF	0x04	0x10	0xFF	0xFF
Judgment			OK	OK	OK	OK

FuncChkMain_Set_ThermistorFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == M_ON)		○	○	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_TEP) == 0x00)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	1	1	0	2
	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0xFFFF	0x0800	0xFFFF
Output	stp_status->us_main_state	0x0000~0xFFFF	0x0800	0xFFFF	0x0000	0xF7FF
Judgment			OK	OK	OK	OK

m_Chk_Range1stAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_GasInfo_addr->ul_1st_alarm != M_OFF_VAL)		○	-	○	○
	if(puc_gas == 1)		○	-	-	-
	else		-	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	puc_gas	0~5	1	4	0	5
	m_Chk_RangeNum()	0~2	1	0	0	2
	stp_GasInfo_addr->ul_1st_alarm	0x00000000~0xFFFFFFFF	0	0xDDDDDDDD	0	0xFFFFFFFF
Output	luc_flg	0~2	1	0	0	2
Judgment			OK	OK	OK	OK

m_Chk_Range2ndAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_GasInfo_addr->ul_2nd_alarm != M_OFF_VAL)		○	-	○	○
	if(puc_gas == 1)		○	-	-	-
	else		-	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	puc_gas	0~5	1	0	0	5
	m_Chk_RangeNum()	0~2	1	0	0	2
	stp_GasInfo_addr->ul_2nd_alarm	0x00000000~0xFFFFFFFF	0	0xDDDDDDDD	0	0xFFFFFFFF
Output	luc_flg	0~2	1	0	0	2
Judgment			OK	OK	OK	OK

m_Chk_Range3rdAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_GasInfo_addr->ul_3rd_alarm != M_OFF_VAL)		○	-	○	○
	if(puc_gas == 1)		○	-	-	-
	else		-	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	puc_gas	0~5	1	0	0	5
	m_Chk_RangeNum()	0~2	1	0	0	2
	stp_GasInfo_addr->ul_3rd_alarm	0x00000000~0xFFFFFFFF	0	0xDDDDDDDD	0	0xFFFFFFFF
Output	luc_flg	0~2	1	0	0	2
Judgment			OK	OK	OK	OK

m_Chk_RangeAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_gas	0~5	0	4	5
	m_Chk_Range1ndAlarm()	0~2	0	1	2
	m_Chk_Range2ndAlarm()	0~2	0	1	2
	m_Chk_Range3rdAlarm()	0~2	0	1	2
	m_Chk_RangeStelAlarm()	0~2	0	1	2
	m_Chk_RangeTwaAlarm()	0~2	0	1	2
Output	luc_ans	0~2	0	1	2
Judgment			OK	OK	OK

m_Chk_RangeCalc

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_gas	0~5	0	4	5
	m_Chk_RangeFSCalc()	0~2	0	1	2
	m_Chk_RangeDigitCalc()	0~2	0	1	2
	m_Chk_RangeZeroCalc()	0~2	0	1	2
	m_Chk_RangeSpanCalc()	0~2	0	1	2
Output	luc_ans	0~2	0	1	2
Judgment			OK	OK	OK

m_Chk_RangeDigitCalc

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_gas != DataStructConst_IR_GAS_CH || stp_GasInfo_addr->uc_gas_comb != DataStructConst_IR_GAS_NUM)		○	○	-	○
	if(stp_GasInfo_addr->ul_digit != GasTableDefData[puc_gas][stp_GasInfo_addr->uc_gas_comb].ul_digit)		○	-	-	○
	else		-	○	-	-
	else		-	-	○	-
	Symbol	Range				
Input	puc_gas	0~5	0	0	4	5
	stp_GasInfo_addr->uc_gas_comb	0~30	29	0	1	30
	stp_GasInfo_addr->ul_digit	0~501	0	1	500	501
Output	luc_flg	0,1	1	0	1	1
Judgment			OK	OK	OK	OK

m_Chk_RangeFloat

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	if((pfl_chk_data < pfl_min) || (pfl_max < pfl_chk_data))		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pfl_chk_data	-69~343.3	-68.9	343.3	-69
	pfl_min	-9999~66	65	-9999	66
	pfl_max	-31~9999	-30.9	9999	-31
Output	luc_flg	0,1	1	0	1
Judgment			OK	OK	OK

m_Chk_RangeFSCalc

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_gas != DataStructConst_IR_GAS_CH || stp_GasInfo_addr->uc_gas_comb != DataStructConst_IR_GAS_NUM)		○	○	-	○
	if(stp_GasInfo_addr->ul_full_scale != GasTableDefData[puc_gas][stp_GasInfo_addr->uc_gas_comb].ul_full_scale)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	-
	Symbol	Range				
Input	puc_gas	0~5	0	0	4	5
	stp_GasInfo_addr->uc_gas_comb	0~29	29	0	1	30
	stp_GasInfo_addr->ul_full_scale	0~100000	0	100	100000	100001
Output	luc_flg	0,1	1	0	1	1
Judgment			OK	OK	OK	OK

m_Chk_RangeNum

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((pul_chk_data < pul_min) || (pul_max < pul_chk_data))		○	-	-
	else		-	○	○
	Symbol	Range			
Input	pul_chk_data	0x00000000~0xFFFFFFFF	0x00000001	0xFFFFFFFF	0x00000000
	pul_min	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0x00000001	0x00000000
	pul_max	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
Output	luc_flg	0,1	1	0	0
Judgment			OK	OK	OK

m_Chk_RangeSpanCalc

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_gas != DataStructConst_IR_GAS_CH || stp_GasInfo_addr->uc_gas_comb != DataStructConst_IR_GAS_NUM)		○	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	puc_gas	0~5	0	4	5	5
	m_Chk_RangeFloat()	0~2	1	0	2	2
	stp_GasInfo_addr->uc_gas_comb	0~30	29	1	30	0
Output	luc_flg	0~2	1	0	2	2
Judgment			OK	OK	OK	OK

m_Chk_RangeStelAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_GasInfo_addr->ul_stel_alarm != M_OFF_VAL)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	puc_gas	0~5	0	4	5
	m_Chk_RangeNum()	0~2	1	0	2
	stp_GasInfo_addr->ul_stel_alarm	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0xFFFFFFFF
Output	luc_flg	0~2	1	0	2
Judgment			OK	OK	OK

m_Chk_RangeTwaAlarm

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_GasInfo_addr->ul_twa_alarm != M_OFF_VAL)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	puc_gas	0~5	0	4	5
	m_Chk_RangeNum()	0,1	1	0	2
	stp_GasInfo_addr->ul_twa_alarm	0x00000000~0xFFFFFFFF	0	0xDDDDDDDD	0xFFFFFFFF
Output	luc_flg	0	1	0	2
Judgment			OK	OK	OK

m_Chk_RangeZeroCalc

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_gas != DataStructConst_IR_GAS_CH || stp_GasInfo_addr->uc_gas_comb != DataStructConst_IR_GAS_NUM)		○	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	puc_gas	0~5	0	4	5	5
	m_Chk_RangeFloat()	0~2	1	0	2	2
	stp_GasInfo_addr->uc_gas_comb	0~30	29	1	30	0
Output	luc_flg	0~2	1	0	2	2
Judgment			OK	OK	OK	OK

m_Set_FailCode

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pus_code	0~144	0	143	144
Output	stp_status->uc_state_error_code[0]	0x30,0x31	0x30	0x31	0x31
	stp_status->uc_state_error_code[1]	0x30~0x34	0x30	0x34	0x34
	stp_status->uc_state_error_code[2]	0x30~0x34	0x30	0x33	0x34
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_confirmation_beep.c

FuncConfirmationBeep_1sec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	○	○	-	-	-
	if(uss_beep_count == 0)		○	○	-	-	-	-
	if(stp_data->uc_beep_time == 0)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	○	○
	Symbol	Range						
Input	FlowMain_Get_ModeStatus()	0~8	1	1	1	0	7	8
	uss_beep_count	0~5941	0	0	5940	0	0	5941
	stp_data->uc_beep_time	0~100	0	99	0	0	0	100
Output	uss_beep_count	0~5941	30	5940	5939	0	0	5941
Judgment			OK	OK	OK	OK	OK	OK

FuncConfirmationBeep_Clr_ActiveFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case2
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_nci_active_flg != FUNC_BEEP_NCI_FLG_OFF)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_nci_active_flg	0~3	0	2	3
Output	(stp_GasInfo_addr+luc_i)->uc_nci_active_flg	0	0	0	0
Judgment			OK	OK	OK

FuncConfirmationBeep_Update_AlarmActiveFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((stp_data->uc_beep_select == M_BEEP_ALM) || (stp_data->uc_beep_select == M_BEEP_B_C_A))		○	○	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	○	-	-	-
	if((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		○	○	-	-	-	-
	if((stp_denst+luc_i)->uc_alarm_lat != 0x00)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	-	-
	if(luc_flg == M_ON)		○	-	-	-	-	-
	else		-	○	○	-	-	-
	else		-	-	-	-	○	○
	Symbol	Range						
Input	stp_data->uc_beep_select	0~7	6	6	6	6	0	7
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	3	3	29	0	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0xFF	0x00	0xFF
	(stp_denst+luc_i)->uc_alarm_lat	0x00~0xFF	0xFF	0x00	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

FuncConfirmationBeep_Update_BumpCalActiveFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((stp_GasInfo_addr+puc_ch)->uc_nci_active_flg != FUNC_BEEP_NCI_FLG_OFF)		○	○	-	○
	if(stp_data->uc_nci_download_flg == M_ON)		○	-	-	-
	else		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	(stp_GasInfo_addr+puc_ch)->uc_nci_active_flg	0~3	2	2	0	3
	stp_data->uc_nci_download_flg	0~2	1	0	0	2
	puc_ch	0~5	0	0	4	5
Output	(stp_GasInfo_addr+puc_ch)->uc_nci_active_flg	0~3	2	0	0	3
Judgment			OK	OK	OK	OK

FuncConfirmationBeep_Update_DownloadActiveFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_nci_active_flg == FUNC_BEEP_NCI_FLG_DL)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	(stp_GasInfo_addr+luc_i)->uc_nci_active_flg	0~3	2	0	3
Output	(stp_GasInfo_addr+luc_i)->uc_nci_active_flg	0~3	0	0	3
Judgment			OK	OK	OK

FuncConfirmationBeep_Update_LimitActiveFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if((stp_data->uc_beep_select == M_BEEP_B_C) || (stp_data->uc_beep_select == M_BEEP_B_C_A))		○	○	○	-	-
	if(stp_data->uc_bump_lmt_set == M_ON)		○	○	-	-	-
	if(FuncBumpLimit_Get_ChkBumpLimitExpired() == M_ON		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	if(FuncCalLimit_Get_CalLimitOnOff() == M_ON)		○	○	-	-	-
	if(FuncCalLimit_Get_ChkCalLimitExpired() == M_ON)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	if(luc_flg == M_ON)		○	-	-	-	-
	else		-	○	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	stp_data->uc_beep_select	0~7	6	6	6	0	7
	stp_data->uc_bump_lmt_set	0~2	1	1	0	0	2
	FuncBumpLimit_Get_ChkBumpLimitExpired()	0~2	1	0	0	0	2
	FuncCalLimit_Get_CalLimitOnOff()	0~2	1	1	0	0	2
	FuncCalLimit_Get_ChkCalLimitExpired()	0~2	1	0	0	0	2
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_Set_BeepEvent

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	switch(stp_data->uc_beep_select)		○	○	○	○	○	○	-	-	-
	case M_BEEP_LED		○	-	-	-	-	-	-	-	-
	case M_BEEP_BUZZ		-	○	-	-	-	-	-	-	-
	case M_BEEP_ALL		-	-	○	-	-	-	-	-	-
	case M_BEEP_B_C		-	-	-	○	-	-	-	-	-
	case M_BEEP_ALM		-	-	-	○	○	-	-	-	-
	case M_BEEP_B_C_A		-	-	-	○	○	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	-	○	○	○	○	-	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	-	-	○	○	○	-	-	-
	if((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	-	○	○	-	-	-	-
	if((stp_GasInfo_addr+luc_i)->uc_nci_active_flg != FUNC_BEEP_NCI_FLG_OFF)		-	-	-	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	○	-	-
	if(luc_flg == M_ON)		-	-	-	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-
	default		-	-	-	-	-	-	-	○	○
	Symbol	Range									
Input	stp_data->uc_beep_select	0~7	1	2	3	4	5	6	6	0	7
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	0	0	3	3	29	0	0	30
	(stp_GasInfo_addr+luc_i)->uc_nci_active_flg	0~3	0	0	0	2	0	0	0	0	3
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_NciActiveFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		○	○	-	○
	if((luc_i != DataStructConst_CO_CAN_H2_H2_CH) || ((stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM))		○	-	-	-
	else		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	3	29	0	30
Output	(stp_GasInfo_addr+luc_i)->uc_nci_active_flg	1	1	1	1	1
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_ec_sensor.c

ECSENSOR_Calc_AD_Main

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2
Root	if(uc_m_sub_ec_chk_flg == M_ON)		○	○
	Symbol	Range		
Input	stp_status->fl_ec_chk_raw_data[0]	0	0	0
	Ad12bit_Calc_Voltage()	0	0	0
	uc_m_sub_ec_chk_flg	1,2	1	2
	uc_m_ec_chk_data[2]	0	0	0
	uc_m_ec_chk_data[9]	0	0	0
Output	stp_status->fl_ec_chk_raw_data[0]	0	0	0
	stp_sub_ad->fl_ec_chk_data[0]	0	0	0
	uc_m_sub_ec_chk_flg	0,2	0	2
	fl_m_ec_chk_difference	0	0	0
Judgment			OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_ec_sensor_chk.c

FuncECSensorChk_10msec_Interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7
Root	for(luc_i=0; luc_i<2; luc_i++)	○	○	○	○	○	○	○
	if(uss_m_start_chk_count[luc_i])	-	-	○	-	-	○	○
	if(uss_m_start_chk_count[luc_i] == 0)	-	-	○	-	-	-	-
	else	-	-	-	-	-	○	○
	else	○	○	-	○	○	○	-
	for(luc_i=0; luc_i<3; luc_i++)	○	○	○	○	○	○	○
	if(uss_m_chk_count[luc_i])	○	○	○	○	-	○	○
	if(uss_m_chk_count[luc_i] == 1)	○	○	○	○	-	-	-
	if(luc_ans != 0x00)	○	○	○	-	-	-	-
	if(uss_m_chk_retry_count[luc_i])	-	○	○	-	-	-	-
	if(uss_m_chk_retry_count[luc_i] == 0)	-	○	-	-	-	-	-
	else	-	-	○	-	-	-	-
	if(luc_i == 0)	-	-	○	-	-	-	-
	else	-	-	○	-	-	-	-
	else	○	-	-	-	-	-	-
	else	-	-	-	○	-	-	-
	else	-	-	-	-	-	○	○
	else	-	-	-	-	○	-	-
	Symbol	Range						
Input	uss_m_start_chk_count[luc_i]	0~2601	0	0	1	0	0	2600	2601
	uss_m_chk_count[luc_i]	0~301	1	1	1	1	0	300	301
	m_Chk_AD()	0x00~0xFF	0x08	0x08	0x08	0x00	0x00	0x00	0xFF
	uss_m_chk_retry_count[luc_i]	0~4	0	1	2	3	3	0	4
Output	uss_m_chk_retry_count[0]	0~4	0	0	1	3	3	0	4
	uss_m_chk_wait[luc_i]	0	0	0	0	0	0	0	0
	uss_m_chk_retry_count[luc_i]	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

FuncECSensorChk_1sec_Interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

		case1	case2	case3	case4	case5	case6
Root	if(stp_sub_ad->fl_ec_ad_data[0] >= M_O2_CHK_START_AD)	○	○	○	○	-	○
	if((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))	○	○	○	-	-	-
	if(uss_m_chk_wait[0] == 0)	○	○	-	-	-	-
	if(uss_m_chk_retry_count[0] == 0)	○	-	-	-	-	-
	else	-	○	-	-	-	-
	else	-	-	○	-	-	-
	else	-	-	-	○	-	○
	else	-	-	-	-	○	-
	if(uss_m_chk_wait[0])	○	○	○	○		○
	else	-	-	-	-	○	-
	for(luc_i=1; luc_i<3; luc_i++)	○	○	○	○	○	○
	if(uss_m_chk_wait[luc_i])	-	-	○	○	-	○
	else	○	○	-	-	○	-
	if((ZIPCNumSet_Get_AlarmFlg() == M_OFF) && (FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER))	○	-	-	-	○	-
	if(uss_m_chk_retry_count[luc_i] == 0)	○	-	-	-	-	-
	else	-	-	-	-	○	-
	else	-	○	-	-	-	-
	Symbol	Range					
Input	stp_sub_ad->fl_ec_ad_data[0]	-800~801	800	800	800	800	-800	801
	uss_m_chk_wait[luc_i]	0~901	0	0	900	900	0	901
	uss_m_chk_retry_count[luc_i]	0~4	0	3	0	3	3	4
	FlowMain_Get_ModeStatus()	0~8	2	2	2	7	2	8
	ZIPCNumSet_Get_AlarmFlg()	0~2	0	1	1	1	0	2
Output	uss_m_chk_retry_count[luc_i]	0~4	3	3	0	3	3	4
	uss_m_chk_wait[luc_i]	0~900	0	0	899	899	0	900
Judgment			OK	OK	OK	OK	OK	OK

FuncECSensorChk_Set_StartChk

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_m_start_chk_count[0]	2600	2600
	uss_m_start_chk_count[1]	1100	1100
Judgment			OK

m_Chk_AD

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	13
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	if(puc_flg == 0)		○	-	○	○	-	-	-	-	-	○	○	-	○
	if(((stp_GasInfo_addr+1)->uc_gas_comb != 0x00) && (!((stp_denst+1)->uc_state_trb &(DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	-	○	○	-	-	-	-	-	○	○	-	○
	if(flc_m_chk_ad_min[0] > stp_sub_ad->fl_ec_chk_data[0])		○	-	○	○	-	-	-	-	-	○	○	-	-
	else if(puc_flg == 1)		-	○	-	-	-	○	○	○	○	-	-	○	-
	if(((stp_GasInfo_addr+2)->uc_gas_comb != 0x00) && (!((stp_denst+2)->uc_state_trb &(DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		-	○	-	-	-	-	-	-	-	-	-	-	-
	if((stp_GasInfo_addr+2)->uc_gas_comb == DataStructConst_SINGLE_H2S_GAS_NUM)		-	-	-	-	-	○	○	○	○	-	-	○	-
	else		-	○	-	-	-	○	○	○	○	-	-	○	-
	if(flc_m_chk_ad_min[luc_flg] > stp_sub_ad->fl_ec_chk_data[luc_flg])		-	○	-	-	-					-	-	-	-
	if(((stp_GasInfo_addr+3)->uc_gas_comb != 0x00) && (!((stp_denst+3)->uc_state_trb &(DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		-	-	-	-	-	-	-	-	-	-	-	-	-
	if(flc_m_chk_ad_min[1] > stp_sub_ad->fl_ec_chk_data[1])		-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-
	Symbol	Range													
Input	puc_flg	0~2	0	1	0	0	2	1	1	1	1	0	0	1	0
	uc_gas_comb	0~29	1	1	0	1	1	0	1	2	3	29	0	0	29
	uc_state_trb	0~255	0	0	0	255	0	0	255	0	0	0	255	0	0
	fl_ec_chk_data[0]	-800~800	0	0	0	0	0	0	0	0	0	0	-800	0	800
	fl_ec_chk_data[1]	0~800	0	0	0	0	0	0	0	0	0	0	0	800	0
Output	luc_ans	0~2	1	2	1	1	0	2	2	2	2	1	1	2	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_ECSensorChk

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(puc_flg)		○	○	○	○
	case 0		○	-	-	-
	case 1		-	○	-	-
	case 2		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	puc_flg	0~3	0	1	2	3
Output	uss_m_chk_count[0]	300	300	300	300	300
	uss_m_chk_count[1]	0,300	0	300	300	300
Judgment			OK	OK	OK	OK

m_Set_Err

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_H2_CH)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	-	-	-
	if(puc_flg & (0x01<<(DataStructConst_CO_CAN_H2_H2_CH - DataStructConst_O2_GAS_CH)))		-	-	-	-	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_CO_CH)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	-	-	-
	if(puc_flg & (0x01<<(DataStructConst_CO_CAN_H2_CO_CH - DataStructConst_O2_GAS_CH)))		-	-	-	-	-
	for(luc_i=DataStructConst_O2_GAS_CH; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○
	if(puc_flg & (0x01<<(luc_i-DataStructConst_O2_GAS_CH)))		-	○	○	○	-
	if(FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		-	-	○	○	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	-	○	○	-
	switch(luc_i)		-	-	○	○	-
	case 1		-	-	○	○	-
	case 2		-	-	-	-	-
	case 3		-	-	-	-	-
	default		-	-	-	-	-
	Symbol	Range					
Input	puc_flg	0x00~0xFF	0x08	0x02	0x01	0xFF	0x00
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	3	3	0	29	30
	FlowMain_Get_ModeStatus()	0~8	0	7	0	0	8
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_Set_StartInitialChk

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == 0)		○	-	-	-
	else		-	○	○	○
	if((stp_GasInfo_addr + 4)->uc_gas_comb != DataStructConst_IR_GAS_NUM)		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	puc_flg	0~2	0	1	1	2
	(stp_GasInfo_addr + 4)->uc_gas_comb	0~29	0	29	1	2
Output	uss_m_chk_retry_count[0]	3	3	3	3	3
	uss_m_chk_retry_count[1]	0,3	0	3	3	3
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_lunch_break.c

FuncLunchBreak_Get_LunchBreakFramData

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	if((luc_ack & 0x0f) && (luc_ack & 0xf0))	○	○	○
	if((luc_ack & 0x0f) && (luc_ack & 0xf0))	-	-	○
	else	-	-	-
	else	○	○	-
	Symbol	Range		
Input	DataFram_LunchBreak_Load	0x00~0xFF	0xF0	0x0F	0xFF
Output	ucs_m_lunch_status_flg	0,1	0	0	1
	luc_ack	0x0F~0xFF	0xF0	0x0F	0xFF
Judgment			OK	OK	OK

FuncLunchBreak_Get_LunchBreakOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

		case1
Root	None	-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_lunch_onoff	0	0
Judgment			OK

FuncLunchBreak_Get_StatusFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

		case1
Root	None	-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_lunch_status_flg	0	0
Judgment			OK

FuncLunchBreak_Set_LunchBreakData

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if((ucs_m_resume_onoff == M_ON) && (ucs_m_lunch_status_flg == M_OK))		○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	-
	for(luc_j=0; luc_j<6; luc_j++)		○	-	-
	else		-	○	○
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	○	○
	Symbol	Range			
Input	ucs_m_resume_onoff	0~2	1	0	2
	ucs_m_lunch_status_flg	0~2	0	1	2
	stp_lb_data->st_param.st_each[luc_i].ul_pmin	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_lb_data->st_param.st_each[luc_i].ul_pmax	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_lb_data->st_param.st_each[luc_i].uc_pmin_flg	0~2	1	0	2
	stp_lb_data->st_param.st_each[luc_i].uc_pmax_flg	0~2	1	0	2
	stp_lb_data->st_param.st_each[luc_i].ul_60_total	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_lb_data->st_param.st_each[luc_i].ul_ave	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_lb_data->st_param.st_each[luc_i].ul_ave_total_cnt	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	stp_lb_data->st_param.st_each[luc_i].ul_twa	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	(stp_denst+luc_i)->ul_pmin	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
	(stp_denst+luc_i)->ul_pmax	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
	(stp_denst+luc_i)->uc_pmin_flg	1	1	1	1
	(stp_denst+luc_i)->uc_pmax_flg	1	1	1	1
	(stp_denst+luc_i)->uc_pmin_tim[luc_j]	99	99	99	99
	(stp_denst+luc_i)->uc_pmax_tim[luc_j]	99	99	99	99
	(stp_denst+luc_i)->ul_60_total	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
	(stp_denst+luc_i)->ul_ave	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
	(stp_denst+luc_i)->ul_ave_total_cnt	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
	(stp_denst+luc_i)->ul_twa	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
Judgment			OK	OK	OK

FuncLunchBreak_Set_LunchBreakFramData

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((stp_status->us_main_state & (DataStructConst_FAIL_FRM | DataStructConst_FAIL_ROM | DataStructConst_FAIL_RAM)) == 0)		○	○	-	-
	if((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))		○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	-	-
	for(luc_j=0; luc_j<6; luc_j++)		-	-	-	-
	else		-	○		-
	else		-	-	○	○
	Symbol	Range				
Input	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF
	FlowMain_Get_ModeStatus()	0~8	1	0	7	8
	(stp_denst+luc_i)->ul_pmin	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+luc_i)->ul_pmax	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+luc_i)->uc_pmin_flg	0~2	0	1	0	2
	(stp_denst+luc_i)->uc_pmax_flg	0~2	0	1	0	2
	(stp_denst+luc_i)->uc_pmin_tim[luc_j]	0~100	0	99	0	100
	(stp_denst+luc_i)->uc_pmax_tim[luc_j]	0~100	0	99	0	100
	(stp_denst+luc_i)->ul_60_total	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+luc_i)->ul_ave	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+luc_i)->ul_ave_total_cnt	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+luc_i)->ul_twa	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	luc_status	0,1	0	1	1	1
	stp_lb_data->st_param.st_each[luc_i].ul_pmin	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].ul_pmax	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].uc_pmin_flg	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].uc_pmax_flg	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i]. uc_pmin_tim[luc_j]	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i]. uc_pmax_tim[luc_j]	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].ul_60_total	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].ul_ave	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].ul_ave_total_cnt	0	0	0	0	0
	stp_lb_data->st_param.st_each[luc_i].ul_twa	0	0	0	0	0
Judgment			OK	OK	OK	OK

FuncLunchBreak_Set_OnOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	0	1	2
Output	ucs_m_resume_onoff	0~2	0	1	2
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_mainte_notice.c

FuncMainteNotice_Chk_MainteLimitDay

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<lHDDEF_GAS_MAX; luc_i++)		○	○	○	○	○
	if(((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00) && (!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	○	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	-	-	-
	if(lul_day == 0)		○	-	-	-	-
	else		-	○	-	-	-
	if(lus_ans > lul_day)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	29	29	3	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0xFF	0xFF
	m_Calc_MainteLimitDay(&((stp_GasInfo_addr + luc_i)->uc_notice_date[0]), stp_data->us_mainte_notice_day)	0~1001	0	1	0	1000	1001
Output	lus_ans	0~1000	0	1	0	1000	1000
Judgment			OK	OK	OK	OK	OK

FuncMainteNotice_Chk_MainteNoticeLimitExpired

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(FuncMainteNotice_Chk_MainteLimitDay() == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	FuncMainteNotice_Chk_MainteLimitDay()	0~1001	0	1000	1001
Output	ucs_mainte_limit_flg	0,1	1	0	0
Judgment			OK	OK	OK

FuncMainteNotice_Get_MainteLimitExpiredBehavior

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_mainte_lmt_chk	0	0
Judgment			OK

FuncMainteNotice_Get_MainteNoticeLimitExpired

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_mainte_limit_flg	0	0
Judgment			OK

FuncMainteNotice_Get_MainteNoticeOnOff

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_data->uc_mainte_notice_onoff == M_ON) && !((stp_data->uc_exp_flg == DSC_TGT_JAPAN)||(stp_data->uc_exp_flg == DSC_TGT_JG)))		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_data->uc_mainte_notice_onoff	0~2	0	1	2
	stp_data->uc_exp_flg	0~4	0	3	4
Output	luc_nas	0,1	0	1	0
Judgment			OK	OK	OK

FuncMainteNotice_Set_MainteNoticeReset

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	stp_GasInfo_addr->uc_notice_date[0]	0	0
	stp_GasInfo_addr->uc_notice_date[1]	0	0
	stp_GasInfo_addr->uc_notice_date[2]	0	0
Judgment			OK

FuncMainteNotice_Set_MainteNoticeStart

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr->uc_notice_date[0] == 0) && (stp_GasInfo_addr->uc_notice_date[1] == 0) && (stp_GasInfo_addr->uc_notice_date[2] == 0))		○	-	-	-
	else		-	○	○	○
	if(luc_flg == M_ON)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	stp_GasInfo_addr->uc_notice_date[0]	0~100	0	99	99	100
	stp_GasInfo_addr->uc_notice_date[1]	1~13	0	1	12	13
	stp_GasInfo_addr->uc_notice_date[2]	1~32	0	1	31	32
Output	stp_GasInfo_addr->uc_notice_date[0]	99~204	204	99	99	100
	stp_GasInfo_addr->uc_notice_date[1]	1~204	204	1	12	13
	stp_GasInfo_addr->uc_notice_date[2]	31~204	204	100	31	32
Judgment			OK	OK	OK	OK

FuncMainteNotice_Set_MainteNoticeUpDate

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_gas_num >= HDDEF_GAS_MAX)		-	-	○
	else		○	○	-
	Symbol	Range			
Input	puc_gas_num	0~5	0	4	5
Output	stp_GasInfo_addr->uc_notice_date[0]	0	0	0	0
	stp_GasInfo_addr->uc_notice_date[1]	0	0	0	0
	stp_GasInfo_addr->uc_notice_date[2]	0	0	0	0
Judgment			OK	OK	OK

m_Calc_MainteLimitDay

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_lmt_serial >= lul_nuw_serial)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	puc_deadline_date	0~100	0	99	100
	pus_limit_day	0~1001	0	1000	1001
	RTCTime_Chg_Day(luc_date[0])	0~36526	36525	0	36526
	RTCTime_Chg_Day(luc_date[1])	0~36526	0	36525	36526
Output	lul_ans_day	0~1001	0	1000	1001
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_margin_chk.c

FuncMarginChk_Calc_SpanMargine

	Responsible	Last update
Coding	Ikarashi	2021/6/25
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if((stp_GasInfo_addr + puc_gas_num)->uc_gas_comb != 0x00)		○	○	○	○	○	○	○	○	○	-
	switch(puc_gas_num)		○	○	○	○	○	○	○	○	○	-
	case 0		○	-	-	-	-	-	-	-	-	-
	case 1		-	○	-	-	-	-	-	-	-	-
	case 2		-	-	○	○	-	-	-	-	-	-
	if((stp_GasInfo_addr+2)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	○	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-
	case 3		-	-	-	-	○	○	-	-	-	-
	if((stp_GasInfo_addr+3)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-
	case 4		-	-	-	-	-	-	○	○	-	-
	if((stp_GasInfo_addr+4)->uc_gas_comb == DataStructConst_IR_GAS_NUM)		-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	○	-	-
	default		-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	○
	Symbol	Range										
Input	puc_gas_num	0~5	0	1	2	2	3	3	4	4	5	0
	(stp_GasInfo_addr + puc_gas_num)->uc_gas_comb	0~30	29	29	29	29	29	29	29	29	30	0
	(stp_GasInfo_addr+2)->uc_gas_comb	0~30	0	0	3	29	0	0	0	0	0	30
	(stp_GasInfo_addr+3)->uc_gas_comb	0~30	0	0	0	0	3	29	0	0	0	30
	(stp_GasInfo_addr+4)->uc_gas_comb	0~30	0	0	0	0	0	0	1	29	0	30
Output	None	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_o2_warm_up.c

O2WarmUp_Func_main

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(lfl_temp_ad < M_TEMP_M_20c_mV)		○	-	-	-	-	-	-	-
	else if(lfl_temp_ad < M_TEMP_P_35c_mV)		-	○	○	-	-	-	-	-
	else if(lfl_temp_ad < M_TEMP_P_45c_mV)		-	-	-	○	○	-	-	-
	else if(lfl_temp_ad < M_TEMP_P_55c_mV)		-	-	-	-	-	○	○	-
	else		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	Ad12bit_Calc_Voltage()	0	0	0	0	0	0	0	0	0
	stp_sub_ad->fl_temp_ad_data	0~249.08	192.37	192.38	233.95	233.96	241.51	241.52	249.07	249.08
	uss_m_ad_num	0	0	0	0	0	0	0	0	0
Output	fls_m_ad_data	0	0	0	0	0	0	0	0	0
	uc_m_temp_get_flg	0~4	0	1	1	2	2	3	3	4
	ucs_m_init_flg	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

O2WarmUp_Get_AD_data

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	fls_m_ad_data	0,255	0	255
Output	fls_m_ad_data	0,255	0	255
Judgment			OK	OK

O2WarmUp_Get_WarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	ucs_m_init_flg	0~2	0	1	2
Output	ucs_m_init_flg	0~2	0	1	2
Judgment			OK	OK	OK

O2WarmUp_Interval_10msec

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(ucs_m_init_flg != M_OFF)		○	-	○	○	○	○	○
	if(ucs_m_port_off_timer != 0)		○	-	-	○	○	○	○
	if(ucs_m_port_off_timer == 0)		○	-	-	-	○	○	○
	if(ucs_m_port_on_timer != 0)		○	-	○	○	-	○	○
	if(ucs_m_port_on_timer == 0)		○		○	○	-	-	○
	if(us_count_end_time != 0)		○	-	○	○	○	○	-
	Symbol	Range							
Input	ucs_m_init_flg	0,1	1	0	1	1	1	1	1
	ucs_m_port_off_timer	0~2	1	1	0	2	1	1	1
	ucs_m_port_on_timer	0~2	1	1	1	1	0	2	1
	us_count_end_time	0,1	1	1	1	1	1	1	0
Output	ucs_m_port_off_timer	0~6	6	1	6	6	0	0	6
	ucs_m_port_on_timer	0,1	0	1	0	0	0	1	0
	us_count_end_time	0,1	0	1	0	0	0	0	0
	ucs_m_approval_ad_flg	0,1	0	0	0	0	1	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

O2WarmUp_Set_AD

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ucs_m_approval_ad_flg == M_ON)		○	-	○
	if(pus_data >= db_m_Threshold[uc_m_temp_get_flg])		-	-	-
	else		○	-	○
	if(++ucs_m_chk_count >= 2)		○	-	○
	if((ucs_m_app_count >= uc_m_Chk_Cmt[uc_m_temp_get_flg]) ‖ (us_count_end_time == 0))		○	-	-
	else		-	-	○
	Symbol	Range			
Input	pus_data	0	0	0	0
	uc_m_temp_get_flg	0	0	0	0
	ucs_m_app_count	0	0	0	0
	ucs_m_approval_ad_flg	1,2	1	2	1
	ucs_m_chk_count	0	0	0	0
	us_count_end_time	0,1	0	0	1
Output	ucs_m_app_count	0	0	0	0
	ucs_approval_ad_flg	0~2	0	2	0
	ucs_m_chk_count	0	0	0	0
	ucs_m_init_flg	0	0	0	0
	ucs_m_port_off_timer	0	0	0	0
	ucs_m_port_on_timer	0,144	144	0	0
	uss_m_ad_num	0	0	0	0
Judgment			OK	OK	OK

O2WarmUp_Set_Init

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	-	-	-
Output	ucs_m_approval_ad_flg	1	1
	ucs_m_init_flg	1	1
	ucs_m_port_off_timer	0	0
	ucs_m_port_on_timer	0	0
Judgment			OK

O2WarmUp_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	-	-	-
Output	fls_m_ad_data	0	0
	uc_m_temp_get_flg	0	0
	ucs_m_app_count	0	0
	ucs_m_approval_ad_flg	0	0
	ucs_m_chk_count	0	0
	ucs_m_init_flg	0	0
	ucs_m_port_off_timer	0	0
	ucs_m_port_on_timer	0	1
	us_count_end_time	2000	2000
	uss_m_ad_num	0	0
Judgment			OK

m_O2WarmUp_Port_Term

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(PORTA.PODR.BIT.B2 == 0U)		○	-
	Symbol	Range		
Input	PORTA.PODR.BIT.B2	-	0	1
Output	PORTA.PODR.BIT.B2	0	1	1
Judgment			OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_ram_chk.c

f_Check_CopyRam

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lus_num == 0U)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	memcmp()	-1~1	-1	0	1
Output	len_ans	0~2	1	0	1
Judgment			OK	OK	OK

f_Check_MainRam

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(lus_loop=0U; lus_loop<F_CHECK_RAM_BUF_SIZE; lus_loop++)		○	○	○	○
	if((fuc_num[0] != F_55_BIT) || (fuc_num[1] != F_AA_BIT))		-	-	-	-
	break		-	-	-	-
	else		○	○	○	○
	if(ful_check_main_address >= lulc_end_address[pen_area])		○	○	○	○
	if(len_ans == FEN_RAMCHECK_ANS_SUB)		-	○	-	○
	else		○	-	○	-
	else		-	-	-	-
	Symbol	Range				
Input	pen_area	0,1	0	0	1	1
	ful_check_main_address	0x00000004~0x0000FFE0	0x00000004	0x00008660	0x0000EB00	0x0000FFE0
Output	fucp_check_point	0	0	0	0	0
	len_ans	0,2	2	0	2	2
Judgment			OK	OK	OK	OK

f_Cyclic_CheckProcess

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	13
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	switch(fen_phase_flag)		○	○	○	○	○	○	○	○	○	○	○	○	○
	case FEN_RAMCHECK_PHASE_START:		○	-	-	-	-	-	-	-	-	-	-	-	-
	case FEN_RAMCHECK_PHASE_SUB:		-	○	○	-	-	-	-	-	-	-	-	-	-
	if(len_chk == FEN_RAMCHECK_ANS_PASS)		-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-
	case FEN_RAMCHECK_PHASE_MAIN:		-	-	-	○	○	○	-	-	-	-	-	-	-
	if(len_chk == FEN_RAMCHECK_ANS_PASS)		-	-	-	○	-	-	-	-	-	-	-	-	-
	else if(len_chk != FEN_RAMCHECK_ANS_SUB)		-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-
	case FEN_RAMCHECK_PHASE_STACK:		-	-	-	-	-	-	○	○	○	-	-	-	-
	if(len_chk == FEN_RAMCHECK_ANS_PASS)		-	-	-	-	-	-	○	-	-	-	-	-	-
	else if(len_chk != FEN_RAMCHECK_ANS_SUB)		-	-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-
	case FEN_RAMCHECK_PHASE_PASS:		-	-	-	-	-	-	-	-	-	○	-	-	-
	case FEN_RAMCHECK_PHASE_FAIL:		-	-	-	-	-	-	-	-	-	-	○	○	-
	if(fuc_err_count >= F_ERR_RETRY_MAX)		-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	○	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range													
Input	fen_phase_flag	0~7	1	2	2	3	3	3	4	4	4	5	6	6	7
	f_Check_CopyRam()	0,2	0	0	2	0	0	0	0	0	0	0	0	0	0
	f_Check_MainRam()	0~3	0	0	0	0	2	1	0	2	1	0	3	0	0
	fuc_err_count	0,3	0	0	0	0	0	0	0	0	0	0	3	0	0
Output	fuc_err_count	0,1	0	0	0	0	0	0	0	0	0	0	1	1	0
	fen_phase_flag	0~6	2	3	6	4	3	6	5	4	6	0	2	2	0
	len_ans	0,1	1	1	1	1	1	1	1	1	1	2	1	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

f_Init_MainRam

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root			-	-
	Symbol	Range		
Input	pen_area	0,1	0	1
Output	ful_check_main_address	0x00000004,0x0000EB00	0x00000004	0x0000EB00
Judgment			OK	OK

FuncRamChk_Cyclic_100msecProcess

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(len_result == EN_RAMCHECK_RESULT_FAIL)		-	○
	else		○	-
	Symbol	Range		
Input	f_Cyclic_CheckProcess()	0~3	0	3
Output	None	-	-	-
Judgment			OK	OK

FuncRamChk_Set_RamFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_flg == M_OK)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_flg	0~2	0	1	2
Output	stp_status->us_main_state	0x0000,0x0008	0x0000	0x0008	0x0008
Judgment			OK	OK	OK

FuncRamChk_Start_Check

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(fen_phase_flag == FEN_RAMCHECK_PHASE_OFF)		○	-
	else		-	○
	Symbol	Range		
Input	fen_phase_flag	0,6	0	6
Output	fen_phase_flag	1,6	1	6
	fuc_err_count	0	0	0
Judgment			OK	OK

FuncRamChk_Start_InitRamChk

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(lul_num=0U; lul_num<F_INIT_LOOP_MAX; lul_num++)		○	○	○	○	○
	if((len_result == EN_RAMCHECK_RESULT_FAIL) || (len_result == EN_RAMCHECK_RESULT_PASS))		○	○	○	-	○
	else		-	-	-	○	-
	if((len_result == EN_RAMCHECK_RESULT_FAIL) || (len_result == EN_RAMCHECK_RESULT_PASS))		-	-	-	○	-
	else		○	○	○	-	○
	Symbol	Range					
Input	f_Cyclic_CheckProcess()	0~4	0	1	2	3	4
Output	luc_ans	0,1	0	0	0	1	0
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_rom_chk.c

FuncRomChk_Chk_EndRomChk

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ucs_m_rom_chk_phase == M_ROM_CHK_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ucs_m_rom_chk_phase	0~4	0	3	4
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

FuncRomChk_Get_SumNum

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_m_sum_num	0	0
Judgment			OK

FuncRomChk_process_RomChk

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(ucs_m_rom_chk_phase != M_ROM_CHK_OFF)		○	○	○	○	-	○
	if(ucs_m_rom_chk_phase != M_ROM_CHK_END)		○	○	-	-	-	○
	if(ucs_m_sum_area_phase == M_AREA_START)		○	-	-	-	-	-
	else		-	○	-	-	-	○
	if(ucs_m_sum_area_phase == M_AREA_END)		-	○	-	-	-	-
	else		○	-	-	-	-	○
	else		-	-	○	○	-	-
	if(uss_m_sum_num != MAIN_Get_SumNum())		-	-	○	-	-	-
	else		-	-	-	○	-	-
	else		-	-	-	-	○	-
	Symbol	Range						
Input	ucs_m_rom_chk_phase	0~4	1	1	3	3	0	4
	ucs_m_sum_area_phase	0~17	0	15	0	0	16	17
	uss_m_sum_num	0x0000,0xFFFF	0x0000	0x0000	0xFFFF	0x0000	0x0000	0xFFFF
Output	ucs_m_sum_area_phase	0~18	1	16	0	0	16	18
	uss_m_sum_num	0x0000,0xFFFF	0x0000	0x0000	0xFFFF	0x0000	0x0000	0x0000
	ucs_m_rom_chk_phase	0~4	2	3	0	0	0	4
	stp_status->us_main_state	0x0000,0x0010	0x0000	0x0000	0x0010	0x0000	0x0000	0x0000
Judgment			OK	OK	OK	OK	OK	OK

FuncRomChk_Start_InitRomChk

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FuncRomChk_Start_RomChk

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_rom_chk_phase	1	1
	ucs_m_sum_area_phase	0	0
Judgment			OK

m_Calc_RomSum

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	while(lsl_now_address >= 0)		○	○	○
	Symbol	Range			
Input	&puc_start[0]	0x00,0xFF	0x00	0x00	0xFF
	ppuc_end	0x00,0xFF	0x00	0xFF	0xFF
	pus_base	0x0000~0xFFFF	0x0000	0xFFFF	0xFF
Output	pus_base	-	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_sdm.c

FuncSDM_Get_CombData

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~5	0	4	5
Output	ucs_m_sdm_disp_comb[puc_num]	0	0	0	0
Judgment			OK	OK	OK

FuncSDM_Get_ExCommandFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_sdm_ex_command_flg	0	0
Judgment			OK

FuncSDM_Set_DispData

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(*puc_data_p == '0')		○	-	-
	for(luc_i=0; luc_i<5; luc_i++)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	&ucs_m_sdm_disp_data[0]	0x00,0xFF	0x30	0x00	0xFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

FuncSDM_Set_ExCommandData

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

FuncSDM_Set_ExCommandFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_onoff	0~2	0	1	2
Output	ucs_m_sdm_ex_command_flg	0,1	0	1	0
Judgment			OK	OK	OK

m_Disp_MainteData

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_data	0x00~0xFF	0x00	0xFF
Output	None	-	-	-
Judgment			OK	OK

m_Disp_MeasData

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○	○	○
	if(luc_unti == 0)		○	○	○	○
	else		-	-	-	-
	switch(luc_i)		○	○	○	○
	case 0		○	○	○	○
	case 1		○	○	○	○
	case 2		○	○	○	○
	case 3		○	○	○	○
	default		-	-	-	-
	if(luc_mes_dot != 0)		○	○	○	○
	else		-	-	-	-
	if(*puc_data == '1')		○	○	○	-
	else		-	-	-	○
	Symbol	Range				
Input	puc_data[3]	48,49	48	49	48	48
	puc_data[4]	48,49	49	49	48	48
	puc_data[11]	49,50	49	49	50	50
Output	stp_nl_disp->uc_lcd_mes_dot[luc_y]	0,32	32	32	0	0
	stp_nl_disp->uc_lcd_mes_blink[luc_y]	0,240	240	240	240	0
	stp_nl_disp->uc_lcd_mes_dot_blink[luc_y]	0,224	224	224	224	0
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	func_temp_warning.c

FuncTempWarning_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(uls_out_temp_timer < M_TEMP_COUNT_MAX)		○	○	○	-
	else		-	-	-	○
	if(uls_out_temp_reset_timer < M_TEMP_COUNT_MAX)		○	○	○	-
	else		-	-	-	○
	if(uls_out_buzzer_timer < M_TEMP_COUNT_MAX)		○	○	○	-
	else		-	-	-	○
	if(ucs_out_buzzer_flg == M_ON)		○	○	-	-
	if(++uls_buzzer_count_timer >= M_BUZZER_COUNT)		-	○	-	-
	else		○	-	-	-
	else		-	-	○	○
	Symbol	Range				
Input	uls_out_temp_timer	0~720001	0	720000	720000	720001
	uls_out_temp_reset_timer	0~720001	0	720000	720000	720001
	uls_out_buzzer_timer	0~720001	0	720000	720000	720001
	ucs_out_buzzer_flg	0~2	1	1	0	2
	uls_buzzer_count_timer	0~500	0	499	499	500
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

FuncTempWarning_Chk_OutOfRange

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(ucs_out_range_flg == M_ON)		○	-	-	-	-	-
	if((M_TEMP_RESET_MAX <= stp_sub_ad->fl_disp_temp) || (M_TEMP_RESET_MIN >= stp_sub_ad->fl_disp_temp))		○	-	-	-	-	-
	else		-	○	-	-	-	-
	if(uls_out_temp_reset_timer >= M_TEMP_RESET_COUNT)		-	○	-	-	-	-
	else		○	-	-	-	-	-
	else		-	-	○	○	○	○
	if((M_TEMP_MAX >= stp_sub_ad->fl_disp_temp) && (M_TEMP_MIN <= stp_sub_ad->fl_disp_temp))		-	-	○	-	-	-
	else		-	-	-	○	○	○
	if(uls_out_temp_timer >= M_TEMP_COUNT)		-	-	-	○	○	○
	if((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))		-	-	-	○	-	-
	else		-	-	-	-	○	○
	else		-	-	○	-	-	-
	if(ucs_out_range_flg == M_ON)		○	-	-	○	-	-
	if((uls_out_buzzer_timer >= M_TEMP_COUNT) || (uls_out_temp_timer > M_BUZZER_RESET_COUNT_MAX))		-	-	-		-	-
	else		○	-	-	-	-	-
	else		-	○	○	-	○	○
	Symbol	Range						
Input	ucs_out_range_flg	0~2	1	1	0	0	0	2
	stp_sub_ad->fl_disp_temp	-55~86	-55	0	0	85	85	86
	uls_out_temp_reset_timer	0~720001	0	720000	0	0	720000	720001
	FlowMain_Get_ModeStatus()	0~8	0	0	0	1	7	8
	uls_out_temp_timer	0~720001	0	0	0	720000	720000	720001
	uls_out_buzzer_timer	0~720001	0	0	0	0	720000	720001
Output	uls_out_temp_reset_timer	0~720001	0	0	0	0	720000	720001
	ucs_out_range_flg	0~2	1	0	0	1	0	2
	uls_out_buzzer_timer	0	0	0	0	0	0	0
	ucs_out_buzzer_flg	0,1	0	0	0	1	0	0
	uls_out_temp_timer	0~720001	0	0	0	720000	720000	720001
Judgment			OK	OK	OK	OK	OK	OK

FuncTempWarning_Get_TempBuzzerFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_out_buzzer_flg	2	2
Judgment			OK

FuncTempWarning_Get_OutOfRangeFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((ucs_out_range_flg == M_ON) && (uls_out_temp_timer >= M_BUZZER_RESET_COUNT_MAX))		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ucs_out_range_flg	0~2	1	0	2
	uls_out_temp_timer	0~720001	720000	0	720001
Output	luc_ans	0,2	2	0	2
Judgment			OK	OK	OK

FuncTempWarning_Reset_OutOfRangeBuzzer

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((ucs_out_buzzer_flg == M_ON) && (uls_out_temp_timer < M_BUZZER_RESET_COUNT_MAX))		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ucs_out_buzzer_flg	0~2	1	0	2
	uls_out_temp_timer	0~720001	0	720000	720001
Output	ucs_out_buzzer_flg	0,2	0	0	2
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc.c

GasCalc_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasCalc_Calc_Gas_Concentration

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(FuncChkMain_Chk_ConfirmEffectiveValue() == M_OK)		○	○	○	-	-
	if((stp_status->us_main_state & (DataStructConst_FAIL_BTA | DataStructConst_FAIL_BTL)) == 0)		○	○	-	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	-	-	-	-
	else		-	○	-	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER || ZIPCNumSet_Get_GasTestFlg() == M_ON)		○	-	-	-	-
	else		-	○	-	-	-
	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF
	FlowMain_Get_ModeStatus()	0~8	1	7	0	0	8
	FuncChkMain_Chk_ConfirmEffectiveValue()	0~2	0	0	0	1	2
	ZIPCNumSet_Get_GasTestFlg()	0~2	1	0	0	0	2
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

GasCalc_Clear_GasFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	stp_denst->uc_state_trb	0x00	0x00
Judgment			OK

GasCalc_Clear_HCPeak

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(!(stp_denst->uc_state_trb & DataStructConst_S_FAIL_ALL))		○	○	-
	else		-	-	○
	Symbol	Range			
Input	stp_denst->ul_denst	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	stp_denst->uc_denst_minus_flg	0~2	0	1	2
	stp_denst->uc_state_trb	0x00~0xFF	0x00	0x00	0xFF
Output	stp_denst->ul_pmin	0	0	0	0
	stp_denst->uc_pmin_flg	0,1	0	1	0
	stp_denst->ul_pmax	0	0	0	0
	stp_denst->uc_pmax_flg	0,1	0	1	0
Judgment			OK	OK	OK

GasCalc_Clear_Peak

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if(!(stp_denst->uc_state_trb & DataStructConst_S_FAIL_ALL))		○	○	-
	else		-	-	○
	Symbol	Range			
Input	stp_denst->ul_denst	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	stp_denst->uc_denst_minus_flg	0~2	0	1	2
	stp_denst->uc_state_trb	0x00~0xFF	0x00	0x00	0xFF
Output	stp_denst->ul_pmin	0	0	0	0
	stp_denst->uc_pmin_flg	0,1	0	1	0
	stp_denst->ul_pmax	0	0	0	0
	stp_denst->uc_pmax_flg	0,1	0	1	0
Judgment			OK	OK	OK

f_Shift_TwaBuf

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	puc_ch	0	0
Output	None	-	-
Judgment			OK

m_alarm1h_op

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	(stp_denst+puc_num)->ul_60_total	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF
	(stp_GasInfo_addr+puc_num)->ul_digit	1~501	500	1	501
Output	(stp_denst+puc_num)->ul_twa	0	0	0	0
Judgment			OK	OK	OK

m_Calc_AverageStelTwa

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i = 0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if((stp_denst+luc_i)->ul_denst != M_OFF_VAL)		○	○	-	○
	if((stp_denst+luc_i)->uc_m_denst_minus_flg)		○	-	-	○
	else		-	○	-	-
	if(puc_stop_flg == M_OFF)		○	-	-	-
	else		-	○	-	○
	else		-	-	○	-
	if(++((stp_denst+luc_i)->ul_ave_60_cnt) == 60)		○	○	-	-
	if(lul_temp1 > (stp_GasInfo_addr+luc_i)->ul_full_scale)		○	○	-	-
	else		○	○	-	-
	if((luc_i == DataStructConst_CO_CH) && (stp_data->uc_1h_alarm_on == M_ON))		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_stop_flg	0~2	0	1	0	2
	(stp_denst+luc_i)->ul_denst	0x00000000~0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xDDDDDDDD	0x00000000
	(stp_denst+luc_i)->uc_denst_minus_flg	0~2	1	0	0	2
	(stp_denst+luc_i)->ul_ave_60_cnt	0~61	59	59	60	61
	(stp_denst+luc_i)->ul_60_cnt	1,61	1	1	1	61
	(stp_denst+luc_i)->ul_60_ave	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+luc_i)->ul_ave_total_cnt	0x00000000~0xFFFFFFFE	0x00000000	0xFFFFFFFD	0x00000000	0xFFFFFFFE
	(stp_GasInfo_addr+luc_i)->ul_full_scale	0~100001	0	0	100000	100001
	(stp_GasInfo_addr+luc_i)->ul_digit	1~501	1	1	500	501
	stp_data->uc_1h_alarm_on	0~2	1	0	0	2
Output	(stp_denst+luc_i)->ul_60_ave	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	(stp_denst+luc_i)->ul_60_total	0x00000000~0x7FFFFFFF	0x00000000	0x7FFFFFFF	0x00000000	0x4210842
	(stp_denst+luc_i)->ul_ave	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	(stp_denst+luc_i)->ul_60_cnt	0	0	0	0	0
	(stp_denst+luc_i)->ul_ave_60_cnt	0	0	0	0	0
	(stp_denst+luc_i)->ul_ave_total_cnt	0x00000000~0xFFFFFFFE	0x00000001	0xFFFFFFFE	0x00000001	0xFFFFFFFE
Judgment			OK	OK	OK	OK

m_Calc_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((stp_GasInfo_addr + 1)->uc_gas_comb != 0x00)		○	○	-	○
	else		-	-	○	-
	if((stp_GasInfo_addr + 0)->uc_gas_comb != 0x00)		○	○	-	○
	else		-	-	○	-
	if((stp_GasInfo_addr + 2)->uc_gas_comb != 0x00)		○	○	-	○
	if((stp_GasInfo_addr + 2)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-
	else		-	○	-	○
	else		-	-	○	-
	if((stp_GasInfo_addr + 3)->uc_gas_comb != 0x00)		○	○	-	○
	if((stp_GasInfo_addr + 3)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-
	else		-	○	-	○
	else		-	-	○	-
	Symbol	Range				
Input	(stp_GasInfo_addr + 0)->uc_gas_comb	0~30	29	29	0	30
	(stp_GasInfo_addr + 1)->uc_gas_comb	0~30	29	29	0	30
	(stp_GasInfo_addr + 2)->uc_gas_comb	0~30	3	29	0	30
	(stp_GasInfo_addr + 3)->uc_gas_comb	0~30	3	29	0	30
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_Calc_DispDenstData

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○	○	○
	if(!(stp_denst->uc_state_trb & DataStructConst_S_SEN_FAIL))		○	○	○	○	○	-	○	-
	if(stp_denst->uc_m_denst_minus_flg)		○	○	○	○	-	-	-	-
	if(stp_denst->uc_m_over_minus_flg)		○	-	-	-	-	-	-	-
	else if((stp_GasInfo_addr->ul_upper_limit / 20) < stp_denst->ul_m_denst)		-	-	-	○	-	-	-	-
	else if(stp_GasInfo_addr->uc_suppress_onoff == M_OFF)		-	-	○	-	-	-	-	-
	else		-	○	-	-	-	-	-	-
	else		-	-	-	-	○	-	○	-
	if(stp_denst->uc_m_over_flg)		-	-	-	-	-	-	○	-
	else		-	-	-	-	○	-	-	-
	else		-	-	-	-	-	○	-	○
	Symbol	Range								
Input	stp_denst->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0xFF	0x00	0xFF
	stp_denst->uc_m_denst_minus_flg	0~2	1	1	1	1	0	0	0	2
	stp_denst->uc_m_over_minus_flg	0~2	1	0	0	0	0	0	0	2
	stp_GasInfo_addr->ul_upper_limit	0x00000000~0xFFFFFFFF	0x00000000	0x00000020	0x00000020	0xDDDDDDDD	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	stp_denst->ul_m_denst	0x00000000~0xFFFFFFFF	0x00000001	0x00000001	0x00000001	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	stp_denst->uc_m_over_flg	0~2	1	1	1	0	0	0	1	2
	stp_GasInfo_addr->uc_suppress_onoff	0~2	1	1	0	1	1	1	1	2
	stp_GasInfo_addr->ul_digit	1~501	500	1	1	1	1	1	1	501
Output	stp_denst->uc_m_over_flg	0,1	0	0	0	0	0	0	1	0
	stp_denst->uc_m_denst_minus_flg	0,1	1	0	1	1	0	0	0	0
	stp_denst->uc_m_over_minus_flg	0,1	1	0	0	0	0	0	0	0
	stp_denst->ul_denst	0x00000000~0xFFFFFFFF	0x000001F4	0x00000000	0x00000001	0xFFFFFFFF	0x00000000	0x00000000	0x00000001	0x00000000
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

m_Calc_Peak

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(puc_stop_flg == M_OFF)	○	○	○	○	○	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	○	○	○	○	○	-	-
	if(!(stp_denst->uc_state_trb & DataStructConst_S_FAIL_ALL))	○	○	○	○	○	○	-	-	-
	if((stp_denst->uc_denst_minus_flg) && (stp_denst->uc_pmin_flg))	○	○	-	-	-	-	-	-	-
	if(stp_denst->ul_pmin < stp_denst->ul_denst)	○	-	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	-	-	-
	else if((stp_denst->uc_denst_minus_flg == 1) && (stp_denst->uc_pmin_flg == 0))	-	-	○	-	-	-	-	-	-
	else if((stp_denst->uc_denst_minus_flg == 0) && (stp_denst->uc_pmin_flg == 0))	-	-	-	○	○	-	-	-	-
	if(stp_denst->ul_pmin > stp_denst->ul_denst)	-	-	-	○	-	-	-	-	-
	else	-	-	-	-	○	-	-	-	-
	else	-	-	-	-	-	○	-	-	-
	if(luc_flg == M_ON)	○	-	○	○	-	-	-	-	-
	else	-	○	-	-	○	○	-	-	-
	if((stp_denst->uc_denst_minus_flg) && (stp_denst->uc_pmax_flg))	○	○	-	-	-	-	-	-	-
	if(stp_denst->ul_pmax > stp_denst->ul_denst)	-	○	-	-	-	-	-	-	-
	else	○	-	-	-	-	-	-	-	-
	else if(!(stp_denst->uc_denst_minus_flg) && (stp_denst->uc_pmax_flg))	-	-	-	-	-	○	-	-	-
	else if((stp_denst->uc_denst_minus_flg == 0) && (stp_denst->uc_pmax_flg == 0))	-	-	-	○	○	-	-	-	-
	if(stp_denst->ul_pmax < stp_denst->ul_denst)	-	-	-	-	○	-	-	-	-
	else	-	-	-	○	-	-	-	-	-
	else	-	-	○	-	-	-	-	-	-
	if(luc_flg == M_ON)	-	○	-	-	○	-	-	-	-
	else	○	-	○	○	-	-	-	-	-
	else	-	-	-	-	-	-	○	-	-
	else	-	-	-	-	-	-	-	○	○
	Symbol	Range								
Input	puc_stop_flg	0~2	0	0	0	0	0	0	0	1	2
	stp_denst->uc_state_trb	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF
	stp_denst->uc_denst_minus_flg	0~2	1	1	1	0	0	0	2	2	1
	stp_denst->uc_pmin_flg	0~2	1	1	0	0	0	1	2	2	1
	stp_denst->ul_pmin	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	stp_denst->ul_denst	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0x00000000
	stp_denst->uc_pmax_flg	0~2	1	1	0	0	0	1	2	2	1
	stp_denst->ul_pmax	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0x00000000
Output	stp_denst->uc_pmin_flg	0~2	1	1	1	0	0	1	2	2	1
	stp_denst->uc_pmax_flg	0~2	1	1	0	0	0	0	2	2	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

m_stel_op

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	for(luc_i = 0 ; luc_i < 15 ; ++luc_i)	○	○	○
	if((stp_denst+puc_num)->ul_stel_point == luc_i)	○	-	-
	else	○	○	○
	if(++(stp_denst+puc_num)->ul_stel_point > 14)	-	○	○
	else	○	-	-
	Symbol	Range		
Input	(stp_denst+puc_num)->ul_stel_point	0~16	0	15	16
	(stp_denst+puc_num)->ul_60_ave	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	(stp_denst+puc_num)->ul_60_cnt	0~61	60	0	61
	(stp_GasInfo_addr+puc_num)->ul_digit	1~501	1	500	501
Output	(stp_denst+puc_num)->ul_stel_point	0,1	1	0	0
	(stp_denst+puc_num)->ul_stel	0,4772185	4772185	0	0
Judgment			OK	OK	OK

m_twa_op

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

		case1	case2	case3
Root	None	-	-	-
	Symbol	Range		
Input	(stp_denst+puc_num)->ul_60_total	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000
	(stp_GasInfo_addr+puc_num)->ul_digit)	0~501	500	0	501
Output	(stp_denst+puc_num)->ul_twa	0	0	0	0
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc_ec.c

GasCalcEC_Back_Linearize_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(puc_ch)		○	○	○	○
	case DataStructConst_O2_GAS_CH		○	-	-	-
	case DataStructConst_H2S_CH		-	○	-	-
	case DataStructConst_CO_CH		-	-	○	-
	default		-	-	-	○
	if(pfl_x_data > 100 || pfl_x_data < 0)		○	○	-	-
	else		-	-	○	-
	for(luc_i = 0U; luc_i < 10U; luc_i++)		-	-	-	○
	if((pfl_x_data >= (float)*(lusp_x_data+luc_i)) && (pfl_x_data <= (float)*(lusp_x_data+luc_i+1))		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	puc_ch	1~4	1	2	3	4
	pfl_x_data	0~110	110	110	100	0
Output	lfl_ret_ans	0~110	110	110	100	0
Judgment			OK	OK	OK	OK

GasCalcEC_Get_ECChkStartFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	0	1	2
Output	uc_m_ec_chk_flg[puc_flg]	0	0	0	0
Judgment			OK	OK	OK

GasCalcEC_Linearize_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(puc_ch)		○	○	○	○
	case DataStructConst_O2_GAS_CH		○	-	-	-
	case DataStructConst_H2S_CH		-	○	-	-
	case DataStructConst_CO_CH		-	-	○	-
	default		-	-	-	○
	if(pfl_y_data > 100)		○	○	-	-
	else		-	-	○	○
	for(luc_i = 0U; luc_i < 10U; luc_i++)		-	-	○	○
	if((pfl_y_data >= (float)*(lusp_y_data+luc_i)) && (pfl_y_data <= (float)*(lusp_y_data+luc_i+1)))		-	-	-	○
	else		-	-	○	
	Symbol	Range				
Input	puc_ch	1~4	1	2	3	4
	pfl_y_data	0~110	110	110	100	0
Output	lfl_ret_ans	4~110	110	110	100	4
Judgment			OK	OK	OK	OK

GasCalcEC_Reset_Tracking

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_ch	0~5	0	4	5
Output	stp_denst->fl_tracking_average	0	0	0	0
	stp_denst->uc_tracking_counter	0	0	0	0
	stp_denst->fl_tracking_offset	0	0	0	0
	stp_denst->fl_tracking_s_data[0]	100000000	100000000	100000000	100000000
	stp_denst->fl_tracking_s_data[1]	100000000	100000000	100000000	100000000
	stp_denst->fl_tracking_s_data[2]	100000000	100000000	100000000	100000000
	stp_denst->fl_tracking_s_data[3]	100000000	100000000	100000000	100000000
Judgment			OK	OK	OK

GasCalcEC_Set_ECChkStartFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	1	0	2
	puc_chk_flg	0~2	1	0	2
Output	uc_m_ec_chk_flg[puc_flg]	0~2	0	0	0
Judgment			OK	OK	OK

m_Get_PlusTrackingDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(puc_ch)		○	○	○	○	○
	case DataStructConst_H2S_CH		○	-	-	-	-
	case DataStructConst_CO_CH		-	○	-	-	-
	case DataStructConst_EXEC_CH		-	-	○	-	-
	default		-	-	-	○	○
	Symbol	Range					
Input	puc_ch	0~5	2	3	4	0	5
	puc_gas_num	0~30	0	0	0	29	30
Output	luc_ans	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_Get_PlusTrackingFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(puc_ch)		○	○	○	○	○
	case DataStructConst_H2S_CH		○	-	-	-	-
	case DataStructConst_CO_CH		-	○	-	-	-
	case DataStructConst_EXEC_CH		-	-	○	-	-
	default		-	-	-	○	○
	Symbol	Range					
Input	puc_ch	0~5	2	3	4	0	5
	puc_gas_num	0~30	0	0	0	29	30
Output	luc_ans	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

GasCalcEC_Calc_Tracking

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	17

Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17
Root	if((FlowMain_Get_ModeStatus() != FlowMain_MEASURE_MODE_NUMBER) && (FlowMain_Get_ModeStatus() != FlowMain_DISP_MODE_NUMBER))	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(stp_GasInfo_addr->uc_follower_onoff == M_OFF)	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	if(uc_flg != M_OFF)	-	-	-	○	○	○	○	○	○	○	-	-	○	○	○	○	○
	if(stp_denst->uc_tracking_counter >= M_TRACKING_COUNT)	-	-	-	○	○	○	○	○	○	○	-	-	-	○	○	○	○
	if(m_Get_PlusTrackingFlg(puc_ch, stp_GasInfo_addr->uc_gas_comb) == M_ON)	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	○
	else	-	-	-	-	-	○	○	○	○	○	-	-	○	○	○	-	-
	if(stp_denst->fl_tracking_s_data[0] != M_FLOAT_OFF_VAL)	-	-	-	○	○	○	○	○	○	-	-	-	-	-	-	○	○
	if(!((fl_disp >= 0) && (fl_d[2] < 0)))	-	-	-	○	○	○	○	○	-	-	-	-	-	-	-	○	○
	if(((fl_d[0] >= fl_al[0][0]) && (fl_d[1] >= fl_al[0][0]) && (fl_d[2] >= fl_al[0][0])) && ((fl_d[0] <= fl_al[1][1]) && (fl_d[1] <= fl_al[0][1]) && (fl_d[2] <= fl_al[1][1])))	-	-	-	○	○	○	○	-	-	-	-	-	-	-	-	○	○
	if(((fl_d[0] >= fl_al[1][0]) && (fl_d[1] >= fl_al[1][0]) && (fl_d[2] >= fl_al[1][0])) && ((fl_d[0] <= fl_al[1][1]) && (fl_d[1] <= fl_al[1][1]) && (fl_d[2] <= fl_al[1][1])))	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	○	○
	if(m_Get_PlusTrackingFlg(puc_ch, stp_GasInfo_addr->uc_gas_comb) == M_ON)	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	○	○
	if(((fl_d[0] >= 0) && (fl_d[1] >= 0) && (fl_d[2] >= 0)) || ((fl_d[0] < 0) && (fl_d[1] < 0) && (fl_d[2] < 0)))	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○	-
	else	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
	else	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
	if(uc_flg == M_ON)	-	-	-	○	-	○	-	-	-	-	-	-	-	-	-	○	-
	else	-	-	-	-	○	-	○	○	○	-	○	-	○	○	○	-	○
	if(stp_denst->fl_tracking_s_data[2] != M_FLOAT_OFF_VAL)	-	-	-	-	○	-	○	○	○	-	○	-	○	○	○	-	-
	if(fl_denst < 0)	-	-	-	-	○	-	-	-	-	-	○	-	○	-	○	-	-
	if(fl_disp < 0)	-	-	-	-	○	-	-	-	-	-	-	-	○	-	○	-	-
	if(fl_d[2] < 0)	-	-	-	-	-	-	-	-	-	-	-	-	○	-	○	-	-
	else	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	○	-	-	-	○	-	-
	else	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	○
	else	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	if(fl_denst >= 0) && (fl_disp < 0))	-	-	-	○	-	-	○	○	-	○	-	○	-	-	-	-	-
	else	-	-	-	-	○	○	-	-	○	-	○	-	-	○	○	○	○
	if(fl_disp > (fl_denst + stp_GasInfo_addr->ul_gas_alarm_min))	-	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-
	else	-	-	-	○	-	○	○	○	○	○	-	○	-	○	○	○	○
	if((fl_disp < 0) && (fl_disp < fl_denst))	-	-	-	○	-	-	○	○	-	○	-	-	-	-	-	-	-
	else	-	-	-	-	○	○	-	-	○	-	○	○	-	○	○	○	○
	if(fl_num >= (float)stp_GasInfo_addr->ul_gas_alarm_min)	-	-	-	○	-	-	○	○	-	-	-	-	-	-	-	○	○
	else	-	-	-	-	○	○	-	-	○	○	○	○	-	○	○	-	-
	switch(uc_judge_flg)	-	-	-	○	○	○	○	○	○	○	-	○	-	○	○	○	○
	case M_TRACKING_JUDGE_GENTLE	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	case M_TRACKING_JUDGE_MINUS	-	-	-	-	-	-	-	-	-	○	-	-	-	○	○	-	-
	case M_TRACKING_JUDGE_RESTRICTION	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case M_TRACKING_JUDGE_EXCESS_MINUS	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
	case M_TRACKING_JUDGE_EXCESS_PLUS	-	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-
	default	-	-	-	○	-	-	○	○	○	-	-	-	-	-	-	○	○
	if(fl_denst < 0)	-	-	-	-	-	○	-	-	-	-	-	-	-	○	○	-	-
	else	-	-	-	○	○	-	○	○	○	○	○	○	-	-	-	○	○
	else	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	Symbol	Range																
Input	puc_ch	0~5	0	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	pfl_denst	-10000~120001	120000	120001	-10000	1500	-300	-600	1500	1500	0	400	-100	50	-30	30	150	150
	puc_minus	0~2	1	2	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	FlowMain_Get_ModeStatus()	0~8	0	7	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	stp_GasInfo_addr->uc_follower_onoff	0~2	0	0	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	stp_denst->uc_tracking_counter	0~30	0	0	30	29	29	29	29	29	29	29	29	0	29	29	29	29	29
	m_Get_PlusTrackingFlg (puc_ch, stp_GasInfo_addr->uc_gas_comb)	0~2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	1	1
	stp_denst->fl_tracking_s_data[0]	-10000~120001	120000	120001	-10000	10	10	-10	10	10	0	100000000	100000000	100000000	100000000	0	0	0	0
	stp_denst->fl_tracking_s_data[1]	-10000~120001	120000	120001	-10000	10	10	-10	10	10	0	100000000	100000000	100000000	0	100000000	100000000	2	3.5
	stp_denst->fl_tracking_s_data[2]	-10000~120001	120000	120001	-10000	20	0	-10	20	20	0	100000000	0	0	0	3	3	3	3
	stp_denst->fl_tracking_s_data[3]	-10000~120001	120000	120001	-10000	30	-10	-10	30	30	1000000	100000000	100000000	100000000	0	4	-2	4	4
	stp_denst->fl_tracking_average	-10000~120001	120000	120001	-10000	1500	-300	-600	1500	1500	0	1000	1000	50	-30	0	0	0	0
	stp_GasInfo_addr->ul_gas_alarm_min	0~501	500	501	0	500	500	500	500	500	0	0	200	500	500	50	50	50	100
	stp_denst->fl_tracking_offset	-10000~120000	120000	120001	-10000	2000	-1000	0	2000	2000	0	1000	300	150	0	0	0	0	0
Output	stp_denst->fl_tracking_s_data[0]	-10000~100000000	120000	120001	-10000	10	10	-10	10	10	0	100000000	100000000	100000000	0	100000000	100000000	2	3.5
	stp_denst->fl_tracking_s_data[1]	-10000~100000000	120000	120001	-10000	20	0	-10	20	20	0	100000000	0	100000000	0	3	3	3	3
	stp_denst->fl_tracking_s_data[2]	-10000~100000000	120000	120001	-10000	30	-10	-10	30	30	1000000	100000000	100000000	0	0	4	-2	4	4
	stp_denst->fl_tracking_s_data[3]	-10000~100000000	120000	120001	-10000	100	-20	-40	100	100	0	46.66	30	100000000	-2	-1	-1	5	5
	stp_denst->fl_tracking_offset	-10000~120000	120000	120001	-10000	2000	-500	0	2000	2000	0	1000	-100	50	-2	-1	-1	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc_ec_denst.c

GasCalcECDenst_AirCal_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((stp_denst->fl_air_max >= lfl_sen_data) && (stp_denst->fl_air_min <= lfl_sen_data))		-	-	○	-
	else		○	○	-	○
	Symbol	Range				
Input	puc_gas_ch	0~5	0	2	4	5
	stp_denst->fl_air_max	0~9	0	0	8.6	9
	stp_denst->fl_air_min	-8.6~9	0	0	-8.6	9
	m_Get_SensorAD(puc_gas_ch, stp_GasInfo_addr->uc_gas_comb)	-800~801	-800	800	0	801
Output	stp_denst->fl_zero[0]	0	0	0	0	0
	stp_GasInfo_addr->fl_zero_buf[0]	0	0	0	0	0
	stp_GasInfo_addr->fl_zero_temp_data	0	0	0	0	0
	luc_ack	0,1	1	1	0	1
Judgment			OK	OK	OK	OK

GasCalcECDenst_AutoCal_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((puc_gas_ch == DataStructConst_EXEC_CH) && (stp_GasInfo_addr->uc_gas_direction))		○	-	-	-
	else		-	○	○	○
	if(lfl_temp3 > 0)		○	○	-	-
	if((stp_denst->fl_span_max[0] >= lfl_span_data) && (stp_denst->fl_span_min[0] <= lfl_span_data))		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_gas_ch	0~5	4	0	0	5
	stp_GasInfo_addr->uc_gas_direction	0~2	1	0	0	2
	pul_cal_denst	0X00000000~0xFFFFFFFF	0x00007D00	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	m_Get_SensorAD (puc_gas_ch, stp_GasInfo_addr->uc_gas_comb)	-800~801	800	800	0	801
	stp_GasInfo_addr->ul_full_scale	0~2001	2000	2000	0	2001
	stp_GasInfo_addr->ul_digit	1~6	1	1	5	6
	GasCalcEC_Back_Linearize_Denst (puc_gas_ch, lfl_temp1)	0~101	0	0	100	101
	GasCalcEC_Back_Linearize_Denst (puc_gas_ch, lfl_temp2)	0~101	0	0	100	101
	stp_denst->fl_span_max[0]	503~521	503	520.98	520.98	521
	stp_denst->fl_span_min[0]	34.97~521	34.97	36.18	520.98	521
	stp_denst->fl_zero[0]	-800~801	-800	-800	0	801
	stp_denst->fl_span_fix	0~2	1	1	0	2
Output	luc_ack	1	1	1	1	1
Judgment			OK	OK	OK	OK

GasCalcECDenst_Calc_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if((!(stp_denst->uc_m_denst_minus_flg)) && (lfl_temp < 100))		○	-	-	-
	else		-	○	○	○
	if((stp_GasInfo_addr->uc_suppress_onoff == M_ON) && (FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER))		○	○	-	-
	if((lfl_denst < lul_z_sup2) && (stp_denst->uc_m_denst_minus_flg == M_OFF))		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	if(stp_denst->ul_m_denst > ((stp_GasInfo_addr->ul_full_scale * 120) / 100))		-	-	○	○
	else		○	○	-	-
	if(stp_denst->uc_m_denst_minus_flg)		-	○	-	○
	else		○	-	○	-
	Symbol	Range				
Input	stp_denst->fl_span_fix	0~2	1	0	2	2
	stp_GasInfo_addr->fl_span_buf[0]	0~520.98	36.18	0	520.98	520.98
	stp_denst->uc_m_denst_minus_flg	0~2	0	1	0	2
	stp_denst->ul_digit	1~6	1	2	5	6
	stp_denst->ul_m_denst	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	stp_GasInfo_addr->ul_upper_limit	0~10001	10000	10000	0	10001
	stp_GasInfo_addr->uc_suppress_onoff	0,2	1	1	0	2
	m_Get_SuppressNum()	2~31	1	2	30	31
	GasCalcEC_Linearize_Denst()	0~10001	0	0	100000	100001
	m_Get_SensorAD()	-800~900	800	900	-800	-800
	m_Get_SpanAD()	-800~900	800	900	-800	-800
	FlowMain_Get_ModeStatus()	0~8	0	0	7	8
	stp_GasInfo_addr->ul_full_scale	0~2001	0	2000	2000	2001
Output	stp_denst->ul_m_denst	0~2400	0	0	2400	0
	stp_denst->uc_m_over_flg	0,1	0	0	1	0
	stp_denst->uc_m_over_minus_flg	0,1	0	1	0	0
Judgment			OK	OK	OK	OK

GasCalcECDenst_Set_SpanMargine

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if((puc_gas_ch == DataStructConst_EXEC_CH) && (stp_GasInfo_addr->uc_gas_direction))		○	○	-	-	-
	if(lfl_ad_data <= stp_denst->fl_zero[0])		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	if(lfl_ad_data >= stp_denst->fl_zero[0])		-	-	○	-	○
	else		-	-	-	○	-
	if(!(luc_margin_minus_flg) && (lfl_temp < 100))		○	-	○	-	-
	else		-	○	-	○	○
	if(stp_GasInfo_addr->ul_span_margin > stp_GasInfo_addr->ul_full_scale)		○	-	○	-	-
	else		-	○	-	○	○
	if(stp_GasInfo_addr->ul_span_margin > (stp_GasInfo_addr->ul_cal_denst * M_EC_CALDENST_MAG))		○	-	○	-	-
	else		-	○	-	○	○
	if(luc_margin_minus_flg == M_ON)		-	○	-	○	-
	else		○	-	○	-	○
	Symbol	Range					
Input	m_Get_SensorAD (puc_gas_ch, stp_GasInfo_addr->uc_gas_comb)	-800~801	50	800	50	-800	801
	puc_gas_ch	0~5	4	4	0	0	5
	stp_GasInfo_addr->uc_gas_direction	0~2	1	1	0	0	2
	stp_denst->fl_zero[0]	-800~801	80	-800	80	800	801
	stp_denst->fl_span_fix	0~2	1	0	1	0	2
	stp_denst->fl_span_min[0]	34.97~37	34.97	34.97	36.18	36.18	37
	GasCalcEC_Linearize_Denst(puc_gas_ch, lfl_temp)	-10000~120001	120000	-10000	120000	-10000	120001
	stp_denst->ul_digit	1~6	1	5	1	5	6
	stp_GasInfo_addr->ul_full_scale	0~2001	2000	0	2000	0	2001
	stp_GasInfo_addr->ul_cal_denst	0x00000000~0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
Output	stp_GasInfo_addr->ul_span_margin	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_Calc_ECSpanTempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<5; luc_i++)		○	○	○	○
	switch(puc_ch)		○	○	○	○
	case DataStructConst_H2S_CH		-	○	-	-
	case DataStructConst_CO_CH		-	-	○	-
	case DataStructConst_EXEC_CH		-	-	-	○
	default		○	-	-	-
	Symbol	Range				
Input	puc_ch	0~4	0	2	3	4
	puc_lin_num	0~7	7	0	1	6
	pdb_temp	-20~51	-20	50	50	51
Output	ldb_ans	0~1	0	0.8	1	0
Judgment			OK	OK	OK	OK

m_Calc_ECZeroTempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<5; luc_i++)		○	○	○	○
	switch(puc_ch)		○	○	○	○
	case DataStructConst_H2S_CH		○	-	-	-
	case DataStructConst_CO_CH		-	○	-	-
	case DataStructConst_EXEC_CH		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	puc_ch	2~5	2	3	4	5
	puc_lin_num	0~7	0	0	6	7
	pdb_temp	-55~86	-55	-55	85	86
Output	ldb_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_Get_SensorAD

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(puc_ch)		○	○	○	○	○
	case DataStructConst_H2S_CH		-	○	-	○	-
	if(puc_gas_num != DataStructConst_SINGLE_H2S_GAS_NUM)		-	○	-	-	-
	else		-	-	-	○	-
	case DataStructConst_CO_CH		-	-	○	-	-
	default		○	-	-	-	○
	Symbol	Range					
Input	puc_ch	2~5	1	2	3	2	5
	puc_gas_num	0~30	29	0	0	2	30
	stp_sub_ad->fl_ec_ad_data[2]	-801~800	-800	0	200	800	-801
	stp_sub_ad->fl_ec_ad_data[1]	-801~800	-800	0	100	800	-801
Output	lfl_ad_ans	0~800	0	0	200	800	0
Judgment			OK	OK	OK	OK	OK

m_Get_SpanAD

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
	if(pfl_sen_ad >= stp_denst->fl_zero[0])		○	-	-	-	-
	else		-	○	-	-	-
	Symbol	Range					
Input	puc_ch	0~5	4	4	2	0	5
	pfl_sen_ad	-800~801	-800	800	500	0	801
	stp_denst->fl_zero[0]	-801~800	800	-800	0	100	-801
Output	stp_denst->uc_m_denst_minus_flg	0,1	0	0	0	1	0
	lfl_span_ad	100~800	800	800	500	100	801
Judgment			OK	OK	OK	OK	OK

m_Get_SuppressNum

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(puc_ch)		○	○	○	○	○
	case DataStructConst_H2S_CH		○	-	-	-	-
	case DataStructConst_CO_CH		-	○	-	-	-
	default		-	-	○	○	○
	Symbol	Range					
Input	puc_ch	0~5	2	3	4	0	5
	puc_gas_num	0~30	0	0	0	29	30
	puc_flg	0~2	0	0	0	1	2
Output	lul_sup_ans	0~3	3	2	0	0	0
Judgment			OK	OK	OK	OK	OK

m_Get_ExportControlsNum

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(puc_ch)		○	○	○	○
	case DataStructConst_H2S_CH		-	○	-	-
	case DataStructConst_CO_CH		-	-	○	-
	default		○	-	-	○
	Symbol	Range				
Input	puc_ch	0~4	0	2	3	4
	puc_gas_num	0	0	0	0	0
Output	lul_sup_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc_ec_h2_cancel_co.c

GasCalcECH2CancelCO_Calc_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8
Root	for(luc_col = 0U; luc_col < M_CANSEL_GAS_SENS_MAX; luc_col++)	○	○	○	○	○	○	○	○
	for(luc_row = 0U; luc_row < M_CANSEL_GAS_SENS_MAX; luc_row++)	○	○	○	○	○	○	○	○
	if(lfl_mmult < 0)	○	-	-	-	-	-	-	○
	else	-	○	○	○	○	○	○	-
	if(lfl_mmult < lfl_eps)	○	○	-	-	-	-	○	-
	if(lfl_we_out_denst[M_WE1_CH] > lfl_we_out_denst[M_WE2_CH])	○	-	-	-	-	-	-	-
	else	-	○	-	-	-	-	○	-
	else if(lfl_temp_val < M_CO_H2_DENST_MIN_TEMP)	-	-	○	-	-	-	-	-
	else	-	-	-	○	○	○	-	○
	if(((stp_GasInfo_addr + M_CMB_CO)->uc_suppress_onoff == M_ON) && (FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER))	-	○	-	-	-	-	○	-
	if((lfl_co_denst < lfl_z_sup2) && (lfl_co_denst > 0.0f))	-	○	-	-	-	-	-	-
	else	-	-	-	-	-	-	○	-
	else	○	-	○	○	○	○	-	○
	if(lsl_co_denst >= 0)	○	○	-	-	-	○	-	○
	if(lsl_co_denst > (S_LNG)(((stp_GasInfo_addr+M_CMB_CO)->ul_full_scale * 120) / 100))	○	-	-	-	-	-	-	-
	else	-	○	-	-	-	○	-	○
	else	-	-	○	○	○	-	○	-
	if(lsl_co_denst > (S_LNG)(((stp_GasInfo_addr+M_CMB_CO)->ul_full_scale * 120) / 100))	-	-	○	-	-	-	-	-
	else	-	-	-	○	○	-	○	-
	if((stp_denst+M_CMB_CO)->uc_m_denst_minus_flg == M_ON)	-	-	○	○	○	-	○	-
	if((stp_denst+M_CMB_CO)->ul_m_denst > ((stp_GasInfo_addr+M_CMB_CO)->ul_full_scale / 5UL))	-	-	○	-	-	-	-	-
	else	-	-	-	○	○	-	○	-
	else	○	○	-	-	-	○	○	○
	if(lsl_h2_denst >= 0)	○	○	○	-	-	○	-	○
	if(lsl_h2_denst > (S_LNG)(((stp_GasInfo_addr+M_CMB_H2)->ul_full_scale * 120) / 100))	-	-	-	-	-	○	-	○
	else	○	○	○	-	-	-	○	-
	else	-	-	-	○	○	-	-	-
	if(lsl_h2_denst > (S_LNG)(((stp_GasInfo_addr+M_CMB_H2)->ul_full_scale * 120) / 100))	-	-	-	-	○	-	-	-
	else	-	-	-	○	-	-	-	-
	if((stp_denst+M_CMB_H2)->uc_m_denst_minus_flg == M_ON)	-	-	-	○	○	-	-	-
	if((stp_denst+M_CMB_H2)->ul_m_denst > ((stp_GasInfo_addr+M_CMB_H2)->ul_full_scale / 5UL))	-	-	-	-	○	-	-	-
	else	-	-	-	○	-	-	-	-
	else	○	○	○	-	-	○	○	○
	Symbol	Range							
Input	(stp_GasInfo_addr + M_CMB_CO)->uc_suppress_onoff	0~2	1	1	0	0	0	0	1	2
	stp_sub_ad->fl_temp_data	-55~86	85	85	-55	0	0	0	0	86
	(stp_denst+M_CMB_CO)->fl_zero[0]	-800~900	0	0	800	10	800	-800	900	801
	(stp_denst+M_CMB_H2)->fl_zero[0]	-800~900	0	0	800	10	800	-800	900	801
	(stp_GasInfo_addr+M_CMB_CO)->fl_span_buf[0]	118.04,119	118.04	118.04	118.04	118.04	118.04	118.04	118.04	119
	(stp_GasInfo_addr+M_CMB_H2)->fl_span_buf[0]	1~34	1	33.8	33.8	33.8	33.8	33.8	33.8	34
	(stp_GasInfo_addr+M_CMB_CO)->fl_span_buf[1]	202.36,203	202.36	202.36	202.36	202.36	202.36	202.36	202.36	203
	(stp_GasInfo_addr+M_CMB_H2)->fl_span_buf[1]	37.8,38	37.8	37.8	1	37.8	37.8	37.8	37.8	38
	(stp_GasInfo_addr+M_CMB_CO)->ul_full_scale	100~4294967295	2000	2000	2000	2000	100	100	2000	4294967295
	(stp_GasInfo_addr+M_CMB_H2)->ul_full_scale	100~2001	2000	2000	2000	2000	100	100	2000	2001
	(stp_GasInfo_addr+M_CMB_CO)->ul_digit	1,2	1	1	1	1	1	1	1	2
	(stp_GasInfo_addr+M_CMB_H2)->ul_digit	1,2	1	1	1	1	1	1	1	2
	m_Chk_PresenceOfInverseMatrix()	-200~117854.89	-1.00E-10	0	1	-200	-154.785	117854.89	0	117855
	m_Get_AveAD(M_WE1_CH)	-900~800	800	0	0	0	-900	-800	0	-801
	m_Get_AveAD(M_WE2_CH)	-801~900	-800	1	0	0	800	900	0	-801
	m_Linearaise_H2CancelCO()	1~2	1	1.152	1	1	1	1	2	2
	FlowMain_Get_ModeStatus()	0~8	7	0	4	4	0	0	0	8
Output	(stp_denst+M_CMB_CO)->uc_m_denst_minus_flg	0,1	0	0	1	0	0	0	1	1
	(stp_denst+M_CMB_H2)->uc_m_denst_minus_flg	0,1	0	0	0	1	1	0	0	1
	(stp_denst+M_CMB_CO)->ul_m_denst	0~400	135	0	79	0	12	0	177	400
	(stp_denst+M_CMB_H2)->ul_m_denst	0~640800	0	0	0	400	20	120	0	640800
	(stp_denst+M_CMB_CO)->uc_m_over_flg	0,1	0	0	0	0	1	0	0	0
	(stp_denst+M_CMB_H2)->uc_m_over_flg	0,1	0	0	0	0	0	1	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

GasCalcECH2CancelCO_Set_AirCal

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(((stp_denst+M_CMB_CO)->fl_air_min <= lfl_ex_adc_curr[M_WE1_CH]) && (lfl_ex_adc_curr[M_WE1_CH] <= (stp_denst+M_CMB_CO)->fl_air_max))		-	-	-	○	○
	if(((stp_denst+M_CMB_H2)->fl_air_min < lfl_ex_adc_curr[M_WE2_CH]) && (lfl_ex_adc_curr[M_WE2_CH] < (stp_denst+M_CMB_H2)->fl_air_max))		-	-	-	-	-
	else		-	-	-	○	○
	else		○	○	○	-	-
	Symbol	Range					
Input	fl_air_max	-800~801	0	0	800	-800	801
	fl_air_min	-800~801	0	800	0	-800	801
	m_Correction_ZeroComp()	-8.6~9	-8.6	-8.6	8.6	0	9
	m_Get_AveAD()	-8.6~9	8.6	8.6	-8.6	0	9
	fl_temp_data	-55~86	-55	85	-55	85	86
Output	fl_zero	0	0	0	0	0	0
	fl_zero_temp_data	0	0	0	0	0	0
	fl_zero_buf	0	0	0	0	0	0
	luc_ack	1	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK

GasCalcECH2CancelCO_Set_AutoCal

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(puc_gas == 0)		○	-
	else		-	○
	if((lfl_ex_adc_curr[M_WE1_CH] > 0) && (lfl_ex_adc_curr[M_WE2_CH] > 0))		-	-
	if(puc_gas == 0)		-	-
	else		-	-
	if((lfl_we1_span_buf_tmp >= (stp_denst+luc_gas_select)->fl_span_min[0]) && (lfl_we1_span_buf_tmp <= (stp_denst+luc_gas_select)->fl_span_max[0]))		-	○
	if((lfl_we2_span_buf_tmp >= (stp_denst+luc_gas_select)->fl_span_min[1]) && (lfl_we2_span_buf_tmp <= (stp_denst+luc_gas_select)->fl_span_max[1]))		-	-
	else		-	-
	else		-	-
	else		○	○
	Symbol	Range		
Input	puc_gas	0,1	0	1
	pul_cal_denst	0	0	0
	fl_span_max[0]	0	0	0
	fl_span_max[1]	0	0	0
	fl_span_min[0]	0	0	0
	fl_spnan_min[1]	0	0	0
	fl_zero[0]	0	0	0
	ul_full_scale	0	0	0
	uc_cal_date[0]	0	0	0
	uc_cal_sdm[0]	0	0	0
	uc_bump_date[0]	0	0	0
	m_Correction_ZeroComp()	0	0	0
	m_Get_AveAD()	0	0	0
	m_Linearaise_H2CancelCO()	0	0	0
	fl_temp_data	0	0	0
Output	luc_ack	0,1	1	1
	fl_span_temp_data	0	0	0
	fl_span_buf	0	0	0
Judgment			OK	OK

GasCalcECH2CancelCO_Set_SpanMargine

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0U; luc_i<M_CANSEL_GAS_SENS_MAX; luc_i++)		○	○	○
	for(luc_j=0U; luc_j<M_CANSEL_GAS_SENS_MAX; luc_j++)		○	○	○
	if(lul_tmp1 < lul_tmp2)		-	-	-
	else		○	○	○
	if((stp_GasInfo_addr+M_CMB_CO)->ul_span_margin > (stp_GasInfo_addr+M_CMB_CO)->ul_full_scale)		-	-	-
	if((stp_GasInfo_addr+M_CMB_CO)->ul_span_margin > ((stp_GasInfo_addr+M_CMB_CO)->ul_cal_denst * M_CO_H2_CALDENST_MAG))		-	-	-
	Symbol	Range			
Input	m_Get_AveAD()	-800~801	801	800	-800
	fl_span_min[0]	0	0	0	0
	fl_span_min[1]	0	0	0	0
	fl_zero[0]	0	0	0	0
	ul_full_scale	0	0	0	0
	ul_digit	0	0	0	0
	ul_cal_denst	0	0	0	0
	ul_span_margin	0	0	0	0
	m_Correction_ZeroComp()	0	0	0	0
	m_Linearaise_H2CancelCO()	0	0	0	0
	fl_temp_data	0	0	0	0
Output	ul_span_margin	0	0	0	0
Judgment			OK	OK	OK

m_Calc_InverseMatrix

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0U; luc_i < luc_Degree; luc_i++)		○	○	○
	for(luc_j = 0U; luc_j < luc_Degree; luc_j++)		○	○	○
	for(luc_i = 0U; luc_i < luc_Degree; luc_i++)		○	○	○
	for(luc_j = 0U; luc_j < luc_Degree; luc_j++)		○	○	○
	for(luc_j = 0U; luc_j < luc_Degree; luc_j++)		○	○	○
	if(luc_i != luc_j)		○	○	○
	for(luc_k = 0U; luc_k < luc_Degree; luc_k++)		○	○	○
	else		○	○	○
	Symbol	Range			
Input	pfl_matrix[M_CANSEL_GAS_SENS_MAX] [M_CANSEL_GAS_SENS_MAX]	-0.008424819~0.3	-0.008424819	0.205383276	0.3
	pfl_inv_matrix[M_CANSEL_GAS_SENS_MAX] [M_CANSEL_GAS_SENS_MAX]	~-390.031853~582	-390.031853	581.275788	582
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_Calc_TempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0U; luc_i<M_CANSEL_GAS_SENS_MAX; luc_i++)		○	○	○
	for(luc_j=0U; luc_j<M_CANSEL_GAS_SENS_MAX; luc_j++)		○	○	○
	Symbol	Range			
Input	pfl_temp_val	-55~86	-55	85	86
	pfl_sens_ratio[0][0]	-0.49851~11	-0.49851	10.86684	11
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_Chk_PresenceOfInverseMatrix

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

		case1	case2	case3	case4
Root	for(luc_i=0U; luc_i<M_CANSEL_GAS_SENS_MAX; luc_i++)	○	○	○	○
	for(luc_j=0U; luc_j<M_CANSEL_GAS_SENS_MAX; luc_j++)	○	○	○	○
	for(luc_i=0U; luc_i<luc_Degree; luc_i++)	○	○	○	○
	for(luc_j = 0U; luc_j < luc_Degree; luc_j++)	○	○	○	○
	if(luc_i < luc_j)	○	○	○	○
	if(lfl_matrix[luc_i][luc_i] == 0)	-	-	○	-
	else	○	○	-	○
	for(luc_k = 0U; luc_k < luc_Degree; luc_k++)	-	-	-	-
	else	○	○	○	○
	if(luc_nan_chk_flg == M_ON)	-	-	○	-
	else	○	○	-	○
	for(luc_i=0U; luc_i<luc_Degree; luc_i++)	○	○	-	○
	if(luc_nan_chk_flg == M_ON)	-	-	○	-
	else	○	○	-	○
	Symbol	Range			
Input	pfl_chk_matrix[M_CANSEL_GAS_SENS_MAX] [M_CANSEL_GAS_SENS_MAX]	-0.008424819~0.205383276	-0.008424819	0.205383276	0	1
Output	lfl_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_Correction_DenstValComp

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0U; luc_i < (luc_size - 1U); luc_i++)		○	○	○
	for(luc_j = (luc_size - 1U); luc_j > luc_i; luc_j--)		○	○	○
	if(lfl_sort_tmp[luc_j-1U] > lfl_sort_tmp[luc_j])		○	-	○
	else		○	○	-
	if(pfl_denst_val_tmp > lfl_sort_tmp[2])		-	-	-
	else if(pfl_denst_val_tmp < lfl_sort_tmp[0])		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pfl_denst_val_tmp	-2001~2000	2000	-2000	-2001
	pfl_we_out_denst[M_WE1_CH]	-36036~41512.81602	-36035.43058	41512.81602	-36036
	pfl_we_out_denst[M_WE2_CH]	-119133~137241.0882	-119132.889	137241.0882	-119133
Output	lfl_denst	-2001~0	0	0	-2001
Judgment			OK	OK	OK

m_Correction_ZeroComp

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3	case3
Root	if(pfl_temp_val > 90.0F)		-	-	○	-
	else		○	○	-	○
	if(pfl_temp_val < -50.0F)		○	-	-	-
	else		-	○	○	○
	for(luc_i = 0U; luc_i < 14U; luc_i++)		○	○	○	○
	if(puc_ad_ch == M_WE1_CH)		○	-	-	○
	if((pfl_temp_val >= fl_we1_temp_zerotbl[0][luc_i]) && (pfl_temp_val <= fl_we1_temp_zerotbl[0][luc_i + 1U]))		○	-	-	○
	else		-	-	-	○
	else		-	○	-	-
	if((pfl_temp_val >= fl_we2_temp_zerotbl[0][luc_i]) && (pfl_temp_val <= fl_we2_temp_zerotbl[0][luc_i + 1U]))		-	○	-	-
	else		-	-	○	-
	if(puc_ad_ch == M_WE1_CH)		○	-	-	○
	else if(puc_ad_ch == M_WE2_CH)		-	○	-	-
	else		-	-	○	-
	Symbol	Range				
Input	puc_ad_ch	0~2	0	1	2	0
	pfl_temp_val	-55~91	-55	85	91	85
Output	lfl_zero_comp_val	0~0.892	0.026	0.042	0	0.892
Judgment			OK	OK	OK	OK

m_Get_AveAD

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_num == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_num	0~2	0	1	2
Output	stp_sub_ad->fl_h2_cancel_co_ad[puc_num]	0	0	0	0
Judgment			OK	OK	OK

m_Linearaise_H2CancelCO

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pfl_per_fullscale > 100.0F)		○	○	-
	else		-	-	○
	if(pfl_per_fullscale < 0.0F)		-	-	○
	else		○	○	-
	for(luc_i = 0U; luc_i < 21U; luc_i++)		○	○	○
	if((pfl_per_fullscale >= (FLOAT)ssc_ec_we1_lineartbl[0][luc_i]) && (pfl_per_fullscale <= (FLOAT)ssc_ec_we1_lineartbl[0][luc_i + 1U]))		-	-	○
	else		○	○	-
	if(puc_ad_ch == M_WE1_CH)		○	-	-
	else if(puc_ad_ch == M_WE2_CH)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	pfl_per_fullscale	-8916.227193~8917	8916.227193	8917	-8916.227193
	puc_ad_ch	0~2	0	1	2
Output	lfl_linear	0~1	0.2	1	0
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc_ec_o2.c

GasCalcECO2_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(ucs_pressure_flg != M_P_FLG_OFF)		○	○	○	○	○	○	○	-
	if(uss_pressure_in_count)		○	○	○	○	○	-	○	-
	if(uss_pressure_in_count == 0)		○	○	○	○	○	-	-	-
	if(lsl_ad1 <= lsl_ad2)		○	○	-	-	-	-	-	-
	if(ucs_pressure_chk_flg == 1)		○	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-
	else if(lsl_ad3 >= lsl_ad2)		-	-	○	○	-	-	-	-
	if(ucs_pressure_chk_flg == 2)		-	-	○	-	-	-	-	-
	else		-	-	-	○	-	-	-	-
	else		-	-	-	-	○	-	-	-
	else		-	-	-	-	-	○	-	-
	if(uss_pressure_out_count)		-	○	-	○	○	-	○	-
	if(uss_pressure_out_count == 0)		-	○	-	○	○	-	-	-
	else		-	-	-	-	-	-	○	-
	else		○	-	○	-	-	○	-	-
	else		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	ucs_pressure_chk_flg	0~3	1	0	2	1	1	1	1	3
	uss_pressure_in_count	0~301	1	1	1	1	1	0	300	301
	uss_pressure_out_count	0~501	1	1	1	1	1	0	500	501
	fl_pressure_pmax_ad	-800~801	801	0	-800	800	0	0	0	0
	fl_pressure_pmin_ad	-800~801	-800	0	800	801	2	0	0	0
	ucs_pressure_flg	0~3	2	1	1	1	1	1	3	0
	stp_sub_ad->fl_ec_ad_data[0]	-801~800	1	1	-801	800	1	0	0	-800
Output	uss_pressure_in_count	0~301	0	0	0	0	0	0	299	301
	uss_pressure_out_count	0~501	0	0	0	0	0	0	499	501
	ucs_pressure_flg	0~3	2	1	1	1	2	1	3	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

GasCalcECO2_AirCal_O2Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(((stp_denst->fl_air_max * lfl_span_temp) >= lfl_sen_data) && ((stp_denst->fl_air_min * lfl_span_temp) <= lfl_sen_data))		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	stp_sub_ad->fl_ec_ad_data[0]	-800~801	0	-800	800	801
	m_Calc_O2TempComp(stp_sub_ad->fl_temp_data)	0.774175~2	0	0.774175	1.0138	2
	stp_denst->fl_air_max	-105.8~1	0	-105.8	0	1
	stp_denst->fl_air_min	-30.9~1	0	-30.9	0	1
	stp_sub_ad->fl_temp_data	-55~86	-55	85	0	86
Output	stp_denst->fl_zero[0]	0	0	0	0	0
	stp_GasInfo_addr->fl_zero_buf[0]	0	0	0	0	0
	stp_GasInfo_addr->fl_zero_temp_data	0	0	0	0	0
	luc_ack	1	1	1	1	1
Judgment			OK	OK	OK	OK

GasCalcECO2_AutoCal_O2Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lfl_temp4 < 0)		-	-	○	-
	else		○	○	-	○
	if(lfl_temp4 < (1E-10F))		○	-	○	○
	else		-	○	-	-
	if((stp_denst->fl_span_max[0] >= lfl_span_data) && (stp_denst->fl_span_min[0] <= lfl_span_data))		-	-	-	○
	else		○	○	○	-
	Symbol	Range				
Input	stp_GasInfo_addr->fl_zero_buf[0]	-800~801	-800	800	801	0
	stp_sub_ad->fl_ec_ad_data[0]	-800~801	800	-800	801	0
	pul_cal_denst	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF
	m_Calc_O2TempComp (stp_GasInfo_addr->fl_zero_temp_data)	0.774175~1.1	0.774175	10.138	1.1	1.1
	stp_GasInfo_addr->ul_full_scale	0~401	0	400	401	401
	GasCalcEC_Back_Linearize_Denst(1, lfl_air_point)	0~101	100	0	1	101
	GasCalcEC_Back_Linearize_Denst(1, lfl_span_point)	0~101	1	0	100	101
	m_Calc_O2TempComp(stp_sub_ad->fl_temp_data)	0.774175~1.1	0.774175	10.138	1.1	1.1
	stp_denst->fl_span_max[0]	-7.83~1	-7.83	0	1	0
	stp_denst->fl_span_min[0]	0~3.75	3.75	0	1	0
Output	luc_ack	0,1	0	1	0	0
Judgment			OK	OK	OK	OK

GasCalcECO2_Calc_O2Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if(stp_GasInfo_addr->uc_suppress_onoff == M_ON && FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER)		○	○	○	-	○	-	-	-
	else		-	-	-	○	-	○	○	○
	if(lfl_sens_volt <= stp_GasInfo_addr->fl_span_buf[0])		○	○	○	-	○	-	-	-
	else		-	-	-	○	-	○	○	○
	if(!(stp_denst->uc_m_denst_minus_flg) && (lfl_temp2 < 100))		○	○	○	○	-	-	-	-
	else		-	-	-	-	○	○	○	○
	if((lfl_denst > lfl_z_sup2) && (lfl_denst < lfl_z_sup3))			-	-	-	-	-	-	○
	else		○	○	○	○	○	○	○	
	if((lfl_denst > lfl_z_sup1) && (lfl_denst <= lfl_z_sup2))		-	○	-	-	-	-	-	-
	else if((lfl_denst >= lfl_z_sup3) && (lfl_denst < lfl_z_sup4))		-	-	○	-	-	-	-	-
	else		-	-	-	○	○	○	○	○
	if(lul_denst == 0)		-	-	-	○	-	-	-	-
	else		○	○	○	-	○	○	○	○
	if(lul_denst > ((stp_GasInfo_addr->ul_full_scale * 120) / 100))		○	○	○	-	-	○	○	○
	else		-	-	-	○	○	-	-	-
	if(luc_denst_minus_flg)		-	-	-	-	-	○	○	○
	else		○	○	○	○	○	-	-	-
	if(luc_pressure_chk_flg != M_OFF){		○	-	-	-	○	○	○	○
	else if(ucs_pressure_flg != M_P_FLG_OFF)		-	○	-	-	-	-	-	-
	else		-	-	○	○	-	-	-	-
	Symbol	Range								
Input	stp_sub_ad->fl_ec_ad_data[0]	-800~801	0	0	0	800	-800	800	801	0
	stp_GasInfo_addr->uc_suppress_onoff	0~2	1	1	1	0	1	1	2	1
	stp_GasInfo_addr->fl_span_buf[0]	36.18~521	36.18	520.98	50	50	0	0	520.98	521
	stp_denst->fl_zero[0]	-800~801	0	0	0	800	-800	800	801	0
	stp_GasInfo_addr->ul_full_scale	0~100001	101	100	100	0	100000	100000	100001	104
	ucs_pressure_flg	0~3	0	1	0	0	0	2	3	0
	stp_denst->ul_digit	1,2	1	1	1	1	1	1	2	1
	stp_GasInfo_addr->ul_upper_limit	0~161	1	1	1	1	0	160	161	1
	m_Chk_PressureDenst()	0~3	1	0	0	0	2	3	3	1
	GasCalcEC_Linearize_Denst()	0~100001	205	196	215	100000	1	100000	100001	205
	GasCalcEC_Back_Linearize_Denst()	0~101	50	50	50	50	0	100	101	50
	m_Calc_O2TempComp()	0.774175~1.0138	0.774175	1.0138	1	1	1	1.1	1.1	0.774175
	FlowMain_Get_ModeStatus()	0~8	0	0	0	7	0	8	8	0
Output	stp_denst->uc_m_denst_minus_flg	0	0	0	0	0	0	0	0	0
	stp_denst->ul_m_denst	0	0	0	0	0	0	0	0	0
	stp_denst->uc_m_over_flg	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

GasCalcECO2_Set_SpanMargine

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_GasInfo_addr->fl_zero_buf[0] <= stp_GasInfo_addr->fl_span_buf[0])		○	○	-	-
	if(lfl_temp2 < 100)		○	-	-	-
	else		-	○	-	-
	if(stp_GasInfo_addr->ul_span_margin > M_O2_SPANMARGIN_MAX)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_GasInfo_addr->fl_zero_buf[0]	-800~801	-800	0	800	801
	stp_GasInfo_addr->fl_span_buf[0]	-0.45~1	-0.45	0	0	1
	m_Calc_O2TempComp (stp_GasInfo_addr->fl_zero_temp_data)	0.774175~1.1	0.774175	1.0138	1.0138	1.1
	stp_denst->fl_air_max	-105.8~1	0	-105.8	0	1
	stp_GasInfo_addr->ul_full_scale	0~401	400	0	400	401
	GasCalcEC_Back_Linearize_Denst(1, lfl_temp2)	0~101	100	0	100	101
	GasCalcEC_Linearize_Denst(1, lfl_temp2)	-10~481	480	-10	10	481
	stp_denst->ul_digit	1,2	1	1	1	2
Output	stp_GasInfo_addr->ul_span_margin	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_Calc_O2TempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pfl_temp	-55~86	-55	85	86
Output	lfl_ans	0.774175~1.007975	0.774175	1.007975	1.006684
Judgment			OK	OK	OK

m_Chk_PressureDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_minus == 1)		○	-	-
	else		-	○	○
	if(lsls_back_denst >= M_PRESSURE_STAND_DENST && lsl_chk_denst < M_PRESSURE_STAND_DENST)		○	-	-
	else if(lsls_back_denst <= M_PRESSURE_STAND_DENST && lsl_chk_denst > M_PRESSURE_STAND_DENST)		-	○	-
	else		-	-	○
	if(FlowMain_Get_ModeStatus() > FlowMain_DISP_MODE_NUMBER)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pul_denst	16~400	16	400	0
	FlowMain_Get_ModeStatus()	0~8	7	0	8
	puc_minus	0~2	1	0	2
Output	luc_ans	0,2	0	2	0
Judgment			OK	OK	OK

m_ForcedStop_Pressure

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(ucs_pressure_flg == M_P_FLG_END)		○	○	○	○	-
	if(ucs_pressure_chk_flg == 2)		○	-	○	-	-
	if(fl_pressure_pmax_ad > stp_sub_ad->fl_ec_ad_data[0])		○	-	-	-	-
	else		-	-	○	-	-
	else		-	○	-	○	-
	if(ucs_pressure_chk_flg == 1)		-	○	-	○	-
	if(fl_pressure_pmin_ad < stp_sub_ad->fl_ec_ad_data[0])		-	○	-	-	-
	else		-	-	-	○	-
	else		○	-	○	-	-
	else		-	-	-	-	○
	Symbol	Range					
Input	ucs_pressure_chk_flg	0~3	2	1	2	1	3
	fl_pressure_pmax_ad	-800~801	800	0	-800	0	801
	fl_pressure_pmin_ad	-800~800	0	-800	0	800	-801
	ucs_pressure_flg	0~3	2	2	2	2	3
	stp_sub_ad->fl_ec_ad_data[0]	-800~801	0	0	801	-800	800
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

m_Peak_Pressure

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(ucs_pressure_flg == M_P_FLG_FIRST)		○	○	-	-	-
	if(fl_pressure_pmax_ad > stp_sub_ad->fl_ec_ad_data[0])		-	○	-	-	-
	else		○	-	-	-	-
	if(fl_pressure_pmin_ad < stp_sub_ad->fl_ec_ad_data[0])		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	Symbol	Range					
Input	fl_pressure_pmax_ad	-800~801	800	-800	801	801	801
	fl_pressure_pmin_ad	-800~800	-800	-800	-801	800	800
	ucs_pressure_flg	0~3	1	1	0	2	3
	stp_sub_ad->fl_ec_ad_data[0]	-800~800	-800	800	800	-801	-801
Output	fl_pressure_pmax_ad	-800,801	-800	-800	801	801	801
	fl_pressure_pmin_ad	-801~800	-800	800	-801	800	800
Judgment			OK	OK	OK	OK	OK

m_Start_Pressure

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(puc_Pressure_flg == 1)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	puc_Pressure_flg	0~3	1	0	2	3
	stp_sub_ad->fl_ec_ad_data[0]	-800~801	-800	800	0	801
Output	ucs_pressure_chk_flg	0~3	1	0	2	3
	uss_pressure_in_count	80,300	80	300	300	300
	uss_pressure_out_count	500	500	500	500	500
	fl_pressure_pmax_ad	-800~900	-800	800	0	801
	fl_pressure_pmin_ad	-800~900	-800	800	0	801
	ucs_pressure_flg	1	1	1	1	1
Judgment			OK	OK	OK	OK

m_Stop_Pressure

	Responsible	Last update
Coding	Ikarashi	2021/6/28
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_pressure_in_count	0	0
	uss_pressure_out_count	0	0
	ucs_pressure_flg	0	0
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc_nc.c

GasCalcNC_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uss_m_nc_warm_count)		○	-	○
	else		-	○	-
	if(uss_m_over_wait_count)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	uss_m_nc_warm_count	0~1501	1500	0	1501
	uss_m_over_wait_count	0~1001	1000	0	1001
Output	uss_m_nc_warm_count	0~1500	1499	0	1500
	uss_m_over_wait_count	0~1000	999	0	1000
Judgment			OK	OK	OK

GasCalcNC_AirCal_HCDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if((stp_denst->fl_air_max >= lfl_sen_data[0]) && (stp_denst->fl_air_min <= lfl_sen_data[0]))		○	○	-	-	-
	if((stp_denst->fl_air_max >= lfl_sen_data[1]) && (stp_denst->fl_air_min <= lfl_sen_data[1]))		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	○	○
	Symbol	Range					
Input	m_Get_NC_ZeroTempComp()	-1.36~1.1	0	1.02	-1.36	1.02	1.1
	stp_sub_ad->fl_nc_ad_data[0][0][1]	-800~801	0	0	-800	800	801
	stp_sub_ad->fl_nc_ad_data[0][1][1]	-800~801	0	-800	-800	800	801
	stp_denst->fl_air_max	0~86	0	85	85	85	86
	stp_denst->fl_air_min	0~66	0	0	65	65	66
	stp_sub_ad->fl_temp_data	-55~86	-55	85	85	85	86
Output	stp_GasInfo_addr->fl_zero_temp_data	0	0	0	0	0	0
	luc_ack	1,248	248	1	1	1	1
Judgment			OK	OK	OK	OK	OK

GasCalcNC_AutoCal_HCDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(lfl_span[0] > 0 || lfl_span[1] > 0)		○	○	○	○	-	-	-
	if((stp_denst->fl_span_max[0] >= lfl_span[0]) && (lfl_span[0] >= stp_denst->fl_span_min[0]))		○	○	-	-	-	-	-
	else		-	-	○	○	-	-	-
	if((stp_denst->fl_span_max[0] >= lfl_span[1]) && (lfl_span[1] >= stp_denst->fl_span_min[0]))		○	-	○	-	-	-	-
	else		-	○	-	○	-	-	-
	if((((HCTableDefData[stp_GasInfo_addr->uc_gas_comb].uc_meas_group & 0x30) == 0x00) && (luc_ack_flg[0] == M_OK || luc_ack_flg[1] == M_OK)) || (((HCTableDefData[stp_GasInfo_addr->uc_gas_comb].uc_meas_group & 0x10) == 0x10) && (luc_ack_flg[0] == M_OK)) ||(((HCTableDefData[stp_GasInfo_addr->uc_gas_comb].uc_meas_group & 0x20) == 0x20) && (luc_ack_flg[1] == M_OK)))		○	○	○	-	-	-	-
	if(luc_ack_flg[0] == M_OK)		○	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	if(luc_ack_flg[1] == M_OK)		○	-	○	-	-	-	-
	else		-	○	-	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	-	○	○	○
	Symbol	Range							
Input	pul_cal_denst	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
	m_Get_NC_ZeroTempComp()	-1.36~1.1	0	0	0	0	-1.36	1.02	1.1
	m_Get_NC_SpanTempComp()	0.99825~1.03665	1	1	1	1	0.99825	1.03665	1
	stp_sub_ad->fl_nc_ad_data[0][0][1]	-800~801	770	770	800	770	-800	800	801
	stp_sub_ad->fl_nc_ad_data[0][1][1]	-800~801	770	800	770	770	-800	800	801
	stp_denst->fl_zero[0]	-800~801	800	800	-800	800	-800	800	801
	stp_denst->fl_zero[1]	-800~801	800	-800	800	800	-800	800	801
	m_Change_LinearRaiseBackOp (stp_GasInfo_addr->uc_lin_number[0], lfl_sp[0])	-50~601	600	600	600	600	-50	0	601
	m_Change_LinearRaiseBackOp (stp_GasInfo_addr->uc_lin_number[0], lfl_sp[1])	-50~601	600	600	600	600	-50	0	601
	stp_denst->fl_span_fix	0~2	1	1	1	1	1	0	2
	stp_GasInfo_addr->us_hc_rate[0]	0~106	100	100	100	100	0	105	106
	stp_denst->fl_span_max[0]	0~9	8	8	8	0	0	8	9
	stp_denst->fl_span_min[0]	0~2	1.6	1.6	1.6	0	0	0	2
	stp_GasInfo_addr->uc_gas_comb	0~30	0	4	4	29	0	0	30
	stp_sub_ad->fl_temp_data	-55~86	-55	85	85	-55	85	85	86
Output	stp_GasInfo_addr->fl_span_temp_data	0,85	0	85	0	0	0	0	0
	stp_data->uc_nc_mode_flg	0	0	0	0	0	0	0	0
	stp_GasInfo_addr->fl_span_buf[0]	0,2	0	0	2	0	0	0	0
	stp_GasInfo_addr->fl_span_buf[1]	0,5	0	0	5	0	0	0	0
	luc_ack	0,1	1	0	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK

GasCalcNC_Calc_NCDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	12
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case10
Root	if((stp_denst+DataStructConst_NC_CH)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)) == 0U)		○	-	-	-	-	-	-	-	-	-	○	○
	else		-	○	○	○	○	○	○	○	○	○	–	–
	if(ucs_m_nc_mode == M_NC_MODE_OVER)		○	○	○	○	-	-	-	-	-	-	–	–
	if(((stp_GasInfo_addr+DataStructConst_O2_GAS_CH)->uc_gas_comb != 0x00) && (!((stp_denst+DataStructConst_O2_GAS_CH)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))))		○	○	○	-	-	-	-	-	-	-	–	–
	if(((stp_denst+DataStructConst_O2_GAS_CH)->uc_m_denst_minus_flg == M_OFF) && ((stp_denst+DataStructConst_O2_GAS_CH)->ul_m_denst >= M_NC_RESET_O2_DENST))		○	○	-	-	-	-	-	-	-	-	–	–
	if(ucs_m_over_extension_start_flg == M_OFF)		○	-	-	-	-	-	-	-	-	-	–	–
	else		-	○	-	-	-	-	-	-	-	-	–	–
	if((uss_m_over_wait_count == 0U) || (uss_m_over_extension_count == 0U))		○	-	-	-	-	-	-	-	-	-	–	–
	else		-	○	-	-	-	-	-	-	-	-	–	–
	else		-	-	○	-	-	-	-	-	-	-	–	–
	else		-	-	-	○	-	-	-	-	-	-	–	–
	else		-	-	-	-	○	○	○	○	○	○	○	○
	if((stp_sub_ad->uc_nc_count != lucs_nc_denst_count_bak) || (stp_sub_ad->uc_nc_Protect_status == M_ON))		-	○	○	○	○	○	-	-	-	-	○	○
	if(ucs_m_nc_mode != M_NC_MODE_OVER)		-	-	-	-	○	○	-	-	-	-	○	○
	if(uss_m_nc_warm_count != 0)		-	-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	○	○
	if((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))		-	-	-	-	-	○	-	-	-	-	○	○
	else		-	-	-	-	-	-	○	○	-	-	-	-
	if((luc_over_mainte_flg == M_OFF) && (stp_sub_ad->uc_nc_Protect_status == M_ON))		-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	○	○	-	-	○	○
	if((ucs_m_nc_output_count >= ucc_nc_output_count_max) && (FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER))		-	-	-	-	-	-	○	○	-	-	○	○
	if(lfl_diff_volt[0] >= flc_nc_diff_limit || lfl_diff_volt[1] >= flc_nc_diff_limit || lfl_diff_volt[2] >= flc_nc_diff_limit || lfl_diff_volt[3] >= flc_nc_diff_limit)		-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	○	-	○	○
	if(stp_GasInfo_addr+DataStructConst_NC_CH) ->uc_follower_onoff == M_ON)		-	-	-	-	-	-	-	○	-	-	-	○
	else		-	-	-	-	-	-	-	-	○	-	○	–
	else		-	-	-	-	-	-	-	-	-	○	-	-
	switch(ucs_m_nc_mode)		-	-	-	-	○	○	-	-	-	-	○	○
	case M_NC_MODE_DIFF		-	-	-	-	-	-	-	○	-	-	-	○
	case M_NC_MODE_WARM		-	-	-	-	○	-	-	○	-	-	○	○
	case M_NC_MODE_OUTPUT		-	-	-	-	○	-	○	○	○	-	○	○
	if(stp_denst->uc_m_over_flg == M_OFF || !((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)))		-	-	-	-	○	-	-	-	○	-	–	–
	else		-	-	-	-	-	-	○	○	-	-	○	○
	if(GasFuncAir_Disp_Get_AirCalFlg() == M_ON)		-	-	-	-	-	-	○	-	-	-	-	○
	else		-	-	-	-	-	-	-	○	-	-	○	–
	case M_NC_MODE_OVER		-	-	-	-	-	○	-	-	-	-	-	-
	default		-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	○	○	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-
	Symbol	Range												
Input	(stp_denst+DataStructConst_NC_CH)->uc_state_trb	0x00,0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	(stp_denst+DataStructConst_O2_GAS_CH) ->uc_state_trb	0x00~0xFF	0x00	0x00	0x00	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	(stp_GasInfo_addr+DataStructConst_O2_GAS_CH) ->uc_gas_comb	1	1	1	1	1	1	1	1	1	1	1	1	1
	(stp_denst+DataStructConst_O2_GAS_CH) ->uc_m_denst_minus_flg	0,1	0	0	1	0	0	0	0	0	0	0	0	0
	(stp_denst+DataStructConst_O2_GAS_CH) ->ul_m_denst	200	200	200	199	200	200	200	200	200	200	200	200	200
	ucs_m_nc_mode	0,3	3	3	3	3	0	0	0	0	0	0	0	0
	ucs_m_over_extension_start_flg	0,1	0	1	0	0	0	0	0	0	0	0	0	0
	uss_m_over_wait_count	0,1	0	1	0	0	0	0	0	0	0	0	0	0
	uss_m_over_extension_count	0~2	0	1	0	0	2	2	2	2	2	2	2	2
	lucs_nc_denst_count_bak	0	0	0	0	0	0	0	0	0	0	0	0	0
	stp_sub_ad->uc_nc_count	0,1	0	1	1	1	1	1	1	1	1	1	1	1
	stp_sub_ad->uc_nc_Protect_status	0,1	0	1	1	1	1	1	0	0	1	1	0	0
	uss_m_nc_warm_count	0	0	0	0	0	1	0	0	0	0	0	0	0
	(stp_denst+DataStructConst_NC_CH) ->uc_m_over_flg	0,1	0	0	0	0	0	0	1	1	0	0	1	1
	FlowMain_Get_ModeStatus()	0~7	0	0	0	0	1	1	0	1	1	7	1	1
	ucs_m_nc_output_count	0~4	0	0	0	0	0	0	4	4	4	0	4	4
	ucc_nc_output_count_max	4	4	4	4	4	4	4	4	4	4	4	4	4
	stp_sub_ad->fl_nc_ad_data[1][0][1]	0,1	0	0	0	0	0	0	1	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[0][0][1]	0	0	0	0	0	0	0	0	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[1][0][2]	0,1	0	0	0	0	0	0	1	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[0][0][2]	0	0	0	0	0	0	0	0	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[1][1][1]	0,1	0	0	0	0	0	0	1	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[0][1][1]	0	0	0	0	0	0	0	0	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[1][1][2]	0,1	0	0	0	0	0	0	1	0	0	0	0	0
	stp_sub_ad->fl_nc_ad_data[0][1][2]	0	0	0	0	0	0	0	0	0	0	0	0	0
	flc_nc_diff_limit	0, 0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0	0.02	0.02
	GasFuncAir_Disp_Get_AirCalFlg()	0,1	0	0	0	0	0	0	1	0	0	0	0	1
	(stp_GasInfo_addr+DataStructConst_NC_CH) ->uc_follower_onoff	0,1	0	0	0	0	0	0	0	1	0	0	0	1
Output	ucs_m_nc_mode	0~3	3	3	3	3	0	3	2	3	3	2	3	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

GasCalcNC_Chk_OffNcPower

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_status->us_main_state & (DataStructConst_FAIL_BTA | DataStructConst_FAIL_BTL)) != 0)		-	○	○
	else		○	-	-
	if((stp_denst->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)) != 0)		-	○	○
	else		○	-	-
	if(luc_flg == M_ON)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0x0003	0xFFFF
	stp_denst->uc_state_trb	0x00~0xFF	0x00	0x14	0xFF
Output	PORT0.PODR.BIT.B3	0	0	0	0
Judgment			OK	OK	OK

GasCalcNC_Chk_SensorFail

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_m_chk_start_flg == M_ON)		○	○	○	-	-
	if(stp_sub_ad->fl_nc_ad_data[0][0][1] > M_NC_SENSOR_CHK_VOLTAGE)		○	○	-	-	-
	if(stp_sub_ad->fl_nc_ad_data[0][1][1] > M_NC_SENSOR_CHK_VOLTAGE)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	uc_m_chk_start_flg	0~2	1	1	1	0	2
	stp_sub_ad->fl_nc_ad_data[0][0][1]	-800~801	800	800	-800	801	-801
	stp_sub_ad->fl_nc_ad_data[0][1][1]	-800~801	800	-800	800	-801	801
Output	luc_ans	0,1	0	1	1	0	0
Judgment			OK	OK	OK	OK	OK

GasCalcNC_Get_LongEnergyFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(GasFuncAir_Disp_Get_AirCalFlg() == M_ON)		○	-	-	-
	else		-	○	○	○
	if(!((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)))		○	-	-	-
	else		-	○	○	○
	if(ucs_m_nc_mode == M_NC_MODE_WARM || ucs_m_nc_mode == M_NC_MODE_OVER)		○	-	○	-
	else		-	○	-	○
	Symbol	Range				
Input	GasFuncAir_Disp_Get_AirCalFlg()	0~2	1	0	0	2
	FlowMain_Get_ModeStatus()	0~8	0	1	7	8
	ucs_m_nc_mode	0~4	0	2	3	4
Output	luc_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

GasCalcNC_Get_NCProtectionCount

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_m_nc_warm_count	0	0
Judgment			OK

GasCalcNC_Get_NCProtectionFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ucs_m_nc_mode == M_NC_MODE_OVER)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ucs_m_nc_mode	0~4	3	0	4
Output	luc_ans	0,1	1	0	0
Judgment			OK	OK	OK

GasCalcNC_Get_NCWarmupFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if (ucs_m_nc_mode==M_NC_MODE_WARM)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	ucs_m_nc_mode	0~2	1	0	2
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

GasCalcNC_Set_NCProtectionFlgOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ucs_m_nc_mode == M_NC_MODE_OVER)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ucs_m_nc_mode	0~4	3	0	4
Output	uss_m_nc_warm_count	0,1500	1500	0	0
	ucs_m_nc_mode	0,4	0	0	4
Judgment			OK	OK	OK

GasCalcNC_Set_SpanMargine

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	for(luc_i=0; luc_i<2; luc_i++)		○	○	○	○	○	○	○	○
	if(lfl_temp1[luc_i] < 0.0F)		○	-	○	○	○	○	○	-
	else		-	○	○	○	○	-	-	○
	if((HCTableDefData[stp_GasInfo_addr->uc_gas_comb]. uc_meas_group & 0x30) == 0x00)		○	○	○	○	○	-	-	-
	if((lfl_temp1[0] >= lfl_temp1[1]) && (luc_minus_flg[0] == M_OFF) && (luc_minus_flg[1] == M_OFF))		○	-	-	-	-	-	-	-
	else if((lfl_temp1[0] <= lfl_temp1[1]) && (luc_minus_flg[0] == M_ON) && (luc_minus_flg[1] == M_ON))		-	○	-	-	-	-	-	-
	else if((luc_minus_flg[0] == M_OFF) && (luc_minus_flg[1] == M_ON))		-	-	○	○	-	-	-	-
	else		-	-	-	-	○	-	-	-
	else if((HCTableDefData[stp_GasInfo_addr->uc_gas_comb]. uc_meas_group & 0x20) == 0x20)		-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	○	○
	if(FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER)		○	-	○	-	-	-	-	-
	else		-	○	-	○	○	○	○	○
	if(lfl_temp2 <= M_NC_LINEAR_MAX)		-	-	○	○	-	-	-	-
	if(FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER)		-	-	○	-	-	-	-	-
	else		-	-	-	○	-	-	-	-
	else		○	○	-	-	○	○	○	○
	if(lfl_denst > stp_GasInfo_addr->ul_full_scale)		-	○	○	○	-	○	○	○
	else		○	-	-	-	○	-	-	-
	if(lfl_denst > (stp_GasInfo_addr->ul_cal_denst * M_NC_CALDENST_MAG))		-	○	○	○	-	○	○	-
	else		○	-	-	-	○	-	-	○
	Symbol	Range								
Input	m_Get_NC_ZeroTempComp()	-1.36~1.1	0	0	0	0	0	-1.36	1.02	1.1
	m_Get_NC_SpanTempComp()	0.99825~1.1	1	1	1	1	1	0.99825	1.03665	1.1
	stp_sub_ad->fl_nc_ad_data[0][0][2]	-800~801	800	800	800	800	800	-800	-800	801
	stp_sub_ad->fl_nc_ad_data[0][1][2]	-800~801	800	800	800	800	800	-800	-800	801
	m_Get_NC_ZeroHumComp (stp_status->fl_absolute_humidity[0])	-800~801	800	-800	800	800	-800	800	800	801
	m_Get_NC_ZeroHumComp (stp_status->fl_absolute_humidity[1])	-800~801	800	-800	-800	-800	800	800	800	801
	stp_GasInfo_addr->uc_gas_comb	0~30	0	0	0	0	29	3	4	30
	FlowMain_Get_ModeStatus()	0~8	0	0	7	7	7	7	7	8
	stp_denst->fl_span_min[luc_flg]	0~1.7	0	1.6	1.6	1.6	1.6	1.6	1.6	1.7
	stp_GasInfo_addr->us_hc_rate[0]	0~121	0	120	120	120	120	120	120	121
	stp_GasInfo_addr->us_hc_rate[1]	0~121	0	120	120	120	120	120	120	121
	m_Change_LinearRaiseOp()	-50~601	0	600	600	600	600	600	600	601
	stp_GasInfo_addr->fl_lel_max_ratio[0]	0~60001	0	60000	60000	60000	60000	600000	600000	60001
	stp_GasInfo_addr->fl_lel_ratio[0]	0~50001	0	50000	50000	50000	50000	500000	500000	50001
	stp_denst->fl_span_fix	0~2	0	1	0	0	1	1	1	2
	stp_GasInfo_addr->ul_full_scale	100~501	100	500	500	500	500	500	500	501
	stp_GasInfo_addr->ul_cal_denst	0x00000000,0xFFFFFFFF	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	stp_GasInfo_addr->ul_digit	1~6	1	1	5	5	5	5	5	6
Output	stp_GasInfo_addr->ul_span_margin	0~4294967295	4294967295	0	0	0	0	0	0	498
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

GasCalcNC_Start_ChkFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_chk_start_flg	1	1
Judgment			OK

m_Calc_NC_DiffMode

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(lfl_temp_diff < 0)		○	-
	else		-	○
	for(luc_i=0; luc_i<2; luc_i++)		○	○
	if(lfl_sen_ad[0][luc_i] < (lfl_vol_diff_lim[0] + stp_GasInfo_addr->fl_zero_buf[luc_i]))		○	○
	if(lfl_sen_ad[0][luc_i] > (lfl_vol_diff_lim[1] + stp_GasInfo_addr->fl_zero_buf[luc_i]))		○	○
	else		-	-
	else		-	-
	Symbol	Range		
Input	fl_absolute_humidity[0]	0	0	0
	fl_absolute_humidity[1]	0	0	0
	fl_zero_temp_data	0	0	0
	fl_zero_buf[0]	0	0	0
	m_Get_NC_ZeroHumComp()	0	0	0
	m_Get_NC_ZeroTempComp()	0	0	0
	fl_nc_ad_data[0][0][2]	0	0	0
	fl_nc_ad_data[0][1][2]	0	0	0
	fl_nc_ad_data[2][0][2]	0	0	0
	fl_nc_ad_data[2][1][2]	0	0	0
	fl_temp_data	-1,0	-1	0
Output	fl_zero	0	0	0
Judgment			OK	OK

m_Calc_AbsoluteHumidity

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0U; luc_i<2U; luc_i++)		○	○	○	-	○
	if((stp_sub_ad->fl_temp_data < flc_nc_temp_ref) || (stp_GasInfo_addr->fl_zero_temp_data < flc_nc_temp_ref))		○	-	-	-	-
	else if(lfl_humidity < M_NC_HUMIDITY_MIN)		-	-	-	-	-
	else if(lfl_humidity > M_NC_HUMIDITY_MAX)		-	○	○	○	○
	else		-	-	-	-	-
	Symbol	Range					
Input	fl_zero_temp_data	0~65535	0	65535	100	200	300
	fl_zero_humidity_voltage[0]	0^65535	0	65535	100	200	300
	m_Get_NC_1000msTempComp()	0~65535	0	65535	100	200	300
	fl_nc_ad_data[0][0][0]	0~65535	0	65535	100	200	300
	fl_nc_ad_data[0][1][0]	0~65535	0	65535	100	200	300
	fl_temp_data	0~65535	0	65535	100	200	300
Output	fl_absolute_humidity	0,100	0	100	100	100	100
Judgment			OK	OK	OK	OK	OK

m_Calc_NC_OutPutMode

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((lfl_sen_ad[0] <= M_NC_RETURN_DIFF) && (lfl_sen_ad[1] <= M_NC_RETURN_DIFF))		-	-	○	○
	if(++ucs_m_nc_output_count > M_NC_OUTPUT_COUNT_MAX)		-	-	○	-
	else		-	-	-	○
	else		○	○	-	-
	Symbol	Range				
Input	stp_sub_ad->fl_nc_ad_data[0][0][1]	-800~801	-800	-800	800	801
	stp_sub_ad->fl_nc_ad_data[0][1][1]	-800~801	-800	-800	800	801
	stp_denst->fl_zero[0]	65~86	65	85	64	86
	stp_denst->fl_zero[1]	65~86	65	85	86	64
	ucs_m_nc_output_count	0~6	5	6	4	3
Output	ucs_m_nc_output_count	0,4	0	0	4	4
Judgment			OK	OK	OK	OK

m_Change_LinearRaiseBackOp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pfl_x_data > 10000)		○	-	-
	else		-	○	○
	for(luc_i = 0; luc_i < 10; luc_i++)		-	○	○
	if((pfl_x_data >= (FLOAT)(luc_i * 1000)) && (pfl_x_data <= (FLOAT)((luc_i+1) * 1000)))		-	○	-
	else		-	○	○
	Symbol	Range			
Input	pfl_x_data	-1~101	101	100	-1
	puc_num	0~7	0	6	7
Output	lfl_ret_ans	0~1110	101	100	1110
Judgment			OK	OK	OK

m_Change_LinearRaiseOp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0; luc_i < 9; luc_i++)		○	○	○
	if((pfl_y_data >= (FLOAT)usc_GasCalc_HC_inearize_y_data[puc_num][luc_i]) && (pfl_y_data <= (FLOAT)usc_GasCalc_HC_inearize_y_data[puc_num][luc_i + 1]))		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_num	0~7	0	6	7
	pfl_y_data	0~101	0	100	101
Output	lfl_ret_ans	0~101	0	100	101
Judgment			OK	OK	OK

m_Get_NC_100msTempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	None		-	-	-	-
	Symbol	Range				
Input	fl_temp_data	-55~85	0	85	-55	-56
Output	lfl_ans_num	-21.2171~0.996	-4.1125	0.996	-20.816	-21.2171
Judgment			OK	OK	OK	OK

m_Get_NC_SpanHumComp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	None		○	-	-	-
	Symbol	Range				
Input	pfl_absolute_humidity	-55~85	0	85	-55	-56
Output	lfl_ans_num	-0.168~0.255	0	0.255	-0.165	-0.168
Judgment			OK	OK	OK	OK

m_Get_NC_SpanTempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((lfl_ans_num < 2E-10F) && (lfl_ans_num > -2E-10F))		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	stp_sub_ad->fl_temp_data	-55~85	0	85	-55	-56
Output	lfl_ans_num	0.99905~1.2765	0.99905	1.27625	1.20625	1.2128
Judgment			OK	OK	OK	OK

m_Get_NC_ZeroHumComp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_sub_ad->fl_temp_data	-55~86	-55	85	86
Output	lfl_ans_num	-0.46182~0.29535	0.29535	-0.45645	-0.46182
Judgment			OK	OK	OK

m_Get_NC_ZeroTempComp

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_sub_ad->fl_temp_data	-55~85	-55	85	86
Output	lfl_ans_num	-0.9567999~1.701473	-0.9567999	1.6668	1.701473
Judgment			OK	OK	OK

m_Set_NC_Denst

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if((stp_GasInfo_addr->uc_suppress_onoff == M_ON) && (FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER))		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○
	for(luc_i=0; luc_i<2; luc_i++)		○	○	○	○	○	○	○	○	○
	if(lfl_temp1[luc_i] < 0.0F)		○	-	-	○	○	-	○	-	-
	else		-	○	○	○	○	○	-	○	○
	if((HCTableDefData[stp_GasInfo_addr->uc_gas_comb].uc_meas_group & 0x30) == 0x00)		○	○	○	○	○	-	○	○	○
	if((HCTableDefData[stp_GasInfo_addr->uc_sub_comb].uc_meas_group & 0x20) == 0x20)		-	-	-	-	-	-	-	○	-
	else		○	○	○	○	○	-	-	-	-
	if((lfl_temp1[0] >= lfl_temp1[1]) && (luc_minus_flg[0] == M_OFF) && (luc_minus_flg[1] == M_OFF))		○	-	-	-	-	-	-	-	-
	else if((lfl_temp1[0] <= lfl_temp1[1]) && (luc_minus_flg[0] == M_ON) && (luc_minus_flg[1] == M_ON))		-	○	○	-	-	-	-	-	-
	else if((luc_minus_flg[0] == M_OFF) && (luc_minus_flg[1] == M_ON))		-	-	-	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-
	else if((HCTableDefData[stp_GasInfo_addr->uc_gas_comb].uc_meas_group & 0x20) == 0x20)		-	-	-	-	-	-	○	-	○
	else		-	-	-	-	-	○	-	-	-
	if(FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER)		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○
	if(lfl_temp2 <= M_NC_LINEAR_MAX)		○	-	○	-	-	○	-	○	-
	if(FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER)		○	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	○	-	○	-
	else		-	○	-	○	○	-	○	-	○
	if(FlowMain_Get_ModeStatus() <= FlowMain_DISP_MODE_NUMBER)		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○
	if(lfl_denst > (stp_GasInfo_addr->ul_full_scale * M_NC_CALC_MAX / 100))		-	○	-	○	-	-	-	-	-
	else		○	-	○	-	○	○	○	○	○
	if((lfl_denst < lul_z_sup2) && (luc_minus_flg[luc_flg] == M_OFF))		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	○
	if(luc_minus_flg[luc_flg] == M_ON)		-	○	○	-	-	○	-	○	○
	if(stp_data->uc_long_energy_flg == M_ON)		-	○	○	-	-	-	-	○	-
	if(luc_minus_over_flg == M_ON)		-	○	-	-	-	-	-	○	-
	if(lul_denst > (stp_GasInfo_addr->ul_full_scale * M_NC_CALC_MAX / 100))		-	○	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	○	-
	else		-	-	-	-	-	○	-	-	-
	if((luc_minus_flg[luc_flg] == M_ON) && (lul_denst > (stp_GasInfo_addr->ul_full_scale / 5UL)))		-	-	○	○	○	○	○	-	○
	else		○	○	-	-	-	-	-	○	-
	else		-	-	-	-	-	○	-	-	○
	if(luc_minus_over_flg == M_ON)		-	-	-	-	-	-	-	-	○
	else		-	-	-	-	-	○	-	-	-
	else		○	-	-	○	○	-	○	-	-
	Symbol	Range									
Input	stp_GasInfo_addr->uc_suppress_onoff	0~2	1	0	0	0	0	0	0	0	2
	stp_GasInfo_addr->ul_full_scale	100~501	100	100	100	100	100	100	100	500	501
	stp_GasInfo_addr->ul_upper_limit	0~501	0	100	100	100	100	100	100	500	501
	stp_sub_ad->fl_nc_ad_data[0][0][2]	-800~801	99	800	1	0	130	1	-800	100	801
	stp_sub_ad->fl_nc_ad_data[0][1][2]	-800~801	99	800	1	130	0	1	-800	100	801
	m_Get_NC_ZeroHumComp (stp_status->fl_absolute_humidity[0])	0~130	100	0	0	130	0	0	85	65	64
	m_Get_NC_ZeroHumComp (stp_status->fl_absolute_humidity[1])	0~130	100	0	0	0	130	0	65	85	64
	stp_GasInfo_addr->uc_gas_comb	0~29	29	29	1	1	1	4	3	0	3
	stp_GasInfo_addr->uc_sub_comb	0~30	0	0	0	0	0	0	29	3	30
	stp_denst->fl_span_fix	1~6.7	1	1	1	1	1	1	6.7	1.2	6.7
	stp_GasInfo_addr->fl_span_buf[0]	4.2,4.3	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.3
	stp_GasInfo_addr->fl_span_buf[1]	4.2,4.3	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.3
	stp_GasInfo_addr->us_hc_rate[0]	0~10000	120	120	120	120	120	120	0	10000	121
	stp_GasInfo_addr->us_hc_rate[1]	0~10000	120	120	120	120	120	120	0	10000	121
	stp_GasInfo_addr->fl_lel_max_ratio[0]	0~50001	0	50000	50000	50000	50000	50000	50000	1000	50001
	stp_GasInfo_addr->fl_lel_max_ratio[1]	0~50001	0	50000	50000	50000	50000	50000	50000	1000	50001
	stp_GasInfo_addr->fl_lel_ratio[0]	0~44001	100	0	44000	44000	44000	44000	44000	100	44001
	stp_GasInfo_addr->fl_lel_ratio[1]	0~44001	100	0	44000	44000	44000	44000	44000	100	44001
	stp_GasInfo_addr->ul_digit	1~6	5	1	1	1	1	1	1	1	6
	stp_data->uc_long_energy_flg	0~2	0	1	1	1	0	0	0	1	2
	m_Change_LinearRaiseOp()	0~130	0	100	0	130	130	0	0	5	0
	m_Get_NC_SpanTempComp()	0.99825~1.04	0.99825	1.03665	1	1	1	1	1	1	1.04
	m_Get_NC_ZeroTempComp()	-1.36~1.03	-1.36	1.02	0	0	0	0	0	0	1.03
	FlowMain_Get_ModeStatus()	0~8	0	3	3	3	3	3	3	7	8
Output	stp_denst->ul_m_denst	0~500	0	120	0	120	120	0	120	500	100
	stp_denst->uc_m_over_flg	0,1	0	1	0	1	1	0	1	0	0
	stp_denst->uc_m_over_minus_flg	0,1	0	0	0	0	0	0	0	0	1
	stp_denst->uc_m_denst_minus_flg	0,1	0	0	1	0	0	1	0	0	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_NC_OffMode

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_GasInfo_addr->ul_full_scale	100~501	100	500	501
	stp_GasInfo_addr->ul_digit	1~6	1	5	6
Output	uss_m_over_wait_count	3000	3000	3000	3000
	uss_m_nc_warm_count	0	0	0	0
	stp_denst->uc_m_over_flg	1	1	1	1
	stp_denst->uc_m_over_minus_flg	0	0	0	0
	stp_denst->uc_m_denst_minus_flg	0	0	0	0
	stp_denst->ul_m_denst	101~507	101	505	507
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_calc_tbl.c

GasCalcTbl_Get_RomNum

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_m_gas_table_rom_num[0]	7328	7328
Judgment			OK

GasCalcTbl_Get_SumNum

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_m_gas_table_sum_num[0]	38394	38394
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_func_air.c

GasFuncAir_Calc_1sec_AirCal

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	14

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case4
Root	if(uc_air_calc_on_flg != M_OFF)		○	-	○	○	○	○	○	○	○	○	○	○	○	○
	if(uc_air_re_count[0])		○	-	-	○	○	○	○	○	○	○	○	○	○	○
	if(GasCalcNC_Get_NCProtectionFlg() == M_ON)		-	-	-	○	-	-	-	-	-	-	-	-	-	-
	else if(GasCalcNC_Get_NCProtectionCount() == 0)		○	-	-	-	○	○	○	○	○	○	○	○	○	○
	if(uc_nc_air_re_count == 0)		○	-	-	-	-	○	○	○	○	○	○	○	-	○
	if(luc_ack == M_NG)		○	-	-	-	-	-	○	○	○	○	○	○	-	○
	if(uc_air_re_count[0] == 0)		○	-	-	-	-	-	-	○	○	○	○	○	-	○
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-
	if(uc_air_re_count[1])		○	-	○	○	○	○	○	-	-	-	-	○	○	○
	if(luc_ack == M_NG)		○	-	○	○	○	○	○	-	-	-	-	○	○	-
	if(uc_air_re_count[1] == 0)		○	-	○	○	○	○	○	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_air_re_count[2])		○	-	○	○	○	○	○	-	-	-	-	○	○	○
	if(luc_ack == M_NG)		○	-	○	○	○	○	○	-	-	-	-	○	○	○
	if(uc_air_re_count[2] == 0)		○	-	○	○	○	○	○	-	-	-	-	-	○	○
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(uc_air_re_count[3])		○	-	○	○	○	○	○	-	-	-	-	○	○	○
	if((stp_GasInfo_addr+3)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		○	-	○	○	○	○	○	-	-	-	-	○	○	○
	if(luc_ack == M_NG)		○	-	○	○	○	○	○	-	-	-	-	○	○	○
	if(uc_air_re_count[3] == 0)		-	-	○	-	-	○	○	-	-	-	-	-	○	○
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if((uc_air_re_count[0] == 0) && (uc_air_re_count[1] == 0) && (uc_air_re_count[2] == 0) && (uc_air_re_count[3] == 0) && (uc_air_re_count[4] == 0))		○	-	○	○	○	○	○	○	○	○	○	-	○	○
	if(uc_air_end_count == 0)		○	-	○	-	-	○	-	○	○	○	-	-	-	○
	if(uc_air_end_count != 0)		-	-	-	-	-	-	-	-	-	-	○	-	-	-
	Symbol	Range														
Input	uc_air_calc_on_flg	0,1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	uc_air_re_count[0]	0~2	1	1	0	1	1	1	2	1	1	1	1	2	1	1
	uc_air_re_count[1]	0~2	1	1	1	1	1	1	1	0	0	0	0	2	1	1
	uc_air_re_count[2]	0~2	1	1	1	1	1	1	1	0	0	0	0	2	1	1
	uc_air_re_count[3]	0~2	1	1	1	1	1	1	1	0	0	0	0	2	1	1
	GasCalcNC_Get_NCProtectionFlg()	0~2	2	2	2	1	2	2	2	2	2	2	2	2	2	2
	GasCalcNC_Get_NCProtectionCount()	0,1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	uc_nc_air_re_count	0,1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	GasCalcNC_AirCal_HCDenst()	1,2	1	1	1	1	1	2	1	1	1	1	1	1	1	1
	uc_gas_comb	0~4	3	3	3	3	3	3	3	3	4	4	4	3	3	3
	GasCalcECDenst_AirCal_Denst()	0~2	1	1	1	1	1	1	1	2	2	1	2	1	1	1
	GasCalcECH2CancelCO_Set_AirCal()	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GasCalcECO2_Air_Cal_O2Denst	0~2	1	1	1	1	1	1	1	2	2	2	2	1	1	2
	uc_air_end_count	0,1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Output	uc_air_calc_on_flg	0,1	0	0	0	1	1	0	1	0	0	0	1	1	1	0
	uc_air_end_count	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_air_re_count[0]	0,1	0	1	0	1	1	0	1	0	0	0	0	1	1	0
	uc_nc_air_re_count	0~5	5	0	0	0	0	0	5	5	5	5	5	5	0	5
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

GasFuncAir_Calc_Get_AirCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_air_calc_on_flg	32	32
Judgment			OK

GasFuncAir_Calc_Set_AirCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)) == 0))		○	○	○	○
	if((luc_i == DataStructConst_CO_CAN_H2_H2_CH) && ((stp_GasInfo_addr+luc_i)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM))		-	-	-	-
	else		○	○	○	○
	if(uc_air_re_count[luc_i] != 0) uc_air_calc_on_flg |= (0x01 << luc_i)		○	○	○	○
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	3	0	29	30
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0xFF	0x00	0x00
	stp_data->uc_ir_air_onoff	0	0	0	0	0
Output	uc_air_calc_on_flg	6,7	7	6	7	7
	uc_air_re_count[luc_i]	0,4	4	0	4	4
	uc_nc_air_re_count	0	0	0	0	0
	uc_air_end_count	5	5	5	5	5
Judgment			OK	OK	OK	OK

GasFuncAir_Disp_Get_AirCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_air_disp_on_flg	0	0
Judgment			OK

GasFuncAir_Disp_Set_AirCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	0	1	2
Output	uc_air_disp_on_flg	0~2	0	1	2
Judgment			OK	OK	OK

GasFuncAir_Set_AirCalErrorOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	for(luc_i = 0U; luc_i < HDDEF_GAS_MAX; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	gas_func_autocal.c

GasFuncAutoCal_Calc_1sec_AutoCal

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	13
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	if(uc_auto_calc_on_flg != M_OFF)	○	-	○	○	○	○	○	○	○	○	○	○	○
	if(uc_auto_re_count[0] == M_ON)	○	-	-	○	○	○	○	○	○	○	○	○	○
	if(ul_auto_cal_denst[0] != M_OFF_VAL)	○	-	-	-	○	○	○	○	○	○	○	○	○
	if(luc_ack == M_NG)	○	-	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	○	○	○	○	○	○	○	○	○
	if(uc_auto_re_count[1] == M_ON)	○	-	○	○	○	-	○	○	○	○	○	○	-
	if(ul_auto_cal_denst[1] != M_OFF_VAL)	○	-	○	○	○	-	○	○	○	○	○	○	-
	if(luc_ack == M_NG)	○	-	○	○	○	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	○	○	○	○	○	○	-
	if(uc_auto_re_count[2] == M_ON)	○	-	○	○	○	○	○	○	○	○	○	○	-
	if(ul_auto_cal_denst[2] != M_OFF_VAL)	○	-	○	○	○	○	○	-	○	○	○	○	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_H2_CH) -> uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	○	-	○	○	○	○	○	-	○	○	○	○	-
	if(luc_ack == M_NG)	○	-	○	○	○	○	○	-	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	○	○	-
	if(uc_auto_re_count[3] == M_ON)	○	-	○	○	○	○	○	○	○	○	○	○	-
	if(ul_auto_cal_denst[3] != M_OFF_VAL)	○	-	○	○	○	○	○	○	○	○	-	○	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_CO_CH) -> uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)	-	-	-	-	-	-	-	-	-	-	-	-	-
	else	○	-	○	○	○	○	○	○	○	○	-	○	-
	if(luc_ack == M_NG)	○	-	○	○	○	○	○	○	○	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	○	-	○	-
	if((uc_auto_re_count[0] == M_OFF) && (uc_auto_re_count[1] == M_OFF) && (uc_auto_re_count[2] == M_OFF) && (uc_auto_re_count[3] == M_OFF))	○	-	-	-	○	-	○	-	○	○	-	○	○
	Symbol	Range												
Input	uc_gas_comb	3,4	3	3	3	3	3	3	3	3	4	4	4	3	3
	GasCalcECDenst_AutoCal_Denst()	1,2	1	1	1	1	1	1	1	1	1	2	2	2	2
	GasCalcECH2CancelCO_Set_AutoCal()	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GasCalcECO2_AutoCal_O2Denst()	1,2	1	1	1	1	1	2	2	2	2	2	2	2	2
	gasCalcNC_AutoCal_HCDenst()	1,2	1	1	1	1	2	2	2	2	2	2	2	2	2
	uc_auto_calc_on_flg	0,1	1	0	1	1	1	1	1	1	1	1	1	1	1
	uc_auto_re_count[0]	1,2	1	1	2	1	1	1	1	1	1	1	1	1	1
	uc_auto_re_count[1]	0,1	1	1	1	1	1	1	1	1	1	1	1	1	0
	uc_auto_re_count[2]	0,1	1	1	1	1	1	1	1	1	1	1	1	1	0
	uc_auto_re_count[3]	0,1	1	1	1	1	1	1	1	1	1	1	1	1	0
	ul_auto_cal_denst[0]	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ul_auto_cal_denst[1]	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ul_auto_cal_denst[2]	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	ul_auto_cal_denst[3]	0xDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE
Output	uc_auto_calc_on_flg	0,1	0	0	1	1	0	1	0	1	0	0	1	0	0
	uc_auto_re_count[0]	0~2	0	1	2	1	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

GasFuncAutoCal_Calc_Get_AutoCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	1
Judgment	OK

		case1
Root	None	-
	Symbol	Range	
Input	None	-	-
Output	uc_auto_calc_on_flg	32	32
Judgment			OK

GasFuncAutoCal_Calc_Set_AutoCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

		case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((stp_GasInfo_addr+luc_i)->ul_cal_denst != M_OFF_VAL))	○	○	○	-
	if((stp_GasInfo_addr+luc_i)->uc_cal_group == puc_group)	○	○	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))	○	-	-	-
	else	-	○	-	-
	else	-	-	○	-
	else	-	-	-	○
	if(uc_auto_re_count[luc_i] == M_ON)	○	-	-	-
	else	-	○	○	○
	Symbol	Range			
Input	puc_group	0x00,0xFF	0x00	0xFF	0x00	0x00
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	30	0
	(stp_GasInfo_addr+luc_i)->ul_cal_denst	0~0xDDDDDDDD	0	50000	50001	0xDDDDDDDD
	(stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0x00	0xFF	0xFF	0x00
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0xFF	0xFF	0x00
Output	stp_data->uc_up_cal_flg	0	0	0	0	0
	uc_auto_re_count[luc_i]	1	1	1	1	1
	uc_auto_calc_on_flg	3	3	3	3	3
Judgment			OK	OK	OK	OK

GasFuncAutoCal_Calc_Set_BumpCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((stp_GasInfo_addr+luc_i)->ul_cal_denst != M_OFF_VAL))		○	○	○	○	-
	if((stp_GasInfo_addr+luc_i)->uc_cal_group == puc_group)		○	○	○	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_BMP)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	if(uc_auto_re_count[luc_i] == M_ON)		○	-	-	-	-
	else		-	○	○	○	○
	Symbol	Range					
Input	puc_group	0x00~0xFF	0x00	0xFF	0x00	0x00	0x00
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	30	29	0
	(stp_GasInfo_addr+luc_i)->ul_cal_denst	0~0xDDDDDDDD	0	50000	50001	0	0xDDDDDDDD
	(stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0x00	0xFF	0x00	0xFF	0xFF
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x80	0x00	0xFF	0x00	0x00
Output	uc_auto_calc_on_flg	1	1	1	1	1	1
	uc_auto_re_count[luc_i]	1	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK

GasFuncAutoCal_Calc_Set_ComAutoCalFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((stp_GasInfo_addr+luc_i)->ul_cal_denst != M_OFF_VAL))		○	○	○	-
	if(puc_flg & (0x01 << luc_i))		○	○	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	-	-	-
	if(uc_auto_re_count[luc_i] == M_ON)		○	-	-	-
	Symbol	Range				
Input	puc_flg	0x00,0xFF	0x00	0xFF	0x00	0x00
	*pucp_serial_p	0~2	1	1	0	2
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	30	0
	(stp_GasInfo_addr+luc_i)->ul_cal_denst	0~0xDDDDDDDD	0	50000	50001	0xDDDDDDDD
	(stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0x00	0xFF	0xFF	0x00
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0xFF	0xFF	0x00
Output	stp_data->uc_up_cal_flg	0,253	0	253	253	0
	uc_auto_re_count[luc_i]	1	1	1	1	1
Judgment			OK	OK	OK	OK

GasFuncAutoCal_Get_AutoCalGroup

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_auto_cal_group	0	0
Judgment			OK

GasFuncAutoCal_Get_AutoCalSDMSelear

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasFuncAutoCal_Set_AutoCalErrorOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(luc_i = 0U; luc_i < HDDEF_GAS_MAX; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasFuncAutoCal_Set_AutoCalGroup

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_group	0x00,0xFF	0x00	0xFF
Output	uc_auto_cal_group	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

GasFuncAutoCal_Set_AutoCalSDMSelear

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

GasFuncAutoCal_Set_ComAutoCalDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~5	0	4	5
	pul_denst	0~50001	0	50000	50001
Output	ul_com_auto_cal_denst[puc_flg]	0	0	0	0
Judgment			OK	OK	OK

m_Set_AutoCalDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_flg == 0)		○	○	-	-
	if((stp_GasInfo_addr+2)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-
	else		-	○	-	-
	if((stp_GasInfo_addr+3)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	puc_flg	0~2	0	0	1	2
	(stp_GasInfo_addr+2)->uc_gas_comb	0~30	4	0	29	30
	(stp_GasInfo_addr+3)->uc_gas_comb	0~30	4	0	29	30
	(stp_GasInfo_addr+0)->ul_cal_denst	0~251	250	0	0	251
	(stp_GasInfo_addr+1)->ul_cal_denst	0~121	120	0	0	121
	(stp_GasInfo_addr+2)->ul_cal_denst	0~501	500	0	0	501
	(stp_GasInfo_addr+3)->ul_cal_denst	0~801	800	0	0	801
Output	ul_auto_cal_denst[0]	0	0	0	0	0
	ul_auto_cal_denst[1]	0	0	0	0	0
	ul_auto_cal_denst[2]	0	0	0	0	0
	ul_auto_cal_denst[3]	0	0	0	0	0
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	interval_timer.c

IntervalTimer_Flag_Acquisition_100msec

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_100msec_flag_time	0	0
Judgment			OK

IntervalTimer_Flag_Acquisition_250msec

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_250msec_flag_time	0	0
Judgment			OK

IntervalTimer_Flag_Set_100msec

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~2	0	1	2
Output	uc_m_100msec_flag_time	0~2	0	1	2
Judgment			OK	OK	OK

IntervalTimer_Flag_Set_250msec

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~2	0	1	2
Output	uc_m_250msec_flag_time	0~2	0	1	2
Judgment			OK	OK	OK

IntervalTimer_Int_CMT_Start

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

IntervalTimer_Int_CMT_Stop

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

IntervalTimer_Int_Timer_CMT0_Op

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(--lucs_t250m == 0)		-	○	-	-	-
	else		○	-	○	○	○
	if(--lucs_t100m == 0)		-	○	-	-	-
	else		○	-	○	○	○
	if(--lucs_t1000m == 0)		-	○	-	-	-
	else		○	-	○	○	○
	Symbol	Range					
Input	lucs_t250m	0~26	0	1	2	25	26
	lucs_t100m	0~11	0	1	2	10	11
	lucs_t1000m	0~101	0	1	2	100	101
Output	lucs_t250m	1~255	255	25	1	24	24
	lucs_t100m	1~255	255	10	1	9	10
	lucs_t1000m	1~255	255	100	1	99	100
	uc_m_250msec_flag_time	0,1	0	1	1	1	1
	uc_m_100msec_flag_time	0,1	0	1	1	1	1
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	key_event.c

KeyEvent_Chk_DispPowerOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(uc_turn_off_flg == M_ON)		○	○	-	-
	if(uc_turn_off_stop_flg == M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	uc_turn_off_flg	0~2	1	1	0	2
	uc_turn_off_stop_flg	0~2	1	0	0	2
Output	luc_ans	0,1	0	1	0	0
Judgment			OK	OK	OK	OK

KeyEvent_Chk_KeyPowerOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(uc_m_switch_buffer != M_MODE_KEY)		○	-	-	○
	else		-	○	○	-
	if(uc_off_disp_count > 3)		-	○	-	-
	else		-	-	○	-
	Symbol	Range				
Input	uc_m_switch_buffer	0x00~0xFF	0x00	0x01	0x01	0xFF
	uc_off_disp_count	0~4	0	3	0	4
Output	uc_off_disp_count	0~4	0	4	1	0
	uc_turn_off_flg	0,1	0	1	0	0
Judgment			OK	OK	OK	OK

KeyEvent_Get_ForcedPowerOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_turn_off_stop_flg	2	2
Judgment			OK

KeyEvent_Get_KeyEvent

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ss_event_zipc	0	0
Judgment			OK

KeyEvent_Get_KeyPort

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

KeyEvent_Get_KeyPowerOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_turn_off_flg	0	0
Judgment			OK

KeyEvent_Get_ReleaseKeyFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_key_release_on_flg	0	0
Judgment			OK

KeyEvent_Get_TimeNum

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	us_timer_count	0	0
Judgment			OK

KeyEvent_no_op_time_set

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(us_no_op_timer_count != 0)		-	○
	else		○	-
	Symbol	Range		
Input	us_no_op_timer_count	0x0000,0xFFFF	0x0000	0xFFFF
Output	uc_no_op_timer_count_flg	0,1	0	1
Judgment			OK	OK

KeyEvent_Reset_KeyTime

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	us_m_switch_on_time	0	0
Judgment			OK

KeyEvent_Scan_KeyPort

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(us_timer_count)		○	-	-	○
	if(lucs_1sec_counter == 0)		○	○	-	-
	if(us_no_op_timer_count)		○	-	-	-
	else		-	-	○	○
	if(luc_KeyBuf != luc_KeyState)		○	-	-	○
	else		-	○	○	-
	Symbol	Range				
Input	us_timer_count	0x0000,0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF
	luc_KeyState	0,1	1	0	0	0
	lucs_1sec_counter	0~101	0	0	100	101
	us_no_op_timer_count	0x0000,0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF
	m_Get_KeyIo()	0~3	2	0	0	3
Output	uc_m_key_data	0,32	32	0	32	32
	us_no_op_timer_count	0~65535	65534	0	0	65535
	us_timer_count	0,65534	65534	0	0	65534
	luc_KeyState	0~3	2	0	0	3
	lucs_1sec_counter	99,100	100	100	99	100
	uc_m_switch_buffer	128	128	128	128	128
Judgment			OK	OK	OK	OK

KeyEvent_Set_ForcedPowerOff

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_onoff == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	luc_onoff	0~2	0	1	2
Output	uc_turn_off_stop_flg	0,1	0	1	1
Judgment			OK	OK	OK

KeyEvent_time_set

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(pus_count != 0)		○	-
	else		-	○
	Symbol	Range		
Input	pus_count	0x0000,0xFFFF	0xFFFF	0x0000
Output	uc_timer_count_flg	0,1	1	0
Judgment			OK	OK

m_Get_KeyIo

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	PORT1.PIDR.BIT.B4	0,1	0	1
	PORT1.PIDR.BIT.B5	0,1	0	1
Output	luc_dummy	0~3	3	0
Judgment			OK	OK

KeyEvent_Make_Event

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	14

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14
Root	if(luc_num)		○	○	-	-	-	-	-	-	-	-	-	○	-	-
	if((uc_m_switch_buffer == M_REMOVE_KEY) && (uc_key_release_on_flg == 1))		○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	○	○	○	○	○	○	○	○	○	-	○	○
	if((uc_no_op_timer_count_flg == M_ON) && (us_no_op_timer_count == 0))		-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else if((uc_timer_count_flg == M_ON) && (us_timer_count == 0))		-	-	-	-	-	-	-	-	-	-	-	-	○	-
	else		-	-	-	○	○	○	○	○	○	○	○	-	-	○
	switch(uc_m_switch_buffer)		-	-	-	○	○	○	○	○	○	○	○	-	-	○
	case M_MODE_KEY:		-	-	-	-	○	-	-	-	-	-	○	-	-	○
	if(uc_m_switch_buffer == M_MODE_KEY)		-	-	-	-	○	-	-	-	-	-	○	-	-	○
	case M_AIR_KEY:		-	-	-	-	○	○	-	-	-	-	○	-	-	○
	if(uc_m_switch_buffer == M_AIR_KEY)		-	-	-	-	○	○	-	-	-	-	○	-	-	○
	case M_MODE_AIR_KEY:		-	-	-	-	○	○	○	-	-	-	○	-	-	○
	if(uc_m_switch_buffer == M_MODE_AIR_KEY)		-	-	-	-	○	○	○	-	-	-	○	-	-	○
	if(uc_m_switch_flag == 1)		-	-	-	-	○	-	-	-	-	-	○	-	-	○
	else		-	-	-	-	-	○	○	-	-	-	-	-	-	-
	for(luc_i=(KEY_TIME, ALL); luc_i>0; luc_i--)		-	-	-	-	-	○	○	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= us_key_time_event[luc_i-1])		-	-	-	-	-	-	○	-	-	-	-	-	-	-
	if(luc_i == 0)		-	-	-	-	-	○	○	-	-	-	-	-	-	-
	case M_REMOVE_KEY:		-	-	-	-	-	-	-	○	-	-	-	-	-	-
	default:		-	-	-	○	-	-	-	-	○	○	-	-	-	-
	if (uc_m_switch_buffer == M_ALL_OFF)		-	-	-	○	-	-	-	-	○	-	-	-	-	-
	if(anychange)		-	-	-	-	○	○	○	○	-	-	○	-	-	○
	if(lus_key_event == lsss_last_off_key)		-	-	-	-	○	-	-	-	-	-	○	-	-	○
	if(luc_i < 7)		-	-	-	-	○	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	○	-	-	-	-	-	-	-
	if(ss_event_zipc == M_NA)		-	-	-	○	○	-	-	○	○	○	○	-	-	○
	if(PORT3.PIDR.BIT.B1 == 0)		-	-	-	○	○	-	-	○	○	○	-	-	-	-
	Symbol	Range														
Input	luc_num	0~2	1	1	0	0	0	0	0	0	0	0	0	0	0	2
	uc_m_switch_buffer	0x00~0xFF	0x10	0x00	0x00	0x00	0x01	0x02	0x03	0x10	0x00	0xFF	0x01	0xFF	0x01	0xFF
	uc_key_release_on_flg	0~3	1	0	0	0	0	0	0	0	0	0	0	0	0	3
	uc_no_op_timer_count_flg	0~2	0	0	1	0	0	0	0	0	0	0	0	0	0	2
	us_no_op_timer_count	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	uc_timer_count_flg	0~2	0	0	0	1	0	0	0	0	0	0	0	0	0	2
	us_timer_count	0x0000,0xFFFF	0x0000	0x0000	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	uc_m_switch_flag	0~12	0	0	0	0	1	0	0	0	0	0	1	0	1	12
	us_m_switch_on_time	0~1001	0	0	0	0	0	500	0	0	0	0	0	0	0	1001
	us_key_time_event	0~1002	0	0	0	0	0	501	0	0	0	0	0	1002	0	0
	PORT3.PIDR.BIT.B1	0~2	0	0	0	0	0	0	0	0	0	0	1	0	1	2
Output	ss_event_zipc	-1~104	49	49	99	98	104	25	104	104	104	104	-1	104	-1	-1
	uc_key_release_on_flg	0~3	2	0	0	0	1	1	1	0	0	0	1	0	1	3
	uc_no_op_timer_count_flg	0~2	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	uc_timer_count_flg	0~2	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	anychange	0,1	1	0	1	1	1	1	1	0	0	0	1	0	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_key_set

	Responsible	Last update
Coding	Ikarashi	2021/6/29
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	12

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12
Root	if(luc_sw_new == luc_sw_old)		○	○	○	○	○	-	-	-	-	-	-	○
	if(luc_sw_new != M_ALL_OFF)		○	○	○	○	-	-	-	-	-	-	-	○
	if(us_m_switch_on_time < 1000)		○	○	-	○	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	○
	if(us_m_switch_on_time >= 20)		-	○	○	-	-	-	-	-	-	-	-	○
	else		○	-	-	○	-	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= 100)		-	-	○	-	-	-	-	-	-	-	-	○
	else		○	○	-	○	-	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= 200)		-	-	○	-	-	-	-	-	-	-	-	○
	else		○	○	-	○	-	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= 300)		-	-	○	-	-	-	-	-	-	-	-	○
	else		○	○	-	○	-	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= 400)		-	-	○	-	-	-	-	-	-	-	-	○
	else		○	○	-	○	-	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= 500)		-	-	○	-	-	-	-	-	-	-	-	○
	else		○	○	-	○	-	-	-	-	-	-	-	-
	if(us_m_switch_on_time >= 1000)		-	-	○	-	-	-	-	-	-	-	-	○
	else		○	○	-	○	-	-	-	-	-	-	-	-
	switch(uc_m_switch_flag)		○	○	○	○	-	-	-	-	-	-	-	○
	case 1		○	-	-	-	-	-	-	-	-	-	-	-
	case 2		-	○	○	-	-	-	-	-	-	-	-	-
	case 3		-	○	○	-	-	-	-	-	-	-	-	-
	case 4		-	○	○	-	-	-	-	-	-	-	-	-
	case 5		-	○	○	-	-	-	-	-	-	-	-	-
	case 6		-	○	○	-	-	-	-	-	-	-	-	-
	case 7		-	○	○	-	-	-	-	-	-	-	-	-
	case 11		-	○	○	-	-	-	-	-	-	-	-	○
	default		-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	○	○	○	○	○	○
	if(luc_sw_new != M_ALL_OFF)		-	-	-	-	-	○	○	○	-	-	-	○
	if(luc_sw_old == M_ALL_OFF)		-	-	-	-	-	-	-	-	-	-	-	○
	else		-	-	-	-	-	○	○	○	-	-	-	-
	if((luc_sw_old == M_MODE_KEY) && (luc_sw_new == (M_MODE_KEY | M_AIR_KEY)))		-	-	-	-	-	○	-	-	-	-	-	-
	else if((luc_sw_old == M_AIR_KEY) && (luc_sw_new == (M_MODE_KEY | M_AIR_KEY)))		-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-	-
	if(luc_ps_flg == ON)		-	-	-	-	-	○	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	○	○	-
	if(uc_m_switch_buffer == M_KEY_END)		-	-	-	-	-	-	-	-	○	○	-	-
	if(KeyEvent_Make_Event(1) == ON)		-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	-
	if(uc_m_switch_active == M_ON)		-	○	○	-	-	○	○	-	○	○	○	○
	if(uc_m_switch_buffer == M_KEY_END)		-	-	-	-	-	○	-	-	-	-	-	-
	else		-	○	○	-	-	-	○	-	○	○	○	○
	else		○	-	-	○	○	-	-	○	-	-	-	-
	Symbol	Range												
Input	uc_m_key_data	0x00~0xFF	0xFF	0xFF	0xFF	0xFF	0x00	0x03	0x03	0x10	0x00	0x00	0x00	0xFF
	us_m_switch_on_time	0~1000	0	20	1000	0	0	0	0	0	0	0	0	1001
	uc_m_switch_flag	0~11	1	2	11	0	0	0	0	0	0	0	0	2
	luc_sw_new	0x00~0xFF	0xFF	0xFF	0xFF	0xFF	0x00	0x01	0x02	0xFF	0xFF	0xFF	0xFF	0x00
	KeyEvent_Make_Event()	0,1	0	0	0	0	0	0	0	0	1	0	0	2
	uc_m_switch_buffer	0x00~0xFF	0x00	0x00	0x00	0x00	0x80	0x80	0x00	0x00	0x80	0x80	0x00	0xFF
Output	uc_m_switch_flag	0~11	1	2	11	0	0	1	1	0	0	0	0	1
	uc_m_switch_active	0,1	0	1	1	0	0	1	1	0	1	1	1	1
	uc_m_switch_buffer	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x03	0x00	0x20	0x10	0x10	0x00	0xFF
	us_m_switch_on_time	0~1000	1	21	1000	1	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	lcd_main.c

LCDMAIN_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3
Root	if(uss_m_back_light_timer)		-	○	○
	else		○	-	-
	if(uss_m_IrDA_disconnect_count)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	uss_m_back_light_timer	0~25501	0	25500	25501
	uss_m_IrDA_disconnect_count	0~18001	0	18000	18001
Output	None	-	-	-	-
Judgment			OK	OK	OK

LCDMAIN_Clr_DispData

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_Set_BackLightTimer

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	stp_data->us_back_light_time	0~256	0	255	256
Output	uss_m_back_light_timer	0~25600	0	25500	25600
Judgment			OK	OK	OK

LCDMAIN_Set_BattIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_num	0~5	0	4	5
Output	None	-	-	-	-
Judgment			OK	OK	OK

LCDMAIN_Set_BumpChkIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(stp_data->uc_bump_lmt_set == M_ON)		○	○	-	-
	if(FuncBumpLimit_Get_ChkBumpLimitExpired() == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	stp_data->uc_bump_lmt_set	0~2	1	1	0	2
	FuncBumpLimit_Get_ChkBumpLimitExpired()	0~2	0	1	0	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

LCDMAIN_Set_Com_Discon_count

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onof	0~2	1	0	2
Output	uss_m_IrDA_disconnect_count	0,18000	18000	0	0
Judgment			OK	OK	OK

LCDMAIN_set_data

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_Set_HartIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_Set_PumpIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(PumpFlow_Get_MotionFlag_isMove() == true)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	PumpFlow_Get_MotionFlag_isMove()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

LCDMAIN_Set_LongIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_Set_ModeIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_Set_OffBackLight

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_onoff	0~2	0	1	2
Output	ucs_m_back_light_off	0~2	0	1	2
Judgment			OK	OK	OK

LCDMAIN_Set_OnSdmBackLight

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_back_liget_sdm	1	1
Judgment			OK

LCDMAIN_Set_TimeIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_start

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_stop

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDMAIN_Set_BackLight

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	11

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11
Root	if(FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	○	-	○	○	-	-	-	○	○	○
	If(uss_m_back_light_timer == 0)		○	○	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	○	-	-	-	○	-	-
	if(ZIPCNumSet_Get_AlarmFlg() == M_ON)		○	○	-	○	-	-	-	-	-	-	-
	else if(FlowMain_Get_ModeStatus() == FlowMain_IRDA_MODE_NUMBER)		-	-	-	-	-	○	-	○	-	-	-
	if(ComCommand_Get_UsbComPartner() == true)		-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-
	else		-	-	○	-	-	-	○	-	-	-	-
	if(stp_data->uc_stealth_on == M_ON)		○	○	○	○	○	○	○	○	-	-	-
	if(ucs_m_back_liget_sdm == M_ON)		○	○	○	○	○	○	○	○	○	-	-
	if(ucs_m_back_light_off == M_ON)		○	○	○	○	○	○	○	○	○	○	-
	Symbol	Range											
Input	ComCommnd_Get_UsbComPartner()	1,2	1	1	1	1	1	1	1	2	1	1	1
	uc_stealth_on	1,2	1	1	1	1	1	1	1	1	2	2	2
	FlowMain\get_ModeStatus()	1~8	1	2	3	1	1	7	8	7	1	1	1
	ZIPCNumSet_Get_AlarmFlg()	1,2	1	1	1	1	2	1	1	1	2	2	2
	ucs_m_back_light_sdm	1,2	1	1	1	1	1	1	1	1	1	2	2
	ucs_m_back_light_off	1,2	1	1	1	1	1	1	1	1	1	1	2
	uss_m_back_light_timer	0,1	0	0	0	1	1	0	0	0	1	1	1
Output	None	-	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	lcd_normal_drv.c

LCDNORMAL_all_display_ram

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	for(luc_i=0; luc_i<4; luc_i++)		○	○
	if(stp_data->uc_ble_onoff == M_OFF)		○	-
	for(luc_i=0; luc_i<2; luc_i++)		○	○
	for(luc_j=0; luc_j<8; luc_j++)		○	○
	for(luc_i=0; luc_i<8; luc_i++)		○	○
	Symbol	Range		
Input	uc_ble_onoff	0,1	0	1
	uc_lcd_icon[0]	0	0	0
Output	uc_lcd_icon_blink	0	0	0
	uc_lcd_time	126	126	126
	uc_lcd_time_blink	0	0	0
	uc_lcd_time_dot	255	255	255
	uc_lcd_time_dot_blink	0	0	0
	uc_lcd_mes[0]	126	126	126
	uc_lcd_mes_blink	0	0	0
	uc_lcd_mes_dot	255	255	255
	uc_lcd_mes_dot_blink	0	0	0
	uc_lcd_com	126	126	126
	uc_lcd_com_blink	0	0	0
	uc_lcd_com_dot	255	255	255
	uc_lcd_com_dot_blink	0	0	0
Judgment			OK	OK

LCDNORMAL_clr_display_ram

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	for(luc_i=0; luc_i<4; luc_i++)		○
	for(luc_i=0; luc_i<2; luc_i++)		○
	for(luc_j=0; luc_j<8; luc_j++)		○
	for(luc_i=0; luc_i<8; luc_i++)		○
	Symbol	Range	
Input	None	-	-
Output	st_normal_disp.uc_lcd_icon[luc_i]	0x00	0x00
	st_normal_disp.uc_lcd_icon_blink[luc_i]	0x00	0x00
	st_normal_disp.uc_lcd_time[luc_i]	0x20	0x20
	st_normal_disp.uc_lcd_mes[luc_i][luc_j]	0x20	0x20
	st_normal_disp.uc_lcd_mes_blink[luc_i]	0x00	0x00
	st_normal_disp.uc_lcd_mes_dot[luc_i]	0x00	0x00
	st_normal_disp.uc_lcd_mes_dot_blink[luc_i]	0x00	0x00
	st_normal_disp.uc_lcd_com[luc_i]	0x20	0x20
	st_normal_disp.uc_lcd_time_blink	0x00	0x00
	st_normal_disp.uc_lcd_time_dot	0x00	0x00
	st_normal_disp.uc_lcd_time_dot_blink	0x00	0x00
	st_normal_disp.uc_lcd_com_dot	0x00	0x00
	st_normal_disp.uc_lcd_com_blink	0x00	0x00
	st_normal_disp.uc_lcd_com_dot_blink	0x00	0x00
Judgment			OK

LCDNORMAL_int_10msec

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ucs_m_timer)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	ucs_m_timer	0~26	0	25	26
Output	None	-	-	-	-
Judgment			OK	OK	OK

LCDNORMAL_Set_BackLight

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_onoff == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_onoff	0~2	1	0	2
Output	PORTA.PODR.BIT.B6	0,1	1	0	0
Judgment			OK	OK	OK

LCDNORMAL_set_data

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(KeyEvent_Chk_DispPowerOff() == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	KeyEvent_Chk_DispPowerOff()	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

LCDNORMAL_Set_DispDataUpDATA

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	Yes
Verification pass	1
Judgment	OK

			case1
Root	while(ucs_m_timer)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDNORMAL_Set_Display_point

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	0

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	&st_normal_disp	-	Indefinite value
Judgment			OK

LCDNORMAL_start

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LCDNORMAL_stop

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_write_normal_disp_data

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	13

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	if(++luc_count == 4)		○	-	-	-	-	-	-	-	-	-	-	-	-
	for(luc_i=0; luc_i<M_WR_TX_LCD_BYTE; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○
	for(luc_i = 0; luc_i < 7; luc_i++)		○	○	○	○	○	○	○	-	○	○	○	○	○
	if(luc_count < 2)		○	-	-	-	○	○	-	-	○	○	○	○	○
	for(luc_i=0; luc_i<4; luc_i++)		○	-	-	-	○	○	-	○	-	-	○	○	○
	else		-	○	○	○	-	-	○	-	-	-	-	-	-
	for(luc_i=0; luc_i<4; luc_i++)		-	○	○	○	-	-	○	-	○	○	-	-	-
	for(luc_i = 0; luc_i < 4; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○
	if(luc_count < 2)		○	-	○	-	○	○	-	○	○	○	○	○	○
	if(st_normal_disp.uc_lcd_time_blink & luc_mask_bit)		○	-	○	-	-	-	-	○	-	-	○	○	○
	if(st_normal_disp.uc_lcd_time_dot_blink & luc_mask_bit)		○	-	○	-	○	-	-	○	-	-	○	○	○
	switch(luc_i)		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 1:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 2:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 3:		○	○	○	○	○	○	○	○	○	○	○	○	○
	for(luc_i = 0; luc_i < 4; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○
	for(luc_k = 0; luc_k < 2; luc_k++)		○	-	○	○	○	○	○	○	○	○	○	○	○
	for(luc_i = 0; luc_i < 8; luc_i++)		○	-	○	○	○	○	○	○	○	○	○	○	○
	if(luc_count < 2)		○	-	○	-	○	○	-	○	○	○	○	○	○
	if(st_normal_disp.uc_lcd_mes_blink[luc_k] & luc_mask_bit)		○	-	○	-	○	○	-	-	-	-	○	-	-
	if(st_normal_disp.uc_lcd_mes_dot_blink[luc_k] & luc_mask_bit)		○	-	○	-	○	○	-	○	○	-	○	-	-
	if(luc_k == 0)		○	○	○	○	○	○	○	○	○	○	○	○	○
	switch(luc_i)		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 1:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 2:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 3:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 4:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 5:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 6:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 7:		○	○	○	○	○	○	○	○	○	○	○	○	○
	else		○	○	○	○	○	○	○	○	○	○	○	○	○
	switch(luc_i)		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 1:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 2:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 3:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 4:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 5:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 6:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 7:		○	○	○	○	○	○	○	○	○	○	○	○	○
	for(luc_i = 0; luc_i < 8; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○
	for(luc_i = 0; luc_i < 8; luc_i++)		○	○	○	○	○	○	○	○	○	○	○	○	○
	if(luc_count < 2)		○	-	○	-	○	○	-	○	○	○	○	○	-
	if(st_normal_disp.uc_lcd_com_blink & luc_mask_bit)		○	-	○	-	○	○	-	○	○	○	-	-	-
	if(st_normal_disp.uc_lcd_com_dot_blink & luc_mask_bit)		○	-	○	-	○	○	-	○	○	○	○	○	○
	switch(luc_i)		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 1:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 2:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 3:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 4:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 5:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 6:		○	○	○	○	○	○	○	○	○	○	○	○	○
	case 7:		○	○	○	○	○	○	○	○	○	○	○	○	○
	Symbol	Range													
Input	uc_code14[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_code14[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	LCDNormalFont_seg_conv7()	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SCI6IIC_Int_Sci6_Send_LCD()	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[35]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[34]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[29]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[72]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[71]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[55]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[51]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[59]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[57]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[69]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[33]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[32]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[31]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[30]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_icon[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_icon_blink[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_time[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_time_blink	0,255	255	255	255	255	0	0	0	255	0	0	255	0	0
	uc_lcd_time_dot	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_time_dot_blink	0,255	255	255	255	255	255	0	0	255	0	0	255	0	0
	uc_lcd_mes[0][0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes_blink[0]	0,255	255	255	255	255	255	255	255	0	0	0	255	255	255
	uc_lcd_mes_dot[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_mes_dot_blink[0]	0,255	255	255	255	255	255	255	255	255	255	0	255	255	255
	uc_lcd_com[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_com_blink	0,255	255	255	255	255	255	255	255	255	255	255	0	0	0
	uc_lcd_com_dot	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_lcd_com_dot_blink	0,255	255	255	255	255	255	255	255	255	255	255	255	255	0
	luc_count	0~4	3	4	0	2	0	0	2	0	0	0	0	0	0
Output	uc_disp_data[0]	128	128	128	128	128	128	128	128	128	128	128	128	128	128
	uc_disp_data[1]	200	200	200	200	200	200	200	200	200	200	200	200	200	200
	uc_disp_data[3]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	uc_disp_data[5]	64	64	64	64	64	64	64	64	64	64	64	64	64	64
	uc_disp_data[7]	224	224	224	224	224	224	224	224	224	224	224	224	224	224
	uc_disp_data[9]	248	248	248	248	248	248	248	248	248	248	248	248	248	248
	uc_disp_data[11]	240	240	240	240	240	240	240	240	240	240	240	240	240	240
	uc_disp_data[13]	232	232	232	232	232	232	232	232	232	232	232	232	232	232
	uc_disp_data[14]	64	64	64	64	64	64	64	64	64	64	64	64	64	64
	luc_count	0~5	0	5	1	3	1	1	3	1	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	lcd_normal_font.c

LCDNormalFont_seg_conv7

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_i = 0 ; lus_i < (sizeof(uc_7seg_table)/sizeof(uc_7seg_table[0])) ; lus_i++)		○	○	○
	if(puc_code == uc_7seg_table[lus_i].uc_asc)		○	○	-
	else		○	○	○
	Symbol	Range			
Input	puc_code	0~68	0	67	68
Output	lucp_ans_pnt	-	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK

LCDNormalFont_seg_conv14

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lus_i = 0 ; lus_i < (sizeof(uc_14seg_table)/sizeof(uc_14seg_table[0])) ; lus_i++)		○	○	○
	if(puc_code == uc_14seg_table[lus_i].uc_asc)		○	○	-
	else		○	○	○
	Symbol	Range			
Input	puc_code	0~43	0	42	43
Output	lucp_ans_pnt	-	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	lcd_normal_show.c

LCDNormalShow_Set_7segGasNameDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch(puc_pos)		○	○	○	○	○	○	○
	case 0		○	○	○	-	-	-	-
	case 1		-	-	-	○	-	-	-
	case 2		-	-	-	-	○	-	-
	case 3		-	-	-	-	-	○	-
	default		-	-	-	-	-	-	○
	switch(puc_pos)		○	○	○	○	○	○	○
	case 0		○	○	○	-	-	-	-
	if((stp_GasInfo_addr+puc_pos)->uc_gas_comb != (stp_GasInfo_addr+puc_pos)->uc_sub_comb)		○	-	-	-	-	-	-
	else		-	○	○	-	-	-	-
	if(Tool_Comp_Pnt(&(stp_GasInfo_addr+puc_pos)->uc_name[0][0], (U_CHR *)&luc_CH4[0], 10) == 0 || Tool_Comp_Pnt(&(stp_GasInfo_addr+puc_pos)->uc_name[0][0], (U_CHR *)&luc_H2[0], 10) == 0)		-	○	-	-	-	-	-
	else		-	-	○	-	-	-	-
	case 1		-	-	-	○	-	-	-
	case 2		-	-	-	○	○	-	-
	case 3		-	-	-	○	○	○	-
	default		-	-	-	-	-	-	○
	Symbol	Range							
Input	puc_pos	0~4	0	0	0	1	2	3	4
	(stp_GasInfo_addr+puc_pos)->uc_gas_comb	0~30	0	0	0	0	0	29	30
	(stp_GasInfo_addr+puc_pos)->uc_sub_comb	0~30	29	0	0	0	0	0	30
	Tool_Comp_Pnt()	0~2	0	0	1	0	0	0	2
Output	None	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK

LCDNormalShow_Set_AlarmCommentDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if((puc_alarm_event & ALARMEVENT_ALM_STL) == ALARMEVENT_ALM_STL)		○	-	-	-	-	-	-	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM_TWA) == ALARMEVENT_ALM_TWA)		-	○	-	-	-	-	-	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM__1H) == ALARMEVENT_ALM__1H)		-	-	○	-	-	-	-	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM_OVR) == ALARMEVENT_ALM_OVR)		-	-	-	○	-	-	-	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM_MIN) == ALARMEVENT_ALM_MIN)		-	-	-	-	○	-	-	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM_3RD) == ALARMEVENT_ALM_3RD)		-	-	-	-	-	○	-	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM_2ND) == ALARMEVENT_ALM_2ND)		-	-	-	-	-	-	○	-	-	-
	else if((puc_alarm_event & ALARMEVENT_ALM_1ST) == ALARMEVENT_ALM_1ST)		-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	○	○
	Symbol	Range										
Input	puc_alarm_event	0x00~0xFF	0x08	0x10	0x40	0x80	0x20	0x04	0x02	0x01	0x00	0xFF
Output	None	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

LCDNormalShow_Set_BattIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(puc_num)		○	○	○	○	○	○
	case 0		○	-	-	-	-	-
	case 1		-	○	-	-	-	-
	case 2		-	-	○	-	-	-
	case 3		-	-	-	○	-	-
	case 4		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	puc_num	0~5	0	1	2	3	4	5
Output	stp_nl_disp->uc_lcd_icon[3]	0x80~0xF0	0x80	0x90	0x90	0xB0	0xF0	0xF
	stp_nl_disp->uc_lcd_icon_blink[3]	0x80,0x90	0x80	0x90	0x90	0x90	0x90	0x90
Judgment			OK	OK	OK	OK	OK	OK

LCDNormalShow_Set_BumpChkIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_icon[0]	0x42	0x42
Judgment			OK

LCDNormalShow_Set_CharacterDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(puc_pos)		○	○	○	○	○
	case 0		○	-	-	-	-
	case 1		-	○	-	-	-
	case 2		-	-	○	-	-
	case 3		-	-	-	○	-
	default		-	-	-	-	○
	if(puc_blink == M_ON)		○	-	-	-	-
	else		-	○	○	○	○
	Symbol	Range					
Input	puc_pos	0~4	0	1	2	3	4
	*pucp_data	0X00~0xFF	0x00	0x00	0x00	0x00	0xFF
	puc_blink	0~2	1	0	0	0	2
Output	stp_nl_disp->uc_lcd_mes_blink[luc_y]	0xF0	0xF0	0	0	0	0
	stp_nl_disp->uc_lcd_mes_dot_blink[luc_y]	0xE0	0xE0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

LCDNormalShow_Set_DenstDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch(puc_pos)		○	○	○	○	○	○	○
	case 0		○	-	-	-	-	-	-
	case 1		-	○	-	-	-	-	-
	case 2		-	-	○	-	-	-	-
	case 3		-	-	-	○	-	-	-
	default		-	-	-	-	○	○	○
	if(pul_val == M_OFF_VAL)		○	-	-	-	-	-	-
	else if(pul_val == M_NA_VAL)		-	○	-	-	-	-	-
	else if(puc_fail == M_ON)		-	-	○	-	-	-	-
	else if(puc_over == M_ON)		-	-	-	○	-	-	-
	else if(puc_m_over == M_ON)		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	○
	if(puc_dec_pos != 0)		-	-	-	-	-	○	-
	else		-	-	-	-	-	-	○
	if(puc_muinus == M_ON)		-	-	-	-	-	○	-
	else		-	-	-	-	-	-	○
	if(puc_blink == M_ON)		○	-	-	-	-	-	-
	else		-	○	○	○	○	○	○
	Symbol	Range							
Input	puc_pos	0~4	0	1	2	3	4	4	4
	pul_val	0x00000000~0xFFFFFFFF	0xDDDDDDDD	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
	puc_over	0~2	0	0	0	1	0	0	2
	puc_m_over	0~2	0	0	0	0	1	0	2
	puc_muinus	0~2	0	0	0	0	0	1	2
	puc_fail	0~2	0	0	1	0	0	0	2
	puc_dec_pos	0~4	0	0	0	0	4	3	0
	puc_blink	0~2	1	0	0	0	0	0	2
Output	stp_nl_disp->uc_lcd_mes_dot[luc_y]	0x00~0x80	0x00	0x00	0x00	0x00	0x00	0x80	0x00
	stp_nl_disp->uc_lcd_mes[luc_y][luc_x]	0x00~0x2D	0x00	0x00	0x00	0x00	0x00	0x2D	0x00
	stp_nl_disp->uc_lcd_mes_blink[luc_y]	0xF0	0xF0	0x00	0x00	0x00	0x00	0x00	0x00
	stp_nl_disp->uc_lcd_mes_dot_blink[luc_y]	0xE0	0xE0	0x00	0x00	0x00	0x00	0x00	0xE0
Judgment			OK	OK	OK	OK	OK	OK	OK

LCDNormalShow_Set_HartIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_icon[0]	0xC2	0xC2
	stp_nl_disp->uc_lcd_icon_blink[0]	0x82	0x82
Judgment			OK

LCDNormalShow_Set_LongIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	○	-	-
	if(stp_data->uc_long_energy_flg == M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	FlowMain_Get_ModeStatus()	0~7	0	0	7	8
	stp_data->uc_long_energy_flg	0,1	1	0	0	2
Output	stp_nl_disp->uc_lcd_icon[0]	0xE2	0xE2	0xE2	0xE2	0xE2
Judgment			OK	OK	OK	OK

LCDNormalShow_Set_MeasurementDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(puc_mode == 0 || puc_mode == 2)		○	○	○	○	○	○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	○	○	-	-	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (luc_on_gas & 0x01))		○	○	○	○	○	-	-	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	○	○	-	-	-	-	-
	if(!((stp_denst+luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)))		○	○	-	-	-	-	-	-	-
	if((stp_denst+luc_i)->uc_alarm_lat != 0x00)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	if(puc_mode == 0)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		-	-	○	○	-	-	-	-	-
	if(puc_mode == 0)		-	-	○	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-
	else		○	○	-	-	○	-	-	-	-
	else		-	-	-	-	-	○	-	-	-
	if((stp_GasInfo_addr+DataStructConst_CO_CAN_H2_H2_CH)->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM)		○	○	○	○	-	-	-	-	-
	if((stp_denst+DataStructConst_CO_CAN_H2_H2_CH)->uc_over_flg == M_ON)		○	○	○	-	-	-	-	-	-
	if(ZIPCNumSet_Get_DispFlicker2() == 0)		○	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	○	○	○
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	-	-	-	-	○	○	○
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (luc_on_gas & 0x01))		-	-	-	-	-	-	○	-	-
	if(luc_i != DataStructConst_CO_CAN_H2_H2_CH || (stp_GasInfo_addr+luc_i)->uc_gas_comb != DataStructConst_CO_CAN_H2_GAS_NUM)		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	○
	Symbol	Range									
Input	puc_mode	0~3	0	2	0	2	0	0	1	1	3
	luc_on_gas	0x00~0xFF	0x01	0x01	0x01	0x01	0xFF	0x00	0xFF	0x00	0xFF
	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	29	29	29	29	3	0	3	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0x00	0xFF	0xFF	0x00	0x00	0x00	0x00	0xFF
	(stp_denst+luc_i)->uc_alarm_lat	0x00,0xFF	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	(stp_GasInfo_addr+DataStructConst_ CO_CAN_H2_H2_CH)->uc_gas_comb	0~30	3	3	3	0	29	29	3	0	30
	(stp_denst+DataStructConst_ CO_CAN_H2_H2_CH)->uc_over_flg	0~2	1	1	0	0	0	0	0	0	2
	ZIPCNumSet_Get_DispFlicker2()	0~2	0	1	0	0	0	0	0	0	2
Output	None	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

LCDNormalShow_Set_ModeIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	FlowMain_Get_ModeStatus()	0~8	0	7	8
Output	stp_nl_disp->uc_lcd_icon[0]	0xE2,0xF2	0xE2	0xF2	0xF2
Judgment			OK	OK	OK

LCDNormalShow_Set_NoALARMIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER || FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER)		○	○	-	-
	if(stp_data->uc_stealth_on == M_ON)		○	-	-	-
	else		-	○	-	-
	if(stp_data->uc_alarm_on == M_OFF)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	FlowMain_Get_ModeStatus()	0~8	0	0	7	8
	stp_data->uc_stealth_on	0~2	1	0	0	2
	stp_data->uc_alarm_on	0~2	1	0	0	2
Output	stp_nl_disp->uc_lcd_icon[0]	0xFA,0xFE	0xFA	0xFE	0xFE	0xFE
Judgment			OK	OK	OK	OK

LCDNormalShow_Set_TimeIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_nl_disp->uc_lcd_time_dot	0x05	0x05
	stp_nl_disp->uc_lcd_time_dot_blink	0x05	0x05
Judgment			OK

LCDNormalShow_Set_PumpIconDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_icon_onoff == M_ON)		○	-	-	○
	else		-	○	○	-
	if(luc_toggle_cnt >= 4U)		○	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	None	-	-	-	-	-
Output	puc_icon_onoff	1,2	1	2	2	1
	luc_togglee_cnt	0,3	3	3	3	0
Judgment			OK	OK	OK	OK

LCDNormalShow_Set_GasNameDisp

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Execution count	34
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20	case21	case22	case23	case24	case25	case26	case27	case28	case29	case30	case31	case32	case33	case34
Root	switch(puc_pos)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case 0		○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_on == M_ON)		○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(Tool_Comp_Pnt(puc_name, (U_CHR *)"CHK", 3) == 0)		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)"HC ",3) == 0)		-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)"H2 ", 3) == 0)		-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)" ", 3) == 0)		-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)"****", 3) == 0)		-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	switch(puc_unit)		○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 0		○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_unit_blink == M_ON)		○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1		-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_unit_blink == M_ON)		-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2		-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_unit_blink == M_ON)		-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 4		-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_unit_blink == M_ON)		-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default		-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 1		-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_on == M_ON)		-	-	-	-	-	-	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(Tool_Comp_Pnt(puc_name, (U_CHR *)"O2 ", 3) == 0)		-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	switch(puc_unit)		-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 4		-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_unit_blink == M_ON)		-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	case 2		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-
	if(puc_name_on == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-
	if(Tool_Comp_Pnt(puc_name, (U_CHR *)"H2S", 3) == 0)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)"SO2", 3) == 0)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	if(puc_name_blink == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)"&("), 3) == 0)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else if(Tool_Comp_Pnt(puc_name, (U_CHR *)"&(****"), 3) == 0)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
	switch(puc_unit)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	case 2		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	if(puc_unit_blink == M_ON)		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
	default		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	Symbol	Range																																		
Input	puc_pos	0~4	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4	
	puc_name_blink	0~2	1	0	1	0	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	2	
	puc_name_on	0~2	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	2	
	puc_unti	0~5	0	0	1	1	2	2	4	4	4	5	5	4	4	5	5	2	2	5	5	5	5	5	5	5	2	2	5	5	5	5	5	5	5	5	
	puc_unti_blink	0~2	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	
	puc_name[0]	0x00~0xFF	0x43	0x43	0x48	0x48	0x48	0x48	0x00	0x00	0xFF	0xFF	0xFF	0x79	0x79	0x00	0x00	0x48	0x48	0x83	0x83	0x84	0x84	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	puc_name[1]	0x00~0xFF	0x48	0x48	0x43	0x43	0x18	0x18	0x00	0x00	0xFF	0xFF	0xFF	0x18	0x18	0x00	0x00	0x18	0x18	0x79	0x79	0x79	0x79	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	puc_name[2]	0x00~0xFF	0x20	0x20	0x00	0x00	0x00	0x00	0x00	0x00	0xFF	0xFF	0xFF	0x00	0x00	0x00	0x00	0x83	0x83	0x18	0x18	0x88	0x88	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	Tool_Comp_Pnt(0)	0~2	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	2	2	
	Tool_Comp_Pnt(1)	0~2	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	2	2	
	Tool_Comp_Pnt(2)	0~2	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	2	2	
	Tool_Comp_Pnt(3)	0~2	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	2	2	
	Tool_Comp_Pnt(4)	0~2	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	2	2	
Output	stp_rt_disp->uc_lcd_icon[1]	0x00~0x64	0x3C	0x3C	0x66	0x66	0x81	0x81	0x04	0x04	0x64	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	stp_rt_disp->uc_lcd_icon_blink[1]	0x00~0x64	0x3C	0x00	0x66	0x00	0x81	0x00	0x04	0x04	0x64	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	stp_rt_disp->uc_lcd_icon[2]	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x05	0x05	0x02	0x02	0x00	0x00	0x00	0x08	0x08	0x00	0x50	0x5F	0x7F	0x7F	0xFF	0xFF	0xFF	0xFF	
	stp_rt_disp->uc_lcd_icon_blink[2]	0x00~0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x05	0x05	0x07	0x07	0x0F	0x0F	0x0F	0x0F	0x0F	0x0F	0x5F	0x50	0x20	0x20	0x00	0x00	0x80	0x00	
	stp_rt_disp->uc_lcd_icon[3]	0x00,0x05	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x05	0x05	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	stp_rt_disp->uc_lcd_icon_blink[3]	0x00,0x05	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x05	0x05	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	

Unit Test Specification(with Results)

Project name	GX-Force
File name	led.c

LED_Control_Led_Port

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(puc_on == M_ON)		○	-	-	-
	else		-	○	○	○
	switch(puc_port)		○	○	○	○
	case LED_LED1		○	-	-	-
	case LED_LED2		-	○	-	-
	case LED_LED3		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	puc_port	1~4	1	2	3	4
	puc_on	0~2	1	0	0	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

LED_10msec_Interrupt

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uss_led_light_on_time != 0U)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	uss_led_light_on_time	0~2	1	0	2
Output	uss_led_light_on_time	0,1	0	0	1
Judgment			OK	OK	OK

LED_Get_LightOnOffStatus

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uss_led_light_on_time != 0U)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	uss_led_light_on_time	0~2	1	0	2
Output	luc_onoff	0,1	1	0	1
Judgment			OK	OK	OK

LED_Set_InitRAM

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_led_light_on_time	0	0
Judgment			OK

LED_Set_LightOff

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uss_led_light_on_time	0	0
Judgment			OK

LED_Set_LightOnTime

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(uss_led_light_on_time != 0U)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	uss_led_light_on_time	0~2	1	0	2
Output	uss_led_light_on_time	0,12000	0	12000	0
Judgment			OK	OK	OK

LED_Test_Command

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_led_onoff == M_OFF)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_led_onoff	0~2	1	0	2
Output	uss_led_light_on_time	0,12000	12000	0	12000
Judgment			OK	OK	OK

m_Set_LedLightPort

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_led_onoff == M_OFF)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_led_onoff	0~2	1	0	2
Output	PORT5.PODR.BIT.B4	0,1	1	0	1
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	logger_GXForce.c

LoggerGXForce_Chk_1secLoggerRun

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_i = 0; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○	-	-	-
	if((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))		○	○	○	-	-	-
	if((st_datalogger_control.uc_state & DATALOGGER_RUN) || (luc_sensor_status != 0) || (luc_main_status != 0))		○	-	-	-	-	-
	else if(st_datalogger_control.uc_state & DATALOGGER_OVERWRITE)		-	○	-	-	-	-
	else		-	-	○	-	-	-
	else		-	-	-	○	○	○
	if(lus_timer >= M_RUN_ERR_TIMER)		-	-	○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	○	-	-	-
	for(luc_j=0; luc_j<15; luc_j++)		-	-	○	-	-	-
	else		○	○	-	○	○	○
	Symbol	Range						
Input	FlowMain_Get_ModeStatus()	0~8	1	1	1	0	7	8
	lus_timer	0~181	0	0	180	0	0	181
	(stp_denst + luc_i)->uc_state_trb	0x00,0xFF	0xFF	0x00	0x00	0x00	0x00	0xFF
	stp_status->us_main_state	0x0000,0xFFFF	0xFFFF	0x00	0x00	0x00	0x00	0xFFFF
	st_datalogger_control.uc_state	0x00,0xFF	0x01	0x08	0x00	0x00	0x00	0xFF
Output	(stp_denst+luc_i)->ul_60_total	0	0	0	0	0	0	0
	(stp_denst+luc_i)->ul_ave	0	0	0	0	0	0	0
	(stp_denst+luc_i)->ul_ave_total_cnt	0	0	0	0	0	0	0
	(stp_denst+luc_i)->ul_twa	0	0	0	0	0	0	0
	(stp_denst+luc_i)->ul_stel	0	0	0	0	0	0	0
	(stp_denst+luc_i)->ul_stel_point	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

LoggerGXForce_Chk_LoggerDataClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

-			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_logger_data_clear	0	0
Judgment			OK

LoggerGXForce_Chk_LoggerPowerClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_logger_power_clear	0	0
Judgment			OK

LoggerGXForce_Clear_LoggerDataClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ucs_m_logger_data_clear)		○	○	○	○
	case M_OFF		○	-	-	-
	case M_ON		-	○	○	-
	if(DataLogger_clear() == 0x00){		-	○	-	-
	else		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ucs_m_logger_data_clear	0~2	0	1	1	2
	DataLogger_clear()	0~3	2	0	1	3
Output	ucs_m_logger_data_clear	0~2	0	0	1	2
Judgment			OK	OK	OK	OK

LoggerGXForce_Clear_LoggerPowerClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(ucs_m_logger_power_clear)		○	○	○	○
	case M_OFF		○	-	-	-
	case M_ON		-	○	○	-
	if(DataLogger_clearPower() == 0x00)		-	○	-	-
	else		-	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	ucs_m_logger_power_clear	0~2	0	1	1	2
	DataLogger_clearPower()	0~3	2	0	1	3
Output	ucs_m_logger_power_clear	0~2	0	0	1	2
Judgment			OK	OK	OK	OK

LoggerGXForce_Clear_PeakClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Get_FastBumpDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_denst_flg == 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_denst_flg	0~2	0	1	2
	puc_gas_num	0~6	0	5	6
	uls_logger_fast_bump_cal_denst[puc_gas_num]	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x0000000
Output	lul_ans	0x00000000	0x00000000	0x00000000	0x0000000
Judgment			OK	OK	OK

LoggerGXForce_Get_LoggerRun

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(st_datalogger_control.uc_state & DATALOGGER_RUN)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	st_datalogger_control.uc_state	0x00~0xFF	0x01	0x00	0xFF
Output	luc_ans	0,1	1	0	1
Judgment			OK	OK	OK

LoggerGXForce_Init_Logger

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	st_datalogger_control.uc_error_state	0x80	0x80
Judgment			OK

LoggerGXForce_Periodic_1secConfirm

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Set_AirEvent

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_gas_num	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

LoggerGXForce_Set_BUMPEvent

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_gas_num	0~17	0	16	17
	puc_pass_num	0~17	0	16	17
	puc_fast_flg	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

LoggerGXForce_Set_FastBumpDenst

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_gas_num	0~6	0	5	6
	puc_denst_flg	0~2	0	1	2
	pul_denst	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF	0x00000000
Output	uls_logger_fast_bump_cal_denst[puc_gas_num]	0x00000000	0x00000000	0x00000000	0x00000000
Judgment			OK	OK	OK

LoggerGXForce_Set_LoggerDefault

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(DataLogger_default() == 0x00)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	DataLogger_default()	0~12	0	11	12
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

LoggerGXForce_Set_LunchFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(puc_flg == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_flg	0~2	1	0	2
Output	st_datalogger_control.uc_lunch_break	0,1	1	0	0
Judgment			OK	OK	OK

LoggerGXForce_Set_PowerOff

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(DataLogger_power_off() != 0x0 && (st_datalogger_control.uc_state & DATALOGGER_FATAL) != DATALOGGER_FATAL)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	DataLogger_power_off()	0~15	15	0	14
	st_datalogger_control.uc_state	0x00~0xFF	0x00	0xFF	0x01
Output	luc_ans	0,1	1	0	1
Judgment			OK	OK	OK

LoggerGXForce_Set_PowerOn

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Set_SettingCheck

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Set_SpanEvent

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case3
Root	None		-
	Symbol	Range	
Input	puc_gas_num	0	0
Output	None	-	-
Judgment			OK

LoggerGXForce_Set_SnapShot

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Set_StartLoggerInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Set_StopLoggerInterval

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(DataLogger_stop() == 0x00)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	DataLogger_stop()	0~5	0	4	5
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

LoggerGXForce_Set_TestData

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

LoggerGXForce_Start_LoggerDataClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_logger_data_clear	1	1
Judgment			OK

LoggerGXForce_Start_LoggerPowerClear

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_logger_power_clear	1	1
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	lvd.c

Lvd_Cyclic_250msecProcess

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(SYSTEM.LVD1SR.BIT.LVD1MON == 1U)		○	-	○
	else		-	○	-
	if(fuc_lvd_count >= M_LVD_COUNT_MAX)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	fuc_lvd_count	0~21	0	20	21
	SYSTEM.LVD1SR.BIT.LVD1MON	0,1	1	0	1
Output	fuc_lvd_count	0,20	0	20	0
Judgment			OK	OK	OK

Lvd_Init_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	fuc_lvd_count	0	0
Judgment			OK

Lvd_Stop_Setting

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	main.c

m_Every_1000msec_Task_A

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Every_1000msec_Task_B

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(RTC_Read_DateAndTime() == M_NG)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	RTC_Read_DateAndTime()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_Every_1000msec_Task_C

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Every_1000msec_Task_D

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Every_250msec_TaskA

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(Battery_Chk_Battery() == M_NG)		○	-	-
	else		-	○	○
	if(FuncSDM_Get_ExCommandFlg() == M_ON)		○	-	-
	else		-	○	○
	if(KeyEvent_Chk_DispPowerOff() != M_OFF)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	Battery_Chk_Battery()	0~2	1	0	2
	FuncSDM_Get_ExCommandFlg()	0~2	1	0	2
	KeyEvent_Chk_DispPowerOff()	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_Every_250msec_TaskB

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Every_Time_Task

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

MAIN_Get_PowerOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None		-
Output	uc_m_power_off_flg	0	0
Judgment			OK

MAIN_Get_RomNum

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	us_m_sum_adr[0][0]	7328	7328
Judgment			OK

MAIN_Get_SumNum

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	us_m_sum_adr[1][0]	30582	30582
Judgment			OK

Main_Process

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

MAIN_Set_ForcedPowerOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root			
	Symbol	Range	
Input	None	-	-
Output	uc_m_forced_power_off_flg	1	1
Judgment			OK

Main_Set_Power_OFF

	Responsible	Last update
Coding	Ikarashi	2021/6/23
Test	Taki	
Judgment	Ikarashi	

Infinite loop	Yes
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_m_power_off_flg == M_ON)		○	○	○	○	-
	if((LoggerGXForce_Set_PowerOff() != ZIPC_NG) || (uc_m_forced_power_off_flg == M_ON))		○	○	○	-	-
	if((FRAM_Get_flg() != M_ON) || (uc_m_forced_power_off_flg == M_ON))		○	○	-	-	-
	if(uc_m_update_flg == M_ON)		○	-	-	-	-
	while(IntervalTimer_Flag_Acquisition_250msec() == M_OFF)		○	-	-	-	-
	else		-	○	-	-	-
	if((uc_m_update_flg == M_ON) && (ComCommand_Get_DeviceTransmitEnable() == false))		○	-	-	-	-
	else		-	○	-	-	-
	while(1U)		-	○	-	-	-
	else		-	–	○	-	-
	else		-	–	-	○	-
	else		-	–	-	-	○
	Symbol	Range					
Input	uc_m_power_off_flg	0,1	1	1	1	1	0
	LoggerGXForce_Set_PowerOff()	0,1	0	0	0	1	0
	uc_m_forced_power_off_flg	0,1	1	1	0	0	0
	FRAM_Get_flg()	0,1	0	0	1	1	0
	IntervalTimer_Flag_Acquisition_250msec()	0,1	0	0	1	1	1
	uc_m_update_flg	0,1	1	0	1	1	0
	ComCommand_Get_DeviceTransmitEnable()	0,1	0	0	0	0	1
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

Main_Set_Power_ON

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	PORT0.PODR.BIT.B5	1	1
Judgment			OK

MAIN_Set_PowerOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_power_off_flg	1	1
Judgment			OK

MAIN_Set_UpdateFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_update_flg	1	1
Judgment			OK

MAIN_Get_UpdateModeFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_update_mode_flg	0,1	0
Judgment			OK

MAIN_Set_UpdateModeFlg

	Responsible	Last update
Coding	Ikarashi	2021/6/30
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_update_mode_flg	0,1	1
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	Port.c

Port_Get_UsbPort_isOn

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	PORT3.PIDR.BIT.B5	0~2	0	1	2
Output	lbl_ans	0~2	0	1	0
Judgment			OK	OK	OK

Port_Init_Setting

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	Pressure.c

Pressure_Change_Device0OutPut

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	pus_output	0	0
	f_Check_DacOutPutPer()	0	0
Output	None	-	-
Judgment			OK

Pressure_Get_FlowValue

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	Ad12bit_Get_Ad_Data()	0	0
Output	lfl_ans	0	0
Judgment			OK

Pressure_Init_Process

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Pressure_Start_Device

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	pus_output	0	0
	f_Check_DacOutPutPer()	0	0
Output	None	-	-
Judgment			OK

Pressure_Stop_Device

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

f_Check_DacOutPutPer

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(lus_ans < PRESSURE_OUTPUT_MIN_PER)		○	○	-	-	-
	else		-	-	○	○	○
	if(lus_ans > PRESSURE_OUTPUT_MAX_PER)		-	-	-	-	○
	else		○	○	○	○	-
	Symbol	Range					
Input	pus_output	0~991	0	9	10	990	991
Output	lus_ans	10~991	10	10	10	990	991
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	pump.c

Pump_Control_PumpMove

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(fbl_back_active != pbl_active)		○	-	○	-	○	-	-	○	○
	else		-	○	-	○	-	○	○	-	-
	if((fuc_1000msec_count >= F_1000MSEC_COUNT_MAX) || (lbl_change_flag == true))		○	○	○	○	○	○	-	○	○
	if(lbl_change_flag == true)		○	-	○	-	○	-	-	○	○
	if(pbl_active == true)		-	-	-	-	-	-	-	○	○
	if(stp_sub_ad->fl_temp_data < F_PUMP_CHK_BOOST_TEMP)		-	-	-	-	-	-	-	-	○
	else		-	-	-	-	-	-	-	○	-
	else		○	-	○	-	○	-	-	-	-
	else		-	○	-	○	-	○	-	-	-
	else		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	pbl_active	0~2	0	0	0	1	2	1	0	1	1
	pbl_lohi	0	0	0	0	0	0	0	0	0	0
	fl_temp.data	-1,0	-1	-1	-1	-1	-1	0	-1	0	-1
	fbl_back_active	0,1	1	0	1	1	1	1	0	0	0
	fuc_1000msec_count	0~3	3	3	2	3	3	3	0	3	3
Output	fbl_back_active	0~2	0	0	0	1	2	1	0	1	1
	fuc_1000msec_count	0,1	0	0	0	0	0	0	1	0	0
	fuc_start_phase_flag	0~17	0	0	0	0	0	0	0	4	17
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

Pump_Get_Phase_isNormalOrEnd

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((fuc_start_phase_flag == FEN_PUMP_START_PHASE_NORMAL) || (fuc_start_phase_flag == FEN_PUMP_START_PHASE_END))		○	○	-
	else		-	-	○
	Symbol	Range			
Input	fuc_start_phase_flag	0~2	0	1	2
Output	lbl_phase_flag	0,1	1	1	0
Judgment			OK	OK	OK

Pump_Init_Process

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	fbl_back_active	0	0
	fuc_1000msec_count	0	0
	fuc_start_phase_flag	0	0
Judgment			OK

f_Drive_Set

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pbl_onoff == true)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pbl_onof	0~2	0	1	2
Output	PORTA.PODR.BIT.B3	0,1	0	1	0
Judgment			OK	OK	OK

f_LoHi_Set

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(pbl_lohi == true)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pbl_lohi	0~2	0	1	2
Output	PORTA.PODR.BIT.B4	0,1	0	1	0
Judgment			OK	OK	OK

f_Power_Set

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(pbl_onoff == true)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pbl_onof	0~2	0	1	2
Output	PORTA.PODR.BIT.B7	0,1	0	1	0
Judgment			OK	OK	OK

f_Set_PumpOperation

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1	case2	case3
Root	if(pbl_onoff == true)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pbl_onof	0~2	0	1	2
	pbl_lohi	0	0	0	0
Output	None	-	-	-	-
Judgment			OK	OK	OK

f_Phase_PumpMove

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	20

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(fuc_start_phase_flag != (U8_CHR)FEN_PUMP_START_PHASE_END)		○	○	○	○	○	○
	switch(fuc_start_phase_flag)		○	-	○	○	○	○
	case FEN_PUMP_START_PHASE_NORMAL:		○	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP3_2:		-	-	○	-	-	-
	case FEN_PUMP_START_PHASE_STOP3_1:		-	-	-	○	-	-
	case FEN_PUMP_START_PHASE_STOP3_0:		-	-	-	-	○	-
	case FEN_PUMP_START_PHASE_STOP2_2:		-	-	-	-	-	○
	case FEN_PUMP_START_PHASE_STOP2_1:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP2_0:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP1_2:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP1_1:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP1_0:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP0_2:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP0_1:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_STOP0_0:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_MOVE3:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_MOVE2:		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_MOVE1:		-	-	-	-	-	-
	if(PumpFlow_Get_BoostCheck_isRetry() == true){		-	-	-	-	-	-
	else		-	-	-	-	-	-
	case FEN_PUMP_START_PHASE_START:		-	-	-	-	-	-
	default:		-	-	-	-	-	-
	Symbol	Range						
Input	pbl_lohi	0	0	0	0	0	0	0
	PumpFlow_Get_BoostCheck_is_Retry()	1,2	1	1	1	1	1	1
	fuc_start_phase_flag	0~18	1	0	2	3	4	6
Output	fuc_start_phase_flag	0~17	0	0	1	2	3	5
Judgment			OK	OK	OK	OK	OK	OK

case7	case8	case9	case10	case11	case12	case13	case14	case15	case16	case17	case18	case19	case20
○	○	○	○	○	○	○	○	○	○	○	○	○	○
○	○	○	○	○	○	○	○	○	○	○	○	○	○
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-
-	○	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	-	-	-	-	-	-	-	-	-	-	-
-	-	-	○	-	-	-	-	-	-	-	-	-	-
-	-	-	-	○	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	○	-	-	-	-	-	-	-
-	-	-	-	-	-	-	○	-	-	-	-	-	-
-	-	-	-	-	-	-	-	○	-	-	-	-	-
-	-	-	-	-	-	-	-	-	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	○	-	-	○
-	-	-	-	-	-	-	-	○	○	○	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	○
-	-	-	-	-	-	-	-	-	-	-	○	-	-
-	-	-	-	-	-	-	-	-	-	-	-	○	-
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	2
7	8	10	11	12	14	15	16	5	9	13	17	18	5
6	7	9	10	11	13	14	15	4	8	12	16	17	0
OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	PumpFlow.c

PumpFlow_Check_FlowTest

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(fbl_pump_move_flag == true)		○	-	○
	if(Pump_Get_Phase_isNormalOrEnd() == true)		○	-	-
	else		-	-	○
	else		-	○	-
	Symbol	Range			
Input	f_Check_FlowFailInPsSensor()	0	0	0	0
	Pump_Get_Phase_isNormalOrEnd()	1,2	1	1	2
	fbl_pump_move_flag	1,2	1	2	1
Output	lbl_result	0	0	0	0
Judgment			OK	OK	OK

PumpFlow_Check_PumpTest

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_pump_test_result	0x0000~0xFFFF	0	1	65535
Output	fbl_pump_test_result	0x0000~0xFFFF	0	1	65535
Judgment			OK	OK	OK

PumpFlow_Check_PurgeActive

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_ch=F_INIT_VAL; luc_ch<F_GAS_NUM_MAX; luc_ch++)		○	○	○	○
	if(((stp_GasInfo_addr+luc_ch)->uc_gas_comb != 0x00U) && (((stp_denst+luc_ch) ->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN)) == 0x00U))		○	○	○	○
	if(f_Check_PurgeLimitCheck_isNeed(luc_ch) == true)		○	○	○	-
	else		-	-	-	○
	else		-	-	-	-
	Symbol	Range				
Input	uc_state_trb[0]	0,4	0	0	4	0
	f_Check_PurgeLimitCheck_isNeed()	1,2	1	1	1	2
	uc_gas_comb[0]	0,1	1	0	1	1
Output	lbl_ans	0	0	0	0	0
Judgment			OK	OK	OK	OK

PumpFlow_Control_DriveMode

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(fbl_pump_move_flag != fbl_req_move_flag)		○	-	-	-	○
	else		-	-	○	○	-
	if(fbl_req_move_flag == true)		-	-	○	○	-
	if(fbl_pump_lohi_flag != fbl_req_lohi_flag)		-	-	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	-
	if(lbl_change_flag == true)		○	-	○	-	○
	if(fbl_pump_move_flag == true)		-	-	○	-	-
	else		-	-	-	-	○
	else		○	-	-	-	-
	Symbol	Range					
Input	fbl_pump_lohi_flag	0,1	1	1	1	0	1
	fbl_pump_move_flag	0,1	1	0	1	1	1
	fbl_req_lohi_flag	0	0	0	0	0	0
	fbl_req_move_flag	0~2	0	0	1	1	2
Output	fbl_pump_lohi_flag	0,1	0	1	0	0	0
	fbl_pump_move_flag	0~2	0	0	1	1	2
Judgment			OK	OK	OK	OK	OK

PumpFlow_Cyclic_250msec

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lbl_flag == true)		○	-	-
	else		-	○	○
	if(lbl_flag == true)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	f_Check_FlowFail()	1,2	1	1	2
	f_Check_PumpFail()	1,2	1	2	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

PumpFlow_Default_Device

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
	us_flow_output[0]	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

PumpFlow_Get_BoostCheck_isRetry

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	fbl_boost_check_ans	0x0000,0xFFFF	0x0000	0xFFFF
Output	fbl_boost_check_ans	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

PumpFlow_Get_DropOutPut

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_data->us_flow_rate_drop_output	0x0000,0xFFFF	0x0000	0xFFFF
Output	stp_data->us_flow_rate_drop_output	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

PumpFlow_Get_FlowOutPut

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	stp_data->us_flow_output	0x0000,0xFFFF	0x0000	0xFFFF
Output	stp_data->us_flow_output	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

PumpFlow_Get_LoHiFlag_isHi

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_req_lohi_flag	0~2	0	1	2
Output	fbl_req_lohi_flag	0~2	0	1	2
Judgment			OK	OK	OK

PumpFlow_Get_MotionFlag_isMove

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	fbl_req_move_flag	0~2	0	1	2
Output	fbl_req_move_flag	0~2	0	1	2
Judgment			OK	OK	OK

PumpFlow_Get_PumpFailPressValue

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	ffls_pump_fail_val	0~2	0	1	2
Output	ffls_pump_fail_val	0~2	0	1	2
Judgment			OK	OK	OK

PumpFlow_Get_PumpOffPressValue

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	ffls_pump_off_flow_val	0~2	0	1	2
Output	ffls_pump_off_flow_val	0~2	0	1	2
Judgment			OK	OK	OK

PumpFlow_Get_SettingData

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	*pstp_data	1	1
Output	None	-	-
Judgment			OK

PumpFlow_Init_Process

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		
	Symbol	Range	
Input	-	-	-
Output	fbl_boost_check_ans	0	0
	fbl_pump_lohi_flag	0	0
	fbl_pump_move_first	0	0
	fbl_pump_move_flag	0	0
	fbl_pump_test_result	0	0
	fbl_req_lohi_flag	0	0
	fbl_req_move_flag	0	0
	ffls_pump_fail_val	0	0
	ffls_pump_off_flow_val	0	0
Judgment			OK

PumpFlow_Set_FlowOutPut

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case1
Root	None		-	-
	Symbol	Range		
Input	pus_num	0x0000,0xFFFF	0x0000	0xFFFF
Output	stp_data->us_flow_output	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

PumpFlow_Set_LoHiRequest_isHi

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pbl_lohi == true)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pbl_lohi	0~2	0	1	2
Output	fbl_req_lohi_flag	0,1	0	1	0
Judgment			OK	OK	OK

PumpFlow_Set_MotionRequest_isMove

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pbl_move == true		-	○	-
	else		○	-	○
	Symbol	Range			
Input	pbl_move	0~2	0	1	2
Output	fbl_req_move_flag	0,1	0	1	0
Judgment			OK	OK	OK

PumpFlow_Set_SettingData

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

PumpFlow_Set_SettingDefaultData

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

PumpFlow_Start_Device

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	stp_data->us_flow_output	0	0
Output	None	-	-
Judgment			OK

PumpFlow_Stop_Device

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

f_Check_FlowFail

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(fbl_pump_move_flag == true)		○	-	○	○	○	○	○
	if(Pump_Get_Phase_isNormalOrEnd() == true)		○	-	-	○	○	○	○
	if((FlowMain_Get_ModeStatus() == FlowMain_INITIAL_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))		○	-	-	○	○	-	○
	if(lbl_h2_meas == false)		○	-	-	○	○	-	-
	else		-	-	-	-	-	-	○
	else		-	-	-	-	-	○	-
	else		-	-	○	-	-	-	-
	if(FlowProcesDisplay_Get_PurgeFlag() == true)		○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-
	Symbol	Range							
Input	f_Check_FlowFailInPsSensor()	0	0	0	0	0	0	0	0
	FlowMain_Get_ModeStatus()	0~3	0	0	0	1	2	3	0
	FlowProcesDisplay_Get_PurgeFlag()	1,2	1	1	1	1	1	1	1
	GasCalc_Get_H2MeasurementFlag()	0,1	0	0	0	0	0	0	1
	Pump_Get_Phase_isNormalOrEnd()	1,2	1	1	2	1	1	1	1
	fbl_pump_move_flag	1,2	1	2	1	1	1	1	1
Output	lbl_ans	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

f_Check_FlowFailInPsSensor

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(lfl_ps_ad > (FLOAT32)stp_data->us_flow_rate_drop_output)		○	-
	else		-	○
	Symbol	Range		
Input	stp_data->us_flow_rate_drop_output	0	0	0
	Pressure_Get_FlowValue()	0	0	0
	ffls_pump_off_flow_val	0,1	1	0
Output	lbl_ans	0,1	1	0
	ffls_pump_fail_val	1	1	1
Judgment			OK	OK

f_Check_PumpFail

	Responsible	Last update
Coding	lkarashi	2021/7/1
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(fbl_pump_move_flag == true)		○	-	○	○	○	○	○
	if(Pump_Get_Phase_isNormalOrEnd() == true)		○	-	-	○	○	-	○
	if(fbl_pump_move_first == true)		○	-	-	-	○	-	○
	if(fbl_boost_check_ans == true)		○	-	-	-	-	-	-
	else		-	-	-	-	○	-	○
	else		-	-	-	○	-	-	-
	else		-	-	○	-	-	○	-
	if(lfl_ps_ad < F_PUMP_CHK_MIN)		○	-	○	-	-	-	-
	else		-	-	-	-	-	○	-
	if(FlowProcesDisplay_Get_PurgeFlag() == true)		○	○	○	○	○	○	-
	else		-	-	-	-	-	-	○
	Symbol	Range							
Input	Pressure_Get_FlowValue()	0	0	0	0	0	0	0	0
	FlowProcesDisplay_Get_PurgeFlag()	1,2	1	1	1	1	1	1	2
	Pump_Get_Phase_isNormalOrEnd()	1,2	1	1	2	1	1	2	1
	fbl_boost_check_ans	1,2	1	1	1	1	2	1	2
	fbl_pump_move_first	1,2	1	1	1	2	1	1	1
	fbl_pump_move_flag	1,2	1	2	1	1	1	1	1
	ffls_pump_off_flow_val	4,5	4	4	4	4	4	5	4
Output	lbl_ans	0	0	0	0	0	0	0	0
	fbl_boost_check_ans	0,1	0	1	1	1	0	0	0
	fbl_pump_move_first	0,1	0	1	1	0	0	1	0
	fbl_pump_test_result	1	1	1	1	1	1	1	1
	ffls_pump_fail_val	0,4	0	0	4	4	4	4	4
Judgment			OK	OK	OK	OK	OK	OK	OK

f_Check_PurgeLimitCheck_isNeed

	Responsible	Last update
Coding	lkarashi	2021/7/1
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(stp_denst->ul_denst != M_OFF_VAL)		○	-	○	○	○	○
	if(stp_denst->uc_denst_minus_flg == M_OFF)		○	-	-	○	○	○
	else		○	-	-	-	○	○
	if(puc_ch == DataStructConst_O2_GAS_CH)		○	-	-	-	-	-
	else		-	-	-	-	○	○
	if((lul_purge_threshold < lul_denst) && (lbl_denst_minus_flg == false))		-	-	-	○	-	-
	else		-	-	○	-	-	-
	else		○	-	○	-	-	-
	Symbol	Range						
Input	puc_ch	1,2	2	2	2	1	2	2
	stp_denst->ul_denst	1~3722305990	3722304990	3722304989	3722304990	3722304990	1	2
	stp_denst->uc_denst_minus_flg	0,1	0	0	1	1	0	0
	stp_GasInfo_addr->ul_full_scale	20	20	20	20	20	20	20
Output	lbl_ans	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

f_Get_OffFlowValue

	Responsible	Last update
Coding	lkarashi	2021/7/1
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	Pressure_Get_FlowValue()	0~2	0	1	2
Output	fbl_pump_move_first	1	1	1	1
	fbl_pump_test_result	0	0	0	0
	ffls_pump_off_flow_val	0~2	0	1	2
Judgment			OK	OK	OK

f_Set_FlowFail

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

f_Set_PumpFail

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_cgc.c

R_CGC_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_cmt.c

R_CMT0_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_CMT0_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_CMT0_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_cmt_user.c

r_cmt_cmi0_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(SelfSubProg_Get_10msecChgFlg() == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	SelfSubProg_Get_10msecChgFlg()	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_hardware_setup.c

HardwareSetup

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_Systeminit

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_intprg.c

r_access_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_brk_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_floatingpoint_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_nmi_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_privileged_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_reserved_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_undefined_exception

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_lvd.c

R_LVD1_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_LVD1_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_LVD1_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root			-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_main.c

main

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	Yes
Verification pass	1
Judgment	OK

			case1
Root	while(1U)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_MAIN_UserInit

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_mtu2.c

R_MTU2_C2_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_MTU2_C2_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_MTU2_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_mtu2_user.c

r_mtu2_tgia2_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_port.c

R_PORT_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_r12da.c

R_R12DA_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_R12DA0_Set_ConversionValue

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_R12DA0_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_R12DA0_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_resetprg.c

PowerON_Reset_PC

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_rspi.c

R_RSPI0_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_RSPI0_Send_Receive

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2
Root	if(tx_num < 1U)		○	-
	else		-	○
	Symbol	Range		
Input	tx_num	0x0000~0xFFFF	0x0000	0xFFFF
	tx_buf	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
	rx_buf	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
Output	gp_rspi0_tx_address	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
	g_rspi0_tx_count	0x0000,0xFFFF	0x0000	0xFFFF
	gp_rspi0_rx_address	0x00000000,0xFFFFFFFF	0x00000000	0xFFFFFFFF
	g_rspi0_rx_length	0x0000,0xFFFF	0x0000	0xFFFF
	g_rspi0_rx_count	0x0000	0x0000	0x0000
	status	0x00,0x81	0x81	0x00
Judgment			OK	OK

R_RSPI0_Send_Receive_Copy

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(RSPI0.SPCR.BIT.SPE == 1U)		○	-	-	-
	else		-	○	○	○
	if(tx_num < 1U)		-	○	-	-
	else		-	-	○	○
	if(ch_num == 0)		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	tx_num	0x0000,0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF
	tx_buf	0x00,0xFF	0x00	0xFF	0x00	0x00
	rx_buf	0x0000,0xFFFF	0x0000	0xFFFF	0x0000	0x0000
	rx_num	0x00,0xFF	0x00	0xFF	0x00	0x00
	ch_num	0~2	2	0	0	1
	RSPI0.SPCR.BIT.SPE	0~2	1	0	2	2
Output	puc_copy_rspi_tx_address	0x00	0x00	0x00	0x00	0x00
	g_rspi0_tx_count	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
	puc_copy_rspi_rx_address	0x0000,0xFFFF	0xFFFF	0xFFFF	0x0000	0x0000
	g_rspi0_rx_length	0x00	0x00	0x00	0x00	0x00
	g_rspi0_rx_count	0x0000	0x0000	0x0000	0x0000	0x0000
	status	0x00,0x81	0x81	0x81	0x00	0x00
Judgment			OK	OK	OK	OK

R_RSPI0_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_RSPI0_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_rspi_user.c

r_rspi0_callback_error

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	err_type	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

r_rspi0_callback_receiveend

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_rspi0_callback_transmitend

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_rspi0_error_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(1U == RSPI0.SPSR.BIT.MODF)		○	-	-
	while(1U != PORTC.PIDR.BIT.B4)		○	-	-
	else		-	○	○
	if(0U != err_type)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	RSPI0.SPSR.BIT.MODF	0~2	1	0	2
	PORTC.PIDR.BIT.B4	0~2	0	1	2
	RSPI0.SPSR.BYTE	0x00~0xA1	0xA0	0	0xA1
Output	None	-	-	-	-
Judgment			OK	OK	OK

r_rspi0_idle_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_rspi0_receive_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for (frame_cnt = 0U; frame_cnt < (_00_RSPI_FRAMES_1 + 1U); frame_cnt++)		○	○	-
	if(g_rspi0_rx_length > g_rspi0_rx_count)		○	○	-
	if(g_rspi0_rx_length == g_rspi0_rx_count)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	g_rspi0_rx_length	0x0000~0xFFFF	0x0001	0xFFFF	0x0000
	g_rspi0_rx_count	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
Output	puc_copy_rspi_rx_address	1,2	0	0	0
Judgment			OK	OK	OK

r_rspi0_transmit_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	for (frame_cnt = 0U; frame_cnt < (_00_RSPI_FRAMES_1 + 1U); frame_cnt++)		○	○
	if(g_rspi0_tx_count > 0U)		○	-
	else		-	○
	Symbol	Range		
Input	g_rspi0_tx_count	0x0000,0xFFFF	0xFFFF	0x0000
Output	None	-	-	-
Judgment			OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_s12ad.c

R_S12AD_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_S12AD_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_S12AD_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_S12AD_Get_ValueResult

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification paths	29
Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11
Root	if(channel == ADSELFDIAGNOSIS)	○	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL0)	-	○	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL1)	-	-	○	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL2)	-	-	-	○	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL3)	-	-	-	-	○	-	-	-	-	-	-
	else if(channel == ADCHANNEL4)	-	-	-	-	-	○	-	-	-	-	-
	else if(channel == ADCHANNEL5)	-	-	-	-	-	-	○	-	-	-	-
	else if(channel == ADCHANNEL6)	-	-	-	-	-	-	-	○	-	-	-
	else if(channel == ADCHANNEL7)	-	-	-	-	-	-	-	-	○	-	-
	else if(channel == ADCHANNEL16)	-	-	-	-	-	-	-	-	-	○	-
	else if(channel == ADCHANNEL17)	-	-	-	-	-	-	-	-	-	-	○
	else if(channel == ADCHANNEL18)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL19)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL20)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL21)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL22)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL23)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL24)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL25)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL26)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL27)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL28)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL29)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL30)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADCHANNEL31)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADTEMPSENSOR)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADINTERREFVOLT)	-	-	-	-	-	-	-	-	-	-	-
	else if(channel == ADDATADUPLICATION)	-	-	-	-	-	-	-	-	-	-	-
	else	-	-	-	-	-	-	-	-	-	-	-
	Symbol	Range										
Input	channel	0~28	0	1	2	3	4	5	6	7	8	9	10
	buffer	0x0000,0xFFFF	0x0000	0xFFFF	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
Output	None	"	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

case12	case13	case14	case15	case16	case17	case18	case19	case20	case21	case22	case23	case24	case25	case26	case27	case28	case29
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_s12ad_user.c

r_s12ad_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/1
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_sci.c

R_SCI5_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI5_Serial_Receive

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(1U > rx_num)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	rx_num	0~2	0	1	2
	rx_buf	0x00,0xFF	0x00	0x00	0xFF
Output	g_sci5_rx_count	0	0	0	0
	g_sci5_rx_length	0~2	0	1	2
	gp_sci5_rx_address	0x00,0xFF	0x00	0x00	0xFF
	status	0x00,0x81	0x81	0x00	0x00
Judgment			OK	OK	OK

R_SCI5_Serial_Send

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(1U > tx_num)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	tx_num	0~3	0	2	3
	tx_buf	0x00,0xFF	0x00	0x00	0xFF
Output	g_sci5_tx_count	0~3	0	2	3
	gp_sci5_tx_address	0x00,0xFF	0x00	0x00	0xFF
	status	0x00,0x81	0x81	0x00	0x00
Judgment			OK	OK	OK

R_SCI5_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI5_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI6_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI6_IIC_Master_Receive

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(1U > rx_num)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	adr	0x00,0xFF	0x00	0x00	0xFF
	rx_buf	0x00,0xFF	0x00	0x00	0xFF
	rx_num	0~8	0	7	8
Output	g_sci6_rx_length	0~8	0	7	8
	g_sci6_rx_count	0	0	0	0
	gp_sci6_rx_address	0x00,0xFF	0x00	0x00	0xFF
	g_sci6_iic_transmit_receive_flag	0x00	0x00	0x00	0x00
	g_sci6_iic_cycle_flag	0x00,0x80	0x00	0x80	0x80
Judgment			OK	OK	OK

R_SCI6_IIC_Master_Send

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(1U > tx_num)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	adr	0x00,0xFF	0x00	0x00	0xFF
	tx_buf	0x00,0xFF	0x00	0x00	0xFF
	tx_num	0~4324	0	4323	4324
Output	gp_sci6_tx_address	0x00,0xFF	0x00	0x00	0xFF
	g_sci6_tx_count	0~4324	0	4323	4324
	g_sci6_slave_address	0x00,0xFF	0x00	0x00	0xFF
	g_sci6_iic_transmit_receive_flag	0x00,0x80	0x00	0x80	0x80
	g_sci6_iic_cycle_flag	0x00,0x80	0x00	0x80	0x80
Judgment			OK	OK	OK

R_SCI6_IIC_StartCondition

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI6_IIC_StopCondition

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI6_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI6_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI9_Create

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI9_Serial_Receive

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(1U > rx_num)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	rx_buf	0x00,0xFF	0x00	0x00	0xFF
	rx_num	0~2	0	1	2
Output	g_sci9_rx_count	0	0	0	0
	g_sci9_rx_length	0~2	0	1	2
	gp_sci9_rx_address	0x00,0xFF	0x00	0x00	0xFF
	status	0x00,0x81	0x81	0x00	0x00
Judgment			OK	OK	OK

R_SCI9_Serial_Send

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(1U > tx_num)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	tx_buf	0x00,0xFF	0x00	0x00	0xFF
	tx_num	0~22	0	21	22
Output	gp_sci9_tx_address	-	Indefinite value	Indefinite value	Indefinite value
	g_sci9_tx_count	0~22	0	21	22
	status	0x00,0x81	0x81	0x00	0x00
Judgment			OK	OK	OK

R_SCI9_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

R_SCI9_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_sci_user.c

r_sci5_callback_receiveend

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci5_callback_receiveerror

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci5_callback_transmitend

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci5_receive_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3
Root	if(g_sci5_rx_length > g_sci5_rx_count)		○	○	-
	if(g_sci5_rx_length <= g_sci5_rx_count)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	g_sci5_rx_length	0x0000~0xFFFF	0x0001	0xFFFF	0x0000
	g_sci5_rx_count	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
Output	None	-	-	-	
Judgment			OK	OK	OK

r_sci5_receiveerror_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci5_transmit_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2
Root	if(0U < g_sci5_tx_count)		○	-
	else		-	○
	Symbol	Range		
Input	g_sci5_tx_count	0x0000~0xFFFF	0xFFFF	0x0000
Output	None	-	-	-
Judgment			OK	OK

r_sci5_transmitend_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci6_callback_receiveend

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci6_callback_transmitend

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci6_receive_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(g_sci6_rx_length > g_sci6_rx_count)		○	-
	else		-	○
	Symbol	Range		
Input	g_sci6_rx_length	0x0000~0xFFFF	0xFFFF	0x0000
	g_sci6_rx_count	0x0000~0xFFFF	0x0000	0xFFFF
Output	None	-	-	-
Judgment			OK	OK

r_sci6_transmit_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(0U == SCI6.SISR.BIT.IICACKR)		○	○	○	○	○	-	-
	if(_80_SCI_IIC_TRANSMISSION == g_sci6_iic_transmit_receive_flag)		○	○	-	-	-	-	-
	if(0U < g_sci6_tx_count)		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	else if(_00_SCI_IIC_RECEPTION == g_sci6_iic_transmit_receive_flag)		-	-	○	○	-	-	-
	if(g_sci6_rx_length == (g_sci6_rx_count + 1))		-	-	○	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	○
	Symbol	Range							
Input	SCI6.SISR.BIT.IICACKR	0~2	0	0	0	0	0	1	2
	g_sci6_iic_transmit_receive_flag	0x00~0x81	0x80	0x80	0x00	0x00	0x81	0x00	0x00
	g_sci6_tx_count	0x0000,0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0x0000	0x000	0x000
	g_sci6_rx_count	0x0000~0xFFFF	0xFFFF	0x0000	0x0001	0xFFFF	0x0000	0x000	0x000
	g_sci6_rx_length	0x0000~0xFFFF	0xFFFF	0x0000	0x0001	0x0000	0x0000	0x000	0x000
Output	g_sci6_iic_cycle_flag	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

r_sci6_transmitend_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(_80_SCI_IIC_START_CYCLE == g_sci6_iic_cycle_flag)		○	-	-	-
	else if(_00_SCI_IIC_STOP_CYCLE == g_sci6_iic_cycle_flag)		-	○	○	-
	if(_80_SCI_IIC_TRANSMISSION == g_sci6_iic_transmit_receive_flag)		-	○	-	-
	else		-	-	○	-
	if(_00_SCI_IIC_RECEPTION == g_sci6_iic_transmit_receive_flag)		-	-	○	-
	else		-	○	-	-
	else		-	-	-	○
	Symbol	Range				
Input	g_sci6_iic_cycle_flag	0x00~0x81	0x80	0x00	0x00	0x81
	g_sci6_iic_transmit_receive_flag	0x00~0x81	0x81	0x80	0x00	0x00
	g_sci6_slave_address	0x00~0xFF	0xFF	0x00	0x00	0x00
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

r_sci9_callback_receiveend

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_use_sci_rx_for_rl78prog == TRUE)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_use_sci_rx_for_rl78prog	0~2	1	0	2
Output	uc_use_sci_rx_for_rl78prog	0,2	0	0	2
Judgment			OK	OK	OK

r_sci9_callback_receiveerror

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci9_callback_transmitend

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_use_sci_rx_for_rl78prog == TRUE)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_use_sci_tx_for_rl78prog	0~2	1	0	2
Output	uc_use_sci_tx_for_rl78prog	0,2	0	0	2
Judgment			OK	OK	OK

r_sci9_receive_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(g_sci9_rx_length > g_sci9_rx_count)		○	○	-
	if(g_sci9_rx_length <= g_sci9_rx_count)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	g_sci9_rx_length	0x0000~0xFFFF	0xFFFF	0x0001	0x0000
	g_sci9_rx_count	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

r_sci9_receiveerror_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

r_sci9_transmit_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(0U < g_sci9_tx_count)		○	-
	else		-	○
	Symbol	Range		
Input	g_sci9_tx_count	0x0000~0xFFFF	0xFFFF	0x0000
Output	None	-	-	-
Judgment			OK	OK

r_sci9_transmitend_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_cg_wdt.c

R_WDT_Restart

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	r_spi.c

m_Wait_Spi_Send_Time

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	while(uc_m_spi_read_flg == 1)		○	-	-
	if(uc_m_time_busy == 0)		○	-	-
	else		-	-	-
	Symbol	Range			
Input	uc_m_spi_read_flg	0~2	1	0	2
Output	luc_ack	0,1	1	0	0
Judgment			OK	OK	OK

RSPI_count_down

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_time_busy)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	uc_m_time_busy	0~6	5	0	6
Output	uc_m_time_busy	0~5	4	0	5
Judgment			OK	OK	OK

RSPI_Int_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

RSPI_Int_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

RSPI_Send_Receive

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(((0 < pus_send_size) && (pus_send_size < M_DATA_SIZE+1))		○	-	○	○
	if(luc_ack[0] != RSPI_ACK || luc_ack[1] != RSPI_ACK)		○	-	-	-
	else		-	-	○	○
	else		-	○	-	-
	Symbol	Range				
Input	pus_send_size	0x0000~0xFFFF	0x0001	0x0001	0xFFFF	0x0000
	ppuc_receive	0x00~0xFF	0x00	0xFF	0x00	0x00
	puc_ch	0~2	0	1	2	0
	pus_send	0x0000,0xFFFF	0x0000	0x0000	0x0000	0xFFFF
	R_RSPI0_Send_Receive_Copy(ppuc_send, pus_send_size, ppuc_receive, pus_send_size, puc_ch)	0~0x82	0x81	0x00	0x82	0x00
	m_Wait_Spi_Send_Time()	0~2	1	0	2	0
Output	luc_ack[0]	0,1	1	0	1	1
	uc_m_spi_read_flg	1	1	1	1	1
Judgment			OK	OK	OK	OK

RSPI_write_chk_off

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_spi_read_flg	0	0
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	ref_volt.c

RefVolt_Cyclic_250msec

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case1	case1
Root	if(lbl_flag == true)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	f_Check_RefVoltFail()	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

RefVolt_Init_HcvCheckDisableFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	bl_hcv_check_disable_flg	0	0
Judgment			OK

RefVolt_Set_HcvCheckDisableFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	bl_hcv_check_disable_flg	1	1
Judgment			OK

f_Set_RefVoltFail

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

f_Check_RefVoltFail

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lbl_ad_init_flag == false)		○	-	○	○
	if((luc_ad_port == AD12BIT_AN025) && (bl_hcv_check_disable_flg == true))		-	-	-	-
	else		○	-	○	○
	if((lfl_refv_ad < flc_refv_threshold[0][luc_ch]) ‖ (lfl_refv_ad > flc_refv_threshold[1][luc_ch]))		○	-	○	○
	else		-	-	-	-
	else		-	○	-	-
	Symbol	Range				
Input	Ad12bit_Check_InitalProcessFlag()	0,1	0	1	0	0
	ucc_refv_ad_port[0]	0	0	0	0	0
	Ad12bit_Get_Ad_Data()	13,191,320	1320	1320	1320	1319
	bl_hcv_check_disable_flg	1,2	1	1	2	1
	flc_refv_threshold[0][0]	0	0	0	0	0
Output	lbl_ans	0,1	1	0	1	1
Judgment			OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	rl78i1e_control.c

RL78I1EControl_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_com_delay_time)		○	○	-
	else		-	-	○
	Symbol	Range			
Input	uc_m_com_delay_time	0~6	5	6	0
Output	None	-	-	-	-
Judgment			OK	OK	OK

RL78I1EControl_Calc_H2CancelCOAdAverage

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<2; luc_i++)		○	○	○
	for(luc_j=0; luc_j<M_H2_CANCEL_CO_AVE_SEC; luc_j++)		○	○	○
	Symbol	Range			
Input	fl_m_h2cancelco_ave_ad[luc_i][luc_j]	-800~801	-800	801	800
Output	RL78DATA.RL78_AD_DATA.fl_h2_cancel_co_ad[luc_i]	-12.5~12.51563	-12.5	12.51563	12.5
Judgment			OK	OK	OK

RL78I1EControl_Calc_H2CancelCOAdSUB

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(++uc_m_h2cancelcoave_num>=M_H2_CANCEL_CO_AVE_SEC)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	pfl_ad_data_1	-800~800	800	-800	801
	pfl_ad_data_2	-800~800	-800	800	-801
	uc_m_h2cancelcoave_num	0~64	63	0	64
Output	uc_m_h2cancelcoave_num	0,1	0	1	0
	fl_m_h2cancelco_ave_ad [0][uc_m_h2cancelcoave_num]	800,801	800	800	801
	fl_m_h2cancelco_ave_ad [1][uc_m_h2cancelcoave_num]	-801,-800	-800	-800	-801
Judgment			OK	OK	OK

RL78I1EControl_Calc_Temp

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(lfl_num == 0)		○	-	-	-	-	-	-
	else		-	○	○	○	○	○	○
	if(lfl_num > M_TEMP_MAX)		-	○	-	○	-	-	-
	else		-	-	○	-	-	○	-
	if(lfl_num < M_TEMP_MIN)		-	-	○	○	○	○	○
	else		-	○	-	-	-	-	-
	if((stp_GasInfo_addr+0)		-	○	-	-	-	-	-
	else		-	-	○	○	○	○	○
	if((stp_GasInfo_addr+4)		-	○	-	-	-	-	-
	else		-	-	○	○	○	○	○
	Symbol	Range							
Input	RL78DATA.RL78_AD_DATA.fl_temp_ad_data	164.5~272.5	207.5	261.5	183.5	165.92	271.76	164.5	272.5
	(stp_GasInfo_addr+0)->uc_gas_comb	0~3	3	2	0	0	0	0	0
	(stp_GasInfo_addr+4)->uc_gas_comb	0~3	3	1	0	0	0	0	0
Output	RL78DATA.RL78_AD_DATA.fl_temp_data	-50~70	0	70	-31.74	-50	70	-50	70
	RL78DATA.RL78_AD_DATA.fl_disp_temp	-50~70	0	68	-31.74	-50	70	-50	70
Judgment			OK	OK	OK	OK	OK	OK	OK

RL78I1EControl_Chk_TempError

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(uc_m_temp_chk_start_flg == M_ON)		○	-	○	-
	if(RL78DATA.RL78_AD_DATA.fl_temp_ad_data > M_TEMP_ERR_MAX_mV || RL78DATA.RL78_AD_DATA.fl_temp_ad_data < M_TEMP_ERR_MIN_mV)		○	-	-	-
	else		-	-	○	-
	else		-	○	-	○
	Symbol	Range				
Input	uc_m_temp_chk_start_flg	0,1	1	0	1	0
	RL78DATA.RL78_AD_DATA.fl_temp_ad_data	-55~170	-55	85	170	86
Output	luc_ans	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

RL78I1EControl_Command_250_Loop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(uc_m_com_status >= RL78I1ECONTROL_CODE_HA)		○	○	○	○	○	○	-
	if(uc_m_sub_err_count > M_COM_ERROR_MAX)		○	-	-	-	-	-	-
	else		-	○	○	○	○	-	-
	if(RL78I1ESumChk_Chk_4Count() == M_ON)		-	○	-	-	-	-	-
	if(RL78I1ESumChk_Get_PhaseFlg() == RL78i1eSumChk_SUM_PHASE_START)		-	○	-	-	-	-	-
	else if(RL78I1ESumChk_Get_PhaseFlg() == RL78i1eSumChk_SUM_PHASE_WAIT)		-	-	○	-	-	-	-
	else if(RL78I1ESumChk_Get_PhaseFlg() == RL78i1eSumChk_SUM_PHASE_END)		-	-	-	○	-	-	-
	else		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	-
	else		-	-	-	-	-	-	○
	Symbol	Range							
Input	uc_m_com_status	0~7	6	7	6	6	6	6	0
	uc_m_sub_err_count	0~6	6	5	0	0	0	0	0
	RL78I1ESumChk_Chk_4Count()	0~2	0	1	1	1	1	0	2
	RL78I1ESumChk_Get_PhaseFlg()	0~4	0	1	2	3	4	0	0
Output	None	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK

RL78I1EControl_Command_Control

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(uc_m_com_status < RL78I1ECONTROL_CODE_HA)		○	○	○	○	○	○	○	○	-
	if(uc_m_sub_err_count > M_COM_ERROR_MAX)		○	-	-	-	-	-	-	-	-
	else		-	○	○	○	○	○	○	○	-
	if(uc_m_com_delay_time == 0)		-	○	○	○	○	○	○	-	-
	switch(uc_m_com_status)		-	○	○	○	○	○	○	-	-
	case RL78I1ECONTROL_CODE_NON		-	○	-	-	-	-	-	-	-
	case RL78I1ECONTROL_CODE_RC		-	-	○	-	-	-	-	-	-
	case RL78I1ECONTROL_CODE_VP		-	-	-	○	-	-	-	-	-
	case RL78I1ECONTROL_CODE_VSC		-	-	-	-	○	-	-	-	-
	case RL78I1ECONTROL_CODE_VSR		-	-	-	-	-	○	-	-	-
	default		-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	○
	Symbol	Range									
Input	uc_m_com_status	0~7	0	0	1	2	3	4	5	0	7
	uc_m_sub_err_count	0~5	6	0	0	0	0	0	0	0	5
	uc_m_com_delay_time	0~6	5	0	0	0	0	0	0	5	6
Output	uc_m_com_delay_time	5,6	5	5	5	5	5	5	5	5	6
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

RL78I1EControl_Command_ReSet

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_com_status	0	0
	uc_m_sub_err_count	0	0
Judgment			OK

RL78I1EControl_Get_Data_point

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	RL78DATA.RL78_AD_DATA	-	Indefinite value
Judgment			OK

RL78I1EControl_Get_MainData_point

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	RL78DATA	-	Indefinite value
Judgment			OK

RL78I1EControl_Set_Control_Status

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(uc_m_com_status >= RL78I1ECONTROL_CODE_HA)		○	○	-	-
	if(puc_num >= RL78I1ECONTROL_CODE_HA)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	for(luc_i=0; luc_i<RL78I1ECONTROL_CODE_HA; luc_i++)		-	-	○	○
	if((uc_m_com_status == luc_i) && (puc_num == (luc_i + 1)))		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	uc_m_com_status	0~7	7	6	5	0
	puc_num	5~7	7	5	6	6
Output	uc_m_sub_err_count	0	0	0	0	0
	uc_m_com_delay_time	0	0	0	0	0
	uc_m_com_status	0~7	7	6	6	0
Judgment			OK	OK	OK	OK

RL78I1EControl_Set_Gas_Status

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case1	case1
Root	None		-	-	-
	Symbol	Range			
Input	pucp_data	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

RL78I1EControl_Set_ROM_NUM

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pucp_data	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

RL78I1EControl_Set_ROM_SUM

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	pus_num	0x0000,0xFFFF	0x0000	0xFFFF
Output	RL78DATA.us_rom_sum	0x0000,0xFFFF	0x0000	0xFFFF
Judgment			OK	OK

RL78I1EControl_Set_ROM_VER

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puscp_data	0x0000,0xFFFF	0x0000	0xFFFF
Output	None	-	-	-
Judgment			OK	OK

RL78I1EControl_Start_ChkFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_temp_chk_start_flg	1	1
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	rl78i1e_sum_chk.c

RL78I1ESumChk_10msec_interrupt

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(ucs_m_sum_phase != RL78i1eSumChk_SUM_PHASE_OFF)		○	○	-
	if(uss_m_wait_count)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	uss_m_wait_count	0~1001	1000	0	1001
	ucs_m_sum_phase	0~4	3	4	0
Output	ucs_m_sum_phase	0,3	3	0	0
Judgment			OK	OK	OK

RL78I1ESumChk_Chk_4Count

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(ucs_m_sum_phase != RL78i1eSumChk_SUM_PHASE_OFF){		○	○	-
	if(++ucs_m_4count >= M_4COUNT_MAX)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	ucs_m_sum_phase	0~4	3	4	0
	ucs_m_4count	0~5	4	0	5
Output	luc_ans	0,1	1	0	0
Judgment			OK	OK	OK

RL78I1ESumChk_Get_PhaseFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_sum_phase	0	0
Judgment			OK

RL78I1ESumChk_Set_PhaseEnd

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_sum_phase	3	3
Judgment			OK

RL78I1ESumChk_Set_PhaseOFF

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_sum_phase	0	0
Judgment			OK

RL78I1ESumChk_Set_PhaseWait

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_sum_phase	2	2
Judgment			OK

RL78I1ESumChk_Set_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(ucs_m_sum_phase == RL78i1eSumChk_SUM_PHASE_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	ucs_m_sum_phase	0~4	0	3	4
Output	uss_m_wait_count	1000	1000	1000	1000
	ucs_m_sum_phase	1~4	1	3	4
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	rtc.c

m_data_check

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(M_YEAR_MAX < puc_year)		-	○	-
	else		○	-	○
	if(M_MONTH_MAX < puc_month || puc_month < M_MONTH_MIN)		-	○	-
	else		○	-	○
	if(M_DAY_MAX < puc_day || puc_day < M_DAY_MIN)		-	○	-
	else		○	-	○
	if(M_HOUR_MAX < puc_hour)		-	○	-
	else		○	-	○
	if(M_MINUTE_MAX < puc_minute)		-	○	-
	else		○	-	○
	if(M_SEC_MAX < puc_sec)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_year	0~100	0	100	99
	puc_month	1~13	1	13	12
	puc_day	1~32	1	32	31
	puc_hour	0~24	0	24	23
	puc_minute	0~60	0	60	59
	puc_sec	0~60	0	60	59
Output	luc_err	0,1	0	1	0
Judgment			OK	OK	OK

m_Get_Oscillation_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if((luc_err != M_NG) && ((luc_data & M_REG_FLAG_VLF_BIT) == M_REG_FLAG_VLF_BIT))		○	○	-	○
	for(luc_i=0; luc_i<5; luc_i++)		○	○	-	○
	if(luc_err != M_NG)		○	-	-	-
	else		-	○	-	○
	if(luc_err == M_NG ||(luc_data & M_REG_FLAG_VLF_BIT) == 0x00)			○	-	○
	else		○	-	-	-
	if(luc_i == 5 || luc_err == M_NG)			○	-	-
	else		○	-	-	○
	else		-	-	○	
	Symbol	Range				
Input	m_Get_RegisterData1	0~2	1	1	1	2
	m_Get_RegisterData2	0~2	1	0	0	2
	m_Set_RegisterData	0~2	1	0	1	2
Output	luc_err	1,2	1	2	1	2
Judgment			OK	OK	OK	OK

m_Get_RegisterData

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_reg_adr	0x00,0xFF	0x00	0xFF
	pucp_data	0x00,0xFF	0x00	0xFF
	puc_size	0x00,0xFF	0x00	0xFF
Output	luc_err	0	0	0
Judgment			OK	OK

m_Read_DateAndTime

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(luc_err == M_OK)		○	○	-	-
	else		-	-	○	○
	if(luc_err == M_OK)		○	○	-	-
	if(RTCTime_Chk_Time_Error(luc_time_data[0], luc_time_data[1]) == M_OK)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	m_Get_RegisterData(M_REG_ADDRESS_SEC, (U_CHR*)&luc_reg_data[0][0], 7)	0~2	0	0	1	2
	RTCTime_Chk_Time_Error(luc_time_data[0], luc_time_data[1])	0~2	0	1	2	2
Output	luc_err	0~2	0	2	1	1
Judgment			OK	OK	OK	OK

m_Rtc_Writ_VerifyCheck

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0 ; luc_i < 2 ; luc_i++)		○	○	○
	if(m_Writ_DateAndTime() == M_NG)		○	-	-
	else		-	○	○
	if(m_Read_DateAndTime() == M_NG)		○	-	-
	else		-	○	○
	if(puc_year != uc_year)		○	-	-
	else		-	○	○
	if(puc_month != uc_month)		○	○	-
	else		-	-	○
	if(puc_day != uc_day)		○	○	-
	else		-	-	○
	if(puc_hour != uc_hour)		○	○	-
	else		-	-	○
	if(puc_minute != uc_minute)		○	○	-
	else		-	-	○
	if(luc_rtc_err == 0)		-	○	○
	else		○	-	-
	if(luc_rtc_err != 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_year	0~100	99	0	100
	puc_month	1~13	12	1	13
	puc_day	1~32	31	1	32
	puc_hour	0~24	23	0	24
	puc_minute	0~60	59	0	60
	puc_sec	0~60	59	0	60
	uc_year	0~100	0	0	100
	uc_month	1~13	1	12	13
	uc_day	1~32	1	31	32
	uc_hour	0~24	0	23	24
	uc_minute	0~60	0	59	60
	uc_sec	0~60	0	59	60
	m_Writ_DateAndTime()	0~2	1	0	2
	m_Read_DateAndTime()	0~2	1	0	2
Output	uc_year	0,100	0	0	100
	uc_month	1,13	1	1	13
	uc_day	1,32	1	1	32
	uc_hour	0,24	0	0	24
	uc_minute	0,60	0	0	60
	uc_sec	0,60	0	0	60
	luc_err	0,1	1	0	0
Judgment			OK	OK	OK

m_Set_RegisterData

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	puc_reg_adr	0x00,0xFF	0x00	0xFF
	pucp_data	0x00,0xFF	0x00	0xFF
	puc_size	0x00,0xFF	0x00	0xFF
Output	luc_err	0	0	0
Judgment			OK	OK

m_Writ_CircuitSetting

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	luc_err	0	0
Judgment			OK

m_Writ_DateAndTime

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(m_Set_RegisterData(M_REG_ADDRESS_CONTROL0, (U_CHR*)&luc_data[0], 1) == M_NG)		-	○	○
	else		○	-	-
	if(m_Set_RegisterData(M_REG_ADDRESS_SEC, (U_CHR*)&luc_data[0], 7) == M_NG)		○	-	-
	else		-	○	○
	if(m_Set_RegisterData(M_REG_ADDRESS_CONTROL0, (U_CHR*)&luc_data[0], 1) == M_NG)		○	-	-
	else		-	○	○
	if(luc_rtc_err != 0)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	m_Set_RegisterData(M_REG_ADDRESS_CONTROL0, (U_CHR*)&luc_data[0], 1)	0~2	1	0	2
	m_Set_RegisterData(M_REG_ADDRESS_SEC, (U_CHR*)&luc_data[0], 7)	0~2	1	0	2
	uc_sec	0~60	0	59	60
	uc_minute	0~60	0	59	60
	uc_hour	0~24	0	23	24
	uc_day	1~32	1	31	32
	uc_month	1~13	1	12	13
	uc_year	0~100	0	99	100
Output	luc_err	0,1	1	0	0
Judgment			OK	OK	OK

RTC_DateAndTime_Config

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_err == M_OK)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_year	0~100	0	99	100
	puc_month	0~13	0	12	13
	puc_day	0~32	0	31	32
	puc_hour	0~24	0	23	24
	puc_minute	0~60	0	59	60
	puc_sec	0~60	0	59	60
	m_Rtc_Writ_VerifyCheck(puc_year, puc_month, puc_day, puc_hour, puc_minute, puc_sec)	0~2	0	1	2
	m_data_check(puc_year, puc_month, puc_day, puc_hour, puc_minute, puc_sec)	0~2	0	1	2
Output	luc_err	0~2	0	1	2
Judgment			OK	OK	OK

RTC_DateAndTime_Set_OnOffFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_onoff	0~2	0	1	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

RTC_Get_Data

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pucp_data	0x00,0xFF	0x00	0xFF	0xFF
	uc_year	0~100	0	99	100
	uc_month	0~13	0	12	13
	uc_day	0~32	0	31	32
	uc_hour	0~24	0	23	24
	uc_minute	0~60	0	59	60
	uc_sec	0~60	0	59	60
Output	None	-	-	-	-
Judgment			OK	OK	OK

RTC_Read_DateAndTime

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	
Output	luc_err	0	0
Judgment			OK

RTC_Start_Up

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(luc_flg == M_YES)		○	-	-	-	-
	else		-	○	○	○	○
	if(luc_flg != M_NG)		○	○	○	-	○
	else		-	-	-	○	-
	if(luc_flg != M_NG)		○	-	-	-	○
	else		-	○	○	○	-
	if(luc_flg == M_OK)		○	○	○	-	-
	if(luc_flg == M_OK)		○	○	-	-	-
	if(luc_first_flg == M_ON)		○	-	-	-	-
	else		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	○	○	○
	Symbol	Range					
Input	m_Get_Oscillation_Stop()	0~2	2	0	0	1	3
	m_Rtc_Writ_VerifyCheck(cuc_def_data[0], cuc_def_data[1], cuc_def_data[2], cuc_def_data[3], cuc_def_data[4], cuc_def_data[5])	0,1	0	1	1	1	2
	m_Read_DateAndTime()	0,1	0	0	1	1	2
	m_Writ_CircuitSetting()	0,1	0	0	0	1	2
	m_data_check(uc_year, uc_month, uc_day, uc_hour, uc_minute, uc_sec)	0,1	0	0	1	1	2
Output	luc_flg	0~2	2	0	1	1	2
Judgment			OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	rtc_time.c

m_Chg_Time

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_bc[0]	0~60	0	59	60
	puc_bc[1]	0~60	0	59	60
	puc_bc[2]	0~24	0	23	24
Output	(lul_sec + lul_mins + lul_hours)	0~90060	0	86399	90060
Judgment			OK	OK	OK

m_Chg_Time_Value

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_bc[0]	0~60	0	59	60
	puc_bc[1]	0~60	0	59	60
	puc_bc[2]	0~24	0	23	24
	puc_bc[3]	1~32	1	31	32
	puc_bc[4]	1~13	1	12	13
	puc_bc[5]	0~100	0	99	100
Output	RTCTime_Chg_Day(puc_bc) * (U_LNG)86400 + m_Chg_Time(puc_bc)	0	0	0	0
Judgment			OK	OK	OK

RTCTime_Chg_Day

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_bc0 > 99)		○	-	-
	else			○	○
	if((luc_bc1 <= 0) || (luc_bc1 > 12))		○	-	○
	else		-	○	○
	f((luc_bc2 <= 0) || (luc_bc2 > 31))		○	-	-
	else		-	○	-
	for (luc_i=1; luc_i<= (luc_bc1 - 1) ;luc_i++)		-	○	-
	switch (luc_i)		-	○	-
	case 1		-	○	-
	case 3		-	○	-
	case 5		-	○	-
	case 7		-	○	-
	case 8		-	○	-
	case 10		-	○	-
	case 12		-	○	-
	case 2		-	○	-
	default		-	○	-
	if(luc_bc0 > 0)		-	○	-
	for(luc_i=1; luc_i<= (luc_bc0 - 1) ;luc_i++)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	puc_bc[3]	0~32	0	31	32
	puc_bc[4]	0~13	0	12	13
	puc_bc[5]	0~100	0	99	100
Output	lul_ans_day	0,36134	0	36134	0
Judgment			OK	OK	OK

RTCTime_Chk_Day

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	switch (puc_month)		○	○	○	○	○	○	○	○	○
	case 1		○	-	-	-	-	-	-	-	-
	case 3		-	○	-	-	-	-	-	-	-
	case 5		-	-	○	-	-	-	-	-	-
	case 7		-	-	-	○	-	-	-	-	-
	case 8		-	-	-	-	○	-	-	-	-
	case 10		-	-	-	-	-	○	-	-	-
	case 12		-	-	-	-	-	-	○	-	-
	case 2		-	-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	-	○
	Symbol	Range									
Input	puc_month	1~13	1	3	5	7	8	10	12	2	13
	pus_year	0~2099	0	2098	2099	0	0	0	0	0	0
Output	luc_ans_day	28~31	31	31	31	31	31	31	31	28	30
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

RTCTime_Chk_LeapYear

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if((pus_year % 4) == 0x00)		-	○	○	○	-
	if((pus_year % 100) == 0x00)		-	○	○	-	-
	if((pus_year % 400) == 0x00)		-	○	-	-	-
	else		-	-	○	-	-
	else		-	-	-	○	-
	else		○	-	-	-	○
	Symbol	Range					
Input	pus_year	0~2099	2098	0	100	4	2099
Output	luc_ans	0,1	0	1	0	1	0
Judgment			OK	OK	OK	OK	OK

RTCTime_Chk_Time_Error

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lul_time[1] >= lul_time[0])		○	○	-
	if((lul_time[1] - lul_time[0]) < 60)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	puc_bc1[6]	0x00,0xFF	0x00	0x00	0xFF
	puc_bc2[6]	0x00,0xFF	0x00	0xFF	0x00
Output	luc_err	0,1	0	1	1
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	sci5_uart.c

SCI5UartMain_Comu_Op

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(uc_rxd_chk_flg == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_m_rxd_chk_flg	-	-	-	-
Output	None	-	-	-	-
Judgment			OK	OK	OK

UartSci5_Get_Rxd_Data

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(us_m_rxd_cnt >= lus_buf_limit)		○	-	-
	else		-	○	○
	if(luc_rxd_one_data == M_C_CR)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_m_one_char_receive[0]	0x00~0xFF	0x0D	0x00	0xFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

UartSci5_Get_TransmitEnable

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	SCI5.SCR.BIT.TE	0	0
Judgment			OK

UartSci5_Init_Setting

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_Txd_Busy

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_rxd_chk_flg	0	0
	us_m_rxd_cnt	0	0
Judgment			OK

UartSci5_Int_Sci5_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

UartSci5_Start_Driver

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

UartSci5_Int_Sci5_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

UartSci5_Interrupt_Sci5Error

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

UartSci5_Interrupt_Sci5SendEnd

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

UartSci5_Set_Rxd_Next_Buf

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	R_SCI5_Serial_Receive	0	0
Output	None	-	-
Judgment			OK

UartSci5_Send_Data

	Responsible	Last update
Coding	Ikarashi	2021/7/2
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case2
Root	if(fuss_uart5_tx_count == 0U)		○	○	-
	if(pus_size >= F_SEND_MIN_SIZE)		-	○	-
	else		○	-	-
	else		-	-	○
	Symbol	Range			
Input	pncp_data	0x00,0xFF	0x00	0x00	0xFF
	fuss_uart5_tx_count	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	pus_size	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	sci6_iic.c

m_set_send_receive_delay

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	while(uc_m_send_receive_end_flg == M_ON)		○	○	-	-
	if(us_m_end_timer == 0)		○	-	-	-
	else		-	○	-	-
	Symbol	Range				
Input	uc_m_send_receive_end_flg	0~2	1	1	0	2
	us_m_end_timer	0~21	0	20	21	21
Output	luc_err	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

m_set_send_receive_flg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_send_receive_end_flg	1	1
	us_m_end_timer	2	2
Judgment			OK

m_set_send_receive_init_flg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_send_receive_end_flg	1	1
	us_m_end_timer	20	20
Judgment			OK

SCI6IIC_count_delay_time

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(us_m_end_timer)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	us_m_end_timer	0~21	20	0	21
Output	None	-	-	-	-
Judgment			OK	OK	OK

SCI6IIC_Int_Sci6_Receive

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(luc_err != M_NG)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	m_set_send_receive_delay()	0~2	1	0	2
	luc_rx_add	0x00,0xFF	0x00	0xFF	0x00
	luc_tx_add	0x00,0xFF	0x00	0xFF	0x00
	pucc_tx_buf	0x00,0xFF	0x00	0xFF	0x00
	pucc_rx_buf	0x00,0xFF	0x00	0xFF	0x00
	lus_tx_num	0x00,0xFF	0x00	0xFF	0x00
	lus_rx_num	0x00,0xFF	0x00	0xFF	0x00
Output	luc_err	0~2	1	0	2
Judgment			OK	OK	OK

SCI6IIC_Int_Sci6_Send

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_send_lcd_end_flg == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_m_send_lcd_end_flg	0~2	0	1	2
	luc_address	0x00,0xFF	0x00	0x00	0xFF
	pucc_tx_buf	0x00,0xFF	0x00	0x00	0xFF
	lus_num	0~4324	0	4323	4324
	m_set_send_receive_delay()	0~2	0	1	2
Output	luc_err	0,1	0	1	1
Judgment			OK	OK	OK

SCI6IIC_Int_Sci6_Send_Init_LCD

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_m_send_lcd_end_flg == M_OFF)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_m_send_lcd_end_flg	0~2	0	1	2
	luc_address	0x00,0xFF	0x00	0x00	0xFF
	pucc_tx_buf	0x00,0xFF	0x00	0x00	0xFF
	lus_num	0~4324	0	4323	4324
	m_set_send_receive_delay()	0~2	0	1	2
Output	luc_err	0,1	0	1	1
Judgment			OK	OK	OK

SCI6IIC_Int_Sci6_Send_LCD

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	luc_address	0x00,0xFF	0x00	0x00	0xFF
	pucc_tx_buf	0x00,0xFF	0x00	0x00	0xFF
	lus_num	0~4324	0	4323	4324
Output	luc_err	0	0	0	0
	uc_m_send_lcd_end_flg	1	1	1	1
Judgment			OK	OK	OK

SCI6IIC_Int_Sci6_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI6IIC_Int_Sci6_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI6IIC_set_send_receive_end_flg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_send_receive_end_flg	0	0
	uc_m_send_lcd_end_flg	0	0
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	sci9_uart_command.c

m_SCI9_Code_DU

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'W' || *(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	○	-
	if(*(pucp_rxd + M_PC_DATA_POS) != C_NAK)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x52~0xFF	0x57	0x52	0xFF
	*(pucp_rxd + M_PC_DATA_POS)	0x00~0xFF	0x15	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0xFF
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_SCI9_Code_HA

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'W' || *(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	○	○	-	-
	if(*(pucp_rxd + M_PC_DATA_POS) != C_NAK)		○	-	○	-	-
	if(stp_sub_ad->uc_nc_count != lucs_nu_count_bak)		○	-	-	-	-
	else		-	-	○	-	-
	for(luc_i=0; luc_i<11; luc_i++)		○	-	○	-	-
	if(stp_sub_ad->uc_status_data != 0x00)		○	-	-	-	-
	else		-	-	○	-	-
	else		-	○	-	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x57	0x52	0x57	0xFF	0x00
	*(pucp_rxd + M_PC_DATA_POS)	0x00~0xFF	0x00	0x15	0x00	0xFF	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0xFF	0xFF
	stp_sub_ad->uc_nc_count	0x00,0xFF	0xFF	0x00	0x00	0xFF	0xFF
	stp_sub_ad->uc_status_data	0x00~0xFF	0xFF	0xFF	0x00	0xFF	0xFF
	stp_sub_ad->fl_nc_ad_data[1][0][0]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[1][0][1]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[1][1][0]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[1][1][1]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[0][0][0]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[0][0][1]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[0][1][0]	-800~801	-800	800	0	0	801
	stp_sub_ad->fl_nc_ad_data[0][1][1]	-800~801	-800	800	0	0	801
Output	stp_sub_ad->fl_nc_ad_data[0][0][0]	-107374176~801	-107374176	800	-107374176	0	801
	stp_sub_ad->fl_nc_ad_data[0][0][1]	-107374176~801	-107374176	800	-107374176	0	801
	stp_sub_ad->fl_nc_ad_data[0][1][0]	-107374176~801	-107374176	800	-107374176	0	801
	stp_sub_ad->fl_nc_ad_data[0][1][1]	-107374176~801	-107374176	800	-107374176	0	801
	stp_sub_ad->fl_ec_ad_data[0]	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
	stp_sub_ad->fl_ec_ad_data[1]	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
	stp_sub_ad->fl_ec_ad_data[2]	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
	stp_sub_ad->fl_temp_ad_data	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
	stp_sub_ad->fl_ec_chk_data[0]	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
	stp_sub_ad->fl_ec_chk_data[1]	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
	stp_sub_ad->fl_ec_chk_data[2]	-107374176	-107374176	-107374176	-107374176	-107374176	-107374176
Judgment			OK	OK	OK	OK	OK

m_SCI9_Code_RC

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	○	-	-
	if(Tool_Comp_Pnt((pucp_rxd + M_PC_DATA_POS), (U_CHR *)&lucc_chk_asc, sizeof(lucc_chk_asc-1)) != 0)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x52	0x52	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0xFF
	Tool_Comp_Pnt((pucp_rxd + M_PC_DATA_POS), (U_CHR *)&lucc_chk_asc, sizeof(lucc_chk_asc-1))	0~2	1	0	2	2
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_SCI9_Code_VP

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	○	-	-
	if(*(pucp_rxd + M_PC_DATA_POS) != C_NAK)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x52	0x52	0x00	0xFF
	*(pucp_rxd + M_PC_DATA_POS)	0x00~0xFF	0x15	0x00	0xFF	0x00
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

m_SCI9_Code_VS

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'R')		○	○	○	-	-	-	-	-	-	-
	if(*(pucp_rxd + M_PC_DATA_POS) != C_NAK)		○	○	-	-	-	-	-	-	-	-
	if(RL78I1ESumChk_Get_PhaseFlg() == RL78i1eSumChk_SUM_PHASE_OFF)		○	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-
	else if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'W')		-	-	-	○	○	-	-	-	-	-
	if(*(pucp_rxd + M_PC_DATA_POS) != C_NAK)		-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-
	else if(*(pucp_rxd + M_PC_SUB_COM_POS) == 'C')		-	-	-	-	-	○	○	○	○	-
	if(*(pucp_rxd + M_PC_DATA_POS) != C_NAK)		-	-	-	-	-	○	○	○	-	-
	if(*(pucp_rxd + M_PC_DATA_POS) == '0')		-	-	-	-	-	○	○	-	-	-
	if(RL78I1ESumChk_Get_PhaseFlg() == RL78i1eSumChk_SUM_PHASE_OFF)		-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	○
	Symbol	Range										
Input	*(pucp_rxd + M_PC_SUB_COM_POS)	0x00~0xFF	0x52	0x52	0x52	0x57	0x57	0x43	0x43	0x43	0x43	0xFF
	*(pucp_rxd + M_PC_DATA_POS)	0x00~0xFF	0x00	0x00	0x15	0xFF	0x15	0x30	0x30	0x01	0x15	0xFF
	pucp_txd	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	RL78I1ESumChk_Get_PhaseFlg()	0x00~0xFF	0x00	0xFF	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0xFF
Output	None	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_Set_Start_Command

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	*(pucp_main + 0)	0x00,0xFF	0x00	0xFF
	*(pucp_main + 1)	0x00,0xFF	0x00	0xFF
	*(puc_sub + 0)	0x00,0xFF	0x00	0xFF
Output	*(pucp_txd + 0)	0x02	0x02	0x02
	*(pucp_txd + 1)	0x30	0x30	0x30
	*(pucp_txd + 2)	0x30	0x30	0x30
	*(pucp_txd + 3)	0x30	0x30	0x30
	*(pucp_txd + 4)	0x30	0x30	0x30
	*(pucp_txd + 5)	0x00,0xFF	0x00	0xFF
	*(pucp_txd + 6)	0x00,0xFF	0x00	0xFF
	*(pucp_txd + 7)	0x2C	0x2C	0x2C
	*(pucp_txd + 8)	0x00,0xFF	0x00	0xFF
	*(pucp_txd + 9)	0x2C	0x2C	0x2C
Judgment			OK	OK

SCI9UartCommand_Command_Op

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(puc_com_mode)		○	○	○	○	○	○
	case Cmd_RC		○	-	-	-	-	-
	case Cmd_VP		-	○	-	-	-	-
	case Cmd_VS		-	-	○	-	-	-
	case Cmd_DU		-	-	-	○	-	-
	case Cmd_HA		-	-	-	-	○	-
	default		-	-	-	-	-	○
	Symbol	Range						
Input	puc_com_mode	1~7	1	3	4	5	6	7
	pucp_rxd	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
	pucp_txd	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

SCI9UartCommand_Get_Cmd_NUM

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	Cmd_NUM	8	8
Judgment			OK

SCI9UartCommand_Get_Comnd_Data

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	luc_cmd_num	0~22	0	21	22
Output	uc_sci9_comnd_tbl[luc_cmd_num].uc_cmd_code	1~103	1	103	73
Judgment			OK	OK	OK

SCI9UartCommand_Get_Comnd_Pointer

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	luc_cmd_num	0~8	0	8	9
Output	(U_CHR*)&uc_sci9_comnd_tbl [luc_cmd_num].uc_cmd_chr[0])	-	Indefinite value	Indefinite value	Indefinite value
Judgment			OK	OK	OK

SCI9UartCommand_Send_DUW

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartCommand_Send_HAW

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if((GasCalcEC_Get_ECChkStartFlg(0) == 0) && (GasCalcEC_Get_ECChkStartFlg(1) == 0))		○	-	-	-	-
	else if((GasCalcEC_Get_ECChkStartFlg(0) == 1) && (GasCalcEC_Get_ECChkStartFlg(1) == 0))		-	○	-	-	-
	else if((GasCalcEC_Get_ECChkStartFlg(0) == 0) && (GasCalcEC_Get_ECChkStartFlg(1) == 1))		-	-	○	-	-
	else if((GasCalcEC_Get_ECChkStartFlg(0) == 1) && (GasCalcEC_Get_ECChkStartFlg(1) == 1))		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	GasCalcEC_Get_ECChkStartFlg(0)	0,1	0	1	0	1	2
	GasCalcEC_Get_ECChkStartFlg(1)	0,1	0	0	1	1	2
Output	None	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK

SCI9UartCommand_Send_RCR

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartCommand_Send_VPR

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartCommand_Send_VSC

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartCommand_Send_VSR

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartCommand_Send_VSW

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	sci9_uart_main.c

m_Comand_Chk

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(m_Rxd_Sum_Chk() == M_OK)		○	○	-	-
	while(luc_cmd_cnt <= SCI9UartCommand_Get_Cmd_NUM())		○	○	-	-
	if(Tool_Comp_Pnt((U_CHR *)&uc_m_rxd_buf[M_PC_COMMAND_POS], (U_CHR *)SCI9UartCommand_Get_Comnd_Pointer(luc_cmd_cnt), 2) == 0)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	m_Rxd_Sum_Chk()	0~2	0	0	1	2
	SCI9UartCommand_Get_Cmd_NUM()	8,9	8	8	8	9
	Tool_Comp_Pnt((U_CHR*)&uc_m_rxd_buf [M_PC_COMMAND_POS], (U_CHR*)SCI9UartCommand_ Get_Comnd_Pointer(luc_cmd_cnt), 2)	0~2	0	1	0	2
Output	luc_com_mode	0x00,0xFF	0x00	0xFF	0xFF	0xFF
Judgment			OK	OK	OK	OK

m_Comu_Op_Sub

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(luc_com_mode != 0xff)		○	-
	else		-	○
	Symbol	Range		
Input	m_Comand_Chk()	0x00,0xFF	0x00	0xFF
Output	None	-	-	-
Judgment			OK	OK

m_Rxd_Sum_Chk

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2	case3	case4
Root	for(lus_i = 0; lus_i < sizeof(uc_m_rxd_buf); lus_i++)		○	○	○	○
	if(*lucp_pnt == M_C_STX)		○	-	-	-
	else		-	○	○	○
	if(luc_stx_flg == M_ON)		○	-	-	-
	else		-	-	○	○
	if(*lucp_pnt == M_C_ETX)		-	○	-	-
	else		○	-	○	○
	if(!(luc_sum_1_dat == luc_sum_ck_dat))		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	uc_m_rxd_buf[0]	0x00~0xFF	0x02	0x03	0x00	0xFF
Output	luc_ans	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

m_Txd_Busy

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	
Output	us_m_rxd_cnt	0	0
	uc_rxd_chk_flg	0	0
Judgment			OK

m_Txd_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pus_txd_size	14~22	14	21	22
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_Txd_Sum_Calc

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3
Root	for(lus_i = 0; lus_i < (lus_txd_count); lus_i++)		-	○	○
	Symbol	Range			
Input	lus_txd_count	0~19	0	18	19
Output	luc_sum_1_dat	0	0	0	0
Judgment			OK	OK	OK

SCI9UartMain_Comu_Op

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_rxd_chk_flg == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_rxd_chk_flg	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

SCI9UartMain_Err_Op

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_txd_buf[M_PC_DATA_POS]	0x15	0x15
Judgment			OK

SCI9UartMain_Get_Rxd_Data

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(us_m_rxd_cnt >= lus_buf_limit)		○	-	○
	else		-	○	-
	if((luc_rxd_one_data == M_C_CR) || (luc_rxd_one_data == M_C_EOT) || (luc_rxd_one_data == M_C_ETB))		○	-	-
	else		-	○	○
	Symbol	Range			
Input	uc_m_one_char_receive[0]	0x00~0xFF	0x0D	0x00	0xFF
	us_m_rxd_cnt	0~301	300	0	301
Output	uc_m_rxd_buf[us_m_rxd_cnt]	0	0	0	0
	uc_rxd_chk_flg	1	1	1	1
Judgment			OK	OK	OK

SCI9UartMain_Head_Set

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	for(luc_i = 0; luc_i < M_PC_DATA_POS; ++luc_i)		○	○
	Symbol	Range		
Input	uc_m_rxd_buf[luc_i]	0x00,0xFF	0x00	0xFF
Output	uc_m_txd_buf[luc_i]	0x00,0xFF	0x00	0xFF
Judgment			OK	OK

SCI9UartMain_Int_Sci9_Start

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UARTMain_Int_Sci9_Stop

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartMain_Set_Rxd_Next_Buf

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

SCI9UartMain_tx_nbyte

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	pus_num	0~8	0	7	8
Output	uc_m_txd_buf[M_PC_DATA_POS + pus_num]	3	3	3	3
	uc_m_txd_buf[13 + pus_num]	4	4	4	4
Judgment			OK	OK	OK

SCI9UartMain_Txd_Buf_Pointre

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		
	Symbol	Range	
Input	None	-	-
Output	uc_m_txd_buf[0]	-	Indefinite value
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	self_sub_RL78FProg.c

m_cmdBaudRateSet

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-
	else		-	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-
	else		-	-	○	○
	if(!m_isRxACK(0))		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	m_reqRx_DataFram(lus_data_bytes)	-1,0	-1	0	0	1
	m_waitForRxComplete(TIM_CS6)	-2~0	0	-2	0	1
	m_isRxACK	0~2	1	1	0	2
Output	m_cmdBaudRateSet@@	-2~0	-1	-2	-1	0
Judgment			OK	OK	OK	OK

m_cmdBlockBlankCheck

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-	-
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-	-
	if(lsc_state == COM_S_ERROR)		-	-	○	-	-
	if(!m_isRxACK(0))		-	-	-	○	-
	Symbol	Range					
Input	pul_start_addr	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	pul_end_addr	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	m_reqRx_DataFram(lus_data_bytes)	0~255	255	0	0	0	1
	m_waitForRxComplete(TIM_CS4)	0~255	0	254	255	0	1
	m_isRxACK(0)	0~2	1	1	1	0	2
Output	m_cmdBlockBlankCheck@@	-2~1	-1	-2	-1	0	1
Judgment			OK	OK	OK	OK	OK

m_cmdBlockErase

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-
	else		-	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-
	else		-	-	○	○
	if(!m_isRxACK(0)		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	pul_elase_addr	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	m_reqRx_DataFram(lus_data_bytes)	-1~1	-1	0	0	1
	m_waitForRxComplete(TIM_CS3)	-2~1	0	-2	0	1
	m_isRxACK(0)	0~2	1	1	0	2
Output	m_cmdBlockErase@@	-2~0	-1	-2	-1	0
Judgment			OK	OK	OK	OK

m_cmdCheckSum

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-	-	-	-
	else		-	○	○	○	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-	-	-	-
	else		-	-	○	○	○	○	○
	if(!m_isRxACK(0))		-	-	○	-	-	-	-
	else		-	-	-	○	○	○	○
	if(lsc_state == COM_S_ERROR)		-	-	-	○	-	-	-
	else		-	-	-	-	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	-	-	-	○	-	-
	else		-	-	-	-	-	○	○
	if(lus_sum != pus_sum)		-	-	-	-	-	○	-
	else		-	-	-	-	-	-	○
	Symbol	Range							
Input	pul_block_head_addr	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	pus_sum	0x0000~0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000	0x0000
	m_reqRx_DataFram(lus_data_bytes1)	-1,0	-1	0	0	0	0	0	1
	m_reqRx_DataFram(lus_data_bytes2)	-1,0	0	0	0	-1	0	0	1
	m_waitForRxComplete(TIM_CS10)	-2~0	0	-2	0	0	0	0	1
	m_waitForRxComplete(TIM_SD10)	-2~0	0	0	0	0	-2	0	1
	m_isRxACK(0)	0,1	1	1	0	1	1	1	2
	uc_rx_buff[FSTATE_POS_DATA]	0x00~0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x00
	uc_rx_buff[FSTATE_POS_DATA + 1]	0x00~0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x00
Output	m_cmdCheckSum@@	-2~1	-1	-2	-1	1	1	0	1
Judgment			OK	OK	OK	OK	OK	OK	OK

m_cmdProgramming_step1

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-
	else		-	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-
	else		-	-	○	○
	if(!m_isRxACK(0))		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	pul_block_head_addr	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	m_reqRx_DataFram(lus_data_bytes)	-1~1	-1	0	0	1
	m_waitForRxComplete(TIM_CS5)	-2~1	-2	-2	0	1
	m_isRxACK(0)	0~2	1	1	0	2
Output	m_cmdProgramming_step1@@	-2~0	-1	-2	-1	0
Judgment			OK	OK	OK	OK

m_cmdProgramming_step2

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(pus_data_size > 256)		○	-	-	-	-	-
	else		-	○	○	○	○	-
	if(pus_data_size == 256)			○	○	○	○	○
	else		○	-	-	-	-	-
	if(lsc_state == COM_S_ERROR)		-	○	-	-	-	-
	else		-	-	○	○	○	-
	if(lsc_state == COM_S_TIMEOUT)		-	-	○	-	-	-
	else		-	-	-	○	○	-
	if(!m_isRxACK(0))		-	-	-	○	-	-
	else		-	-	-	-	○	○
	if(!m_isRxACK(1))		-	-	-	-	○	-
	else		-	-	-	-	-	○
	Symbol	Range						
Input	pus_data_size	0x0000~0xFFFF	0xFFFF	0x0100	0x0100	0x0100	0x0100	0x0000
	lucp_data	0x00,0xFF	0xFF	0x00	0x00	0x00	0x00	0x00
	puc_is_last_frame	0x00,0xFF	0xFF	0x00	0x00	0x00	0x00	0x00
	m_reqRx_DataFram(lus_data_bytes)	-1,0	1	-1	0	0	0	0
	m_waitForRxComplete(TIM_DS5)	-2~1	1	0	-2	0	0	0
	m_isRxACK(0)	0~2	2	1	1	0	1	1
	m_isRxACK(1)	0~2	2	1	1	1	0	1
Output	m_cmdProgramming_step2@@	-2~0	-1	-1	-2	-1	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_cmdProgramming_step3

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-
	else		-	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-
	else		-	-	○	○
	if(!m_isRxACK(0))		-	-	○	-
	else		-	-	-	○
	Symbol	Range				
Input	m_reqRx_DataFram(lus_data_bytes)	-1~1	-1	0	0	1
	m_waitForRxComplete(TIM_SS5)	-2~1	-2	-2	0	1
	m_isRxACK(0)	0~2	1	1	0	2
Output	m_cmdProgramming_step3@@	-2~0	-1	-2	-1	0
Judgment			OK	OK	OK	OK

m_cmdReset

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(lsc_state == COM_S_ERROR)		○	-	-	-	-
	else		-	○	○	○	○
	if(lsc_state == COM_S_TIMEOUT)		-	○	-	-	-
	else		-	-	○	○	○
	if(lsc_state == COM_S_ERROR)		-	-	○	-	-
	else		-	-	-	○	○
	if(!m_isRxACK(0))		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	m_reqRx_DataFram(lus_data_bytes)	-1~1	-1	0	0	0	1
	m_waitForRxComplete(TIM_CS1)	-2~1	0	-2	-1	0	1
	m_isRxACK(0)	0~2	1	1	1	0	2
Output	m_cmdReset@@	-2~0	-1	-2	-1	-1	0
Judgment			OK	OK	OK	OK	OK

m_csum_ui16

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(;pus_bytes;pus_bytes--)		○	-	○
	Symbol	Range			
Input	pus_bytes	0x0000~0xFFFF	0x0001	0x0000	0xFFFF
	pucp_data	0x00,0xFF	0xFF	0x00	0x00
Output	lus_sum	0x0000~0x00FF	0x00FF	0x0000	0x07F8
Judgment			OK	OK	OK

m_csum_ui8

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(;pus_bytes;pus_bytes--)		○	-	○
	Symbol	Range			
Input	pus_bytes	0x0000~0xFFFF	0x0001	0x0000	0xFFFF
	pucp_data	0x00,0xFF	0xFF	0x00	0x00
Output	luc_sum	0x00~0xFF	0xFF	0x00	0xf8
Judgment			OK	OK	OK

m_eraseUntiLastBlock

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(; luc_retry; luc_retry--)		○	○	○
	if(lsc_state < COM_S_SUCCESS)		○	-	-
	else		-	○	○
	if(lsc_state < COM_S_SUCCESS)		-	○	-
	else		-	-	○
	if(pul_address != USER_MAX_ADDRESS+1)		-	-	○
	else		-	-	-
	Symbol	Range			
Input	pul_address	0x00000000,0x00007400	0x00000000	0x00000000	0x00007400
	m_cmdBlockErase(pul_address)	-2~1	-2	0	1
	m_cmdBlockBlankCheck(pul_address,pul_address+MMC_BLOCK_SIZE-1)	-2~1	0	-2	1
Output	m_eraseUntiLastBlock@@	0,1	0	0	1
Judgment			OK	OK	OK

m_isRxACK

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(uc_rx_buff[FSTATE_POS_DATA + puc_data_num] == ASCII_ACK)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_data_num	0x00,0xFF	0x00	0xFF	0x00
	uc_rx_buff[FSTATE_POS_DATA + puc_data_num]	0x00,0xFF	0x06	0x00	0xFF
Output	m_isRxACK@@	0,1	1	0	0
Judgment			OK	OK	OK

m_reqRx

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(1U > rx_num)		○	-	-
	else		-	○	○
	if(uc_use_sci_rx_for_rl78prog == TRUE)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	rx_buf	0x00,0xFF	0xFF	0x00	0x00
	rx_num	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	uc_use_sci_rx_for_rl78prog	0~2	0	1	2
Output	uc_use_sci_rx_for_rl78prog	0,1	0	1	1
	g_sci9_rx_count	0	0	0	0
	g_sci9_rx_length	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	SCI9.SCR.BIT.RIE	0,1	0	0	1
	SCI9.SCR.BIT.RE	0,1	0	0	1
	m_reqRx@@	0,1	0	0	1
Judgment			OK	OK	OK

m_reqRx_DataFram

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(lsc_result == FALSE)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	pus_data_bytes	1~4	1	3	4
	m_reqRx(uc_rx_buff, pus_data_bytes)	-1~2	0	-1	2
Output	us_rx_bytes	5~8	5	7	8
	m_reqRx_DataFram	-1,0	-1	0	0
Judgment			OK	OK	OK

m_reqTx

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(1U > tx_num)		○	-	-
	else		-	○	○
	if(uc_use_sci_tx_for_rl78prog == TRUE)		-	○	-
	else		○	-	○
	Symbol	Range			
Input	tx_buf	0x00,0xFF	0x00	0xFF	0xFF
	tx_num	0x0000~0xFFFF	0x0000	0xFFFF	0xFFFF
	uc_use_sci_tx_for_rl78prog	0~2	0	1	2
Output	uc_use_sci_tx_for_rl78prog	0,1	0	1	1
	g_sci9_tx_count	0x0000,0xFFFF	0x0000	0x0000	0xFFFF
	PORTB.PMR.BYTE	227	227	227	227
	SCI9.SCR.BIT.TIE	0,1	0	0	1
	SCI9.SCR.BIT.TE	0,1	0	0	1
	m_reqTx@@	0,1	0	0	1
Judgment			OK	OK	OK

m_reqTx_CmdFrame

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	while(uc_use_sci_tx_for_rl78prog)		○	-	○
	Symbol	Range			
Input	puc_com	0x00~0xB1	0x00	0xB	0xB1
	pus_info_bytes	0~8	0	7	8
	pucp_info_addr	0x00,0xFF	0xFF	0x00	0x00
	uc_use_sci_tx_for_rl78prog	0~2	1	0	2
Output	uc_tx_buff[0]	1	1	1	1
	uc_tx_buff[1]	1~9	1	8	9
	uc_tx_buff[2]	0~177	0	11	177
Judgment			OK	OK	OK

m_reqTx_DataFrame

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(pus_data_bytes == 0)		○	-	-
	if(puc_is_last_frame == TRUE)		○	-	-
	else		-	○	○
	while(uc_use_sci_tx_for_rl78prog)		○	-	○
	Symbol	Range			
Input	pus_data_bytes	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	pucp_data_addr	0x00,0xFF	0x00	0xFF	0xFF
	puc_is_last_frame	0~2	1	0	2
	uc_use_sci_tx_for_rl78prog	0~2	1	0	2
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_restartComDrv

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	
Output	SCI9.SMR.BYTE	9	9
	SCI9.SEMR.BYTE	128	128
	SCI9.BRR	0	0
Judgment			OK

m_sendComMethodData

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	for(;lul_wait_count;lul_wait_count--)		○
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

m_updateWaitTime

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(pfl_freq_mhz < 16.0)		○	○	○	-
	else		-	-	-	○
	for(lus_i = 0; lus_i < TIM_END_OF_NUM; lus_i++)		○	○	○	○
	if(lul_temp_time[lus_i] == 0)		○	○	○	○
	else		○	○	○	○
	Symbol	Range				
Input	pfl_freq_mhz	0.75~17	0.75	1	2	17
Output	ul_wait_time[0]	14	14	14	14	14
Judgment			OK	OK	OK	OK

m_wait

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case2
Root	do		○	○	○
	while(lul_wait_count--)		○	-	-
	Symbol	Range			
Input	pus_wait_id	0~40	0	39	40
	ul_wait_time[pus_wait_id]	0x00,0xFF	0xFF	0x00	0x00
Output	None	-	-	-	-
Judgment			OK	OK	OK

m_waitForRxComplete

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	do		○	○	○
	if(uc_use_sci_rx_for_rl78prog != TRUE)		○	-	-
	else		-	○	○
	while(lul_wait_count--)		○	○	-
	Symbol	Range			
Input	pus_timeout_id	7~38	7	37	38
	ul_wait_time[pus_timeout_id]	0~546214	0	546213	546214
	us_rx_bytes	0x00,0xFF	0xFF	0xFF	0x00
	uc_use_sci_rx_for_rl78prog	0~2	1	0	2
Output	prev_tx_delay_id	-2,0	-2	0	0
Judgment			OK	OK	OK

RL78FProg_execTargetPowerOff

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	MPC.PB7PFS.BYTE	0	0
	PORTB.PODR.BYTE	0	0
	PORTB.PMR.BYTE	0	0
Judgment			OK

RL78FProg_rewriteBlock

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	9

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9
Root	if(us_split_data_counter == 1)		○	○	-	-	-	-	-	-	-
	if(us_split_data_counter == SPLIT_BLOCK_MAX_NUM)		-	-	-	○	○	○	○	○	○
	if(ul_current_block_top_addr >= USER_MAX_ADDRESS)		-	-	○	-	-	-	-	-	-
	if(luc_is_first_frame == TRUE)		○	○	-	-	-	-	-	-	-
	if(lsc_state < COM_S_SUCCESS)		○	-	-	-	-	-	-	-	-
	if(lsc_state < COM_S_SUCCESS)		-	○	-	-	-	-	-	-	-
	if(luc_is_last_frame == TRUE)		-	-	-	○	○	○	○	○	-
	if(lsc_state < COM_S_SUCCESS)		-	-	-	○	-	-	-	-	-
	if(lsc_state < COM_S_SUCCESS)		-	-	-	-	○	-	-	-	-
	if(lsc_state == COM_R_UNMATCH)		-	-	-	-	-	○	-	-	-
	if(ul_current_block_top_addr > USER_MAX_ADDRESS)		-	-	-	-	-	-	○	-	-
	Symbol	Range									
Input	pucp_write_data	0x00~0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFF
	us_split_data_counter	0~5	1	1	0	4	4	4	4	4	5
	m_cmdProgramming_step1()	0~-1	-1	0	0	0	0	0	0	0	1
	m_cmdProgramming_step2()	0~-1	0	-1	0	0	0	0	0	0	1
	m_cmdProgramming_step3()	0~-1	0	0	0	0	-1	0	1	1	1
	m_cmdCheckSum()	0~-1	0	0	0	0	-1	0	1	1	1
	ul_current_block_top_addr	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00007E70	0x00000000	0x00000000
Output	RL78FProg_rewriteBlock@@	0~2	0	0	2	0	0	0	2	1	1
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK

RL78FProg_setStartBlock

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(lul_block_chk != pul_address)		-	-	○
	else		○	○	-
	if(pul_address > USER_MAX_ADDRESS)		-	○	-
	else		○	-	-
	Symbol	Range			
Input	pul_address	0x00000000~0xFFFFFFFF	0x00000000	0x00008000	0xFFFFFFFF
Output	ul_current_block_top_addr	0x00000000	0x00000000	0x00000000	0x00000000
	us_split_data_counter	1	1	1	1
	RL78FProg_setStartBlock@@	0	0	0	0
Judgment			OK	OK	OK

RL78FProg_startRewritingProg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	for(; luc_retry; luc_retry--)		○	○	○
	if(lsc_state < COM_S_SUCCESS)		○	-	-
	else		-	○	○
	if(lsc_state < COM_S_SUCCESS)		-	○	-
	else		-	-	○
	Symbol	Range			
Input	m_cmdBaudRateSet()	-2~1	-2	0	1
	m_cmdReset()	-2~1	0	-2	1
Output	RL78FProg_startRewritingProg@@	0,1	0	0	1
Judgment			OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	self_sub_prog.c

SelfSubProg_Get_10msecChgFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	-		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_10msec_chg_flg	0	0
Judgment			OK

SelfSubProg_Get_ModeOnFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	-		-
	Symbol	Range	
Input	None	-	-
Output	fbl_mode_on_flg	0	0
Judgment			OK

SelfSubProg_Int_10msec

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(ucs_m_timer)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	ucs_m_timer	0~26	25	0	26
Output	None	-	-	-	-
Judgment			OK	OK	OK

SelfSubProg_Main_RewriteProcessing

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	Yes
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	if(ucs_m_rewrite_op_flg == M_ON)		○	○	○	○	-
	if(LoggerGXForce_Set_PowerOff() != ZIPC_NG)		○	○	○	-	-
	if(FRAM_Get_flg() != M_ON)		○	○	-	-	-
	while(IntervalTimer_Flag_Acquisition_250msec() == M_OFF)		○	○	-	-	-
	while(ucs_m_timer)		○	○	-	-	-
	while(ucs_m_timer)		○	○	-	-	-
	while(ucs_m_rewrite_op_flg == M_ON)		○	○	-	-	-
	if((MAIN_Get_PowerOffFlg() == M_ON) && (ucs_m_End_op_flg == M_OFF))		○	-	-	-	-
	else		-	○	-	-	-
	if(ucs_m_End_op_flg || KeyEvent_Get_KeyEvent() == EV_MODE_3s)		○	-	-	-	-
	else		-	○	-	-	-
	while(1)		○	○	-	-	-
	else		-	–	○	-	-
	else		-	-	-	○	-
	else		-	-	-	-	○
	Symbol	Range					
Input	ucs_m_rewrite_op_flg	0,1	1	1	1	1	0
	LoggerGXForce_Set_PowerOff()	0,1	0	0	0	1	1
	FRAM_Get_flg()	0,1	0	0	1	1	1
	IntervalTimer_Flag_Acquisition_250msec()	0,1	0	0	0	0	0
	MAIN_Get_PowerOffFlg()	0,1	1	0	0	0	0
	ucs_m_End_op_flg	0,1	0	1	0	0	0
	KeyEvent_Get_KeyEvent()	0~5	5	4	0	0	0
Output	ucs_m_10msec_chg_flg	1	1	1	1	1	1
	ucs_m_timer	0,99	99	0	0	0	0
	ucs_m_End_op_flg	0,1	1	0	0	0	0
	ucs_m_irda_chg_flg	1	1	1	1	1	1
Judgment			OK	OK	OK	OK	OK

SelfSubProg_Set_ModeOnFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	-		-
	Symbol	Range	
Input	None	-	-
Output	fbl_mode_on_flg	0	0
Judgment			OK

SelfSubProg_Set_RewiteStart

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	-		-
	Symbol	Range	
Input	None	-	-
Output	ucs_m_rewrite_op_flg	1	1
Judgment			OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	tool.c

Tool_Asc_BinL

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	if((puc_num == 8) && (Tool_Comp_Pnt(&ucc_off_val_data[0], pucp_pnt, 8) == 0))		○	-	-
	else		-	○	○
	if(*pucp_pnt == '+' || *pucp_pnt == '-')		-	○	-
	else		-	-	○
	for(luc_i; luc_i < puc_num; luc_i++)		-	○	○
	for(luc_i=(puc_num-1); luc_i > 0; luc_i--)		-	○	○
	Symbol	Range			
Input	puc_num	0~9	8	0	9
	Tool_Comp_Pnt(&ucc_off_val_data[0], pucp_pnt, 8)	0~2	0	1	2
	*pucp_pnt	0x00~0xFF	0x00	0x2B	0xFF
Output	lul_ans	0x615AB054~0xDDDDDDDD	0xDDDDDDDD	0x00000000	0x635750A2
Judgment			OK	OK	OK

Tool_Asc_Dec

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○	-
	if(('.' == *(pucp_str)) || (',' == *(pucp_str)))		○	-	-	-
	else if(' ' == *(pucp_str))		-	○	-	-
	else		○	○	○	-
	Symbol	Range				
Input	*(pucp_str)	0x00~0xFF	0x2E	0x20	0xFF	0x00
	pucp_col_dot	0x00,0xFF	0x00	0x00	0x00	0xFF
	puc_col	0~9	8	8	9	0
Output	lul_ans	0x00000000~0x5AE835E9	0x0DB58569	0x0DB58569	0x5AE825E9	0x00000000
Judgment			OK	OK	OK	OK

Tool_Asc_Hex

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	for(i = 0; i < puc_num; i++)		○	○	-	○
	for(luc_loop = 0; luc_loop < 2; luc_loop++)		○	○	-	○
	if(('pucp_pnt1 >= '0') && (*pucp_pnt1 <= '9'))		○	-	-	-
	if(luc_loop == 0)		○	-	-	-
	else		○	-	-	-
	else		-	○	-	○
	if((*pucp_pnt1 >= 'A') && (*pucp_pnt1 <= 'F'))		-	○	-	-
	if(luc_loop == 0)		-	○	-	-
	else		-	○	-	-
	else		-	-	-	○
	if(puc_num == 2)		○	-	-	-
	else		-	○	-	○
	if(puc_num == 4)		-	○	-	-
	else		○	-	-	○
	Symbol	Range				
Input	*pucp_pnt1	0x00~0xFF	0x30	0x41	0x00	0xFF
	*pucp_pnt2	0x00,0xFF	0x00	0x00	0xFF	0x00
	puc_num	0~5	2	4	0	5
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

Tool_Bin_Asc

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	2
Judgment	OK

			case1	case2
Root	-		-	-
	Symbol	Range		
Input	pus_num	0x0000,0xFFFF	0x0000	0xFFFF
	ppuc_asc	0x00,0xFF	0x00	0xFF
Output	ppuc_asc[0]	48,54	48	54
	ppuc_asc[1]	48,53	48	53
Judgment			OK	OK

Tool_CHR_Chg_OnOff

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	3
Judgment	OK

			case1	case2	case3
Root	if(puc_pnt == M_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_pnt	0~2	0	1	2
Output	luc_ans	0,1	1	0	1
Judgment			OK	OK	OK

Tool_Comp_Pnt

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification point	4
Judgment	OK

			case1	case2	case3	case4
Root	for(lul_num=0; lul_num<pul_size; lul_num++)		○	○	○	○
	if(*ppuc_add1 != *ppuc_add2)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	ppuc_add1	0x00,0xFF	0x00	0x00	0x00	0xFF
	ppuc_add2	0x00~0xFF	0x00	0x01	0x01	0xFF
	pul_size	0~11	10	10	0	11
Output	luc_ACK	0,1	1	1	0	1
Judgment			OK	OK	OK	OK

Tool_Copy_Pnt

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lul_loop = 0; lul_loop < pul_num ; ++lul_loop)		○	○	○
	Symbol	Range			
Input	ppuc_pnt1	0x00,0xFF	0x00	0xFF	0xFF
	ppuc_pnt2	0x00,0xFF	0x00	0xFF	0xFF
	pul_num	0~257	0	256	257
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_Copy_Reverse_Pnt

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lul_loop = 0; lul_loop < pul_num ; ++lul_loop)		○	○	○
	Symbol	Range			
Input	ppuc_pnt1	0x00,0xFF	0x00	0xFF	0xFF
	ppuc_pnt2	0x00,0xFF	0x00	0xFF	0xFF
	pul_num	0~257	0	256	257
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_Copy_SHORT_CHR

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(lul_loop = 0; lul_loop < pul_num ; ++lul_loop)		○	○	○
	Symbol	Range			
Input	ppus_pnt1	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	ppuc_pnt2	0x00,0xFF	0x00	0xFF	0xFF
	pul_num	0~257	0	256	257
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_Copy_SHORT_Pnt

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3
Root	for(lul_loop = 0; lul_loop < pul_num ; ++lul_loop)		○	○	○
	Symbol	Range			
Input	ppus_pnt1	0x0000,0xFFFF	0x0000	0xFFFF	0xFFFF
	ppuc_pnt2	0x00,0xFF	0x00	0xFF	0xFF
	pul_num	0~257	0	256	257
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_Disp_OnOff_Menu

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(puc_sel_num)		○	○	○
	case M_OFF		○	-	-
	case M_ON		-	○	-
	default		-	-	○
	Symbol	Range			
Input	puc_sel_num	0~2	0	1	2
	puc_disp_num	0~16	0	15	16
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_Dump_Hex_Asc

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	for(luc_i = 0; luc_i < pul_num; luc_i++)		○	○	○	○	○	○	-	-	○	○
	for(luc_loop = 0; luc_loop < 2; luc_loop++)		○	○	○	○	○	○	-	-	○	○
	switch(*pucp_pnt2)		○	○	○	○	○	-	-	-	○	○
	case 10		○	-	-	-	-	-	-	-	-	-
	case 11		-	○	-	-	-	-	-	-	-	-
	case 12		-	-	-	-	○	-	-	-	○	○
	case 13		-	-	○	-	-	-	-	-	-	-
	case 14		-	-	-	-	-	○	-	-	-	-
	case 15		-	-	-	-	-	-	○	-	○	○
	default		○	-	-	○	-	-	-	-	○	○
	Symbol	Range										
Input	pucp_pnt1	0xA0~0xFF	0xA0	0xB0	0xD0	0xFF	0xC0	0xE0	0xF0	0xF0	0xF0	0xF0
	pucp_pnt2	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	pul_num	0~257	256	256	256	256	256	256	256	0	256	257
Output	luc_conv	0,2	0	0	0	0	0	0	0	0	0	2
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Tool_Hex_Asc

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	for(luc_i = 0; luc_i < puc_num; luc_i++)		○	○	○	○	○	○	-	-	○	○
	for(luc_loop = 0; luc_loop < 2; luc_loop++)		○	○	○	○	○	○	-	-	○	○
	switch(*pucp_pnt2)		○	○	○	○	○	-	-	-	○	○
	case 10		○	-	-	-	-	-	-	-	-	-
	case 11		-	○	-	-	-	-	-	-	-	-
	case 12		-	-	-	-	○	-	-	-	○	○
	case 13		-	-	○	-	-	-	-	-	-	-
	case 14		-	-	-	-	-	○	-	-	-	-
	case 15		-	-	-	-	-	-	○	-	○	○
	default		○	-	-	○	-	-	-	-	○	○
	Symbol	Range										
Input	pucp_pnt1	0xA0-0xFF	0xA0	0xB0	0xD0	0xFF	0xC0	0xE0	0xF0	0xF0	0xF0	0xF0
	pucp_pnt2	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	pul_num	0~257	256	256	256	256	256	256	256	0	256	257
Output	luc_conv	0,2	0	0	0	0	0	0	0	0	0	2
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Tool_Hex_ShortAsc

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	for(luc_i = 0; luc_i < puc_num; luc_i++)		○	○	○	○	○	○	-	-	○	○
	for(luc_loop = 0; luc_loop < 2; luc_loop++)		○	○	○	○	○	○	-	-	○	○
	switch(*pusp_pnt2)		○	○	○	○	○	-	-	-	○	○
	case 10		○	-	-	-	-	-	-	-	-	-
	case 11		-	○	-	-	-	-	-	-	-	-
	case 12		-		-	-	○	-	-	-	○	○
	case 13		-	-	○	-	-	-	-	-	-	-
	case 14		-	-	-	-	-	○	-	-	-	-
	case 15		-	-	-	-	-	-	○	-	○	○
	default		○	-	-	○	-	-	-	-	○	○
	Symbol	Range										
Input	pucp_pnt1	0xA0-0xFF	0xA0	0xB0	0xD0	0xFF	0xC0	0xE0	0xF0	0xF0	0xF0	0xF0
	pusp_pnt2	0x0000~0xFFFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xFFFF
	puc_num	0x00~0xFF	0x01	0x01	0x01	0x01	0x01	0x01	0x01	0x00	0x02	0xFF
Output	luc_conv	0~254	2	2	2	2	2	2	2	0	4	254
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Tool_OneCount

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	13
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13
Root	if(puc_off_val == M_ON)		○	○	○	○	○	○	○	-	-	-	-	-	-
	if(puc_flg == 0)		○	○	○	-	-	-	-	-	-	-	-	-	-
	if(pul_set_value == M_OFF_VAL)		○	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	○	-	-	-	-	-	-	-	-	-	-
	if(pul_set_value > pul_max_value)		-	○	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	○	○	○	○	-	-	-	-	-	-
	if(pul_set_value == M_OFF_VAL)		-	-	-	○	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	○	○	○	-	-	-	-	-	-
	if(pul_set_value < pul_digit)		-	-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	○	○	-	-	-	-	-	-
	if((pul_set_value - pul_digit) < pul_min_value)		-	-	-	-	-	○	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	○	○	○	○	○
	if(puc_flg == 0)		-	-	-	-	-	-	-	○	○	-	-	-	-
	if(pul_set_value > pul_max_value)		-	-	-	-	-	-	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	○	○	○	○
	if(pul_set_value < pul_digit)		-	-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	○	○
	if(pul_set_value - pul_digit) < pul_min_value)		-	-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	○	○
	Symbol	Range													
Input	pul_set_value	0x00000000~0xFFFFFFFF	0xDDDDDDDD	0x00000001	0x00000000	0xDDDDDDDD	0x00000000	0x00000001	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000	0x00000001	0xFFFFFFFF
	pul_digit	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000001	0x00000000	0x00000000	0x00000001	0x00000000	0x00000001	0x00000000	0x00000000	0xFFFFFFFF
	pul_max_value	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000001	0x00000000	0x00000000	0xFFFFFFFF
	pul_min_value	0x00000000~0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000002	0x00000000	0x00000000	0x00000000	0x00000000	0x00000001	0x00000000	0xFFFFFFFF
	puc_flg	0~2	0	0	0	1	1	1	1	0	0	0	1	1	2
	puc_off_val	0~2	1	1	1	1	1	1	1	0	0	0	0	0	2
Output	pul_set_value	0x00000000~0xFFFFFFFF	0x00000000	0xDDDDDDDD	0x00000000	0x00000000	0xDDDDDDDD	0xDDDDDDDD	0x00000001	0x00000000	0x00000000	0x00000001	0x00000000	0x00000001	0xFFFFFFFF
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

Tool_set_Dec_ChrAscii

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_i = 0; luc_i < 16; luc_i++)		○	○	○	○	○	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○	○	○	○
	if((luc_i != 0) && (puc_dot == luc_i))		-	-	○	-	-	-
	else		○	○	○	○	○	○
	if(pul_val > 0)		○	-	-	-	-	-
	else		-	○	○	○	○	○
	if((pul_val <= 0) && (luc_i > puc_dot) && (puc_app != 11))		○	-	-	-	-	-
	else		-	-	○	-	-	○
	if((pul_val <= 0) && (puc_dot == 0) && (puc_app != 11))		-	○	-	○	○	-
	else		-	-	○	-	-	○
	if(puc_app != 0)		○	○	○	○	○	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○	-	-	○
	else		-	-	-	○	○	-
	for(luc_i = 0; luc_i < puc_col; luc_i++)		-	-	-	○	○	-
	if(luc_buff[luc_i] != 0x20)		-	-	-	○	-	-
	else		-	-	-	-	○	-
	for(luc_i; luc_i < puc_col; luc_i++)		-	-	-	○	○	-
	Symbol	Range						
Input	pucp_str	0x00,0xFF	0x00	0x00	0x00	0x00	0x00	0xFF
	pul_val	0x00000000~0xFFFFFFFF	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	puc_col	0~9	5	5	8	8	0	9
	puc_dot	0~4	3	0	1	0	0	4
	puc_app	0~12	1	1	11	0	0	12
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

Tool_set_Dec_ChrSymbolAscii

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(pul_val == M_OFF_VAL)		○	-	-	-
	else		-	○	○	○
	Symbol	Range				
Input	pucp_str	0x00,0xFF	0x00	0xFF	0x00	0xFF
	pul_val	0x00000000~0xDDDDDDDD	0xDDDDDDDD	0xFFFFFFFF	0x00000000	0xFFFFFFFF
	puc_col	0~9	0	8	8	9
	puc_dot	0~4	0	3	3	4
	puc_app	0~12	0	1	11	12
Output	None	-	-	-	-	-
Judgment			OK	OK	OK	OK

Tool_set_Dec_DenstAscii

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0; luc_i < 16; luc_i++)		○	○	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○
	if(pul_val > 0)		○	-	○
	else		-	○	-
	for(luc_i=(puc_col-(puc_dot+1))		○	○	○
	if(luc_buff[luc_i] == ' ')		-	○	-
	else		○	-	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○
	Symbol	Range			
Input	pucp_str	0x00~0xFF	0x00	0x00	0xFF
	pul_val	0x00000000~0xFFFFFFFF	0x00000001	0x00000000	0xFFFFFFFF
	puc_col	0~9	0	8	9
	puc_dot	0~4	0	3	4
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_set_Dec_DenstShortAscii

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i = 0; luc_i < 16; luc_i++)		○	○	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○
	if(pul_val > 0)		○	-	○
	else		-	○	-
	for(luc_i=(puc_col-(puc_dot+1)); luc_i < puc_col; luc_i++)		○	○	○
	if(luc_buff[luc_i] == ' ')		-	○	-
	else		○	-	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○
	Symbol	Range			
Input	pusp_str	0x00~0xFF	0x00	0x00	0xFF
	pul_val	0x00000000~0xFFFFFFFF	0x00000001	0x00000000	0xFFFFFFFF
	puc_col	2~9	2	8	9
	puc_dot	1~4	1	3	4
Output	None	-	-	-	-
Judgment			OK	OK	OK

Tool_set_Dec_ShortAscii

	Responsible	Last update
Coding	Ikarashi	2021/7/5
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	for(luc_i = 0; luc_i < 16; luc_i++)		○	○	○	○	○	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		○	○	○	○	-	○
	if((luc_i != 0) && (puc_dot == luc_i))		-	-	○	-	-	-
	else		○	○	○	○	-	○
	if(pul_val > 0)		○	-	-	-	-	-
	else		-	○	○	○	-	○
	if((pul_val <= 0) && (luc_i > puc_dot) && (puc_app != 11))		○	-	-	-	-	-
	else		-	-	○	-	-	○
	if((pul_val <= 0) && (puc_dot == 0) && (puc_app != 11))		-	○	-	○	-	-
	else		-	-	○	-	-	○
	if(puc_app != 0)		○	○	○	○	-	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		-	-	○	-	-	○
	else		-	-	-	○	○	○
	for(luc_i = 0; luc_i < puc_col; luc_i++)		-	-	-	○	-	-
	if(lus_buff[luc_i] != 0x0020)		-	-	-	○	-	-
	else		-	-	-	○	-	-
	for(luc_i; luc_i < puc_col; luc_i++)		-	-	-	○	-	-
	Symbol	Range						
Input	pusp_str	0x0000~0xFFFF	0x0000	0x0000	0x0000	0x0000	0x0000	0xFFFF
	pul_val	0x00000000~0xFFFFFFFF	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	puc_col	0~9	5	0	8	8	0	9
	puc_dot	0~4	3	0	1	0	0	4
	puc_app	0~12	1	1	11	0	0	12
Output	None	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	usb_com.c

USBCom_Cmd_Proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

		case1	case2	case3	case4	case5
Root	if(m_Cmp_SUM(pucp_io_buff))	○	○	○	-	○
	while(!stp_tbl->uc_cmd[0])	○	○	○	-	○
	if((!stp_tbl->uc_cmd[0] == !pst_cmd->command[0]) && (!stp_tbl->uc_cmd[1] == !pst_cmd->command[1]))	-	-	-	-	○
	if(!stp_tbl->uc_cmd[0] == 0U)	○	○	○	-	○
	else	-	-	-	-	-
	else	-	-	-	○	-
	if (!ss_result_data == -1)	○	○	○	○	○
	if(!uc_error_num == 1U)	-	-	-	○	-
	else if(!uc_error_num == 2U)	○	○	○	-	-
	else	-	-	-	-	-
	Symbol	Range				
Input	pucp_io_buff[5]	0~57	1	0	1	0	57
	pucp_io_buff[6]	0~51	1	1	0	0	51
	m_cmd_ResponsTerm()	0	0	0	0	0	0
	m_Cmp_SUM()	0,1	1	1	1	0	1
Output	pucp_io_buff[0]	12,21	12	12	12	21	12
	!ss_result_data	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_AsciiDenst_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(Tool_Comp_Pnt(pucp_data, &uc_m_non_data[0], 8) == 0)	○	
	else	-	○
	Symbol	Range	
Input	*pucp_data	0	0	0
	Tool_Asc_Dec()	0	0	0
	Tool_Comp_Pnt()	0,1	0	1
Output	lul_ans	0~3722304989	3722304989	0
Judgment			OK	OK

m_Calc_PCSum

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

		case1	case2	case3
Root	for(lus_i = 0U; lus_i < pus_len; lus_i++)	○	-	-
	Symbol	Range		
Input	pucp_dat[0]	0	0	0	0
	pus_len	0~2	1	0	2
Output	luc_sum	0,1	0	0	1
Judgment			OK	OK	OK

m_Chk_AsciiOnOffDec

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(puc_data == '0')	○	-
	else	-	○
	Symbol	Range	
Input	puc_data	48,49	48	49
Output	luc_ans	0,1	0	1
Judgment			OK	OK

m_Chk_DecOnOffAscii

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(puc_data == M_OFF)	○	-
	else	-	○
	Symbol	Range	
Input	puc_data	48,49	0	1
Output	luc_ans	0,1	48	49
Judgment			OK	OK

m_Cmp_SUM

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

		case1	case2	case3	case4	case5	case6
Root	for(lus_i=0U;((lus_i<IrDaMain_TX_RX_BUFF_SIZE/*sizeof(ircommTxRxBuffer)*/) && (luc_result_data == 0U));lus_i++)	○	○	○	○	○	○
	if((buff[lus_i] == IRDA_EOT) || (buff[lus_i] == IRDA_CR) || (buff[lus_i] == IRDA_ETX))	○	-	○	○	-	-
	switch(buff[lus_i])	○	-	○	○	-	-
	case IRDA_EOT:	○	-	-	-	-	-
	case IRDA_CR:	-	-	-	○	-	-
	case IRDA_ETX:	-	-	○	-	-	-
	default:	-	-	-	-	-	-
	if (luc_result_data == 0U)	-	○		-	○	○
	if(("0123456789ABCDEF"[luc_sum>>4] == buff[lus_i]) && ("0123456789ABCDEF"[luc_sum&0xFU] == buff[lus_i+1U]))	-	-	-	-	-	-
	Symbol	Range					
Input	buff[0]	2~14	4	5	3	13	14	2
Output	luc_result_data	0,1	1	0	0	1	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_DenstAscii_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(pul_denst == M_OFF_VAL)	○	○
	else	○	-
	Symbol	Range	
Input	*pucp_data	0	0	0
	pul_denst	3722304989, 3722304990	3722304989	3722304990
	puc_dot	0	0	0
Output	None	-	-	-
Judgment			OK	OK

m_Get_UnitPnt

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(pul_denist == M_OFF_VAL)		○	○
	else		○	-
	Symbol	Range		
Input	*pucp_data	0x00~0xFF	0x00	0xFF
	puc_uni_code	0	0	0
Output	None	-	-	
Judgment			OK	OK

m_cmd_90_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'S')		○	○	-	-
	if(MAIN_Get_UpdateModeFlg() == M_ON)		○	-	-	-
	else		-	○	-	-
	else		-	-	○	○
	Symbol	Range				
Input	*pucp_io_buff	0	0	0	0	0
	pucp_io_buff[8]	83,84	83	83	84	84
	m_cmd_ResponsTerm()	0	0	0	0	0
	MAIN_Get_UpdateModeFlg()	1,2	1	2	1	2
Output	*pucp_io_buff	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_93_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case1	case1	case2
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	-	-
	case 'S':		○	○	-	-
	if((bl_mode_on_flg == true) && (lbl_sensor_on_flg == false))		-	○	-	-
	else		○	-	-	-
	default		-	-	○	○
	Symbol	Range				
Input	*pucp_io_buff	0	0	0	0	0
	pucp_io_buff[8]	83,84	83	83	84	84
	m_cmd_ResponsTerm()	0	0	0	0	0
	SelfSubProg_Set_ReWiteStart()	0,1	0	1	0	1
Output	*pucp_io_buff	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_94_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○	○	○	○
	case 'S':		○	○	-	-	-	-	-	-
	if(lsc_result != (S8_CHR)FALSE)		-	○	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-
	case 'W':		-	-	○	○	○	-	-	-
	if(fbl_frame_receive_flg == false)		-	-	○	-	-	-	-	-
	else		-	-	-	○	○	-	-	-
	if(lsc_result != (S8_CHR)FALSE)		-	-	-	-	○	-	-	-
	for(lus_i = 0U; lus_i < (U16_SHT)sizeof(luc_write_data); lus_i++)		-	-	-	-	○	-	-	-
	else		-	-	-	○	-	-	-	-
	case 'E':		-	-	-	-	-	○	○	-
	if(fus_total_sum == fus_received_sum)		-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	○	-
	default		-	-	-	-	-	-	-	○
	Symbol	Range								
Input	*pucp_io_buff	0	0	0	0	0	0	0	0	0
	pucp_io_buff[8]	32~83	83	83	87	87	87	69	69	32
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0
	RL78FProg_rewiteBlock()	0,1	0	0	0	0	1	0	0	0
	ul_current_block_top_addr	0	0	0	0	0	0	0	0	0
	us_split_data_counter	0	0	0	0	0	0	0	0	0
	us_write_sum	0	0	0	0	0	0	0	0	0
	RL78FProg_setStartBlock()	0,1	0	1	0	0	0	0	0	0
	RL78FProg_startRewritingProg()	0	0	0	0	0	0	0	0	0
	fbl_frame_receive_flg	0,1	0	0	0	1	1	0	0	0
	fus_received_sum	0,1	0	0	0	0	0	0	1	0
	fus_total_sum	0	0	0	0	0	0	0	0	0
	luc_write_data[0]	0	0	0	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0
	fbl_frame_receive_flg	0	0	0	1	0	0	0	0	0
	fus_received_sum	0~65280	0	0	0	0	65280	0	1	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

m_cmd_A7_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		○	-	-	-
	else		-	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		○	-	-	-
	else		-	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		○	-	-	-
	else		-	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		-	○	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_1st_alarm == M_OFF_VAL)		○	-	○	-
	else		-	○	-	-
	if((stp_GasInfo_addr+luc_ch)->ul_2nd_alarm == M_OFF_VAL)		○	-	○	-
	else		-	○	-	-
	if((stp_GasInfo_addr+luc_ch)->ul_3rd_alarm == M_OFF_VAL)		○	-	○	-
	else		-	○	-	-
	if((stp_GasInfo_addr+luc_ch)->ul_stel_alarm == M_OFF_VAL)		○	-	○	-
	else		-	○	-	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm == M_OFF_VAL)		○	-	○	-
	else		-	○	-	-
	default		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~82	87	87	82	32
	uc_uni_dec	0	0	0	0	0
	ul_1st_alarm	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
	ul_2nd_alarm	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
	ul_3rd_alarm	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
	ul_stel_alarm	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
	ul_twa_alarm	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD	0xDDDDDDDD
	m_AsciiDenst_proc()	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	Tool_Comp_Pnt()	0,1	0	1	1	1
	Tool_Hex_Asc()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	ul_1st_alarm	0, 0xDDDDDDDD	0xDDDDDDDD	0	0xDDDDDDDD	0xDDDDDDDD
	ul_2nd_alarm	0, 0xDDDDDDDD	0xDDDDDDDD	0	0xDDDDDDDD	0xDDDDDDDD
	ul_3rd_alarm	0, 0xDDDDDDDD	0xDDDDDDDD	0	0xDDDDDDDD	0xDDDDDDDD
	ul_stel_alarm	0, 0xDDDDDDDD	0xDDDDDDDD	0	0xDDDDDDDD	0xDDDDDDDD
	ul_twa_alarm	0, 0xDDDDDDDD	0xDDDDDDDD	0	0xDDDDDDDD	0xDDDDDDDD
Judgment			OK	OK	OK	OK

m_cmd_AF_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	pucp_io_buff[0]	0	0	0	0
	pucp_io_buff[10]	0	0	0	0
	uc_alarm_on	0,1	0	0	1
	m_Chk_AsciiOnOffDec()	0,1	0	0	1
	m_Chk_DecOnOffAscii()	0,1	0	0	1
	mlcmd_ResponsTerm()	0,1	0	0	1
Output	*picp_io_buff	0	0	0	0
	m_cmd_ResponsTerm()	0,1	0	0	1
	uc_alarm_on	0,1	0	0	1
Judgment			OK	OK	OK

m_cmd_AJ_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(luc_ret == M_OK)		○	-	-	-	-
	else		-	○	-	-	-
	case 'R':		-	-	○	○	-
	if(luc_check == M_ON)		-	-	○	-	-
	else		-	-	-	○	-
	if(luc_ret == M_OK)		-	-	○	-	-
	if(luc_ret == M_OK)		-	-	○	-	-
	else		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	0~87	87	87	82	82	32
	DataFram_AlarmReset_Load()	0,1	0	0	0	0	0
	DataFramFactry_Alarmpoint_Load()	0,1	0	0	0	0	1
	DataFramFactry_Chk_AlarmResetOnOffFlg()	0,1	1	0	1	0	1
	FlowGasSelDisplay_End_ResetAlarmPoint()	0,1	0	1	1	1	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_AM_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	Data_Detector_Set_DataAddress()	0	0	0	0
	uc_1h_alarm_on	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_1h_alarm_on	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_AP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	Data_Detector_Set_DataAddress()	0	0	0	0
	uc_alarm_pattern	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_alarm_pattern	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_B0_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	pucp_io_buff[12]	0	0	0	0
	Data_Detector_Set_DataAddress()	0	0	0	0
	uc_stealth_on	0	0	0	0
	uc_stealth_motor_on	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_stealth_on	0	0	0	0
	uc_stealth_motor_on	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_B7_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	case 'R':		○	○	○	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm_min == M_OFF_VAL)		-	○	○	-
	else		○	-	-	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm_max == M_OFF_VAL)		-	○	○	-
	else		○	-	-	-
	default		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	0~87	87	87	82	32
	*pucp_io_buff	0,1	1	1	1	1
	GasCalc_Get_GasData_Add()	0	0	0	0	0
	uc_uni_dec	0	0	0	0	0
	ul_twa_alarm_min	0	0	0	0	0
	ul_twa_alarm_max	0	0	0	0	0
	m_AsciiDenst_proc	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	Tool_Comp_Pnt()	0	1	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0
Output	*pucp_io_buff	0,1	1	1	1	1
	ul_twa_alarm_min	0	0	0	0	0
	ul_twa_alarm_max	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_B8_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	case 'R':		○	○	○	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm_min == M_OFF_VAL)		-	○	○	-
	else		○	-	-	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm_max == M_OFF_VAL)		-	○	○	-
	else		○	-	-	-
	default		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	0~87	87	87	82	32
	*pucp_io_buff	0,1	1	1	1	1
	GasCalc_Get_GasData_Add()	0	0	0	0	0
	uc_uni_dec	0	0	0	0	0
	ul_twa_alarm_min	0	0	0	0	0
	ul_twa_alarm_max	0	0	0	0	0
	m_AsciiDenst_proc	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	Tool_Comp_Pnt()	0	1	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0
Output	*pucp_io_buff	0,1	1	1	1	1
	ul_twa_alarm_min	0	0	0	0	0
	ul_twa_alarm_max	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_B9_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	case 'R':		○	○	○	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm_min == M_OFF_VAL)		-	○	-	-
	else		○	-	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_twa_alarm_max == M_OFF_VAL)		-	○	-	-
	else		○	-	○	-
	default		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	0~87	87	87	82	32
	*pucp_io_buff	0,1	1	1	1	1
	GasCalc_Get_GasData_Add()	0,1	1	1	1	1
	uc_uni_dec	0	0	0	0	0
	ul_twa_alarm_min	0	0	0	0	0
	ul_twa_alarm_max	0	0	0	0	0
	m_AsciiDenst_proc	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	Tool_Comp_Pnt()	0	1	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0
Output	*pucp_io_buff	0,1	1	1	1	1
	ul_twa_alarm_min	0~0xDDDDDDDD	0	0xDDDDDDDD	0	0
	ul_twa_alarm_max	0~0xDDDDDDDD	0	0xDDDDDDDD	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_BJ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	uc_bump_lmt_set	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	1	1	1
	uc_bump_lmt_set	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_BK_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	*pucp_io_buff	0	0	0	0
	us_bump_lmt_day	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_BL_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	○	-	-
	if(pucp_io_buff[ius_r_pos] == '2')		○	-	-	-	-
	else if(pucp_io_buff[ius_r_pos] == '1')		-	-	○	-	-
	else		-	○	-	-	-
	case 'R':		○	○	○	○	-
	if(stp_data->uc_bump_lmt_chk == 2)		○	-	-	○	-
	else if(stp_data->uc_bump_lmt_chk == 1)		-	-	○	-	-
	else		-	○	-	-	-
	default		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	0~87	87	87	87	82	32
	pucp_io_buff[10]	0~51	50	51	49	48	48
	uc_bump_lmt_chk	0~2	1	2	1	2	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	uc_bump_lmt_chk	0~2	2	0	1	2	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_BO_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-
	if(luc_i != 0U)		○	-
	else		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	0~82	82	32
	*pucp_io_buff	1	1	1
	uc_uni_dec	0	0	0
	uc_bump_date[0]	0	0	0
	uc_bump_date[1]	0	0	0
	uc_bump_date[2]	0	0	0
	uc_bump_date[3]	0	0	0
	uc_bump_date[4]	0	0	0
	uc_bump_date[5]	0	0	0
	ul_bump_val	0	0	0
	ul_bump_test_val	0	0	0
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	0	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_BP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(pucp_io_buff[ius_r_pos] == '0')		○	-	-	-	-
	else		-	○	-	-	-
	if(pucp_io_buff[ius_r_pos+2U] == '0')		○	-	-	-	-
	else		-	○	-	-	-
	case 'R':		○	○	○	○	-
	if(PumpFlow_Get_LoHiFlag_isHi() == true)		○	-	○	-	-
	else		-	○	-	○	-
	if(PumpFlow_Get_MotionFlag_isMove() == true)		○	○	-	-	-
	else		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	0~87	87	87	82	82	32
	pucp_io_buff[10]	0~49	48	49	48	48	48
	pucp_io_buff[12]	48,49	48	49	49	48	48
	PumpFlow_Get_LoHiFlag_IsHi()	0~2	1	0	1	2	1
	Pump_Flow_Get_MotionFlag_isMove()	0~2	1	1	1	0	2
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_BS_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	*pucp_io_buff	0	0	0	0
	uc_sep_no[0]	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_BU_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	pucp_io_buff[19]	0	0	0	0
	uc_bump_time	0	0	0	0
	uc_bump_limit	0	0	0	0
	uc_bump_cal_time	0	0	0	0
	uc_bamp_cal_flg	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	pucp_io_buff[12]	0~44	44	44	0
	uc_bump_cal_flg	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_CP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	if((us_flow_fail_threshold >= PUMPFLOW_FLOW_FAIL_SPAN_MIN) && (us_flow_fail_threshold <= PUMPFLOW_FLOW_FAIL_SPAN_MAX))		○	-	-
	else		-	-	-
	case 'R':		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	0~87	87	82	32
	*pucp_io_buff	0,1	1	1	1
	us_flow_rate_drop_output	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	us_flow_rate_drop_output	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_D2_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-
	if((luc_result_temp & 0x01) == 0U)		○	-	○	○	-
	else		-	○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-
	if(luc_bump_result[luc_i] == M_OK)		○	-	○	○	-
	if((stp_denst + luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))		○	-	-	-	-
	else		○	○	○	○	-
	else		-	○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	○	○	-
	if(lul_bump_result_denst != M_OFF_VAL)		○	-	○	-	-
	else		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	0~87	87	87	87	87	32
	uc_bump_limit	0	0	0	0	0	0
	uc_state_trb	0~255	255	255	0	0	0
	FlowBumpTestDisplay_Com_BumpTest()	0,1	0	1	0	0	0
	FlowBumpTestDisplay_Get_BumpResultDenst()	0x00000000~0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDD
	uc_uni_dec	0	0	0	0	0	0
	ul_cal_denst	0	0	0	0	0	0
	m_AsciiDenst_proc	0	0	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0~50	50	49	48	48	12
	uc_bump_limit	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_D3_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	-
	case 'W':		○	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-
	if(lul_fast_cal_denst[luc_i] != M_OFF_VAL)		○	○	○	○	-
	if((luc_success_flg & (0x01 << luc_i)) != 0x00)		○	○	-	-	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32,87	87	87	87	87	32
	pucp_io_buff[49]	48~50	48	48	49	49	50
	uc_bump_date[0]	0,1	0	1	0	1	0
	uc_bump_sdm[0]	0,1	0	1	0	1	0
	m_AsciiDenst_proc()	0,1	0	1	0	1	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	ul_bump_val	0,1	0	1	0	0	0
	ul_bump_test_val	0,1	0	1	0	0	0
	m_cmd_ResponsTerm()	0,1	0	1	0	1	0
Judgment			OK	OK	OK	OK	OK

m_cmd_DH_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	11

Judgment	OK

		case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')	○	○	○	○	○	○	○	○	○	○	○
	if(FlowMain_Get_ModeStatus() == FlowMain_IRDA_MODE_NUMBER)	○	-	○	○	○	○	○	○	○	○	-
	else	-	○	-	-	-	-	-	-	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	○	○	○	○	○	○	○	○	-
	switch(Battery_Get_ComBatteryIconAD())	○	○	○	○	○	○	○	○	○	○	-
	case BATTERY_BATT_STATUS_FULL:	○	○	-	-	-	-	-	-	-	○	-
	case BATTERY_BATT_STATUS_MANY:	-	-	○	-	-	-	-	-	-	-	-
	case BATTERY_BATT_STATUS_NORMAL:	-	-	-	○	-	-	-	-	-	-	-
	case BATTERY_BATT_STATUS_WARNING:	-	-	-	-	○	-	-	-	-	-	-
	case BATTERY_BATT_STATUS_ALARM:	-	-	-	-	-	○	-	-	-	-	-
	case BATTERY_BATT_STATUS_SHUT_DOWN:	-	-	-	-	-	-	○	-	-	-	-
	default:	-	-	-	-	-	-	-	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	○	○	○	○	○	○	○	○	-
	if(luc_i != 0U)	○	○	○	○	○	○	○	○	○	○	-
	else	○	○	○	○	○	○	○	○	○	○	-
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && ((stp_denst+luc_i) ->ul_denst != M_OFF_VAL))	○	○	○	○	○	○	○	○	○	○	-
	if((luc_i == DataStructConst_CO_CAN_H2_H2_CH) && ((stp_GasInfo_addr+luc_i) ->uc_gas_comb == DataStructConst_CO_CAN_H2_GAS_NUM))	-	-	-	-	-	-	-	-	-	-	-
	else	○	○	○	○	○	○	○	○	○	○	-
	if((stp_denst + luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS))	-	-	-	-	-	-	-	-	-	○	-
	else if((stp_denst + luc_i)->uc_state_trb & (DataStructConst_S_FAIL_SEN))	-	-	-	-	-	-	-	-	-	-	-
	else	○	○	○	○	○	○	○	○	○	○	-
	if((stp_denst+luc_i)->uc_denst_minus_flg == M_ON)	○	○	○	○	○	○	○	○	-	-	-
	else	○	○	○	○	○	○	○	○	○	○	-
	else	○	○	○	○	○	○	○	○	○	○	-
	else	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range										
Input	pucp_io_buff[8]	32,87	87	87	87	87	87	87	87	87	87	32
	*pucp_io_buff	1	1	1	1	1	1	1	1	1	1	1
	Battery_Get_ComBatteryAD()	0	0	0	0	0	0	0	0	0	0	0
	Battry_Get_ComBatteryIconAD()	0~6	5	5	4	3	2	1	0	6	6	5	5
	ul_denst	0x00000000~0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE
	uc_denst_minus_flg	1,2	1	1	1	1	1	1	1	1	1	1	2
	uc_state_trb	0,255	0	0	0	0	0	0	0	0	0	255	0
	uc_alarm_lat	0	0	0	0	0	0	0	0	0	0	0	0
	us_main_state	0	0	0	0	0	0	0	0	0	0	0	0
	FlowMain_Get_ModeStatus()	7,8	7	8	7	7	7	7	7	7	7	7	7
	uc_gas_comb	1	1	1	1	1	1	1	1	1	1	1	1
	uc_uni_dec	0	0	0	0	0	0	0	0	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0	0	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1	1	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_cmd_DP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

		case1	case2	case3	case4	case5
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')	○	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	○	○	-
	if(luc_i != 0U)	○	○	○	○	-
	else	-	-	-	-	-
	if(luc_i == DataStructConst_O2_GAS_CH)					
	else					
	if(((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00) && (lul_peak_denst != M_OFF_VAL))	○	○	○	○	-
	if((stp_denst + luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))	-	-	-	○	-
	else	○	○	○	○	-
	if(luc_peak_flg == M_ON)	○	-	-	-	-
	else	○	○	○	○	-
	else	○	○	○	○	-
	else	-	-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	0~82	82	82	82	82	32
	*pucp_io_buff	1	1	1	1	1	1
	uc_state_trb	9~255	0	0	255	0	0
	ul_pmin	0	0	0	0	0	0
	ul_pmax	0	0	0	0	0	0
	uc_pmin_flg	0	0	0	0	0	0
	uc_pmax_flg	0~2	1	1	1	2	0
	uc_gas_comb	0,1	0	1	0	0	0
	uc_uni_dec	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_DW_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'W'	○	-
	else	-	○
	Symbol	Range	
Input	pucp_io_buff[8]	32,87	87	32
	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_EF_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

		case1	case2	case3
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')	○	○	-
	if(luc_ans == M_OK)	○	-	-
	else	-	○	-
	else	-	-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~82	82	82	32
	*pucp_io_buff	1	1	1	1
	DataFram_get_Dumpload()	0,1	0	1	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	pucp_io_buff[14]	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_EP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0]		○	○	○
	case 'R':		○	○	-
	if((0x100000 / M_TRANS_DUMP_SIZE) <= ius_block)		-	-	-
	else		○	○	-
	default		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32,82	82	82	32
	flash_read()	0,1	0	1	1
Output	pucp_io_buff[14]	0,44	44	44	0
	ius_w_pos	15, 275	275	275	15
Judgment			OK	OK	OK

m_cmd_EX_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'W')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,87	87	32
	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0,1	0	1
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0,1	0	1
Judgment			OK	OK

m_cmd_FB_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(luc_ret == M_OK)		○	-	-	-	-
	if(luc_ret == M_OK)		○	-	-	-	-
	else		-	○	-	-	-
	case 'R':		-	-	○	-	-
	if(luc_ret == M_OK)		-	-	○	-	-
	if(luc_ret == M_OK)		-	-	-	-	-
	else		-	-	-	○	-
	default		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	*pucp_io_buff	1	1	1	1	1	1
	DataFramFactory_All_load()	0,1	0	1	0	1	1
	DataFramFactory_write_op()	0,1	0	1	0	1	1
	FlowGasSelDisplay_End_ResetAlarmPoint()	0,1	0	1	0	1	1
	m_cmd_ResponsTerm()	0,1	0	1	0	1	1
Output	*pucp_io_buff	1	1	1	1	1	1
	m_cmd_ResponsTerm()	0,1	0	1	0	1	1
Judgment			OK	OK	OK	OK	OK

m_cmd_G0_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if((FlowMain_Get_ModeStatus() == FlowMain_MEASURE_MODE_NUMBER) || (FlowMain_Get_ModeStatus() == FlowMain_DISP_MODE_NUMBER))		○	○	○	○	-	○
	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	-	○
	case 'W':		○	○	-	-	-	-
	if(pucp_io_buff[ius_r_pos] == '1')		○	○	-	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	-	-	-	-
	for(luc_j=0; luc_j<15; luc_j++)		○	-	-	-	-	-
	for(luc_j=0U; luc_j<9U; luc_j++)		○	-	-	-	-	-
	else		-	○	-	-	-	-
	case 'R':		-	-	○	○	-	-
	if(LoggerGXForce_Get_LoggerRun() == M_ON)		-	-	-	○	-	-
	else		-	-	○	-	-	-
	default:		-	-	-	-	-	○
	else		-	-	-	-	○	-
	Symbol	Range						
Input	pucp_io_buff[8]	32~87	87	87	82	82	82	32
	pucp_io_buff[10]	49,50	49	50	50	50	50	49
	FlowMain_Get_ModeStatus()	1~3	1	2	1	1	3	1
	LoggerGXForce_Get_LoggerRun()	0~2	1	1	0	1	0	2
	LoggerGXForce_Set\StopLoggerInterval()	0	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0	0
	ul_60_total	0	0	0	0	0	0	0
	ul_ave	0	0	0	0	0	0	0
	ul_ave_total_cnt	0	0	0	0	0	0	0
	ul_stel_point	0	0	0	0	0	0	0
	ul_stel_buf	0	0	0	0	0	0	0
	ul_stel	0	0	0	0	0	0	0
	ul_twa	0	0	0	0	0	0	0
	ul_60min_buff	0	0	0	0	0	0	0
	uc_60min_count	0	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_cmd_GJ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

		case1	case2	case3	case4
Root	switch((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])	○	○	○	○
	case 'W':	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos+(luc_i*3)], &uc_m_non_data[0], 2) != 0)	○	-	-	-
	if((luc_gas < HDDEF_GAS_TABLE_MAX) && ((stp_GasInfo_addr+luc_i)->uc_gas_comb != luc_gas))	○	-	-	-
	if(Tool_Comp_Pnt(&GasTableDefData[luc_i][luc_gas].uc_name[0], &uc_m_non_data[0], 10) != 0)	○	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	○	-	-
	if(luc_i != 0U)	○	○	-	-
	else	○	○	-	-
	case 'R':	-	-	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	-	-	○	-
	if(luc_i != 0U)	-	-	○	-
	else	-	-	○	-
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)	-	-	○	-
	else	-	-	○	-
	default:	-	-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	87	82	32
	*pucp_io_buff	1	1	1	1	1
	uc_gas_comb	0,1	1	0	0	1
	uc_disp_name[0][0]	0,1	0	0	0	1
	uc_name[0][0]	0,1	0	0	0	1
	ul_full_scale	0,1	0	0	0	1
	ul_digit	0,1	0	0	0	1
	uc_uni_dec	0,1	0	0	0	1
	Tool_Comp_Pnt()	0,1	1	1	0	1
	Tool_Hex_Asc()	0,1	0	0	0	1
	m_cmd_ResponsTerm()	0,1	0	0	0	1
Output	*pucp_io_buff	1	1	1	1	1
	m_cmd_ResponsTerm()	0,1	0	0	0	1
Judgment			OK	OK	OK	OK

m_cmd_GT_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

		case1	case2	case3
Root	switch((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])	○	○	○
	case 'W':	○	-	-
	case 'R':	○	○	-
	default:	-	-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	uc_long_energy_flg	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_long_energy_flg	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_GY_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

		case1	case2	case3	case4	case5
Root	switch((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])	○	○	○	○	○
	case 'W':	○	○	-	-	-
	if(pucp_io_buff[lus_r_pos] == '0')	○	-	-	-	-
	else	-	○	-	-	-
	case 'R':	○	○	○	○	-
	if(stp_data->uc_disp_long_e_flg == GAS_CALC_NC_DISP_LONG_OFF)	○	-	-	○	-
	else	-	○	○	-	-
	default:	-	-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	pucp_io_buff[10]	48~50	48	49	50	50	50
	uc_disp_long_e_flg	0,1	0	0	0	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	uc_disp_long_e_flg	0,1	1	0	0	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_H1_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	-
	if(luc_i != 0U)	○	-
	else	○	-
	else	-	○
	Symbol	Range	
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	uc_disp_long_e_flg	1	1	1
	uc_cal_sdm[0]	0,1	0	1
	m_cmd_ResponsTerm()	1	1	1
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0,1	0	1
Judgment			OK	OK

m_cmd_H3_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

		case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)	○	-
	if(luc_i != 0U)	○	-
	else	○	-
	else	-	○
	Symbol	Range	
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	uc_bump_sdm[0]	0	0	0
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_HA_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	for(luc_i=0U; luc_i<AD12BIT_CH_MAX; luc_i++)		○	-
	for(luc_i=0U; luc_i<2U; luc_i++)		○	-
	for(luc_i=0U; luc_i<3U; luc_i++)		○	-
	for(luc_i=0U; luc_i<3U; luc_i++)		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	Ad12bit_Get_Ad_Data()	0	0	0
	fl_nc_ad_data[0][0]	0	0	0
	fl_ec_ad_data[0]	0	0	0
	fl_temp_ad_data	0	0	0
	fl_temp_data	0	0	0
	Tool_Hex_Asc()	0	0	0
	m_cmd_ResponsTerm()	0	0	0
Output	pucp_io_buff[18]	0,44	0	44
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_HC_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	RTC_DateAndTime_Config()	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_HF_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	if(((us_dac_output >= PRESSURE_OUTPUT_MIN_PER) && (us_dac_output <= PRESSURE_OUTPUT_MAX_PER))		-	-	-
	else		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	us_flow_output	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	us_flow_output	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_HN_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	uc_f_serial[0]	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_IN_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	uc_f_serial[0]	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_JZ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_key_tone_onoff == M_OFF)		○	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	82	82	32
	pucp_io_buff[10]	0,48	48	0	0	0
	uc_key_tone_onoff	0,1	1	0	0	0
	m_cmd_ResponsTerm()	0,1	1	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_key_tone_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0,1	1	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_KW_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	KeyEvent_Get_KeyPort()	0	0	0
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_LB_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(luc_ret == M_OK)		○	-	-	-	-
	else		-	○	-	-	-
	case 'R':		-	-	○	○	-
	if(luc_ret == M_OK)		-	-	○	○	-
	else		-	-	-	-	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	*pucp_io_buff	1	1	1	1	1	1
	FuncLunchBreak_Get_LunchBreakFramData()	0,1	0	1	0	1	0
	FuncLunchBreak_Set_LunchBreakFramData()	0,1	0	1	0	1	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_LI_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	○	-
	if(luc_gas < HDDEF_GAS_MAX && luc_list < HDDEF_GAS_TABLE_MAX)		○	○	-
	else		-	-	-
	else		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~82	82	82	32
	*pucp_io_buff	1	1	1	1
	Tool_Hex_Asc()	0,1	0	1	0
	m_cmd_ResponsTerm()	0,1	0	1	0
Output	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0,1	0	1	0
Judgment			OK	OK	OK

m_cmd_LJ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '2')		○	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '1')		-	○	-	-	-
	else		-	-	○	-	-
	case 'R':		○	○	○	○	-
	if(stp_data->uc_lel_sel_flg == 2)		○	-	-	○	-
	else if(stp_data->uc_lel_sel_flg == 1)		-	○	-	-	-
	else		-	-	○	-	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	87	82	32
	pucp_io_buff[10]	0~50	50	49	48	0	0
	uc_lel_sel_flg	0~2	2	1	0	2	0
	uc_gas_comb	0	0	0	0	0	0
	uc_sub_comb	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	uc_lel_sel_flg	0~2	2	1	0	2	0
	fl_lel_ratio	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_LK_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	○
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	uc_nc_mode_flg	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_LP_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	if(luc_data != 0 && luc_data < HDDEF_GAS_TABLE_MAX)		-	-	-	-
	else		○	-	-	-
	case 'R':		○	○	-	-
	case 'G':		-	-	○	-
	if((luc_data < HDDEF_GAS_TABLE_MAX) && ((GasTableDefData[0][luc_data].uc_uni_dec & 0xF0) == 0x10))		-	-	-	-
	if((HCTableDefData[luc_data].uc_meas_group & 0x01) == 0x01)		-	-	-	-
	else		-	-	-	-
	else		-	-	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	82	71	32
	*pucp_io_buff	1	1	1	1	1
	Tool_Hex_Asc()	0	0	0	0	0
	uc_sub_comb	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1
	uc_sub_comb	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_LQ_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	-
	case 'R':		○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	-
	if(luc_i != 0U)		○	○	-
	else		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	uc_cal_group	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	uc_cal_group	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_MO_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	-	-
	if(stp_data->uc_user_pass_onoff == M_OFF)		○	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~50	48	49	50	0
	uc_user_pass_onoff	0,1	1	0	0	0
	us_user_pass_num	0	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_user_pass_onoff	0,1	0	1	0	0
	us_user_pass_num	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_MP_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	-	-
	if(stp_data->uc_user_pass_onoff == M_OFF)		○	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~50	48	49	50	0
	uc_mainte_pass_onoff	0,1	1	0	0	0
	us_mainte_pass_num	0	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_mainte_pass_onoff	0,1	0	1	0	0
	us_mainte_pass_num	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_MR_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(pucp_io_buff[lus_r_pos+7] == '2')		○	-	-	-	-
	else if(pucp_io_buff[lus_r_pos+7] == '1')						
	else		-	○	-	-	-
	case 'R':		○	○	-	-	-
	if(stp_data->uc_mainte_lmt_chk == 2)		○	-	○	○	-
	else if(stp_data->uc_mainte_lmt_chk == 1)						
	else		-	○	-	-	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	pucp_io_buff[15]	0,1	0	0	0	0	1
	pucp_io_buff[17]	0~50	50	49	48	0	0
	uc_mainte_notice_onoff	0	0	0	0	0	0
	us_mainte_lmt_chk	0~2	2	1	0	2	0
	us_mainte_notice_day	0	0	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1
	uc_mainte_notice_onoff	0	0	0	0	0	0
	us_mainte_lmt_chk	0~2	2	1	0	2	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_MX_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_MY_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(stp_data->uc_nci_download_flg == M_OFF)		○	-	-	-
	case 'R':		○	○	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0,1	0	0	0	1
	uc_nci_download_flg	0,1	0	1	0	0
	m_Chk_AsciiOnOffDec()	0,1	0	1	0	0
	m_Chk_DecOnOffAscii()	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0,1	0	1	0	0
Output	*pucp_io_buff	1	1	1	1	1
	uc_nci_download_flg	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_MZ_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	-	-
	if(pucp_io_buff[lus_r_pos+luc_i] == '1')		○	-	-	-	-
	else if(pucp_io_buff[lus_r_pos+luc_i] == '2')		-	○	○	-	-
	else		○	○	-	-	-
	case 'R':		○	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○	-
	if((stp_GasInfo_addr+luc_i)->uc_nci_active_flg == FUNC_BEEP_NCI_FLG_ON)		○	-	-	-	-
	else if((stp_GasInfo_addr+luc_i)->uc_nci_active_flg == FUNC_BEEP_NCI_FLG_DL)		-	○	○	○	-
	else		○	○	-	○	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	87	82	32
	pucp_io_buff[10]	0~51	49	50	51	0	0
	uc_nci_active_flg	0~2	0	1	2	2	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	uc_nci_active_flg	0~2	1	2	0	2	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_OI_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	if(pucp_io_buff[lus_r_pos+2] == '0'){		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_bump_end_initial == M_OFF)		○	-	○	-
	else		-	○	-	-
	if(stp_data->uc_cal_end_initial == M_OFF)		○	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	pucp_io_buff[12]	0~49	48	49	0	0
	uc_bump_end_initial	0,1	1	0	0	0
	uc_cal_end_initial	0,1	1	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_bump_end_initial	0,1	0	1	0	0
	uc_cal_end_initial	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_PG_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_protect_pword_onoff == M_OFF)		○	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_protect_pword_onoff	0,1	1	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_protect_pword_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_PN_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	uc_margin_disp_onoff	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_margin_disp_onoff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_PS_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	-
	case 'R':		○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	-
	if(luc_i != 0U)		○	○	-
	else		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	uc_notice_date	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_notice_date	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_RA_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	○	-	-
	else		-	○	-	-
	if(luc_buzzer_flg > 2)		-	-	-	-
	case 'R':		○	○	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	BUZZER_Chk_TestFlg()	0	0	0	0	0
	Tool_Asc_Dec()	0~2	0	2	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_RC_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_RL_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	-
	case 'S':		○	-	-
	case 'E':		-	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~83	83	69	32
	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_RM_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'S':		○	-	-
	case 'E':		-	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~83	83	69	32
	pucp_io_buff[10]	0	0	0	0
	pucp_io_buff[12]	0	0	0	0
	pucp_io_buff[14]	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_RO_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'W')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,87	87	32
	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_ResponsTerm

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	None		-	-
	Symbol	Range		
Input	*pucp_io_buff	0,255	0	255
	pucp_io_buff	0,65535	0	65535
	m_Calc_PCSum()	0,1	0	1
	str_set_uw_hex	0,1	0	1
Output	pus_w_pos	2	2	2
	*pucp_io_buff	3,255	3	255
Judgment			OK	OK

m_cmd_S2_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	14

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12	case13	case14
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○	○	○	○	○	○	○	○	○	○
	case 'W':		○	○	○	-	-	-	-	-	-	-	-	-	-	-
	for(luc_i = 0U; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	○	-	-	-	-	-	-	-	-	-	-	-
	if(GasFuncAutoCal_Calc_Get_AutoCalFlg() == M_OFF)		○	-	○	-	-	-	-	-	-	-	-	-	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	○	-	-	-	-	-	-	-	-	-	-	-
	if(lul_denst != M_OFF_VAL)		○	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-	-	-	-
	case 'D':		-	-	-	-	○	○	○	○	○	-	-	-	-	-
	if(!(stp_data->uc_up_cal_flg & 0x01))		-	-	-	-	-	○	○	○	○	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-	-	-
	if(!(stp_data->uc_up_cal_flg & 0x02))		-	-	-	-	○	-	○	○	○	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-	-	-	-
	if(!(stp_data->uc_up_cal_flg & 0x04))		-	-	-	-	○	○	-	○	○	-	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-	-	-	-	-
	if(!(stp_data->uc_up_cal_flg & 0x08))		-	-	-	-	○	○	○	-	○	-	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-	-	-	-
	case 'R':		-	-	-	○	○	○	○	○	○	-	-	-	-	-
	case 'C':		-	-	-	-	-	-	-	-	-	○	○	○	○	-
	if(GasFuncAutoCal_Calc_Get_AutoCalFlg() == M_OFF)		-	-	-	-	-	-	-	-	-	○	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	-	-	-	-	-	-	-	○	○	○	-	-
	if(lucs_cal_flg == M_ON)		-	-	-	-	-	-	-	-	-	○	○	○	-	-
	if((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00)		-	-	-	-	-	-	-	-	-	-	○	○	-	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_SPN)		-	-	-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	○	-	-
	if((stp_denst + luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	○	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	○	-	-
	else		-	-	-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	-	-	○	-
	default:		-	-	-	-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range														
Input	pucp_io_buff[8]	32~87	87	87	87	82	68	68	68	68	68	67	67	67	67	32
	*pucp_io_buff	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	uc_up_cal_flg	0~8	0	0	0	0	1	2	4	8	0	0	0	0	1	1
	uc_state_trb	0,255	255	255	255	255	255	255	255	255	255	255	255	0	0	0
	uc_gas_comb	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	uc_cal_sdm[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	fl_span_temp_data	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	fl_span_buf[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	fl_span_buf[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GasFuncAutoCal_Calc_Get_AutoCalFlg()	0,1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	m_AsciiDenst_proc()	0XDDDDDDDD,0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE	0xDDDDDDDE
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	lucs_cal_flg	1,2	1	1	1	1	1	1	1	1	1	2	1	1	1	1
	lucs_cal_result[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	uc_cup_cal_flg	0~15	0	0	0	0	15	15	15	15	15	0	0	0	1	1
	fl_span_temp_data	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	lucs_cal_flg	0,1	1	1	1	1	1	1	1	1	1	0	0	0	1	1
	lucs_cal_result[0]	0~49	48	48	48	0	0	0	0	0	0	0	49	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_cmd_S4_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	case 'R':		○	○	○	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	○	-	-
	else		○	-	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_cal_denst_min == M_OFF_VAL)		-	○	-	-
	else		○	-	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_cal_denst_max == M_OFF_VAL)		-	○	-	-
	else		○	-	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	*pucp_io_buff	1	1	1	1	1
	uc_uni_dec	0	0	0	0	0
	ul_cal_denst_min	0xDDDDDDDD,DDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD
	ul_cal_denst_max	0xDDDDDDDD,DDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD
	m_AsciiDenst_proc()	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	Tool_Comp_Pnt()	0,1	1	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1
	ul_cal_denst_min	0xDDDDDDDD,DDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD
	ul_cal_denst_max	0xDDDDDDDD,DDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_SA_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	-	-	-
	else		○	○	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 8) == 0)		-	○	-	-
	else		○	-	-	-
	case 'R':		○	○	○	-
	if(luc_ch >= HDDEF_GAS_MAX)		-	○	-	-
	else		○	-	○	-
	if((stp_GasInfo_addr+luc_ch)->ul_cal_denst_min == M_OFF_VAL)		-	○	-	-
	else		○	-	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	*pucp_io_buff	1	1	1	1	1
	uc_uni_dec	0	0	0	0	0
	ul_cal_denst	0xDDDDDDDD,DDDDDDDE	0xDDDDDDDE	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD
	m_AsciiDenst_proc()	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
	Tool_Comp_Pnt()	0,1	1	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1
	ul_cal_denst	0x00000000~0xDDDDDDDE	0	0xDDDDDDDD	0xDDDDDDDE	0xDDDDDDDD
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_SC_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32,87	87	82	32
	*pucp_io_buff	1	1	1	1
	us_cal_lmt_day	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_SD_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32,87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	us_cal_lmt_set	0	0	0	0
	m_Chk_AsciiOnOff()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_SG_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_demand_zero_onoff == M_OFF)		-	○	○	-
	else		○	-	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32,87	87	82	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_demand_zero_onoff	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_demand_zero_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_SJ_proc

	Responsible	Last update
Coding	lkarashi	2021/7/6
Test	Taki	
Judgment	lkarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_auto_zero_onoff == M_OFF)		-	○	○	-
	else		○	-	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	82	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_auto_zero_onoff	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_auto_zero_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_SM_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '2')		○	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '1')		-	○	-	-	-
	else		-	-	○	-	-
	case 'R':		○	○	○	○	-
	if(stp_data->uc_cal_lmt_chk == 2)		○	-	○	○	-
	else if(stp_data->uc_cal_lmt_chk == 1)		-	○	-	-	-
	else		-	-	○	-	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	87	82	32
	pucp_io_buff[10]	0~50	50	49	0	0	0
	uc_cal_lmt_chk	0~2	2	1	0	2	0
	m_cmd_ReponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	uc_cal_lmt_chk	0~2	2	1	0	2	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_SO_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-
	if(luc_i != 0U)		○	-
	else		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32~82	82	32
	*pucp_io_buff	1	1	1
	uc_uni_dec	0	0	0
	uc_cal_date[0]	0	0	0
	uc_cal_date[1]	0	0	0
	uc_cal_date[2]	0	0	0
	uc_cal_date[3]	0	0	0
	uc_cal_date[4]	0	0	0
	uc_cal_date[5]	0	0	0
	ul_cal_before	0	0	0
	ul_cal_after	0	0	0
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_SZ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	7

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○	○	○
	case 'W':		○	○	-	○	-	-	-
	for(luc_i = 0U; luc_i < HDDEF_GAS_MAX; luc_i++)		○	○	-	-	-	-	-
	if(GasFuncAir_Calc_Get_AirCalFlg() == M_OFF)		○	-	-	-	-	-	-
	else		-	○	-	-	-	-	-
	case 'R':		-	-	○	-	-	-	-
	for(luc_i=DataStructConst_O2_GAS_CH; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	○	-	-	-	-
	case 'C':		-	-	-	○	○	○	-
	if(GasFuncAir_Calc_Get_AirCalFlg() == M_OFF)		-	-	-	○	○	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	-	-	○	○	-	
	if(lucs_cal_flg == M_ON)		-	-	-	○	○	-	-
	if((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00)		-	-	-	○	-	-	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_ZRO)		-	-	-	○	-	-	-
	else		-	-	-	○			-
	if((stp_denst + luc_i)->uc_state_trb & (DataStructConst_S_FAIL_INS | DataStructConst_S_FAIL_SEN))		-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	○	-	-	-
	else		-	-	-	○	○	-	-
	else		-	-	-	○	-	○	-
	default:		-	-	-	-	-	-	○
	Symbol	Range							
Input	pucp_io_buff[8]	32~87	87	87	82	67	67	67	32
	*pucp_io_buff	1	1	1	1	1	1	1	1
	uc_state_trb	0,255	255	255	255	255	255	255	0
	uc_gas_comb	1	1	1	1	1	1	1	1
	fl_zero_temp_data	0	0	0	0	0	0	0	0
	fl_zero_buf[0]	0	0	0	0	0	0	0	0
	fl_zero_buf[1]	0	0	0	0	0	0	0	0
	fl_zero_humidity_voltage[0]	0	0	0	0	0	0	0	0
	fl_zero_humidity_voltage[1]	0	0	0	0	0	0	0	0
	GasFuncAir_Calc_Get_AirCalFlg()	0,1	0	1	1	0	0	1	0
	lucs_cal_flg	0,1	0	1	1	1	0	1	1
	lucs_cal_result[0]	0	0	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1	1	1
	lucs_cal_flg	0,1	1	1	1	0	0	1	1
	lucs_cal_result[0]	0~49	48	48	0	49	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK

m_cmd_U8_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	-	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_disp_setting_item_onoff == M_OFF)		○	-	○	-
	else		-	-	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_disp_setting_item_onoff	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_disp_setting_item_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_UJ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○	○
	case 'W':		○	○	○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '1')		-	○	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '2')		-	-	○	-	-	-
	else		-	-	-	○	-	-
	case 'R':		○	○	○	○	○	-
	if(stp_data->uc_exp_flg == DSC_TGT_JAPAN)		○	-	-	-	○	-
	else if(stp_data->uc_exp_flg == DSC_TGT_EXPORT)		-	○	-	-	-	-
	else if(stp_data->uc_exp_flg == DSC_TGT_OEM)		-	-	○	-	-	-
	else		-	-	-	○	-	-
	default:		-	-	-	-	-	○
	Symbol	Range						
Input	pucp_io_buff[8]	32~87	87	87	87	87	82	32
	pucp_io_buff[10]	0~51	48	49	50	51	0	0
	uc_exp_flg	0~3	0	1	2	3	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0	0
	uc_exp_flg	0~3	0	1	2	3	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

m_cmd_US_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	pucp_io_buff[12]	0	0	0	0
	pucp_io_buff[14]	0	0	0	0
	pucp_io_buff[16]	0	0	0	0
	pucp_io_buff[18]	0	0	0	0
	uc_disp_suppress_val	0	0	0	0
	uc_suppress_onoff	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_disp_suppress_val	0	0	0	0
	uc_suppress_onoff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_UX_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	case 'C':		-	○	○	-
	if (FRAM_Get_DefaultEndFlg() == M_OFF)		-	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	67	67	32
	*pucp_io_buff	1	1	1	1	1
	FRAM_Get_DefaultEndFlg()	0,1	0	1	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_UZ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	pucp_io_buff[12]	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_disp_zero_filwr	0	0	0	0
	uc_follower_onoff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_VL_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	GasCalcTbl_Get_RomNum()	0	0	0
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_VP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	MAIN_Get_RomNum()	0	0	0
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_VQ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	uc_rom_num[0]	0	0	0
	us_rom_sum	0	0	0
	uc_rom_ver[0]	0	0	0
	Tool_Hex_Asc()	0	0	0
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_VS_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	FuncRomChk_Get_SumNum()	0	0	0
	Tool_Hex_Asc()	0	0	0
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_VT_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_VV_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	GasCalcTbl_Get_SumNum()	0	0	0
	Tool_Hex_Asc()	0	0	0
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_W0_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'R')		○	-
	else		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,82	82	32
	*pucp_io_buff	1	1	1
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_W8_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5
Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-	-
	else		-	○	-	-	-
	case 'R':		-	-	○	○	-
	if(lbl_mode_on_flg == false)		-	-	-	○	-
	else		-	-	○	-	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	pucp_io_buff[10]	0~49	48	49	0	0	0
	SelfSubProg_Get_ModeOnFlg()	0,1	0	0	1	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_W9_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	-	-
	case 'R':		-	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		-	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	uc_chg_batt_date	0	0	0	0
	uc_chg_pump_date	0	0	0	0
	uc_chg_sen_date	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	uc_chg_batt_date	0	0	0	0
	uc_chg_pump_date	0	0	0	0
	uc_chg_sen_date	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_WU_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6
Judgment	OK

			case1	case2	case3
Root	if(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0] == 'W')		○	○	○
	if(pucp_io_buff[lus_r_pos] == '1')		○	-	-
	else		○	-	-
	else		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	87	32
	pucp_io_buff[10]	48,49	49	48	48
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_X8_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_disp_pump_off == M_OFF)		○	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_disp_pump_off	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_disp_pump_off	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_XA_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	pucp_io_buff[10]	0	0	0	0
	uc_alarm_silence	0	0	0	0
	m_Chk_AsciiOnOffDec()	0	0	0	0
	m_Chk_DecOnOffAscii()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	0	0	0	0
	uc_alarm_silence	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_XB_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	10

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○	○	○	○	○	○
	case 'W':		○	○	○	○	○	○	○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '6')		○	-	-	-	-	-	-	○	-	-
	else if(pucp_io_buff[lus_r_pos] == '5')		-	○	-	-	-	-	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '4')		-	-	○	-	-	-	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '3')		-	-	-	○	-	-	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '2')		-	-	-	-	○	-	-	-	-	-
	else if(pucp_io_buff[lus_r_pos] == '1')		-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-
	if(luc_flg != stp_data->uc_beep_select)		○	○	○	○	○	○	○	-	-	-
	case 'R':		○	○	○	○	○	○	○	○	○	-
	if(stp_data->uc_beep_select == 6)		○	-	-	-	-	-	-	-	-	-
	else if(stp_data->uc_beep_select == 5)		-	○	-	-	-	-	-	-	-	-
	else if(stp_data->uc_beep_select == 4)		-	-	○	-	-	-	-	-	-	-
	else if(stp_data->uc_beep_select == 3)		-	-	-	○	-	-	-	-	-	-
	else if(stp_data->uc_beep_select == 2)		-	-	-	-	○		-	-	-	-
	else if(stp_data->uc_beep_select == 1)		-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	○	○	-
	default:		-	-	-	-	-	-	-	-	-	○
	Symbol	Range										
Input	pucp_io_buff[8]	32~87	87	87	87	87	87	87	87	87	82	32
	pucp_io_buff[10]	0~54	54	53	52	51	50	49	48	54	0	0
	uc_beep_select	0~6	6	5	4	3	2	1	0	0	0	0
	uc_beep_time	0	0	0	0	0	0	0	0	0	0	0
	Tool_Hex_Asc()	0	0	0	0	0	0	0	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0	0	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0	0	0	0	0	0	0
	uc_beep_select	0~6	6	5	4	3	2	1	0	6	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

m_cmd_XC_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	case 'C':		-	○	○	-
	if (LoggerGXForce_Chk_LoggerDataClear() == 0)		-	○	-	-
	else		-	-	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	0~87	87	67	67	32
	*pucp_io_buff	1	1	1	1	1
	LoggerGXForce_Chk_LoggerDataClear()	0	0	0	1	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_XH_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2

Judgment	OK

			case1	case2
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○
	case 'W':		○	-
	default:		-	○
	Symbol	Range		
Input	pucp_io_buff[8]	32,87	87	32
	*pucp_io_buff	1	1	1
	m_cmd_ReponsTerm()	0	0	0
Output	*pucp_io_buff	1	1	1
	m_cmd_ResponsTerm()	0	0	0
Judgment			OK	OK

m_cmd_XI_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_id_disp_onoff == M_OFF)		-	○	-	-
	else		○	-	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_id_disp_onoff	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_id_disp_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_XL_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_lunch_onoff == M_OFF)		-	○	-	-
	else		○	-	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_lunch_onoff	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_lunch_onoff	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_XP_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	us_interval_time	0	0	0	0
	Tool_Asc_Dec()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	us_interval_time	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_XS_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 2) == 0)		○	-	-	-	-
	else		-	○	-	-	-
	if(HDDEF_STATION_ID_NUM <= luc_idx)		-	-	-	-	-
	else		-	○	-	-	-
	case 'R':		○	○	○	○	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_w_pos], &uc_m_non_data[0], 2) == 0)		○	-	○	-	-
	for(luc_i = 0; luc_i < HDDEF_STATION_ID_SIZE; luc_i++)		○	-	○	-	-
	if(stp_data->uc_st_id[luc_i] == NULL)		○	-	○	-	-
	else		○	-	○	-	-
	else		-	○	-	○	-
	if(HDDEF_STATION_ID_NUM <= luc_idx)		-	-	-	-	-
	else		-	○	-	○	-
	for(luc_i = 0; luc_i < HDDEF_STATION_ID_SIZE; luc_i++)		-	○	-	○	-
	if(stp_st_id_data_addr->uc_st_id[luc_idx][luc_i] == NULL)		-	○	-	○	-
	else		-	-	-	○	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	*pucp_io_buff	1	1	1	1	1	1
	uc_st_id[0]	0,1	1	0	1	0	0
	uc_st_id[0][0]	0,1	0	0	0	1	0
	Tool_Comp_Pnt()	0,1	0	1	0	1	0
	Tool_Hex_Asc()	0	0	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

m_cmd_XT_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○
	case 'W':		○	-	-
	case 'R':		○	○	-
	default:		-	-	○
	Symbol	Range			
Input	pucp_io_buff[8]	32~87	87	82	32
	*pucp_io_buff	1	1	1	1
	us_back_light_time	0,1	1	0	0
	Tool_Hex_Asc()	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0
Output	*pucp_io_buff	1	1	1	1
	us_back_light_time	0	0	0	0
	m_cmd_ResponsTerm()	0	0	0	0
Judgment			OK	OK	OK

m_cmd_XW_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	○	-	-
	if(pucp_io_buff[lus_r_pos] == '0')		○	-	-	-
	else		-	○	-	-
	case 'R':		○	○	○	-
	if(stp_data->uc_over_write == M_OFF){		○	-	-	-
	else		-	○	○	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	87	82	32
	pucp_io_buff[10]	0~49	48	49	0	0
	uc_over_write	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	0	0	0	0	0
	uc_over_write	0,1	0	1	0	0
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_XY_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4

Judgment	OK

			case1	case2	case3	case4
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○
	case 'W':		○	-	-	-
	case 'C':		-	○	○	-
	if(LoggerGXForce_Chk_LoggerPowerClear() == M_OFF)		-	-	○	-
	else		-	○	-	-
	default:		-	-	-	○
	Symbol	Range				
Input	pucp_io_buff[8]	32~87	87	67	67	32
	*pucp_io_buff	1	1	1	1	1
	LoggerGXForce_Chk_LoggerPowerClear()	0,1	1	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0
Judgment			OK	OK	OK	OK

m_cmd_XZ_proc

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	switch(((IrDA_COMMAND_DATA *)pucp_io_buff)->sub_command[0])		○	○	○	○	○
	case 'W':		○	○	-	-	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_r_pos], &uc_m_non_data[0], 2) == 0)		○	-	-	-	-
	else		-	○	-	-	-
	if(HDDEF_USER_ID_NUM <= luc_idx)		-	-	-	-	-
	else		-	○	-	-	-
	case 'R':		○	○	○	○	-
	if(Tool_Comp_Pnt(&pucp_io_buff[lus_w_pos], &uc_m_non_data[0], 2) == 0)		○	-	○	-	-
	for(luc_i = 0; luc_i < HDDEF_USER_ID_SIZE; luc_i++)		○	-	○	-	-
	if(stp_data->uc_ur_id[luc_i] == NULL)		○	-	○	-	-
	else		○	-	○	-	-
	else		-	○	-	○	-
	if(HDDEF_USER_ID_NUM <= luc_idx)		-	-	-	-	-
	else		-	○	-	○	-
	for(luc_i = 0; luc_i < HDDEF_USER_ID_SIZE; luc_i++)		-	○	-	○	-
	if(stp_user_id_data_addr->uc_ur_id[luc_idx][luc_i] == NULL)		-	○	-	○	-
	else		-	-	-	○	-
	default:		-	-	-	-	○
	Symbol	Range					
Input	pucp_io_buff[8]	32~87	87	87	82	82	32
	*pucp_io_buff	1	1	1	1	1	1
	uc_ur_id[0]	0,1	1	0	1	0	0
	uc_ur_id[0][0]	0,1	0	0	0	1	0
	Tool_Comp_Pnt()	0,1	0	1	0	1	0
	Tool_Hex_Asc()	0	0	0	0	0	0
	m_cmd_ReponsTerm()	0	0	0	0	0	0
Output	*pucp_io_buff	1	1	1	1	1	1
	m_cmd_ResponsTerm()	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK

str_set_uw_hex

	Responsible	Last update
Coding	Ikarashi	2021/7/6
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	8

Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8
Root	if((col < 1U) || ((U_CHR)sizeof(buff) < col))		○	○	○	○	○	○	○	○
	else		-	-	-	-	-	-	-	-
	for(i = 0; i < (S_SHORT)col; i++)		-	○	○	-	○	○	-	○
	if(wk < 10U)		-	○	○	-	○	-	-	○
	else		-	-	-	-	-	○	-	○
	for(i = 0; i < (S_SHORT)col; i++)		-	○	○	○	○	○	-	○
	Symbol	Range								
Input	val	9,10	9	9	9	9	9	10	10	10
	col	0~9	0	1	8	9	2	1	0	2
Output	str[0]	0~65	0	57	48	0	48	65	0	48
	ret	0~8	0	1	8	0	2	1	0	2
Judgment			OK	OK	OK	OK	OK	OK	OK	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	vibration.c

VIBRATION_Control_VIB_Port

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3

Judgment	OK

			case1	case2	case3
Root	if(puc_on == VIBRATION_SET_ON)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	puc_on	0~2	0	1	2
Output	PORTC.PODR.BIT.B6	0,1	0	1	0
Judgment			OK	NG	OK

Unit Test Specification(with Results)

Project name	GX-Force
File name	zipc_num_set.c

m_Get_SensorFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	4
Judgment	OK

			case1	case2	case3	case4
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	○	○
	if(!((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_INS))		-	○	○	-
	if((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_SEN)		-	-	○	-
	else		-	○	-	-
	else		-	-	-	○
	else		○	-	-	-
	Symbol	Range				
Input	(stp_GasInfo_addr+luc_i)->uc_gas_comb	0~30	0	29	29	30
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x00	0x00	0x04	0xFF
Output	luc_ans	0,1	0	0	1	0
Judgment			OK	OK	OK	OK

ZIPCNumSet_Chk_AirFail

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	2
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_ZRO)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	(stp_denst + luc_i)->uc_state_trb	0x00~0xFF	0x01	0x00	0xFF
Output	luc_ans	0,1	1	0	1
Judgment			OK	OK	OK

ZIPCNumSet_Chk_AutoCal_EscNum

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(st_zipu_num.ul_item_select_num == HDDEF_GAS_MAX)		○	-	-
	else		-	○	○
	Symbol	Range			
Input	st_zipu_num.ul_item_select_num	0x00000000~0xFFFFFFFF	0x00000004	0x00000000	0xFFFFFFFF
Output	luc_ans	0,1	1	0	0
Judgment			OK	OK	OK

ZIPCNumSet_Chk_DemandZero_onoff

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_demand_zero_onoff	0	0
Judgment			OK

ZIPCNumSet_Chk_SensorAllInitFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00 && !((stp_denst+luc_i)->uc_state_trb & DataStructConst_S_FAIL_INS))		○	-	-
	else		-	○	○
	if(stp_status->us_main_state & DataStructConst_FAIL_SUB)		-	○	○
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	29	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00,0xFF	0x00	0xFF	0xFF
	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0XFFFF	0XFFFF
Output	luc_ans	0	0	0	0
Judgment			OK	OK	OK

ZIPCNumSet_Chk_SpanCalFail

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_cal_group == GasFuncAutoCal_Get_AutoCalGroup())		○	○	-
	if((stp_denst + luc_i)->uc_state_trb & DataStructConst_S_FAIL_SPN)		○	-	-
	else		-	○	-
	else		-	-	○
	Symbol	Range			
Input	GasFuncAutoCal_Get_AutoCalGroup()	0x00,0xFF	0x00	0xFF	0x00
	stp_GasInfo_addr+luc_i)->uc_cal_group	0x00,0xFF	0x00	0x00	0xFF
	(stp_denst + luc_i)->uc_state_trb	0x00~0xFF	0x02	0x00	0xFF
Output	luc_ans	0	0	0	0
Judgment			OK	OK	OK

ZIPCNumSet_Get_AlarmFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr+luc_i)->uc_gas_comb != 0x00)		-	○	○
	if((stp_denst+luc_i)->uc_alarm_lat)		-	-	○
	break		-	-	○
	else		-	○	-
	else		○	-	-
	Symbol	Range			
Input	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	0	29	30
	(stp_denst+luc_i)->uc_alarm_lat	0x00,0xFF	0x00	0x00	0xFF
Output	luc_ans	0	0	0	0
Judgment			OK	OK	OK

ZIPCNumSet_Get_AlarmManDownFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_status->uc_man_down != 0) || (ZIPCNumSet_Get_AlarmFlg() == M_ON) || (FuncTempWarning_Get_TempBuzzerFlg() == M_ON))		-	○	-
	else		○	-	○
	Symbol	Range			
Input	stp_status->uc_man_down	0x00~0x20	0x00	0x1F	0x20
	ZIPCNumSet_Get_AlarmFlg()	0~2	0	1	2
	FuncTempWarning_Get_TempBuzzerFlg()	0~2	0	1	2
Output	luc_ans	0,1	0	1	0
Judgment			OK	OK	OK

ZIPCNumSet_Get_AlarmPattern

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_alarm_pattern	0	0
Judgment			OK

ZIPCNumSet_Get_AutoZeroFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_auto_zero_onoff)	0	0
Judgment			OK

ZIPCNumSet_Get_DispFlicker2

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_disp_flicker2sec2	0	0
Judgment			OK

ZIPCNumSet_Get_DispFlicker3

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_disp_flicker2sec3	0	0
Judgment			OK

ZIPCNumSet_Get_DispFlicker4

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_disp_flicker2sec4	0	0
Judgment			OK

ZIPCNumSet_Get_GasTestFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	uc_m_gastest_mode	0	0
Judgment			OK

ZIPCNumSet_Get_NotManDownFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if((stp_status->us_main_state & DataStructConst_FAIL_SIS) || m_Get_SensorFailFlg())		-	○	○
	else		○	-	-
	Symbol	Range			
Input	stp_status->us_main_state	0x0000~0xFFFF	0x0000	0x0001	0xFFFF
	m_Get_SensorFailFlg()	0~2	0	1	2
Output	luc_ans	0,1	0	1	1
Judgment			OK	OK	OK

ZIPCNumSet_Get_SensorInitFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	if(stp_status->us_main_state & DataStructConst_FAIL_SUB)		○	-	○
	else		-	○	-
	for(luc_i=0; luc_i<HDDEF_GAS_MAX; luc_i++)		○	○	○
	if((stp_GasInfo_addr + luc_i)->uc_gas_comb != 0x00 && (stp_denst+luc_i)-> uc_state_trb & DataStructConst_S_FAIL_INS)		○	-	○
	else		-	○	-
	Symbol	Range			
Input	stp_status->us_main_state	0x0000~0xFFFF	0x0200	0x0000	0xFFFF
	(stp_GasInfo_addr + luc_i)->uc_gas_comb	0~30	29	0	30
	(stp_denst+luc_i)->uc_state_trb	0x00~0xFF	0x10	0x00	0xFF
Output	luc_ans	0,1	1	0	1
Judgment			OK	OK	OK

ZIPCNumSet_Get_ZipcModeNum

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	6

Judgment	OK

			case1	case2	case3	case4	case5	case6
Root	if(puc_type == 0)		○	-	-	-	-	-
	else if(puc_type == 1)		-	○	-	-	-	-
	else if(puc_type == 2)		-	-	○	-	-	-
	else if(puc_type == 3)		-	-	-	○	-	-
	else if(puc_type == 4)		-	-	-	-	○	-
	else		-	-	-	-	-	○
	Symbol	Range						
Input	puc_type	0~5	0	1	2	3	4	5
	st_zipu_num.ul_mode_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	st_zipu_num.ul_sub_mode_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	st_zipu_num.ul_sub2_mode_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	st_zipu_num.ul_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
	st_zipu_num.ul_sub_item_select_num	0x00000000,0xFFFFFFFF	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0xFFFFFFFF
Output	luc_ans	0	0	0	0	0	0	0
Judgment			OK	OK	OK	OK	OK	OK

ZIPCNumSet_Get_ZipcNumAddress

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	None	-	-
Judgment			OK

ZIPCNumSet_Int_1sec

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	5

Judgment	OK

			case1	case2	case3	case4	case5
Root	if(uc_num == M_ON)		○	○	○	-	-
	if(++uc_disp_flicker2sec2 == 2)		-	○	-	-	-
	else		○	-	○	-	-
	if(++uc_disp_flicker2sec3 == 3)		-	○	-	-	-
	else		○	-	○	-	-
	if(++uc_disp_flicker2sec4 == 4)		-	○	-	-	-
	else		○	-	○	-	-
	else		-	-	-	○	○
	Symbol	Range					
Input	uc_num	0~2	1	1	1	0	2
	uc_disp_flicker2sec2	0~2	0	1	2	0	1
	uc_disp_flicker2sec3	0~3	0	2	3	0	2
	uc_disp_flicker2sec4	0~4	0	3	4	0	3
Output	uc_disp_flicker2sec2	0,1	1	0	0	0	1
	uc_disp_flicker2sec3	0,2	1	0	0	0	2
	uc_disp_flicker2sec4	0~3	1	0	0	0	3
Judgment			OK	OK	OK	OK	OK

ZIPCNumSet_mainte_pass_onoff

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1

Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_mainte_pass_onoff	0	0
Judgment			OK

ZIPCNumSet_pass_protect_onoff

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_protect_pword_onoff	0	0
Judgment			OK

ZIPCNumSet_Reset_InitPossibleResetFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_status->us_main_state	0	0
Judgment			OK

ZIPCNumSet_Reset_PossibleResetFailFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_status->us_main_state	0	0
Judgment			OK

ZIPCNumSet_Set_GasTestFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	3
Judgment	OK

			case1	case2	case3
Root	None		-	-	-
	Symbol	Range			
Input	puc_flg	0~2	1	0	2
Output	uc_m_gastest_mode	0~2	1	0	2
Judgment			OK	OK	OK

ZIPCNumSet_user_pass_onoff

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	1
Judgment	OK

			case1
Root	None		-
	Symbol	Range	
Input	None	-	-
Output	stp_data->uc_user_pass_onoff	0	0
Judgment			OK

ZIPCNumSet_Get_FailFlg

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	12
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11	case12
Root	if(puc_type == 0)		○	○	-	-	-	-	-	-	-	-	-	-
	if((stp_status->us_main_state & DataStructConst_FAIL_ALL) || m_Get_SensorFailFlg())		-	○	-	-	-	-	-	-	-	-	-	-
	else		○	-	-	-	-	-	-	-	-	-	-	-
	else if(puc_type == 1)		-	-	○	○	-	-	-	-	-	-	-	-
	if(stp_status->us_main_state & (DataStructConst_FAIL_BTA | DataStructConst_FAIL_BTL))		-	-	-	○	-	-	-	-	-	-	-	-
	else		-	-	○	-	-	-	-	-	-	-	-	-
	else if(puc_type == 2)		-	-	-	-	○	○	-	-	-	-	-	-
	if(stp_status->us_main_state & (DataStructConst_FAIL_RAM | DataStructConst_FAIL_ROM | DataStructConst_FAIL_BTA | DataStructConst_FAIL_BTL | DataStructConst_FAIL_CLK | DataStructConst_FAIL_BKB | DataStructConst_FAIL_FRM | DataStructConst_FAIL_FLH))		-	-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	○	-	-	-	-	-	-	-
	else if(puc_type == 3)		-	-	-	-	-	-	○	○	-	-	-	-
	if(stp_status->us_main_state & (DataStructConst_FAIL_FLH | DataStructConst_FAIL_CLK | DataStructConst_FAIL_BKB | DataStructConst_FAIL_ACC | DataStructConst_FAIL_BLE))		-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-	-	-
	if(stp_status->us_main_state & (DataStructConst_FAIL_MDN | DataStructConst_FAIL_MDP))		-	-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	○	-	-	-	-	-
	else if(puc_type == 4)		-	-	-	-	-	-	-	-	○	○	-	-
	if(stp_status->us_main_state & (DataStructConst_FAIL_MDN | DataStructConst_FAIL_MDP))		-	-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	○	-	-	-
	else		-	-	-	-	-	-	-	-	-	-	○	○
	Symbol	Range												
Input	puc_type	0~6	0	0	1	1	2	2	3	3	4	5	5	6
	stp_status->us_main_state	0x0000,0xFFFF	0x0000	0xFFFF	0x0000	0xFFFF	0x0000	0xFFFF	0x0000	0xFFFF	0x0000	0xFFFF	0x0000	0x0000
	m_Get_SensorFailFlg()	0~2	0	1	0	0	0	0	0	0	0	0	0	2
Output	luc_ans	0,1	0	1	0	1	0	1	0	1	0	1	0	0
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

ZIPCNumSet_Select_ZipcNumUpDown

	Responsible	Last update
Coding	Ikarashi	2021/7/7
Test	Taki	
Judgment	Ikarashi	

Infinite loop	None
Verification pass	11
Judgment	OK

			case1	case2	case3	case4	case5	case6	case7	case8	case9	case10	case11
Root	if(puc_type == 0)		○	○	-	-	-	-	-	-	-	-	-
	if(st_zipu_num.uc_mode_select_updown == M_ON)		○	-	-	-	-	-	-	-	-	-	-
	else		-	○	-	-	-	-	-	-	-	-	-
	else if(puc_type == 1)		-	-	○	○	-	-	-	-	-	-	-
	if(st_zipu_num.uc_sub_mode_select_updown == M_ON)		-	-	○	-	-	-	-	-	-	-	-
	else		-	-	-	○	-	-	-	-	-	-	-
	else if(puc_type == 2)		-	-	-	-	○	○	-	-	-	-	-
	if(st_zipu_num.uc_sub2_mode_select_updown == M_ON)		-	-	-	-	○	-	-	-	-	-	-
	else		-	-	-	-	-	○	-	-	-	-	-
	else if(puc_type == 3)		-	-	-	-	-	-	○	○	-	-	-
	if(st_zipu_num.uc_item_select_updown == M_ON)		-	-	-	-	-	-	○	-	-	-	-
	else		-	-	-	-	-	-	-	○	-	-	-
	else if(puc_type == 4)		-	-	-	-	-	-	-	-	○	○	-
	if(st_zipu_num.uc_sub_item_select_updown == M_ON)		-	-	-	-	-	-	-	-	○	-	-
	else		-	-	-	-	-	-	-	-	-	○	-
	else		-	-	-	-	-	-	-	-	-	-	○
	Symbol	Range											
Input	puc_type	0~5	0	0	1	1	2	2	3	3	4	4	5
	st_zipu_num.uc_mode_select_updown	0~2	1	0	2	0	0	0	0	0	0	0	0
	st_zipu_num.uc_sub_mode_select_updown	0~2	2	0	1	0	0	0	0	0	0	0	0
	st_zipu_num.uc_sub2_mode_select_updown	0~2	2	0	0	0	1	0	0	0	0	0	0
	st_zipu_num.uc_item_select_updown	0~2	2	0	0	0	0	0	1	0	0	0	0
	st_zipu_num.uc_sub_item_select_updown	0~2	0	0	0	0	0	0	0	0	1	0	2
Output	None	-	-	-	-	-	-	-	-	-	-	-	-
Judgment			OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK