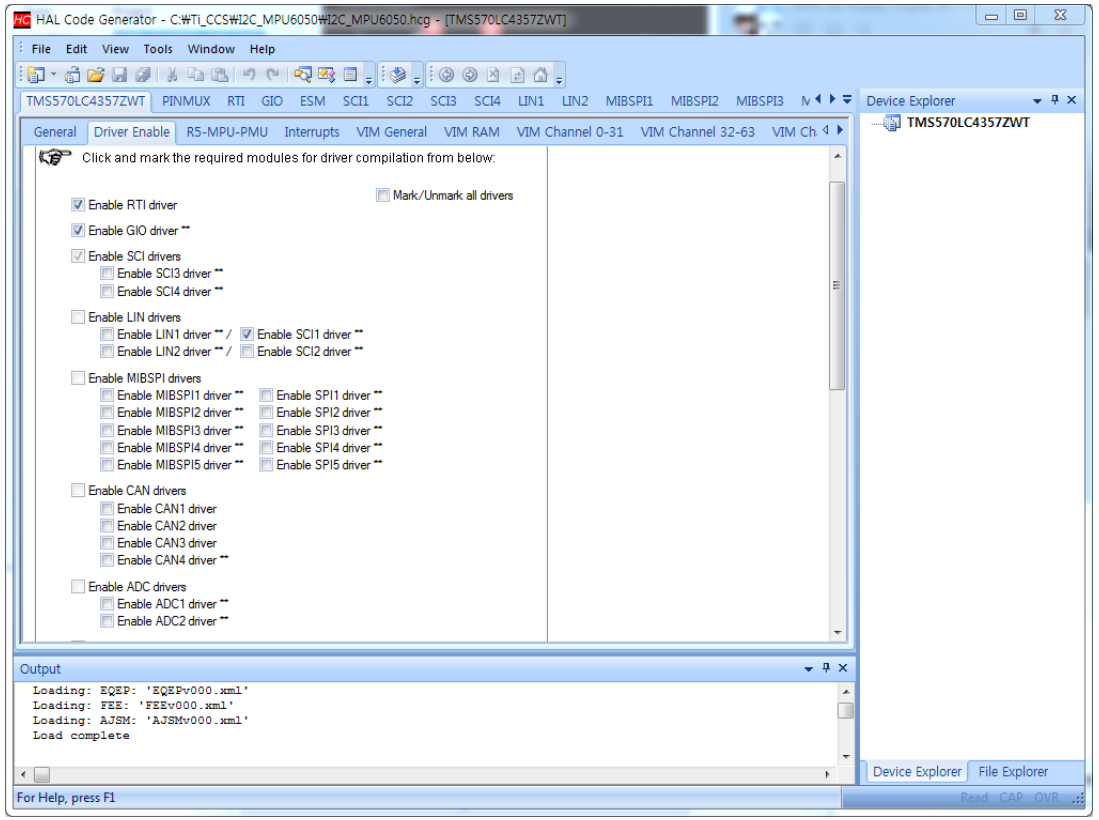
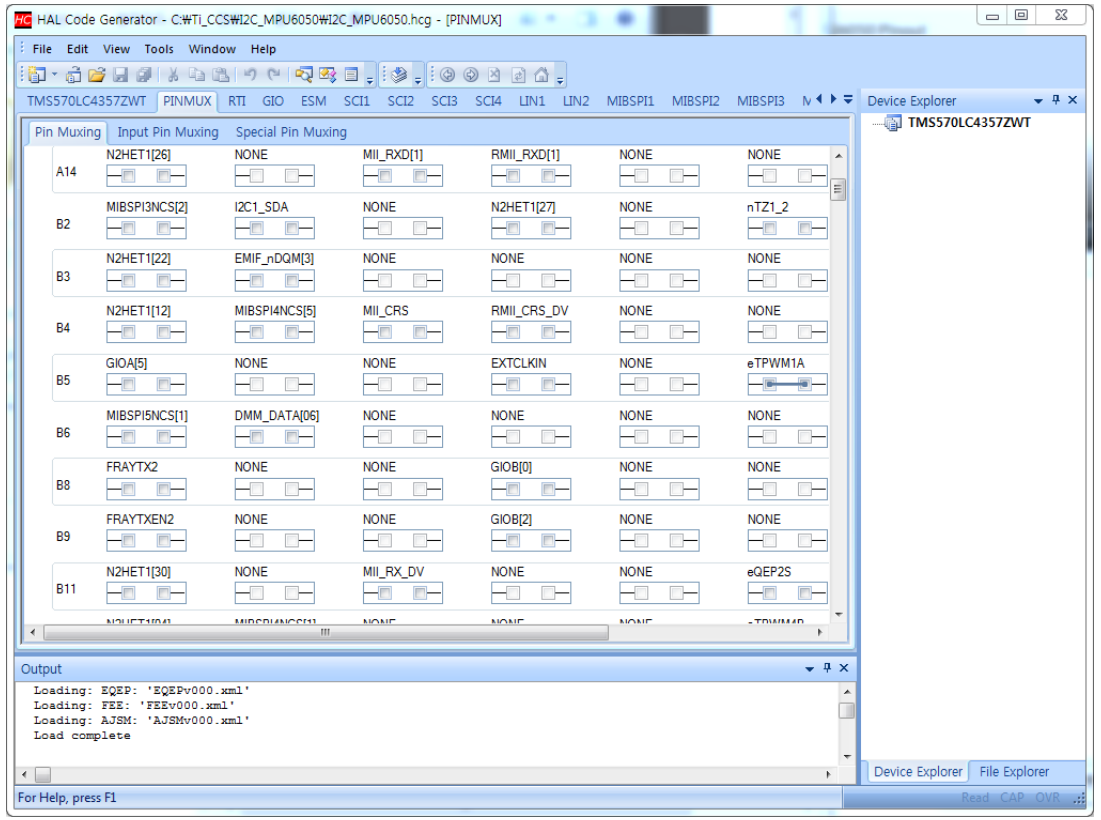
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

Innova Lee(이상훈) 강사
gcccompil3r@gmail.com

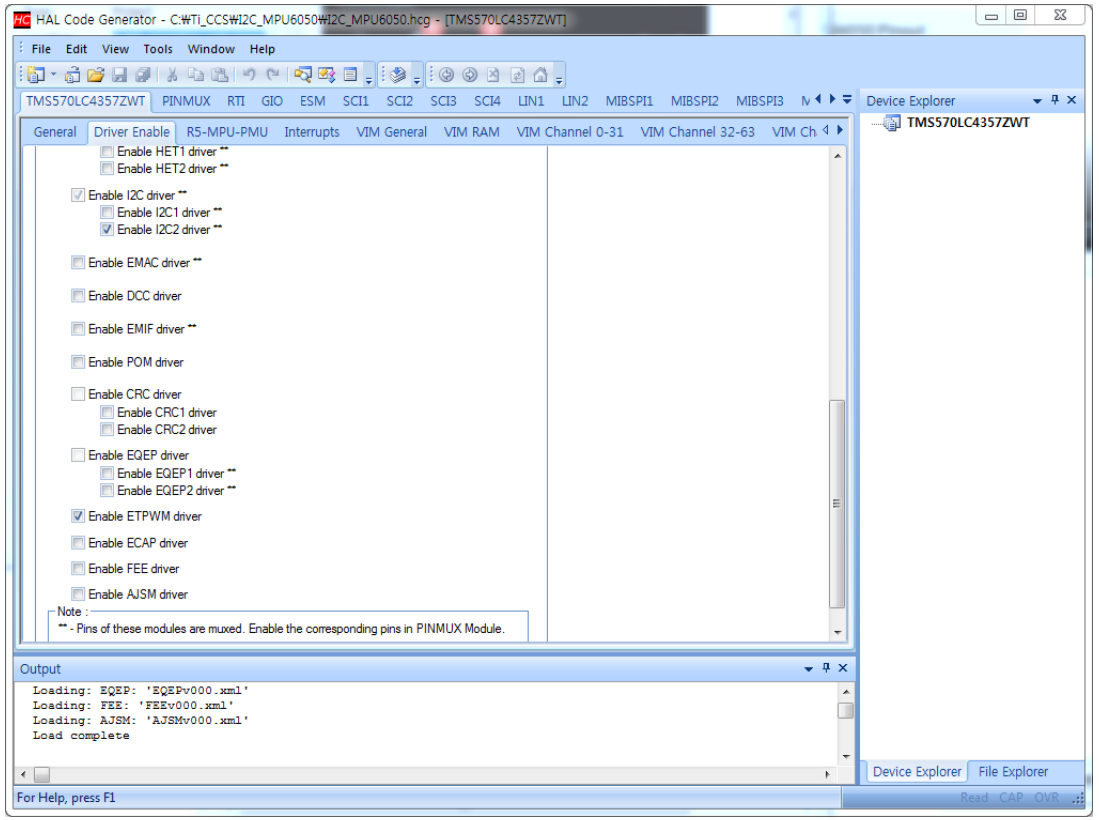
은태영 학생
zero_bird@naver.com

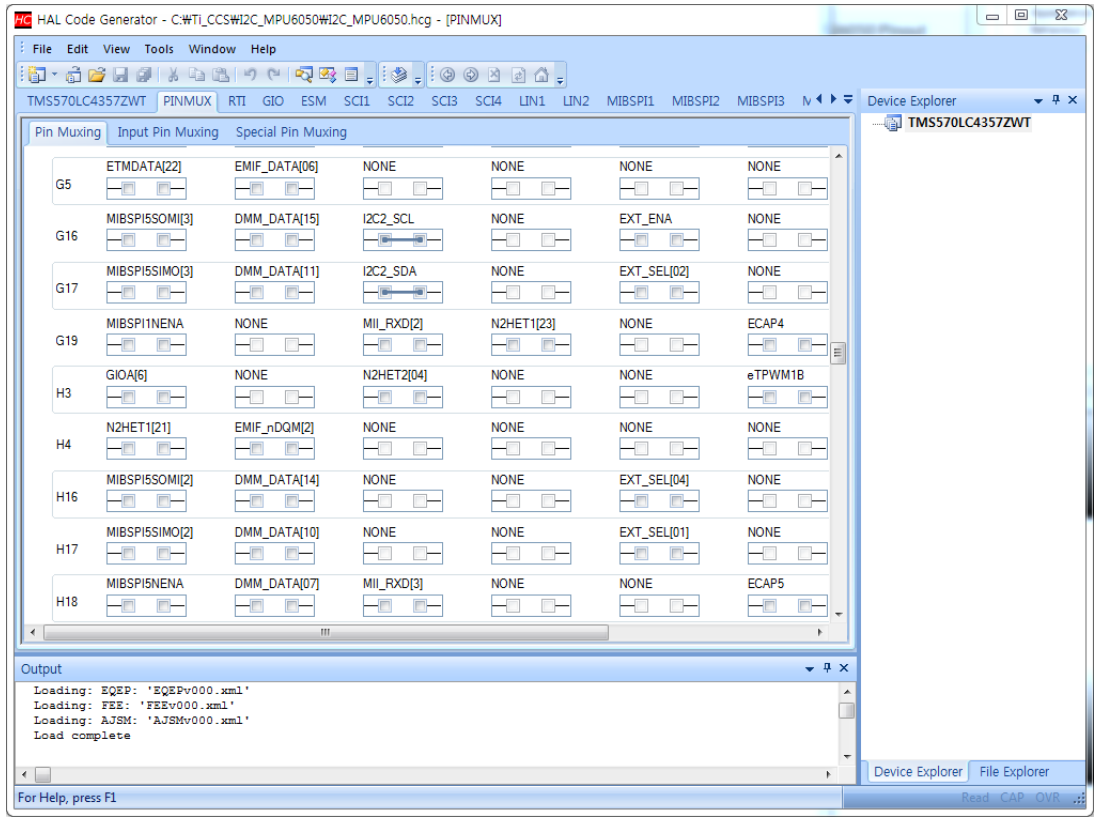
CCS


HAL Code Generator - C:¥Ti_CCS¥I2C_MPU6050¥I2C_MPU6050.hcg - [TMS570LC4357ZWT]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3
General Driver Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31 VIM Channel 32-63
Click and mark the required modules for driver compilation from below:
Mark/Unmark all drivers
Enable RTI driver
Enable GIO driver **
Enable SCI drivers
Enable SCI3 driver **
Enable SCI4 driver **
Enable LIN drivers
Enable LIN1 driver ** / Enable SCI1 driver **
Enable LIN2 driver ** / Enable SCI2 driver **
Enable MIBSPI drivers
Enable MIBSPI1 driver ** Enable SPI1 driver **
Enable MIBSPI2 driver ** Enable SPI2 driver **
Enable MIBSPI3 driver ** Enable SPI3 driver **
Enable MIBSPI4 driver ** Enable SPI4 driver **
Enable MIBSPI5 driver ** Enable SPI5 driver **
Enable CAN drivers
Enable CAN1 driver
Enable CAN2 driver
Enable CAN3 driver
Enable CAN4 driver **
Enable ADC drivers
Enable ADC1 driver **
Enable ADC2 driver **
Device Explorer
TMS570LC4357ZWT
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer File Explorer
For Help, press F1

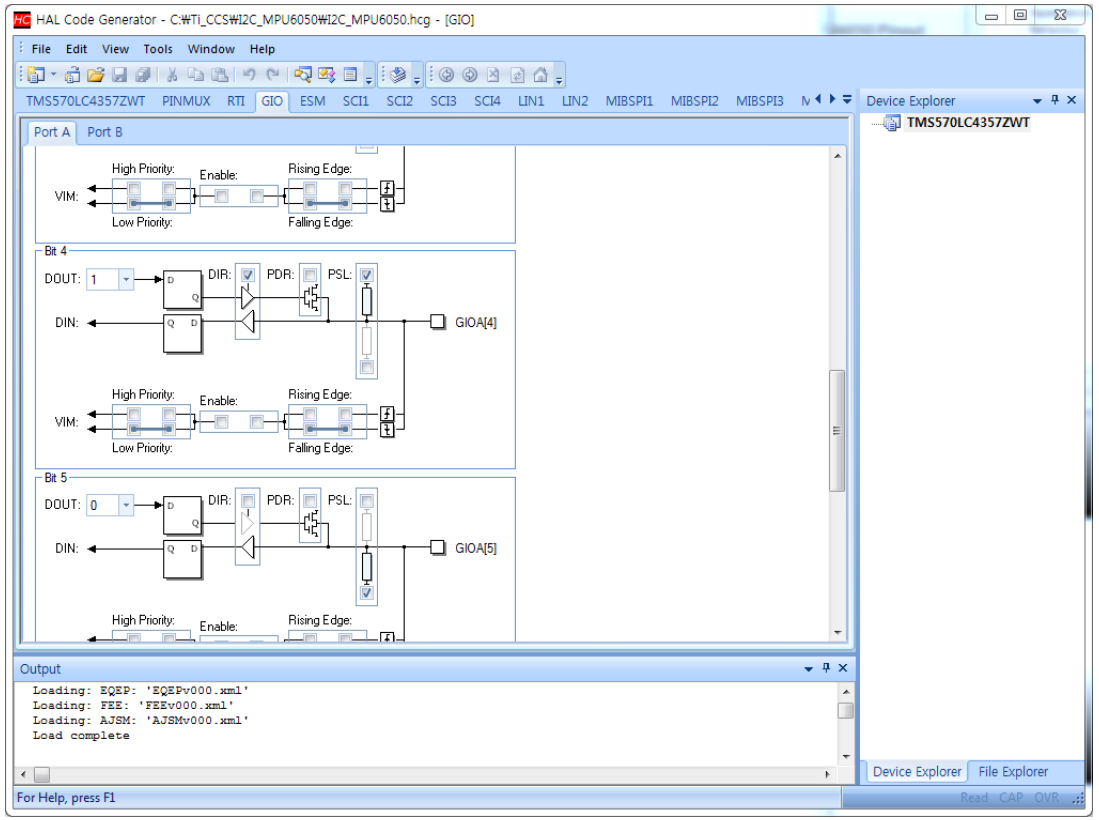

HAL Code Generator - C:¥Ti_CCS¥I2C_MPU6050¥I2C_MPU6050.hcg - [PINMUX]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3
Pin Muxing Input Pin Muxing Special Pin Muxing
A14 N2HET1[26] NONE MII_RXD[1] RMII_RXD[1] NONE NONE
B2 MIBSPI3NCS[2] I2C1_SDA NONE N2HET1[27] NONE nTZ1_2
B3 N2HET1[22] EMIF_nDQM[3] NONE NONE NONE NONE
B4 N2HET1[12] MIBSPI4NCS[5] MII_CRS RMII_CRS_DV NONE NONE
B5 GIOA[5] NONE NONE EXTCLKIN NONE eTPWM1A
B6 MIBSPI5NCS[1] DMM_DATA[06] NONE NONE NONE NONE
B8 FRAYTX2 NONE NONE GIOB[0] NONE NONE
B9 FRAYTXEN2 NONE NONE GIOB[2] NONE NONE
B11 N2HET1[30] NONE MII_RX_DV NONE NONE eQEP2S
Device Explorer
TMS570LC4357ZWT
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer File Explorer
For Help, press F1

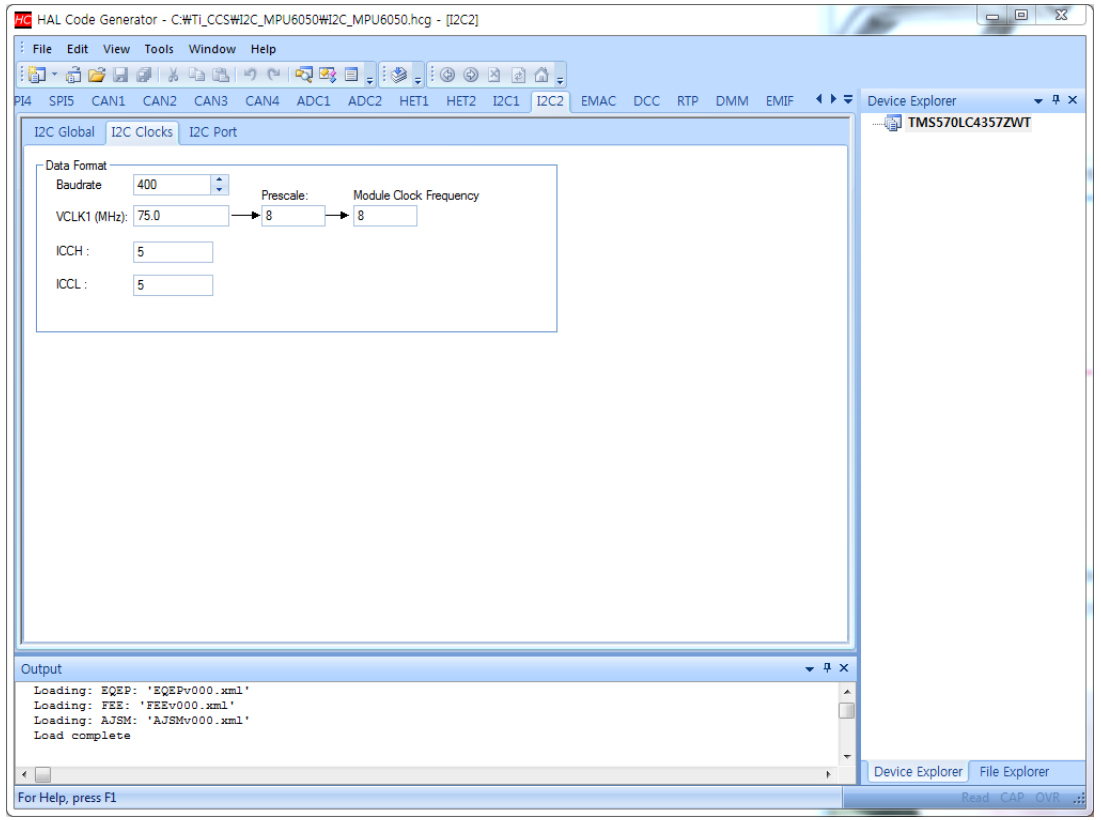
CCS


HAL Code Generator - C:¥Ti_CCS¥I2C_MPU6050¥I2C_MPU6050.hcg - [TMS570LC4357ZWT]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3
General Driver Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31 VIM Channel 32-63
Enable HET1 driver **
Enable HET2 driver **
Enable I2C driver **
Enable I2C1 driver **
Enable I2C2 driver **
Enable EMAC driver **
Enable DCC driver
Enable EMIF driver **
Enable POM driver
Enable CRC driver
Enable CRC1 driver
Enable CRC2 driver
Enable EQEP driver
Enable EQEP1 driver **
Enable EQEP2 driver **
Enable ETPWM driver
Enable ECAP driver
Enable FEE driver
Enable AJSM driver
Note :
** - Pins of these modules are muxed. Enable the corresponding pins in PINMUX Module.
Device Explorer
TMS570LC4357ZWT
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer File Explorer
For Help, press F1

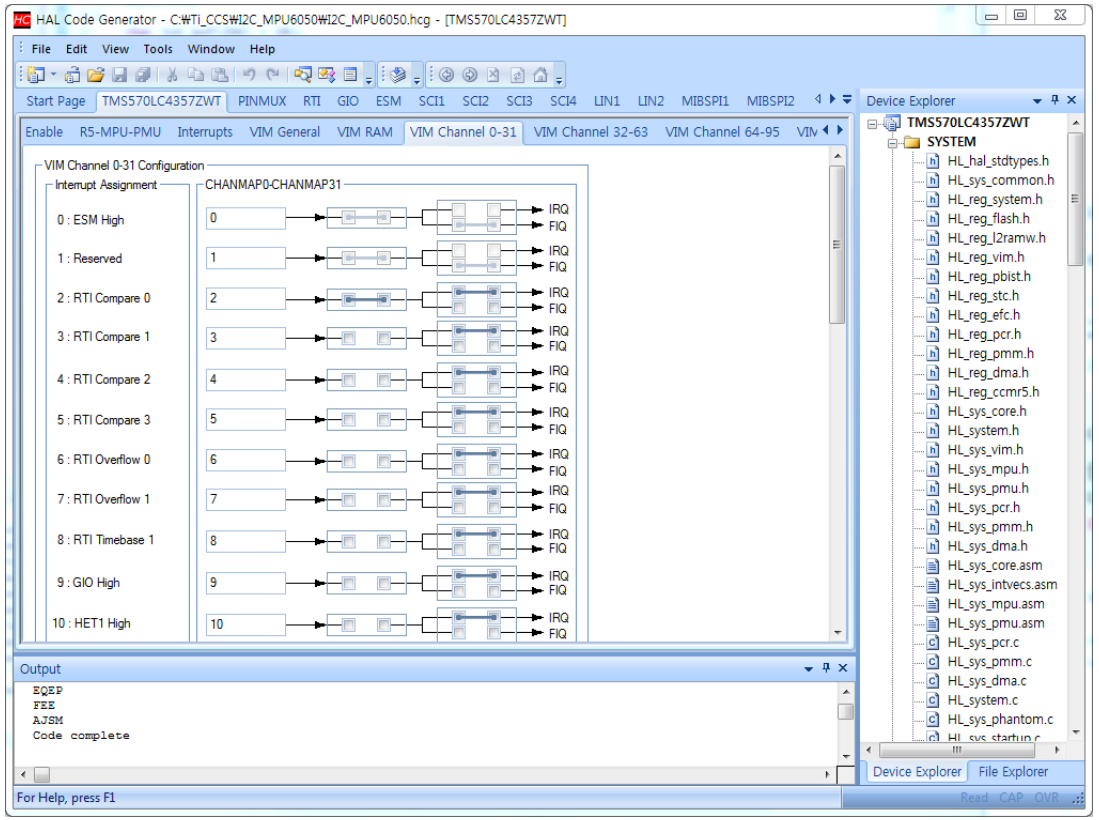

HAL Code Generator - C:¥Ti_CCS¥I2C_MPU6050¥I2C_MPU6050.hcg - [PINMUX]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3
Pin Muxing Input Pin Muxing Special Pin Muxing
G5 ETMDATA[22] EMIF_DATA[06] NONE NONE NONE NONE
G16 MIBSPI5SOMI[3] DMM_DATA[15] I2C2_SCL NONE EXT_ENA NONE
G17 MIBSPI5SIMO[3] DMM_DATA[11] I2C2_SDA NONE EXT_SEL[02] NONE
G19 MIBSPI1NENA NONE MII_RXD[2] N2HET1[23] NONE ECAP4
H3 GIOA[6] NONE N2HET2[04] NONE NONE eTPWM1B
H4 N2HET1[21] EMIF_nDQM[2] NONE NONE NONE NONE
H16 MIBSPI5SOMI[2] DMM_DATA[14] NONE NONE EXT_SEL[04] NONE
H17 MIBSPI5SIMO[2] DMM_DATA[10] NONE NONE EXT_SEL[01] NONE
H18 MIBSPI5NENA DMM_DATA[07] MII_RXD[3] NONE NONE ECAP5
Device Explorer
TMS570LC4357ZWT
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer File Explorer
For Help, press F1

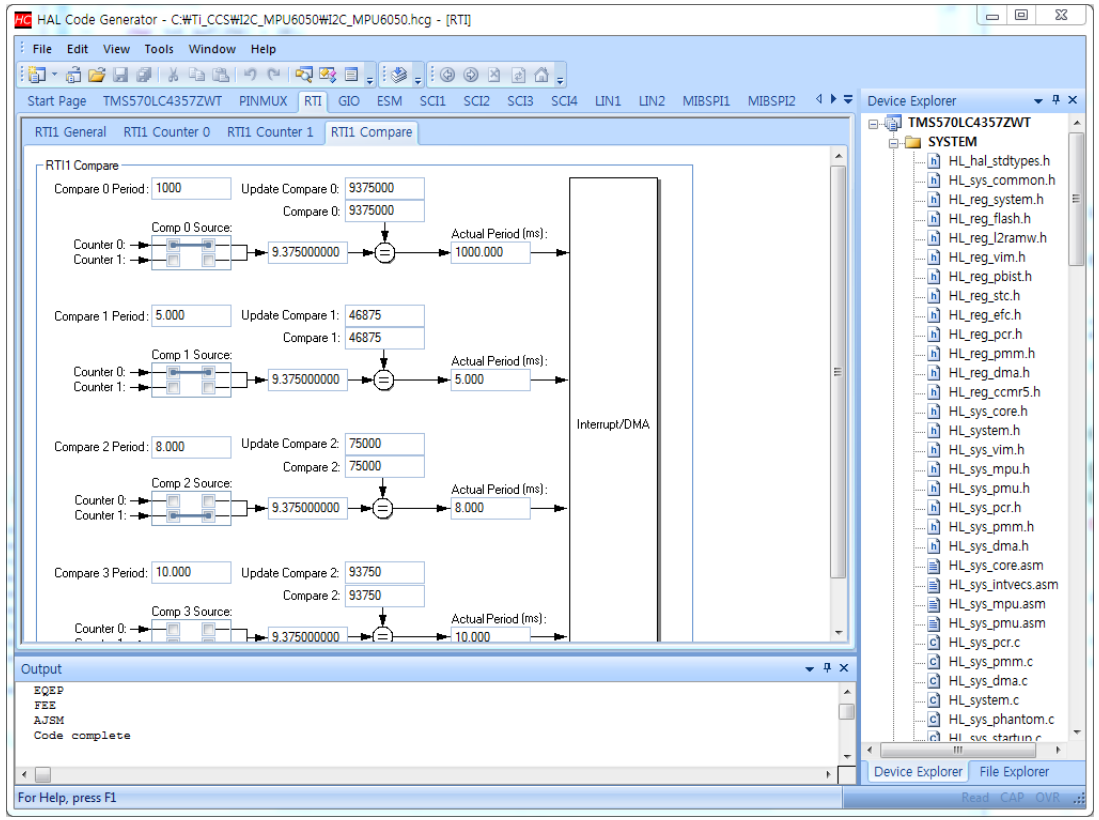
CCS


HAL Code Generator - C:₩Ti_CCS₩I2C_MPU6050₩I2C_MPU6050.hcg - [GIO]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3
Port A Port B
High Priority:
Enable:
Rising Edge:
VIM:
Low Priority:
Falling Edge:
Bit 4
DOUT: 1
DIR:
PDR:
PSL:
DIN:
GIOA[4]
Bit 5
DOUT: 0
GIOA[5]
Device Explorer
TMS570LC4357ZWT
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer File Explorer
For Help, press F1

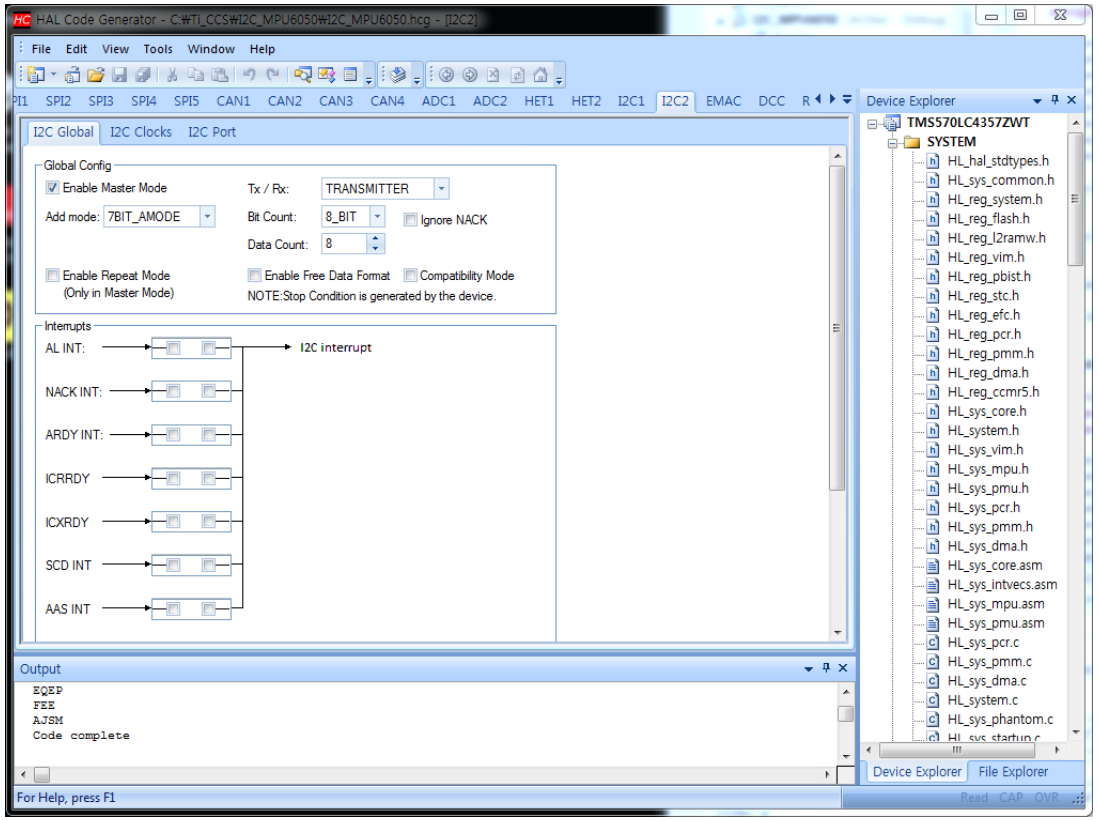

HAL Code Generator - C:₩Ti_CCS₩I2C_MPU6050₩I2C_MPU6050.hcg - [I2C2]
File Edit View Tools Window Help
PI4 SPI5 CAN1 CAN2 CAN3 CAN4 ADC1 ADC2 HET1 HET2 I2C1 I2C2 EMAC DCC RTP DMM EMIF
I2C Global I2C Clocks I2C Port
Data Format
Baudrate 400
Prescale:
Module Clock Frequency
VCLK1 (MHz): 75.0
8
8
ICCH : 5
ICCL : 5
Device Explorer
TMS570LC4357ZWT
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer File Explorer
For Help, press F1

CCS


HAL Code Generator - C:\Ti_CCS\I2C_MPU6050\I2C_MPU6050.hcg - [TMS570LC4357ZWT]
File Edit View Tools Window Help
Start Page TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2
Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31 VIM Channel 32-63 VIM Channel 64-95
VIM Channel 0-31 Configuration
Interrupt Assignment
CHANMAP0-CHANMAP31
0 : ESM High
1 : Reserved
2 : RTI Compare 0
3 : RTI Compare 1
4 : RTI Compare 2
5 : RTI Compare 3
6 : RTI Overflow 0
7 : RTI Overflow 1
8 : RTI Timebase 1
9 : GIO High
10 : HET1 High
IRQ
FIQ
Output
EQEP
FEE
AJSM
Code complete
For Help, press F1
Device Explorer
TMS570LC4357ZWT
SYSTEM
HL_hal_stdtypes.h
HL_sys_common.h
HL_reg_system.h
HL_reg_flash.h
HL_reg_l2ramw.h
HL_reg_vim.h
HL_reg_pbist.h
HL_reg_stc.h
HL_reg_efc.h
HL_reg_pcr.h
HL_reg_pmm.h
HL_reg_dma.h
HL_reg_ccmr5.h
HL_sys_core.h
HL_system.h
HL_sys_vim.h
HL_sys_mpu.h
HL_sys_pmu.h
HL_sys_pcr.h
HL_sys_pmm.h
HL_sys_dma.h
HL_sys_core.asm
HL_sys_intvecs.asm
HL_sys_mpu.asm
HL_sys_pmu.asm
HL_sys_pcr.c
HL_sys_pmm.c
HL_sys_dma.c
HL_system.c
HL_sys_phantom.c
Device Explorer
File Explorer


HAL Code Generator - C:\Ti_CCS\I2C_MPU6050\I2C_MPU6050.hcg - [RTI]
File Edit View Tools Window Help
Start Page TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2
RTI1 General RTI1 Counter 0 RTI1 Counter 1 RTI1 Compare
RTI1 Compare
Compare 0 Period: 1000
Update Compare 0: 9375000
Compare 0: 9375000
Comp 0 Source:
Counter 0:
Counter 1:
9.375000000
Actual Period (ms):
1000.000
Compare 1 Period: 5.000
Update Compare 1: 46875
Compare 1: 46875
Comp 1 Source:
Counter 0:
Counter 1:
9.375000000
Actual Period (ms):
5.000
Compare 2 Period: 8.000
Update Compare 2: 75000
Compare 2: 75000
Comp 2 Source:
Counter 0:
Counter 1:
9.375000000
Actual Period (ms):
8.000
Compare 3 Period: 10.000
Update Compare 2: 93750
Compare 2: 93750
Comp 3 Source:
Counter 0:
9.375000000
Actual Period (ms):
10.000
Interrupt/DMA
Output
EQEP
FEE
AJSM
Code complete
For Help, press F1
Device Explorer
TMS570LC4357ZWT
SYSTEM
HL_hal_stdtypes.h
HL_sys_common.h
HL_reg_system.h
HL_reg_flash.h
HL_reg_l2ramw.h
HL_reg_vim.h
HL_reg_pbist.h
HL_reg_stc.h
HL_reg_efc.h
HL_reg_pcr.h
HL_reg_pmm.h
HL_reg_dma.h
HL_reg_ccmr5.h
HL_sys_core.h
HL_system.h
HL_sys_vim.h
HL_sys_mpu.h
HL_sys_pmu.h
HL_sys_pcr.h
HL_sys_pmm.h
HL_sys_dma.h
HL_sys_core.asm
HL_sys_intvecs.asm
HL_sys_mpu.asm
HL_sys_pmu.asm
HL_sys_pcr.c
HL_sys_pmm.c
HL_sys_dma.c
HL_system.c
HL_sys_phantom.c
Device Explorer
File Explorer

CCS


HAL Code Generator - C:\TI_CCS\I2C_MPU6050\I2C_MPU6050.hcg - [I2C2]
File Edit View Tools Window Help
SPI2 SPI3 SPI4 SPI5 CAN1 CAN2 CAN3 CAN4 ADC1 ADC2 HET1 HET2 I2C1 I2C2 EMAC DCC
I2C Global I2C Clocks I2C Port
Global Config
Enable Master Mode
Tx / Rx: TRANSMITTER
Add mode: 7BIT_AMODE
Bit Count: 8_BIT
Ignore NACK
Data Count: 8
Enable Repeat Mode
(Only in Master Mode)
Enable Free Data Format
Compatibility Mode
NOTE:Stop Condition is generated by the device.
Interrupts
AL INT:
I2C interrupt
NACK INT:
ARDY INT:
ICRRDY
ICXRDY
SCD INT
AAS INT
Output
EQEP
FEE
AJSM
Code complete
For Help, press F1
Device Explorer
TMS570LC4357ZWT
SYSTEM
HL_hal_stdtypes.h
HL_sys_common.h
HL_reg_system.h
HL_reg_flash.h
HL_reg_l2ramw.h
HL_reg_vim.h
HL_reg_pbist.h
HL_reg_stc.h
HL_reg_efc.h
HL_reg_pcr.h
HL_reg_pmm.h
HL_reg_dma.h
HL_reg_ccmr5.h
HL_sys_core.h
HL_system.h
HL_sys_vim.h
HL_sys_mpu.h
HL_sys_pmu.h
HL_sys_pcr.h
HL_sys_pmm.h
HL_sys_dma.h
HL_sys_core.asm
HL_sys_intvecs.asm
HL_sys_mpu.asm
HL_sys_pmu.asm
HL_sys_pcr.c
HL_sys_pmm.c
HL_sys_dma.c
HL_system.c
HL_sys_phantom.c
Device Explorer
File Explorer

```
#include <string.h>
#include <stdio.h>

#include "HL_sys_common.h"
#include "HL_sys_core.h"
#include "HL_sci.h"
#include "HL_gio.h"
#include "HL_i2c.h"
#include "HL_rti.h"

#define UART           sciREG1
#define MPU6050_ADDR    0x68

void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len);
void mpu6050_enable(void);
void mpu6050_acc_config(void);
```

```
volatile char g_acc_xyz[6];
volatile int g_acc_flag;

void wait(uint32 t)
{
    t--;
}

int main(void)
{
    volatile int i;
    char txt_buf[256];
    unsigned int buf_len;
    signed short acc_x, acc_y, acc_z;
    double real_acc_x, real_acc_y, real_acc_z;

    gioInit();
    sciInit();
```

```
wait(10000000);

  i2cInit();

  wait(10000000);

  mpu6050_enable();
  sprintf(txt_buf, "MPU6050 Enabled₩n₩r₩0");
  buf_len = strlen(txt_buf);
  sciDisplayText(sciREG1, (uint8 *)txt_buf, buf_len);
  wait(200);

  mpu6050_acc_config();
  sprintf(txt_buf, "MPU6050 Accelerometer Configured₩n₩r₩0");
  buf_len = strlen(txt_buf);
  sciDisplayText(sciREG1, (uint8 *)txt_buf, buf_len);
  wait(200);
```

```
rtiInit();
  rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE2);
  _enable_IRQ_interrupt_();
  rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK1);

  sprintf(txt_buf, "RTI Enabled₩n₩r₩0");
  buf_len = strlen(txt_buf);
  sciDisplayText(sciREG1, (uint8 *)txt_buf, buf_len);
```

```
for(;;)
    {
        if(g_acc_flag)
        {
            acc_x = acc_y = acc_z = 0;
            real_acc_x = real_acc_y = real_acc_z = 0.0;

            acc_x = g_acc_xyz[0];
            acc_x = acc_x << 8;
            acc_x |= g_acc_xyz[1];
            real_acc_x = ((double)acc_x) / 2048.0;

            acc_y = g_acc_xyz[2];
            acc_y = acc_y << 8;
            acc_y |= g_acc_xyz[3];
            real_acc_y = ((double)acc_y) / 2048.0;
```

```
            acc_z = g_acc_xyz[4];
            acc_z = acc_z << 8;
            acc_z |= g_acc_xyz[5];
            real_acc_z = ((double)acc_z) / 2048.0;

            sprintf(txt_buf, "acc_x = %2.5lf₩tacc_y=%2.5lf₩tacc_z=%2.5lf₩n₩r₩0", real_acc_x, real_acc_y, real_acc_z);

            buf_len = strlen(txt_buf);
            sciDisplayText(sciREG1, (uint8 *)txt_buf, buf_len);

            g_acc_flag = 0;
        }
    }

    return 0;
}
```

```
void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len)
{
    while(len--)
    {
        while((UART->FLR & 0x4) == 4)
            ;
        sciSendByte(UART, *text++);
    }
}
```

```
void mpu6050_enable(void)
{
    volatile unsigned int cnt = 2;
    unsigned char data[2] = {0x00U, 0x00U};
    unsigned char slave_word_address = 0x6bU;

    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);
    i2cSend(i2cREG2, cnt, data);

    while(i2cIsBusBusy(i2cREG2) == true)
        ;
```

```
    while(i2cIsStopDetected(i2cREG2) == 0)
        ;

    i2cClearSCD(i2cREG2);

    wait(1000000);
}
```

```
void mpu6050_acc_config(void)
{
    volatile unsigned int cnt = 1;
    unsigned char data[1] = {0x18U};
    unsigned char slave_word_address = 0x1cU;

    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);
    i2cSend(i2cREG2, cnt, data);

    while(i2cIsBusBusy(i2cREG2) == true)
        ;
```

```
    while(i2cIsStopDetected(i2cREG2) == 0)
        ;

    i2cClearSCD(i2cREG2);

    wait(1000000);
}
```

```
void rtiNotification(rtiBASE_t *rtiREG, uint32 notification)
{
    unsigned char slave_word_address = 0x3B;

    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);

    while(i2cIsBusBusy(i2cREG2) == true)
        ;

    while(i2cIsStopDetected(i2cREG2) == 0)
        ;
```

```
    i2cClearSCD(i2cREG2);

    i2cSetDirection(i2cREG2, I2C_RECEIVER);
    i2cSetCount(i2cREG2, 6);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStart(i2cREG2);
    i2cReceive(i2cREG2, 6, (unsigned char *)g_acc_xyz);
    i2cSetStop(i2cREG2);

    while(i2cIsBusBusy(i2cREG2) == true)
        ;
    while(i2cIsStopDetected(i2cREG2) == 0)
        ;

    i2cClearSCD(i2cREG2);

    g_acc_flag = 1;
}
```