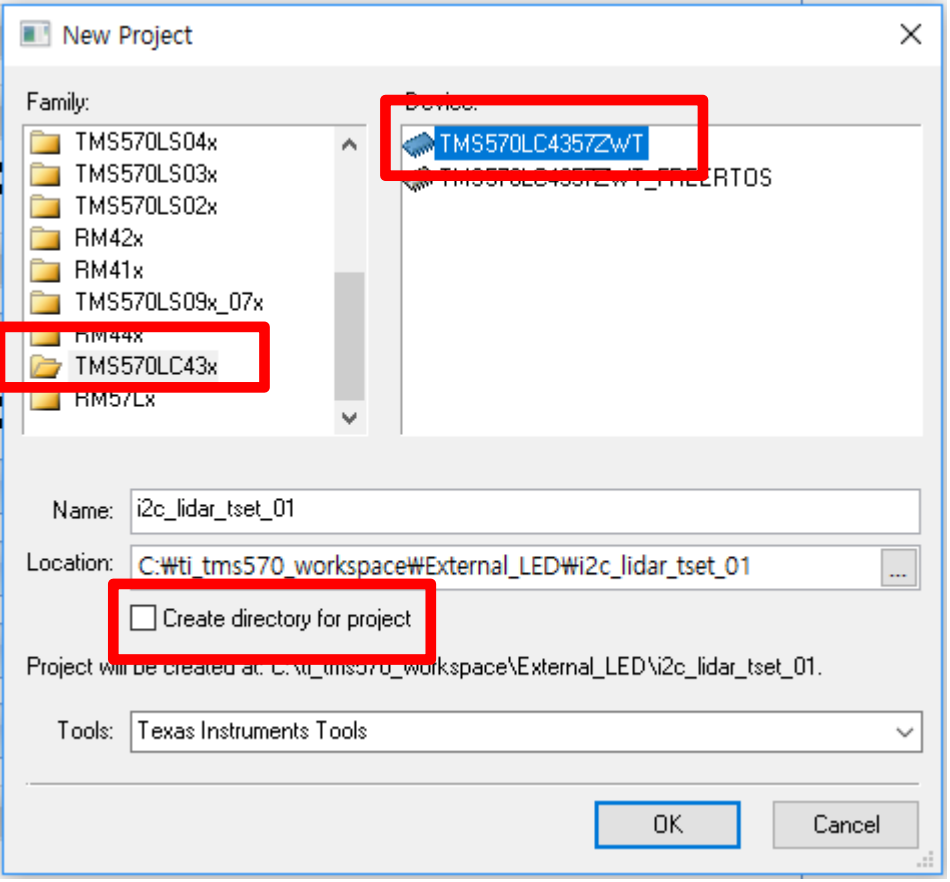
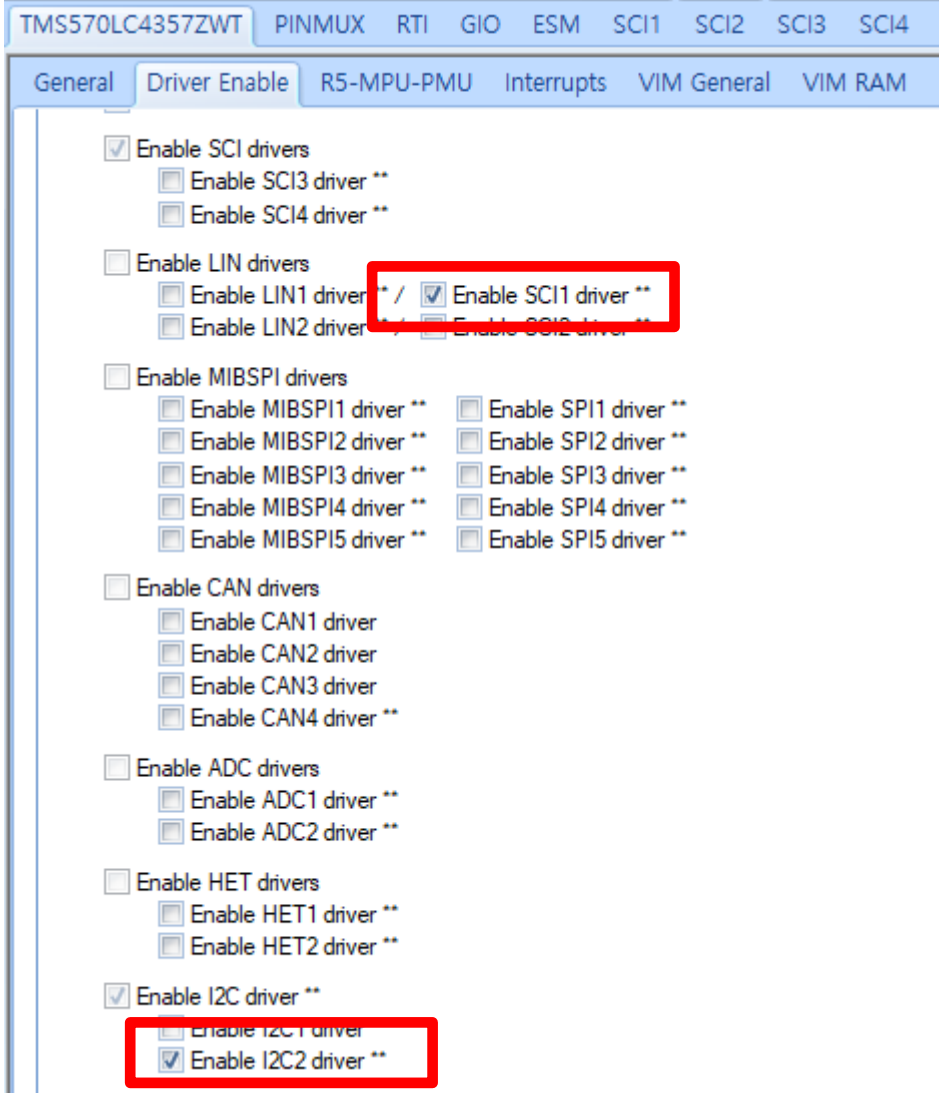
Xilinx
Zynq FPGA
TI DSP MCU 기반의
프로그래밍 및 회로 설계 전문가

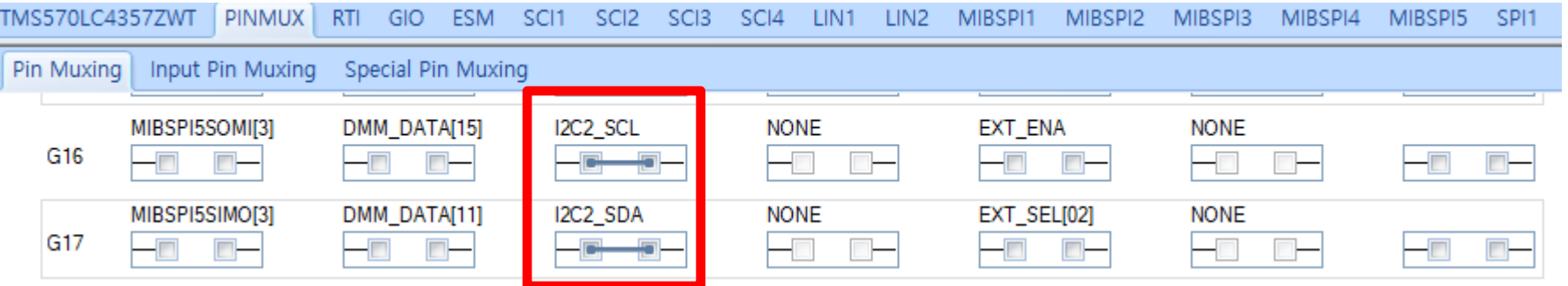
강사 이상훈
gcccompil3r@gmail.com

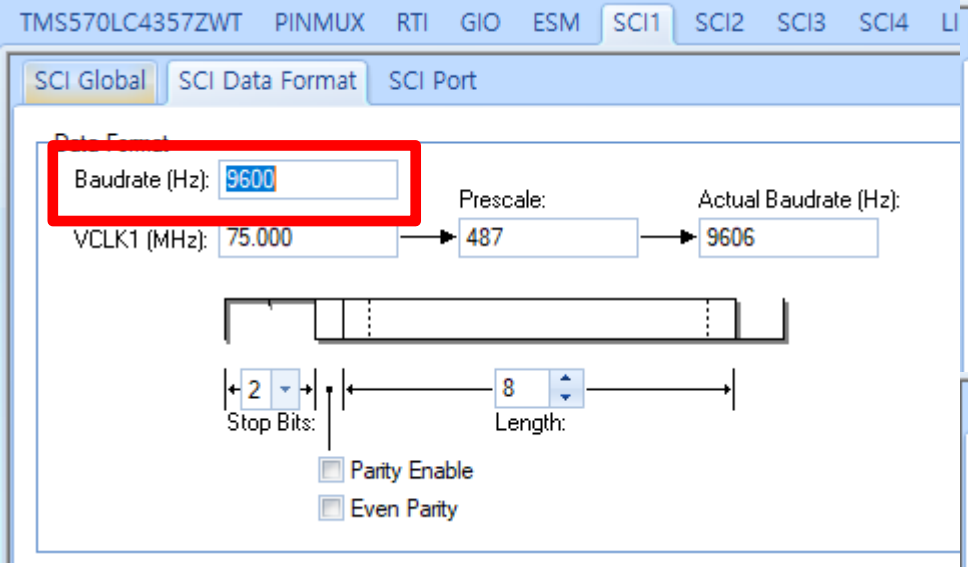
학생 김민호
minking12@naver.com

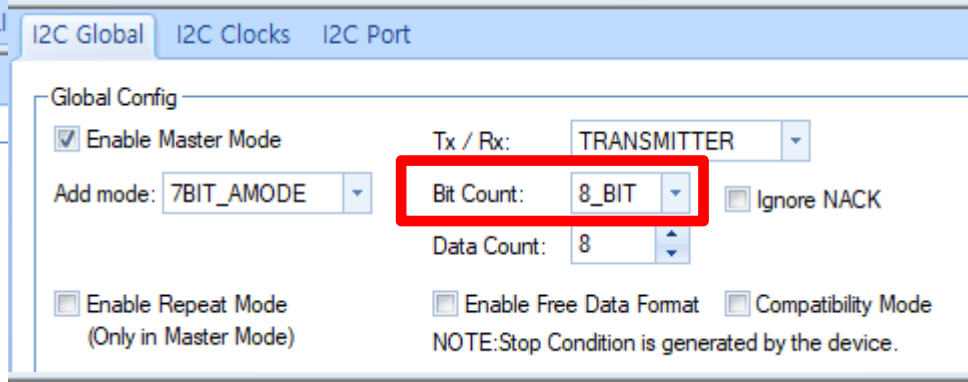


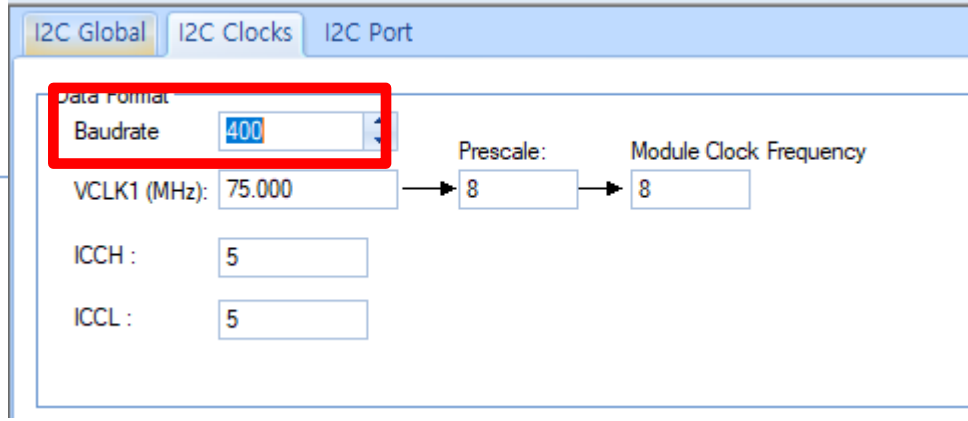


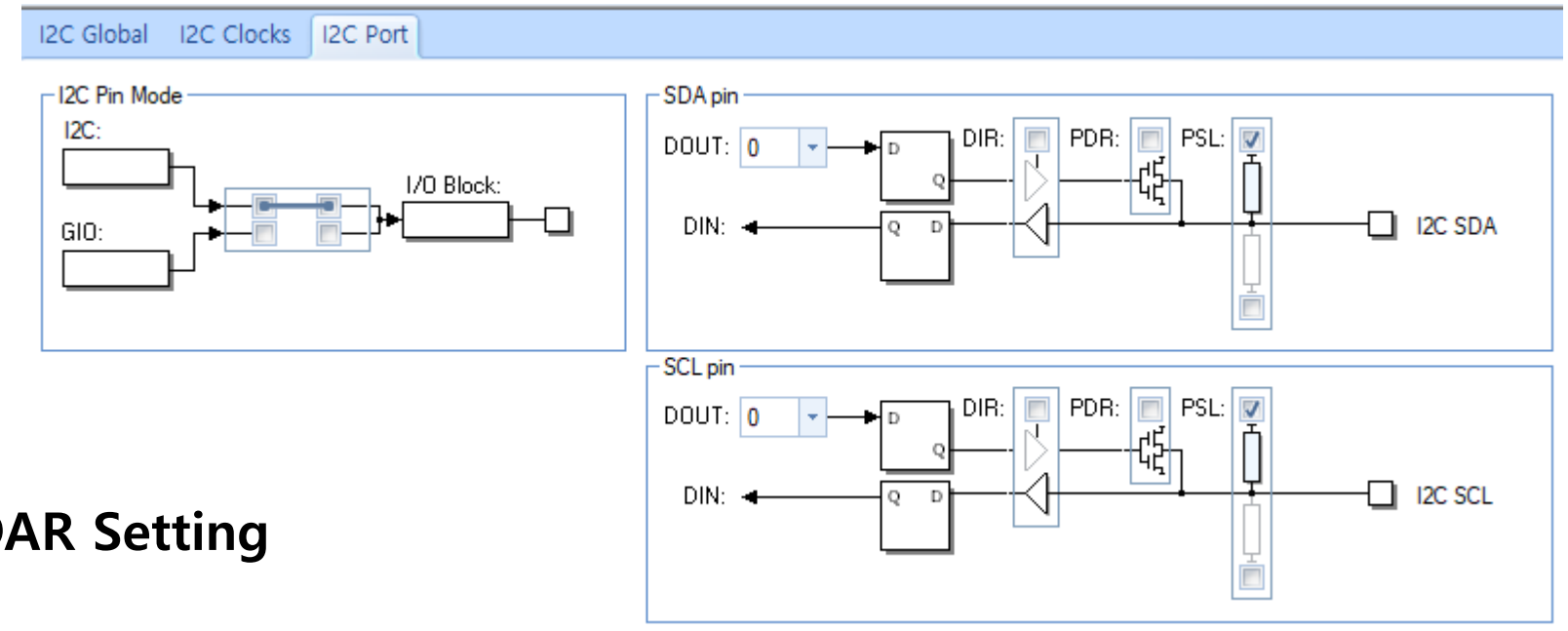
LIDAR Setting











LIDAR Setting

```
#include <HL_hal_stdtypes.h>
#include <HL_i2c.h>
#include <HL_reg_sci.h>
#include <HL_sci.h>
#include <HL_sys_core.h>
#include <stdbool.h>
#include <stdio.h>
#include <string.h>

#define UART sciREG1
#define LIDAR_SLAVE_ADDR 0x62

#define ACQ_COMMAND 0x00
#define STATUS 0x01
#define SIG_COUNT_VAL 0x02
#define ACQ_CONFIG_REG 0x04
#define THRESHOLD_BYPASS 0x1C
#define READ_FROM 0x8f

#define FULL_DELAY_HIGH 0x0f

uint8 receives[2];
volatile int g_acc_flag;

uint8 bias_cnt = 0;

void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len);
void pwmSet(void);
void wait(uint32 delay);
void Lidar_without_bias(void);
void Lidar_bias(void);
void Get_Data(void);

void Lidar_enable(void);
```

```
void wait(uint32 delay)
{
    int i;

    for (i = 0; i < delay; i++)
        ;
}

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

void disp_set(char *str)
{
    char txt_buf[256] = { 0 };
    unsigned int buf_len;
    sprintf(txt_buf, str);
    buf_len = strlen(txt_buf);
    sciDisplayText(sciREG1, (uint8 *) txt_buf, buf_len);
    wait(100000);
}
```

LIDAR CODE

```
int main(void)
{
    char txt_buf[256] = { 0 };
    unsigned int buf_len;
    volatile int i =0;
    int cnt = 1;

    uint16 ave[4] = { 0 };

    sciInit();

    disp_set("SCI Configuration Success!!₩n₩r₩0");

    i2cInit();
    wait(10000000);

    disp_set("I2C Init Success!!₩n₩r₩0");

    Lidar_enable();

    disp_set("Lidar Enable Success!!₩n₩r₩0");

    wait(1000000);

    for (;;)
    {
        Get_Data();

        if (g_acc_flag)
        {
            uint16 tmp;
            tmp = receives[0] << 8;
            tmp |= receives[1];

            if (cnt % 5 == 0)
            {
                tmp = (ave[0] + ave[1] + ave[2] + ave[3]) / 4;

                sprintf(txt_buf, "Distance = %d₩n₩r₩0", tmp);
                buf_len = strlen(txt_buf);
                sciDisplayText(sciREG1, (uint8 *) txt_buf, buf_len);

                i = 0;
                cnt++;

                g_acc_flag = 0;

            }
            else
            {
                ave[i] = tmp;

                i++;
                cnt++;

                g_acc_flag = 0;
            }
        }
        Lidar_without_bias();
        bias_cnt++;
        if(bias_cnt == 100){
            Lidar_bias();
            bias_cnt = 0;
        }
    }
    return 0;
}
```

LIDAR CODE

```
void Lidar_enable(void)
{
    uint8 tmp[4] = { 0x80, 0x08, 0x00, 0x04 };

    volatile unsigned int cnt = 7;
    i2cSetSlaveAdd(i2cREG2, LIDAR_SLAVE_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);

    disp_set("test1 -!!₩n₩r₩0");
    i2cSendByte(i2cREG2, SIG_COUNT_VAL);

    disp_set("test2 -!!₩n₩r₩0");
    i2cSend(i2cREG2, 1, &tmp[0]);
    disp_set("test3 -!!₩n₩r₩0");

    i2cSendByte(i2cREG2, ACQ_CONFIG_REG);
    i2cSend(i2cREG2, 1, &tmp[1]);
    i2cSendByte(i2cREG2, THRESHOLD_BYPASS);
    i2cSend(i2cREG2, 1, &tmp[2]);
    i2cSendByte(i2cREG2, ACQ_COMMAND);
    i2cSend(i2cREG2, 1, &tmp[3]);

    disp_set("Lidar tmp 1 Enable Success!!₩n₩r₩0");

    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    wait(100000);
}
```

```
void Get_Data()
{
    i2cSetSlaveAdd(i2cREG2, LIDAR_SLAVE_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, READ_FROM);

    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);

    i2cSetDirection(i2cREG2, I2C_RECEIVER);
    i2cSetCount(i2cREG2, 2);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStart(i2cREG2);
    i2cReceive(i2cREG2, 2, (unsigned char *) receives);
    i2cSetStop(i2cREG2);

    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);

    g_acc_flag = 1;
}
```

LIDAR CODE

```
void Lidar_bias(void)
{
    volatile unsigned int cnt = 1;
    unsigned char data[1] = { 0x04U };
    i2cSetSlaveAdd(i2cREG2, LIDAR_SLAVE_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, ACQ_COMMAND);
    i2cSend(i2cREG2, cnt, data);
    i2cSetStop(i2cREG2);

    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    wait(1000000);
}
```

```
void Lidar_without_bias(void)
{
    volatile unsigned int cnt = 1;
    unsigned char data[1] = { 0x03U };
    i2cSetSlaveAdd(i2cREG2, LIDAR_SLAVE_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER);
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, ACQ_COMMAND);
    i2cSend(i2cREG2, cnt, data);
    i2cSetStop(i2cREG2);

    while (i2cIsBusBusy(i2cREG2) == true)
        ;
    while (i2cIsStopDetected(i2cREG2) == 0)
        ;
    i2cClearSCD(i2cREG2);
    wait(1000000);
}
```

LIDAR CODE

```
Distance = 179
Distance = 179
Distance = 178
Distance = 180
Distance = 183
Distance = 180
Distance = 10
Distance = 22
Distance = 14
Distance = 1
Distance = 145
Distance = 110
Distance = 1
Distance = 45
Distance = 12
Distance = 13
Distance = 20
Distance = 14
Distance = 20
Distance = 8
Distance = 13
Distance = 17
Distance = 25
Distance = 5
Distance = 15
Distance = 11
Distance = 21
Distance = 20
Distance = 20
Distance = 11
Distance = 15
Distance = 20
Distance = 13
Distance = 5
Distance = 9
Distance = 178
Distance = 178
Distance = 179
Distance = 181
Distance = 180
```