

# TI DSP, MCU, Xilinx Zynq FPGA 프로그래밍 전문가 과정

2018.07.25

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 황수정

sue100012@naver.com

## I2C - MPU6050

```
int main(void)
{
    char txt_buf[256] = {0};
    unsigned int buf_len;
    volatile int i;
    signed short acc_x, acc_y, acc_z;
    double real_acc_x, real_acc_y, real_acc_z;

    sdInit();
    disp_set("SCI Configuration Success!!\n\r\0"); //디버깅용

    gpioInit();
    disp_set("GPIO Init Success!!\n\r\0");

    i2cInit();
    wait(10000000); //delay
    disp_set("I2C Init Success!!\n\r\0");

    MPU6050_enable();
    disp_set("MPU6050 Enable Success!!\n\r\0");

    MPU6050_acc_config();
    disp_set("MPU6050 Accelerometer Configure Success!!\n\r\0");

    rtiInit();
    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
    _enable_IRQ_interrupt();
    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
```

```
disp_set("RTI Init Success!!\n\r\0");

etpwmInit();

disp_set("ETPWM Configuration Success!!\n\r\0");

etpwmStartTBCLK();
wait(1000000);

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;
```

## I2C - MPU6050

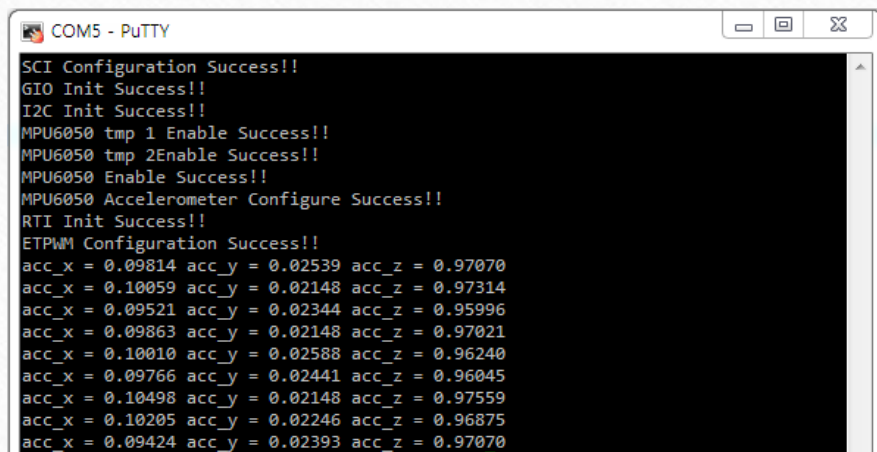
```
acc_z = g_acc_xyz[4];
acc_z = acc_z << 8;
acc_z |= g_acc_xyz[5];
real_acc_z = ((double)acc_z) / 2048.0;
```

```
if (real_acc_x > 0){
    pwmSet1();
}
else{
    pwmSet2();
}
```

```
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;
```

```
    }
}
}
```



```
COM5 - PuTTY
SCI Configuration Success!!
GPIO Init Success!!
I2C Init Success!!
MPU6050 tmp 1 Enable Success!!
MPU6050 tmp 2Enable Success!!
MPU6050 Enable Success!!
MPU6050 Accelerometer Configure Success!!
RTI Init Success!!
ETPWM Configuration Success!!
acc_x = 0.09814 acc_y = 0.02539 acc_z = 0.97070
acc_x = 0.10059 acc_y = 0.02148 acc_z = 0.97314
acc_x = 0.09521 acc_y = 0.02344 acc_z = 0.95996
acc_x = 0.09863 acc_y = 0.02148 acc_z = 0.97021
acc_x = 0.10010 acc_y = 0.02588 acc_z = 0.96240
acc_x = 0.09766 acc_y = 0.02441 acc_z = 0.96045
acc_x = 0.10498 acc_y = 0.02148 acc_z = 0.97559
acc_x = 0.10205 acc_y = 0.02246 acc_z = 0.96875
acc_x = 0.09424 acc_y = 0.02393 acc_z = 0.97070
```

## I2C - MPU6050

```
void MPU6050_acc_config(void)
{
    volatile unsigned int cnt = 1;
    unsigned char data[1] = {0x18U};
    unsigned char slave_word_address = 0x1dU;

    i2cSetSlaveAdd(i2cREG2, MPU6050_ADDR); //슬레이브 장치 주소 지정.
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER); //통신모드 설정
    i2cSetCount(i2cREG2, cnt + 1); //CNT+1만큼 받겠다 설정한 것
    i2cSetMode(i2cREG2, I2C_MASTER); //i2cREG2 모듈을 마스터모드로 설정
    /*마스터 모드를 설정했으므로 STOP하고 START해 준 것이다.*/
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address); //slave word address를 전송
    i2cSend(i2cREG2, cnt, data); //data를 cnt 길이만큼 데이터 블록을 전송

    while(i2cIsBusBusy(i2cREG2) == true) //버스가 사용 중일 때, 참
        ;
    while(i2cIsStopDetected(i2cREG2) == 0) //통신이 stop할 때까지
        ;
    i2cClearSCD(i2cREG2); //SCD 플래그를 클리어

    wait(1000000); //delay
}
```

## I2C - MPU6050 문제점

양방향으로 전기 신호를 주기 때문에 플로팅 상태에 빠질 수 있다. 이 때에는 잡음에 매우 취약해 지므로 신호를 확실히 잡아 줄 역할이 필요하다.

→ SDA, SCL에 풀업 저항을 달아주면 된다.

그러나 HAL COGEN에서 I2C를 지정할 때, 내부에서 풀업 저항을 설정하기 때문에 크게 문제 되지 않는다.

### 전력 문제

→ 전원을 주는 곳에 캐패시터를 달아주어 전력이 떨어지지 않도록 해준다. 용량클수록 안정적으로 쓸 수 있다. 그러나 너무 큰 용량은 충전하는데 시간이 걸리므로 적당한 크기를 달아준다.

### 노이즈, Crosstalk 문제

→ 도선에 자기장이 발생하여 인접한 다른 도선에 피해를 줄 수 있으므로 도선은 인접하지 않도록 한다. 또한 도선의 종류와 길이가 도선 저항에 영향을 주며 고주파 신호일수록 도선 길이가 길면 신호가 점점 약해져 통신에 문제가 생길 수 있다. 그러므로 도선은 짧은 것이 좋다. 노이즈는 고주파신호를 걸러주는 캐패시터를 이용해 전원과 접지에 필터를 달아 잡아준다.

## I2C LIDAR

```
int main(void)
{
    /* USER CODE BEGIN (3) */

    gpioInit();
    sciInit();
    rtiInit();
    i2cInit();
    wait(10000000); //delay

    disp_set("GIO, SCI, RTI, I2C Init Success!!\n\r\0"); //디버깅용

    LIDAR_enable();

    wait(10000000); //delay
    disp_set("LIDAR Enable Success!!\n\r\0"); //디버깅용

    LIDAR_acc_config();
    disp_set("MPU6050 Accelerometer Configure Success!!\n\r\0");

    rtiInit();
    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
    _enable_IRQ_interrupt();
    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
```

```
void LIDAR_enable(void)
{
    volatile unsigned int cnt = 2;
    unsigned char data[2] = {0x00U, 0x00U};
    unsigned char slave_word_address = 0x66U;

    i2cSetSlaveAdd(i2cREG2, LIDAR_ADDR);
    i2cSetDirection(i2cREG2, I2C_TRANSMITTER); //전송 방향 결정
    i2cSetCount(i2cREG2, cnt + 1);
    i2cSetMode(i2cREG2, I2C_MASTER);
    i2cSetStop(i2cREG2);
    i2cSetStart(i2cREG2);
    i2cSendByte(i2cREG2, slave_word_address);
    disp_set("MPU6050 tmp 1 Enable Success!!\n\r\0");
    i2cSend(i2cREG2, cnt, data);
    disp_set("MPU6050 tmp 2Enable Success!!\n\r\0");

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

    wait(100000);
}
```

감사합니다