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

강사 이상훈
gcccompil3r@gmail.com

학생 김민호
minking12@naver.com

RTOS_MPU6050 Setting

```c
#include <FreeRTOS.h>
#include <FreeRTOSConfig.h>
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_sci.h"
#include "HL_i2c.h"
#include <string.h>
#include <stdio.h>
#include "HL_sys_common.h"
#include "FreeRTOS.h"
#include "os_task.h"

#define UART        sciREG1
#define MPU6050_ADDR    0x68

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;
void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len);
void wait(uint32 delay);
void eeee();
void MPU6050_enable(void);
void MPU6050_acc_config(void);
void disp_set(char *);
uint32 rx_data = 0;
uint32 tmp = 0;
uint32 value = 0;
volatile char g_acc_xyz[6];
volatile int g_acc_flag;
#define IDX     2
uint32 duty_arr[IDX] = { 1000, 2000 };
xTaskHandle xTask1Handle;


void vTask1(void *pbParameters)
{
    for (;;)
    {
        wait(10000000);
        eeee();
        {
            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);
            wait(1000000);
            sciDisplayText(sciREG1, (uint8 *) txt_buf, buf_len);
        }
    }
}
```

```c
int main(void)
{
    sciInit();
    disp_set("SCI Configuration Success!!\n\r\0");
    i2cInit();
    wait(10000000);
    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");
    if (xTaskCreate(vTask1, "task1", configMINIMAL_STACK_SIZE * 8, NULL,
1,&xTask1Handle) != pdTRUE)

    {
        while (1)
            ;
    }

    vTaskStartScheduler();

    while (1)
        ;
    return 0;
}
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++);
    }

}
```

```c
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);
    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);
}

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);
}
```

RTOS_MPU6050 Code
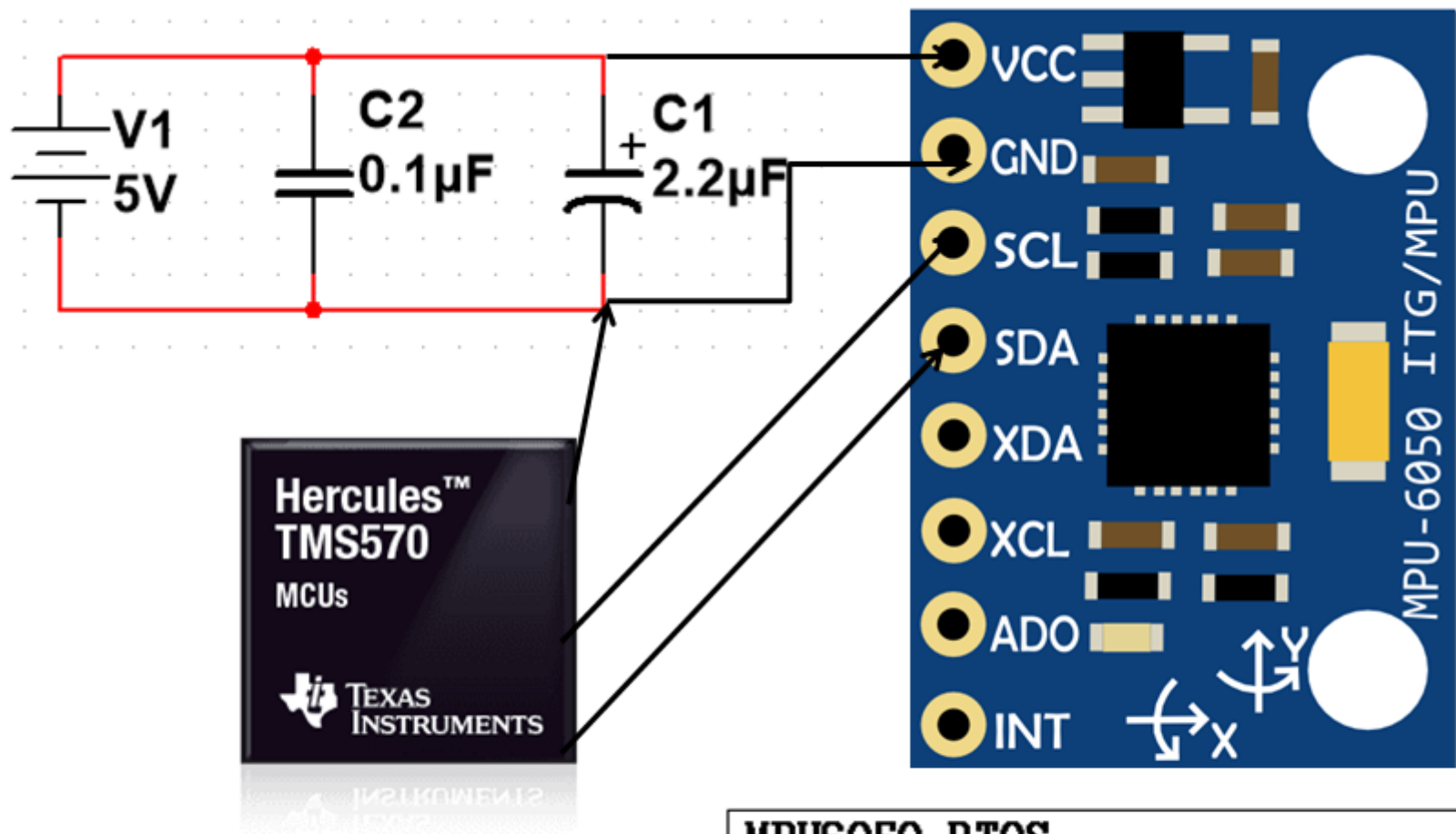
```c
void eeee()

{

    unsigned char slave_word_address = 0x3B;
    wait(1000000);
    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;
}
```

```c
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);
}
```
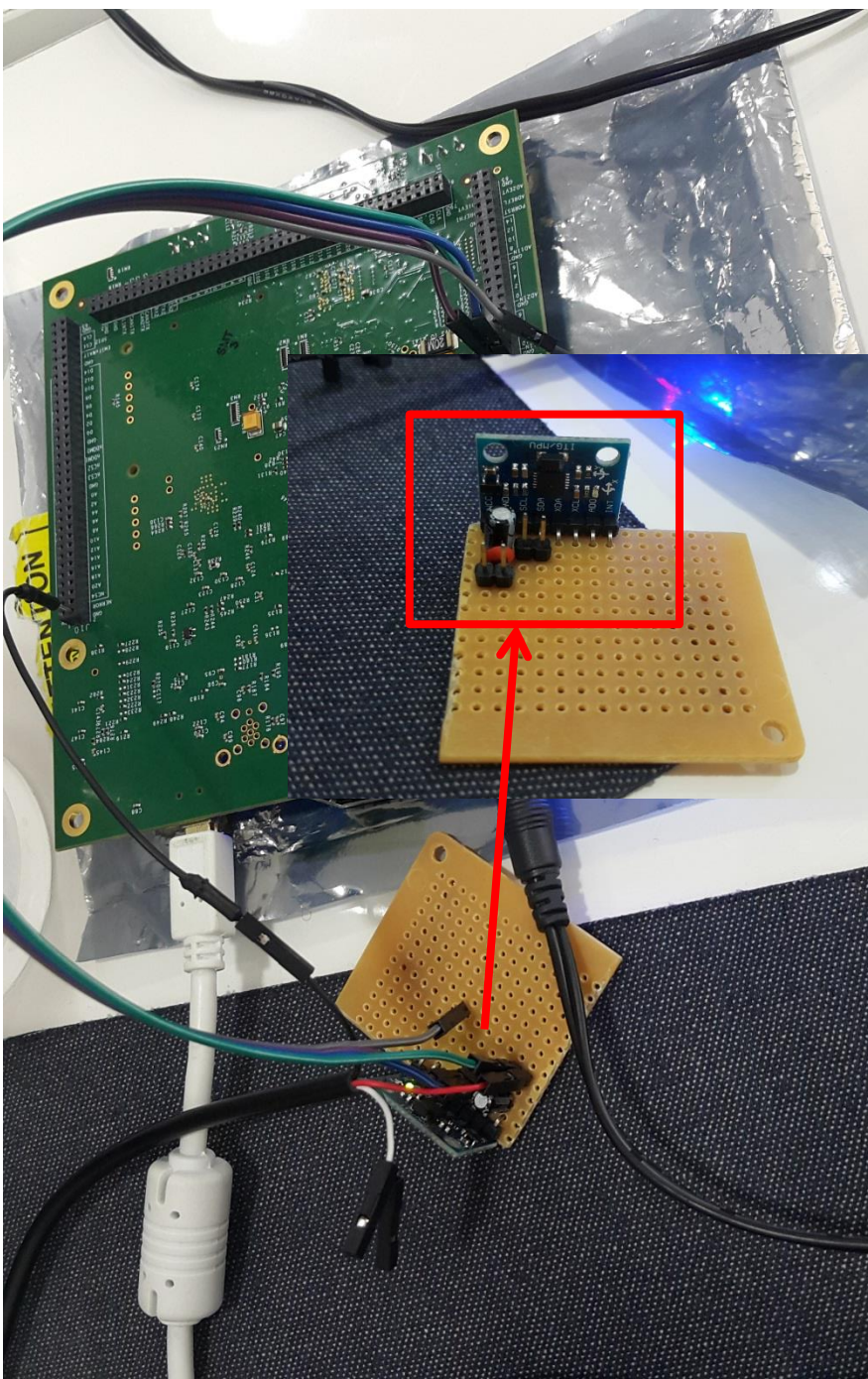
RTOS_MPU6050 Code

| V1 5V | C2 0.1μF | C1 2.2μF |

Hercules™
TMS570
MCUs

TEXAS
INSTRUMENTS

MPU-6050 ITG/MPU

VCC
GND
SCL
SDA
XDA
XCL
ADO
INT

| MPU6050_RTOS |
| Designer : 황수정 , 김민호 |
| 2018.08.14 |
| Project AI CAR |

```
acc_x = 0.63965 acc_y = 0.33789 acc_z = 0.85791
acc_x = 0.78906 acc_y = 0.40527 acc_z = 0.64600
acc_x = 0.71289 acc_y = 0.36963 acc_z = 0.75488
acc_x = 0.77588 acc_y = 0.40527 acc_z = 0.68408
acc_x = 0.61768 acc_y = 0.31543 acc_z = 0.90674
acc_x = 0.71094 acc_y = 0.35010 acc_z = 0.82715
acc_x = 0.60693 acc_y = 0.33545 acc_z = 0.92969
acc_x = 0.55225 acc_y = 0.32373 acc_z = 0.94434
acc_x = 0.63818 acc_y = 0.29639 acc_z = 0.87549
acc_x = 0.75439 acc_y = 0.35596 acc_z = 0.61279
acc_x = 0.53271 acc_y = 0.13867 acc_z = 1.05566
acc_x = 0.38086 acc_y = -0.16113          acc_z = 0.95850
acc_x = 0.19873 acc_y = -0.27393          acc_z = 1.08984
acc_x = 0.87207 acc_y = 0.58350 acc_z = 0.40088
acc_x = 0.78125 acc_y = 0.19287 acc_z = 0.89990
acc_x = 0.90430 acc_y = 0.50098 acc_z = 0.06982
acc_x = 0.68994 acc_y = 0.47949 acc_z = -0.25293
acc_x = 0.65918 acc_y = 0.48975 acc_z = -0.50781
acc_x = 0.52588 acc_y = 0.44727 acc_z = -0.50977
acc_x = 0.85449 acc_y = 0.48535 acc_z = -0.13330
acc_x = 0.89014 acc_y = 0.14600 acc_z = 0.71387
acc_x = 0.60742 acc_y = -0.15527          acc_z = 0.95459
acc_x = 0.69678 acc_y = -0.09766          acc_z = 0.82861
acc_x = 0.47168 acc_y = -0.19580          acc_z = 1.07031
acc_x = 0.33691 acc_y = -0.23389          acc_z = 1.10156
acc_x = 0.33838 acc_y = -0.21777          acc_z = 1.10791
acc_x = 0.74414 acc_y = 0.25781 acc_z = 0.67578
acc_x = 0.75098 acc_y = 0.35498 acc_z = 0.69775
acc_x = 0.87061 acc_y = 0.54150 acc_z = 0.05957
acc_x = 0.82373 acc_y = 0.44775 acc_z = -0.24121
acc_x = 0.63574 acc_y = 0.40381 acc_z = -0.53174
acc_x = 0.52979 acc_y = 0.36914 acc_z = -0.60938
acc_x = 0.93408 acc_y = 0.41699 acc_z = 0.10742
acc_x = 0.46729 acc_y = -0.13281          acc_z = 0.98975
acc_x = 0.44873 acc_y = -0.15967          acc_z = 1.05762
acc_x = 0.37402 acc_y = -0.19336          acc_z = 1.06494
acc_x = 0.43848 acc_y = -0.17139          acc_z = 1.01221
```