

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

- MPU6050 칼만필터 내장형 (UART BASE) -

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – GJ (박현우)
uc820@naver.com

목차

MPU6050 + STM32 (칼만필터 내장형)

1. DATASHEET
2. HALCOGEN 설정
3. CODE
4. SIMULATION

1. DATASHEET

Precision six-axis inertial navigation module specification

1 product description..

This six-axis module uses advanced digital filtering technology (Kalman filter), can effectively reduce the measurement noise, improve the measurement accuracy..

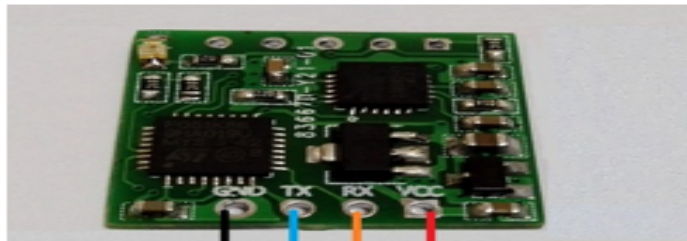
The module integrates motion engine DMP, and obtain quaternion to get the current attitude. Attitude measurement accuracy of 0.01 degrees, high stability, performance and even better than some of the professional inclinometer!..

This module uses high-precision six-axis gyro accelerometer MPU6050, read the measurement data MPU6050 by a processor and output via the serial port, eliminating the need for users to develop their own MPU6050 complex I2C protocol, and careful PCB layout and processes ensure MPU6050 Receive external interference minimum, the highest precision measurement..

2 Performance parameters..

1. Voltage: 3V ~ 6V..
2. Current: <10mA..
3. Volume: 17.8mm X 17.8mm Weight: 1.1g..
4. Pad pitch: up and down 100mil (2.54mm), left and right 600mil (15.24mm)..
5. Measuring dimensions: Acceleration: 3-dimensional angular velocity: 3-dimensional attitude angle: 3-dimensional..
6. Range: Acceleration: $\pm 16g$, angular velocity: $\pm 2000^{\circ} / s...$
7. Resolution: Acceleration: $6.1e-5g$, the angular velocity: $7.6e-3^{\circ} / s...$
8. Stability: acceleration: 0.01g, angular velocity 0.05 $^{\circ} / s...$
9. Attitude Stability Measurement: 0.01 $^{\circ}...$
10. Data output frequency 100Hz (baud rate 115200) / 20Hz (baud rate 9600). 11, data interface: Serial (TTL level), I2C (directly connected MPU6050, no attitude output)..
- 10, baud rate 115200kps / 9600kps...

3 Pin Description:..



1. 허용 전압과 전류 확인
2. 각가속도와 각속도 범위 확인
3. 오차범위 확인
4. 통신속도 확인

1. DATASHEET

5.1 Host computer to the module.

Instruction content.	Features.	Remarks.
Lt; / RTI & gt;	Angle initialization.	The Z-axis angle to zero.

0x61.	Using the serial port, disable I2C.	Power-down save, it is recommended to use the host computer to modify.
0x62.	Disable the serial port, using the I2C interface.	Power-down save, it is recommended to use the host computer to modify.
0x63 ('c').	115200 baud rate, frame rate 100Hz.	Power-down save, it is recommended to use the host computer to modify.
0x64 ('d').	9600 baud rate, frame rate 10Hz.	Power-down save, it is recommended to use the host computer to modify.

Description:

1. module after power required to remain stationary inside the module MCU will perform automatic calibration module when stationary (elimination of gyro drift), after the calibration angle Z axis will be re-initialized to 0, Z axis angle output is 0, Can be regarded as the signal that the automatic calibration completes.

2. The factory default settings using the serial port baud rate 115200, frame rate 100Hz. Configuration can be configured through the host computer software, because all configuration is saved, so only need to configure once.

5.2 module first bit machine:

Module sends first bit machine each frame of data is divided into three packets, each packet acceleration, angular velocity and angle bag package, 3.

Packet sequence output. 115200 baud rate when every 10ms output a data baud rate of 9600 when every 50ms output a data.

1. Stm32에 0x61 또는 0x62를 보내서 어떤 모드를 쓸지를 선택.
2. 시리얼 창에 0x63 또는 0x64로 통신 속도도 설정할 수 있음.
3. 공장에서 기본 셋팅으로 통신속도를 115200으로 설정되어 나옴.

1. DATASHEET

The data number.,	Data content.,	meaning.,
Lt; / RTI & gt;.,	0x55 ('U').,	Baotou.,
1.,	0x51 ('Q').,	Identify this package is the acceleration package.,
2.,	<u>AxL</u> .,	X-axis acceleration low byte.,
3.,	<u>AxH</u> .,	X-axis acceleration high byte.,
4.,	<u>AyL</u> .,	y-axis acceleration low byte.,
5.,	<u>AyH</u> .,	y-axis acceleration high byte.,
6.,	<u>AzL</u> .,	z-axis acceleration low byte.,
7.,	<u>AzH</u> .,	z-axis acceleration high byte.,
8.,	TL.,	Temperature low byte.,
9.,	TH.,	Temperature high byte.,
10.,	Sum.,	Checksum.,

Acceleration calculation formula:.,

$a_x = ((AxH < 8) | AxL) / 32768 * 16g$ (g is the gravitational acceleration, preferably $9.8m/s^2$).,

$a_y = ((AyH < 8) | AyL) / 32768 * 16g$ (g of acceleration of gravity, it is desirable $9.8m/s^2$).,

$a_z = ((AzH < 8) | AzL) / 32768 * 16g$ (g is the gravitational acceleration, the desirability of $9.8m/s^2$) temperature calculation formula:.,

$T = ((TH < 8) | TL) / 340 + 36.53 \text{ } ^\circ C$.,

Checksum:.,

$Sum = 0x55 + 0x51 + AxH + AxL + AyH + AyL + AzH + AzL + TH + TL$.,

1. 시리얼 창에서 0x51 이후에 나오는 값이 각가속도이다.

2. 주의할 점!!

Low , High 순으로 나옴.

1. DATASHEET

5.2.2 Angular velocity output:

The data number.	Data content.	meaning.
Lt / RTI & gt;	0x55 ('U').	Baotou.
1.	0x52 ('R').	Identifies this package as angular velocity packets.
2.	<u>WxL</u> .	X-axis angular velocity low byte.
3.	<u>WxH</u> .	X-axis acceleration high byte.
4.	<u>WyL</u> .	y-axis acceleration low byte.
5.	<u>WyH</u> .	y-axis acceleration high byte.
6.	<u>WzL</u> .	z-axis acceleration low byte.
7.	<u>WzH</u> .	z-axis acceleration high byte.
8.	TL.	Temperature low byte.
9.	TH.	Temperature high byte.
10.	Sum.	Checksum.

Angular velocity calculation formula:

$$w_x = ((w_{xH} \ll 8) | w_{xL}) / 32768 * 2000 (^{\circ}/s)$$

$$w_y = ((w_{yH} \ll 8) | w_{yL}) / 32768 * 2000 (^{\circ}/s)$$

$$w_z = ((w_{zH} \ll 8) | w_{zL}) / 32768 * 2000 (^{\circ}/s)$$

temperature calculation formula:

$$T = ((TH \ll 8) | TL) / 340 + 36.53 (^{\circ}C)$$

Checksum:

$$Sum = 0x55 + 0x52 + w_{xH} + w_{xL} + w_{yH} + w_{yL} + w_{zH} + w_{zL} + TH + TL$$

1. 시리얼 창에서 0x52 이후에 나오는 값이 각 속도이다.

2. 주의할 점!!

Low , High 순으로 나옴.

1. DATASHEET

5.2.3 Angle Output:

The data number.	Data content.	meaning.
Lt / RTI & gt.	0x55 ('U').	Baotou.
1.	0x53 ('S').	Identifies this package as an angle package.
2.	RollL.	X axis angle low byte.
3.	RollH.	X axis angle high byte.
4.	PitchL.	y-axis angle of the low byte.
5.	PitchH.	y-axis angle high byte.
6.	YawL.	z axis angle low byte.
7.	YawH.	z axis angle high byte.
8.	TL.	Temperature low byte.
9.	TH.	Temperature high byte.
10.	Sum.	Checksum.

Angular velocity calculation formula:

Roll angle (x-axis) Roll = ((RollH << 8) | RollL) / 32768 * 180 (°)

pitch angle (y-axis) Pitch = ((PitchH << 8) | PitchL) / 32768 * 180 (°)

yaw angle (z-axis) yaw = ((YawH << 8) | YawL) / 32768 * 180

(°) temperature is calculated:

T = ((TH << 8) | TL) / 340 + 36.53 °C.

1. 시리얼 창에서 0x53 이후에 나오는 값이 각도이다.

2. 주의할 점!!

Low , High 순으로 나옴.

2. HALCOGEN 설정

- Enable Driver Compilation

Click and mark the required modules for driver compilation from below:

☒ Enable RTI driver ☐ Mark/Unmark all drivers

☒ 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 **

☐ Enable HET drivers

☐ Enable HET1 driver **

☐ Enable HET2 driver **

☒ Enable I2C driver **

☐ Enable I2C1 driver **

☒ Enable I2C2 driver **

☐ Enable EMAC driver **

☐ Enable DCC driver

Enable / Disable Peripherals					
<input type="checkbox"/> HET1	<input type="checkbox"/> GIOA	<input type="checkbox"/> MIBSPI2	<input type="checkbox"/> MIBSPI1	<input checked="" type="checkbox"/> SCI3	<input type="checkbox"/> RMI
<input type="checkbox"/> HET2	<input type="checkbox"/> GIOB	<input type="checkbox"/> MIBSPI4	<input type="checkbox"/> MIBSPI3	<input type="checkbox"/> SCI4	<input type="checkbox"/> MII
<input type="checkbox"/> EMIF	<input type="checkbox"/> EQEP	<input type="checkbox"/> AD1EVT	<input type="checkbox"/> MIBSPI5	<input type="checkbox"/> LIN2/SCI2	<input type="checkbox"/> CAN4
<input type="checkbox"/> ETPWM	<input type="checkbox"/> ECAP	<input type="checkbox"/> AD2EVT	<input type="checkbox"/> I2C1	<input checked="" type="checkbox"/> I2C2	

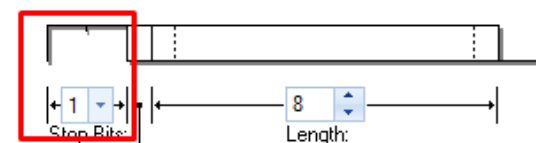
Ball	Default Mux	Mux Option 1	Mux Option 2	Mux Option 3	Mux
A4	N2HET1[16]	NONE	NONE	ETPWM1SYNCI	NOI
A13	N2HET1[17]	EMIF_nOE	SCI4RX	NONE	NOI
A14	N2HET1[26]	NONE	MII_RXD[1]	RMII_RXD[1]	NOI
B2	MIBSPI3NCS[2]	I2C1_SDA	NONE	N2HET1[27]	NOI
B3	N2HET1[22]	EMIF_nDQM[3]	NONE	NONE	NOI
B4	N2HET1[12]	MIBSPI4NCS[5]	MII_CRS	RMII_CRS_DV	NOI
B5	GIOA[5]	NONE	NONE	EXTCLKIN	NOI
B6	MIBSPI5NCS[1]	DMM_DATA[06]	NONE	NONE	NOI
B8	FRAYTX2	NONE	NONE	GIOB[0]	NOI
B9	FRAYTXEN2	NONE	NONE	GIOB[2]	NOI

Note
GIO pins are mapp
and alternate termi
MII have dedicate
RMII and MII chec
Special Pinmuxing

Level: TTL-level (non-RS232 level, when the module is connected to the wrong RS232 level could cause damage to the modules) Baud rate: 115200/9600, 1 stop bit, parity bit 0...

- Data Format

Baudrate (Hz): 115200
VCLK1 (MHz): 75.000 → Prescale: 40 → Actual Baudrate (Hz): 114329



☐ Parity Enable

☐ Even Parity

3. CODE

```
#include "HL_sys_common.h"
#include "HL_etpwm.h"
#include "HL_system.h"
#include "HL_sci.h"
#include "HL_rti.h"

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

#define UART sciREG1

void sciDisplayText(sciBASE_t *sci, uint8 *text, uint32 len);
void pwmSet(void);
void wait(uint32 delay);
void disp_set(char *str);

uint32 rx_data =0;
uint32 tmp =0;
uint32 value =0;
uint32 duty_arr[10] = {1000, 1020, 1040, 1060, 1080, 1100, 1150, 1200, 1400, 2000};

int acc_l_x, acc_l_y, acc_l_z;
int acc_h_x, acc_h_y, acc_h_z;

double real_acc_x, real_acc_y, real_acc_z;

int gyro_x, gyro_y, gyro_z;
double real_gyro_x, real_gyro_y, real_gyro_z;

int l_roll, l_pitch, l_yaw;
int h_roll, h_pitch, h_yaw;
double real_roll, real_pitch, real_yaw;
```

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

    sciInit();
    disp_set("sci Init Success!\n\r");
    etpwmInit();
    disp_set("pwm Init Success!\n\r");

    rtiInit();
    rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
    _enable_IRQ_interrupt_();
    rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);

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

    sciSendByte(sciREG3, 0x61); // using the serial port , disable I2C

    etpwmStartTBCLK();
    wait(10000);

    for (;;)
    {
        if (sciReceiveByte(sciREG3) == 0x53)
        {
            /*roll*/
            l_roll = sciReceiveByte(sciREG3);
            h_roll = sciReceiveByte(sciREG3);
            h_roll <<= 8;
            h_roll |= l_roll;
            real_roll = ((double) h_roll) / 32768.0 * 180;
            /*pitch*/
            l_pitch = sciReceiveByte(sciREG3);
            h_pitch = sciReceiveByte(sciREG3);
            h_pitch <<= 8;
            h_pitch |= l_pitch;
            real_pitch = ((double) h_pitch) / 32768.0 * 180;
            /*yaw*/
            l_yaw = sciReceiveByte(sciREG3);
            h_yaw = sciReceiveByte(sciREG3);
            h_yaw <<= 8;
            h_yaw |= l_yaw;
            real_yaw = ((double) h_yaw) / 32768.0 * 180;

            sprintf(txt_buf, "roll = %lf \t pitch = %lf \t yaw = %lf \n\r\0",
                    real_roll, real_pitch, real_yaw);
```

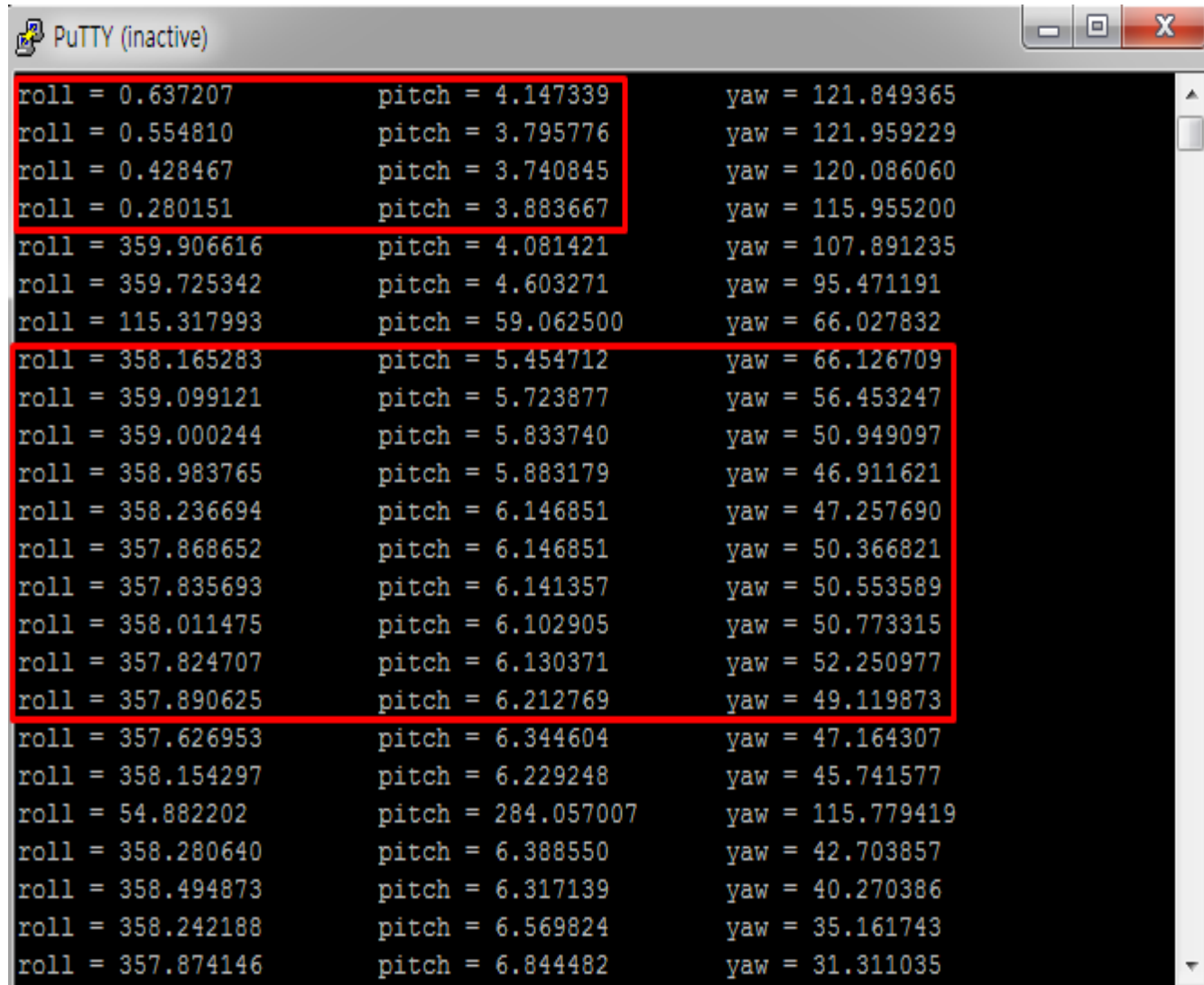
```
void pwmSet(){
    value = duty_arr[rx_data];
    etpwmSetCmpA(etpwmREG1, value);
    wait(10000);
}

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);
}
```

4. SIMULATION



```
PuTTY (inactive)
roll = 0.637207    pitch = 4.147339    yaw = 121.849365
roll = 0.554810    pitch = 3.795776    yaw = 121.959229
roll = 0.428467    pitch = 3.740845    yaw = 120.086060
roll = 0.280151    pitch = 3.883667    yaw = 115.955200
roll = 359.906616  pitch = 4.081421    yaw = 107.891235
roll = 359.725342  pitch = 4.603271    yaw = 95.471191
roll = 115.317993  pitch = 59.062500   yaw = 66.027832
roll = 358.165283  pitch = 5.454712    yaw = 66.126709
roll = 359.099121  pitch = 5.723877    yaw = 56.453247
roll = 359.000244  pitch = 5.833740    yaw = 50.949097
roll = 358.983765  pitch = 5.883179    yaw = 46.911621
roll = 358.236694  pitch = 6.146851    yaw = 47.257690
roll = 357.868652  pitch = 6.146851    yaw = 50.366821
roll = 357.835693  pitch = 6.141357    yaw = 50.553589
roll = 358.011475  pitch = 6.102905    yaw = 50.773315
roll = 357.824707  pitch = 6.130371    yaw = 52.250977
roll = 357.890625  pitch = 6.212769    yaw = 49.119873
roll = 357.626953  pitch = 6.344604    yaw = 47.164307
roll = 358.154297  pitch = 6.229248    yaw = 45.741577
roll = 54.882202   pitch = 284.057007   yaw = 115.779419
roll = 358.280640  pitch = 6.388550    yaw = 42.703857
roll = 358.494873  pitch = 6.317139    yaw = 40.270386
roll = 358.242188  pitch = 6.569824    yaw = 35.161743
roll = 357.874146  pitch = 6.844482    yaw = 31.311035
```

1. roll 과 pitch 값이 잘 나오는 것을 확인 할 수 있다.
2. 반대 방향일 때는 360에서 역으로 빠지는 것을 확인이 된다.