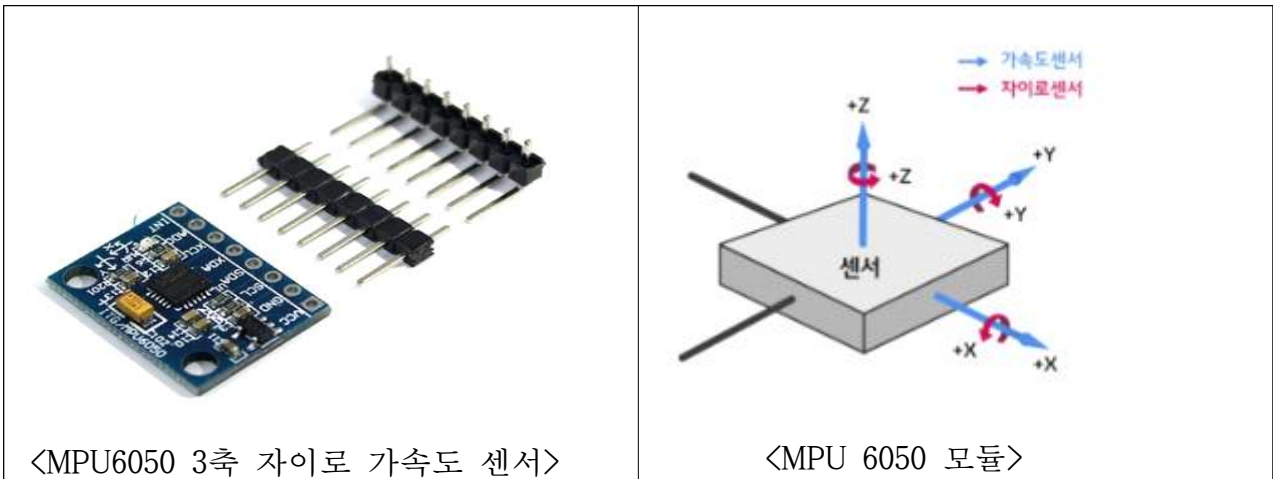


## <라즈베리파이 자이로 가속도 센터(MPU6050)을 활용한 걸음수>

### ■ MPU 6050 3축 자이로 가속도 센터 특징



<MPU6050 3축 자이로 가속도 센서>

<MPU 6050 모듈>

- I2C 통신을 이용하여 X, Y, Z 축 값을 받아 올 수 있음
- 기울기, 가속도를 측정하는 데 적합함
- MPU6050 모듈은 가속도 3축, 자이로 3축, 온도센터의 7가지의 센서를 포함(온도센서가 있는 이유는 가속도 자이로 값은 온도에 영향을 받게 되는데 일반적인 온도에서는 크게 영향이 없기에 해당 측정값은 사용되지 않음)
- 게임 및 가상현실 입력장치, GPS네비게이션, 로봇시스템, 드론 등에 유용

### ■ 자이로 가속도 센터(MPU6050)을 활용한 걸음수 설정

#### 1. 설정

- ◎ sudo raspi-config 설정  
→ 설정창에서 I2C 라는 목록 클릭 후 “YES” 선택
- ◎ sudo vi /etc/modules 입력  
→ i2c-bcm2708 한줄 아래 i2c-dev 입력 후 저장

#### 2. 관련 파일들을 설치

- ◎ sudo apt-get install i2c-tools 입력
- ◎ sudo apt-get install python-smbus 입력

#### 3. 자이로 가속도 센서와 라즈베리파이 연결

- ◎ VCC, GND, SDA(데이터를 주고받는 역할), SCL(동기화를 위한 Clock 역할) 연결

#### 4. 연결테스트

- ◎ sudo i2cdetect -y 0(라즈베리파이 버전 2 미만)
- ◎ sudo i2cdetect -y 1(라즈베리파이 버전 2 이상)

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- 68 -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

< 출력 화면 - 정상 연결 >

## 5. 소스코드 입력

© sudo vi mpu6050\_test1.py

```
#-*-coding:utf-8 -*- #python3 버전 이하는 한글 인식이 안 되기 때문에 입력
import smbus
import math
import time
import urllib.request

#전원 관리 레지스터
power_mgmt_1=0x6b
power_mgmt_2=0x6c
step_flag=0 #걸음을 인식하는 동작을 할 경우를 저장하는 플래그 변수
sum_step=0 # 걸음 수(현재 걸음 수로 저장하나 엔드디바이스 개발 시 신호만 보
내서 어플에서 카운트

def read_byte(adr):
    return bus.read_byte_data(address,adr)

def insertCloud(data):
    api_key=' UGV6IKHB2C3WDIDT' #write API Key
    url=' https://api.thingspeak.com/update'
    url=url+' ?api_key=%s' % api_key
    url=url+' &field1=%s' % data

    urllib.request.urlopen(url)

def read_word(adr):
    high=bus.read_byte_data(address, adr)
    low=bus.read_byte_data(address, adr+1)
    val=(high<<8)+low
    return val

def read_word_2c(adr):
    val=read_word(adr)

    if(val>=0x8000):
        return -((65535 - val) +1)

    else:
        return val
```

```

def dist(a,b):
    return math.sqrt((a*a)+(b*b))

def get_y_rotation(x,y,z):
    radians=math.atan2(x, dist(y,z))
    return -math.degrees(radians)

def get_x_rotation(x,y,z):
    radians=math.atan2(x, dist(y,z))
    return -math.degrees(radians)

bus=smbus.SMBus(1)
address=0x68

bus.write_byte_data(address, power_mgmt_1, 0)

try:
    while True:
        time.sleep(1)
        print "-----"
        print " => X, Y, Z 자이로 센서 데이터"
        print "----- "

        gyro_xout=read_word_2c(0x43)
        gyro_yout=read_word_2c(0x45)
        gyro_zout=read_word_2c(0x47)

        print "gyro_xout: " , gyro_xout, " scaled: ",(gyro_xout/131)
        print "gyro_yout: " , gyro_yout, " scaled: ",(gyro_yout/131)
        print "gyro_zout: " , gyro_zout, " scaled: ",(gyro_zout/131)

        print
        print "=>가속 데이터"
        print "-----"

        accel_xout=read_word_2c(0x3b)
        accel_yout=read_word_2c(0x3d)
        accel_zout=read_word_2c(0x3f)

        accel_xout_scaled=accel_xout / 16384.0
        accel_yout_scaled=accel_yout / 16384.0
        accel_zout_scaled=accel_zout / 16384.0

```

```

    print "accel_xout: ", accel_xout, " scaled: " , accel_xout_scaled
    print "accel_yout: ", accel_yout, " scaled: " , accel_yout_scaled
    print "accel_zout: ", accel_zout, " scaled: " , accel_zout_scaled
    print "x rotation: " , get_x_rotation(accel_xout_scaled, accel_yout_scaled,
accel_zout_scaled)
    print "y rotation: " , get_y_rotation(accel_xout_scaled, accel_yout_scaled,
accel_zout_scaled)
    print "-----"

    if accel_xout>0 and step_flag==0:
        step_flag=1
    elif accel_xout <0 and step_flag ==1:
        step_flag=0
        sum_step=sum_step +1

    print(step_flag)
    print "내 걸음 수 : ", sum_step

insertCloud(10)

except KeyboardInterrupt:
    GPIO.cleanup()

```

## 6. 실행

© python mpu6060\_test1.py

<자료 출처>

1. <https://m.blog.naver.com/dlwjddns5/220725348476>
2. <https://blog.naver.com/3246902/221965791915>