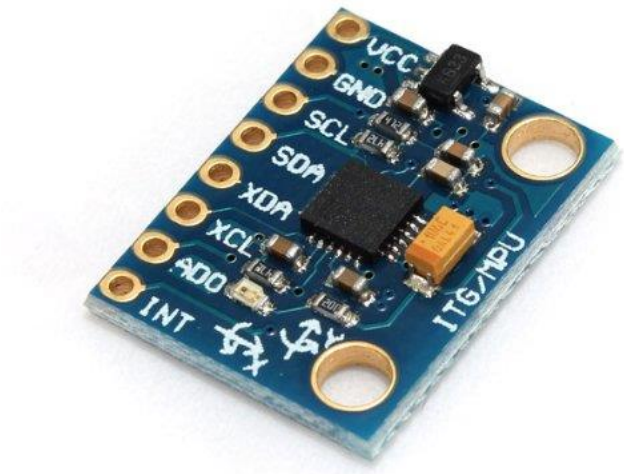
Practical Aim:- Interface with an Accelerometer Gyro Mpu6050 on the i2c bus and send sensor values over the internet via mqtt.
The Gyro Meter look like this:-



Measuring Rotation and acceleration with the Raspberry Pi:-

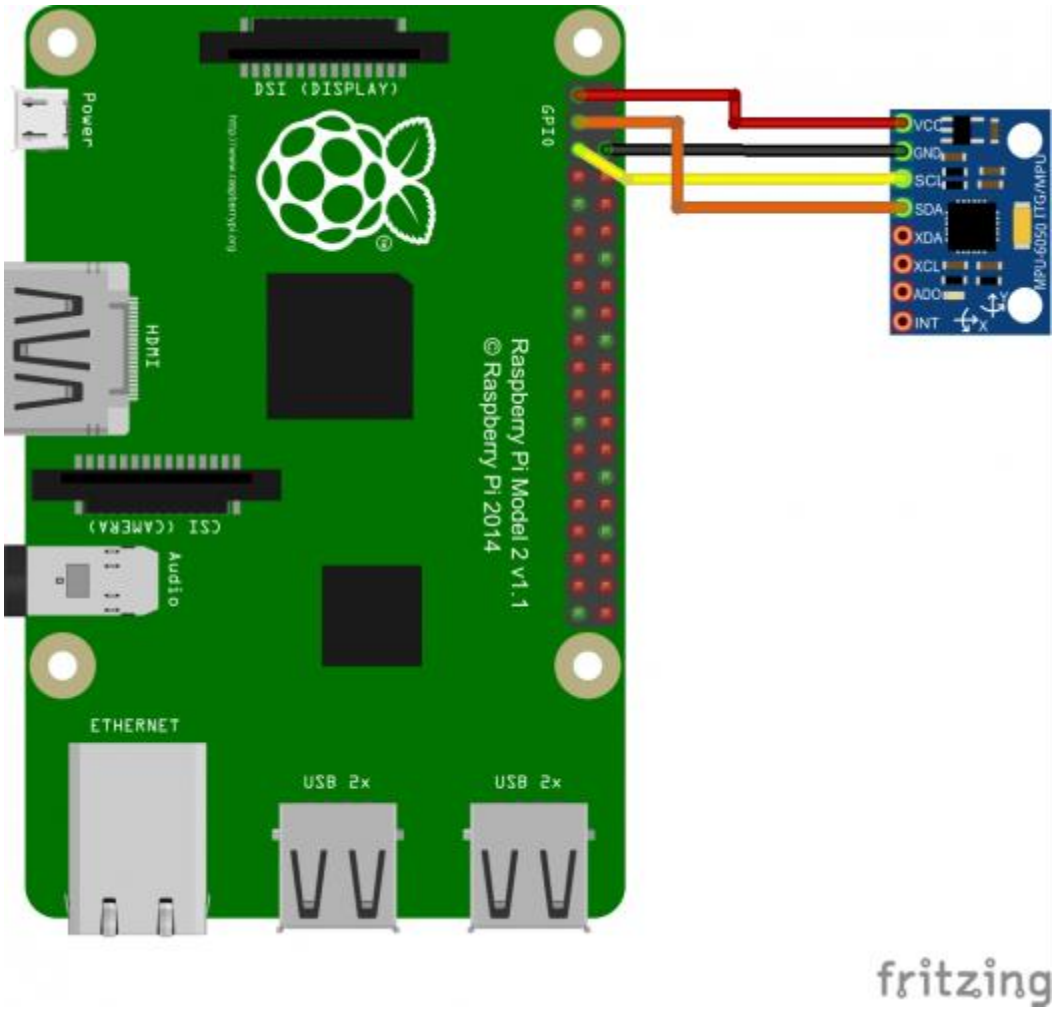
Acceleration and rotation sensors are most known from smartphones. The rotation of the device can be detected and can be addressed directly.
With the Raspberry Pi and a Gyroscope / Accelerator sensor it is possible to measure the rotation as well as the acceleration of the 3 axes.

Setup

Raspberry Pi	MPU 6050
Pin 1 (3.3V)	VCC
Pin 3 (SDA)	SDA

Pin 5 (SCL)	SCL
Pin 6 (GND)	GND

The practical thing about I2C is that very few pins are used. Of the eight pins of the sensor, we only need to connect the upper 4:



Activate I²C on the Raspberry Pi

First we enable SPI and I2C. If you have already done this in a previous tutorial, you can skip this step.

```
sudo raspi-config
```

We then edit the modules file:

```
sudo nano /etc/modules
```

If the following lines are not already included, add them and restart the Pi (`sudo reboot`):

Shell

1 i2c-bcm2708

2 i2c-dev

Now we can quickly install the necessary tools:

```
sudo apt-get install i2c-tools python-smbus
```

Let's start a small test. The parameter `-y 1`stands for revision 2. If you have a completely old Pi (before 2013), you would have to specify a 0 instead:

```
sudo i2cdetect -y 1
```

If the gyroscope is properly connected, you will see this output (if you have other I2C modules connected, their hex addresses should be displayed):

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1

 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: -- -- -- -- -- -- -- --
```

To address the device with address 68 (Attention: Hexadecimal) under the register, enter the following:

```
sudo i2cget -y 1 0x68 0x75
```

Read the MPU-6050 Modul

The most convenient way to read the acceleration sensor is probably Python. Therefore, we create a file and paste the following code.

```
sudo nano gyro.py
```

```
#!/usr/bin/python

import smbus

import math


# Register

power_mgmt_1 = 0x6b

power_mgmt_2 = 0x6c


def read_byte(reg):

    return bus.read_byte_data(address, reg)


def read_word(reg):

    h = bus.read_byte_data(address, reg)

    l = bus.read_byte_data(address, reg+1)

    value = (h << 8) + l

    return value
```

```
def read_word_2c(reg):
    val = read_word(reg)
    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(y, dist(x,z))
    return math.degrees(radians)

bus = smbus.SMBus(1) # bus = smbus.SMBus(0) fuer Revision 1
address = 0x68       # via i2cdetect

# Aktivieren, um das Modul ansprechen zu koennen
bus.write_byte_data(address, power_mgmt_1, 0)

print "Gyroskop"
```

```
print "--------"

 gyroskop_xout = read_word_2c(0x43)

gyroskop_yout = read_word_2c(0x45)

gyroskop_zout = read_word_2c(0x47)

print "gyroskop_xout: ", ("%5d" % gyroskop_xout), " skaliert: ", (gyroskop_xout / 131)

print "gyroskop_yout: ", ("%5d" % gyroskop_yout), " skaliert: ", (gyroskop_yout / 131)

print "gyroskop_zout: ", ("%5d" % gyroskop_zout), " skaliert: ", (gyroskop_zout / 131)

print "Beschleunigungssensor"

print "---------------------"


beschleunigung_xout = read_word_2c(0x3b)

beschleunigung_yout = read_word_2c(0x3d)

beschleunigung_zout = read_word_2c(0x3f)


beschleunigung_xout_skaliert = beschleunigung_xout / 16384.0

beschleunigung_yout_skaliert = beschleunigung_yout / 16384.0

beschleunigung_zout_skaliert = beschleunigung_zout / 16384.0


print "beschleunigung_xout: ", ("%6d" % beschleunigung_xout), " skaliert: ",
beschleunigung_xout_skaliert

print "beschleunigung_yout: ", ("%6d" % beschleunigung_yout), " skaliert: ",
beschleunigung_yout_skaliert

print "beschleunigung_zout: ", ("%6d" % beschleunigung_zout), " skaliert: ",
beschleunigung_zout_skaliert


print "X Rotation: " , get_x_rotation(beschleunigung_xout_skaliert, beschleunigung_yout_skaliert,
beschleunigung_zout_skaliert)
```

```
print "Y Rotation: " , get_y_rotation(beschleunigung_xout_skaliert, beschleunigung_yout_skaliert, beschleunigung_zout_skaliert)
```

Save it with CTRL + O and exit the editor with CTRL + X. You can then run the script.

```
sudo python gyro.py
```

Now you'll see an output that contains all the captured data:

```
Gyroskop

--------

gyroskop_xout:   -260  skaliert:  -2

gyroskop_yout:   -154  skaliert:  -2

gyroskop_zout:     78  skaliert:  0


Beschleunigungssensor

---------------------

beschleunigung_xout:   -1048  skaliert:  -0.06396484375

beschleunigung_yout:    -676  skaliert:  -0.041259765625

beschleunigung_zout:   16644  skaliert:  1.01586914062

X Rotation:  -2.32121150537

Y Rotation:  3.59994842011
```