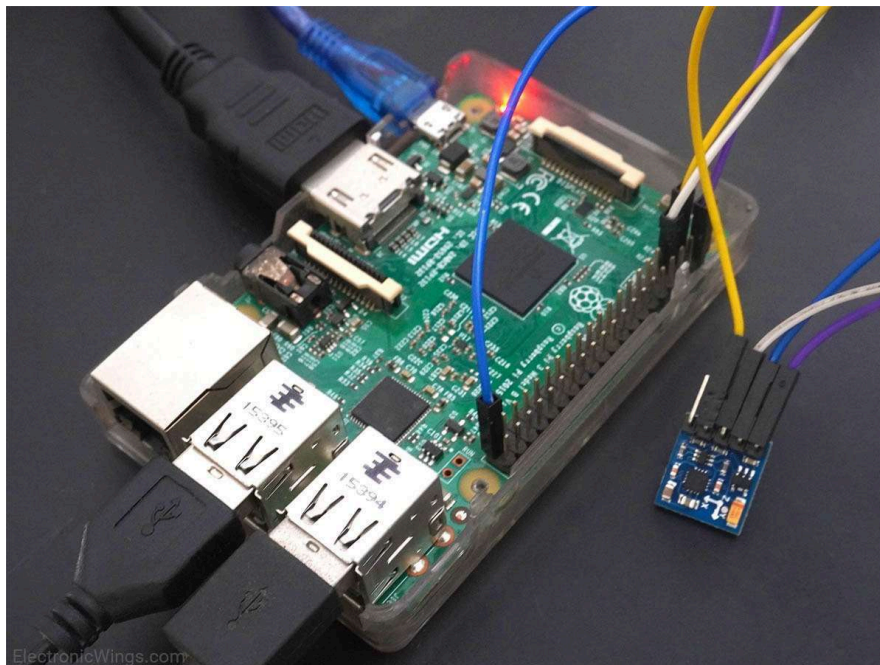


Triple Axis Magnetometer HMC5883L Interfacing with Raspberry Pi



Overview of Magnetometer HMC5883L



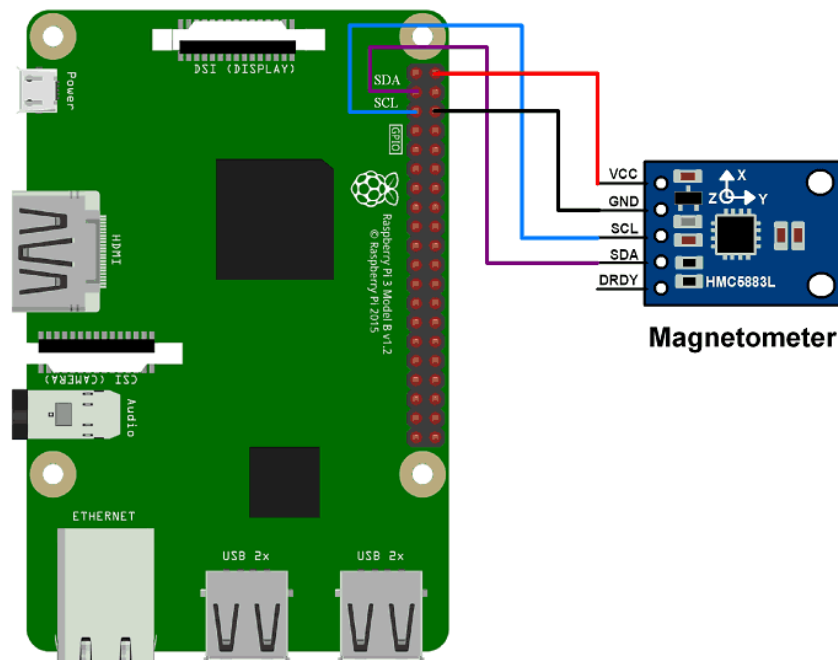
HMC5883L Magnetometer Module

- Magnetometer HMC5883L is used for measuring the direction and magnitude of the Earth's magnetic field. It is used for low cost compassing and magnetometry.
- It measures the Earth's magnetic field value along the X, Y, and Z axes from milli-gauss to 8 gauss.
- It can be used to find the direction of the heading of the device.
- It uses the I2C protocol for communication with the microcontroller.

For more information about Magnetometer HMC5883L and how to use it, refer the topic **HMC5883L Magnetometer Module** (<http://www.electronicwings.com/sensors-modules/hmc5883l-magnetometer-module>) in the sensors and modules section.

To interface the HMC5883L magnetometer module with Raspberry Pi, we should ensure that the I2C protocol on Raspberry Pi is enabled. So before going for interfacing HMC5883L with Raspberry Pi, we need to make some I2C configurations on Raspberry Pi for which you can refer **Raspberry Pi I2C** (<http://www.electronicwings.com/raspberry-pi/raspberry-pi-i2c>).

Connection Diagram of HMC5883L Magnetometer with Raspberry Pi



HMC5883L Magnetometer Interfacing with Raspberry Pi 3

Get the angle in degrees using HMC5883L Magnetometer and Raspberry Pi

Reading magnetic field strength along the x, y, and z axes from the HMC5883L magnetometer module, calculate the heading angle from these three axes and displaying heading angle on Serial Monitor.

Let's interface magnetometer HMC5883L with Raspberry Pi and calculate its heading angle. HMC5883L uses the I2C protocol for communication. We can interface it with Raspberry by using Python.

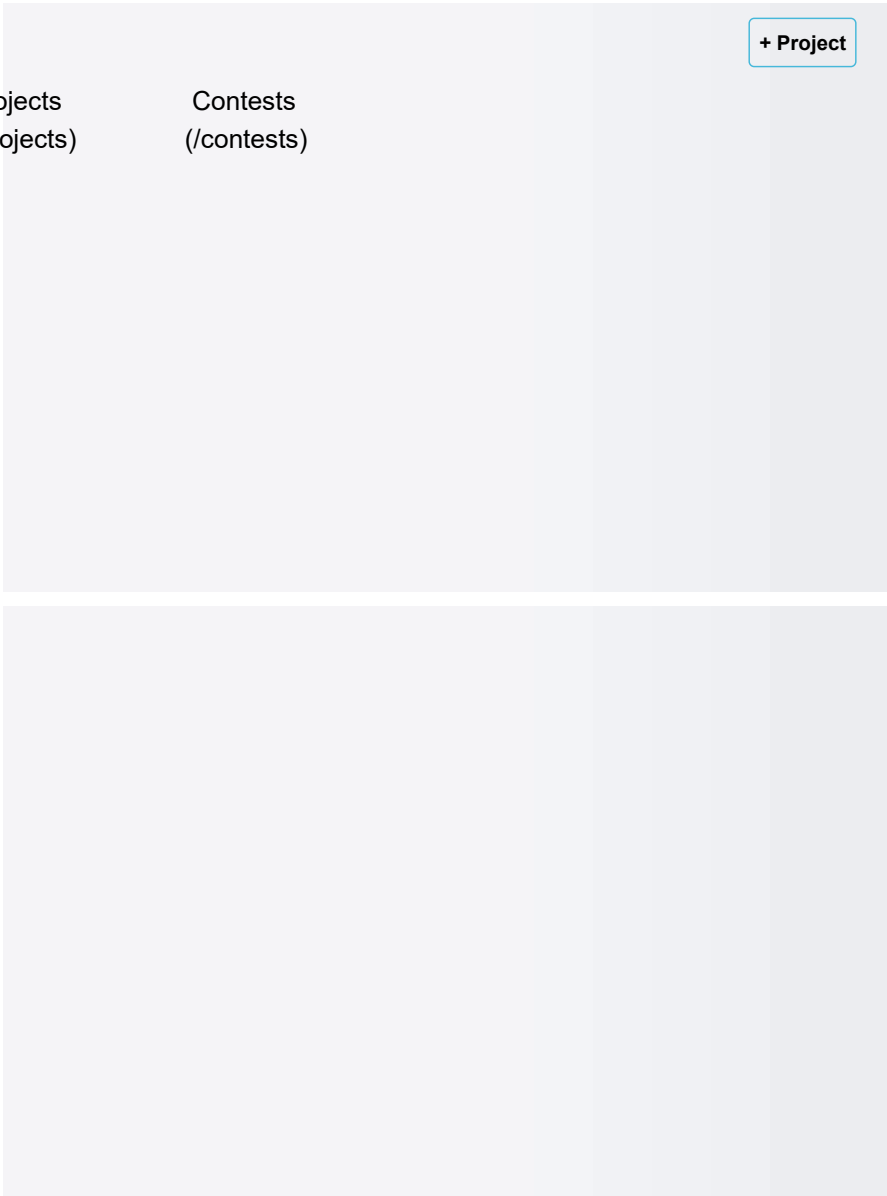
To know about Python based I2C functions used on Raspberry Pi, you can refer **Python based I2C functions for Raspberry Pi**
(<http://www.electronicwings.com/raspberry-pi/python-based-i2c-functions-for-raspberry-pi>)

Programming steps

- First we need to set configuration register A for average of 8-sample measurement with 15 Hz default data output rate
- Set Gain using Configuration Register B i.e. here its 0xA0. (or we can choose any other desired gain)
- Select Continuous measurement mode of operation in Mode Register. Hence Mode Register value will become 0x00.

After initialization, read X, Y and Z-axis registers raw values.

Calculate Heading value by using following formula,



HMC5883L Python Program for Raspberry Pi



Platforms

[\(/explore\)](#)

Projects

[\(/projects\)](#)

Contests

[\(/contests\)](#)

Find Heading by using HMC5883L interface with Raspberry Pi using Python

<http://www.electronicwings.com>[+ Project](#)[\(/login#login\)](#)

```

import smbus          #import SMBus module of I2C
from time import sleep #import sleep
import math

#some MPU6050 Registers and their Address
Register_A    = 0           #Address of Configuration register A
Register_B    = 0x01        #Address of configuration register B
Register_mode  = 0x02        #Address of mode register

X_axis_H      = 0x03         #Address of X-axis MSB data register
Z_axis_H      = 0x05         #Address of Z-axis MSB data register
Y_axis_H      = 0x07         #Address of Y-axis MSB data register
declination = -0.00669       #define declination angle of location whe
pi            = 3.14159265359 #define pi value

def Magnetometer_Init():
    #write to Configuration Register A

```

HMC5883L Output

EW (/)

Platforms (/explore)

Projects (/projects)

Contests (/contests)

+ Project

(/login#login)

EW

Platforms

[\(/explore\)](#)

Projects

[\(/projects\)](#)

Contests

[\(/contests\)](#)

+ Projects

[\(/login#login\)](#)

Components Used

Powered by 

 (/)

Platforms

[\(/explore\)](#)

Projects

(/projects)


Contests

(/contests)

HMC5883L Magnetometer Module

Magnetometer HMC5883L is developed by Honeywell...


X 1


 (https://www.mouser.in/ProductDetail/Olinex-Ltd/MOD-HMC5883L?qs=%2Fha2pyFaduiM2FizGGE3eZs8JvW%2Fsm6fbEJBqFTwYwfN63cbOSCmqA%3D%3D&utm_source=electronicwings&utm_medium=referral&utm_campaign=mouser-componentlisting)

Raspberry Pi 4B

Raspberry Pi 4BRaspberry Pi 4B

X 1


 (https://www.mouser.com/ProductDetail/Raspberry-Pi/RPI4-MODBP-8GB-BULK?qs=sPbYRqrBIVlrjhMid19vUA%3D%3D&utm_source=electronicwings&utm_medium=referral&utm_campaign=mouser-componentlisting)

 Datasheet (https://www.mouser.com/ProductDetail/Raspberry-Pi/RPI4-MODBP-8GB-BULK?qs=sPbYRqrBIVlrjhMid19vUA%3D%3D&utm_source=electronicwings&utm_medium=referral&utm_campaign=mouser-componentlisting)

Raspberry Pi Zero

Raspberry Pi Zero

X 1

 (https://www.mouser.in/ProductDetail/Seeed-Studio/102110357?qs=sGAepiMZZMtyU1cDF2RqUKjVR%252BsHljvWiLJcsapOR6w%3D&utm_source=electronicwings&utm_medium=referral&utm_campaign=mouser-componentlisting)



Platforms
(/explore)

Downloads

[+ Project](#)
[\(/login#login\)](#)

Projects
(/projects)

HMC5883 Interface with Raspberry using Python

Contests
(/contests)

[Download](#)

Comments

Please login/signup to comment and reply.

HG8

(/users/HG8/profile)
2018-07-06 05:44:25

Hi I need help with this code. I can't read any values of x, y or z axis. Furthermore, I only get 0's instead of any value. Here is the code (I changed the device adress to 0x0d, I got this with previous codes I tried to implement):

```
"Python 3.5.3 (default, Jan 19 2017, 14:11:04) "
```

```
"[GCC 6.3.0 20170124] on linux"
```

```
import smbus #import SMBus module of I2C
```

```
from time import sleep #import sleep
```

```
import math
```

```
#some MPU6050 Registers and their Address
```

```
Register_A = 0 #Address of Configuration register A
```

```
Register_B = 0x01 #Address of configuration register B
```

```
Register_mode = 0x02 #Address of mode register
```

```
X_axis_H = 0x03 #Address of X-axis MSB data register
```

```
Z_axis_H = 0x05 #Address of Z-axis MSB data register
```

```
Y_axis_H = 0x04 #Address of Y-axis MSB data register
```

```
declination = 0.07389 #define declination angle (in radians) of location where  
measurement going to be done
```

```
pi = 3.14159265359 #define pi value
```

```
def Magnetometer_Init():
```

```
#write to Configuration Register A
```

```
bus.write_byte_data(Device_Address, Register_A, 0x70)
```

```
#Write to Configuration Register B for gain
```

```
bus.write_byte_data(Device_Address, Register_B, 0xa0)
```

```
#Write to mode Register for selecting mode
```

```
bus.write_byte_data(Device_Address, Register_mode, 0)
```

```
def read_raw_data(addr):
```

```
#Read raw 16-bit value
```

```
high = bus.read_byte_data(Device_Address, addr)
```

```
low = bus.read_byte_data(Device_Address, addr+1)
```

```
#concatenate higher and lower value
```

```
value = ((high << 8) | low)
```

```
#to get signed value from module
```

```
if(value > 32768):
```

```
value = value - 65536
```

```
return value
```

```
bus = smbus.SMBus(1) # or bus = smbus.SMBus(0) for older version boards
```



Platforms

[\(/explore\)](#)

Device_Address = 0x0d # HMC5883L magnetometer device address

[+ Project](#)[\(/login#login\)](#)

Projects **Contests**

[\(/projects\)](#) [\(/contests\)](#)

```
Magnetometer_Init() # initialize HMC5883L magnetometer
print (" Reading Reading Angle")
```

```
while True:
```

```
#Read Accelerometer raw value
x = read_raw_data(X_axis_H)
z = read_raw_data(Z_axis_H)
y = read_raw_data(Y_axis_H)
```

```
heading = math.atan2(y, x) + declination
```

```
#Due to declination check for >360 degree
if(heading > 2*pi):
    heading = heading - 2*pi
```

```
#check for sign
if(heading < 0):
    heading = heading + 2*pi
```

```
#convert into angle
heading_angle = float(heading * 180/pi)
```

```
print ("Heading angle = %0.1f°" %heading_angle)
print (x)
print (y)
sleep(1)
```

What I obtain is this:

```
Heading angle = 4.2°
0
0
Reply Like
```

parisotericmail
[\(/users/parisotericmail/profile\)](#)
 2018-11-03 20:52:53

hi I had the same problem and it looks like the HMC5883L is actually a QMC5883L, working code can be found here :
https://github.com/RigacciOrg/py-qmc5883/blob/master/py_qmc5883/__init__.py
 Reply Like

UsamaAmjad
[\(/users/UsamaAmjad/profile\)](#)
 2020-08-19 17:27:52

hello...! i am using raspberry pi 3 b+ and trying to get value from HM5883L in python code but i am getting this error.
 bus.write_byte_data(Device_Address, Register_A, 0x70)
 OSError: [Errno 121] Remote I/O error
 I tried multiple times but getting same error even after changing wires.
 I need help for this thanku.
 Reply Like

AxxelUtime
[\(/users/AxxelUtime/profile\)](#)
 2020-09-14 02:59:41

Hello, I think we all have the same problem (see reply of the other post), we received a QMC5883L that works on different addresses and with a different code (I had to change Device_Address to 0x0d)
 More infos: <https://www.raspberrypi.org/forums/viewtopic.php?t=172244>
 Reply Like



Platforms

[\(/explore\)](#)

NeftaliGonzalez

(/users/NeftaliGonzalez/profile)
2021-04-08 20:52:34[+ Project](#)[\(/login#login\)](#)

Projects if the previous contest is run an displays error "OSError: [Errno 121] Remote I/O error" you may have a QMC5883L instead, corrected code is here:

Contests (/contests)

```
import smbus #comunicación I2C
from time import sleep
import math

#direcciones requeridas
#-----config
Registro_A = 0x0B
Registro_B = 0x09
RegStatus = 0x06
RegCtrl = 0x09
#-----direcciones de la conexión
bus=smbus.SMBus(1)
deviceAdress = 0x0d
#-----datos
eje_X_Mag = 0x00
eje_Y_Mag = 0x02
eje_Z_Mag = 0x04
declination = -0.00669
pi=3.14159265359

def MagnetometerInit():
    #configurar registro A
    bus.write_byte_data(deviceAdress, Registro_A, 0x01)
    #configurar registro B
    bus.write_byte_data(deviceAdress, Registro_B, 0x1D)
    #configurar registro para seleccionar el modo
    #bus.write_byte_data(deviceAdress, ModoRegistro, 0)

def read_raw_data(addr):
    #leer doble byte (16 bits)
    low = bus.read_byte_data(deviceAdress, addr)
    high = bus.read_byte_data(deviceAdress, addr+1)
    #concatenar los bytes
    valor = ((high << 8) | low)
    #obtener el signo
    if(valor > 32768):
        valor = valor - 65536
    return valor
#-----main
MagnetometerInit()
print('leyendo magnetometro...')
while True:
    bandera = bus.read_byte_data(deviceAdress, RegStatus)
    a="{0:b}".format(bandera)
    if a[len(a)-1] == 0:
        bandera = bus.read_byte_data(deviceAdress, RegStatus)
    x=read_raw_data(eje_X_Mag)
    y=read_raw_data(eje_Y_Mag)
    z=read_raw_data(eje_Z_Mag)
    heading = math.atan2(y,x)+ declination
    #compensar superiores a 360
    if(heading > 2*pi):
        heading = heading - 2*pi
    #revisar el signo
    if(heading < 0):
        heading=heading+2*pi
    #convertir a grados
    heading_angle = int(heading * (180/pi))
    print("angulo = %d°" %heading_angle)
    sleep(0.5)
```

