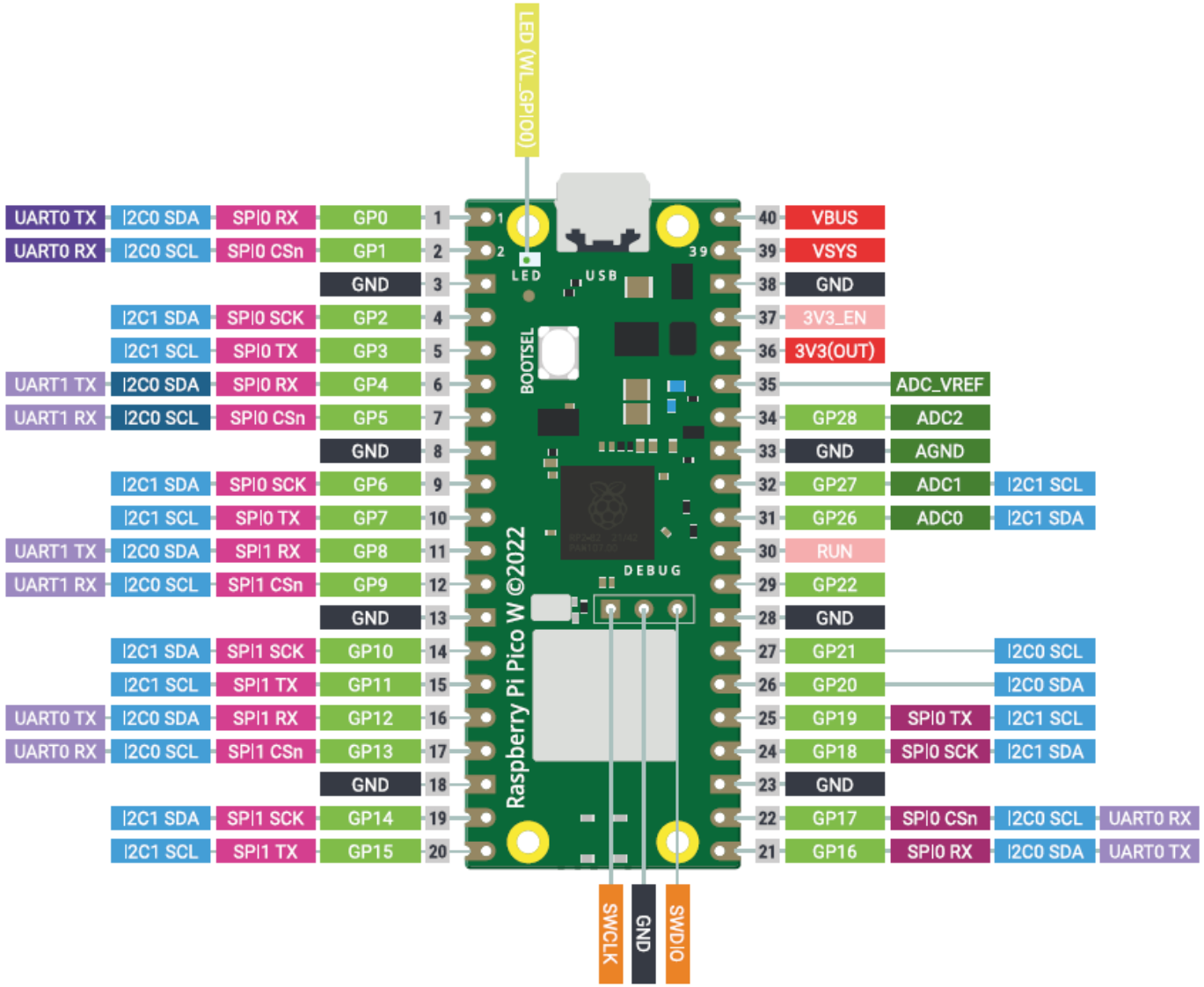
Application of GY-87 IMU Module

Home Projects | RPI Pico W Specification



RP2040

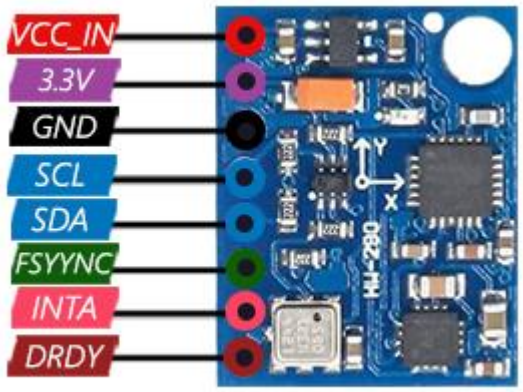
- Power
- Ground
- UART / UART (default)
- GPIO, PIO, and PWM
- ADC
- SPI / SPI (default)
- I2C / I2C (default)
- System Control
- Debugging

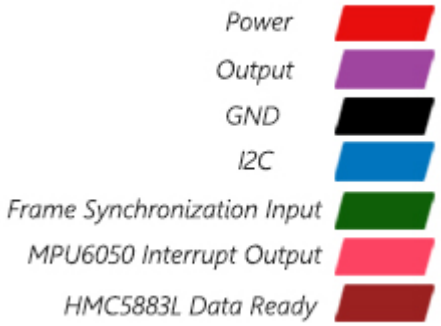
Infineon 43439

- GPIO

- RAM : 256K
- ROM : 2M
- Clock speed: 133MHz
- WiFi: 2.4GHz 802.11 b/g/n
- BT 5.2

Home Projects | GY-87 IMU Module



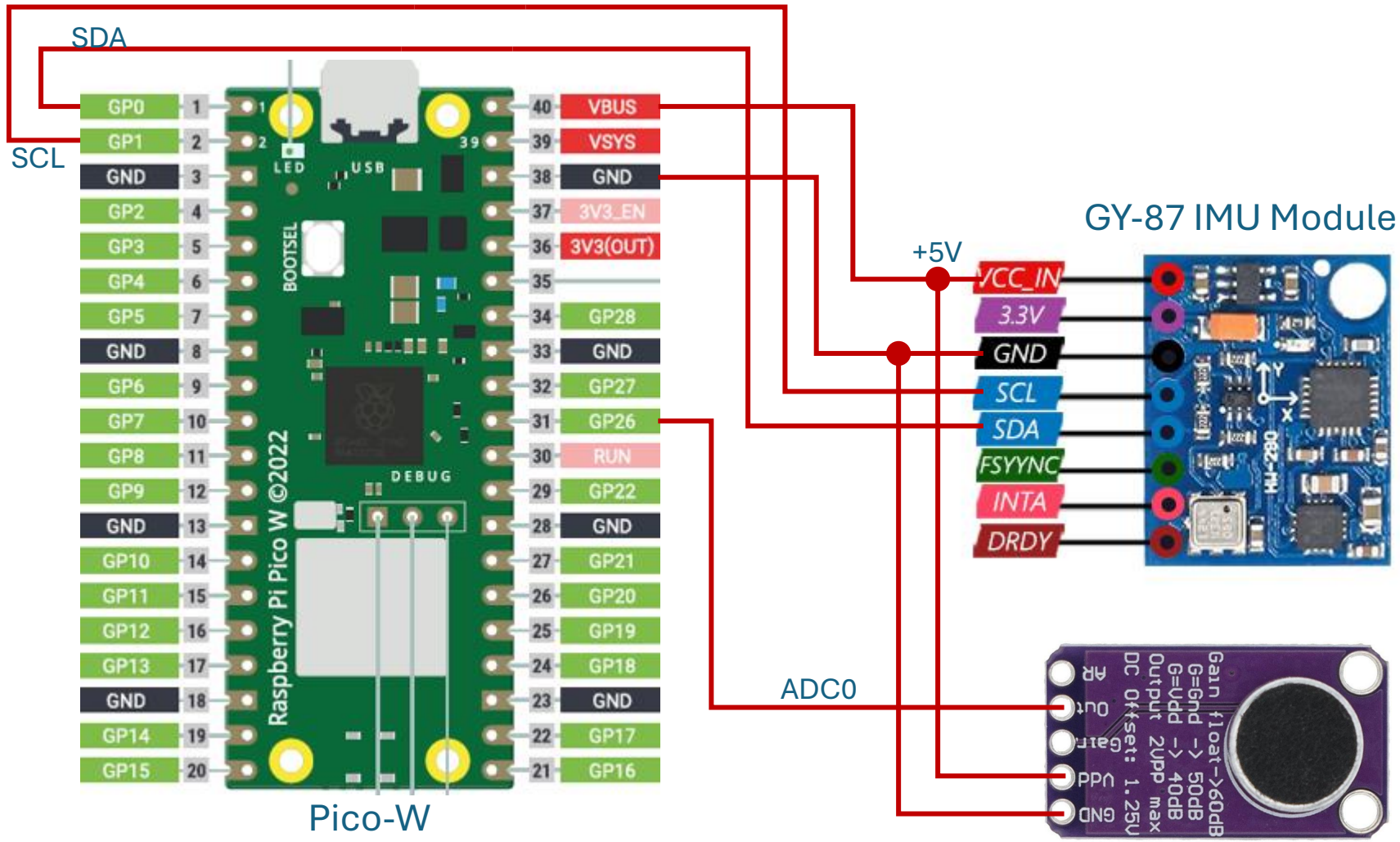


- BMP180 (0x77) Bosch Sensortec air pressure sensor
  - Raspberry Pi Pico and BMP180 Sensor : 4 Steps – Instructables
  - Interface BMP180 Air Pressure Sensor with Raspberry Pi 4 - The Engineering Projects
  - https://github.com/ramjipatel041/Raspberry-Pi-Pico-and-BMP180
- QMC5883L (0x0D) 3-axis digital compass (Not compatible with HMC5883L)
  - QMC5883L in Micropython - Raspberry Pi Forums
- MPU6050 (0x68) Accelerometer and Gyroscope
  - How to Use MPU6050 With Raspberry Pi Pico or Pico W – Instructables
  - https://github.com/shillehbean/youtube-channel/blob/main/vector3d.py
  - https://github.com/shillehbean/youtube-channel/blob/main/imu.py
- Misc. tools
  - Raspberry Pi Pico: I2C Scanner (MicroPython) | Random Nerd Tutorials

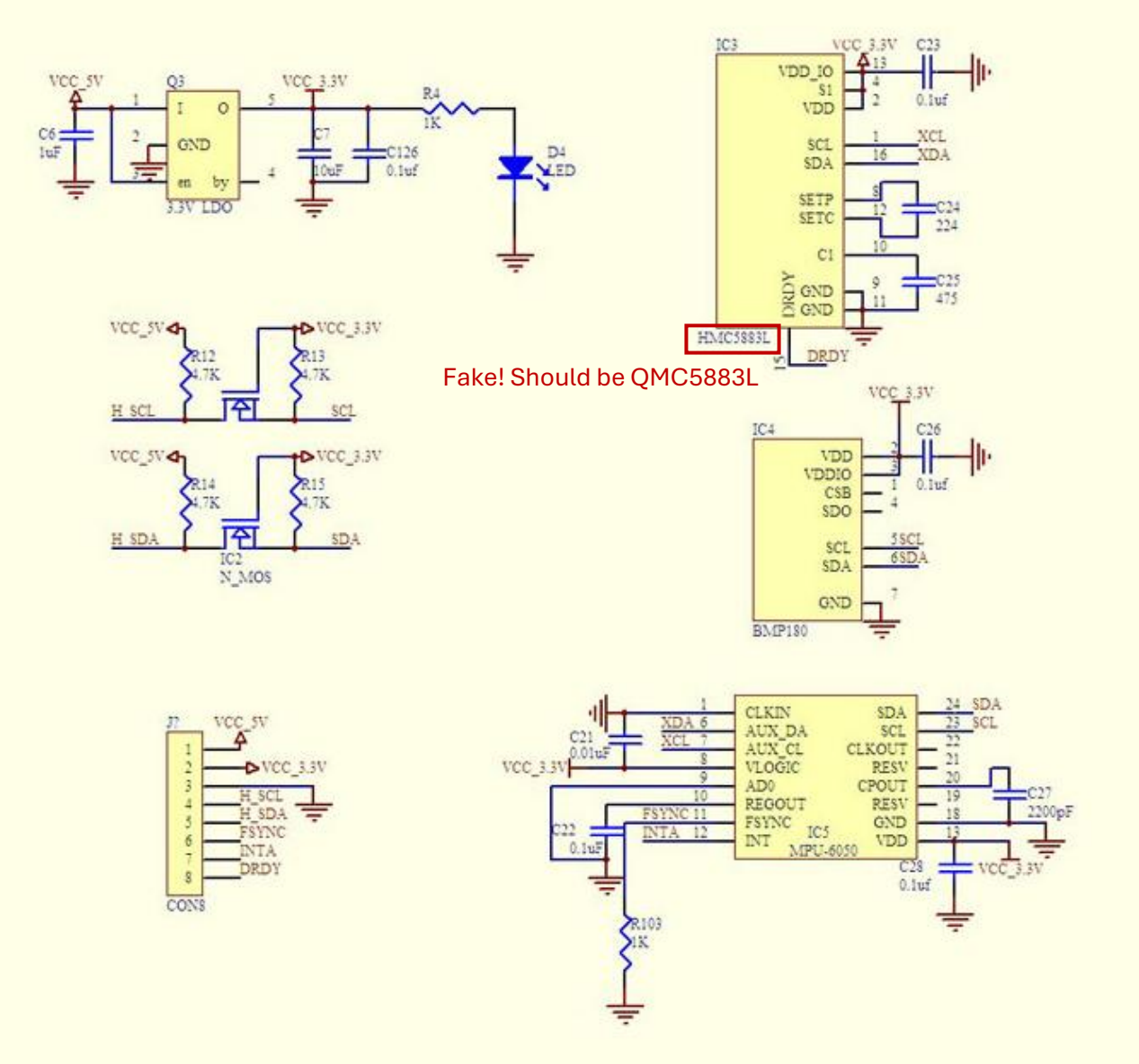
Home Projects | MAX9814 Microphone module

- Reference link
  - Clap Switch With PICO 2 and MAX9814 : 5 Steps (with Pictures) – Instructables
  - How to use ADC in Raspberry Pi Pico | ADC Example Code

Home Projects | Schematic



Home Projects | GY-87 IMU Module - Schematic



- For MPU6050, AD0 is tied to LOW. So, the device ID (read value from WHO_AM_I register (0x75) will be 0x34. If AD=High, ID will be 0x68

- For the I2C connection of QMC5883L, it is connected to the AUX I2C of MPU6050. So, to access the QMC chip via MPU, the following MPU registers should be set as below:-
  - Register 0x37's bit 2 = High (I2C_BYPASS_EN)
  - Register 0x6A's bit 1 = Low (I2C_MST_RESET)
  - Detail please see next slide and/or MPU6050 register map.
  - Here below is the example MicroPython® code to do this action:-

```
print ('Unlock the HMC5883L IIC address')
buf = bytearray(2)
buf[0]=2
i2c.writeto_mem( addr: 0x68,  memaddr: 0x37, buf)
buf[0]=0
i2c.writeto_mem( addr: 0x68,  memaddr: 0x6A, buf)

reg_0x37 = i2c.readfrom_mem( addr: 0x68,  memaddr: 0x37,  nbytes: 1)
print(f'read reg 0x37, content = 0x{reg_0x37.hex()}')

print('I2C SCANNER')
devices = i2c.scan()
```

- After running this, the I2C address of QMC (0x0D) will be visible via i2c.scan()

Home Projects | MPU key registers for by-passing QML I2C address

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
37	55	INT_LEVEL	INT_OPEN	LATCH _INT_EN	INT_RD _CLEAR	FSYNC_ INT_LEVEL	FSYNC_ INT_EN	I2C _BYPASS _EN	-

Set to High (0x02)

I2C_BYPASS_EN When this bit is equal to 1 and *I2C_MST_EN* (Register 106 bit[5]) is equal to 0, the host application processor will be able to directly access the auxiliary I^2C bus of the MPU-60X0.

When this bit is equal to 0, the host application processor will not be able to directly access the auxiliary I^2C bus of the MPU-60X0 regardless of the state of *I2C_MST_EN* (Register 106 bit[5]).

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6A	106	-	FIFO_EN	I2C_MST _EN	I2C_IF _DIS	-	FIFO _RESET	I2C_MST _RESET	SIG_COND _RESET

Set to Low (0x00)

I2C_MST_EN When set to 1, this bit enables I^2C Master Mode.

When this bit is cleared to 0, the auxiliary I^2C bus lines (AUX_DA and AUX_CL) are logically driven by the primary I^2C bus (SDA and SCL).

Home Projects | Major differences between HMC and QMC

- Registers
  - All register addresses of HMC5883L are different from those of QMC5883L. See detail at RHS.
- I2C address
  - HMC5883L: 0x1E
  - QMC5883L: 0x0D
- Chip ID
  - HMC5883L: ‘H’, ‘4’ or ‘3’ (Three identification registers A, B, C)
  - QMC5883L: 0xFF (Chip ID register)
- ADC
  - HMC5883L: 12-bit ADC
  - QMC5883L: 16-bit ADC

HMC5883L

Address Location	Name	Access
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

QMC5883L

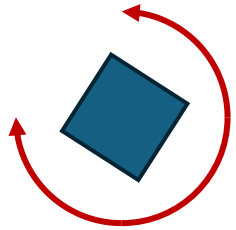
Addr.	7	6	5	4	3	2	1	0	Access
00H	Data Output X LSB Register XOUT[7:0]								Read only
01H	Data Output X MSB Register XOUT[15:8]								Read only
02H	Data Output Y LSB Register YOUT[7:0]								Read only
03H	Data Output Y MSB Register YOUT[15:8]								Read only
04H	Data Output Z LSB Register ZOUT[7:0]								Read only
05H	Data Output Z MSB Register ZOUT[15:8]								Read only
06H						DOR	OVL	DRDY	Read only
07H	TOUT[7:0]								Read only
08H	TOUT[15:8]								Read only
09H	OSR[1:0]		RNG[1:0]		ODR[1:0]		MODE[1:0]		Read/Write
0AH	SOFT_RST	ROL_PNT						INT_ENB	R/W, Read only on blanks
0BH	SET/RESET Period FBR [7:0]								Read/Write
0CH	Reserved								Read only
0DH	Chip ID								Read only

Home Projects | QMC5883L calibration

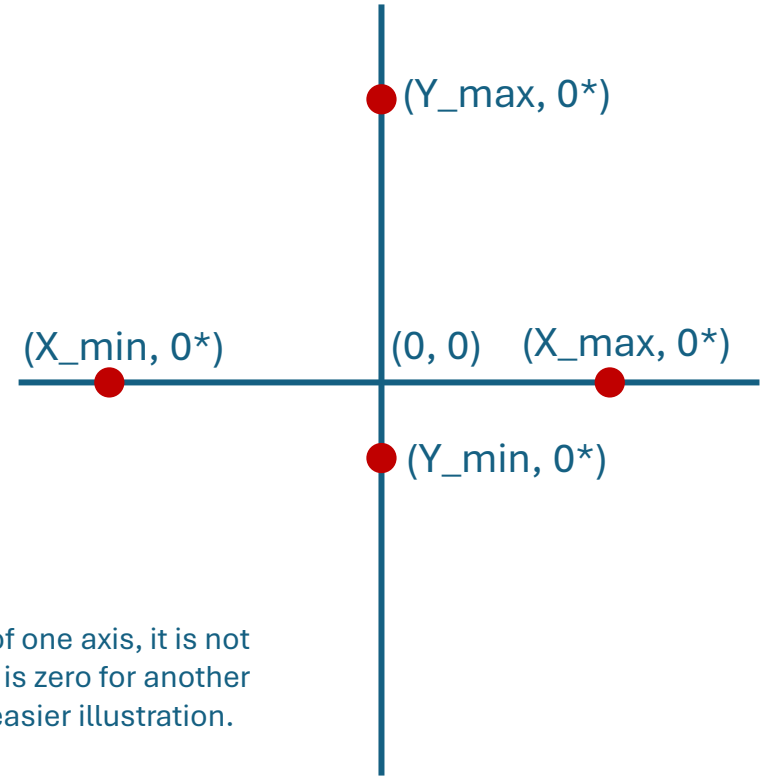
- The heading value is calculated from the values of x, y, z data output registers of the device.
- In reality, due to the non-linearity of the actual device, the calibration process is necessary.
- The procedure of calibration is to look for the minimum and maximum values of x, y, z data. In simply, we just want to get the min./max. values of x, y data. Therefore, we will rotate the device in horizontal plane. That is Z value is fixed.

Home Projects | QMC5883L calibration process

Calibration process



Rotate the device by 360^0 to and fro



* For getting the min/max values of one axis, it is not necessary to make sure the value is zero for another axis. We set zero value is just for easier illustration.

Calibration calculation

Calibration Data

1. X_offset = (X_max + X_min) / 2.0
2. Y_offset = (Y_max + Y_min) / 2.0
3. Y_scale = 1.0
4. X_scale = (Y_max – Y_min) / (X_max – X_min)

Calibrated data X_pos and Y_pos output

1. X_pos = (X_raw – X_offset) * X_scale
2. Y_pos = (Y_raw – Y_offset) * Y_scale

Raw data from normal operation

X_RAW
Y_RAW

Home Projects | QMC5883L measurement stability

- As QMC5883L, the ADC is 16-bit, the sensitivity is much higher than that of HMC5883L which has 12-bit ADC for data conversion.

- However, high sensitivity leads a data getting unstable issue and very difficult to do either calibration or normal heading measurement.

- To solve this issue, two software solutions are deployed:-
  - Simple Moving Average (SMA) filtering for raw data.
  - 'Lower' the sampling 'bit' by >> 4 (right bit shift by 4) to truncate LSB by 4 bit or divided the result by 16.0. Here below is the sample coding to implement them.

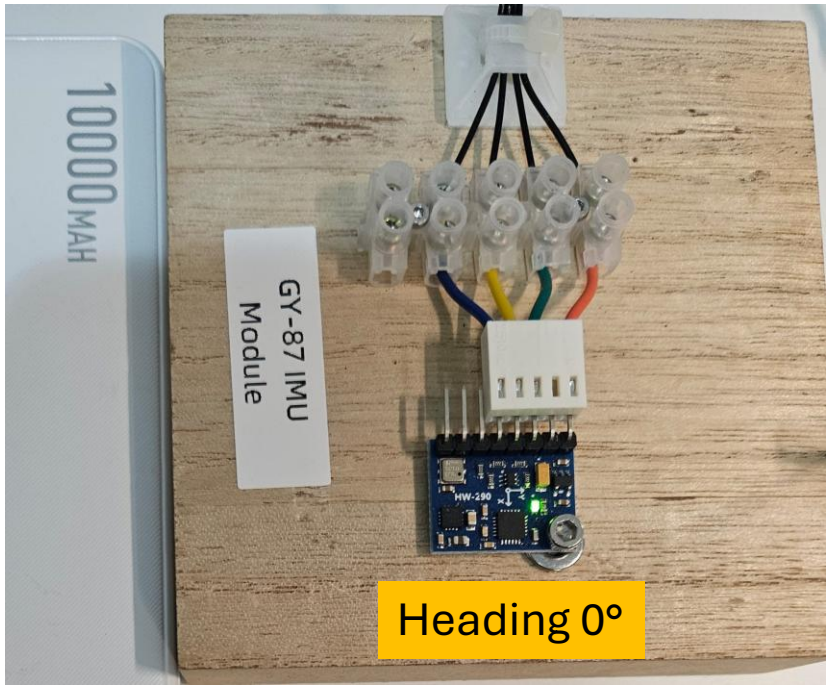
```
def simple_moving_average(self, val: int, array_list: list) -> float:  3 usages
    """
    Moving average
    :param val:
    :param array_list:
    :return:
    """

    sum_val = 0
    # Buffer element filling
    if len(array_list) > self.sma_buffer_size:
        return 0.0
    elif len(array_list) == self.sma_buffer_size: # Buffer full
        array_list.pop(0) # Remove the first element
        array_list.append(val)
    else:
        array_list.append(val)

    for each_element in range(0, len(array_list)):
        sum_val += array_list[each_element]

    return (sum_val / len(array_list)) * 1.0 # multiplied by 1.0 to convert to float
```

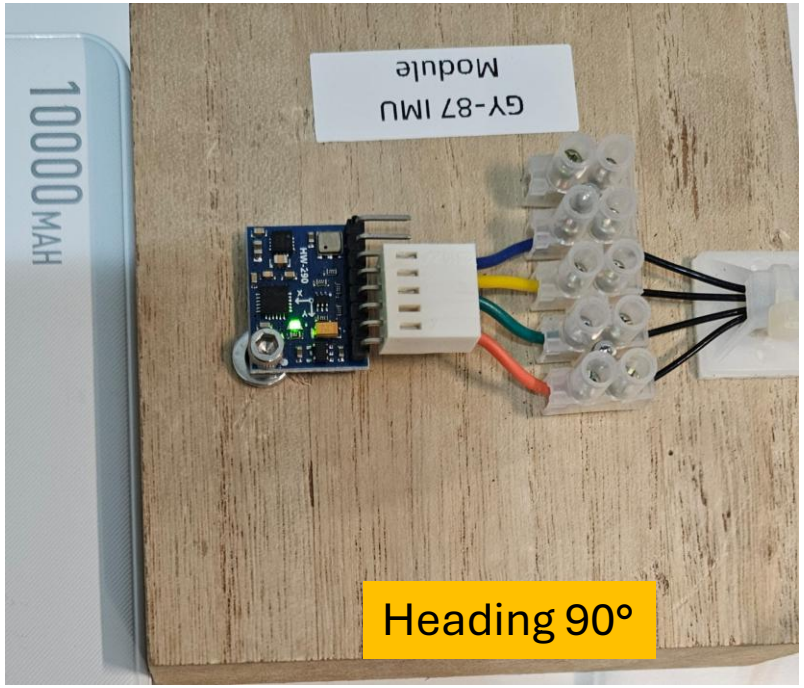
```
#Do SMA
_pos_x_raw_sma = self.simple_moving_average(_pos_x_raw, temp_buf_x) / 16.0
_pos_y_raw_sma = self.simple_moving_average(_pos_y_raw, temp_buf_y) / 16.0
_pos_z_raw_sma = self.simple_moving_average(_pos_y_raw, temp_buf_z) /16.0
```

Home Projects | QMC5883L measurement result

```
MicroPython  REPL
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(328.0321, -10.74496, 169.9756, 358.12)
MPU6050 Data: ax=0.03,ay=-0.01,az=0.99, gx=-9, gy=0, gz=-2, Temperature=27.56C
BMP180 Data: temp=28.4C, Pressure=1002.95hPa, Altitude=86.11m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(327.8129, -9.430603, 171.29, 358.35)
MPU6050 Data: ax=0.03,ay=-0.01,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.52C
BMP180 Data: temp=28.4C, Pressure=1002.95hPa, Altitude=86.11m
QMC5883L Device ID: 0xff
```

```
MicroPython  REPL
BMP180 Data: temp=28.4C, Pressure=1002.98hPa, Altitude=85.86m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(-1.615683, 330.4419, 511.1625, 90.28)
MPU6050 Data: ax=0.03,ay=0.01,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.56C
BMP180 Data: temp=28.4C, Pressure=1002.98hPa, Altitude=85.86m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(-1.844521, 330.035, 510.7556, 90.32)
MPU6050 Data: ax=0.04,ay=0.0,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.66C
BMP180 Data: temp=28.4C, Pressure=1002.98hPa, Altitude=85.86m
QMC5883L Device ID: 0xff
```





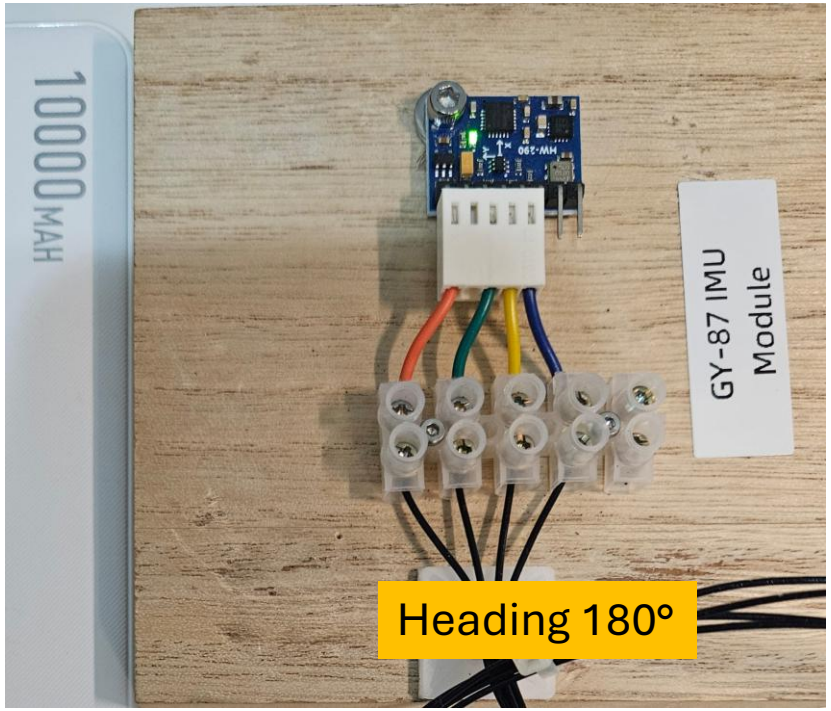
Home Projects | QMC5883L measurement result

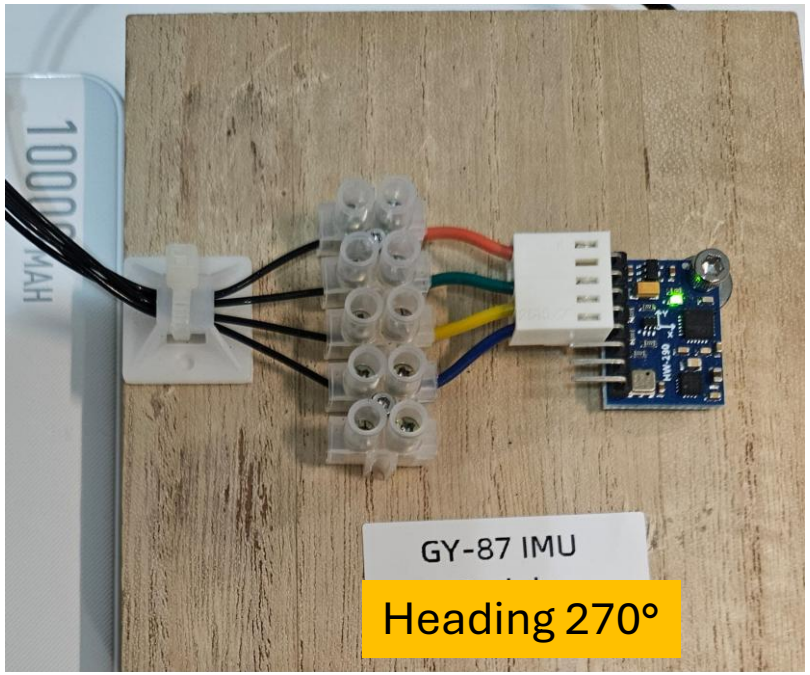
MicroPython REPL

```
BMP180 Data: temp=28.3C, Pressure=1003.01hPa, Altitude=85.61m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(-349.6669, -17.77934, 162.9413, 182.91)
MPU6050 Data: ax=0.01,ay=0.0,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.56C
BMP180 Data: temp=28.3C, Pressure=1003.01hPa, Altitude=85.61m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(-350.289, -21.37497, 159.3456, 183.49)
MPU6050 Data: ax=0.01,ay=0.0,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.56C
BMP180 Data: temp=28.3C, Pressure=1003.04hPa, Altitude=85.35m
```

MicroPython REPL

```
BMP180 Data: temp=28.6C, Pressure=1003.05hPa, Altitude=85.27m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(5.394397, -336.8737, -156.1531, 270.92)
MPU6050 Data: ax=0.02,ay=-0.0,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.71C
BMP180 Data: temp=28.6C, Pressure=1003.05hPa, Altitude=85.27m
QMC5883L Device ID: 0xff
(X,Y,Z, Heading)=(5.587778, -335.7631, -155.0425, 270.95)
MPU6050 Data: ax=0.01,ay=-0.0,az=0.99, gx=-8, gy=0, gz=-2, Temperature=27.75C
BMP180 Data: temp=28.6C, Pressure=1003.08hPa, Altitude=85.02m
```






END