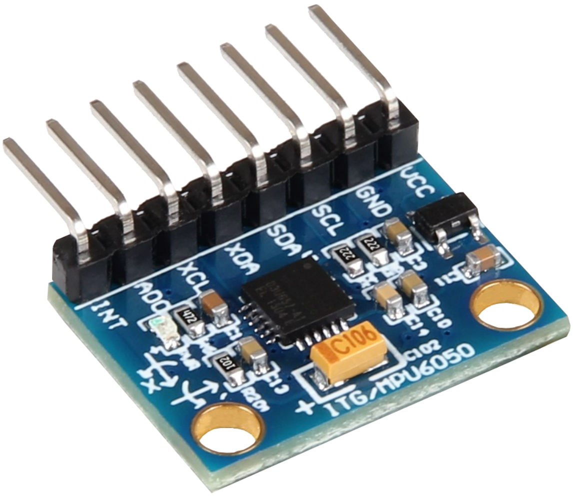
GYROSCOPE & ACCELEROMETER MODULE

SEN-MPU6050



1. GENERAL INFORMATION

Dear costumer,

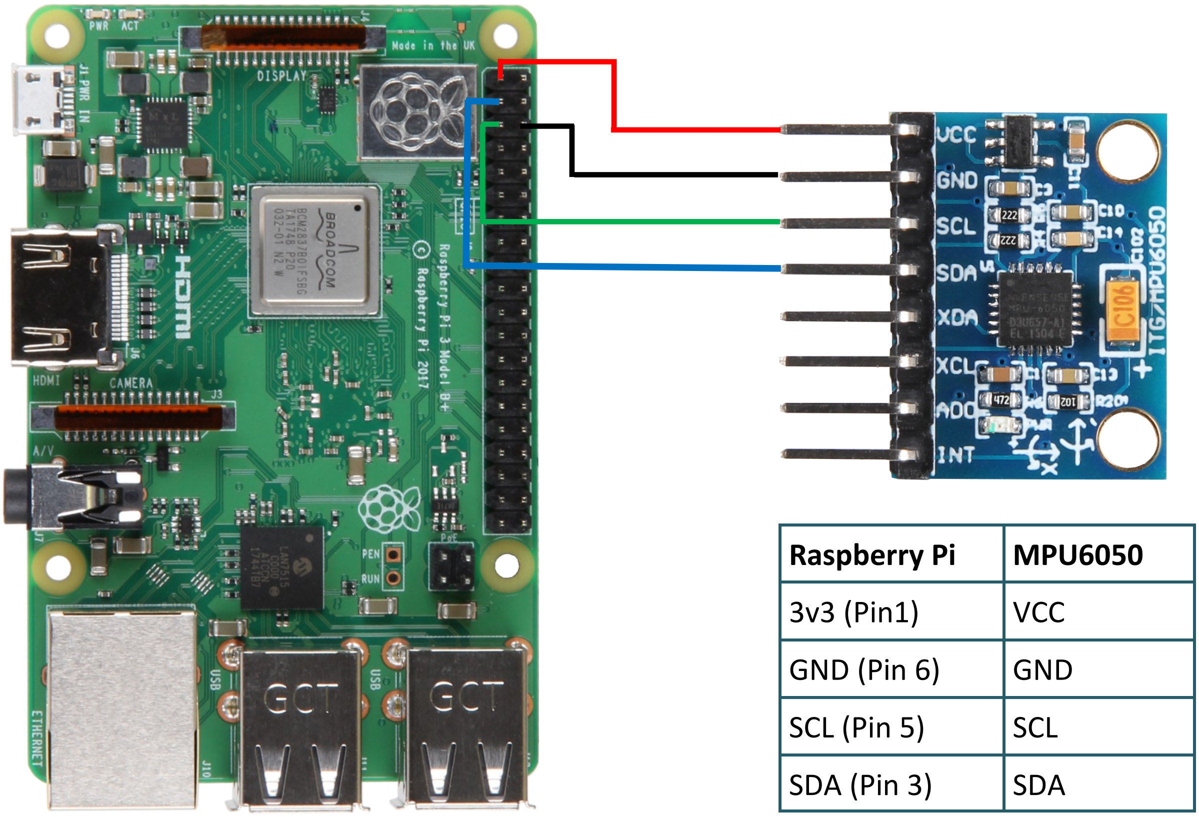
thank you very much for choosing our product.

In following, we will introduce you to what to observe while starting up and using this product.

Should you encounter any unexpected problems during use, please do not hesitate to contact us.

2. COMMISSIONING WITH A RASPBERRY PI

Wiring:



Raspberry Pi	MPU6050
3v3 (Pin1)	VCC
GND (Pin 6)	GND
SCL (Pin 5)	SCL
SDA (Pin 3)	SDA

After the sensor is connected, you have to activate I2C. Open the terminal and enter the following command:

```
sudo raspi-config
```

Now enable the I2C interface under **Interfacing Options** -> **I2C**.

You can now create the program file.

First, we create a new file in the document folder with following commands:

```
cd Documents
```

```
sudo nano MPU6050.py
```

Then copy the example code, shown on the following pages, completely into the recently created file.

```
#!/usr/bin/python
import smbus
import math
import time

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)
address = 0x68       # I2C-Address


bus.write_byte_data(address, power_mgmt_1, 0)

while True:
 gyroskop_xout = read_word_2c(0x43)
 gyroskop_yout = read_word_2c(0x45)
 gyroskop_zout = read_word_2c(0x47)
```

Continuation of the example code on the next page.

Continuation of the source code of the last page:

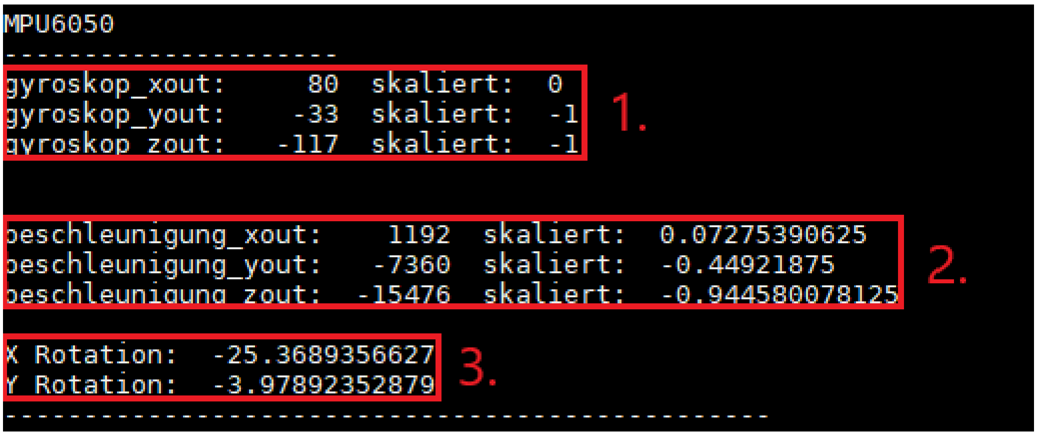
```
 print
 print ("MPU6050")
 print ("---------------------")

 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 ("")
 beschleunigung_xout = read_word_2c(0x3b)
 beschleunigung_yout = read_word_2c(0x3d)
 beschleunigung_zout = read_word_2c(0x3f)
 print ("")
 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 ("")
 print ("X Rotation: ") , get_x_rotation(beschleunigung_xout_skaliert, be-
schleunigung_yout_skaliert, beschleunigung_zout_skaliert)
 print ("Y Rotation: ") , get_y_rotation(beschleunigung_xout_skaliert, be-
schleunigung_yout_skaliert, beschleunigung_zout_skaliert)
 print ("------------------------------------------------")
 time.sleep(1) # Time between the Measurements in seconds
```

Now you can run the program with the following command :

```
sudo python MPU6050.py
```

2. COMMISSIONING WITH A RASPBERRY PI

Here is the explanation of the console output:



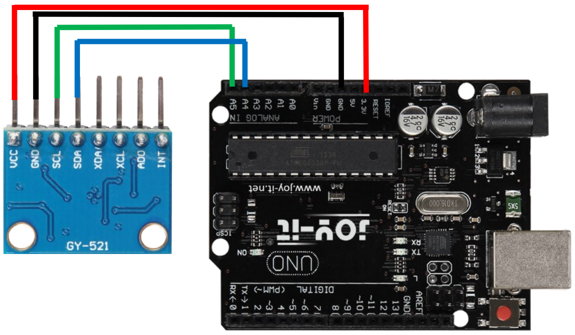
```
MPU6050
---------------------
gyroskop_xout:      80  skaliert:  0
gyroskop_yout:     -33  skaliert:  -1
gyroskop_zout:    -117  skaliert:  -1


beschleunigung_xout:     1192  skaliert:  0.07275390625
beschleunigung_yout:    -7360  skaliert:  -0.44921875
beschleunigung_zout:   -15476  skaliert:  -0.944580078125

X Rotation:  -25.3689356627
Y Rotation:  -3.97892352879
------------------------------------------------
```

1.: These 3 values display the values of the x, y and z axis of the gyroscope.

2.: These 3 values display the values of the x, y and z axis of the accelerometer.

3.: These values display the value the current tilt of the sensor in degrees.

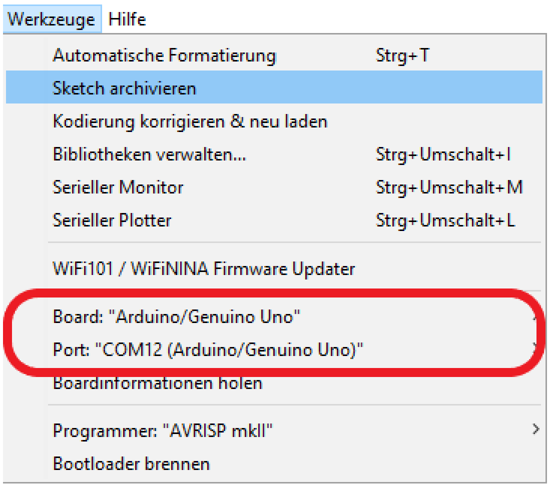
3. COMMISSIONING WITH AN ARDUINO



Arduino Uno	MPU6050
3v3	VCC
GND	GND
SCL (A5)	SCL
SDA (A4)	SDA

For usage of MPU6050 with an Arduino, you have to modulate your Arduino IDE to the used Board and the used port, first.

In this case, we use an Arduino UNO and the Port is COM12.


Werkzeuge
Hilfe
Automatische Formatierung Strg+T
Sketch archivieren
Kodierung korrigieren & neu laden
Bibliotheken verwalten... Strg+Umschalt+I
Serieller Monitor Strg+Umschalt+M
Serieller Plotter Strg+Umschalt+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM12 (Arduino/Genuino Uno)"
Boardinformationen holen
Programmer: "AVRISP mkII"
Bootloader brennen

Now enter the following source code into your Arduino IDE and press "Upload" .

```
#include "Wire.h" // Importing the I2C library.
const int I2C_adress_MPU = 0x68; // I2C Address of the MPU6050.
int16_t Beschleunigung_x, Beschleunigung_y, Beschleunigung_z; // Variables for the Accelerometer sensor
int16_t gyro_x, gyro_y, gyro_z; // Variables for the Gyroscope
int16_t Temperatur; // Variable in which the temperature is saved
char tmp_str[7];
char* convert_int16_to_str(int16_t i) {
  sprintf(tmp_str, "%6d", i);
  return tmp_str;
}
void setup() {
  Serial.begin(9600);
  Wire.begin();
  Wire.beginTransmission(I2C_adress_MPU); // Starting the I2C transmission
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
}
```

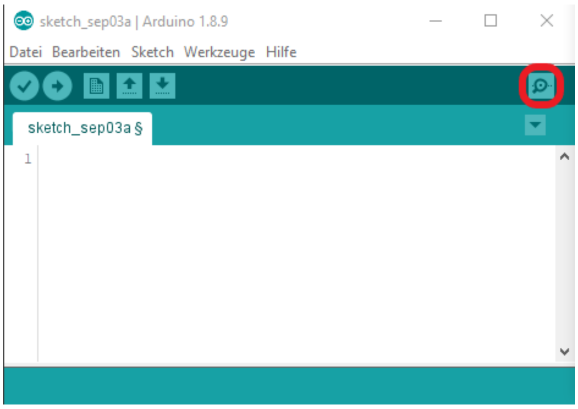
Continuation of the source code on the following page.

3. COMMISSIONING WITH AN ARDUINO

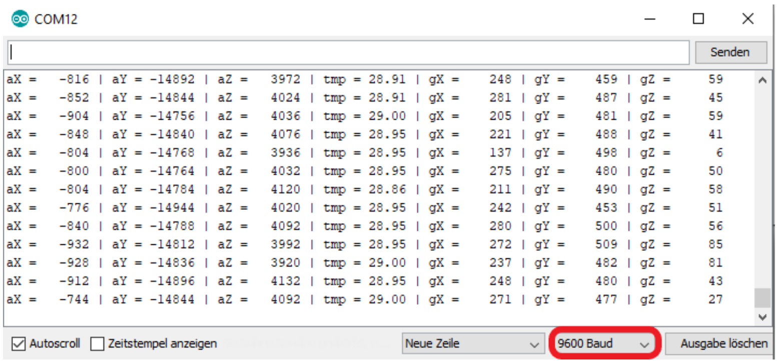
Continuation of the source code:

```
void loop() {
  Wire.beginTransmission(I2C_adress_MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(I2C_adress_MPU, 7*2, true);
  Beschleunigung_x = Wire.read()<<8 | Wire.read();
  Beschleunigung_y = Wire.read()<<8 | Wire.read();
  Beschleunigung_z = Wire.read()<<8 | Wire.read();
  Temperatur = Wire.read()<<8 | Wire.read();
  gyro_x = Wire.read()<<8 | Wire.read();
  gyro_y = Wire.read()<<8 | Wire.read();
  gyro_z = Wire.read()<<8 | Wire.read();
  // Output of Data
  Serial.print("aX = "); Serial.print(convert_int16_to_str(Beschleunigung_x));
  Serial.print(" | aY = "); Serial.print(convert_int16_to_str(Beschleunigung_y));
  Serial.print(" | aZ = "); Serial.print(convert_int16_to_str(Beschleunigung_z));
  Serial.print(" | tmp = "); Serial.print(Temperatur/340.00+36.53);
  Serial.print(" | gX = "); Serial.print(convert_int16_to_str(gyro_x));
  Serial.print(" | gY = "); Serial.print(convert_int16_to_str(gyro_y));
  Serial.print(" | gZ = "); Serial.print(convert_int16_to_str(gyro_z));
  Serial.println();
  // 1 Second Pause
  delay(1000);
}
```

Press the red-marked button, to call up the serial monitor.


sketch_sep03a | Arduino 1.8.9
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_sep03a §

Make sure that the baud rate (red marked field in the picture) is set to 9600.

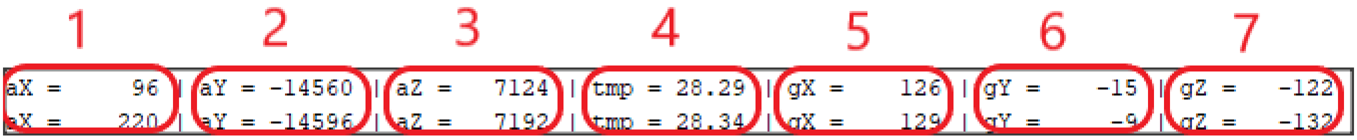


```
aX =   -816 | aY = -14892 | aZ =   3972 | tmp = 28.91 | gX =    248 | gY =    459 | gZ =     59
aX =   -852 | aY = -14844 | aZ =   4024 | tmp = 28.91 | gX =    281 | gY =    487 | gZ =     45
aX =   -904 | aY = -14756 | aZ =   4036 | tmp = 29.00 | gX =    205 | gY =    481 | gZ =     59
aX =   -848 | aY = -14840 | aZ =   4076 | tmp = 28.95 | gX =    221 | gY =    488 | gZ =     41
aX =   -804 | aY = -14768 | aZ =   3936 | tmp = 28.95 | gX =    137 | gY =    498 | gZ =      6
aX =   -800 | aY = -14764 | aZ =   4032 | tmp = 28.95 | gX =    275 | gY =    480 | gZ =     50
aX =   -804 | aY = -14784 | aZ =   4120 | tmp = 28.86 | gX =    211 | gY =    490 | gZ =     58
aX =   -776 | aY = -14944 | aZ =   4020 | tmp = 28.95 | gX =    242 | gY =    453 | gZ =     51
aX =   -840 | aY = -14788 | aZ =   4092 | tmp = 28.95 | gX =    280 | gY =    500 | gZ =     56
aX =   -932 | aY = -14812 | aZ =   3992 | tmp = 28.95 | gX =    272 | gY =    509 | gZ =     85
aX =   -928 | aY = -14836 | aZ =   3920 | tmp = 29.00 | gX =    237 | gY =    482 | gZ =     81
aX =   -912 | aY = -14896 | aZ =   4132 | tmp = 28.95 | gX =    248 | gY =    480 | gZ =     43
aX =   -744 | aY = -14844 | aZ =   4092 | tmp = 29.00 | gX =    271 | gY =    477 | gZ =     27
```

Now you can read the values measured by the sensor.

3. COMMISSIONING WITH AN ARDUINO

Explanation of console output:



```
aX =       96 | aY = -14560 | aZ =    7124 | tmp = 28.29 | gX =      126 | gY =      -15 | gZ =    -122
aX =      220 | aY = -14596 | aZ =    7192 | tmp = 28.34 | gX =      129 | gY =       -9 | gZ =    -132
```

1.: This value is the value of the x-axis of the accelerometer.

2.: This value is the value of the y-axis of the accelerometer. .

3.: This value is the value of the z-axis of the accelerometer. .

4.: This value states the current temperature.

5.: This value is the value of the x-axis of the Gyroscope.

6.: This value is the value of the y-axis of the Gyroscope.

7.: This value is the value of the x-axis of the Gyroscope.

4. ADDITIONAL INFORMATION

Our information and take-back obligations according to the Electrical and Electronic Equipment Act (ElektroG)

Symbol on electrical and electronic equipment:

This crossed-out dustbin means that electrical and electronic appliances do not belong in the household waste. You must return the old appliances to a collection point.
Before handing over waste batteries and accumulators that are not enclosed by waste equipment must be separated from it.

Return options:
As an end user, you can return your old device (which essentially fulfils the same function as the new device purchased from us) free of charge for disposal when you purchase a new device.
Small appliances with no external dimensions greater than 25 cm can be disposed of in normal household quantities independently of the purchase of a new appliance.

Possibility of return at our company location during opening hours:
Simac GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn, Germany

Possibility of return in your area:
We will send you a parcel stamp with which you can return the device to us free of charge. Please contact us by e-mail at Service@joy-it.net or by telephone.

Information on packaging:
If you do not have suitable packaging material or do not wish to use your own, please contact us and we will send you suitable packaging.

5. SUPPORT

If there are still any issues pending or problems arising after your purchase, we will support you by e-mail, telephone and with our ticket support system.

E-Mail: service@joy-it.net

Ticket system: http://support.joy-it.net

Telephone: +49 (0)2845 98469-66 (10-17 o‘clock)

For further information please visit our website:

www.joy-it.net

Veröffentlicht: 29.10.2019