

# MPU650 test tool for raspberry Pi

## Background information

<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>

<https://www.raspberry-pi-geek.de/ausgaben/rpg/2017/08/3-achsen-lage-und-beschleunigungssensor-mpu6050/>

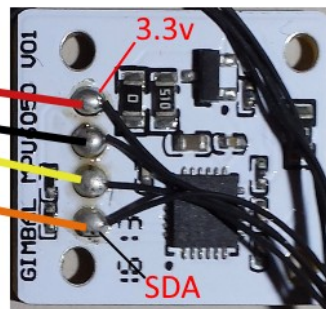
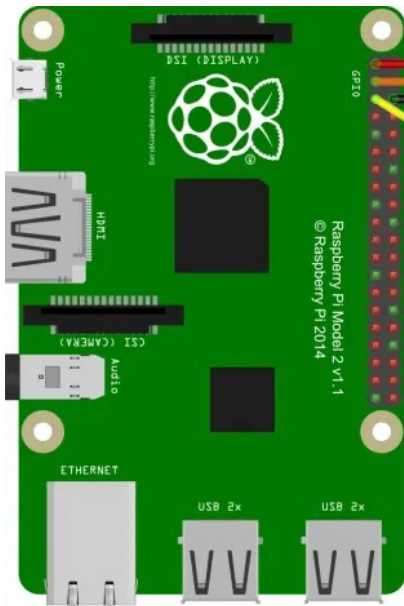
<https://github.com/Blokkendoos/mpu-calibration>

## Preparations

Enable I2C:

`sudo raspi-config` > Interface Options > I2C > Yes

Connect MPU and check wiring.



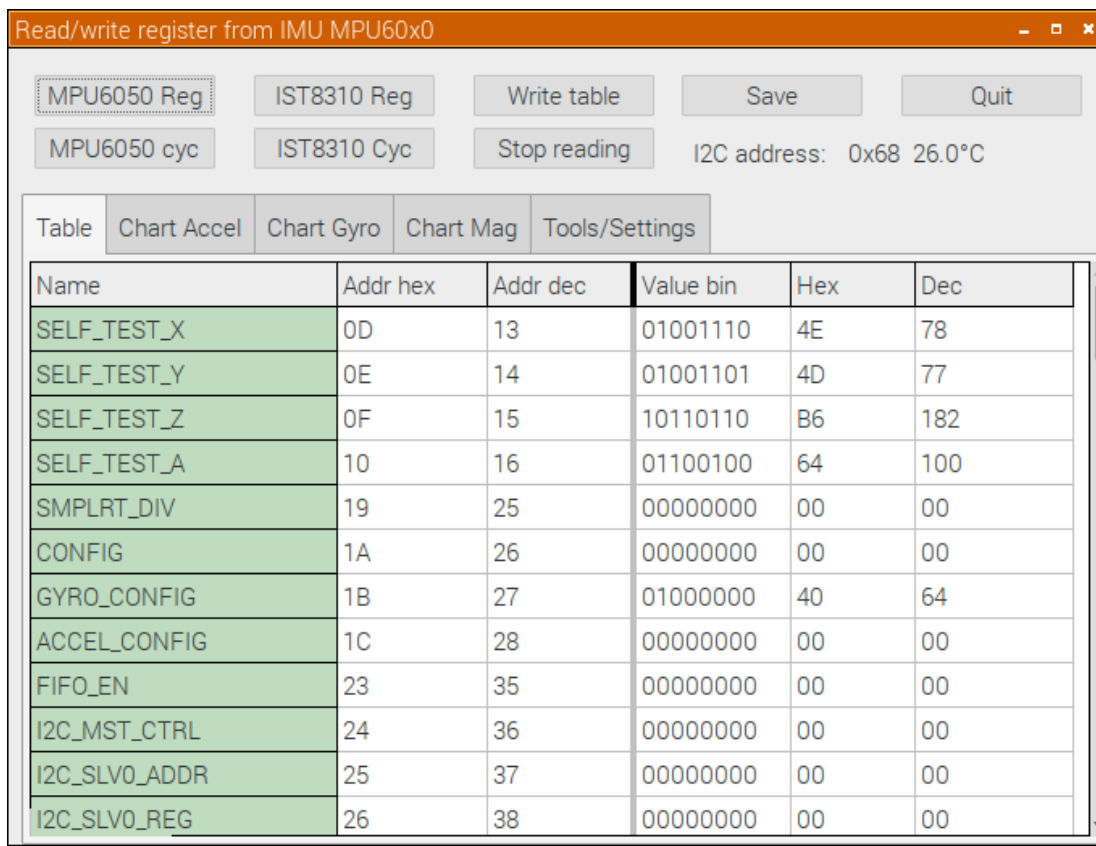
Raspberry Pi	MPU6050	
pin 1 - +3.3V	Vcc	red
pin 6 - GND	GND	black
pin 5 - SCL	SCL	yellow
pin 3 - SDA	SDA	orange

Check if MPU6050 is available: `i2cdetect -y 1` > should be appear at address 0x68

```
pi@raspigui: ~  
Datei Bearbeiten Reiter Hilfe  
pi@raspigui:~ $ 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: -- -- -- -- -- -- -- -- -- -- -- -- -- --  
pi@raspigui:~ $
```

## IMU\_test

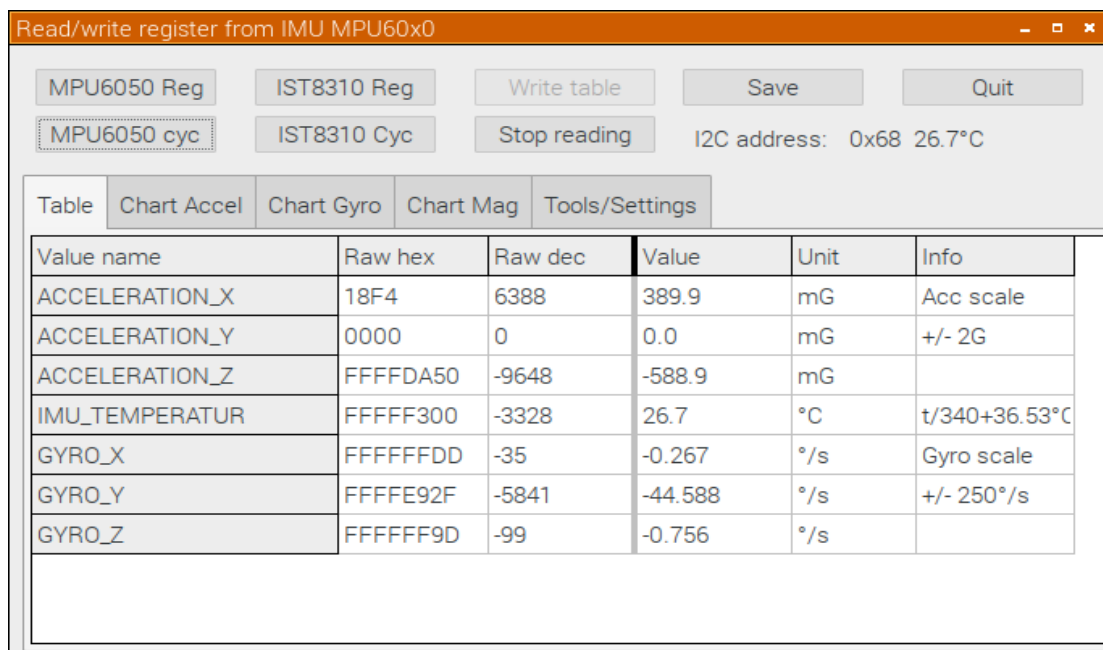
This is a test tool to check and learn something about the Motion Processing Unit MPU6050. One can read all register and save the settings to a CSV file just as test.



The screenshot shows the 'Read/write register from IMU MPU60x0' application window. The 'MPU6050 Reg' button is selected. The 'I2C address' is 0x68 and the temperature is 26.0°C. The 'Table' tab is active, displaying a list of registers with their addresses and values in binary, hex, and decimal.

Name	Addr hex	Addr dec	Value bin	Hex	Dec
SELF_TEST_X	0D	13	01001110	4E	78
SELF_TEST_Y	0E	14	01001101	4D	77
SELF_TEST_Z	0F	15	10110110	B6	182
SELF_TEST_A	10	16	01100100	64	100
SMPLRT_DIV	19	25	00000000	00	00
CONFIG	1A	26	00000000	00	00
GYRO_CONFIG	1B	27	01000000	40	64
ACCEL_CONFIG	1C	28	00000000	00	00
FIFO_EN	23	35	00000000	00	00
I2C_MST_CTRL	24	36	00000000	00	00
I2C_SLV0_ADDR	25	37	00000000	00	00
I2C_SLV0_REG	26	38	00000000	00	00

It's also possible to cyclic read the Accelerometer, Temperature and Gyroscope values.

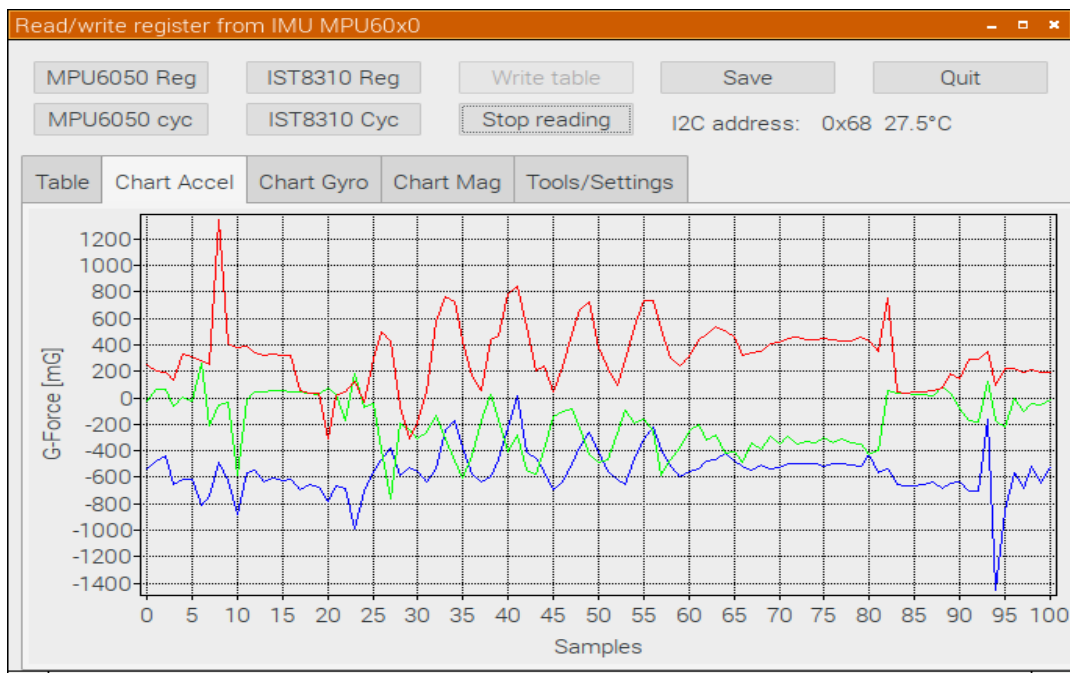


The screenshot shows the same application window, but the 'MPU6050 cyc' button is selected. The 'I2C address' is 0x68 and the temperature is 26.7°C. The 'Table' tab is active, displaying a list of sensor values with their raw data, converted values, units, and additional information.

Value name	Raw hex	Raw dec	Value	Unit	Info
ACCELERATION_X	18F4	6388	389.9	mG	Acc scale
ACCELERATION_Y	0000	0	0.0	mG	+/- 2G
ACCELERATION_Z	FFFFDA50	-9648	-588.9	mG	
IMU_TEMPERATUR	FFFFFF300	-3328	26.7	°C	t/340+36.53°C
GYRO_X	FFFFFFD0	-35	-0.267	°/s	Gyro scale
GYRO_Y	FFFFE92F	-5841	-44.588	°/s	+/- 250°/s
GYRO_Z	FFFFFF9D	-99	-0.756	°/s	

You will also see the current scale setting for Accelerometer and Gyroscope in Info column.

The same can be seen in a rolling chart.



For testing and settings it is possible to write into a register or overwrite all write-able with zero.

The screenshot shows the "Tools/Settings" tab of the "Read/write register from IMU MPU60x0" software. The top bar shows "I2C address: 0x0E 26.3°C". The "Tools/Settings" tab is active, displaying several configuration options:

- Overwrite register:** Includes fields for "Address decimal" (RegNo), "Value decimal 0..255" (0.255), and "Binary" (0x00). Buttons for "Write byte" and "Write all zero" are present.
- IST8310 options:** Includes checkboxes for "Single measurement mode" (checked), "Data ready enable", and "Self test".
- MPU6050 options:** Includes buttons for "Self test" (Pending) and "Add slave".
- Sample timer [ms]:** Includes radio buttons for "MC6050" and "IST8310" with options 100, 150, 200 (selected), 250, 500, and 1000.
- Byte calculator:** Includes input fields for "Decimal" (dec), "Hexadecimal" (hex), and "Binary" (bin).

Some options, settings and special actions are on the Tools/Settings page too.

**Some terminal commands – good to know:**

Read a byte from MPU: `i2cget -y 1 0x68 0x75` (Who am I, it's own address)  
Read a word from MPU: `i2cget -y 1 0x68 65 w` (result comes as big endian)  
Write a byte to MPU register: `i2cset y 1 0x68 107 0` (wake-up command)  
Read temperature cyclic (raw): `watch -n 0.5 'i2cget -y 1 0x68 65 w'`  
Read ADC PCF8591: `i2cget -y 1 0x48 0x42` (channel AIN2)