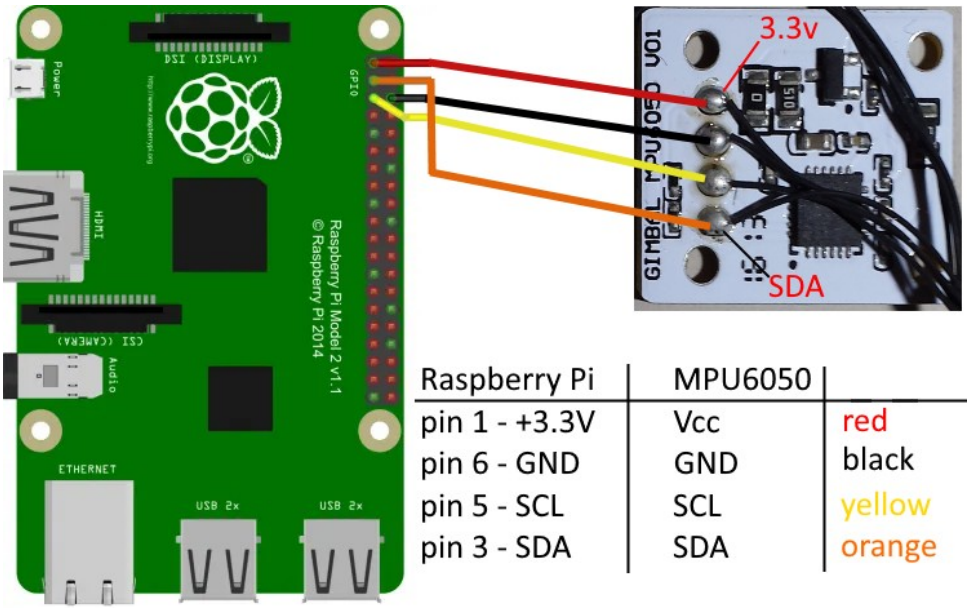
MPU6050 test tool for raspberry Pi

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://github.com/Blokkendoos/mpu-calibration

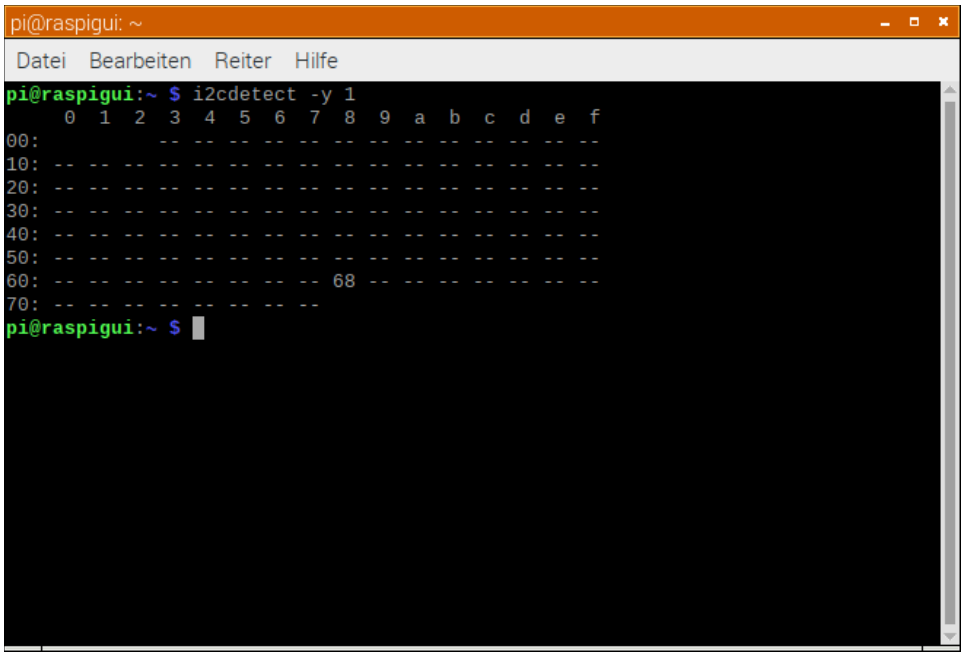
Preparations

Enable I2C: **`sudo raspi-config`** > Interface Options > I2C > Yes
Install i2cdetect: **`sudo apt-get install i2c-tools`**
Connect MPU and check wiring.



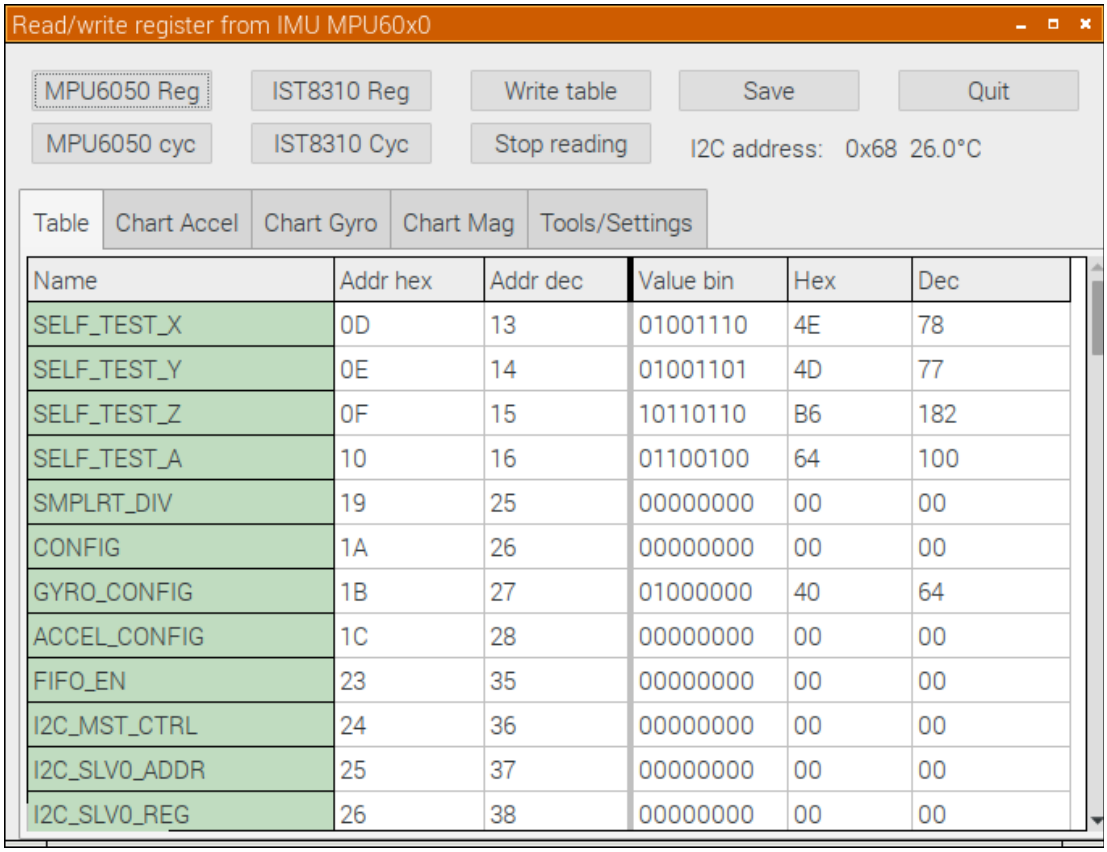
Raspberry Pi	MPU6050	
pin 1 - +3.3V	Vcc	red
pin 6 - GND	GND	black
pin 5 - SCL	SCL	yellow
pin 3 - SDA	SDA	orange

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



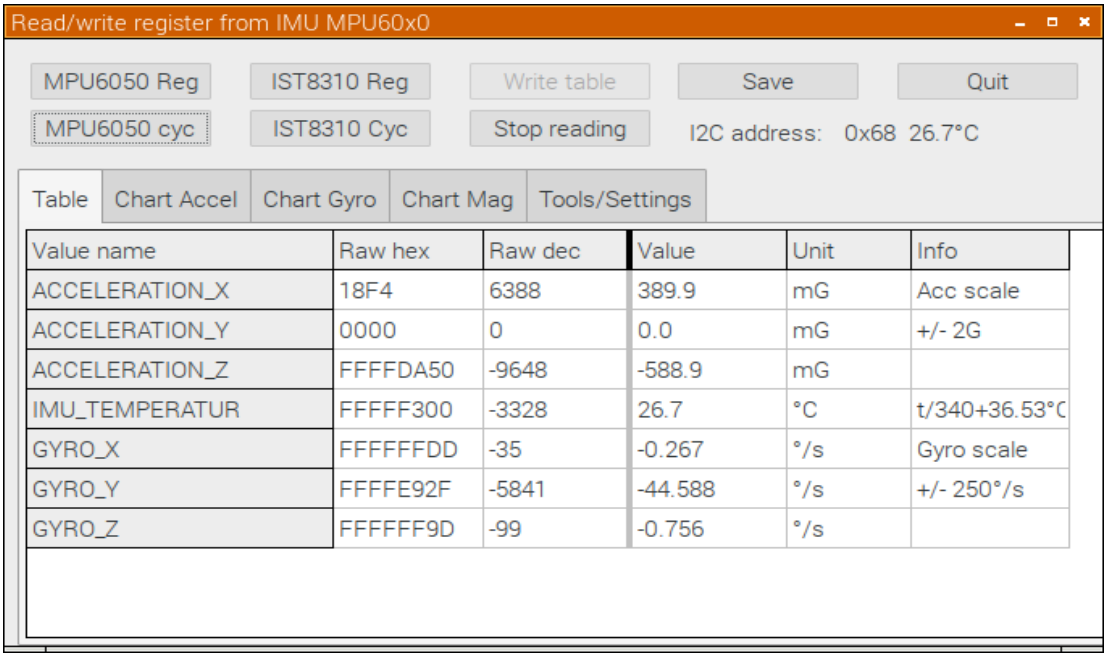
IMU_test

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.



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

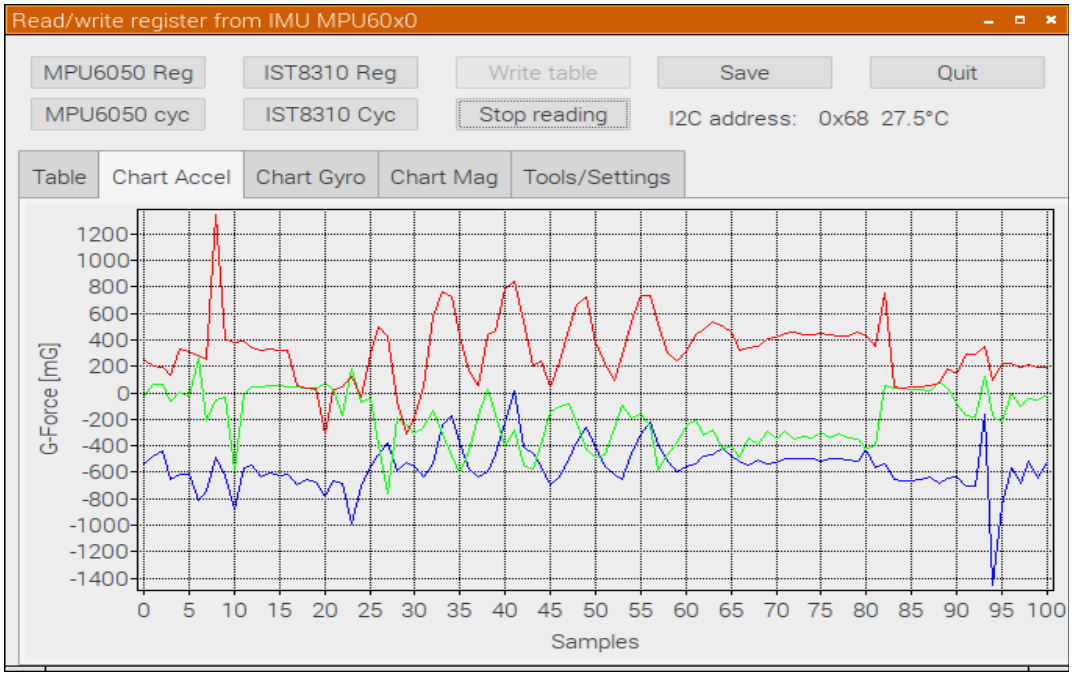
It's also possible to cyclic read the Accelerometer, Temperature and Gyroscope values.



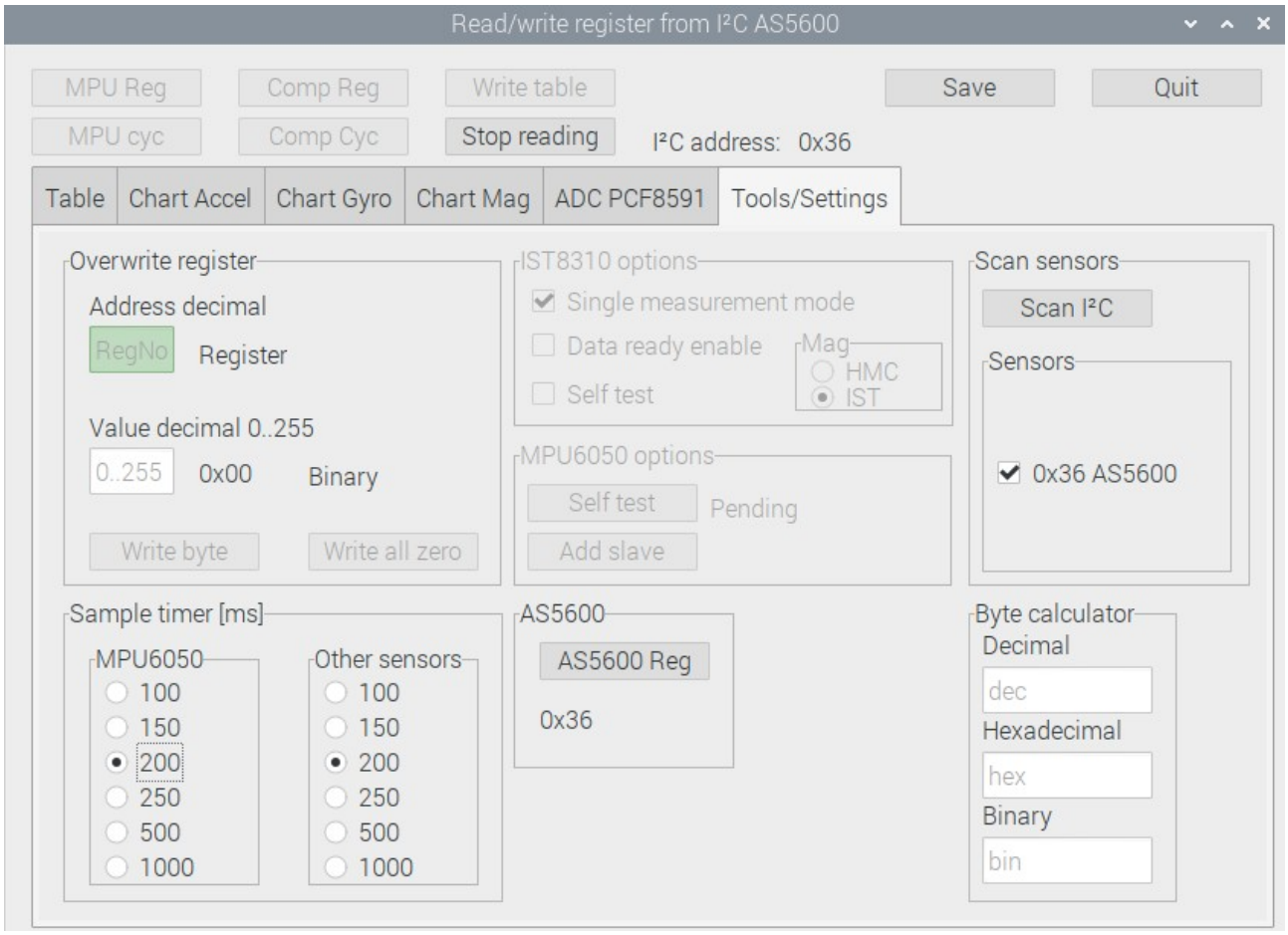
Value name	Raw hex	Raw dec	Value	Unit	Info
ACCELERATION_X	18F4	6388	389.9	mG	Acc scale
ACCELERATION_Y	0000	0	0.0	mG	+/- 2G
ACCELERATION_Z	FFFFDA50	-9648	-588.9	mG	
IMU_TEMPERATUR	FFFFF300	-3328	26.7	°C	t/340+36.53°C
GYRO_X	FFFFFFDD	-35	-0.267	°/s	Gyro scale
GYRO_Y	FFFFE92F	-5841	-44.588	°/s	+/- 250°/s
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.



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'`**

Programming contactless potentiometer AS5600

AS5600 Register

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Configuration Registers (1), (2)										
0x00	ZMCO	R							ZMCO(1:0)	
0x01	ZPOS	R/W/P					ZPOS(11:8)			
0x02			ZPOS(7:0)							
0x03	MPOS	R/W/P					MPOS(11:8)			
0x04			MPOS(7:0)							
0x05	MANG	R/W/P					MANG(11:8)			
0x06			MANG(7:0)							
0x07	CONF	R/W/P			WD	FTH(2:0)			SF(1:0)	
0x08			PWMF(1:0)		OUTS(1:0)		HYST(1:0)		PM(1:0)	
Output Registers										
0x0C	RAW ANGLE	R					RAW ANGLE(11:8)			
0x0D		R	RAW ANGLE(7:0)							
0x0E	ANGLE	R					ANGLE(11:8)			
0x0F		R	ANGLE(7:0)							
Status Registers										
0x0B	STATUS	R			MD	ML	MH			
0x1A	AGC	R	AGC(7:0)							
0x1B	MAGNITUDE	R					MAGNITUDE (11:8)			
0x1C		R	MAGNITUDE(7:0)							
Burn Commands										
0xFF	BURN	W	Burn_Angle = 0x80; Burn_Setting = 0x40							

Note(s):

1. To change a configuration, read out the register, modify only the desired bits and write the new configuration. Blank fields may contain factory settings.
2. During power-up, configuration registers are reset to the permanently programmed value. Not programmed bits are zero.

Read Register with button "AS5600 Reg". Same as button Read CONF but jumps to Table to see all register values for verification.
To read values from data registers click on "AS5600 Cyc" (Read cyclic).

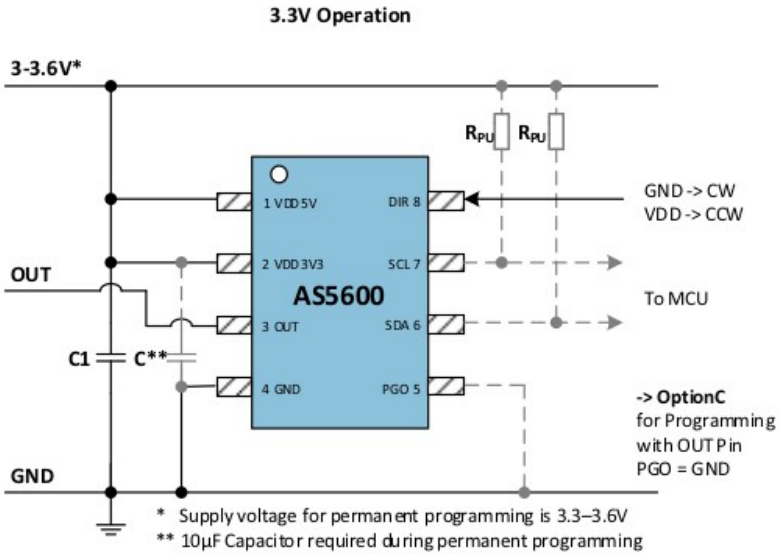
Note: To change Conf register do not change n/a bits. Those may contain factory settings.

Programming (Option A with I²C)

Use the correct hardware configuration (Option A and C).

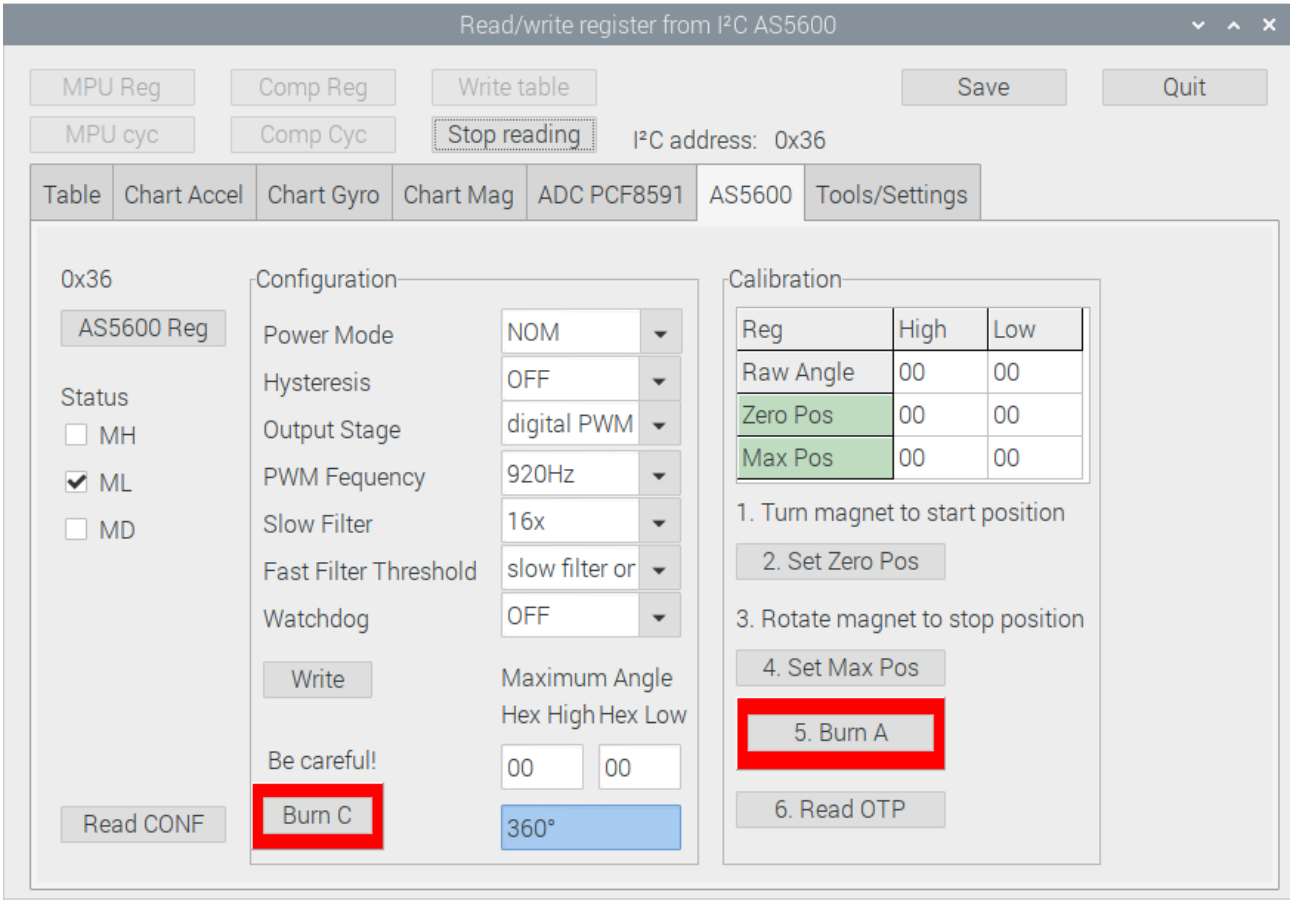
Option A: Write CONF register
Option C: Set Maximum Angle

Burn: Permanently save both.



Usage

Connect AS5600 to Raspberry Pi with I²C interface and power up the AS5600.
Start the tool "IMU_test". Go to Page "AS5600". It will go to "AS5600" page if this is the only detected sensor on I²C bus.



Read the register of AS5600.

- Button "AS5600 Reg" or button "Read CONF"

Calibration procedure

Step 1: Turn the magnet to the start position.
Step 2: Read the RAW ANGLE register. Write the RAW ANGLE value into the ZPOS register. Wait at least 1 ms.

- Button "2. Set Zero Pos"

Step 3: Rotate the magnet in the direction defined by the level on the DIR pin (GND for clockwise, VDD for counterclockwise) to the stop position. The amount of rotation must be greater than 18 degrees.
Step 4: Read the RAW ANGLE register. Write the RAW ANGLE value into the MPOS register. Wait at least 1 ms. Proceed with Step 6 to permanently program the configuration.

- Button "4. Set Max Pos"

Step 5: Perform a BURN_ANGLE command to permanently program the device. Wait at least 1 ms.

- Button 5. Burn A

Step 6: Verify the BURN_ANGLE command: Write the commands 0x01, 0x11 and 0x10 sequentially into the register 0xFF to load the actual OTP content. Read the ZPOS and MPOS registers to verify that the BURN_ANGLE command was successful.

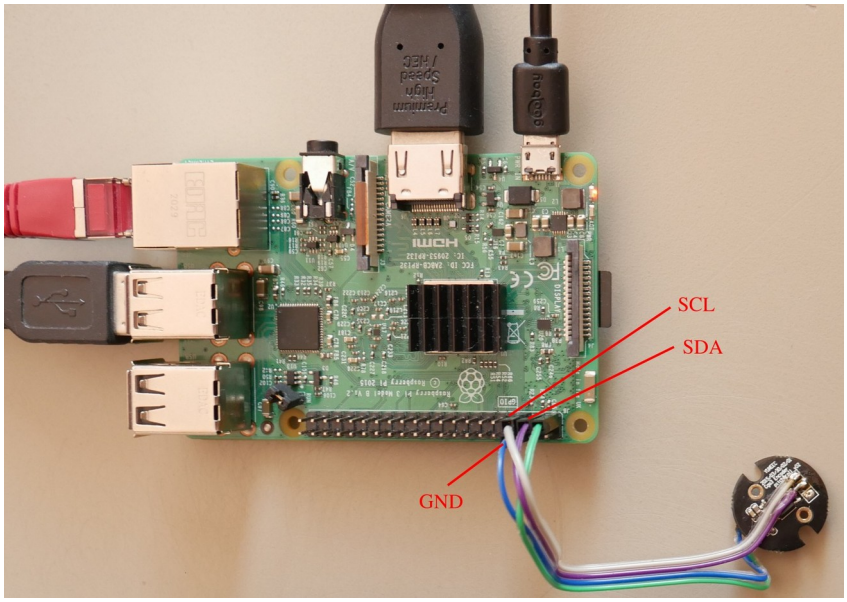
- Button "6. Read OTP"

Step 7: Read and verify the ZPOS and MPOS registers again after a new power-up cycle.

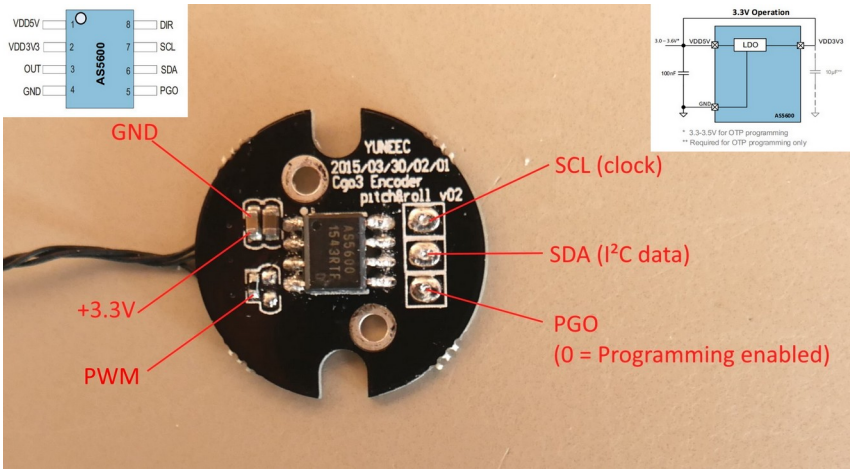
- Button "AS5600 Reg"

Note: Do not touch Configuration, at least do not burn config.
Basically, do only something when you exactly know what you are doing, especially "Burn" actions.
Before perform a burn-command read carefully the **rules and restictions** for the OTP mentioned in AS5600 datasheet.

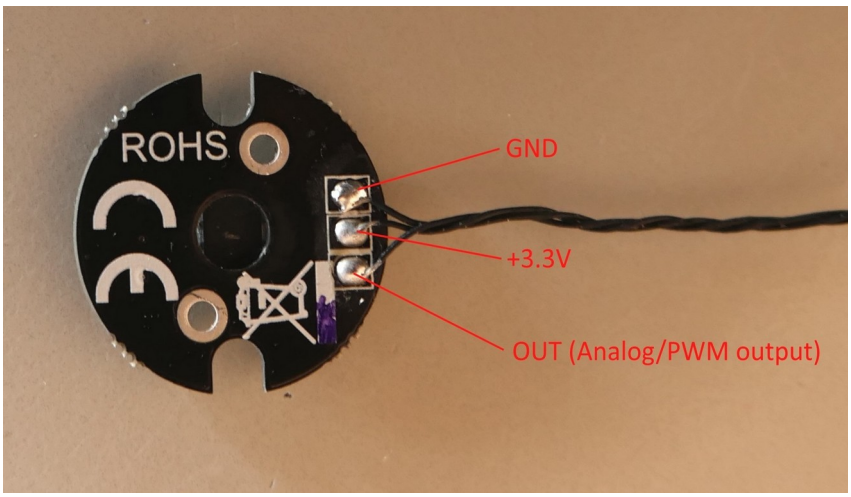
HW settings



Sensor connected to Raspberry Pi by I²C interface.



I²C bus SCL and SDA is 3.3V signal.
PGO pin is not needed when programming via I²C.



Power supply is 3.3V. It canbe done by camera or Raspberry PI.
GND must be connected to Raspberry Pi.