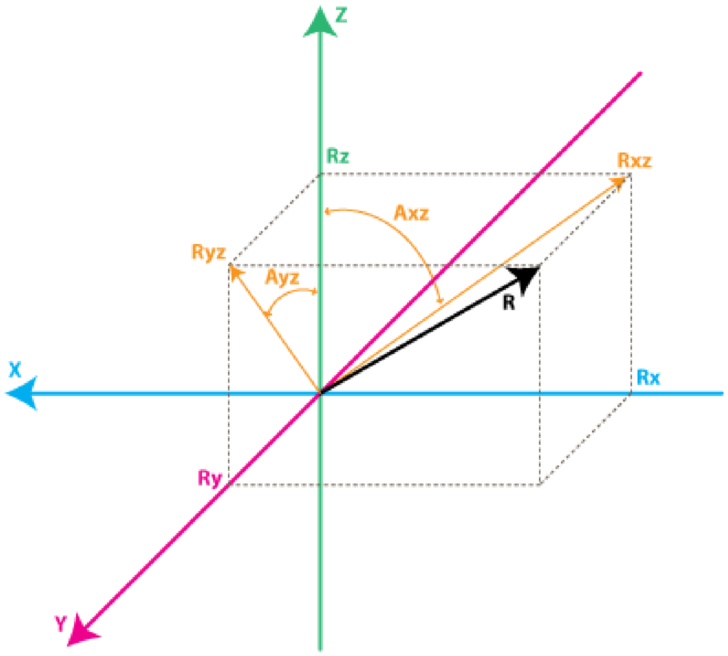
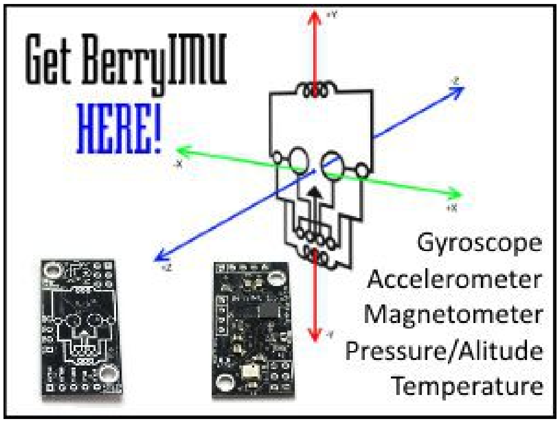


Guide to interfacing a Gyro and Accelerometer with a Raspberry Pi

April 29, 2013 | Mark Williams | 65 Comments

A lot of people have asked how I got the readings from the Gyro, Accelerometer, and Compass inertial measurement unit(IMU) which is used to keep PiBBOT upright.

UPDATED And updated guide has been published here, with cleaner and simpler code.



In this guide I will explain how to get readings from the IMU and convert these raw readings into usable angles. I will also show how to read some of the information in the datasheets for these devices. The theory and principals below can be applied to any digital IMU, just some minor modifications need to be made. Eg register values, sensitivity level…

The code can be found here;
https://github.com/mwilliams03/Raspberry-Gyro-Acc

A note about Gyros and Accelerometers

When using the IMU to calculate angles, readings from both the gyro and accelerometer are needed, which are then combined. This is because using either on their own will result in inaccurate readings.

Here is why;
Gyros - A gyro measures the rate of rotation, which has to be tracked over time to calculate the current angle. This tracking causes the gyro to drift. However, gyros are good at measuring quick sharp movements.

Accelerometers - Accelerometers are used to sense both static (e.g. gravity) and dynamic (e.g. sudden starts/stops) acceleration. They don't need to be tracked like a gyro and can measure the current angle at any given time. Accelerometers however are very noisy and are only useful for tracking angles over a long period of time.

Here is an excellent tutorial about accelerometers and gyros;
http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/?ALLSTEPS

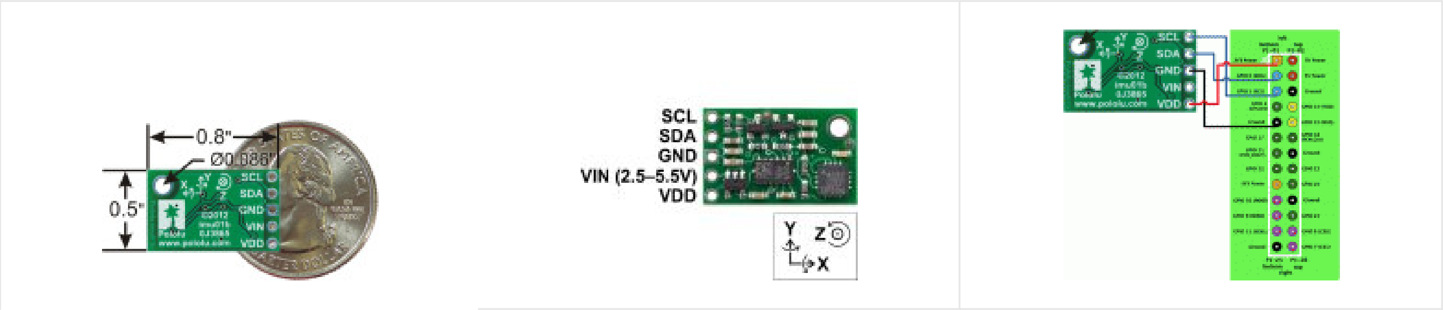
Setting up the IMU and I2C

The IMU I use is a MinIMU-9 v2 Gyro, Accelerometer, and Compass which uses a L3GD20 3-axis gyroscope and a

LSM303DLHC 3-axis accelerometer and 3-axis magnetometer.
The datasheets are needed if you want to use these devices;
L3GD20 datasheet
LSM303DLHC datasheet

This IMU communicates via the I2C interface.

From the images below, you can see how to connect it to the Raspberry Pi. You can also see the orientation of the X, Y and Z axis.



Once connected, you need to install the I2C-tools

```
pi@raspberrypi ~ $ sudo apt-get install i2c-tools libi2c-dev
```

You will then need to comment out the driver from the blacklist. currently the I2C driver isn't being loaded.

```
pi@raspberrypi ~ $ sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Place a hash '#' in front of **blacklist i2c-bcm2708**

You now need to edit the modules conf file.

```
pi@raspberrypi ~ $ sudo nano /etc/modules
```

Add these two lines;

```
i2c-dev
i2c-bcm2708
```

Now reboot.

Once your Raspberry Pi reboots, you should see the addresses of the components on the IMU when using i2cdetect;

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
```

Or port 0 on the older Raspberry Pi

```
pi@raspberrypi ~ $ sudo i2cdetect -y 0
```

```
pi@raspberrypi ~ $ sudo /usr/sbin/i2cdetect -y 1
0 1 2 3 4 5 6 7 8 9 a b c d e f
00: — — — — — — — — — — — — —
10: — — — — — — — — — 19 — — — — 1e —
20: — — — — — — — — — — — — — — — —
30: — — — — — — — — — — — — — — — —
40: — — — — — — — — — — — — — — — —
50: — — — — — — — — — — — — — — — —
60: — — — — — — — — — — — 6b — — — —
70: — — — — — — — —
pi@raspberrypi ~ $
```

The gyro (L3GD2) is using the address of 0x6b, this can also be verified with page 23 on the datasheet.

Quoted from the datasheet with the address highlighted;
"The Slave ADdress (SAD) associated with the L3GD20 is 110101xb. The SDO pin can be used to modify the less significant bit of the device address. If the SDO pin is connected to voltage supply, LSb is '1' **(address 1101011b)**. Otherwise, if the SDO pin is connected to ground, the LSb value is '0' (address 1101010b). This solution allows to connect and address two different gyroscopes to the same I2C bus"

The accelerometer (LSM303DLHC) is using address 0x19, this can be seen on page 20 of the datasheet;
"For linear acceleration the default (factory) 7-bit slave address is 0011001b."

The third address shown above, 0x1e, is used for the magnetometer on the LSM303DLHC, from the datasheet;
"For magnetic sensors the default (factory) 7-bit slave address is 0011110xb."

I will not be covering the magnetometer in this post.

Reading and Writing to the Accelerometer and Gyro

We read and write values from different register in the above addresses to do different things. Eg Turn the Gyro on, set the sensitivity level, read accelerometer values. etc..
The git repository includes i2c-dev.h, which contains the functions to write and read to the I2C bus.
The git repository also includes the files for the L3GD20 and LSM303DLHC which lists all the registers used by these sensors.

We use ioctl calls to read and write information to the I2C bus.

Open I2C Bus

First, you need to open the I2C bus device file, as follows;

```
char filename[20];
sprintf(filename, "/dev/i2c-%d", 1);
file = open(filename, O_RDWR);
if (file<0) {
        printf("Unable to open I2C bus!");
        exit(1);
}
```

Enable the Accelerometer

```
// Enable accelerometer.
writeAccReg(LSM303_CTRL_REG1_A, 0b01010111); //  z,y,x axis enabled , 100Hz data ra
writeAccReg(LSM303_CTRL_REG4_A, 0b00101000); // +/- 8G full scale: FS = 10 on DLHC,
```

To enable to accelerometer, we need to write a byte to the CTRL_REG1_A(0x20) register on the LSM303. This can be see on page 24 of the datasheet, where you will also find other values that can be used.

After this, we then need to write a byte to the CTRL_REG4_A(0x23) register to set our sensitivity. This information is on page 26 of the datasheet.

Enable the Gyro

Enabling the gyro is very similar to the accelerometer.

```
// Enable Gyro
writeGyrReg(L3G_CTRL_REG1, 0b00001111); // Normal power mode, all axes enabled
writeGyrReg(L3G_CTRL_REG4, 0b00110000); // Continuous update, 2000 dps full scale
```

Reading the Values from the Accelerometer and Gyro

Once you enable the accelerometer and gyro, you can start reading the raw values from these sensors.
Reading the accelerometer

```
void readACC(int  *a)
{
        uint8_t block[6];
        selectDevice(file,ACC_ADDRESS);

        readBlock(0x80 | LSM303_OUT_X_L_A, sizeof(block), block);

        *a = (int16_t)(block[0] | block[1] << 8) >> 4;
        *(a+1) = (int16_t)(block[2] | block[3] << 8) >> 4;
        *(a+2) = (int16_t)(block[4] | block[5] << 8) >> 4;
}
```

The above function will read the raw values from the accelerometer, here is a description of what is happening;

1. An array of 6 bytes is first created to store the values.
2. The I2C slave address is selected for the accelerometer, by passing the accelerometer address of

ACC_ADDRESS or 0x19 (0x32 >> 1) to the selectDevice() function.

3. Using the readBlock() function from i2c-dev.h, we read 6 bytes starting at LSM303_OUT_X_L_A (0x28). This is shown on page 28 of the datasheet.
4. The values are expressed in 2's complement (MSB for the sign and then 15 bits for the value) so we need to combine;
   block[0] & block[1] for X axis
   block[2] & block[3] for Y axis
   block[4] & block[5] for Z axis

Reading the gyro

```
void readGYR(int *g)
{
        uint8_t block[6];
        selectDevice(file,GYR_ADDRESS);

        readBlock(0x80 | L3G_OUT_X_L, sizeof(block), block);

        *g = (int16_t)(block[1] << 8 | block[0]);
        *(g+1) = (int16_t)(block[3] << 8 | block[2]);
        *(g+2) = (int16_t)(block[5] << 8 | block[4]); }
```

The above function reads the values from the gyro. It is very similar to the readACC() function.
Once we have the raw values, we can start converting them into meaningful angle values.

Convert the Raw Values to Usable Angles

Gyro

```
//Convert Gyro raw to degrees per second
rate_gyr_x = (float) *gyr_raw * G_GAIN;
rate_gyr_y = (float) *(gyr_raw+1) * G_GAIN;
rate_gyr_z = (float) *(gyr_raw+2) * G_GAIN;
```

For the gyro, we need to work out how fast it is rotating in degrees per second(dps). This is easily done by multiplying the raw value by the sensitivity level that was used when enabling the gyro. G_GAIN = 0.07, which is based off a sensitivity level of 2000 dps. Looking at the table on page 9 of the datasheet, we can see that we need to multiple it by 0.07 to get degrees per second. The table shows it as 70 mdps (milli degrees per second). Another example: Looking at the table, if we chose a sensitivity level of 250dps when enabling the gyro, then we would have to multiply the raw values by 0.00875.

We now need to track this gyro overtime.

```
//Calculate the angles from the gyro
gyroXangle+=rate_gyr_x*DT;
gyroYangle+=rate_gyr_y*DT;
gyroZangle+=rate_gyr_z*DT;
```

DT = loop period. I use a loop period of 20ms. so DT = 0.02.

gyroXangle = is the current X angle calculated from the gyro X data.
I have DT set to 20ms. This is how long it takes to complete one cycle of the main loop. This loop period has to be constant and accurate, otherwise your gyro will drift more then it should.

Accelerometer

```
//Convert Accelerometer values to degrees

AccXangle = (float) (atan2(*(acc_raw+1),*(acc_raw+2))+M_PI)*RAD_TO_DEG;
AccYangle = (float) (atan2(*(acc_raw+2),*acc_raw)+M_PI)*RAD_TO_DEG;
```

The angles can be calculated by using trigonometry and the raw values from the other accelerometer axis. This is done using the Atan2 function to return the principal value of the tangent of the other angles, expressed in radians. We add π to the radians so that we get a result between 0 and 2. We then convert the radians to degrees by multiplying the radians by 57.29578 (180/π).
M_PI = 3.14159265358979323846
RAD_TO_DEG = 57.29578 , 1 radian = 57.29578 degrees

Change rotation value of Accelerometer

```
//Change the rotation value of the accelerometer to -/+ 180
if (AccXangle >180)
{
          AccXangle -= (float)360.0;
}
         if (AccYangle >180)
         AccYangle -= (float)360.0;
```

The above code will convert the values for the X and Y axis on the accelerometer so that the values is 0 when the IMU is upright. This isn't needed, however I like all axis to be 0 when the device is upright.

Combining Angles from the Accelerometer and Gyro

Once you have the both the angles from both sensors, you will have to combined them to overcome the gyro drift and the accelerometer noise.
We can do this by using a filter which will trust the gyro for short periods of time and the accelerometer for longer periods of time.

There are two filers you could use, the Kalman Filter or the Complementary Filter. I used the Complementary filter as it is simple to understand and less CPU intensive. The Kalman filter is way to far complex and would be very processor intensive.

Complementary filter;

Current angle = 98% x (current angle + gyro rotation rate) + (2% * Accelerometer angle)

or

```
CFangleX=AA*(CFangleX+rate_gyr_x*DT) +(1 - AA) * AccXangle;
CFangleY=AA*(CFangleY+rate_gyr_y*DT) +(1 - AA) * AccYangle;
```

CFangleX & CFangleY are the final angles to use.

Timing

The gyro tracking will not be accurate if you can not run a well timed loop.
The code includes the mymillis() function for the timing.

At the start of each loop, we get the current time;

```
startInt = mymillis();
```

then at the end of the loop, we make sure that at least 20ms has passed before continuing;

```
//Each loop should be at least 20ms.
while(mymillis() - startInt < 20)
{
        usleep(100);
}
```

If your loop takes longer than 20ms to run, 45ms for example. Jut update the above value to make sure that every loop cycle runs at the same speed and set this value in DT for the gyro tracking.

ACCELEROMETER GYRO RASPBERRY PI

65 THOUGHTS ON "GUIDE TO INTERFACING A GYRO AND ACCELEROMETER WITH A RASPBERRY PI"

Chris Hudson

MAY 14, 2013 AT 11:16 AM

Is it possible to write the values to a txt file?

mwilliams03

MAY 14, 2013 AT 3:06 PM

I don't see why it isn't possible.
Right at the end of the main loop you would use the values in these variables;
gyroXangle = Gyro X angle
AccXangle = Acce X angle
CFangleX = Filtered X angle, which is the angel to use for your application
gyroYangle = Gyro Y angle

AccYangle = Acce Y angle
CFangleY = Fitlered Y angle, which is the angel to use for your application

vonkohorn
MAY 24, 2013 AT 6:55 AM

This is awesome. Thank you for posting.

m
JUNE 9, 2013 AT 5:47 PM

Did you try something like this using cheap electronics from dx? (e.g. http://dx.com/p/gy-52-mpu-6050-3-axis-gyroscope-triaxial-accelerometer-6-axis-stance-tilt-module-for-arduino-148654)

Pingback: RPi and Accelerometers/Gyros | misperious chunks

JP
JUNE 12, 2013 AT 12:42 AM

Really wonderful. Thanks for documenting the whole work.

Shantanu Bhadoria
SEPTEMBER 3, 2013 AT 9:23 AM

Thanks for this tutorial. I would like to add a correction in your guide. You will also need to install libi2c-dev to be able to properly detect all i2c devices normally.
replace the line :
sudo apt-get install i2c-tools
with :
sudo apt-get install i2c-tools libi2c-dev

If libi2c-dev is missing then i2cdetect will return an empty result without any error This can be confusing and you might wonder if you have connected your multiimu correctly.

PS: I am using a altimu board, which is the same device as minimu except with a added altimeter/barometer chip which also gives you altitude readings. That's useful for my own hobby project.
Cheers!
-Shantanu Bhadoria

★ mwilliams03

SEPTEMBER 3, 2013 AT 10:39 AM

thanks Shantanu, I have updated the blog.

rudo

SEPTEMBER 4, 2013 AT 6:10 AM

This is wonderful posting. Thank you.

I have a question. How do you get DT?
i'm using a STM32 MCU with Keil but I don't konw how to get DT data.
can you help me, please?

★ mwilliams03

SEPTEMBER 4, 2013 AT 8:33 AM

This is something you define. You can see it being defined at the start of the code;

```
#define DT 0.02 // [s/loop] loop period. 20ms
```

Neil

SEPTEMBER 16, 2013 AT 3:21 PM

Hi Mark, Thanks very much for your blog – it's very valuable and interesting. I'm just about to start working on connecting a Mini-IMU-9 to RPi. Likely use will involve movement at speed and rapid acceleration / deceleration. I assume it's very important that the IMU is securely mounted to get accurate readings. Any tips on (or pix of) physical mounting and connectivity of the IMU? Thanks!

★ mwilliams03

SEPTEMBER 18, 2013 AT 4:02 AM

To try and reduce vibrations from the device mine is attached to, I placed a small piece of foam about 2cm square under the IMU. I than used an elastic band to keep it secure. I'm thinking of removing the elastic band and using a screw. Still keeping the foam in-between the device and the IMU.

Neil

SEPTEMBER 18, 2013 AT 5:03 AM

Great, thanks – I will play around with that (once the jumper cables arrive)

Pingback: MinMU-9 v2 (L3GD20) Python Library | mpolaczyk | software, robotics, engineering & geeky stuff...

wime77

OCTOBER 21, 2013 AT 8:58 PM

Colud I rotate raspberry desktop with gyro like tablet ?

codac

NOVEMBER 18, 2013 AT 7:32 PM

My whole project is written in python. I came across the website of Marcin as well (http://mpolaczyk.pl/?p=257). The only thing is that his python code doesn't include the accelerometer and magentometer. Can anyone help me with code snippets in order to add them to Marcin's python script as well?
I'm a pretty bad coder and any help is greatly appreciated…

★ **mwilliams03**

NOVEMBER 19, 2013 AT 4:42 AM

this page may help, it has some python code for the LSM303DLHC ;
http://forums.adafruit.com/viewtopic.php?f=50&t=36212&start=0&hilit=lsm303

Than only thing you would need to worry about is converting the raw data to usable angles:

```
AccXangle = (float) (atan2(*(acc_raw+1),*(acc_raw+2))+M_PI)*RAD_TO_DEG;
AccYangle = (float) (atan2(*(acc_raw+2),*acc_raw)+M_PI)*RAD_TO_DEG;
```

codac

NOVEMBER 23, 2013 AT 1:18 PM

So what I did is using Marcin's Python script in order to get the Gyro data with his script and the following code:

```
# Fetch angle data from MinIMU-9 V2 L3GD20 sensor
# Calculate angle
angleX = 0
angleY = 0
angleZ = 0

dxyz = s.Get_CalOut_Value()
angleX += dxyz[0]*sleepTime;
angleY += dxyz[1]*sleepTime;
angleZ += dxyz[2]*sleepTime;
```

Then I added the LSM303DLHC python code that you’ve suggested and the following code in order to get usable angles:

```
# Fetch Acceleration, Magnetic Gauss, Temperature and Heading from MinIMU-9 V2 LSM303DLHC sensor

acceleration = lsm.readAccelerationsG()
magnetic = lsm.readMagneticsGauss()
temp3 = lsm.readTemperatureCelsius()
heading = lsm.readMagneticHeading()

angleDegX = math.degrees(math.atan2(acceleration.y, acceleration.z))
angleDegY = math.degrees(math.atan2(acceleration.z, acceleration.x))
angleDegZ = math.degrees(math.atan2(acceleration.x, acceleration.y))

# Change the rotation value of the Accelerometer to +/- 180
if (angleDegX > 180):
angleDegX -= 360
if (angleDegY > 180):
angleDegY -= 360
if (angleDegZ > 180):
angleDegZ -= 360

# Complementary filter used to combine the Accelerometer and Gyro values
cfAngleX=0.98*(cfAngleX+angleX)+(1-0.98)*angleDegX
cfAngleY=0.98*(cfAngleY+angleY)+(1-0.98)*angleDegY
cfAngleZ=0.98*(cfAngleZ+angleZ)+(1-0.98)*angleDegZ
```

*Note: I’ve removed the DT as in my case I’ve multiplied it before already (angleX += dxyz[0]*sleepTime;)

The values look like this:

```
Gyro-X-Angle: 0.00
Gyro-Y-Angle: 0.00
Gyro-Z-Angle: 0.00
Accel X: 0.000 G
Accel Y: 92.095 G
Accel Z: -90.000 G
Magn X: 0.166 Gauss
Magn Y: -0.901 Gauss
Magn Z: -0.877 Gauss
CF-Angle X: 0.000
CF-Angle Y: 1.842
CF-Angle Z: -1.800
Heading: -79.551
```

Does that code and the values make sense to you?

★ mwilliams03

NOVEMBER 24, 2013 AT 6:26 AM

Some of your results don't look right. even if the IMU isn't moving, you should have values that are not 0. Its best to start seeing if the raw angles you have are correct. Because if this isn't correct, then you will never have the final angles correct.
The data below is when my IMU is upright an not moving.

```
Loop Time 35 gyrXraw -18 gyrXrate -1.260 gyroXangle -0.078
Loop Time 35 gyrXraw -16 gyrXrate -1.120 gyroXangle -0.118
Loop Time 35 gyrXraw -11 gyrXrate -0.770 gyroXangle -0.145
Loop Time 35 gyrXraw -19 gyrXrate -1.330 gyroXangle -0.191
Loop Time 35 gyrXraw -15 gyrXrate -1.050 gyroXangle -0.228
Loop Time 35 gyrXraw -12 gyrXrate -0.840 gyroXangle -0.257
Loop Time 35 gyrXraw -13 gyrXrate -0.910 gyroXangle -0.289
Loop Time 35 gyrXraw -20 gyrXrate -1.400 gyroXangle -0.338
Loop Time 35 gyrXraw -16 gyrXrate -1.120 gyroXangle -0.377
Loop Time 35 gyrXraw -16 gyrXrate -1.120 gyroXangle -0.417
Loop Time 35 gyrXraw -15 gyrXrate -1.050 gyroXangle -0.453
Loop Time 35 gyrXraw -18 gyrXrate -1.260 gyroXangle -0.497
Loop Time 35 gyrXraw -14 gyrXrate -0.980 gyroXangle -0.532
Loop Time 35 gyrXraw -12 gyrXrate -0.840 gyroXangle -0.561
Loop Time 35 gyrXraw -17 gyrXrate -1.190 gyroXangle -0.603
Loop Time 35 gyrXraw -15 gyrXrate -1.050 gyroXangle -0.639
Loop Time 35 gyrXraw -11 gyrXrate -0.770 gyroXangle -0.666
Loop Time 35 gyrXraw -20 gyrXrate -1.400 gyroXangle -0.715
Loop Time 35 gyrXraw -17 gyrXrate -1.190 gyroXangle -0.757
```

Pingback: Google Alert – RaspberryPi | Raspberry Pi Made Easy

codac

NOVEMBER 27, 2013 AT 8:17 PM

hey mwilliams03,

the reason for the 0-values of the gyro is that at that point in time it already has been calibrated with no movement.

So, I'll take out the gyro and keep the Accelerometer, Magnetometer and heading.

The raw values look like this (except for the CF and heading values which are not raw of course):
Accel X: 1.160 G
Accel Y: 86.756 G
Accel Z: 70.346 G
Magn X: -0.363 Gauss
Magn Y: -0.644 Gauss
Magn Z: -0.619 Gauss
CF-Angle X: 0.023
CF-Angle Y: 1.735
CF-Angle Z: 1.407
Heading: -119.363

How do I know or how can I check if those values make any sense.
Are there proper values I can check against with, any further measurements, or websites with calculation tools!?

★ mwilliams03
NOVEMBER 28, 2013 AT 2:11 AM

Even without movement, the gyro will drift. My raw log i pasted in is when the IMU is not moving, however the values for gyro and accelerometer are still changing. This is normal.
Can you log the results for about 30 loops and include the gyro raw

Ami
NOVEMBER 28, 2013 AT 8:35 AM

Thanks so much for this great post. Just one question:
For the gyro, how do you know that the “register address order is X,Z,Y with high bytes first”?

I wasn’t able to deduce this by looking at the datasheet. In fact, the orders of sensor readings for the gyro seem exactly like the accelerometer (according to both their datasheets).

★ mwilliams03
NOVEMBER 28, 2013 AT 9:43 AM

you are absolutely right. The gyro is the same as the accelerometer.
We worked this out some time ago, I removed the statement from the GIT repository.. but I forgot to update the blog.

BTW: it is the MAG that is xzy.

Hugo

DECEMBER 17, 2013 AT 2:43 PM

Hi, your work is very great: Thank you for that!

If i would like to have the acceleration in the three directions of space, did i just want to take *acc_raw, *(acc_raw+1) and *(acc_raw+2) and put them in a file, or not? With these values, it should be easy to have the speed and the position of my device, isn't it?

Thank you

★ mwilliams03

DECEMBER 17, 2013 AT 10:23 PM

Yes and no. 😃
You can measure the acceleration in Gs. Which you then can use to calculate distance and direction over time. However, you will notice drift and the drift gets worse,to the point that it is not useful anymore. This is due to the noise of the accelerometer.
If you are still interested, google "accelerometer double integration"

codac

DECEMBER 22, 2013 AT 2:04 PM

Here is my gyro raw data (somehow I was not able to reply to your last comment):

```
Calibrating axis X, please do not move sensor…
Done: (min=-334;mean=-319.85;max=-304)
Calibrating axis Y, please do not move sensor…
Done: (min=74;mean=94.25;max=112)
Calibrating axis Z, please do not move sensor…
Done: (min=-511;mean=-320.35;max=21)

(-15.149999999999977, 0, 0)
(-16.149999999999977, 30.75, 0)
(16.850000000000023, 0, 0)
(30.850000000000023, 0, 0)
(16.850000000000023, 0, 0)
(23.850000000000023, 0, 0)
(-16.149999999999977, 24.75, 0)
(-23.149999999999977, 0, 0)
(24.850000000000023, 0, 0)
(16.850000000000023, -25.25, 0)
```

```
(20.850000000000023, 0, 0)
(24.850000000000023, 0, 0)
(0, 0, 345.35000000000002)
```

where it is (x, y, z)

★ mwilliams03

DECEMBER 23, 2013 AT 12:51 AM

Mmmm.. the values don't look right…are they the raw values?
The raw values from the gyro are read as 2's complement . The results should be a whole number, not a decimal.

codac

DECEMBER 23, 2013 AT 2:29 PM

Yea, I thought the same thing… I modified the code a bit (using https://github.com/mpolaczyk/Gyro-L3GD20-Python/blob/master/Example_ReadRawData.py) so that it displays not only the z but x and y axis as well.

Here is the modified part:

```
s.CalibrateX()
s.CalibrateY()
s.CalibrateZ()

plotEnabled = False
if plotEnabled:
plt.subplot(111)

# Read values loop
starttime = time.time() # this is our start time
t = 0
prev_x = 0
prev_y = 0
prev_z = 0
tmax = 10
T = []
X = []
Y = []
Z = []
while t < tmax:
while s.Get_AxisDataAvailable_Value()[2] == 0:
time.sleep(0.001)
```

```
t = time.time() – starttime
x = s.Get_CalOutX_Value()
y = s.Get_CalOutY_Value()
z = s.Get_CalOutZ_Value()
T.append(t)
X.append(x)
Y.append(y)
Z.append(z)

print(x, y, z);

if plotEnabled:
plt.plot(T, X, 'ro')
plt.plot(T, Y, 'ro')
plt.plot(T, Z, 'ro')

plt.show()
```

★ mwilliams03

DECEMBER 26, 2013 AT 2:52 AM

I had a look through that code.
The actual values are not raw, as the values being printed to the screen are after the GAIN has been applied.
And I think the wrong gain is being applied.
I dont use python often, however I have updated the code in two files;
Example_ReadRawData.py http://pastebin.com/6fzZv3AS
and
L3GD20.py, I only made a change to Get_RawOutX_Value(), I hard coded the correct GAIN value.
http://pastebin.com/1tf16bqW

codac

DECEMBER 27, 2013 AT 1:20 PM

Wow, thank you SO much for your help!
Using your code, the result looks like this:

```
Calibrating axis X, please do not move sensor…
Done: (min=-5.53;mean=-5.159875;max=-4.83)
Initiating…
Gain set to:0.00875
Gyro X Raw 0 Gyro X Turn Rate = 0.000000 Gyro X Angle 9.900000
```

```
loopTime 0.020137 Gyro X Raw 0 Gyro X Turn Rate = 0.000000 Gyro X Angle 9.943295
loopTime 0.020088 Gyro X Raw 0 Gyro X Turn Rate = 0.000000 Gyro X Angle 9.943295
loopTime 0.020129 Gyro X Raw 0 Gyro X Turn Rate = 0.000000 Gyro X Angle 9.943295
loopTime 0.020107 Gyro X Raw 0 Gyro X Turn Rate = 0.000000 Gyro X Angle 9.943295
loopTime 0.020104 Gyro X Raw 0 Gyro X Turn Rate = 0.000000 Gyro X Angle 9.943295

loopTime 0.020037 Gyro X Raw 0 Gyro X Turn Rate = -0.008452 Gyro X Angle 9.942490
```

Gyro X Raw is always "0", Gyro X Turn Rate ist mostly "0" but gets the value "0.007647" and "-0.008452″ during the whole run.

★ **mwilliams03**

DECEMBER 27, 2013 AT 10:52 PM

That still doesn't look right.. Raw can sometimes be 0.. but not all the time, the noise should cause it to show other values.
I made some more changes to a number of files and uploaded to my site. You can get the new files like this;
wget http://www.ozzmaker.com/downloads/Gyro-L3GD20-Python-update.tar
Untar them and run Example_ReadRaw.
This is what my output looks like when the IMU is not moving;

```
loopTime 0.020520 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.017640
loopTime 0.020132 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.017640
loopTime 0.020053 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.017640
loopTime 0.020068 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.017640
loopTime 0.020074 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.017640
loopTime 0.020028 raw 00006 @@@GAIN@@@ 0.00875 xrate 0.483000 Xangle = 0.002940
loopTime 0.020059 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.002940
loopTime 0.020017 raw 00006 @@@GAIN@@@ 0.00875 xrate 0.483000 Xangle = 0.012600
loopTime 0.020027 raw 00007 @@@GAIN@@@ 0.00875 xrate 0.553000 Xangle = 0.023660
loopTime 0.020123 raw -0009 @@@GAIN@@@ 0.00875 xrate -0.637000 Xangle = 0.010920
loopTime 0.020036 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.010920
loopTime 0.020024 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.010920
```

And this is when I rotate it 180 degrees;

```
loopTime 0.020084 raw -0225 @@@GAIN@@@ 0.00875 xrate -15.816500 Xangle = 3.250660
loopTime 0.020027 raw -0590 @@@GAIN@@@ 0.00875 xrate -41.366500 Xangle = 2.423330
loopTime 0.020075 raw -4575 @@@GAIN@@@ 0.00875 xrate -320.316500 Xangle = -145.672520
loopTime 0.020071 raw -4105 @@@GAIN@@@ 0.00875 xrate -287.416500 Xangle = -151.420850
loopTime 0.020083 raw -1904 @@@GAIN@@@ 0.00875 xrate -133.346500 Xangle = -154.087780
loopTime 0.020039 raw -1489 @@@GAIN@@@ 0.00875 xrate -104.296500 Xangle = -156.173710
loopTime 0.020108 raw -1297 @@@GAIN@@@ 0.00875 xrate -90.856500 Xangle = -157.990840
loopTime 0.020017 raw -1771 @@@GAIN@@@ 0.00875 xrate -124.036500 Xangle = -172.494420
loopTime 0.020124 raw -1690 @@@GAIN@@@ 0.00875 xrate -118.366500 Xangle = -174.861750
loopTime 0.020024 raw -1245 @@@GAIN@@@ 0.00875 xrate -87.216500 Xangle = -176.606080
loopTime 0.020041 raw -0365 @@@GAIN@@@ 0.00875 xrate -25.616500 Xangle = -177.118410
```

codac

DECEMBER 29, 2013 AT 12:35 PM

Brilliant!

This is my output when IMU is not moving at all:

```
loopTime 0.020092 raw 00003 @@@GAIN@@@ 0.00875 xrate 0.262500 Xangle = 0.136150
loopTime 0.020135 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.136150
loopTime 0.020144 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.157150
loopTime 0.020038 raw 00003 @@@GAIN@@@ 0.00875 xrate 0.262500 Xangle = 0.162400
loopTime 0.020033 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
loopTime 0.020098 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
loopTime 0.020121 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
loopTime 0.020113 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
loopTime 0.020108 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
loopTime 0.020087 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
loopTime 0.020035 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.162400
```

raw is mostly “0” but changes to “0003” inbetween.
xrate is mostly 0.00000 but changes to “0.262500” inbetween.
looks much better now i guess.

This is what I get when I move the IMU 180 degrees:

```
loopTime 0.020037 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.000000
loopTime 0.020067 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = 0.000000
loopTime 0.020051 raw -0016 @@@GAIN@@@ 0.00875 xrate -1.141000 Xangle = -0.467040
loopTime 0.020034 raw -0013 @@@GAIN@@@ 0.00875 xrate -0.931000 Xangle = -0.485660
loopTime 0.020053 raw -0013 @@@GAIN@@@ 0.00875 xrate -0.931000 Xangle = -0.504280
loopTime 0.020019 raw -0016 @@@GAIN@@@ 0.00875 xrate -1.141000 Xangle = -0.527100
loopTime 0.020027 raw -0020 @@@GAIN@@@ 0.00875 xrate -1.421000 Xangle = -0.555520
loopTime 0.020054 raw -0032 @@@GAIN@@@ 0.00875 xrate -2.261000 Xangle = -1.220800
loopTime 0.020076 raw 00006 @@@GAIN@@@ 0.00875 xrate 0.469000 Xangle = -2.313080
loopTime 0.020106 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = -2.313080
loopTime 0.020063 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = -2.313080
loopTime 0.020094 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = -2.313080
loopTime 0.020109 raw 00006 @@@GAIN@@@ 0.00875 xrate 0.469000 Xangle = -2.303700
loopTime 0.020116 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = -2.303700
```

codac

JANUARY 17, 2014 AT 12:36 PM

Hi Mark,
any news on my last results?
It might be easier to do a Skype session (or something similar) in order to finally get the IMU working.
Would that be possible? I can't find any properly working solution on Python basis including the gyro, accelerometer and magenetometer on the whole www.

★ mwilliams03

JANUARY 17, 2014 AT 1:11 PM

those last results look good. What does it look like when you turn the IMU 90 degrees along the X-asis?

codac

JANUARY 18, 2014 AT 8:58 AM

Please have a look of the second part of those values, beginning with: "This is what I get when I move the IMU 180 degrees:"

★ mwilliams03

JANUARY 19, 2014 AT 1:03 AM

sorry… I missed that.
It doesn't look correct.
Can you tar up all the source, send it to me and I'll see if it works with my IMU.
mark< @>ozzmaker.com

codac

FEBRUARY 1, 2014 AT 9:13 AM

did you receive my files?

★ mwilliams03

FEBRUARY 1, 2014 AT 9:43 PM

yes I did. I replied the same day, but it must not have gone through.

I downloaded your code… uncompressed it and ran it without changing anything. And it works.

Maybe there is something wrong with your IMU?
Are you rotating it at all along the X?
what IMU do you have?

This is me rotating it 180 degrees;

```
loopTime 0.020065 raw -0139 @@@GAIN@@@ 0.00875 xrate -9.789500 Xangle = -0.393540
loopTime 0.020126 raw -0223 @@@GAIN@@@ 0.00875 xrate -15.669500 Xangle = -0.706930
loopTime 0.020081 raw -0665 @@@GAIN@@@ 0.00875 xrate -46.609500 Xangle = -1.639120
loopTime 0.020132 raw -1152 @@@GAIN@@@ 0.00875 xrate -80.699500 Xangle = -3.253110
loopTime 0.020096 raw -2035 @@@GAIN@@@ 0.00875 xrate -142.509500 Xangle = -6.103300
loopTime 0.020115 raw -3571 @@@GAIN@@@ 0.00875 xrate -250.029500 Xangle = -11.103890
loopTime 0.020051 raw -6191 @@@GAIN@@@ 0.00875 xrate -433.429500 Xangle = -19.772480
loopTime 0.020075 raw -8905 @@@GAIN@@@ 0.00875 xrate -623.409500 Xangle = -32.240670
loopTime 0.020046 raw -12687 @@@GAIN@@@ 0.00875 xrate -888.149500 Xangle = -50.003660
loopTime 0.020111 raw -6898 @@@GAIN@@@ 0.00875 xrate -482.919500 Xangle = -59.662050
loopTime 0.020061 raw -3447 @@@GAIN@@@ 0.00875 xrate -241.349500 Xangle = -64.489040
loopTime 0.020035 raw -2668 @@@GAIN@@@ 0.00875 xrate -186.819500 Xangle = -68.225430
loopTime 0.020068 raw -2813 @@@GAIN@@@ 0.00875 xrate -196.969500 Xangle = -72.164820
loopTime 0.020032 raw -2738 @@@GAIN@@@ 0.00875 xrate -191.719500 Xangle = -75.999210
loopTime 0.020037 raw -3229 @@@GAIN@@@ 0.00875 xrate -226.089500 Xangle = -80.521000
loopTime 0.020121 raw -3053 @@@GAIN@@@ 0.00875 xrate -213.769500 Xangle = -84.796390
loopTime 0.020195 raw -1975 @@@GAIN@@@ 0.00875 xrate -138.309500 Xangle = -87.562580
loopTime 0.020119 raw -1373 @@@GAIN@@@ 0.00875 xrate -96.169500 Xangle = -89.485970
loopTime 0.020135 raw -1109 @@@GAIN@@@ 0.00875 xrate -77.689500 Xangle = -91.039760
loopTime 0.020187 raw 01157 @@@GAIN@@@ 0.00875 xrate 81.000500 Xangle = -89.419750
loopTime 0.020131 raw -6097 @@@GAIN@@@ 0.00875 xrate -426.849500 Xangle = -97.956740
loopTime 0.020018 raw -10283 @@@GAIN@@@ 0.00875 xrate -719.869500 Xangle = -112.354130
loopTime 0.020059 raw -12267 @@@GAIN@@@ 0.00875 xrate -858.749500 Xangle = -129.529120
loopTime 0.020116 raw -11152 @@@GAIN@@@ 0.00875 xrate -780.699500 Xangle = -145.143110
loopTime 0.020110 raw -9977 @@@GAIN@@@ 0.00875 xrate -698.449500 Xangle = -159.112100
loopTime 0.020123 raw -3595 @@@GAIN@@@ 0.00875 xrate -251.709500 Xangle = -164.146290
loopTime 0.020082 raw -1933 @@@GAIN@@@ 0.00875 xrate -135.369500 Xangle = -166.853680
loopTime 0.020390 raw -0856 @@@GAIN@@@ 0.00875 xrate -59.979500 Xangle = -168.053270
loopTime 0.020095 raw -0296 @@@GAIN@@@ 0.00875 xrate -20.779500 Xangle = -168.468860
loopTime 0.020087 raw -0058 @@@GAIN@@@ 0.00875 xrate -4.119500 Xangle = -168.551250
loopTime 0.020115 raw -0102 @@@GAIN@@@ 0.00875 xrate -7.199500 Xangle = -168.695240
loopTime 0.020111 raw -0269 @@@GAIN@@@ 0.00875 xrate -18.889500 Xangle = -169.073030
loopTime 0.020026 raw -0170 @@@GAIN@@@ 0.00875 xrate -11.959500 Xangle = -169.312220
loopTime 0.020102 raw -0065 @@@GAIN@@@ 0.00875 xrate -4.609500 Xangle = -169.404410
loopTime 0.020305 raw -0017 @@@GAIN@@@ 0.00875 xrate -1.249500 Xangle = -169.429400
loopTime 0.020159 raw -0024 @@@GAIN@@@ 0.00875 xrate -1.739500 Xangle = -169.464190
loopTime 0.020058 raw -0057 @@@GAIN@@@ 0.00875 xrate -4.049500 Xangle = -169.545180
loopTime 0.020094 raw -0043 @@@GAIN@@@ 0.00875 xrate -3.069500 Xangle = -169.606570
loopTime 0.020126 raw -0022 @@@GAIN@@@ 0.00875 xrate -1.599500 Xangle = -169.638560
loopTime 0.020101 raw -0010 @@@GAIN@@@ 0.00875 xrate -0.759500 Xangle = -169.653750
loopTime 0.020097 raw -0009 @@@GAIN@@@ 0.00875 xrate -0.689500 Xangle = -169.667540
loopTime 0.020080 raw -0011 @@@GAIN@@@ 0.00875 xrate -0.829500 Xangle = -169.684130
loopTime 0.020088 raw -0015 @@@GAIN@@@ 0.00875 xrate -1.109500 Xangle = -169.706320
loopTime 0.020104 raw -0029 @@@GAIN@@@ 0.00875 xrate -2.089500 Xangle = -169.748110
loopTime 0.020037 raw -0020 @@@GAIN@@@ 0.00875 xrate -1.459500 Xangle = -169.777300
loopTime 0.020086 raw -0011 @@@GAIN@@@ 0.00875 xrate -0.829500 Xangle = -169.793890
loopTime 0.020066 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = -169.793890
loopTime 0.020085 raw -0010 @@@GAIN@@@ 0.00875 xrate -0.759500 Xangle = -169.809080
loopTime 0.020030 raw -0009 @@@GAIN@@@ 0.00875 xrate -0.689500 Xangle = -169.822870
loopTime 0.020120 raw -0010 @@@GAIN@@@ 0.00875 xrate -0.759500 Xangle = -169.838060
loopTime 0.020063 raw 00000 @@@GAIN@@@ 0.00875 xrate 0.000000 Xangle = -169.838060
loopTime 0.020014 raw -0010 @@@GAIN@@@ 0.00875 xrate -0.759500 Xangle = -169.853250
loopTime 0.020076 raw -0009 @@@GAIN@@@ 0.00875 xrate -0.689500 Xangle = -169.867040
loopTime 0.020025 raw -0011 @@@GAIN@@@ 0.00875 ^Cxrate -0.829500 Xangle = -169.883630
```

codac

FEBRUARY 2, 2014 AT 7:49 AM

I’m using the MinIMU-9 V2 (http://www.pololu.com/product/1268 using the L3GD20 and LSM303DLHC) on a breadboard which is connected via a flatcable to my RasPi and the RasPi is connected with the ethernet and power cables. Using the cables, it’s a bit hard to do a proper 180° turn. So that might be the issue.

Ok the code outputs the values for the Gyro now, adding the accelerometer and magnetometer would be an amazing benefit and my project would finally reach the finishing line, thanks to you mark!
Would it be possible to add those values as well?

Cyril

JANUARY 22, 2014 AT 8:56 PM

Thank you for this post !
The code works as well with “FLORA Accelerometer/Compass Sensor – LSM303″
3 lines to put in comments since there is no gyro :
main.c line 114 : // readGYR(Pgyr_raw);
sensor.c line 130 : // writeGyrReg(L3G_CTRL_REG1, 0b00001111);
sensor.c line 131 : // writeGyrReg(L3G_CTRL_REG4, 0b00110000);
And replace :
main.c line 149 by : printf (” AccXangle \e[m %7.3f \t AccYangle %7.3f \n”,AccXangle,AccYangle);
Now I can play!

Harpreet Singh

FEBRUARY 6, 2014 AT 5:27 AM

i need to interface the raspberry pi with accelerometer…. i need coding to interface that. can you help me…
my mail id is hsjabbal19@yahoo.com

★ **mwilliams03**

FEBRUARY 10, 2014 AT 12:35 AM

what type of accelerometer?

Ahmad Nawaz

MARCH 3, 2014 AT 6:40 PM

I am working at Self balancing robot.Can u give me the code of this robot..?
my email id is kpalizai08@gmail.com

Ahmad Nawaz

MARCH 3, 2014 AT 6:43 PM

Also please tell me the setting of accelerometer sensitivity in c laguange code?
pleasee…………………..

★ **mwilliams03**

MARCH 3, 2014 AT 11:27 PM

Have a look here,
http://ozzmaker.com/2013/08/12/pibbot-v2-balancing-robot-using-a-raspberry-pi/

There is a link to the code on the above page

Tyrone Robinson

JUNE 15, 2014 AT 3:41 AM

Hello I am new to the raspberry pi and I am looking forward to building my Pibbot. I purchased all the parts needed to construct it along with the LCD 20×2 + extras. I am having a hard time understanding the pins configuration for all the components to the raspberry pi. I see in the Pibbot V2 master there is a read me for the pins, but I am having a hard time understanding it. Could you please put it in dummy form so that In may get my Pibbot balancing. Thank you.

★ mwilliams03

JUNE 15, 2014 AT 12:38 PM

I would try and get the IMU going first.
The instructions for this are on the web page.

Tyrone

JUNE 24, 2014 AT 11:07 AM

Hello just a little confused about a couple of things. (Actually 5)
1. Is LCD_E connected to GPIO 0(SDA) along with the IMU?
2. Encoder pin 2 says it is going to GPIO 1 which is 3V3. Is that right? And if thats true are you using one or two encoders?
3. Are Wheel gain , and LCD_7 both grouped together on GPIO pin 5.
4. Is LCD_ Green going to GPIO pin 2 witch is 5 volts ?
5. And am I grounding backlights Red, Green, and Blue?
Thank you for you help. If you could please explain in detail. I would appreciate it.

★ mwilliams03

JUNE 24, 2014 AT 1:36 PM

I see your issue. You are GPIO pin numbers.. when you should be using WiringPi pin numbers.
http://wiringpi.com/pins/
1. These are WiringPi pin numbers. WiringPi = 0, BCM = 17, GPIO = 0
2. Using WiringPi numbers, pin 1 is GPIO1 (BCM18).
3. That WHEELGAIN is commented out.. so it isnt being used. You can delete it. WiringPi pin number 5 (BCM24).
4. Again, WiringPi pin numbers
5. Only ground the colour you want to use for the back light.

Tyrone

JUNE 24, 2014 AT 10:51 PM

Thank you so very much for the quick reeply.

Jim Keane

AUGUST 8, 2014 AT 6:57 PM

Hey there, love the code. It’s worked best out of all the options I’ve come across in my search!

I’m trying to now adapt the imu data for use in an AUV…but for some reason whatever changes I make to the main.c file don’t come into effect when I run

sudo ./main

Could someone please point me in the direction of a python tute which could explain what i’m doing wrong – i don’t doubt it’s elementary…but that’s where I’m stuck!

Regards and thanks,

Jim

★ mwilliams03

AUGUST 8, 2014 AT 8:38 PM

Hi Jim
I’m not aware of any python tutorials, but I do know there is some Python code out there on the Net for the Adafruit IMU. I just had a look but I couldn’t find it, sorry.

Are you recompiling after you make the changes?

BTW: This may interest you, I will soon post eanentry on how to use the magnetometer and accelerometer as a tilt compensated compass.

Tiago

OCTOBER 11, 2014 AT 2:30 AM

Hi Mark, thanks alot for the tutorial, but something’s not working here…
I’ve already debugged the code and it seems that things are going wrong, as soon as I call “writeGyrReg(L3G_CTRL_REG1, 0b00001111)”, here
i2c_smbus_access, in the return statement “ioctl(file,I2C_SMBUS,&args);”
is it because of the I2C_SMBUS?

I have a similar IMU, the ALTIMU, the devices aren’t the same, except the gyro, that’s the only thing that I’m trying to teste.

Thanks,
Tiago

★ Mark Williams

OCTOBER 11, 2014 AT 11:09 PM

Does the i2c address in the code match up with the i2c addresses used by your sensors?

James Elger

OCTOBER 14, 2014 AT 6:27 AM

It looks like you've invented the protester's 3rd arm. If anything, that thing could march around and fly a sign.

Paul Cox

OCTOBER 31, 2014 AT 4:25 AM

Thanks a bunch for the code. By the way this is useful on any Linux board with i2c bus, not just RasPi. For example, works on Minnowboard MAX and would also for beaglebone….

Pingback: Accelerometru/giroscop – Raspberry PI | Robofun Blog

Shrestha Nishu

JANUARY 14, 2015 AT 6:07 PM

Do you have something similar for Arduino???

★ Mark Williams

JANUARY 14, 2015 AT 6:10 PM

I haven't sorry.
The math and IMU registers would be the same, the code would just have to be ported over to Arduino

gary morton

APRIL 1, 2015 AT 3:37 AM

First, let me thank you very much for the hardware product, and also the python files. Everything worked for me!
The inclinometer functions perfectly when rotated about the x and y axis. So giving the x and y angles you use in the example showing the inclination of the car.
I had hoped to use the device to also allow me to determine the rotation of the device. ie keeping the device flat and level, and rotating it to measure the direction. For such a orientation CF angle X and Y remain constant.
I assume I need to compute a different set of angles, but cannot see which combination to use. I set up the equivalent of CF Angle Z, but that too stays constant. (which at least i did expect!).
Would you have any suggestions?

★ Mark Williams

APRIL 1, 2015 AT 6:45 PM

To measure rotation around the Z axis , you will need to use the gyro Z axis and the Compass (Magnetometer) heading axis. First try with just the compass heading, this may be good enough.

Here, the heading is calculated using python;
https://github.com/mwilliams03/BerryIMU/tree/master/python-LSM9DS0-gryo-accel-compass

Here it is calculated using C;
https://github.com/mwilliams03/BerryIMU/tree/master/compass_tutorial01_basics
And with tilt compensation;
https://github.com/mwilliams03/BerryIMU/tree/master/compass_tutorial02_tilt_compensation

BTW: you can not use the accelerometer to measure rotation around the Z axis as the accelerometer measures gravity, gravity doesn't change when it is being rotated on this axis.

gary morton

APRIL 1, 2015 AT 7:56 PM

Thank you for your prompt reply. I suspected that it was not possible.
I did make use of your python program and used the heading. And of course I found that the motors in the machine were distorting the magnetic field, so although the heading did change as the unit rotated, it was not in correct relation!
I will now investigate how to mount the IMU where it is less impacted by the motors fields. Or even try to screen the motors a bit.
Regards

★ Mark Williams

APRIL 1, 2015 AT 9:15 PM

For the compass, you could try calibration for hard iron distortion.. But to be honest, I not sure what the magnet field of a spinning motor would look like.

If you want to have a look at some information on compass calibration;
`http://ozzmaker.com/2015/01/23/compass3/`

akbar

APRIL 3, 2015 AT 2:42 PM

nice tutorial,but im so confused to run that main.c?
how to run your program,im first time use program c in raspberry

like python to run like “python ./(name file).py
how to program c?

thanks alot dude