

Bitify

Tinkering with the Raspberry Pi and other geeky stuff

Saturday, 16 November 2013

3D OpenGL visualisation of the data from an MPU-6050 connected to a Raspberry Pi

In this post I'll show how to serve the data over http and display a 3D representation in OpenGL extending on a [previous blog post](#) detailing how to read data from the MPU-6050 sensor and convert it into a something useful.

Using a simple web server to serve up the data

Let's start by setting up a simple server based on [web.py](#), which is installed via apt-get

```
sudo apt-get install python-webpy
```

Now create a directory to put the code in and create a simple test program

```
mkdir webpy
cd webpy
vi server.py
```

Use the following as a test

```
01 #!/usr/bin/python
02 import web
03
04 urls = (
05     '/', 'index'
06 )
07
08 class index:
09     def GET(self):
10         return "Hello, world!"
11
12 if __name__ == "__main__":
13     app = web.application(urls, globals())
14     app.run()
```

Save the and then set it as executable with

```
chmod +x server.py
```

and then run the code

```
./server.py
```

you will see something like this showing the server is waiting for a request (pressing Ctrl+C will stop the server)

```
http://0.0.0.0:8080/
```

Now point your browser at <http://ip-address-of-your-pi:8080> and it will show a web page with the content of **Hello, world!**. We can make use of this to read data from a remote machine, in my case my Linux desktop.

Adding the sensor code to the server

Replace the contents of server.py with

```
01 #!/usr/bin/python
02 import web
03 import smbus
04 import math
05
06 urls = (
07     '/', 'index'
08 )
09
10 # Power management registers
11 power_mgmt_1 = 0x6b
12 power_mgmt_2 = 0x6c
13
14 bus = smbus.SMBus(0) # or bus = smbus.SMBus(1) for Revision 2 boards
15 address = 0x68 # This is the address value read via the i2cdetect command
16
17
18 def read_byte(adr):
19     return bus.read_byte_data(address, adr)
20
21 def read_word(adr):
22     high = bus.read_byte_data(address, adr)
23     low = bus.read_byte_data(address, adr+1)
24     val = (high << 8) + low
25     return val
26
27 def read_word_2c(adr):
28     val = read_word(adr)
29     if (val >= 0x8000):
```

Search

Popular Posts



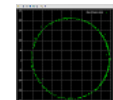
Interfacing Raspberry Pi and MPU-6050

I wanted to interface my Pi to a Six-Axis Gyro + Accelerometer sensor and the one I settled on was based on a MPU-6050 chip. I went for thi...



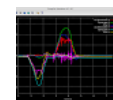
Reading data from the MPU-6050 on the Raspberry Pi

In a previous post I showed how to connect an Accelerometer & Gyro sensor to the Raspberry Pi, in this post I'll show some simple P...



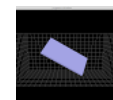
Connecting and calibrating a HMC5883L Compass on the Raspberry Pi

Here is how to connect a HMC5883L Compass to the Raspberry Pi, calibrate it and read the data. Connecting the compass is simple enough, fo...



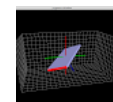
Using a complementary filter to combine Accelerometer and Gyroscopic data

This post shows how to combine data from the accelerometer and gyroscope using a complementary filter to produce a better readings from the...



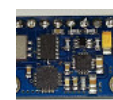
3D OpenGL visualisation of the data from an MPU-6050 connected to a Raspberry Pi

In this post I'll show how to serve the data over http and display a 3D representation in OpenGL extending on a [previous blog post](#) det...



Pitch, Roll and Yaw using MPU6050 & HMC5883L (with tilt compensation and complementary filter)

Combining the data from an MPU605 and a HMC5883L to give tilt compensated pitch, roll and yaw. Pitch, roll and yaw (with tilt compensati...



GY80 (L3G4200D, ADXL345, HMC5883L, BMP085) Python library for Raspberry Pi

A while back I bought a GY80 board, which comprises of: L3G4200D - Three axis Gyroscope ADXL345 - Three axis accelerometer HMC5883L - C...



Temperature logging with a DS18B20 and a Raspberry Pi

I wanted to do some temperature logging so I hooked up a DS18B20 temperature sensor to a Raspberry Pi. About the DS18B20 Dallas DS18B...

```

30     return -((65535 - val) + 1)
31 else:
32     return val
33
34 def dist(a,b):
35     return math.sqrt((a*a)+(b*b))
36
37 def get_y_rotation(x,y,z):
38     radians = math.atan2(x, dist(y,z))
39     return -math.degrees(radians)
40
41 def get_x_rotation(x,y,z):
42     radians = math.atan2(y, dist(x,z))
43     return math.degrees(radians)
44
45
46 class index:
47     def GET(self):
48         accel_xout = read_word_2c(0x3b)
49         accel_yout = read_word_2c(0x3d)
50         accel_zout = read_word_2c(0x3f)
51
52         accel_xout_scaled = accel_xout / 16384.0
53         accel_yout_scaled = accel_yout / 16384.0
54         accel_zout_scaled = accel_zout / 16384.0
55
56         return str(get_x_rotation(accel_xout_scaled, accel_yout_scaled,
57 accel_zout_scaled))+" "+str(get_y_rotation(accel_xout_scaled, accel_yout_scaled,
58 accel_zout_scaled))
59
60 if __name__ == "__main__":
61     # Now wake the 6050 up as it starts in sleep mode
62     bus.write_byte_data(address, power_mgmt_1, 0)
63
64     app = web.application(urls, globals())
65     app.run()

```

The server has to be run as sudo so you have permissions to read from the I²C

```
sudo ./server.py
```

Connecting via your browser will now return the X & Y rotation values e.g.

```
-28.7291281627 -39.4833542336
```

3D visualisation

I'm using a Linux desktop and that is all I have tested this simple code on, I've no idea if it works on Windows or Macs and it certainly won't run on the Pi itself. I'm no OpenGL guru so this code is just hacked together to get something visible.

Setting up OpenGL and pygame

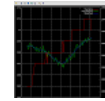
```
sudo apt-get install python-opengl
sudo apt-get install python-pygame
```

Now save the following to a file (in my case level.py) and run it

```

001 #!/usr/bin/python
002
003 import pygame
004 import urllib
005 from OpenGL.GL import *
006 from OpenGL.GLU import *
007 from math import radians
008 from pygame.locals import *
009
010 SCREEN_SIZE = (800, 600)
011 SCALAR = .5
012 SCALAR2 = 0.2
013
014 def resize(width, height):
015     glViewport(0, 0, width, height)
016     glMatrixMode(GL_PROJECTION)
017     glLoadIdentity()
018     gluPerspective(45.0, float(width) / height, 0.001, 10.0)
019     glMatrixMode(GL_MODELVIEW)
020     glLoadIdentity()
021     gluLookAt(0.0, 1.0, -5.0,
022              0.0, 0.0, 0.0,
023              0.0, 1.0, 0.0)
024
025 def init():
026     glEnable(GL_DEPTH_TEST)
027     glClearColor(0.0, 0.0, 0.0, 0.0)
028     glShadeModel(GL_SMOOTH)
029     glEnable(GL_BLEND)
030     glEnable(GL_POLYGON_SMOOTH)
031     glHint(GL_POLYGON_SMOOTH_HINT, GL_NICEST)
032     glEnable(GL_COLOR_MATERIAL)
033     glEnable(GL_LIGHTING)
034     glEnable(GL_LIGHT0)
035     glLightfv(GL_LIGHT0, GL_AMBIENT, (0.3, 0.3, 0.3, 1.0));
036
037 def read_values():
038     link = "http://192.168.1.65:8080" # Change this address to your settings
039     f = urllib.urlopen(link)
040     myfile = f.read()

```



Interfacing a BMP085 Digital Pressure sensor to the Raspberry Pi

I recently bought a sensor with a BMP085 Digital Pressure sensor on it so I thought I'd write a post on how to read the data from the R...

Labels

1-wire (1)
ADXL345 (1)
BMP085 (2)
DS18B20 (1)
gnuplot (3)
GY80 (1)
HMC5883L (3)
L3G4200D (1)
MPU-6050 (6)
OpenGL (2)
Python (8)
Raspberry Pi (8)
Raspbian (4)
temperature (1)

Blog Archive

► 2014 (2)
▼ 2013 (7)
► December (1)
▼ November (6)
Interfacing a BMP085 Digital Pressure sensor to th...
Connecting and calibrating a HMC5883L Compass on t...
Using a complementary filter to combine Accelerome...
3D OpenGL visualisation of the data from an MPU-60...
Reading data from the MPU-6050 on the Raspberry Pi...
Interfacing Raspberry Pi and MPU-6050

About Me



Andrew Birkett

View my complete profile

```

041     return myfile.split(" ")
042
043 def run():
044     pygame.init()
045     screen = pygame.display.set_mode(SCREEN_SIZE, HWSURFACE | OPENGL | DOUBLEBUF)
046     resize(*SCREEN_SIZE)
047     init()
048     clock = pygame.time.Clock()
049     cube = Cube((0.0, 0.0, 0.0), (.5, .5, .7))
050     angle = 0
051
052     while True:
053         then = pygame.time.get_ticks()
054         for event in pygame.event.get():
055             if event.type == QUIT:
056                 return
057             if event.type == KEYUP and event.key == K_ESCAPE:
058                 return
059
060         values = read_values()
061         x_angle = values[0]
062         y_angle = values[1]
063
064         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
065
066         glColor((1.,1.,1.))
067         glLineWidth(1)
068         glBegin(GL_LINES)
069
070         for x in range(-20, 22, 2):
071             glVertex3f(x/10., -1, -1)
072             glVertex3f(x/10., -1, 1)
073
074         for x in range(-20, 22, 2):
075             glVertex3f(x/10., -1, 1)
076             glVertex3f(x/10., 1, 1)
077
078         for z in range(-10, 12, 2):
079             glVertex3f(-2, -1, z/10.)
080             glVertex3f( 2, -1, z/10.)
081
082         for z in range(-10, 12, 2):
083             glVertex3f(-2, -1, z/10.)
084             glVertex3f(-2,  1, z/10.)
085
086         for z in range(-10, 12, 2):
087             glVertex3f( 2, -1, z/10.)
088             glVertex3f( 2,  1, z/10.)
089
090         for y in range(-10, 12, 2):
091             glVertex3f(-2, y/10., 1)
092             glVertex3f( 2, y/10., 1)
093
094         for y in range(-10, 12, 2):
095             glVertex3f(-2, y/10., 1)
096             glVertex3f(-2, y/10., -1)
097
098         for y in range(-10, 12, 2):
099             glVertex3f(2, y/10., 1)
100             glVertex3f(2, y/10., -1)
101
102         glEnd()
103         glPushMatrix()
104         glRotate(float(x_angle), 1, 0, 0)
105         glRotate(-float(y_angle), 0, 0, 1)
106         cube.render()
107         glPopMatrix()
108         pygame.display.flip()
109
110 class Cube(object):
111
112     def __init__(self, position, color):
113         self.position = position
114         self.color = color
115
116     # Cube information
117     num_faces = 6
118
119     vertices = [ (-1.0, -0.05, 0.5),
120                 (1.0, -0.05, 0.5),
121                 (1.0, 0.05, 0.5),
122                 (-1.0, 0.05, 0.5),
123                 (-1.0, -0.05, -0.5),
124                 (1.0, -0.05, -0.5),
125                 (1.0, 0.05, -0.5),
126                 (-1.0, 0.05, -0.5) ]
127
128     normals = [ (0.0, 0.0, +1.0), # front
129                (0.0, 0.0, -1.0), # back
130                (+1.0, 0.0, 0.0), # right
131                (-1.0, 0.0, 0.0), # left
132                (0.0, +1.0, 0.0), # top
133                (0.0, -1.0, 0.0) ] # bottom
134
135     vertex_indices = [ (0, 1, 2, 3), # front
136                       (4, 5, 6, 7), # back
137                       (1, 5, 6, 2), # right
138                       (0, 4, 7, 3), # left
139                       (3, 2, 6, 7), # top
140                       (0, 1, 5, 4) ] # bottom
141
142     def render(self):
143         then = pygame.time.get_ticks()
144         glColor(self.color)

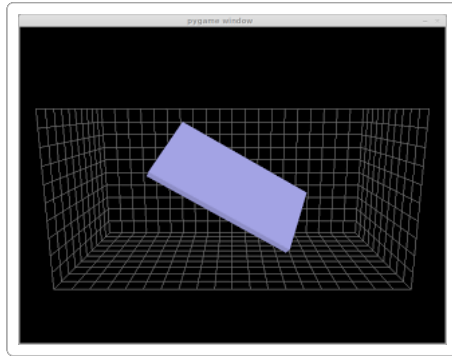
```

```

145
146     vertices = self.vertices
147
148     # Draw all 6 faces of the cube
149     glBegin(GL_QUADS)
150
151     for face_no in xrange(self.num_faces):
152         glNormal3dv(self.normals[face_no])
153         v1, v2, v3, v4 = self.vertex_indices[face_no]
154         glVertex(vertices[v1])
155         glVertex(vertices[v2])
156         glVertex(vertices[v3])
157         glVertex(vertices[v4])
158     glEnd()
159
160 if __name__ == "__main__":
161     run()

```

Remember to change the URL line **038** to your specific value (the address you used earlier to test the server). When you run it a window will open showing the orientation of the sensor, rotating the sensor will update the display.



You'll notice that when the sensor isn't being physically moved the noisy data is causing it to wobble. The [next blog post](#) shows how to reduce this.

Posted by [Andrew Birkett](#) at 17:09



Labels: [MPU-6050](#), [OpenGL](#), [Python](#), [Raspberry Pi](#), [Raspbian](#)

19 comments



Add a comment as Julien Louette

Top comments



Andrew Birkett via Google+ 2 years ago - Shared publicly

#RaspberryPi

+1 1 - Reply



Luis Emilio López Guillén 1 year ago

it works on windows, activepython 2.7 pygame 2.7 and pyopengl 3, gj andrew, thx for your great work and share with us.



Ash Anjelo Fernando 6 months ago - Shared publicly

Hey Andrew your code has been really helpful. But I'm having trouble writing the python code to change the sensitivity of the sensor. would you be able to help me out with that ??

1 - Reply



John Wood 1 year ago - Shared publicly

I've just set this up on my pi, using PCLinux OS for the level.py code, but I get a blank pygame screen... I know nothing of pygame, so not sure what to look for... any guesses?

+1 1 - Reply



Andrew Birkett 1 year ago

Do your Linux video driver support OpenGL as it will need this feature for the rendering.



John Wood 1 year ago

+**Andrew Birkett** possibly not! I'm not much of a linux GUI guy - more at home in the console! I'm

only using an old 32bit Dell at the moment, and PCLinux OS had the only support for my dual-monitor graphics card out of the box. I'll try to find better nVidia drivers and try again... Thanks!



Siddharth Gupta 1 year ago - Shared publicly

I can confirm this works on Mac. Thank you so much for the awesome tutorials!

+1 · Reply



Adward5477 2 years ago - Shared publicly

Great Work. but I have some problem

```
$sudo python level.py
```

Traceback (most recent call last):

File "level.py", line 161, in <module>

Read more

+1 · Reply

[View all 7 replies](#)



alex b 5 months ago

so that means that we have to install the code on our laptop, not on Pi?



Andrew Birkett 5 months ago

Yes that's right.



Andy 2 years ago

I have corrected a bug in the render code and the server code so the orientations are calculated correctly.



Andy 2 years ago

Glad it worked ok. I'd be interested to see any changed / improvements you make to the code.



Alex Withthescar 2 years ago

Excellent Work! Up and running on OSX with only few changes! Thankyou!



Alex Withthescar 2 years ago

Once again, Thank you Andy! Counting the hours until the day job finishes so i can go play Pi! :)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)