

EXPERIMENT 02 - GYROSCOPE AND ACCELEROMETER DEMO WITH PYTHON AND BLE DONGLE

Learn how to access the gyroscope and accelerometer from Angel Sensor with Python and Bluetooth Low Energy by Bluegiga.

Get the source code [here](#).

If you haven't yet, check out [experiment 01](#) to get more familiar with your Angel Sensor's heart rate functions, set up your environment for this tutorial, and get a better understanding of some of the basic capabilities of the device. We're going to use the [BLED112 dongle](#) and Python once again to communicate with the Angel Sensor, but this time we're aiming for the gyroscope and accelerometer. The purpose of today's post is to emulate this [video](#) and get you familiar with how to communicate with the gyro and accelerometer.

A bit about Gyroscopes and Accelerometers

The crew here at Angel Sensor is pretty excited to get a gyroscope and accelerometer on board the device! A gyroscope uses a rotor to determine orientation and is widely used in UAVs and boats for navigation. Accelerometers are able to determine acceleration of an object but are susceptible to the gravitational pull of the earth. This makes accelerometers not as useful in things such as aircrafts but are great for doing tasks such as determining when to flip a screen on a smartphone. Now that we have the basics, let's get into how we access the gyro on Angel Sensor in Python.

Gyroscopes, Accelerometers and Python, oh my!

This experiment adds a more visual element than experiment 01 by using pygame. If you look at the source code, 'cube.py' is our pygame driver that simulates a 3D cube to which we are going to send our gyroscope and accelerometer data. However, you'll find most of the code I'm going to be referring to in the 'mpu.py' file which interacts with the Angel Sensor. First thing we need to do is find our ID's for the accelerometer and gyroscope. With the latest firmware they can be found with the following UUIDs:

```
WAVEFORM_SERVICE_UUID = '481d178c10dd11e4b514b2227cce2b54'  
ACCEL_WAVEFORM_UUID = '4e92f4abc01b4b5ab328699856a7c2ee'  
GYRO_WAVEFORM_UUID = '5df14ec3fed1442883bf28ade00b0d98'
```

Just as with the heart rate notifications, the gyro and accelerometer are handled with events. Let's start with the gyroscope. The Angel Sensor will send gyroscope data as 6 bit serialized data, so we will need to parse that into our x, y and z values.

```
def handleGyroEvent(event, gfx):
    d = event.data
    while len(d) >= 6:
        x = Sint16().deserialize(d[0:2])
        y = Sint16().deserialize(d[2:4])
        z = Sint16().deserialize(d[4:6])
        del d[0:6]
        gfx.rotateX(convertGyro(x))
        gfx.rotateY(convertGyro(z))
        gfx.rotateZ(convertGyro(y))
```

The raw data might not mean too much to us so we also do a bit of scaling to get it into a nice pretty angular velocity (°/sec).

```
def convertGyro(gyro):
    '''Convert gyro output to angle change in degrees based on sampling
    rate and sensitivity'''
    sampling_rate = 100.0 # Hz
    sensitivity = 500.0 # deg/sec
    half_range = 2 << 15 # 16-bit signed value return gyro * sensitivity /
    half_range / sampling_rate
```

Accelerometers are a little bit simpler with only 3 bit serialized data. For this experiment we have just consolidated the x, y and z values into one waveform but you can easily switch that back to individual values with a bit of tinkering.

```
def handleAccelEvent(event, gfx): d = event.data while len(d) >= 3:
    mag = d[0] + (d[1] << 8) + (d[2] << 16)
    del d[0:3]
    gfx.addAccel(mag)
```

Hopefully that will get you a good start into understanding how to access the gyroscope and accelerometer in your own projects.

Setting up your environment

Now that we have the code out of the way, let's get to the good stuff! You hopefully have your environment set up from [experiment 01](#) already, so I'll just list off what you need here: [Python](#), [PySerial](#) and [Pygame](#). Also make sure you have updated Angel Sensor to the latest firmware or you'll likely run into some errors during execution. From here, just plug in your BLE112 dongle you should be good to start.

Running the program

You should already have the source code (download link) if not you'll need to grab that and unpack it. Now open up a command prompt and navigate to the '02-Python-Gyro-3D' directory. Now press the button on Angel sensor and execute this command:

```
python mpu.py -a
```

...and voila! Your pygame module should have popped up to show you real time gyroscope data as the cube and accelerometer waveform beneath it. Play around with it and get used to how it moves. We're excited to see what you can come up with using the gyroscope and accelerometer. Happy coding!

*Special thanks to **Leonel Machava** for his original post "[Rotating 3D Cube using Python and Pygame](#)". Leonel's source code used under [MIT license](#).*