Overview

9-axis absolute orientation sensor
3-axis accelerometer
3-axis gyroscope
3-axis magnetometer
Capable of providing sensor fused data for euler angles and quaternions

Libraries

1. <Adafruit_Sensor.h>
2. <Adafruit_BNO055.h>
3. <utility/imumaths.h>

Library 3 should be defaultly installed with arduino.
To install Library 2, go to Tools → Manage Libraries.
Search "Adafruit BNO055". Click instal.
This should prompt you to install the other dependencies.

Example Code

More information at the following link:
learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/arduino-code

Additionally, video tutorials available from Top Tech Boy
toptechboy.com/arduino-based-9-axis_inertial-measurement-unit-imu-based-on-bno055-sensor/

- Instantiation
  `Adafruit_BNO055 myIMU = Adafruit_BNO055();`
- In set up
  `myIMU.begin();`
  `myIMU.setExternalCrystalUse(true);`

14 | BNO055 Testing

05/23/23
Wesley Cecka

- Getting a vector with 3 components.

```
imu::Vector<3> acc = myIMU.getVector(Adafruit_BNO055::VECTOR_ACCELEROMETER);
```

1 2 3 4

1. Data type of the variable. This is similar to int or float.
2. Variable Name. This name could be anything you chose. Make it descriptive.
3. This is the method to call.
4. This is the specific Vector that you want.

Available Vectors

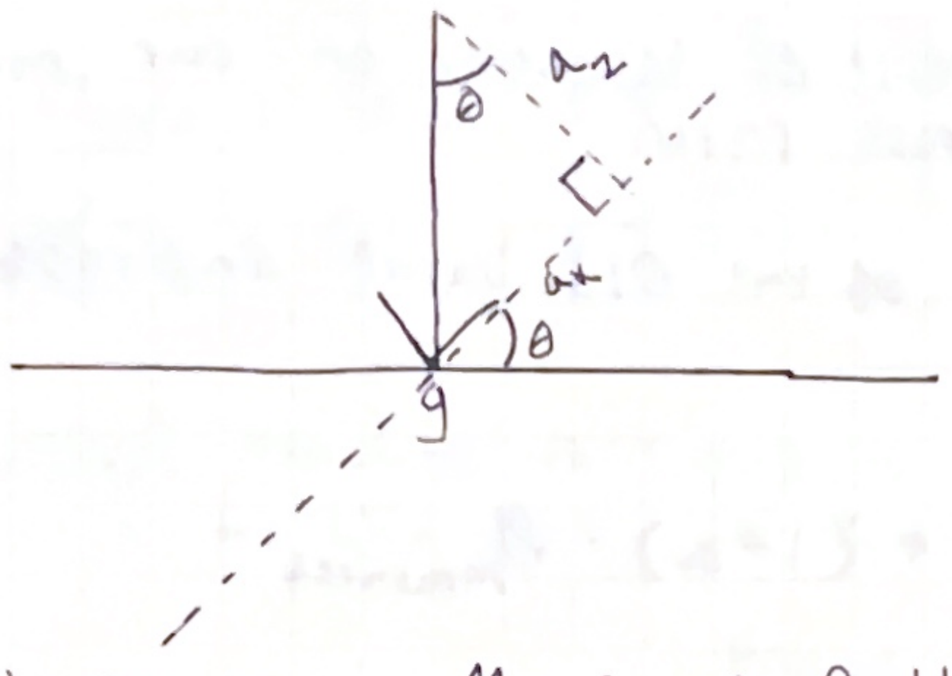
```
Adafruit_BNO055::VECTOR_MAGNETOMETER
Adafruit_BNO055::VECTOR_GYROSCOPE
Adafruit_BNO055::VECTOR_ACCELEROMETER
Adafruit_BNO055::VECTOR_EULER
Adafruit_BNO055::VECTOR_LINEARACCEL
Adafruit_BNO055::VECTOR_GRAVITY
```

Quaternion Data

```
imu::Quaternion quat = myIMU.getQuat();
```

Vectors have .x(), .y(), and .z() methods.
Quaternions have .x(), .y(), .z(), and .w() methods.

Measuring Pitch and Roll from Accelerometer



Measured Pitch

$$\theta_M = \tan^{-1}\left(\frac{a_x}{a_z}\right)$$

Measured Roll

$$\phi_M = \tan^{-1}\left(\frac{a_y}{a_z}\right)$$

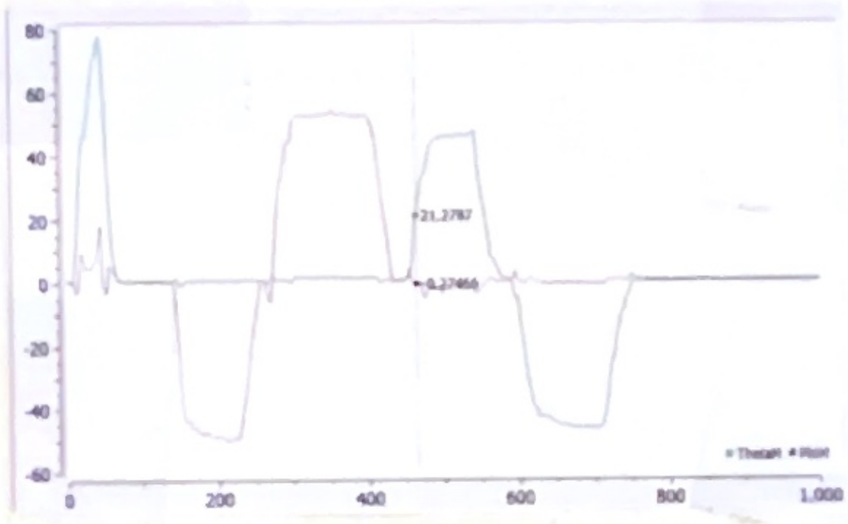


Figure 1

Testing our pitch and roll measurements. This is acomplished by rotating the bread board about the x any y axies. The measurements are good based on eyeball judgement.

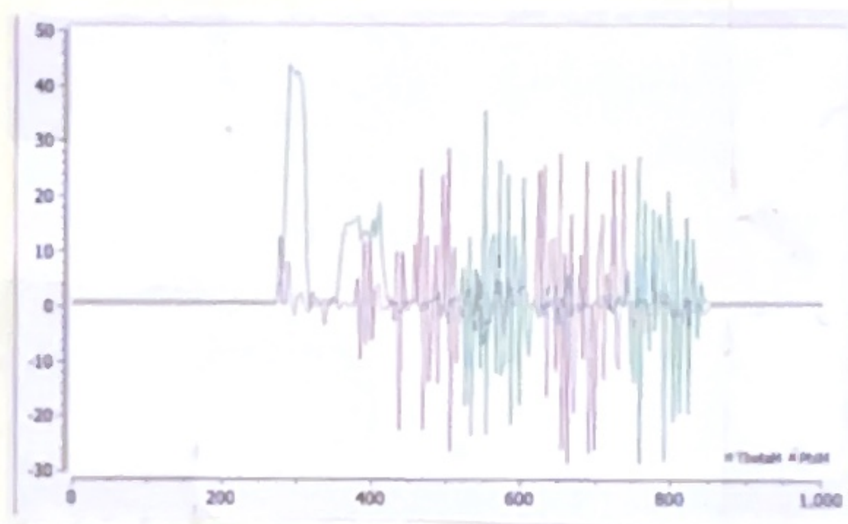


Figure 2

Testing our pitch and roll measurements when vibration is introduced. Notice that we didn't move the pitch and roll from 0. Howevere, there were still changes due to the vibration.

Low Pass Filter on Pitch and roll

To combat the effects of Vibration on our measurements, we can use a low pass filter.

We will take 90% of the Old Value and 10% of the New.

$a = 0.9$

$$\theta_{Filtered\ New} = a \cdot \theta_{Filtered\ old} + (1-a) \cdot \theta_{Measured}$$

$$\phi_{Filtered\ New} = a \cdot \phi_{Filtered\ old} + (1-a) \cdot \phi_{measured}$$

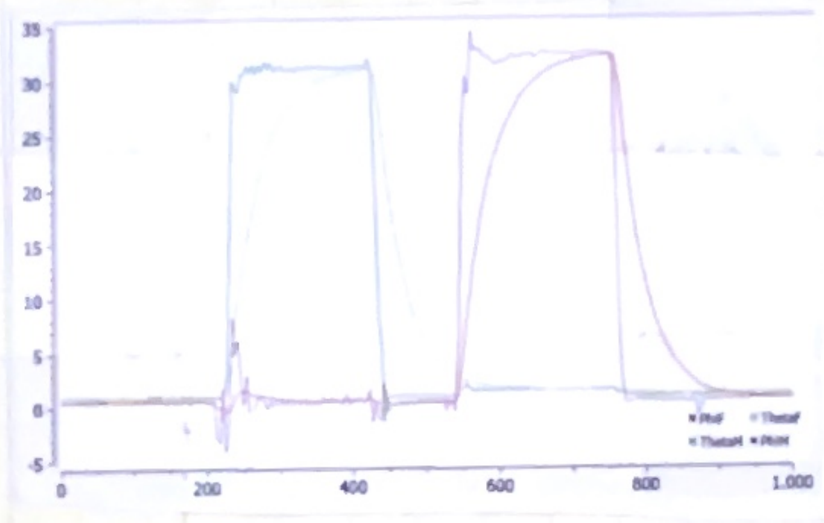


Figure 3

Showing the effects of the low pass filter on our measurements.

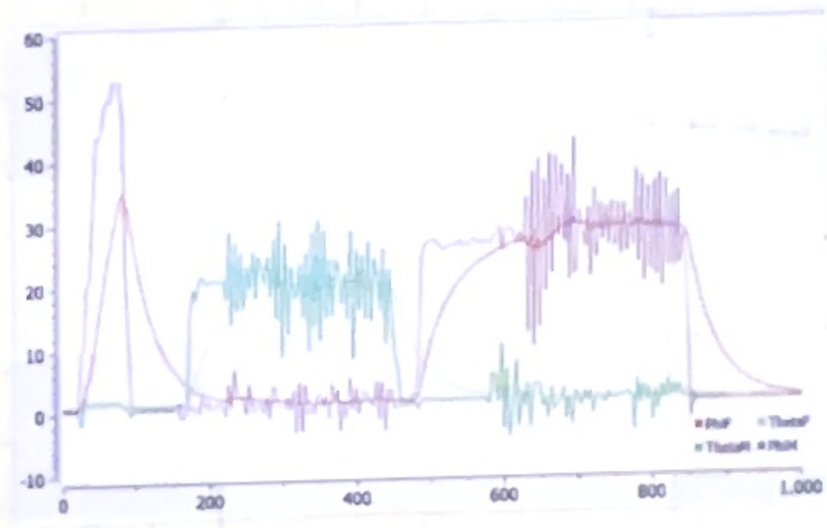


Figure 4

Showing vibration is not as bad on the filtered values Compared to the ~~accel~~ measured.

Measuring pitch and Roll from the gyroscope

$$\theta_{gyro} = \theta_{gyro} + \omega_y \cdot dt$$

$$\phi_{gyro} = \phi_{gyro} + \omega_x \cdot dt$$

The gyroscope measures the angular velocity. If we keep up with the time delta, we can measure pitch and Roll.

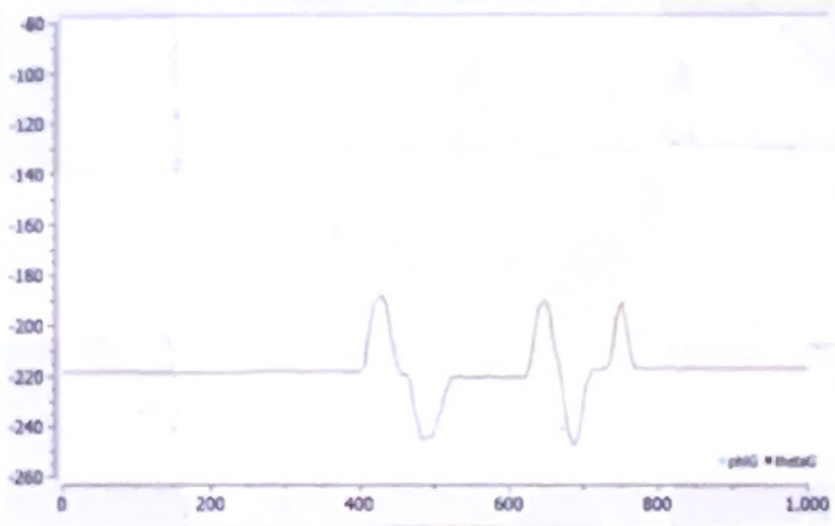


Figure 5

Testing the measurements from the gyroscope. Notice that the data in the short term, is good, but overtime the readings began to drift further from the actual value.

Combining Accelerometer and Gyroscope data using a Complimentary Filter

$$\theta_{system} = [\theta_{system} + \omega_y \cdot dt] \cdot a + \theta_{measured} \cdot [1 - a]$$

$$\phi_{system} = [\phi_{system} + \omega_x \cdot dt] \cdot a + \phi_{measured} \cdot [1 - a]$$

If ω is changing, we will use more of the data from it and less of the data from the accelerometer. If ω is near 0, our system will approach the value from the accelerometer.

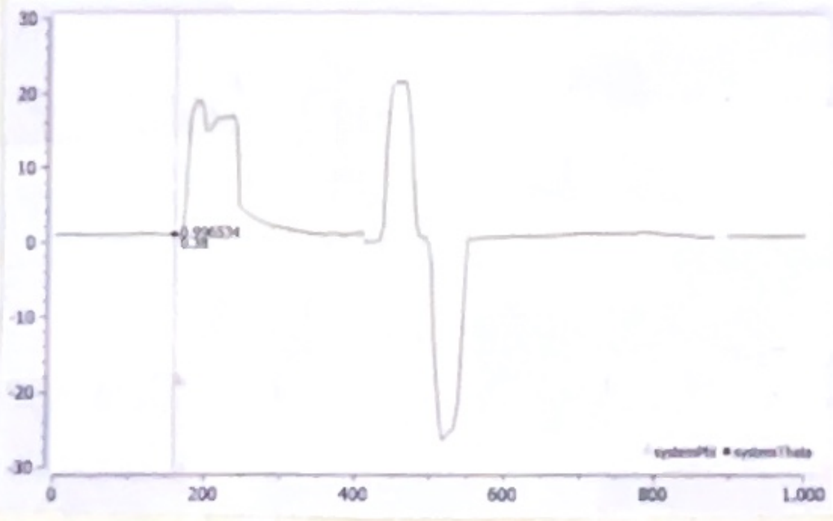


Figure 6

Testing the complimentary filter. Notice that the value changes realy quick, but when left still, it will approach a more stable value.

BNo055 Testing

05/23/23
Wesley Cooke

Basic Yaw Measurement

Assuming we stay in the XY plane, we can estimate our heading using $\tan^{-1}(M_y/M_x)$ where M_y and M_x are the y and x readings from the magnetometer. However, this stops working once you add Pitch and Roll. The derivation of the tilt compensated equation is left as an exercise for the reader.

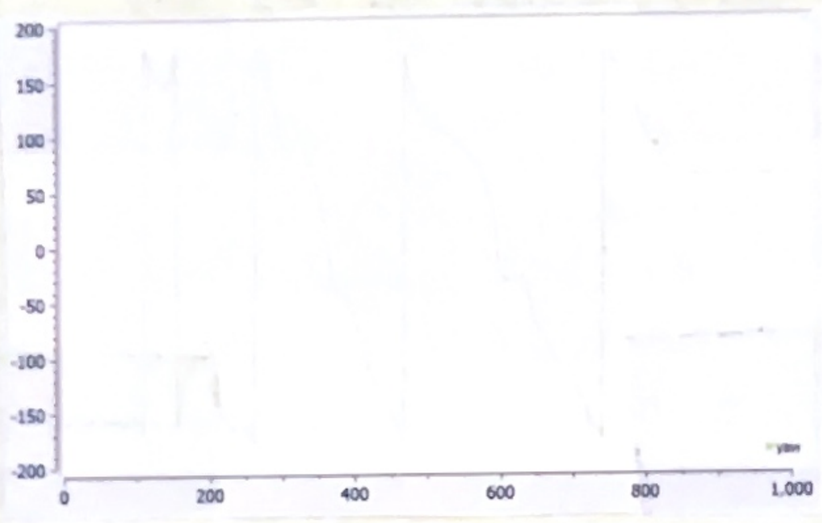


Figure 7
Testing the basic yaw measurement.

Comparing Complimentary Filter to onboard Euler and Quaternion data

The previous exercises are important to help build your intuition about how euler angles work. However, this sensor has registers that are keeping track of these values already.

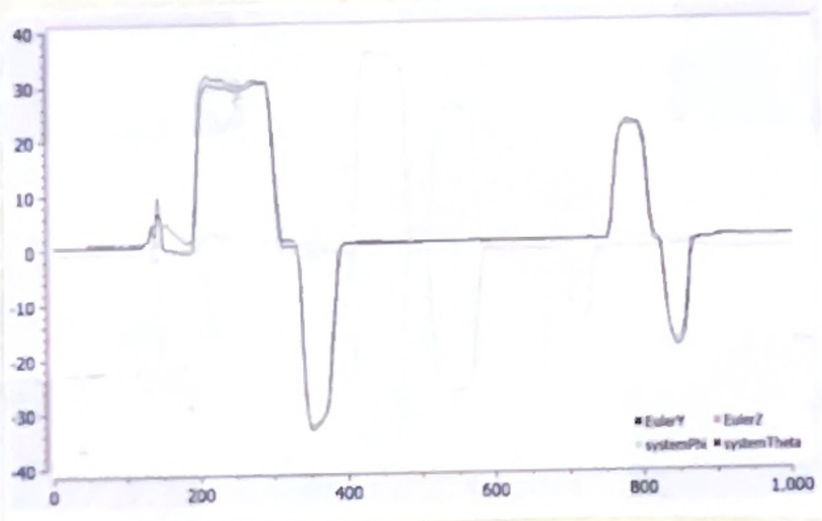
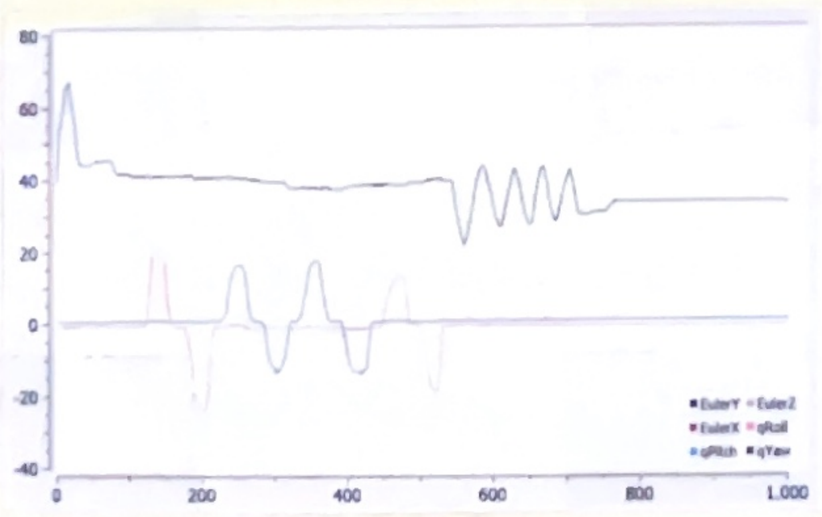


Figure 8
Comparing the onboard Euler angles to our filtered angles. Aside from having an axis flipped, we notice that the values are very similar.



~~The yaw is not shown because it assumed a different forward direction.~~ → See Page 21

Figure 9
If we read the quaternion data and convert them to euler angles, we notice they almost exclusively overlap the eular angles of the board.

Code

Not all ~~pea~~ pieces of the code are shown for spacial reasons. Most is available on top tech boy's website.

```
62 void setup()
63 {
64   Serial.begin(115200);
65   offerEncouragement();
66
67   myIMU.begin();
68   delay(1000);
69
70   // Tell the sensor to make a temperature measurement.
71   // Some measurements may be based on this
72   int8_t temp = myIMU.getTemp();
73   Serial.print("Temp: ");
74   Serial.println(temp);
75
76   // Don't use the crystal on the chip itself.
77   // Instead use the crystal on the board. -- More accurate
78   myIMU.setExtCrystalUse(true);
79
80   millisOld = millis();
81 }
```

```
Void offerEncouragement()
{
  Serial.println("Stand by to Get Some!");
  Serial.println("#GOPhysics");
  Serial.println("Water is Life");
  Serial.println("--- --- ---------");
}
```

BNO055 Testing

05/23/23
Wesley Costa

```
83 void loop()
84 {
85
86   // Calibration Qualities. Measured on a scale of 0-3.
87   uint8_t system, gyro, accel, mg = 0;
88   myIMU.getCalibration(&system, &gyro, &accel, &mg);
89
90   // Using the unified library
91   // Go the IMU and bring me a vector with 3 components
92   imu::Vector<3> acc = myIMU.getVector(Adafruit_BNO055::VECTOR_ACCELEROMETER);
93   imu::Vector<3> gyr = myIMU.getVector(Adafruit_BNO055::VECTOR_GYROSCOPE);
94   imu::Vector<3> mag = myIMU.getVector(Adafruit_BNO055::VECTOR_MAGNETOMETER);
95   imu::Vector<3> euler = myIMU.getVector(Adafruit_BNO055::VECTOR_EULER);
96   imu::Quaternion quat = myIMU.getQuat();
97
98   // Measured values from the accelerometer
99   thetaM = atan2(acc.x()/9.8, acc.z()/9.8)/2/3.141592654*360; // Pitch
100  phiM = atan2(acc.y()/9.8, acc.z()/9.8)/2/3.141592654*360;   // Roll
```

```
102  // Calculate the new filtered values
103  phiFnew = a * phiFold + (1-a) * phiM;
104  thetaFnew = a * thetaFold + (1-a) * thetaM;
105
106  // Get the current dt, and remember the old millis.
107  dt = (millis() - millisOld)/1000.0;
108  millisOld = millis();
109
110  // System theta calculation using a complimentary filter
111  theta = (theta-(gyr.y()*dt))*a + thetaM*(1-a); // Pitch
112  phi = (phi+(gyr.x()*dt))*a + phiM*(1-a);       // Roll
113
114  // theta = thetaold + omega * dt. Omega comes from gryo.
115  thetaG = thetaG-gyr.y()*dt;
116  phiG = phiG+gyr.x()*dt;
117
118  phiRad = phi/360*(2*3.14);
119  thetaRad = theta/360*(2*3.14);
120
```

```
190   thetaFold = thetaFnew;
191   phiFold = phiFnew;
192   delay(BNO055_SAMPLERATE_DELAY_MS);
```

```
// Calculate Euler angles from quaternions.
// using https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles
double qRoll = atan2(2*(quat.w()*quat.x() +  quat.y()*quat.z()), 1-2*(quat.x()*quat.x()+quat.y()*quat.y()));
double qPitch = -3.14/2 + 2*atan2(sqrt(1+2*(quat.w()*quat.y() - quat.x()*quat.z())), sqrt(1-2*(quat.w()*quat.y() - quat.x()*quat.z())));
double qYaw = atan2(2*(quat.w()*quat.z()+quat.x()*quat.y()), 1-2*(quat.y()*quat.y() + quat.z()*quat.z()));
```

Relative vs. Absolute orientation

The BNO is capable of providing a yaw reading relative to where the sensor turned on at, or a yaw reading relative to the earths magnetic field. (ie absolute).

To enable the Absolute readings, the BNO needs to be set to the compass mode. More Modes and details are on page 22 and 23 of the datasheet.
https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/~~bno~~ bst-bno055-ds000.pdf

```
11 void setup()
12 {
13   Serial.begin(115200);
14   myIMU.begin();
15   delay(1000);
16   myIMU.setMode(adafruit_bno055_opmode_t::OPERATION_MODE_COMPASS);
17   myIMU.setExtCrystalUse(true);
18   Serial.println("BNO Connected!");
19 }
20
21 void loop()
22 {
23   euler = myIMU.getVector(Adafruit_BNO055::VECTOR_EULER);
24   Serial.println(euler.x(), 2);
25   delay(500);
26 }
```

The set mode method allows us to set the BNO to the compass mode. If this isn't here, the yaw we recieve from the BNO is relative to the position it powered up in.