```
// POST Gyroscope test code
// Test to see if Gyroscope is working

// Needed Libraries
   #include <Wire.h>
    #include"gyro.h"

// Global variables
// variables for Gryo X,Y,Z axes
   int gyX;
   int gyY;
   int gyZ;

void setup(){
// Start Serial communication.
   Serial.begin(9600);
   Serial.println("POST Gyro code started...");

// Start I2C Communication
   Serial.println("START WIRE");
   Wire.begin();

// Start Gyro
   Serial.println("Start Gyro");
// Starts the gyro
   Gyro::setupGyroITG();
// Prints addres of gyro
   Serial.println(Gyro::itgRead(Gyro::itgAddress,
}
```

```
void loop(){

// Read Gyro
// The values after the Gyro::read_() command are
// Calculate the initialization values by placing
// surface and not moving or shaking it. Those va
// here in the code and must be a whole number
    gyX = Gyro::readX() +16/* + INSERT INITIALIZA
    gyY = Gyro::readY() -49/* + INSERT INITIALIZA
    gyZ = Gyro::readZ() +1/* + INSERT INITIALIZAT

// Print the output rates to the terminal, sepera
// Right now we do not have a way to calibrate th
// shows that tilting the board results in a chan
   Serial.print(gyX);
   Serial.print('\t');
   Serial.print(gyY);
   Serial.print('\t');
   Serial.println(gyZ);

delay(1);
}
```