**SHREYA KUMARI**

[**Krshreya2402@gmail.com**](mailto:Krshreya2402@gmail.com)

**IoT June-July batch**

**OUTSTANDING PROJECT-3**

**3.Sensor Fusion Using Accelerometer, Gyroscope:**

**Calibrate and Use Sensor Fusion to Convert the Raw data coming from your MPU6050 Sensor and make sure that it is applicable to resist the vibrations on the sensor.**

**Components:**

* One Arduinocompatible board
* One MPU6050 Acceleration Gyroscope Sensor Module
* jumper wires

**Connections:**

* Connect **5V VCC Power** (**Red wire**), **Ground** (**Black wire**), **SCL** (**Yellow wire**), and **SDA** (**Green wire**) to the **MPU6050 Module**
* Connect the other end of the **Ground wire** (**Black wire**) to a **Ground** pin of the **Arduino** board
* Connect the other end of the **5V VCC Power wire** (**Red wire**) to the **5V** power pin of the **Arduino** board
* Connect the other end of the **SDA wire** (**Green wire**) to **SDA/Analog pin 4** of the **Arduino** board
* Connect the other end of the **SCL wire** (**Yellow wire**) to **SCL/Analog pin 5** of the **Arduino** board
* **Picture 4** shows where are the **Ground**, **5V** Power, **SDA/Analog pin 4**, and **SCL/Analog pin 5**, pins of the **Arduino**

**Code:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | #include "MPU9250.h" | |  |  |  | | --- | | #include "Streaming.h" // needed for the Serial output | |  |  |  | | --- | | #include "SensorFusion.h" | |  |  |  | | --- | | SF fusion; | |  |  |  | | --- | |  | |  |  |  | | --- | | float gx, gy, gz, ax, ay, az, mx, my, mz, temp; | |  |  |  | | --- | | float pitch, roll, yaw; | |  |  |  | | --- | | float deltat; | |  |  |  | | --- | |  | |  |  |  | | --- | | #define SS\_PIN PB12 | |  |  |  | | --- | | SPIClass mySPI (2); | |  |  |  | | --- | | MPU9250 IMU(mySPI, SS\_PIN); | |  |  |  | | --- | | int status; | |  |  |  | | --- | |  | |  |  |  | | --- | | #define EULER\_DATA | |  |  |  | | --- | | //#define RAW\_DATA | |  |  |  | | --- | | //#define PROCESSING | |  |  |  | | --- | | //#define SERIAL\_PLOTER | |  |  |  | | --- | |  | |  |  |  | | --- | |  | |  |  |  | | --- | | void setup()  { | |  |  |  | | --- | | // serial to display data | |  |  |  | | --- | | Serial.begin(115200); | |  |  |  | | --- | | while (!Serial)  {} | |  |  |  | | --- | |  | |  |  |  | | --- | | // start communication with IMU | |  |  |  | | --- | | status = IMU.begin(); | |  |  |  | | --- | | if (status < 0)  { | |  |  |  | | --- | | Serial.println("IMU initialization unsuccessful"); | |  |  |  | | --- | | Serial.println("Check IMU wiring or try cycling power"); | |  |  |  | | --- | | Serial.print("Status: "); | |  |  |  | | --- | | Serial.println(status); | |  |  |  | | --- | | while (1)  {} | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | void loop()  { | |  |  |  | | --- | |  | |  |  |  | | --- | | IMU.readSensor(); | |  |  |  | | --- | |  | |  |  |  | | --- | | ax = IMU.getAccelX\_mss(); | |  |  |  | | --- | | ay = IMU.getAccelY\_mss(); | |  |  |  | | --- | | az = IMU.getAccelZ\_mss(); | |  |  |  | | --- | | gx = IMU.getGyroX\_rads(); | |  |  |  | | --- | | gy = IMU.getGyroY\_rads(); | |  |  |  | | --- | | gz = IMU.getGyroZ\_rads(); | |  |  |  | | --- | | mx = IMU.getMagX\_uT(); | |  |  |  | | --- | | my = IMU.getMagY\_uT(); | |  |  |  | | --- | | mz = IMU.getMagZ\_uT(); | |  |  |  | | --- | | temp = IMU.getTemperature\_C(); | |  |  |  | | --- | |  | |  |  |  | | --- | | #ifdef RAW\_DATA | |  |  |  | | --- | | Serial << "From last Update:\t"; Serial.println(deltat, 6); | |  |  |  | | --- | | Serial << "GYRO:\tx:" << gx << "\t\ty:" << gy << "\t\tz:" << gz << newl; | |  |  |  | | --- | | Serial << "ACC:\tx:" << ax << "\t\ty:" << ay << "\t\tz:" << az << newl; | |  |  |  | | --- | | Serial << "MAG:\tx:" << mx << "\t\ty:" << my << "\t\tz:" << mz << newl; | |  |  |  | | --- | | Serial << "TEMP:\t" << temp << newl << newl; | |  |  |  | | --- | | #endif | |  |  |  | | --- | |  | |  |  |  | | --- | | deltat = fusion.deltatUpdate(); | |  |  |  | | --- | | //fusion.MahonyUpdate(gx, gy, gz, ax, ay, az, mx, my, mz, deltat); //mahony is suggested if there isn't the mag | |  |  |  | | --- | | fusion.MadgwickUpdate(gx, gy, gz, ax, ay, az, mx, my, mz, deltat); //else use the magwick | |  |  |  | | --- | |  | |  |  |  | | --- | | roll = fusion.getRoll(); | |  |  |  | | --- | | pitch = fusion.getPitch(); | |  |  |  | | --- | | yaw = fusion.getYaw(); | |  |  |  | | --- | |  | |  |  |  | | --- | | #ifdef EULER\_DATA | |  |  |  | | --- | | Serial << "Pitch:\t" << pitch << "\t\tRoll:\t" << roll << "\t\tYaw:\t" << yaw << newl << newl; | |  |  |  | | --- | | #endif | |  |  |  | | --- | |  | |  |  |  | | --- | | #ifdef PROCESSING | |  |  |  | | --- | | roll = fusion.getRollRadians(); | |  |  |  | | --- | | pitch = fusion.getPitchRadians(); | |  |  |  | | --- | | yaw = fusion.getYawRadians(); | |  |  |  | | --- | | Serial << pitch << ":" << roll << ":" << yaw << newl; | |  |  |  | | --- | | #endif | |  |  |  | | --- | |  | |  |  |  | | --- | | #ifdef SERIAL\_PLOTER | |  |  |  | | --- | | Serial << pitch << " " << roll << " " << yaw << endl; | |  |  |  | | --- | | #endif | |  |  |  | | --- | |  | |  |  |  | | --- | | //delay(200); //for readability | |  |  |  | | --- | |  | |  |   } |

**Sample code:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | |  |  |  | | --- | | #include "SensorFusion.h" //SF | |  |  |  | | --- | | SF fusion; | |  |  |  | | --- | |  | |  |  |  | | --- | | float gx, gy, gz, ax, ay, az, mx, my, mz; | |  |  |  | | --- | | float pitch, roll, yaw; | |  |  |  | | --- | | float deltat; | |  |  |  | | --- | |  | |  |  |  | | --- | |  | |  |  |  | | --- | | void setup()  { | |  |  |  | | --- | |  | |  |  |  | | --- | | Serial.begin(115200); //serial to display data | |  |  |  | | --- | | // your IMU begin code goes here | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | void loop()  { | |  |  |  | | --- | |  | |  |  |  | | --- | | // now you should read the gyroscope, accelerometer (and magnetometer if you have it also) | |  |  |  | | --- | | // NOTE: the gyroscope data have to be in radians | |  |  |  | | --- | | // if you have them in degree convert them with: DEG\_TO\_RAD example: gx \* DEG\_TO\_RAD | |  |  |  | | --- | |  | |  |  |  | | --- | | deltat = fusion.deltatUpdate(); //this have to be done before calling the fusion update | |  |  |  | | --- | | //choose only one of these two: | |  |  |  | | --- | | fusion.MahonyUpdate(gx, gy, gz, ax, ay, az, deltat); //mahony is suggested if there isn't the mag and the mcu is slow | |  |  |  | | --- | | //fusion.MadgwickUpdate(gx, gy, gz, ax, ay, az, mx, my, mz, deltat); //else use the magwick, it is slower but more accurate | |  |  |  | | --- | |  | |  |  |  | | --- | | pitch = fusion.getPitch(); | |  |  |  | | --- | | roll = fusion.getRoll(); //you could also use getRollRadians() ecc | |  |  |  | | --- | | yaw = fusion.getYaw(); | |  |  |  | | --- | |  | |  |  |  | | --- | | Serial.print("Pitch:\t"); Serial.println(pitch); | |  |  |  | | --- | | Serial.print("Roll:\t"); Serial.println(roll); | |  |  |  | | --- | | Serial.print("Yaw:\t"); Serial.println(yaw); | |  |  |  | | --- | | Serial.println(); | |  |   } |