# Code for Arduino IDE

#include <Wire.h>

#include <MadgwickAHRS.h>

Madgwick filter;

unsigned long microsPerReading, microsPrevious;

float accelScale, gyroScale;

//initializations

float accAngleX, accAngleY, gyroAngleX, gyroAngleY, gyroAngleZ;

float roll, pitch, yaw;

float AccErrorX, AccErrorY, GyroErrorX, GyroErrorY, GyroErrorZ;

float elapsedTime, currentTime, previousTime;

int c = 0;

// MPU9250 Slave Device Address

const uint8\_t MPU9250SlaveAddress = 0x68;

// Pins for serial data

const uint8\_t scl = D6;

const uint8\_t sda = D7;

// sensitivity scale factor of accelerometer and gyroscope

const uint16\_t AccelScaleFactor = 16384;

const uint16\_t GyroScaleFactor = 131;

// MPU9250 few configuration register addresses

const uint8\_t MPU9250\_REGISTER\_SMPLRT\_DIV = 0x19;

const uint8\_t MPU9250\_REGISTER\_USER\_CTRL = 0x6A;

const uint8\_t MPU9250\_REGISTER\_PWR\_MGMT\_1 = 0x6B;

const uint8\_t MPU9250\_REGISTER\_PWR\_MGMT\_2 = 0x6C;

const uint8\_t MPU9250\_REGISTER\_CONFIG = 0x1A;

const uint8\_t MPU9250\_REGISTER\_GYRO\_CONFIG = 0x1B;

const uint8\_t MPU9250\_REGISTER\_ACCEL\_CONFIG = 0x1C;

const uint8\_t MPU9250\_REGISTER\_FIFO\_EN = 0x23;

const uint8\_t MPU9250\_REGISTER\_INT\_ENABLE = 0x38;

const uint8\_t MPU9250\_REGISTER\_ACCEL\_XOUT\_H = 0x3B;

const uint8\_t MPU9250\_REGISTER\_SIGNAL\_PATH\_RESET = 0x68;

int16\_t AccelX, AccelY, AccelZ, Temperature, GyroX, GyroY, GyroZ;

void setup() {

Serial.begin(9600);

Wire.begin(sda, scl);

MPU9250\_Init();

filter.begin(25);

// initialize variables to pace updates to correct rate

microsPerReading = 1000000 / 25;

microsPrevious = micros();

}

void I2C\_Write(uint8\_t deviceAddress, uint8\_t regAddress, uint8\_t data){

Wire.beginTransmission(deviceAddress);

Wire.write(regAddress);

Wire.write(data);

Wire.endTransmission();

}

// read all 14 register

void Read\_RawValue(uint8\_t deviceAddress, uint8\_t regAddress){

Wire.beginTransmission(deviceAddress);

Wire.write(regAddress);

Wire.endTransmission();

Wire.requestFrom(deviceAddress, (uint8\_t)14);

AccelX = (((int16\_t)Wire.read()<<8) | Wire.read());

AccelY = (((int16\_t)Wire.read()<<8) | Wire.read());

AccelZ = (((int16\_t)Wire.read()<<8) | Wire.read());

Temperature = (((int16\_t)Wire.read()<<8) | Wire.read());

GyroX = (((int16\_t)Wire.read()<<8) | Wire.read());

GyroY = (((int16\_t)Wire.read()<<8) | Wire.read());

GyroZ = (((int16\_t)Wire.read()<<8) | Wire.read());

}

//configure MPU9250

void MPU9250\_Init(){

delay(150);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_SMPLRT\_DIV, 0x07);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_PWR\_MGMT\_1, 0x01);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_PWR\_MGMT\_2, 0x00);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_CONFIG, 0x00);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_GYRO\_CONFIG, 0x00);//set +/-250 degree/second full scale

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_ACCEL\_CONFIG, 0x00);// set +/- 2g full scale

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_FIFO\_EN, 0x00);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_INT\_ENABLE, 0x01);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_SIGNAL\_PATH\_RESET, 0x00);

I2C\_Write(MPU9250SlaveAddress, MPU9250\_REGISTER\_USER\_CTRL, 0x00);

}

void loop() {

float Ax, Ay, Az, T, Gx, Gy, Gz;

unsigned long microsNow;

Read\_RawValue(MPU9250SlaveAddress, MPU9250\_REGISTER\_ACCEL\_XOUT\_H);

//divide each with their sensitivity scale factor

Ax = (float)AccelX/AccelScaleFactor;

Ay = (float)AccelY/AccelScaleFactor;

Az = (float)AccelZ/AccelScaleFactor;

T = (float)Temperature/340+36.53; //temperature formula

Gx = (float)GyroX/GyroScaleFactor;

Gy = (float)GyroY/GyroScaleFactor;

Gz = (float)GyroZ/GyroScaleFactor;

// check if it's time to read data and update the filter

microsNow = micros();

if (microsNow - microsPrevious >= microsPerReading) {

// update the filter, which computes orientation

filter.updateIMU(Gx, Gy, Gz, Ax, Ay, Az);

// print the heading, pitch and roll

roll = filter.getRoll();

pitch = filter.getPitch();

yaw = filter.getYaw();

Serial.print(roll);

Serial.print("/");

Serial.print(pitch);

Serial.print("/");

Serial.println(yaw);

// increment previous time, so we keep proper pace

microsPrevious = microsPrevious + microsPerReading;

// Print the values on the serial monitor

}

/\*

Serial.print("Ax: "); Serial.print(Ax);

Serial.print(" Ay: "); Serial.print(Ay);

Serial.print(" Az: "); Serial.print(Az);

Serial.print(" T: "); Serial.print(T);

Serial.print(" Gx: "); Serial.print(Gx);

Serial.print(" Gy: "); Serial.print(Gy);

Serial.print(" Gz: "); Serial.println(Gz);

\*/

//calculating error

/\*

while (c<200){

float AccErrorX, AccErrorY, GyroErrorX, GyroErrorY, GyroErrorZ;

AccErrorX = AccErrorX + ((atan((Ay) / sqrt(pow((Ax), 2) + pow((Az), 2))) \* 180 / PI));

AccErrorY = AccErrorY + ((atan(-1 \* (Ax) / sqrt(pow((Ay), 2) + pow((Az), 2))) \* 180 / PI));

c++;

}

AccErrorX = AccErrorX / 200;

AccErrorY = AccErrorY / 200;

c = 0;

// Read gyro values 200 times

while (c < 200) {

// Sum all readings

GyroErrorX = GyroErrorX + (Gx);

GyroErrorY = GyroErrorY + (Gy);

GyroErrorZ = GyroErrorZ + (Gz);

c++;

}

//Divide the sum by 200 to get the error value

GyroErrorX = GyroErrorX / 200;

GyroErrorY = GyroErrorY / 200;

GyroErrorZ = GyroErrorZ / 200;

// Print the error values on the Serial Monitor

Serial.print("AccErrorX: ");

Serial.println(AccErrorX);

Serial.print("AccErrorY: ");

Serial.println(AccErrorY);

Serial.print("GyroErrorX: ");

Serial.println(GyroErrorX);

Serial.print("GyroErrorY: ");

Serial.println(GyroErrorY);

Serial.print("GyroErrorZ: ");

Serial.println(GyroErrorZ);

\*/

delay(100);

}

# Code for Processing

import processing.serial.\*;

import java.awt.event.KeyEvent;

import java.io.IOException;

Serial myPort;

String data="";

float roll, pitch,yaw;

void setup() {

size (1366 , 768, P3D);

myPort = new Serial(this, "COM6", 9600); // starts the serial communication

myPort.bufferUntil('\n');

}

void draw() {

translate(width/2, height/2, 0);

background(233);

textSize(22);

text("Roll: " + int(roll) + " Pitch: " + int(pitch), -100, 265);

// Rotate the object

rotateX(radians(-pitch));

rotateZ(radians(-roll));

rotateY(radians(yaw));

// 3D 0bject

textSize(30);

fill(0, 76, 153);

box (386, 40, 200); // Draw box

textSize(25);

fill(255, 255, 255);

text("Shaoor", -183, 10, 101);

//delay(10);

//println("ypr:\t" + angleX + "\t" + angleY); // Print the values to check whether we are getting proper values

}

// Read data from the Serial Port

void serialEvent (Serial myPort) {

// reads the data from the Serial Port up to the character '.' and puts it into the String variable "data".

data = myPort.readStringUntil('\n');

// if you got any bytes other than the linefeed:

if (data != null) {

data = trim(data);

// split the string at "/"

String items[] = split(data, '/');

if (items.length > 1) {

//--- Roll,Pitch in degrees

roll = float(items[0]);

pitch = float(items[1]);

yaw = float(items[2]);

}

}

}