**CODING**

#include <Wire.h>

#include <SoftwareSerial.h>

#include <TinyGPS++.h>

#include "MPU6050\_6Axis\_MotionApps20.h"

SoftwareSerial mySerial(10, 11);

SoftwareSerial mySerialGps(9, 10);

static const int RXPin = 4, TXPin = 3;

static const uint32\_t GPSBaud = 9600;

// The TinyGPS++ object

TinyGPSPlus gps;

// The serial connection to the GPS device

SoftwareSerial ss(RXPin, TXPin);

#define default\_co\_ordinates "13.0837722,77.4820204"

#define buzzer 6

#define button 11

#define command 8

String data, url = "";

int count = 1;

static boolean flag = true;

MPU6050 mpu;

// MPU6050 Slave Device Address

const uint8\_t MPU6050SlaveAddress = 0x68;

// Select SDA and SCL pins for I2C communication

const uint8\_t scl = A5;

const uint8\_t sda = A4;

// sensitivity scale factor respective to full scale setting provided in datasheet

const uint16\_t AccelScaleFactor = 16384;

const uint16\_t GyroScaleFactor = 131;

// MPU6050 few configuration register addresses

const uint8\_t MPU6050\_REGISTER\_SMPLRT\_DIV = 0x19;

const uint8\_t MPU6050\_REGISTER\_USER\_CTRL = 0x6A;

const uint8\_t MPU6050\_REGISTER\_PWR\_MGMT\_1 = 0x6B;

const uint8\_t MPU6050\_REGISTER\_PWR\_MGMT\_2 = 0x6C;

const uint8\_t MPU6050\_REGISTER\_CONFIG = 0x1A;

const uint8\_t MPU6050\_REGISTER\_GYRO\_CONFIG = 0x1B;

const uint8\_t MPU6050\_REGISTER\_ACCEL\_CONFIG = 0x1C;

const uint8\_t MPU6050\_REGISTER\_FIFO\_EN = 0x23;

const uint8\_t MPU6050\_REGISTER\_INT\_ENABLE = 0x38;

const uint8\_t MPU6050\_REGISTER\_ACCEL\_XOUT\_H = 0x3B;

const uint8\_t MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET = 0x68;

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

void setup() {

Serial.begin(9600);

///////////////////////////GPS SETUP//////////////////////////////////////////

mySerialGps.begin(9600);

Serial.println("Hi initilizing the system.");

delay(100);

ss.begin(GPSBaud);

Serial.println("In recieving mode.");

mySerialGps.println("AT+CNMI=2,2,0,0,0\r"); // AT Command to receive a live SMS

delay(1000);

////////////////////////////////////////////////////////////////////////////

//////////////////////////MPU 6050 SETUP/////////////////////////////////////

Wire.begin();

// sgps.begin(9600);

Serial.println(F("Initializing I2C devices..."));

mpu.initialize();

// verify connection

Serial.println(F("Testing device connections..."));

Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));

// load and configure the DMP

Serial.println(F("Initializing DMP..."));

mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity

mpu.setXGyroOffset(220);

mpu.setYGyroOffset(76);

mpu.setZGyroOffset(-85);

mpu.setZAccelOffset(1788);

pinMode(command, OUTPUT);

pinMode(buzzer, OUTPUT);

pinMode(button, INPUT);

///////////////////////////////////////////////////////////////////////////////

}

void loop() {

if (flag) {

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

Read\_RawValue(MPU6050SlaveAddress, MPU6050\_REGISTER\_ACCEL\_XOUT\_H);

//divide each with their sensitivity scale factor

Ax = (double) AccelX / AccelScaleFactor;

Ay = (double) AccelY / AccelScaleFactor;

Az = (double) AccelZ / AccelScaleFactor;

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

Gx = (double) GyroX / GyroScaleFactor;

Gy = (double) GyroY / GyroScaleFactor;

Gz = (double) GyroZ / GyroScaleFactor;

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);

if (Gy >= 05.0 || Gy <= -05.0 || Gx >= 05.0 || Gx <= -05) {

Serial.println("ANGLE CHANGE IN Z DIRECTION IS SUDDEN!!!!");

//////////////

sendSms();

////////////////

int x = millis();

int time = millis() - x;

while (time <= 10000 && digitalRead(button) == LOW) {

time = millis() - x;

digitalWrite(buzzer, HIGH);

delay(1000);

}

digitalWrite(buzzer, LOW);

if (time >= 1000) {

flag = false;

digitalWrite(command, HIGH);

}

}

}

if (digitalRead(button) == HIGH) {

flag = true;

digitalWrite(command, LOW);

}

while (ss.available() > 0)

if (gps.encode(ss.read()))

displayInfo();

if (millis() > 5000 && gps.charsProcessed() < 10) {

//Serial.println(F("No GPS detected: check wiring."));

//while (true);

displayInfo();

}

delay(100);

}

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 MPU6050

void MPU6050\_Init() {

delay(1500);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SMPLRT\_DIV, 0x07);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_PWR\_MGMT\_1, 0x01);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_PWR\_MGMT\_2, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_CONFIG, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_GYRO\_CONFIG, 0x00); //set +/-250 degree/second full scale

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_ACCEL\_CONFIG, 0x00); // set +/- 2g full scale

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_FIFO\_EN, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_INT\_ENABLE, 0x01);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_USER\_CTRL, 0x00);

}

void sendSms() {

mySerialGps.println("AT+CMGF=1\r"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1 second

mySerialGps.println("AT+CMGS=\"+919986624722\"\r"); // Replace x with mobile number

delay(1000);

String mesage = "ATTENTION!! Your device has met with a serious accident the vehicle location is : ";

mesage.concat(url);

mySerialGps.println(mesage); // The SMS text you want to send

delay(100);

mySerialGps.println((char) 26); // ASCII code of CTRL+Z for saying the end of sms to the module

delay(1000);

Serial.println("In recieving mode.");

mySerialGps.println("AT+CNMI=2,2,0,0,0\r");

}

void displayInfo() {

//Serial.print(F("Location: "));

if (gps.location.isValid()) {

url = "http://www.google.com/maps/place/";

url.concat(String(gps.location.lat(), 6));

url.concat(",");

url.concat(String(gps.location.lng(), 6));

Serial.println(url);

delay(1000);

} else {

Serial.print(F("INVALID"));

url = "http://www.google.com/maps/place/";

url.concat(default\_co\_ordinates);

delay(1000);

//Serial.println(url);

}

if (mySerial.available() > 0) {

Serial.print(mySerial.read());

}

Serial.println();

count++;

//delay(1000);

}