#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h> #include <WiFiClient.h>

#include <SoftwareSerial.h> #define DHTPIN D2 #define DHTTYPE DHT11

#include <Adafruit\_SSD1306.h> #include <MPU6050.h>

#define OLED\_Address 0x3C // 0x3C device address of I2C OLED. Few other OLED has 0x3D

MPU6050 mpu;

int a=0;

int lasta=0; int lastb=0;

int LastTime=0; int ThisTime;

bool BPMTiming=false; bool BeatComplete=false; int BPM=0;

static String fdata; static String fdata1;

#define UpperThreshold 560

#define LowerThreshold 530 float sensorValue;

void setup() { Serial.begin(9600);

if (!oled.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { Serial.println(F("SSD1306 allocation failed"));

while (true);

}

float temp; //Stores temperature value String PIRData;

String MOData; String MOData1; int bPIR = 0;

int bMO = 0;

int flag=0; int flag1=0; int fcount=0;

String gpsString=""; const int buttonPin = D5; int buttonState = 0;

const int buttonPin1 = D6; int buttonState1 = 0;

// Your WiFi credentials.

// Set password to "" for open networks.

//const char \*ssid = "jes";

//const char \*pass = "pass"; const char \*ssid = "IOT"; const char \*pass = "password";

const char\* host = "jescloud.in"; void setup()

{

dht.begin();

pinMode(buttonPin, INPUT); pinMode(buttonPin1, OUTPUT); Serial.begin(115200); gps.begin(9600); Serial.println("Connecting to "); Serial.println(ssid); WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500); Serial.print(".");

}

Serial.println(""); Serial.println("WiFi connected");

delay(4000);

}

void loop()

{

String postData;

buttonState = digitalRead(buttonPin); hum = dht.readHumidity();

temp= dht.readTemperature();

gpsString="";

while (gps.available()>0 && gps.available()>0)

{

char inChar = (char)gps.read(); gpsString+= inChar;

}

gpsString.trim();

String Result=gpsString.substring(1, 2);

String Result1=gpsString.substring(3, 4); Serial.println(Result); Serial.println(Result1); MOData1=gpsString;

WiFiClient client; HTTPClient http,http1; if (connect(host,80))

{

postData = "?t=" + String(Result) + "+g=" + String(Result1) + "+c=" + String(temp); http.begin("<http://gladecloud.in/24_IOT_Traffic_Health.aspx>"+ postData);

//Specify request destination

//http.addHeader("Content-Type", "application/x-www-form-urlencoded"); int httpCode = http.GET(); //Send the request

String payload = http.getString(); //Get the response payload

//String payload = http.getString(); //Get the response payload Serial.println(postData); //Print HTTP return code

// Serial.println(httpCode); http.end();

if (buttonState == HIGH)

{

flag=1;

}

if(flag==1)

{

digitalWrite(buttonPin1, HIGH); http1.begin("<http://gladecloud.in/24_frm_IOT_SMS.aspx>"); int httpCode1 = http1.GET();

String payload1 = http1.getString(); http1.end();

Serial.println("SMS Send.."); delay(1000);

delay(1000); digitalWrite(buttonPin1, LOW); flag=0;

}

//Close connection Serial.println("Waiting..."); delay(1000);

}

}

mpu.calibrateGyro(); mpu.setThreshold(3); Serial.println("MPU6050 Interface");

oled.clearDisplay(); // clear display oled.setTextSize(2); // text size

oled.setTextColor(WHITE); // text color oled.setCursor(0, 10); // position to display oled.println("WOMEN"); // text to display oled.println("SAFETY"); // text to display oled.println("SYSTEM"); // text to display oled.display(); // show on OLED delay(2000);

}

void loop()

{

unsigned int airdata = analogRead(A1); sensorValue = airdata; ThisTime=millis();

int value=analogRead(0); oled.setTextColor(WHITE); lastb=10;

lasta=a; if(value>UpperThreshold)

{

if(BeatComplete)

{

BPM=ThisTime; BPM=int(60/(float(BPM))); BPMTiming=false; BeatComplete=false;

}

if(BPMTiming==false)

{

BPMTiming=true;

}

}

if((value<LowerThreshold)&(BPMTiming))

{

BeatComplete=true;

}

Vector rawAccel = mpu.readRawAccel();

Vector normAccel = mpu.readNormalizeAccel(); Vector rawGyro = mpu.readRawGyro();

Vector normGyro = mpu.readNormalizeGyro();

if(normAccel.XAxis>=2)

{

digitalWrite(8, LOW); oled.clearDisplay(); // clear display

oled.setTextSize(2); // text size oled.setTextColor(WHITE); // text color oled.setCursor(0, 10); // position to display oled.print("BPM:");

oled.println(BPM); oled.println("Postion:"); oled.println("Normal"); oled.display();

}

else

{

oled.clearDisplay(); // clear display oled.setTextSize(2); // text size oled.setTextColor(WHITE); // text color oled.setCursor(0, 10); // position to display oled.print("BPM:");

oled.println(BPM); oled.println("Postion:"); oled.println("ABNormal"); oled.display();

}

}

void accelShow