**Program used in Arduino Nano:**

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include "MAX30105.h"

#include "DHT.h"

#include <floatToString.h>

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include "heartRate.h"

#include <RTClib.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

#define SCREEN\_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32

#define DHTPIN 2

#define DHTTYPE DHT11

#define GPSRXPin 3

#define GPSTXPin 4

#define GSMRXPin 5

#define GSMTXPin 6

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

MAX30105 pulseSensor;

DHT dht(DHTPIN, DHTTYPE);

TinyGPSPlus gps;

SoftwareSerial GPS\_serial(GPSRXPin, GPSTXPin);

SoftwareSerial GSM\_serial(GSMRXPin, GSMTXPin); // RX, TX for GSM

const byte RATE\_SIZE = 4; //Increase this for more averaging. 4 is good.

byte rates[RATE\_SIZE]; //Array of heart rates

byte rateSpot = 0;

long lastBeat = 0; //Time at which the last beat occurred

double latitude = 0.0;

double longitude = 0.0;

const char phoneNumber1[] = "+919483926205";

uint8\_t bpm\_min = 50;

uint8\_t bpm\_max = 90;

uint8\_t bpm\_alert\_counter = 0;

long last\_alert\_sent\_millis = 0;

float beatsPerMinute;

void setup()

{

Serial.begin(9600);

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

// Initialize display

if(!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {

Serial.println(F("SSD1306 allocation failed"));

for(;;); // Don't proceed, loop forever

}

delay(500);

display\_clear();

display\_write("Initializing...", 5, 20, 1);

// Initialize pulse sensor

if (!pulseSensor.begin(Wire, I2C\_SPEED\_FAST)) //Use default I2C port, 400kHz speed

{

Serial.println(F("MAX30105 was not found. Please check wiring/power. "));

while (1);

}

pulseSensor.setup(); //Configure sensor with default settings

pulseSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running

pulseSensor.setPulseAmplitudeGreen(0); //Turn off Green LED

dht.begin();

GPS\_serial.begin(9600);

delay(500);

display\_clear();

display\_write("READY", 20, 20, 3)

delay(100);

Serial.println(F("READY"));

}

void loop()

{

double lat, lng;

bool location\_updated = 0;

int bpm = get\_BPM();

if((bpm > -1) && ((bpm < bpm\_min) || (bpm > bpm\_max))) bpm\_alert\_counter++;

else bpm\_alert\_counter = 0;

float temperature = dht.readTemperature();

char tempString[6];

dtostrf(temperature, 4, 2, tempString);

float humidity = dht.readHumidity();

char humidString[7];

dtostrf(humidity, 4, 0, humidString);

bool gas\_detected = digitalRead(A0) ? 0 : 1;

// while(!location\_updated) location\_updated = update\_location();

for(int i=0; i<50; i++)

{

if(update\_location()) break;

delay(1);

}

// update\_time();

display\_clear();

display\_write("BPM: ", 1, 1, 3);

display\_write("T(C): ", 1, 30, 2);

display\_write("H(%): ", 1, 49, 2);

if(bpm > -1) display\_write(String(bpm), 70, 1, 3);

display\_write(tempString, 60, 30, 2);

display\_write(String(humidity), 60, 49, 2);

Serial.println("BPM: " + String(bpm) + ", Temperature: " + tempString + ", Humidity: " + String(humidity) + "\n");

if(gas\_detected)

{

display\_clear();

display\_write("Gas", 32, 5, 3);

display\_write("detect!", 2, 35, 3);

delay(1000);

if(millis() > (last\_alert\_sent\_millis + 60000))

{

send\_alert(bpm, temperature, humidity, gas\_detected);

last\_alert\_sent\_millis = millis();

}

delay(1000);

display\_clear();

}

if((bpm\_alert\_counter > 15) && (millis() > (last\_alert\_sent\_millis + 60000)))

{

send\_alert(bpm, temperature, humidity, gas\_detected);

last\_alert\_sent\_millis = millis();

bpm\_alert\_counter = 0;

}

}

void display\_write(String text, int x, int y, int s)

{

display.setTextColor(BLACK);

display.setCursor(x,y);

display.setTextSize(s);

display.println(text);

display.display();

}

void display\_clear()

{

// display.clearDisplay();

display.fillRect(0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT, SSD1306\_WHITE);

}

int get\_BPM()

{

long irValue, beatAvg;

long start\_time = millis();

while(millis() < (start\_time + 3000))

{

irValue = pulseSensor.getIR();

if (checkForBeat(irValue) == true)

{

//We sensed a beat!

long delta = millis() - lastBeat;

lastBeat = millis();

beatsPerMinute = 60 / (delta / 1000.0);

if (beatsPerMinute < 255 && beatsPerMinute > 20) {

rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array

rateSpot %= RATE\_SIZE; //Wrap variable

//Take average of reading

beatAvg = 0;

for (byte x = 0 ; x < RATE\_SIZE ; x++)

beatAvg += rates[x];

beatAvg /= RATE\_SIZE;

}

}

}

if (irValue < 50000)

{

// Serial.println(F("Finger not detected"));

return -1;

}

else

{

Serial.print(F("IR="));

Serial.print(irValue);

Serial.print(F(", BPM="));

Serial.print(beatsPerMinute);

Serial.print(F(", Avg BPM="));

Serial.println(beatAvg);

return beatAvg;

}

}

void send\_alert(int heart\_rate, float temperature\_C, float humidity\_percent, bool gas\_detected)

{

char latBuffer[12];

char lngBuffer[12];

dtostrf(latitude, 10, 6, latBuffer); // Convert latitude to string with 6 decimal places

dtostrf(longitude, 10, 6, lngBuffer);

Serial.println(F("Sending alert!"));

display\_clear();

display\_write("Sending", 2, 5, 3);

display\_write("Alert!", 3, 35, 3);

GSM\_serial.print("AT+CMGS=\"");

GSM\_serial.print(phoneNumber1);

GSM\_serial.println("\"");

delay(1000);

GSM\_serial.print("Heart Rate: ");

GSM\_serial.print(heart\_rate);

GSM\_serial.print("\nTemperature: ");

GSM\_serial.print(temperature\_C);

GSM\_serial.print("\nHumidity: ");

GSM\_serial.print(humidity\_percent);

if(gas\_detected)

{

GSM\_serial.print("\nGas detected!");

}

GSM\_serial.print("\n\nLatitude: ");

GSM\_serial.print(latitude);

GSM\_serial.print("\nLongitude: ");

GSM\_serial.println(longitude);

delay(1000);

GSM\_serial.write(26); // ASCII code for CTRL+Z to send the SMS

delay(1000);

delay(1000);

display\_clear();

}

bool update\_location()

{

Serial.println("Reading coordinates...");

// GPS\_serial.begin(9600);

// delay(50);

while(GPS\_serial.available() > 0)

{

if (gps.encode(GPS\_serial.read()))

{

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

if (gps.location.isValid())

{

// Serial.print(gps.location.lat(), 6);

// Serial.print(F(","));

// Serial.print(gps.location.lng(), 6);

latitude = gps.location.lat();

longitude = gps.location.lng();

Serial.print(latitude, 6);

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

Serial.println(longitude, 6);

return 1;

}

}

}

return 0;

}

void update\_time()

{

if(GPS\_serial.available() > 0)

{

if (gps.encode(GPS\_serial.read()))

{

if (gps.time.isUpdated() && gps.date.isUpdated())

{

// Get UTC time from GPS

uint16\_t year = gps.date.year();

uint8\_t month = gps.date.month();

uint8\_t day = gps.date.day();

uint8\_t hour = gps.time.hour();

uint8\_t minute = gps.time.minute();

uint8\_t second = gps.time.second();

// Convert to DateTime object

DateTime utcTime(year, month, day, hour, minute, second);

// Convert UTC to IST

DateTime istTime = convertToIST(utcTime);

// Print IST time

char buffer[25];

sprintf(buffer, "%02d-%02d-%02d %02d:%02d:%02d", istTime.year(), istTime.month(), istTime.day(), istTime.hour(), istTime.minute(), istTime.second());

Serial.println(buffer);

}

}

}

}

DateTime convertToIST(DateTime utcTime)

{

long totalSeconds = utcTime.unixtime() + (5 \* 3600) + (30 \* 60);

return DateTime(totalSeconds);

}