Arduino code:

#include "ThingSpeak.h"

#include <Wire.h>

#include "MAX30100\_PulseOximeter.h"

#include <LiquidCrystal\_I2C.h>

#include <ESP8266WiFi.h>

#include <SoftwareSerial.h>

#define PIN\_TX D6

#define PIN\_RX D7

SoftwareSerial mySerial(PIN\_TX,PIN\_RX);

WiFiClient client;

LiquidCrystal\_I2C lcd(0x3F,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

char ssid[] = "HM"; // your network SSID (name)

char pass[] = "12345678"; // your network password

#define REPORTING\_PERIOD\_MS 2000

#define REPORTING\_PERIOD\_online 200000

unsigned long myChannelNumber = 1229863;

const char \* myWriteAPIKey = "PAT8NLMKRHVGBPSZ";

int bpm;

int spo2;

int temp;

int m;

int x1;

int only1=0;

int bpmc;

int spo2c;

int tempc;

// PulseOximeter is the higher level interface to the sensor

// it offers:

// \* beat detection reporting

// \* heart rate calculation

// \* SpO2 (oxidation level) calculation

PulseOximeter pox;

uint32\_t tsLastReport = 0;

uint32\_t pre = 0;

// Callback (registered below) fired when a pulse is detected

void onBeatDetected()

{

pox.update();

Serial.println("Beat!");

bpm=pox.getHeartRate();

spo2=pox.getSpO2();

x1=1;

lcd.setCursor(7,1);

lcd.print("beat");

}

void setup()

{

pinMode(D5,OUTPUT);digitalWrite(D5,0);

delay(3000);

digitalWrite(D8,0);

Serial.begin(115200);

mySerial.begin(9600);

delay(500);

delay(500);

digitalWrite(D0,0);

Serial.print("Initializing pulse oximeter..");

lcd.init(); // initialize the lcd

// Print a message to the LCD.

lcd.backlight();

lcd.clear();

lcd.setCursor(2,0);

lcd.print("smart health !");

lcd.setCursor(2,1);

lcd.print("initilizeing");

if (!pox.begin()) {

Serial.println("FAILED");

for(;;);

} else {

Serial.println("SUCCESS");

}

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

delay(100);

Serial.print("Connecting Wifi: ");

Serial.println(ssid);

WiFi.begin(ssid,pass);

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

Serial.print(".");

delay(1000);

}

ThingSpeak.begin(client); // Initialize ThingSpeak

// Initialize the PulseOximeter instance

// Failures are generally due to an improper I2C wiring, missing power supply

// or wrong target chip

if (!pox.begin()) {

Serial.println("FAILED");

for(;;);

} else {

Serial.println("SUCCESS");

}

mySerial.println("AT"); //Once the handshake test is successful, it will back to OK

pox.setOnBeatDetectedCallback(onBeatDetected);

}

void loop()

{

// Make sure to call update as fast as possible

pox.update();

// Asynchronously dump heart rate and oxidation levels to the serial

// For both, a value of 0 means "invalid"

if (millis() - tsLastReport > REPORTING\_PERIOD\_MS) {

temp=72-analogRead(0)/5;

if(temp<=31)

temp=0;

if(temp>=35)

tempc=temp;

if(bpm>90 || bpm<70){

bpm=0;spo2=0;

}

if(spo2>100 || spo2<80){

bpm=0;spo2=0;

}

if (bpm>70 & bpm<90){

tempc=temp;

bpmc=bpm;

spo2c=spo2;

Serial.println("sending");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("HR=");

lcd.print(bpm);

lcd.setCursor(6,0);

lcd.print(" Temp=");

lcd.print(temp);

lcd.setCursor(0,1);

lcd.print("spo2=");

lcd.print(spo2);

if(WiFi.status() == WL\_CONNECTED){

ThingSpeak.setField(1, tempc);

ThingSpeak.setField(2, spo2c);

ThingSpeak.setField(3,bpmc );

lcd.setCursor(7,1);

lcd.print(" send!");

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

if(x == 200){

Serial.println("Channel update successful.");

}

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

} delay(500); ESP.restart();

}

Serial.print("temp: ");

Serial.println(temp);

Serial.print("Heart rate:");

Serial.print(bpm);

Serial.print("bpm / SpO2:");

Serial.print(spo2);

Serial.println("%");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("HR=");

lcd.print(bpm);

lcd.setCursor(6,0);

lcd.print(" Temp=");

lcd.print(temp);

lcd.setCursor(0,1);

lcd.print("spo2=");

lcd.print(spo2);

if(x1==1){

// m++;

}

if(m>=3){

x1=0;

bpm=0;

spo2=0;

lcd.setCursor(7,1);

lcd.print(" ");

}

if(bpm==0){

lcd.setCursor(7,1);

lcd.print(" ");}

if(temp>28){

// bpm= random(80,85);

//spo2=random(94,98);

}

else{

bpm=0;

spo2=0;

lcd.setCursor(9,1);

lcd.print(" ");

}

if(temp>38 &temp<=42 &only1==0){

Serial.println("send massage");

mySerial.println("AT+CMGF=1"); // Configuring TEXT mode

updateSerial();

mySerial.println("AT+CMGS=\"+967776877668\"");//change ZZ with country code and xxxxxxxxxxx with phone number to sms

updateSerial();

mySerial.print("high temp of the patient "+String(temp)); //text content

updateSerial();

mySerial.write(26);

updateSerial();

mySerial.println("ATH");

updateSerial();

only1=1;

}

if(temp>42 &only1==0){

Serial.println("call");

mySerial.println("AT+CMGF=1"); // Configuring TEXT mode

updateSerial();

updateSerial();

updateSerial();

mySerial.println("ATD + +967775246785;");

updateSerial();

delay(15000);

mySerial.println("ATH");

updateSerial();

only1=1;

}

if(temp<35){

only1=0;

}

tsLastReport = millis();

}

}

void updateSerial()

{

delay(500);

while (Serial.available())

{

// mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port

}

while(mySerial.available())

{

Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port

}

}