#include "MAX30100\_PulseOximeter.h"

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

#include "UbidotsEsp32Mqtt.h"

#define SENSOR A0

#define REPORTING\_PERIOD\_MS 1000

const char \*UBIDOTS\_TOKEN ="BBUS-rsXMWZn0no4jgVn9AR2p2fuZMRRYI9";

const char \*WIFI\_SSID="POCO F5";

const char \*WIFI\_PASS="1234567890";

const char \*DEVICE\_LABEL="ESP32";

const char \*VARIABLE\_LABEL1="Pulse Rate";

const char \*VARIABLE\_LABEL2="SpO2";

const char \*VARIABLE\_LABEL3="ECG";

unsigned long timer;

Ubidots ubidots (UBIDOTS\_TOKEN);

PulseOximeter pox;

uint32\_t tsLastReport=0;

void onBeatDetected(){

  ;

}

void callback(char \*topic,byte \*payload,unsigned int length)

{

  Serial.print("Message arrived[");

  Serial.print(topic);

  Serial.print("]");

  for(int i=0;i<length;i++)

  {

    Serial.print((char)payload[i]);

  }

  Serial.println();

}

void setup(){

  Serial.begin(115200);

  lcd.init();

  lcd.backlight();

  ubidots.connectToWifi(WIFI\_SSID,WIFI\_PASS);

  ubidots.setCallback(callback);

  ubidots.setup();

  ubidots.reconnect();

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

  if(!pox.begin()){

    Serial.println("FAILED");

    for(;;)

    {

      ;

    }

  }else{

    Serial.println("SUCESS");

    digitalWrite(1,HIGH);

  }

  pox.setIRLedCurrent(MAX30100\_LED\_CURR\_7\_6MA);

  pox.setOnBeatDetectedCallback(onBeatDetected);

}

void loop()

  {

  if(!ubidots.connected()){

    ubidots.reconnect();

  }

  pox.update();

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

    ubidots.add(VARIABLE\_LABEL1,pox.getHeartRate());

    ubidots.publish(DEVICE\_LABEL);

    ubidots.add(VARIABLE\_LABEL2,pox.getSpO2());

    ubidots.publish(DEVICE\_LABEL);

    float sensor = analogRead(SENSOR);

    ubidots.add(VARIABLE\_LABEL3,sensor);

    ubidots.publish(DEVICE\_LABEL);

    lcd.setCursor(0,0);

    lcd.print("BPM:");

    lcd.print(pox.getHeartRate());

    lcd.setCursor(0,1);

    lcd.print("SpO2:");

    lcd.print(pox.getSpO2());

    lcd.print("%");

    ubidots.loop();

    tsLastReport=millis();

  }

}