//KODLAR

#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 9);

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //lcd(9, 8, 4, 3, 2, 1); lcd(7, 8, 9, 10, 11, 12);

// Değişkenler

int pulsePin = 0; // Turuncu kabloyu analog 0 pinine bağladık

int blinkPin = 13; // pin her atışta ledin yanması için

int fadePin = 5; // pin her atışta buzzerin çalması için

int fadeRate = 0; // PWM devresinde led ve buzzerin sönmesi için kullandık

// Geçici değerler, aralıklı servis rutini için kullanılır.

int BPM; // int işlenmemiş analog 0 değerlerini 2ms de bir günceller

volatile int Signal; // yeni işlenmemiş data girişlerini tutar

volatile int IBI = 600; // int Atışlar arasındaki aralıklı verileri tutar

volatile boolean Pulse = false; // Kalp atışı tespit edildiğinde doğru, edilemezse yanlış

volatile boolean QS = false; // Arduino atış bulduğunda doğru olur.

//Seri çıkışları sunar ihiyacımıza göre ayarlarız

static boolean serialVisual = false; // Varsayılanı yanlış ayarlar. Arduino seri ekranındaki sanal sinyalleri görmek için tekrardan doğru ayarla

void setup(){

lcd.begin(16, 2);

lcd.print("Baslatiliyor...");

pinMode(blinkPin,OUTPUT); // pin kalp atışına göre yanacaktır

pinMode(fadePin,OUTPUT); // pin kalp atışına göre sönecektir.

Serial.begin(115200); // Hızlı konuştuğumuzu kabul edelim! 115200

mySerial.begin(9600);

interruptSetup(); // Nabız sensörünü 2ms de bir okunmasını ayarlar

// Eğer nabız sensörünün voltu boardın voltundan düşük ise

// Sıradaki çizgiyi yorumlama ve bu voltaj değerini referans olarak kabul et

// analogReference(EXTERNAL);

delay(3000);

//lcd.setCursor(0,0);

lcd.clear();

lcd.print("Baslatildi");

delay(3000);

lcd.clear();

lcd.print("KalpHiziniz");

}

// Hadi başlayalım!

void loop(){

serialOutput() ;

if (QS == true){ // Bir kalp atışı bulundu.

// Dakika başına atış ve İnterbeat aralığı hesaplandı.

// Arduino kalp atışı bulduğunda "QS" doğru

digitalWrite(blinkPin,HIGH);

fadeRate = 255; // Led e sönme efekti yap.

// FadeRate komutu için led'in değerini 255'e ayarla(Ledin atışa göre sönmesi)

serialOutputWhenBeatHappens(); // Atış gerçekleştiğinde seri olarak çıktı verir

QS = false; // Sonraki zamana kadar QS'i resetler

}

ledFadeToBeat(); // Led'e söndürme efekti yap

lcd.setCursor(11,0);

lcd.print(" ");

lcd.setCursor(13,0);

lcd.print(BPM);

mySerial.print(BPM);

if(BPM <= 75)

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Hareket Ediniz");

mySerial.println(" Hareket Ediniz");

}

else if(BPM >= 120)

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Dinlen");

mySerial.println(" Dinlen");

}

else

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Stabil");

mySerial.println(" Stabil");

}

delay(1000); // Ara ver

}

void ledFadeToBeat(){

fadeRate -= 15; // Ledin sönme değerini ayarla

fadeRate = constrain(fadeRate,0,255); // Negatif değerler geldikçe ledi sönük tut

analogWrite(fadePin,fadeRate); // Ledi söndür

}

//EKRAN ÇIKTISI KODLARI

///////// Bütün kullanılan seri kodlar

///////// Okunan değere göre değişebilir

///////// Kodları başladığı bildirildiğinde doğru ve yanlış olarak ayarla

void serialOutput(){ // Çıkışın nasıl gerçekleşeğine karar ver.

if (serialVisual == true){

arduinoSerialMonitorVisual('-', Signal); // bu fonksiyon ekran görüntüleyiciye gider

} else{

sendDataToSerial('S', Signal); // seri data gönderme komutuna gider

}

}

// Dakika başına atış ve interbeat aralığının nasıl çıkış vereceğine karar verir.

void serialOutputWhenBeatHappens(){

if (serialVisual == true){ // Ekran görüntüleyicinin çalışma kodu

Serial.print("\*\*\* Heart-Beat Happened \*\*\* ");

Serial.print("BPM: ");

Serial.print(BPM);

Serial.print(" ");

} else{

sendDataToSerial('B',BPM); // kalp atışını "B" ye gönder

sendDataToSerial('Q',IBI); // atışlar arasındaki zaman farkını "Q" ya gönder

}

}

void sendDataToSerial(char symbol, int data ){

Serial.print(symbol);

Serial.println(data);

}

// Ekran görüntüleyecinin çalışmasını sağlayan kodlar

void arduinoSerialMonitorVisual(char symbol, int data ){

const int sensorMin = 0;

const int sensorMax = 1024;

int sensorReading = data;

// 12 farklı çeşitte sensörü yorumlar:

int range = map(sensorReading, sensorMin, sensorMax, 0, 11);

// sensör rangeine bağlı olarak farklı sonuç çıkartır

switch (range) {

case 0:

Serial.println(""); /////ASCII Art Madness

break;

case 1:

Serial.println("---");

break;

case 2:

Serial.println("------");

break;

case 3:

Serial.println("---------");

break;

case 4:

Serial.println("------------");

break;

case 5:

Serial.println("--------------|-");

break;

case 6:

Serial.println("--------------|---");

break;

case 7:

Serial.println("--------------|-------");

break;

case 8:

Serial.println("--------------|----------");

break;

case 9:

Serial.println("--------------|----------------");

break;

case 10:

Serial.println("--------------|-------------------");

break;

case 11:

Serial.println("--------------|-----------------------");

break;

}

}

//KAPATMA KODLARI

volatile int rate[10];

volatile unsigned long sampleCounter = 0;

volatile unsigned long lastBeatTime = 0;

volatile int P =512;

volatile int T = 512;

volatile int thresh = 525;

volatile int amp = 100;

volatile boolean firstBeat = true;

volatile boolean secondBeat = false;

void interruptSetup(){

TCCR2A = 0x02;

TCCR2B = 0x06;

OCR2A = 0X7C;

TIMSK2 = 0x02;

sei();

}

ISR(TIMER2\_COMPA\_vect){

cli();

Signal = analogRead(pulsePin);

sampleCounter += 2;

int N = sampleCounter - lastBeatTime;

if(Signal < thresh && N > (IBI/5)\*3){

if (Signal < T){

T = Signal;

}

}

if(Signal > thresh && Signal > P){

P = Signal;

}

if (N > 250){

if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)\*3) ){

Pulse = true;

digitalWrite(blinkPin,HIGH);

IBI = sampleCounter - lastBeatTime;

lastBeatTime = sampleCounter;

if(secondBeat){

secondBeat = false;

for(int i=0; i<=9; i++){

rate[i] = IBI;

}

}

if(firstBeat){

firstBeat = false;

secondBeat = true;

sei();

return;

}

word runningTotal = 0;

for(int i=0; i<=8; i++){

rate[i] = rate[i+1];

runningTotal += rate[i];

}

rate[9] = IBI;

runningTotal += rate[9];

runningTotal /= 10;

BPM = 60000/runningTotal;

QS = true;

}

}

if (Signal < thresh && Pulse == true){

digitalWrite(blinkPin,LOW);

Pulse = false;

amp = P - T;

thresh = amp/2 + T;

P = thresh;

T = thresh;

}

if (N > 2500){

thresh = 512;

P = 512;

T = 512;

lastBeatTime = sampleCounter;

firstBeat = true;

secondBeat = false;

}

sei();

}

// end isr