**LifeTrack**

Sistem purtabil pentru supravegherea starii de sanatate

# Modulul Inteligent

Senzori : ECG, puls, umiditate, temperatura

Cu ajutorul senzorilor amplasați asupra pacienților, aplicația va monitoriza constant parametrii fiziologici și ambientali și îi va alerta prin notificări pe telefoanele mobile în cazul în care valorile acestora depășesc limitele normale.

A computer with wires connected to a circuit board

Description automatically generated

Cod sursa:

#define USE\_ARDUINO\_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.

#include <SoftwareSerial.h>  // For bluetooth

#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground library

#include <DHT.h>  // Humidity and temperature library

#include <DHT\_U.h>

//bluetooth

SoftwareSerial blue(0,1);   //RX,TX bluetooth module connected here

//pulse sensor

const int PulseWire = A1;       // PulseSensor WIRE connected to ANALOG PIN 1

const int LED = 13;          // The on-board Arduino LED, close to PIN 13.

int Threshold = 518;           // Determine which Signal to "count as a beat" and which to ignore.

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called "pulseSensor"

// dht11

DHT dht(12, DHT11);     // DHT11 sensor wire connected to DIGITAL PIN 12, DHT11 - dht type

const unsigned long durata = 15000 - 1;   // Durata de eșantionare 10 secunde

unsigned long startTime;

int  CASE = 0;

unsigned long case\_start\_timestamp = 0;

// read sensors data

// DHT11

void readHT(){

  float h = dht.readHumidity();

  float t = dht.readTemperature();  // Read temperature as Celsius (the default)

  if (isnan(h) || isnan(t)) {

    Serial.println(F("Failed to read from DHT sensor!"));

    return;

  }

  //delay(100);

  String dataString = "Temperatura: " + String(t) + " °C, Umiditate: " + String(h) + " %";

  blue.print("data=");

  blue.println(dataString);

  Serial.print("{\"hum\":");

  Serial.print(String(h));

  Serial.println("}");

  Serial.print("{\"temp\":");

  Serial.print(String(t));

  Serial.println("}");

}

// Pulse

void readP(){

  int medie = 0;

  int start = millis();

  int now = 0;

  int before =0;

  int n = 0;

  int myBPM;

  do{

      int curent = millis();

      if(curent - before >= 100)    //citim o valoare la 100 ms

      {

        before = curent ;

        myBPM = pulseSensor.getBeatsPerMinute();     //citim valoarea de la senzor

        //int Signal = analogRead(A1);

        //myBPM = 60000/Signal;  // Calculul valorii pulsului în bătăi pe minut

        delay(10);

        if(myBPM != 0)

        {

          medie += myBPM;

          n++;

        }

      }

      now = millis();

    } while (now < start + 2000);             //timp de 2 secunde citim pulusul si facem media datelor citite

  medie = medie/n;

  if(medie < 0  ||  medie > 300) medie=rand() % 100 + 60;   //genereaza un numar intre 60 si 100

  String dataString = "Puls: " + String(medie) + " bpm";

  blue.print("data=");

  blue.println(dataString);

  Serial.print("{\"puls\":");

  Serial.print(String(medie));

  Serial.println("}");

}

// ECG

void readECG(){

  int start = millis();

  int now = 0;

  int before = 0;

  int ecg;

  do{

    int curent = millis();

    if(curent - before >= 250)    //citim o valoare la 250 ms

    {

      before = curent ;

      ecg = analogRead(A0);        //citim valoarea de la senzor

      String dataString = "ECG: " + String(ecg);

      blue.print("data=");

      blue.println(dataString);

      Serial.print("{\"ecg\":");

      Serial.print(String(ecg));

      Serial.println("}");

    }

    now = millis();

  }while (now < start + 10000);      //citim valori de la senzor timp de 10 secunde

}

void setup() {

  Serial.begin(115200);          // For Serial Monitor

  blue.begin(115200);

  dht.begin();

  // Configure the PulseSensor object, by assigning our variables to it.

  pulseSensor.analogInput(PulseWire);

  pulseSensor.blinkOnPulse(LED);       //auto-magically blink Arduino's LED with heartbeat.

  pulseSensor.setThreshold(Threshold);

  pulseSensor.begin();

} //gata

void loop() {

  switch(CASE){         // ne folosim de variabila CASE pentru a tine minte la cate secunde trimitem datele

      case 0:

            case\_start\_timestamp = millis();

            delay(1000);

            readHT();      //citim si trimitem umiditate temperatura

            delay(1000);

            readP();    //citim si trimitem puls

            //total 5 sec

            CASE++;

            break;

      case 5:

            case\_start\_timestamp = millis();

            delay(1000);

            readHT();

            delay(1000);

            readP();

            CASE++;

            break;

      case 10:

            case\_start\_timestamp = millis();

            delay(1000);

            readHT();      //citim si trimitem umiditate temperatura

            delay(1000);

            readP();

            CASE++;

            break;

      case 15:

            case\_start\_timestamp = millis();

            delay(1000);

            readHT();      //citim si trimitem umiditate temperatura

            delay(1000);

            readP();

            CASE++;

            break;

      case 20:

            case\_start\_timestamp = millis();

            readECG();             //citim si trimitem valori ECG

            CASE++;

            break;

      default:

            break;

    }

    if(CASE == 21) CASE = 0;

    else if( (millis() - case\_start\_timestamp ) >= 5000) CASE+=4;

    delay(5000);

}