

SerenITy

Introducere

SerenITy este un altfel de ursulet de plus, destinat persoanelor care au nevoie sa isi gestioneze anxietatea pentru a se putea odihni. Acest dispozitiv inteligent ne poate monitoriza pulsul si oferi indicatii si informatii referitoare la acesta, pentru a ne ghida catre o stare de liniste.

- sistemul poate fi integrat in orice animalut de plus
- reduce anxietatea utilizatorului si monitorizeaza pulsul
- utilitatea lui consta in integrarea unei monitorizari esentiale intr-un mod familiar, linistitor

Descriere generală

Sistemul va monitoriza pulsul pe baza senzorului hearbeat si va trimite datele catre sistemul Arduino si catre ecranul LCD, avand posibilitatea de a observa datele si de pe telefonul mobil.

Hardware Design

Componente hardware necesare:

- Arduino Uno
- ESP 32
- Senzor de puls
- LCD cu adaptor I2C

Software Design

Descrierea codului aplicației:

- am lucrat in Arduino IDE si am doua programe, unul ruleaza pe Arduino pentru a face legatura cu senzorul de puls iar celalalt pe ESP32 pentru a rula serverul HTTP si a afisa datele pe ecranul LCD
- librării: <LiquidCrystal_I2C.h>
- ESP32 se conecteaza la hotspot-ul meu personal

Rezultate Obținute

Am tras firele pentru senzor prin ursulețul de plus si am pus ecranul LCD in bratele acestuia. Pe site se pot verifica in timp real valorile pulsului, care apar si pe ecranul LCD.

COD ESP32:

```
#include <Arduino.h> #include "WiFi.h" #include "WiFiClient.h" #include "WebServer.h" #include "ESPmDNS.h" #include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x3F, 16, 2); const char *ssid = "Diana"; const char *password = "pisi0000";
```

```
WebServer server(80);
```

```
const unsigned long BAUD_RATE = 9600; const unsigned long MEASUREMENT_DELAY = 1000;
```

```
float heartRate = 0.0; const int heartRateThreshold = 100;
```

```
void handleRoot() {
```

```
char msg[1500];
```

```
snprintf(msg, 1500,
          "<html>\
```

```
<head>\
```

```
<meta http-equiv='refresh' content='4' />\
```

```
<meta name='viewport' content='width=device-width, initial-scale=1'>\
```

```
<link rel='stylesheet'
```

```
href='https://use.fontawesome.com/releases/v5.7.2/css/all.css'
```

```
integrity='sha384-fnm0CqbTlWIlj8LyTjo7m0UStjsKC4p0pQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr' crossorigin='anonymous'>\
```

```
<title>ESP32 Heart Rate Server</title>\
```

```
<style>\
```

```
html { font-family: Arial; display: inline-block; margin: 0px auto;
text-align: center;}\
```

```
h2 { font-size: 3.0rem; }\
```

```
p { font-size: 3.0rem; }\
```

```
.units { font-size: 1.2rem; }\
```

```
.dht-labels{ font-size: 1.5rem; vertical-align:middle; padding-bottom:
15px;}\
</style>\
</head>\
<body>\
  <h2>Home</h2>\
  <p>\
    <i class='fas fa-heart' style='color:#FF0000;'></i>\
    <span class='dht-labels'>Heart Rate</span>\
    <span>%.2f</span>\
  </p>\
</body>\
```

```
</html>", heartRate);
```

```
server.send(200, "text/html", msg);
```

```
}
```

```
void setup() {
```

```
Serial.begin(115200);
lcd.begin();
lcd.backlight();
```

```
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
Serial.println("");
```

```
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
```

```
Serial2.begin(BAUD_RATE, SERIAL_8N1, 16, 17);
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
```

```
if (MDNS.begin("esp32")) {
  Serial.println("MDNS responder started");
}
```

```
server.on("/", handleRoot);
server.begin();
Serial.println("HTTP server started");
```

```
}
```

```
void loop() {  
  
  if (Serial2.available()) {  
    String data = Serial2.readStringUntil('\n');  
    heartRate = data.toFloat();  
    Serial.println("Received heart rate: " + String(heartRate));  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Heart Rate: ");  
    lcd.print(heartRate);  
  }  
  
  server.handleClient();  
  delay(2);  
  
}
```

COD ARDUINO:

```
const int PULSE_SENSOR_PIN = A0; const unsigned long BAUD_RATE = 9600; const unsigned long  
MEASUREMENT_DELAY = 1000;
```

```
void setup() {
```

```
  Serial.begin(BAUD_RATE);
```

```
}
```

```
void loop() {
```

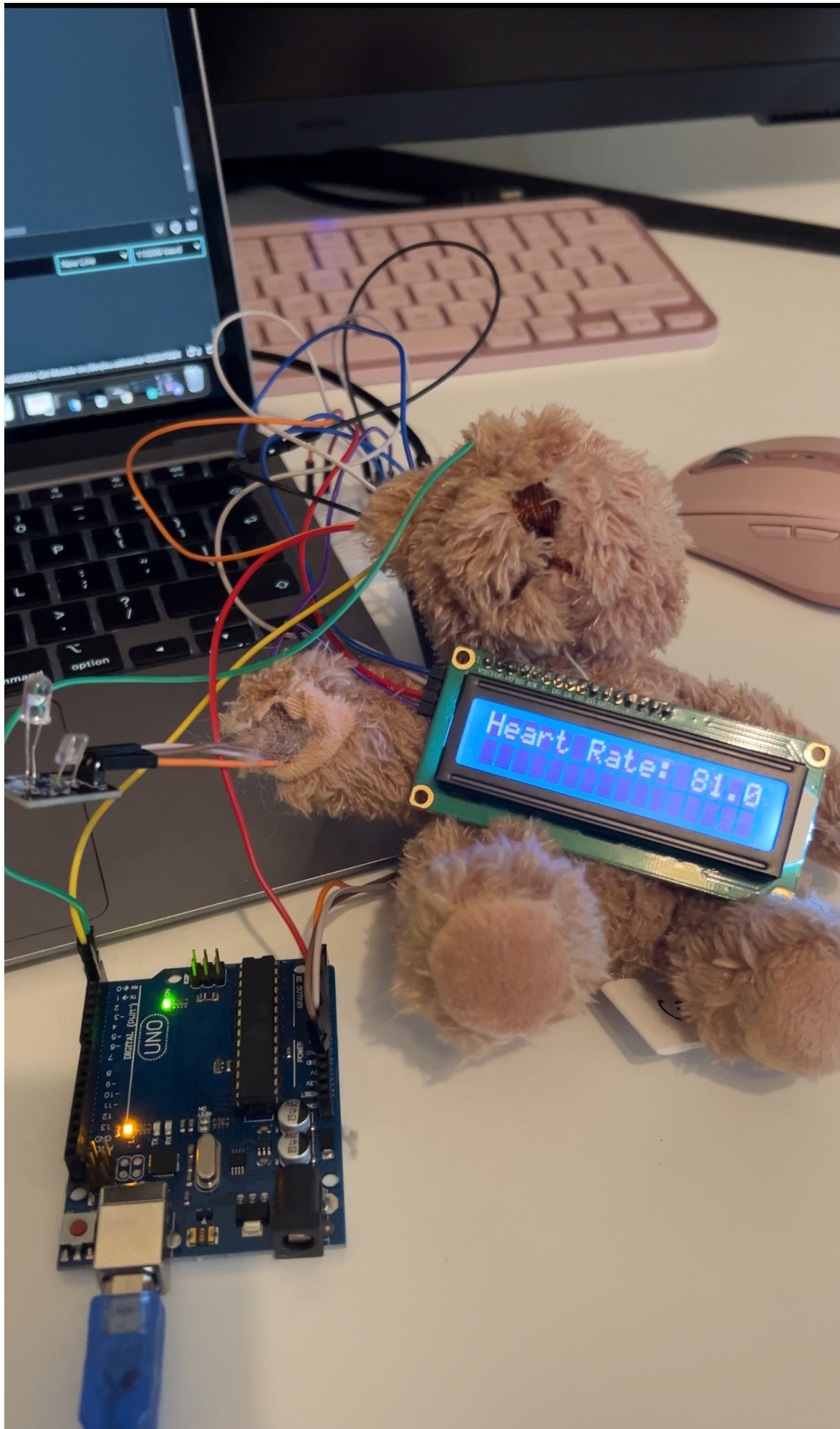
```
  int pulseValue = analogRead(PULSE_SENSOR_PIN);  
  int heartRate = pulseValue / 10;
```

```
  Serial.println(heartRate);
```

```
  delay(MEASUREMENT_DELAY);
```

```
}
```

Media



Home



Heart Rate **96.00**

AA	172.20.10.2	
----	-------------	--

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/rpascale/diana.dervis>



Last update: **2024/05/27 12:48**