

Sport Watch

Autor : **Dinu Eduard - Adiel**

Introducere

Proiectul consta in implementarea unui ceas de tip SportWatch.

Acesta contine integrat un senzor de luat temperatura si unul pentru puls si o alarma pentru sport. Alarma va suna timp de 30 secunde, iar un LED se va aprinde daca senzorul detecteaza o temperatura prea ridicata si un alt LED pentru puls.

Descriere generală

In schema de mai jos se poate observa inputul si outputul proiectului legat la ARDUINO. Exista 2 inputuri : citirea facuta de senzor si timpul configurat de noi.\

Sunt 3 iesiri : sunetul facut de Buzzer cand trebuie sa sune alarma la o anumita ora, LED-ul care se va aprinde cu rosu daca temperatura este in zona de pericol, Ora, minutul si secunda afisata pe ecran.\



Hardware Design

Piese necesare sunt :

1. Arduino
2. 1 Breadboard mare / 2 Breadboard mici
3. 1 Buzzer
4. 1 ecran LCD

5. 1 senzor de temperatura TMP36
6. 1 LED
7. 3 rezistente
8. fire
 9. Potentiometru
 10. Senzor de puls

Ecranul LCD trebuie sa fie suficient de mare pentru a afisa Ora Minutul Secunda.

[dinueduardadiel335caschemaelectrica.pdf](#)

Software Design

In prima faza, proiectul este implementat in tinkercad etapa de software(codarea). Dupa ce testarea proiectului in Tinkercad se termina cu success, se implementeaza fizic.

S-a folosit LiquidCrystal pentru LCD. Functiile folosite sunt : tone, setcursor, print, analogread, digitalWrite. Am definit functii ajutatoare : 1.ConfigClock si ConfigAlarm : seteaza parametrii interni(de la ce ora, minut, secunda incepe ceasul sa afiseze respectiv cand este alarma)
2.toneBuzzer : apeleaza functia tone cu o perioada data parametru, o frecventa si un pin preselectat.
3. temperature: verifica daca senzorul detecteaza o temperatura prea ridicata pentru a face sport, caz in care aprinde LED ul. 4. Check Alarm: verifica daca timpul curent coincide cu alarma, in caz afirmativ porneste buzzer ul timp de 30 secunde.



Cod :

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#include <PulseSensorPlayground.h>
int button;
int h=1;
int m=45;
int s=00;
int flag=1; //PM
int sensePin = A0; //This is the Arduino Pin that will read the sensor
output
int sensorInput; //The variable we will use to store the sensor input
double temp; //The variable we will use to store temperature in degrees.
int alarmMin = 21;
int alarmHour = 19;
int TempLED = 8;
int PulseLED = 10;
```

```
void showPulse(int value) {
  lcd.print("Pulse : ");
  int res = 0;
  value = value / 100;
  if(value <= 0)
    value = 0;
  while(value)
  {
    res = res * 10 + value % 10;
    value /= 10;
  }

  while(res > 0) {
    char c = '0' + res % 10;
    lcd.print(c);
    res /= 10;
  }
  if(res == 0)
    lcd.print("0");
}

void configClock(int hour, int mins, int secs) {
  h = hour;
  m = mins;
  s = secs;
}

void configAlarm(int alarmH, int alarmM) {
  alarmMin = alarmM;
  alarmHour = alarmH;
}

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(7, OUTPUT);
  pinMode(6, INPUT);
  pinMode(TempLED, OUTPUT);
  pinMode(PulseLED, OUTPUT);
  configClock(21,32,50);
  configAlarm(21,33);
}

void temperature() {
  sensorInput = analogRead(A0);
  temp = (double)sensorInput / 1024; //find percentage of input reading
  temp = temp * 5; //multiply by 5V to get voltage
  temp-= 0.5;
  temp = temp * 100; //Convert to degrees
  Serial.println(temp);
  if(temp > 10) {
    digitalWrite(TempLED, HIGH);
  }
  else digitalWrite(TempLED, LOW);
}
```

```
}

void pulse() {
  int  sensorPulse = analogRead(A1);
  Serial.println(sensorPulse);
  showPulse(sensorPulse);
  if(sensorPulse > 750)
  {
    digitalWrite(PulseLED, HIGH);
  }
  else digitalWrite(PulseLED, LOW);
}

void toneBuzzer(int duration) {
  tone(7, 220, duration);
}

void checkAlarm(int min, int hour, int sec) {
  if(min == alarmMin && hour == alarmHour && sec < 30)
    toneBuzzer(1000); //1 sec
}

void loop() {
  temperature();

  lcd.setCursor(0,0);
  lcd.print("Time ");
  if(h<10)lcd.print("0");
  lcd.print(h);
  lcd.print(":");
  if(m<10) lcd.print("0");
  lcd.print(m);
  lcd.print(":");
  if(s<10) lcd.print("0");
  lcd.print(s);
  if(flag==0) lcd.print(" AM");
  if(flag==1) lcd.print(" PM");
  lcd.setCursor(0,1); // for Line 2

  delay(987);
  temperature();
  pulse();
  s=s+1;
  if(s==60){
    s=0;
    m=m+1;
  }
  if(m==60)
  {
    m=0;
    h=h+1;
  }
  if(h==13)
```

```
{  
  h=1;  
  flag=flag+1;  
  
  if(flag==2)flag=0;  
}  
  checkAlarm(m, h, s);  
}
```

Rezultate Obținute

Ecranul LCD afiseaza cu succes timpul cu precizie, LED urile se lumineaza in caz de pericol, iar alarma suna corespunzator.

Concluzii

Consider ca proiectul este important deoarece foloseste in scop didactic cunostinte de intreruperi, senzori.

Jurnal

Main intai am rezolvat partea de afisaj timp.
Apoi am rezolvat partea de alarma.
In final am facut partea de senzor de temperatura si puls, alaturi iluminarea LED urilor.

Bibliografie/Resurse

<https://create.arduino.cc/projecthub/munir03125344286/add-buzzer-to-arduino-bf010b>

<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

<https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>

<https://www.arduino.cc/en/Tutorial/LibraryExamples/LiquidCrystalSetCursor>

<https://www.arduino.cc/reference/en/libraries/liquidcrystal/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/sionescu/sportwatch>



Last update: **2022/06/02 14:19**