

Ceas multifunctional

Autor: [Bianca - Ioana Mașala](#)

Introducere

- Proiectul are ca scop crearea unui ceas multifuncțional ce poate fi utilizat în viața de zi cu zi.
- Pe langa funcțiile generale de afișare a datei, orei, zilei săptămânii, acesta va avea și posibilitatea de a seta alarme, dar și de a vizualiza temperatura.
- Informațiile vor fi afișate pe un ecran. Ceasul va avea posibilitatea de a detecta mișcare și a opri viitoarele alarme automat.

Descriere generală

Ceasul va afișa pe un ecran LCD, controlat cu ajutorul unui potențiomtru toate măsurătorile luate. Pentru a avea un afișaj corect al timpului, vom conecta la placa Arduino Uno un modul RTC. Alarma va avea o melodie prestabilită. Am introdus și un senzor de sunet ce îmi va determina starea ceasului. Temperatura va fi măsurată cu ajutorul unui senzor de temperatură și umiditate, iar în funcție de cât de ridicată sau scăzută este față de limita impusă, va interpreta această informație prin aprinderea fie a ledului albastru, fie a celui roșu. Proiectul poate fi conectat și la o baterie, dar fără a putea seta alarma, la această variantă.

[Schema bloc:](#)



Hardware Design

Componente:

- Arduino Uno
- Modul RTC
- Senzor de sunet
- Ecran LCD
- Senzor de temperatură și umiditate
- Potențiomtru
- Piezo
- Led RGB
- Rezistente

- Breadboard
- Fire tată - mamă
- Fire tată - tată
- Fire mamă - mamă
- Butoane

Schema electrică



Software Design

- Limbaj utilizat: C
- Biblioteci: Serial library, LiquidCrystal, DS3231, DHT
- Mediu de dezvoltare:
 - Schema bloc: draw.io
 - Schema Electrică: Tinkercad
 - Software: Arduino IDE

Structura codului

Codul este scris în Arduino IDE și interpretează datele primite de pe plăcuța Arduino. Pe un ecran LCD îmi afișez toate datele de care am nevoie. Pentru a naviga mai ușor între paginile LCD-ului și a modifica anumite date, mă folosesc de 4 butoane ce funcționează după cum urmează:

Butonul 1: → face posibilă navigarea între pagini, pentru a afișa diverse informații;

Butonul 2: → plasează cursorul pentru a modifica data sau ora;

Butonul 3: → incrementează valorile pentru a fi modificate;

Butonul 4: → determină formatul orei

Codul este structurat cu ajutorul mai multor funcții:

- setup() - setez pinii de INPUT/OUTPUT;
- fbuttonState() - contorizez apăsarea butonului;
- isButtonPushed() - verific dacă a fost apăsat vreun buton;
- setAlarm() - citește caractere date în Serial Monitor și setează alarma;
- play() - redă o melodie;

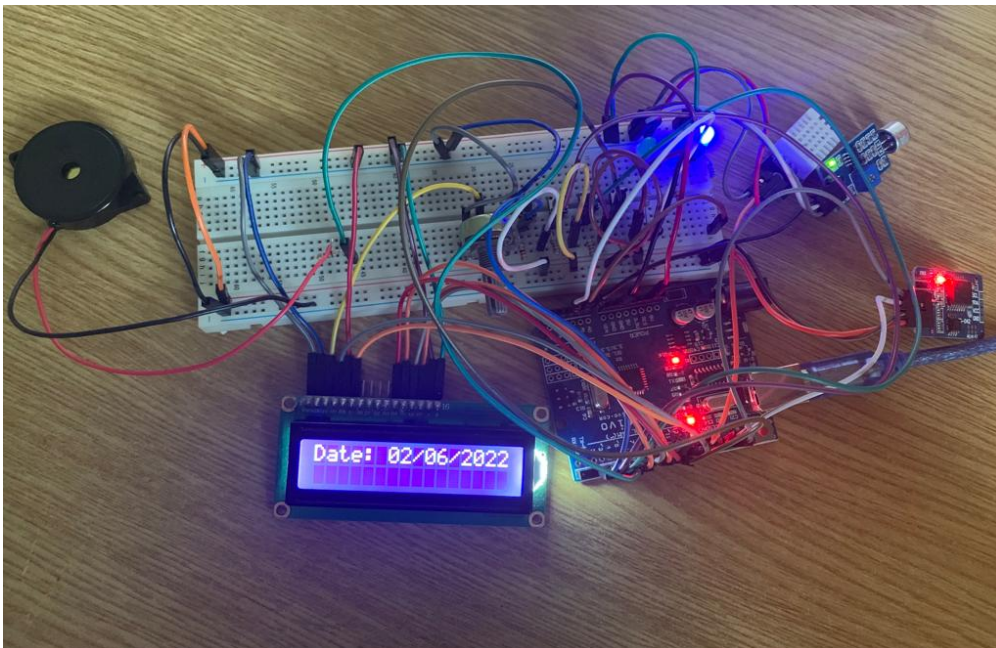
- loop() - citesc datele transmise de senzori și în funcție de pagina pe care mă aflu fac modificările necesare. Ecranul se repornește constant atunci când detectează zgomot.

Alarma

- Alarma a fost inițial gândită pentru a fi setată din telefon, doar că datorită problemelor întâmpinate de la modulul Bluetooth aceasta este setată din Serial Monitor. Acolo trimit constant caractere corespunzătoare orelor, minutelor și secundelor, iar atunci când timpul setat corespunde cu ora care este piezo-ul va porni melodia predefinită și se va aprinde led-ul RGB.

Rezultate Obținute

Proiect:



Demo:

<https://youtu.be/aKvPsj2Aelc>

Concluzii

Aplicația a fost una foarte interesantă de implementat. Parte de legare a componentelor a fost una mai dificilă, dar și puțin din partea de implementare a codului, deoarece la un moment dat mă pierdusem în toate variabilele avute. Per total proiectul consider că este unul chiar util și îmi doresc să lucrez în continuare la el, pentru a reuși să conectez proiectul la aplicația din telefon pe care am realizat-o în AppInventor, dar și să conectez modulul Pir, atunci când acesta îmi detectează mișcare, toate alarmele curente să fie oprite.

Download

Arhiva proiectului poate fi descărcată de aici: [ceasmultifunctional.zip](#)

Jurnal

- 21 Aprilie - alegere temă proiect
- 5 mai - prima comandă de piese
- 13 mai - primul Milestone
- 27 mai - a doua comandă de piese
- 29 mai - finalizare

Bibliografie/Resurse

<https://www.youtube.com/watch?v=nm-lrZTYoew>

<https://ocw.cs.pub.ro/courses/pm/lab/lab2-2022>

<https://github.com/robsoncouto/arduino-songs>

https://github.com/fariha6412/Digital-Clock-in-arduino-with-alarm-timer-and-stopwatch/blob/main/digital_clock_with_alarm_timer_and_stopwatch.ino

<https://howtomechatronics.com/tutorials/arduino/arduino-ds3231-real-time-clock-tutorial/>

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody/>

<http://www.rinkydinkelectronics.com/library.php?id=73>

<https://create.arduino.cc/projecthub/karmette/basic-led-setup-for-beginners-0a124a>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2022/arosca/ceasmultifunctional>



Last update: **2022/06/02 05:21**