

Ceas inteligent cu alarmă și senzori de mediu

Introducere

Proiectul meu constă în realizarea unui **ceas electronic multifuncțional** care îmi va oferi informații despre ora curentă, data calendaristică, temperatura mediului înconjurător și funcționalitate de alarmă.

- **Ce face:** Afișează ora, data, temperatura, permite setarea orei/datei, a formatului de afișare 12/24 ore și a unei alarme programabile, care alege random dintr-o serie de melodii cum să sune.
- **Scop:** Crearea unui dispozitiv funcțional, independent, care combină elemente hardware și software pentru a oferi funcționalitățile unui ceas digital modern, demonstrând în același timp capacitatea de integrare a diferitelor tehnologii și protocoale de comunicație.
- **Ideea de pornire:** Totul a început din nevoia de a avea un ceas digital în cameră, dar cu funcționalități extinse.
- **Utilitate:** Acest proiect rezolvă problema dependenței de telefon pentru verificarea orei și a temperaturii. Mulți dintre noi ne uităm la telefon înainte de culcare și dimineața, ceea ce poate afecta somnul și concentrarea. Ceasul meu oferă aceste informații esențiale fără distragerea atenției și tentația de a verifica notificările sau social media. În plus, funcția de alarmă personalizabilă îmbunătățește rutina zilnică, fiind adaptată nevoilor specifice ale utilizatorului, spre deosebire de soluțiile comerciale care oferă opțiuni limitate.

Descriere generală

Funcționalități:

1. Afișarea și setarea orei
 1. Ora curentă este obținută de la modulul RTC și afișată pe ecranul LCD.
 2. Formatul orei poate fi setat în mod 12 ore (AM/PM) sau 24 ore prin intermediul butoanelor.
 3. Setarea orei se face prin intermediul celor trei butoane.
2. Afișarea datei și a temperaturii
 1. Data calendaristică este obținută de la modulul RTC și afișată pe ecranul LCD.
 2. Temperatura este citită de la senzorul DHT22 și afișată pe ecranul LCD.
 3. Actualizarea informațiilor se face automat la intervale regulate de timp.
3. Sistem de alarmă programabil
 1. Setarea alarmei se face prin interfața cu butoane, permițând configurarea orei exacte.
 2. Semnalul sonor este generat de buzzer atunci când ora curentă coincide cu ora alarmei.
4. Interfață cu utilizatorul
 1. Navigare prin meniuri folosind butoanele pentru setarea diverselor parametri.
 2. Feedback vizual prin intermediul LED-urilor pentru confirmarea acțiunilor.

Laboratoare folosite:

- **Laboratorul 0 (GPIO)** - utilizat pentru configurarea butoanelor și LED-urilor, formând interfața fizică prin care utilizatorul interacționează cu ceasul pentru setarea orei și alarmei.
- **Laboratorul 2 (Întrepereri)** - implementate pentru detectarea apăsărilor butoanelor cu funcționalitate de debounce, asigurând o citire precisă a comenzilor utilizatorului și pentru declanșarea alarmei la momentul programat.
- **Laboratorul 4 (ADC)** - folosit pentru citirea valorilor de la senzorul de temperatură și convertirea lor în valori de temperatură afișabile pe ecran.
- **Laboratorul 6 (I2C)** - utilizat pentru comunicarea cu modulul RTC și cu ecranul LCD I2C, permițând transferul eficient al datelor.

Module Hardware:

1. Unitate de procesare - Arduino UNO bazat pe microcontrollerul ATmega328p
2. Afișare - Display LCD cu interfață I2C pentru afișarea informațiilor
3. Cronometrare precisă - Modul RTC (Real-Time Clock) DS1307 pentru menținerea timpului
4. Sensori - Senzor de temperatură DHT22 pentru măsurarea parametrilor de mediu
5. Interfață utilizator - Trei butoane pentru setarea orei, datei și controlul alarmei
6. Feedback - Buzzer pentru alarmă și LED-uri pentru indicatori vizuali
7. Alimentare - Baterie/acumulator pentru funcționare independentă

Module Software:

1. Control display - Gestionarea afișării informațiilor pe LCD
2. Procesare timp - Citirea și procesarea datelor de la modulul RTC
3. Interfață utilizator - Interpretarea intrărilor de la butoane și actualizarea stării sistemului
4. Managementul temperaturii - Citirea și procesarea datelor de la senzorul de temperatură
5. Sistem alarmă - Gestionarea alarmei programabile
6. Management energetic - Optimizarea consumului de energie

Interacțiunea între module se realizează astfel:

- Microcontrollerul comunică cu RTC și LCD-ul prin protocolul I2C
- Senzorii și butoanele sunt interfațate prin pini
- Modulul de procesare timp actualizează periodic informațiile afișate pe display
- Interfața utilizator interpretează apăsările butoanelor și modifică setările
- Sistemul de alarmă verifică continuu ora curentă și declanșează alarma la momentul programat

Schema bloc:



Hardware Design

Listă de componente:

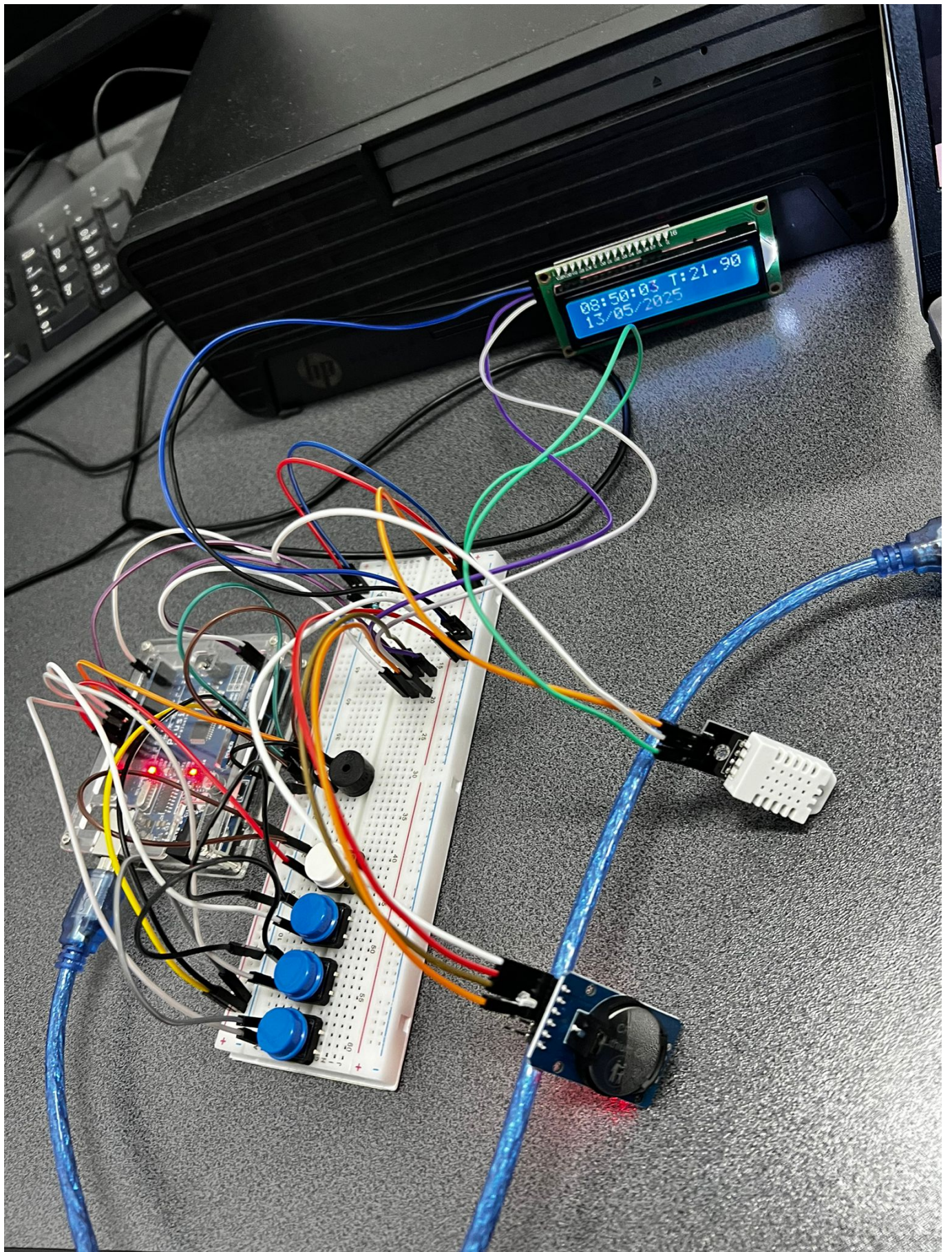
- Arduino UNO - Unitatea centrală de procesare
- Display LCD I2C (16x2 caractere) - Afișarea informațiilor
- Modul RTC - Menținerea timpului precis
- Senzor de temperatură - Măsurarea temperaturii ambientale
- Buzzer Activ - Sonerie pentru alarmă

- Butoane tactile - Interfața utilizator pentru setări
- Breadboard
- Fire de conexiune - Realizarea conexiunilor
- Rezistențe 330Ω - Limitare curent pentru LED-uri
- Rezistențe 4.7kΩ - Pull-up pentru butoane
- LED-uri cu catod comun (verde, roșu) - Indicatori vizuali

Schema electrica realizata in fusion

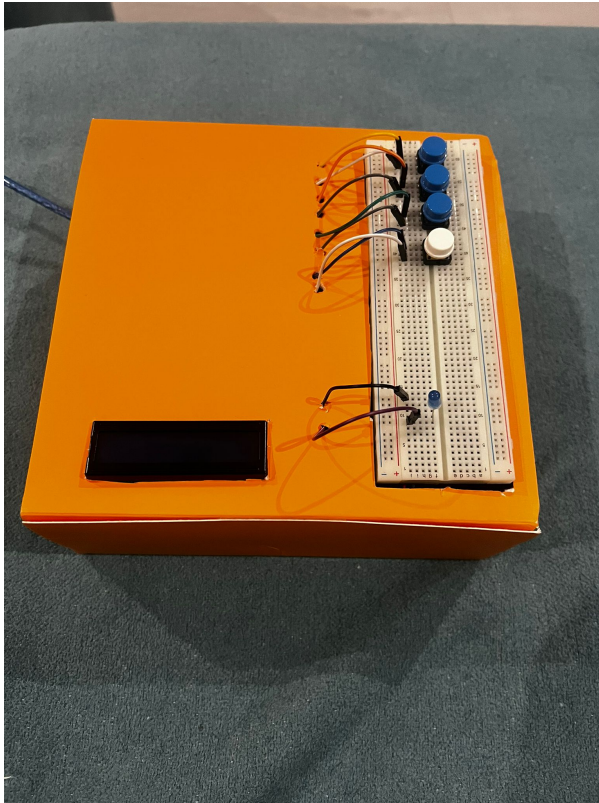


Componentele hardware conectate



!!! In cadrul schemei am folosit o baterie pentru alimentare, iar proiectul momentan este alimentat prin intermediul laptopului.

Proiectul cu tot cu carcasa



Software Design

Mediu de dezvoltare utilizat Aplicația este scrisă în Arduino C++. Mediul de dezvoltare folosit este Arduino IDE.

Librării și surse third-party utilizate Aplicația folosește mai multe librării externe:

1. Wire.h - Comunicație I2C. Folosită pentru a comunica cu modulul RTC și LCD.
2. RTCLib.h - Parte din Adafruit RTCLib, pentru lucrul cu modulul RTC DS3231 și permite obținerea și setarea timpului.
3. LiquidCrystal_I2C.h - Pentru afișaj LCD 16x2 cu interfață I2C și permite afișarea textului fără control direct al pinilor LCD.
4. DHT.h - Pentru lucrul cu senzorul DHT22 (temperatură și umiditate).

Algoritmi și structuri implementate

1. **Afișare oră, dată, temperatură**
 1. Se utilizează modulul RTC (`rtc.now()`).
 2. Se afișează în format 12h sau 24h, configurabil.
 3. Afișare temperatură obținută de la DHT22.
2. **Moduri de operare prin buton multifuncțional (`buttonModeAlarmPin`)** - Toggle între moduri:
 1. Modul normal
 2. Modul de setare alarmă
 3. Modul de setare dată
 4. Comutare între 12h și 24h
3. **Setarea orei și alarmei cu 3 butoane separate:**

1. buttonHourPin – modifică ora
2. buttonMinutePin – modifică minutul
3. buttonSecondPin – modifică secunda
4. **Verificare alarmă (checkAlarm())**
 1. Compară ora actuală cu valorile setate.
 2. Dacă sunt egale și alarma este activată → se pornește o melodie și un LED.
5. **Alarma**
 1. Redă o melodie aleatorie din 3 variante (Twinkle Star, Happy Birthday, Jingle Bells).
 2. Se folosește tone() pe pinul buzzer.
 3. LED-ul pentru alarmă clipește sincron cu notele.
 4. Se poate opri manual cu un buton.
6. **Setarea datei**
 - Se pot incrementa ziua, luna și anul, ținând cont de numărul de zile în lună și anii bisecți.
7. **Anti-rebound pentru butoane**
 - Se folosește un algoritm simplu de debounce bazat pe millis() și un delay de 200ms.

STRUCTURA GENERALĂ A CODULUI

Firmware-ul este structurat astfel:

- Inițializare componente hardware și variabile globale
- Loop principal (loop()): gestionează modurile de funcționare
- Funcții de afișare (LCD)
- Funcții pentru setarea timpului/datei/alarmei
- Funcții pentru alarmă (sunet și LED)
- Funcții auxiliare (conversii, formate, ani bisecți etc.)

Rezultate Obținute

În urma implementării proiectului, am obținut un sistem funcțional care îndeplinește următoarele cerințe:

- Afișarea corectă a orei, datei și temperaturii pe un ecran LCD 16×2 cu interfață I2C;
- Comutare între format 12h și 24h pentru ora afișată;
- Setare manuală a datei și a alarmei folosind 4 butoane;
- Alarmă activă cu melodii aleatorii redată printr-un buzzer și semnalizată vizual printr-un LED;
- Funcție de oprire a alarmei prin apăsarea unui buton dedicat;
- Feedback vizual pe ecran pentru fiecare mod de funcționare, inclusiv modul de setare a alarmei și datei;
- Implementarea corectă a debounce-ului software pentru a preveni multiple declanșări false ale butoanelor;
- Trecerea automată între moduri (afișare, setare alarmă, setare dată) printr-un buton multifuncțional.

Link videoclip youtube:

<https://youtube.com/shorts/8U7rjtlfug?si=eXz-076XvNCoucf>

(In cadrul acestui videoclip, proiectul nu se afla la varianta lui finala, intre timp am modificat modalitatea de a suna a buzzerului. I-am adăugat acestuia posibilitatea de a suna cu una din trei melodii alese de mine pe frecvente, frecvente care se transmit și către led astfel încât să clipească concomitent cu ritmul melodiei.)

Concluzii

Download

Contul de git cu restul resurselor proiectului:

<https://github.com/biancascirtocea/Ceas-electronic-multifunctional>

Jurnal

□ Configurare inițială Am început cu conectarea componentelor principale: modulul RTC DS3231, senzorul de temperatură DHT22, LCD-ul I2C și butoanele. Am întâmpinat dificultăți în conectarea firelor pe breadboard – unele jumpere nu făceau contact corect, iar alimentarea era instabilă. A fost nevoie să schimb mai multe fire pentru a obține conexiuni stabile.

□ Probleme hardware neașteptate Am ars un LED (care semnală activarea alarmei) deoarece l-am conectat fără rezistență de limitare. A fost o lecție bună privind protejarea componentelor și calculul corect al curentului. Am înlocuit componenta și am inclus un rezistor de 220Ω.

□ Debugging RTC Inițial, ora nu se seta corect. După mai multe încercări de a regla ora din cod, am descoperit că bateria de backup de pe modulul RTC era descărcată. După ce am înlocuit-o, modulul a început să rețină corect ora și data, chiar și după resetări.

□ Implementare alarmă și testarea sunetului Am integrat partea de alarmă și redare melodii. A fost o provocare sincronizarea buzzerului cu LED-ul și testarea fără a deranja vecinii. De asemenea, m-am asigurat că oprirea alarmei funcționează fiabil – adăugând debounce corect pentru buton.

□ Finalizare și testare completă Am testat toate scenariile: schimbarea orei, comutare 12h/24h, afișare temperatură, setarea și declanșarea alarmei. Toate funcțiile au fost verificate și funcționează stabil. Am făcut și un mic stres-test, simulând resetări și reporniri pentru a verifica persistența funcționării RTC-ului.

Bibliografie/Resurse

- <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- <https://docs.arduino.cc/hardware/uno-rev3/>
- <https://github.com/adafruit/RTClib>
- <https://docs.arduino.cc/libraries/liquidcrystal-i2c/>
- <https://github.com/adafruit/DHT-sensor-library>
- <https://projecthub.arduino.cc/tmekinyan/playing-popular-songs-with-arduino-and-a-buzzer-546f4a>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/rnedelcu/bianca.scirtocea>



Last update: **2025/05/27 20:53**