

# Smart desk clock

## Introducere

Proiectul constă în realizarea unui ceas de birou inteligent bazat pe microcontrolerul ATmega328p (placă compatibilă Arduino UNO). Acest dispozitiv afișează ora curentă, temperatura și umiditatea mediului ambiant, toate prezentate pe un ecran LCD cu interfață I2C. De asemenea, ceasul este capabil să emită alarme sonore prin intermediul unui buzzer piezo, în funcție de condiții presetate sau ora programată.

Scopul principal al proiectului este de a crea un dispozitiv accesibil, funcțional și ușor de personalizat, care să îmbine mai multe funcții utile într-un format compact. Am pornit de la ideea de a îmbunătăți ceasurile clasice, adăugând funcții de monitorizare a mediului și posibilitatea alertelor sonore, fără a depinde de componente scumpe sau complexe.

Consider că acest proiect este util atât pentru mine, ca mod de aprofundare a cunoștințelor în domeniul sistemelor embedded, cât și pentru alți utilizatori care doresc un ceas inteligent personalizat, ideal pentru birou, casă sau chiar ca bază pentru proiecte mai avansate precum sisteme de automatizare sau stații meteo

## Descriere generală

Voi folosi microcontrolerul Arduino UNO (ATmega328p) care procesează datele de la senzori, controlează afișajul și buzzerul. LCD 1602 I2C se folosește pentru a afișa ora, temperatura și umiditatea, primește date de la Arduino prin protocol I2C. DHT22 este senzorul de temperatură și umiditate, trimite date către Arduino pentru a fi afișate. Folosesc buzzer-ul pentru semnal sonor (alarmă sau avertizare) și este controlat de Arduino cu funcția `tone()`. Butonul (opțional) poate seta ora sau declanșa manual alarma și este citit de Arduino ca intrare digitală Alimentare va fi prin USB/Baterie.

Fluxul de date:

1. Senzorul DHT22 măsoară temperatura și umiditatea.
2. Arduino preia aceste date și:
  - Le afișează pe LCD.
  - Activează buzzerul dacă sunt depășite praguri prestabilite (ex:  $>30^{\circ}\text{C}$ ).
1. Butonul poate fi folosit pentru a seta ora sau porni o alarmă manual.
2. Sistemul este alimentat fie prin USB, fie cu o baterie de 9V prin mufă DC.



# Hardware Design

Piese:

- DHT22
- LCD 1602 I2C
- Buzzer
- Buton (opțional)
- Arduino UNO (ATmega328p)
- Alimentare (USB/Baterie)

## Modul în care sunt conectate modulele la microcontroller-ul ATmega328P

### LCD 1602 cu interfață I2C

A fost conectat la pinii SDA și SCL ai microcontroller-ului, care corespund cu A4 (SDA) și A5 (SCL) pe placa Arduino UNO. Alimentarea display-ului este realizată prin conectarea pinului VCC la +5V și GND la GND. Comunicarea se face prin protocol I2C, ceea ce reduce semnificativ numărul de fire necesare comparativ cu conexiunea paralelă.

### Senzorul de temperatură și umiditate DHT22

Pinul de date este conectat la unul dintre pinii digitali ai ATmega328P, de exemplu PD2 (D2). Pentru funcționarea corectă, este necesară o rezistență de pull-up de 10kΩ între pinul de date și VCC (așa cum este indicat în datasheet-ul senzorului). Alimentarea senzorului este asigurată prin conectarea pinului VCC la 5V și GND la masă.

### Buzzer pasiv

Acesta este conectat la unul dintre pinii digitali de pe ATmega328P, de exemplu PD12 (D12) și la GND.

### Alimentare

Toate componentele sunt alimentate la +5V, care poate proveni fie de la un cablu USB conectat la Arduino, fie de la o sursă externă (ex. baterie de 9V cu regulator onboard).

## Comunicare pe I2C

Pentru comunicarea între microcontroller și perifericele I2C (precum LCD-ul 1602 cu interfață I2C și, opțional, alți senzori compatibili), am decis să folosesc magistrala I2C/TWI integrată în ATmega328P. Protocolul I2C este ideal în acest context deoarece permite conectarea mai multor dispozitive pe aceleași două fire de semnal, reducând semnificativ complexitatea cablajului.

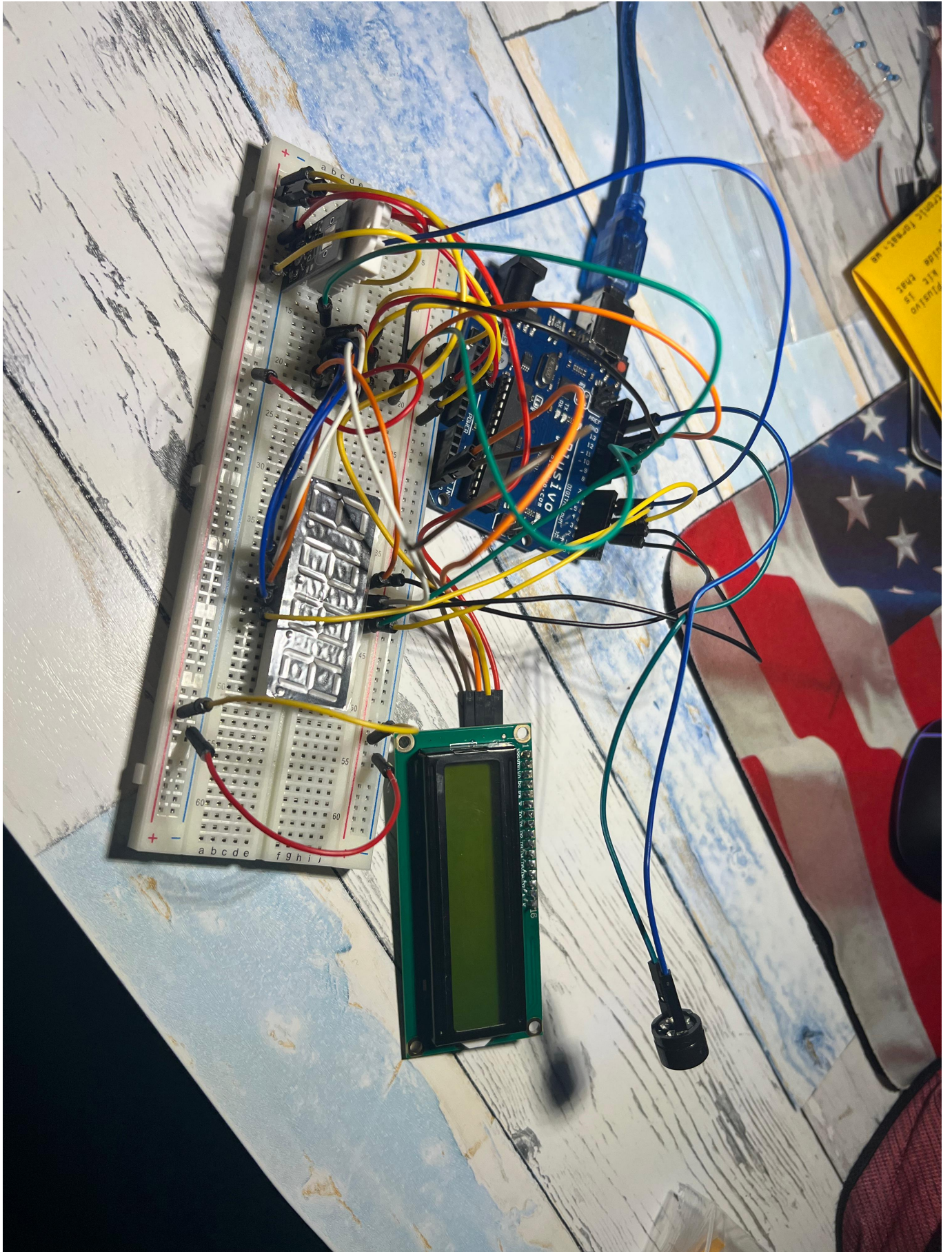
Am realizat conexiunile astfel: SDA (Date) → conectat la pinul A4 al ATmega328P. SCL (Clock) → conectat la pinul A5. VCC → alimentat la +5V, întrucât modulul LCD suportă această tensiune. GND → comun cu restul circuitului.

## Ilustrație făcută pe TinkerCad



## Ilustrație luată din gidul Plusivo





# Software Design

Aplicația constă în implementarea unui ceas digital inteligent cu afișare pe LCD și display 7-segmente controlat prin shift register (74HC595), alături de măsurarea temperaturii și umidității folosind senzorul DHT22 și de redarea unei melodii la fiecare oră fixă.

Componentele controlate de firmware:

- Afișarea timpului curent (HH:MM:SS) pe LCD 16×2.
- Afișarea unei animații numerice 1-2-3-4 pe display 7-segmente sincronizat cu redarea melodiei.
- Citirea temperaturii și umidității de la senzorul DHT22.
- Trimiterea datelor către monitorul serial prin USART implementat manual.
- Redarea unei melodii ("Twinkle, Twinkle, Little Star") printr-un buzzer la ora fixă.

Actualizarea timpului se face printr-un timer hardware (Timer1) configurat în mod CTC care declanșează întreruperi la fiecare 1 secundă. Afișarea se face ciclic în bucla principală, iar funcțiile sunt bine separate pentru lizibilitate și întreținere.

## 2. Mediu de dezvoltare

- PlatformIO în Visual Studio Code (cu platform = atmelavr)
- Placă de dezvoltare: ATmega328P compatibil Arduino UNO R3
- Upload via USB, monitorizare serială prin USART (implementată manual)

## 3. Librării utilizate

- LiquidCrystal\_I2C - pentru afișajul LCD 16×2
- Adafruit DHT sensor library - pentru senzorul DHT22
- Adafruit Unified Sensor - suport general pentru senzori
- pitches.h - pentru definirea notelor muzicale (buzzer)

## 4. Algoritmi și structuri implementate

- Multiplexare pentru 7-segmente

Se activează secvențial fiecare cifră (DIG1-DIG4) cu semnal LOW, în timp ce se trimite către 74HC595 codul binar corespunzător cifrei dorite. Codificarea segmentelor este adaptată pentru cazul în care segmentul A este conectat la Q7 și DP (nefolosit) este pe Q0.

- Menținerea timpului software cu Timer1 CTC + ISR:

Timer1 este configurat la 1Hz (OCR1A = 15624 cu prescaler 1024) și incrementează seconds, minutes, hours. La fiecare oră fixă, melodyShouldPlay este setat pe true.

- Citirea senzorului DHT22

Se citește temperatura și umiditatea la fiecare 2 secunde. Se validează datele (isnan()) și se afișează pe LCD, sau se afișează un mesaj de eroare.

- Redare melodie pe buzzer + secvență vizuală

Buzzerul redă o melodie cu durate variabile pentru fiecare notă. La fiecare 500ms, se schimbă cifra

activă, afișând 1→2→3→4 sincronizat vizual.

- Comunicație serială USART implementată direct:

Se configurează registrele UBRR/UCSR0A/B/C pentru baudrate 9600. Se trimit mesaje text pentru ora curentă și sensor prin USART\_sendString().

#### 5. Surse și funcții implementate Funcție Descriere

- void shiftOutFast() = Trimite rapid un octet spre 74HC595 folosind porturi directe.
- void sendTo74HC595() = Trimite date spre segment display prin latch.
- void deactivateAllDigits() = Dezactivează toate cele 4 cifre.
- void USART\_init() = Configurează USART pentru 9600 baud
- void USART\_sendChar() = Trimite un caracter prin USART
- void USART\_sendString() = Trimite un șir de caractere
- void updateTemperatureAndHumidity() = Citește sensorul și afișează pe LCD și USART
- void updateDigitDisplay() = Comută între cifrele 1-2-3-4 la fiecare 500ms
- void playMelody() = Redă melodia notă cu notă și oprește după final
- ISR(TIMER1\_COMPA\_vect) = Actualizează timpul o dată pe secundă și declanșează melodia la fix
- void loop() = Afișează ora, actualizează sensorul, gestionează melodia și afișajul

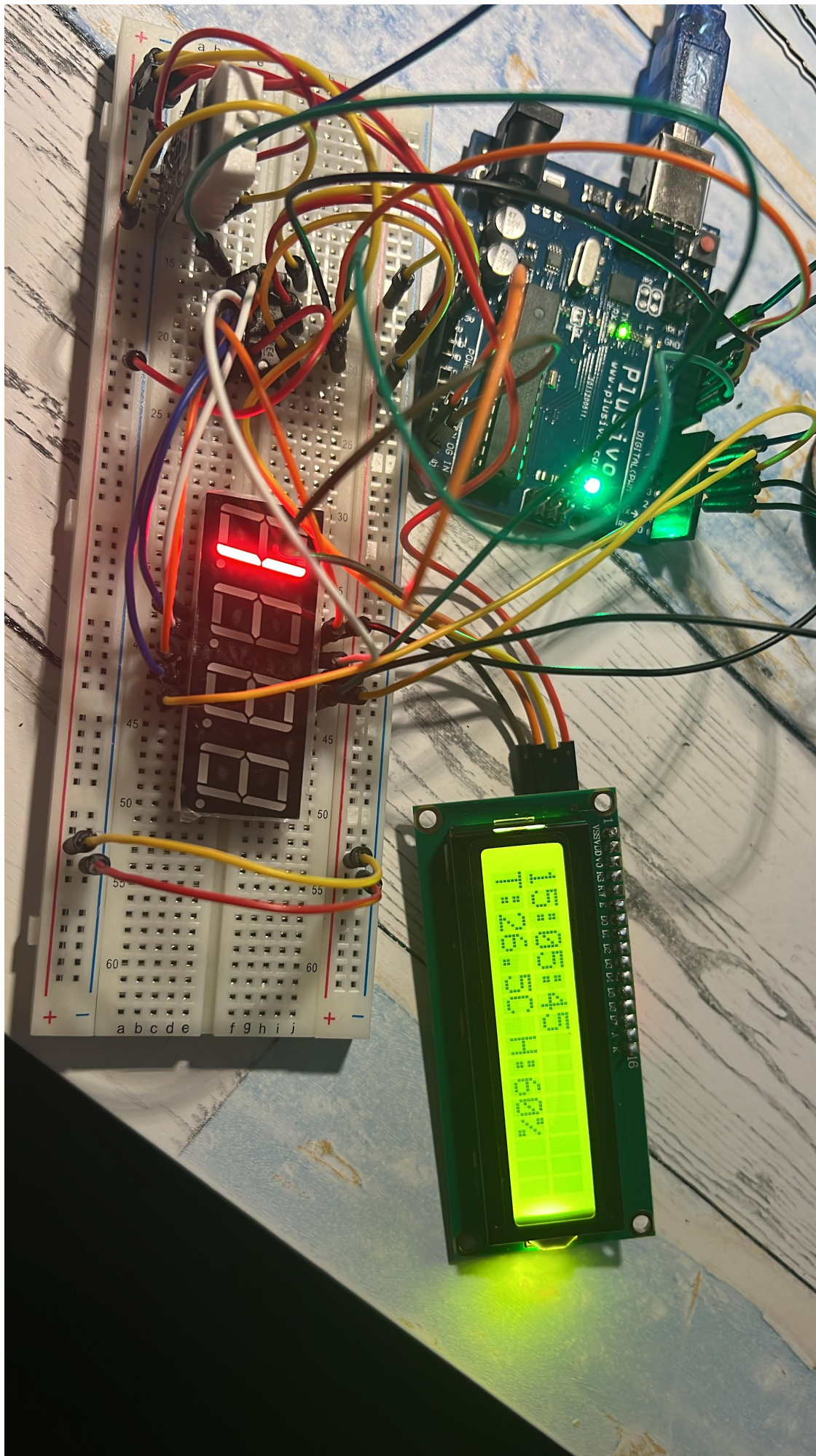
## Rezultate Obținute

### Serial Monitor



### Real life





## Videoclip

<https://youtube.com/shorts/hlM-PZVYdmU?feature=share>

## Concluzii

În concluzie a fost foarte captivant să lucrez la acest proiect, a trebuit să învăț rapid cum să mă descurc, cum să-mi fac viața mai ușoară și codul mai frumos.

## Download

[smartclock\\_mocanualexia\\_333cd.zip](#)

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

[Export to PDF](#)

<https://cdn.sparkfun.com/assets/f/7/d/9/c/DHT22.pdf>

[https://www.handsontec.com/dataspecs/module/I2C\\_1602\\_LCD.pdf](https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf)

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/125741>



Last update: **2025/05/30 08:29**

