

Book Mood Lamp

Introducere

Proiectul consta intr-o lampa inteligenta pentru citit, realizata cu ajutorul unei placi Arduino Uno. Lampa va avea forma de carte deschisa, iluminata din interior cu LED-uri RGB care vor crea o atmosfera relaxanta pentru citit.

Pe langa iluminare, aceasta va include mai multe functionalitati precum :

- schimbarea culorii si a intensitatii luminii prin buton, respectiv potentiometru
- afisarea timpului petrecut citind pe LCD
- alarma pentru pauze
- masurarea temperaturii si umiditatii in camera
- redarea de sunete ambientale prin buzzer

Scopul proiectului este de a imbunatati experienta de citit si de a crea un spatiu relaxant si personalizat pentru cititor.

Ideea proiectului a pornit de la pasiunea pentru citit. Imi doresc sa realizez un obiect care sa redea o atmosfera cat mai cozy si sa-ti dea un imbold sa citesti ca mod de relaxare in timpul liber.

Consider ca proiectul este util deoarece ofera o lumina ambientala si confortabila pentru citit, ajuta utilizatorul sa-si organizeze timpul petrecut citind si poate contribui la reducerea oboselii ochilor prin ajustarea intensitatii luminii.

Descriere generală



Proiectul este construit pe baza unei placute Arduino Uno, care reprezinta unitatea centrala de control si coordoneaza toate modulele hardware si software.

Iluminarea ambientala este realizata cu ajutorul unor LED-uri RGB, controlate prin semnale PWM, ceea ce permite modificarea culorii si intensitatii. Intensitatea LED-urilor este reglata folosind un potentiometru, conectat pe un pin analogic si citit prin intermediul modulului ADC al microcontrolerului.

Pentru afisarea informatiilor, proiectul foloseste un LCD conectat prin protocolul I2C, ceea ce reduce numarul de pini necesari. Pe acesta o sa se afiseze timpul petrecut citind, temperatura, umiditatea si alarma. Cu ajutorul senzorului DHT11 masor temperatura si umiditatea din camera si transmit datele catre Arduino prin comunicatie digitala de tip GPIO. Aceste valori o sa fie afisate periodic pe LCD.

Interactiunea utilizatorului cu sistemul se realizeaza cu ajutorul a patru butoane conectate pe pini digitali. Primul buton este folosit pentru pornirea si oprirea sesiunii de citit, al doilea pentru schimbarea culorii LED-urilor, al treilea pentru setarea alarmei care la terminarea timpului va activa un buzzer pasiv, iar al patrulea pentru activarea buzzerului pentru muzica ambientala.

Buzzerul este controlat prin PWM si timere, iar sunetele sunt redade prin modificarea frecventei semnalului.

Din punct de vedere software, proiectul e impartit in mai multe module: modulul principal de control, cel de iluminare RGB, cel de citire a senzorilor, de afisare pe LCD, de gestionare a alarmei si de generare a sunetelor.

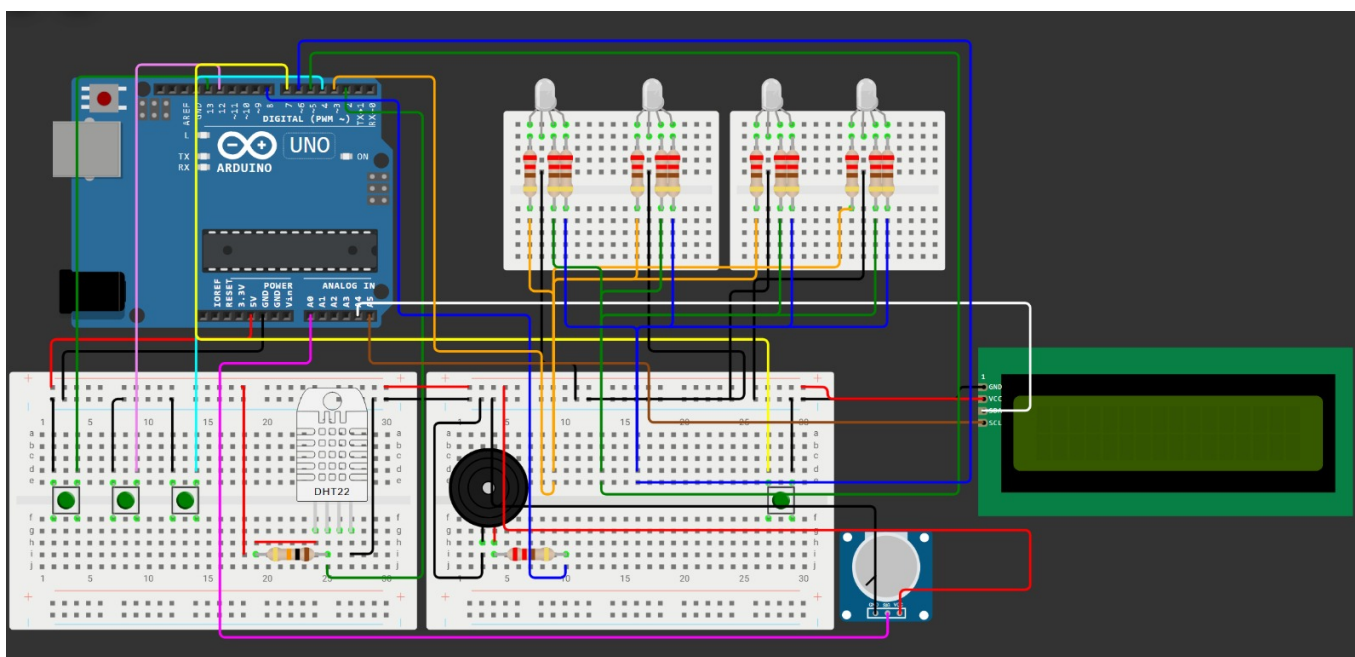
Hardware Design

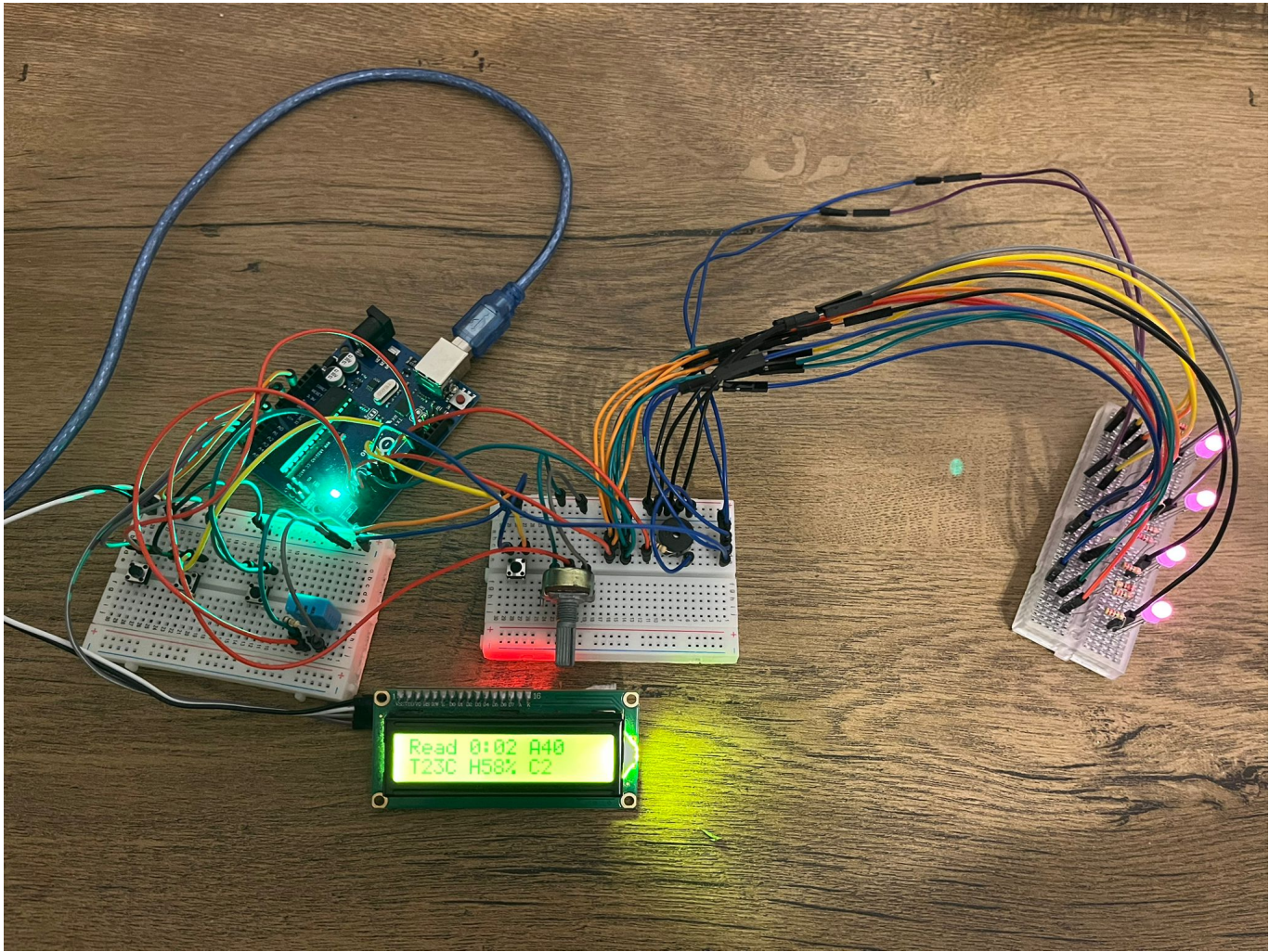
Lista componente:

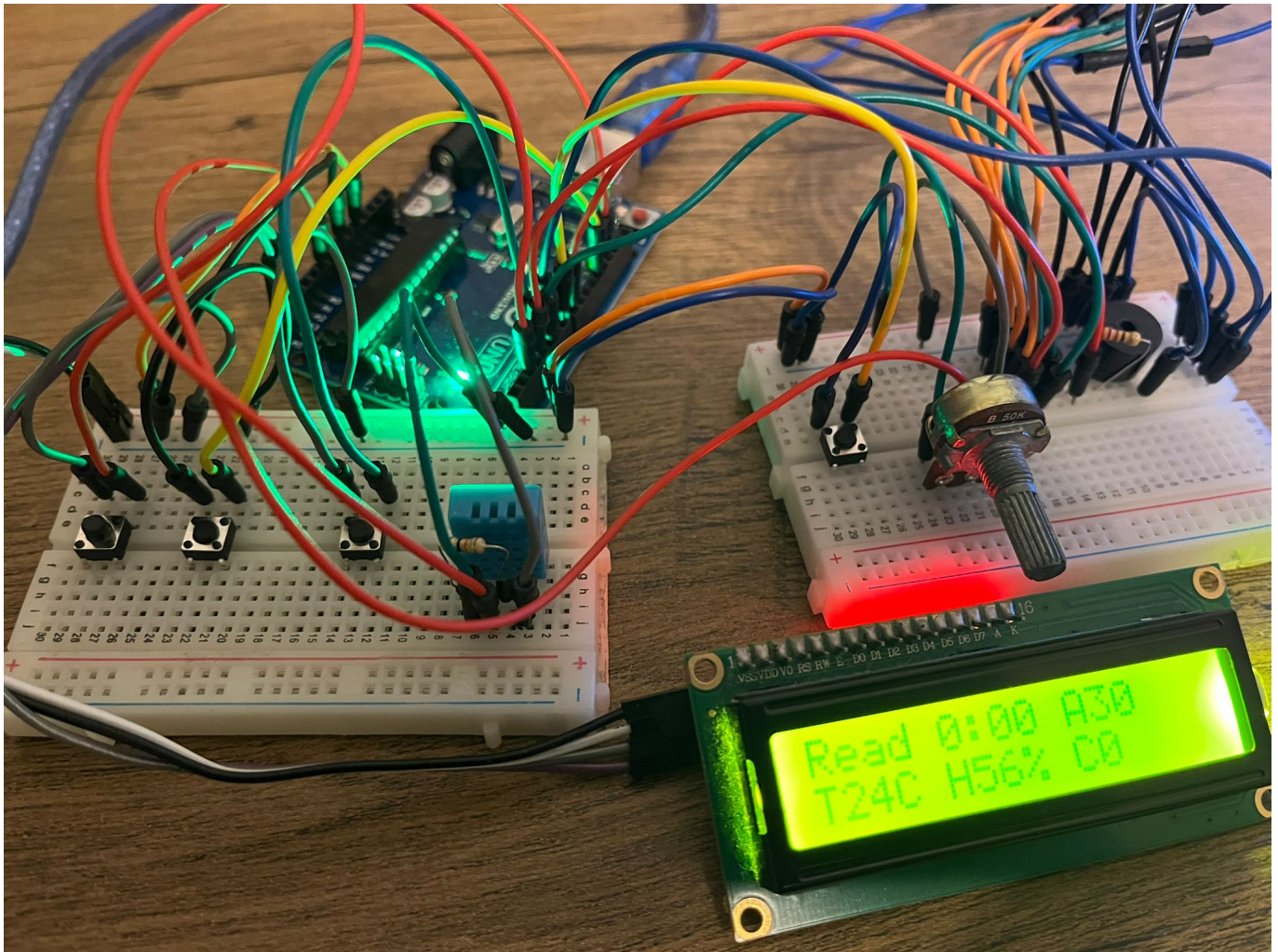
- Arduino Uno
- LCD + I2C
- LED-uri RGB
- 4 butoane
- potentiometru
- DHT11
- Buzzer pasiv
- rezistente
- breadboard
- fire

Componenta	Pin Componenta	Pin Arduino	Rol
LCD I2C	VCC	5V	Alimentare modul LCD
LCD I2C	GND	GND	GND-ul comun al circuitului
LCD I2C	SDA	A4	Linie Serial Data pentru protocolul I2C
LCD I2C	SCL	A5	Linie Serial Clock pentru protocolul I2C
LED-uri RGB	RED	D5	Pin PWM folosit pentru controlul intensitatii culorii rosii
LED-uri RGB	GREEN	D6	Pin PWM folosit pentru controlul intensitatii culorii verzi
LED-uri RGB	BLUE	D7	Pin PWM folosit pentru controlul intensitatii culorii albastre
LED-uri RGB	COMUN (catod)	GND	GND comun pentru toate LED-urile RGB
Potentiometru	Pin stanga	5V	Alimentare potentiometru
Potentiometru	Pin mijloc	A0	Intrare analogica ADC pentru reglarea intensitatii LED-urilor
Potentiometru	Pin dreapta	GND	GND comun
DHT11	VCC	5V	Alimentare senzor temperatura si umiditate

DHT11	DATA	D3	Comunicare digitala GPIO cu microcontrolerul
DHT11	NC	—	Pin neutilizat
DHT11	GND	GND	GND comun
Buton schimbare culoare	P1	D10	Intrare digitala pentru schimbarea culorii LED-urilor RGB
Buton schimbare culoare	P2	GND	Foloseste rezistenta interna pull-up
Buton start/stop citit + muzica ambientala	P1	D13	Intrare digitala pentru pornirea/oprirea sesiunii de citit
Buton start/stop citit + muzica ambientala	P2	GND	Foloseste rezistenta interna pull-up
Buton alarma - crestere timp	P1	D12	Intrare digitala pentru setarea alarmei
Buton alarma - crestere timp	P2	GND	Foloseste rezistenta interna pull-up
Buton alarma - scadere timp	P1	D11	Intrare digitala pentru activarea muzicii ambientale
Buton alarma - scadere timp	P2	GND	Foloseste rezistenta interna pull-up
Buzzer pasiv	(+)	D2	Pin controlat prin Timer/PWM pentru generarea frecventelor audio
Buzzer pasiv	(-)	GND	GND comun







In etapa actuala au fost implementate si verificate:

- conectarea LED-urilor RGB
- controlul culorilor LED-urilor
- conectarea butoanelor
- conectarea potentiometrului
- conectarea buzzerului pasiv
- conectarea senzorului DHT11
- conectarea LCD-ului prin I2C

De asemenea, a fost testata functionarea LED-urilor RGB prin schimbarea culorilor si reglarea intensitatii luminoase.

Software Design

Cod proiect: <https://github.com/micuco/Book-Mood-Lamp.git>

Codul pentru placa Arduino Uno a fost dezvoltat in PlatformIO, folosind C si acces direct la registrele microcontrollerului ATmega328P.

Fisier / Biblioteca	Rol
---------------------	-----

main.c	Contine logica principala a proiectului: stari, butoane, LCD, alarma, muzica si interactiunea dintre module.
usart.c / usart.h	Configureaza comunicatia seriala USART0 pentru mesaje de debug si comenzi simple din Serial Monitor.
adc.c / adc.h	Configureaza ADC-ul si citeste valoarea potentiometrului de pe A0.
twi.c / twi.h	Implementeaza comunicatia I2C/TWI prin registre, folosita pentru LCD.
lcd_i2c.c / lcd_i2c.h	Controleaza LCD-ul 16x2 prin adaptor I2C.
timers.c / timers.h	Configureaza Timer0 si Timer1 pentru PWM LED-uri, millis software si buzzer pasiv.
dht11.c / dht11.h	Citeste senzorul DHT11 folosind protocol digital single-wire pe pinul D3.
Biblioteca	Rol
<avr/io.h>	Permite accesul direct la registrele microcontrollerului.
<avr/interrupt.h>	Permite activarea intreruperilor si definirea rutinei ISR pentru Timer1.
<util/delay.h>	Folosita pentru intarzieri scurte necesare la initializarea LCD-ului si la protocolul DHT11.
<string.h>	Folosita pentru operatii pe siruri de caractere.
Functie	Rol
main()	Initializeaza modulele hardware, opreste LED-urile si buzzerul, la pornire, afiseaza mesajul initial pe LCD si ruleaza bucla principala a sistemului.
gpio_init()	Configureaza pinii butoanelor ca intrari cu pull-up intern si seteaza pinul DHT11 ca intrare initiala.
button_is_down_b(bit)	Citeste starea unui buton conectat pe PORTB. Deoarece se foloseste pull-up intern, buton apasat inseamna nivel LOW.
set_reading_state(active, now)	Porneste sau opreste sesiunea de citit. Cand citirea porneste, se aprinde lumina; cand se opreste, se stinge lumina si se opreste animatia.
update_brightness_from_pot()	Citeste potentiometrul prin ADC, filtreaza valoarea pentru reducerea zgomotului si actualizeaza intensitatea LED-urilor.
transition_task()	Realizeaza tranzitia smooth intre culori si ruleaza animatia de culori daca aceasta este activata.
lcd_update()	Actualizeaza informatiile de pe LCD: timpul de citit, temperatura, umiditatea si starea/timpul alarmei.
handle_main_button()	Gestioneaza butonul principal D13: apasare lunga pentru start/stop citit si lumina, dublu click pentru pornire/oprire muzica.
handle_alarm_buttons()	Gestioneaza butoanele D12 si D11 pentru cresterea/scaderea alarmei si combinatia simultana pentru setare/editare/oprire alarma.
check_alarm_trigger()	Verifica daca alarma setata a ajuns la zero si porneste starea de alarma.
alarm_sound_task()	Genereaza sunetul alarmei pe buzzer pana cand utilizatorul o opreste manual.
handle_color_button()	Gestioneaza butonul D10 pentru schimbarea culorii sau pornirea animatiei prin apasare lunga.
music_task()	Reda pe buzzer o versiune simplificata a Beethoven - Bagatelle No. 25 in A minor.
handle_serial()	Permite comenzi simple prin UART si transmite mesaje de debug catre PC.
Functie auxiliara	Rol

USART0_init(ubrr)	Configureaza baud rate-ul si activeaza transmisia/receptia USART0.
USART0_transmit(data)	Trimite un caracter catre PC prin UART.
USART0_receive_nonblocking(data)	Verifica daca exista un caracter primit fara a bloca executia programului.
USART0_print(text)	Trimite un sir de caractere prin UART.
adc_init()	Configureaza ADC-ul cu referinta AVcc si prescaler 128.
adc_read(channel)	Porneste conversia ADC pe canalul ales si returneaza valoarea intre 0 si 1023.
twi_init()	Configureaza interfata TWI/I2C la aproximativ 100 kHz.
twi_start(), twi_stop(), twi_write(data)	Realizeaza secventele de start, stop si scriere pe I2C.
lcd_init(), lcd_clear(), lcd_set_cursor(), lcd_print()	Controleaza LCD-ul prin I2C si permite afisarea mesajelor.
timers_init()	Configureaza Timer0 pentru PWM hardware si Timer1 pentru tick de timp, buzzer si PWM software.
millis_reg()	Returneaza timpul intern al sistemului in milisecunde.
rgb_set(r,g,b,brightness)	Aplica valoarea RGB si intensitatea pe LED-uri, folosind PWM.
light_enable(on)	Activeaza/dezactiveaza complet iesirile PWM pentru a evita aprinderea slaba a LED-urilor cand lampa este oprita.
buzzer_start(freq), buzzer_stop(), buzzer_beep()	Controleaza buzzerul pasiv prin frecvente generate de Timer1.
dht11_read(temp, hum)	Citeste temperatura si umiditatea de la DHT11 si valideaza checksum-ul.

Element calibrat	Mod de calibrare
Potentiometru pe A0	Valoarea ADC 0-1023 este transformata in domeniul 0-255 pentru PWM. Pentru a reduce zgomotul, se foloseste filtrare de tip medie exponentiala: valoarea noua este combinata cu valoarea anterioara. A fost introdus si un deadband, astfel incat diferentele foarte mici sa nu actualizeze PWM-ul.
LED-uri RGB	La intensitati foarte mici, LED-urile pot palpai. Pentru reducerea fenomenului, codul ridica usor pragul minim PWM cand lampa este pornita si dezactiveaza complet iesirile PWM cand lampa este oprita.
DHT11	Pentru DHT11 cu 4 pini s-a folosit rezistenta de pull-up de 10k intre VCC si DATA. Citirea se face la aproximativ 2.5 secunde, nu continuu, pentru a respecta timpul de raspuns al senzorului. Datele sunt validate prin checksum.
Butoane	S-a folosit pull-up intern si logica activa pe LOW. Pentru apasari simultane la alarma s-a introdus o fereastra scurta de protectie, astfel incat valoarea alarmei sa nu se modifice accidental cand se seteaza sau se opreste alarma.
Buzzer	Au fost testate frecvente diferite pentru buzzer. Alarma foloseste o frecventa in zona in care buzzerul se aude mai clar.

Laboratoare folosite:

- Laborator 0 - GPIO: Folosit pentru configurarea pinilor digitali, citirea butoanelor D10-D13, citirea liniei DATA pentru DHT11 si controlul iesirilor digitale.
- Laborator 1 - UART / USART: Folosit pentru comunicatia dintre Arduino Uno si PC. Codul transmite mesaje de stare in Serial Monitor si permite comenzi simple de debug, fara sa blocheze bucla principala.

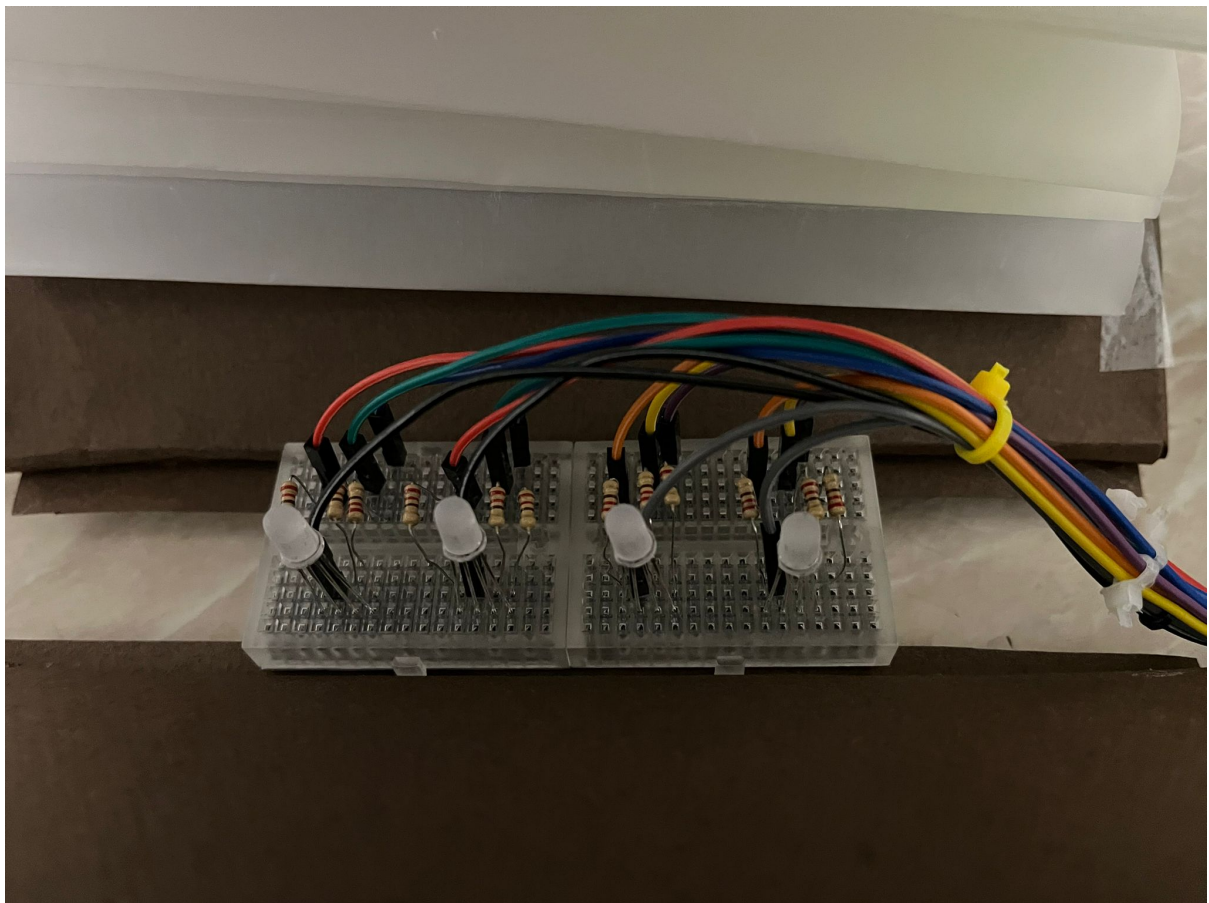
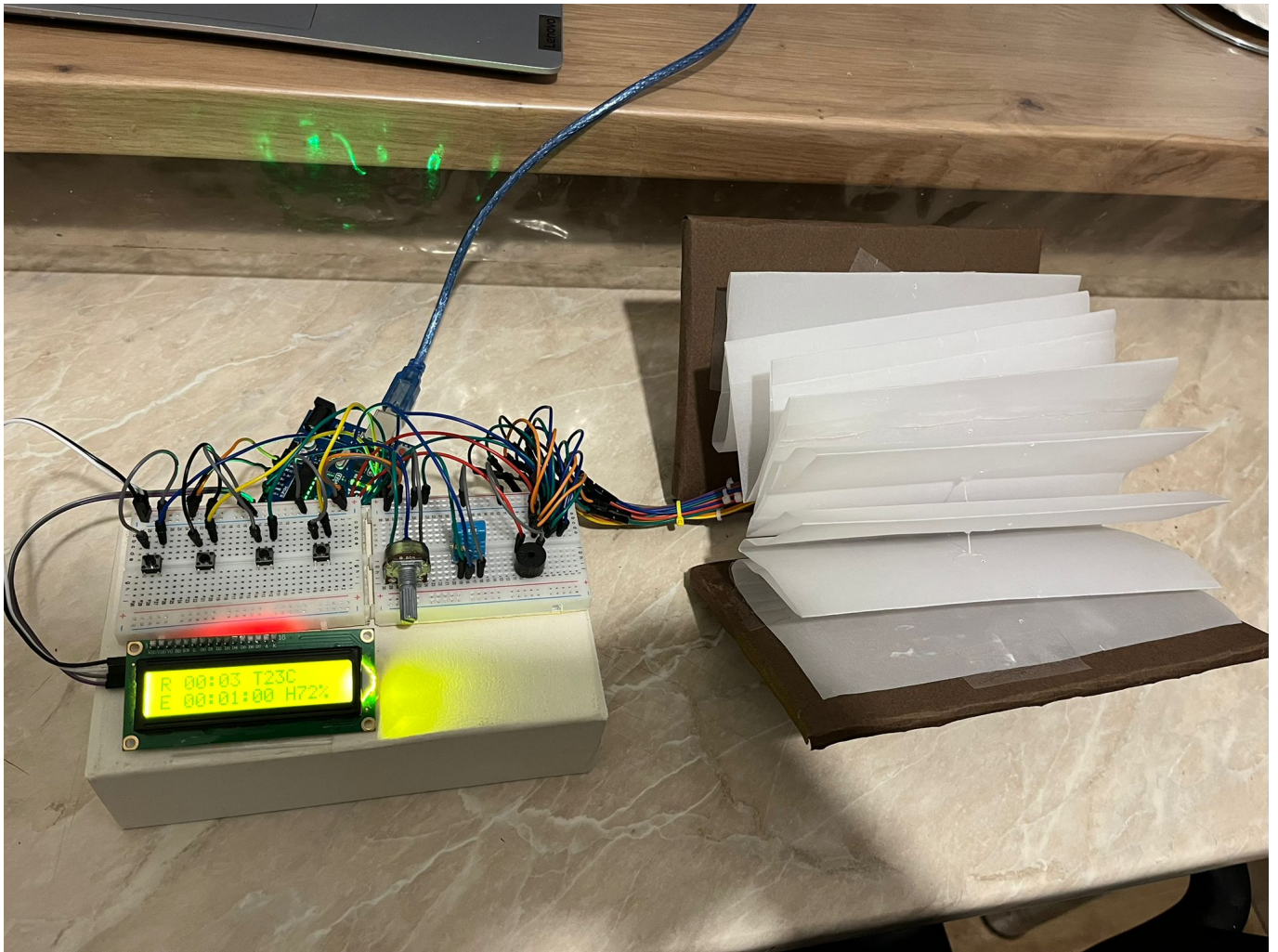
- Laborator 3 - Timere / PWM: Folosit pentru controlul LED-urilor RGB, pentru generarea timpului intern millis, pentru animatii si pentru frecventele buzzerului pasiv. Timer0 controleaza RED/GREEN prin PWM hardware, iar Timer1 genereaza tick-ul de timp, buzzerul si PWM software pentru BLUE.
- Laborator 4 - ADC: Folosit pentru citirea potentiometrului de pe A0. Valoarea ADC controleaza intensitatea LED-urilor RGB.
- Laborator 6 - I2C: Folosit pentru comunicatia cu LCD-ul I2C. Arduino trimite comenzi si date catre adaptorul LCD prin TWI/I2C pe pinii A4/SDA si A5/SCL.

Rezultate Obținute

Demo: https://youtu.be/b1M8POujwxU?is=X4ejjwqw6nSt_CQ1

Sistemul implementat realizeaza functiile:

- pornirea si oprirea sesiunii de citit si a luminii prin apasare lunga pe butonul principal (cel mai din stanga buton)
- reglarea intensitatii LED-urilor prin potentiometru
- schimbarea culorilor cu tranzitie smooth prin apasare de buton scurta si rularea unei animatii de culori prin apasare de buton lunga (cel mai din dreapta buton)
- afisarea pe LCD a timpului de citit (stanga sus), a alarmei (stanga jos), a temperaturii (dreapta sus) si a umiditatii (dreapta jos)
- setarea alarmei pana la 24 de ore, cu afisare HH:MM:SS si countdown (setare prin butoanele din mijloc - buton dreapta scade timpul, buton stanga creste timpul, apasarea simulatana a celor 2 butoane seteaza alarma sau o pune in modul de editat); o apasare de buton creste/scade timpul cu un minut, o apasare prelungita creste/scade timpul din minut in minut mult mai rapid
- redarea unei melodii simple pe buzzer (prin apasare de 2 ori pe cel mai din dreapta buton) si pornirea unei alarme sonore pana la oprire manuala (prin apasarea simultana a celor 2 butoane din mijloc)
- transmiterea mesajelor de debug prin UART catre Serial Monitor



Concluzii

Proiectul arata cum pot fi combinate mai multe functionalitati hardware folosind Arduino Uno si programare la nivel de registre. In cadrul proiectului au fost folosite concepte din mai multe laboratoare: GPIO, UART/USART, Timere/PWM, ADC si I2C. Sistemul permite controlul iluminarii ambientale, afisarea temperaturii si umiditatii, redarea unei melodii pe buzzer si setarea unei alarme pentru timpul de citit. Pe viitor, proiectul poate fi imbunatatit prin realizarea unei aplicatii pentru telefon din care utilizatorul sa controleze lampa de la distanta, sa schimbe culorile si intensitatea LED-urilor sau sa selecteze diferite melodii ambientale.

Bibliografie/Resurse

- https://ww1.microchip.com/downloads/en/devicedoc/atmel-7810-automotive-microcontrollers-atmega328p_datasheet.pdf
- <https://components101.com/sensors/dht11-temperature-sensor>
- https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf
- <https://docs.arduino.cc/tutorials/generic/secrets-of-arduino-pwm/>
- Laboratoarele rezolvate

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2026/ciprian.popescu0411/elena_corina.micu



Last update: **2026/05/26 08:49**