

# Ambient Mood Light

Proiectul constă într-un sistem care citește în timp real temperatura și luminozitatea din mediul înconjurător și generează o lumină ambientală a cărei culoare și intensitate reflectă condițiile detectate.

## Introducere

Ideea a pornit de la modul în care lumina influențează starea oamenilor, iar aceasta de obicei este setată static. Astfel, proiectul creează un sistem ambiental care observă mediul și se adaptează automat la el.

Astfel, o zi însorită și caldă va induce o lumină roșie-portocalie vibrantă, iar o zi rece și întunecată va genera o lumină albastră cu intensitate scăzută. De asemenea, utilizatorul va putea ajusta luminozitatea manual printr-un buton, și va putea vedea temperatura și luminozitatea prin intermediul unui display.

scopul este de a avea un obiect decorativ care poate fi ușor integrat în orice locuință, care reacționează la contextul dat și poate

## Descriere generală



Schema bloc de mai sus prezintă arhitectura completă a sistemului, structurată pe trei niveluri:

**Strat de intrare (senzori):** DHT22 comunică pe protocolul 1-Wire și furnizează temperatura în grade Celsius. BH1750 comunică pe I2C și returnează valoarea luminozității în lux. Un buton tactil permite comutarea manuală între moduri de funcționare. Microcontroller (ATmega328P XPlained Mini): Coordonează citirea senzorilor, rulează algoritmul de mapare HSV și controlează perifericele de ieșire. Operează la 5V / 16MHz.

**Strat firmware:** Format din trei module — modulul de citire senzori, motorul de mapare HSV (temperatura controlează Hue: albastru→roșu; luminozitatea controlează Value: întunecat→strălucitor) și managerul de moduri (Auto, Manual).

**Strat de ieșire:** Banda WS2812B primește valorile RGB calculate și afișează culoarea corespunzătoare. Display-ul OLED SSD1306 afișează în timp real valorile numerice de temperatură și luminozitate.

# Hardware Design

Listă de componente

Componentă	Model	Cantitate
Microcontroller	ATmega328P XPlained Mini	1
Senzor temperatură/umiditate	DHT22 AM2302 (modul)	1
Senzor luminozitate	GY-302 BH1750	1
Bandă LED adresabilă	WS2812B 60LED/m, 10cm	2
Display	OLED 0.96" SSD1306 I2C	1
Buton	PCB mini 6x6x5, 4 pini	3
Breadboard	MB-102 830 puncte	1
Fire	Jumper set 140 cabluri	1
Condensator	Electrolitic 10.000µF/25V	1
Rezistoare	Kit 30 valori, 600 buc, 1/4W	1

Schema electrică realizată in KiCad:



Componentele au fost legate în configurația următoare

Componentă	Pin componentă	Pin ATmega328P XPlained Mini	Observații
<b>DHT22</b>	VCC	5V (J202)	—
	DATA	PD2 (J201)	Pull-up inclus pe modul
	GND	GND (J202)	—
<b>BH1750</b>	VCC	5V (J202)	—
	GND	GND (J202)	—
	SDA	PC4 (J200)	Magistrală I2C partajată
	SCL	PC5 (J200)	Magistrală I2C partajată
	ADD	GND	Adresă I2C → 0x23
<b>SSD1306 OLED</b>	VCC	5V (J202)	—
	GND	GND (J202)	—
	SDA	PC4 (J203)	Magistrală I2C partajată
	SCL	PC5 (J203)	Magistrală I2C partajată
<b>WS2812B</b>	+5V	5V (J202)	—
	GND	GND (J202)	—
	DIN	PD6 (J201)	Prin rezistor 300Ω
<b>Buton</b>	Pin 1	PD4 (J201)	Pull-up intern activat
	Pin 2	GND	—
<b>Rezistor 300Ω</b>	Terminal 1	PD6 (J201)	Protecție date WS2812B
	Terminal 2	WS2812B DIN	—

## Componente folosite și rolul lor:

ATmega328P XPlained Mini — Coordonează citirea senzorilor, rulează algoritmul de mapare a culorii și gestionează toate perifericele de intrare și ieșire

DHT22 AM2302 — senzor de temperatură și umiditate. Furnizează temperatura în grade Celsius, valoare folosită pentru a determina culoarea benzii LED prin mapare HSV: temperaturi scăzute generează nuanțe de albastru, temperaturi ridicate generează nuanțe de portocaliu-roșu.

BH1750 GY-302 — senzor de luminozitate. Returnează valoarea iluminării în lux. Valoarea este folosită în modul automat pentru a controla intensitatea benzii LED: mediu întunecat produce lumină mai slabă, mediu luminos produce lumină mai intensă.

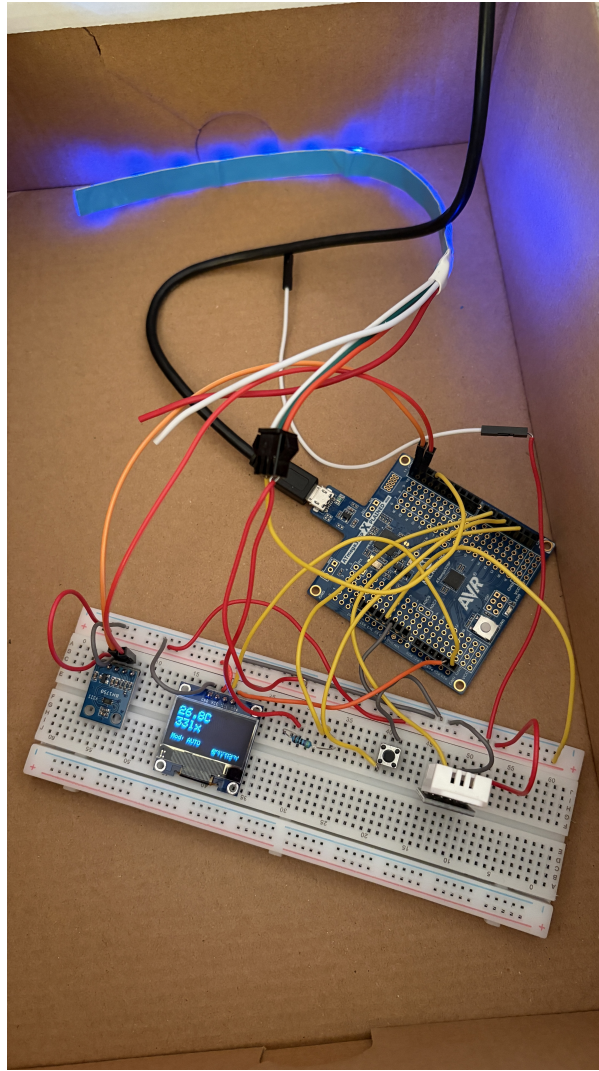
WS2812B (banda LED) - elementul de ieșire principal al sistemului. Culoarea și intensitatea sunt calculate de microcontroller pe baza datelor de la senzori.

SSD1306 OLED 0.96" — display auxiliar care afișează în timp real valorile de temperatură, luminozitate și modul curent de funcționare (AUTO / MANUAL).

Buton - permite utilizatorului să regleze manual intensitatea, unde aceasta crește în trepte la fiecare apăsare. Revenirea la modul automat se face automat după 5 secunde de inactivitate.

## Stadiu actual + Componentă funcțională

La momentul de față întreg proiectul este ansamblat, fiecare componentă fiind funcțională. Temperatura și luminozitatea sunt citite corect și afișate pe display, alături de modul AUTO/MANUAL + NIVELUL DE LUMINOZITATE.



Banda LED a afișat culoarea corespunzătoare temperaturii măsurate - nuanță de albastru, conform mapării implementate pentru intervalul 24-31°C, cu intensitatea reglementată automat în funcție de luminozitatea ambientală. În link-ul următor se poate observa un video în care acesta își schimbă culoarea, având un feon care era țintit către senzorul de temperatură DHT22:

<https://drive.google.com/drive/folders/14bySmPVbB63vOo3YLh20lzzRB1eumNvZ?usp=sharing>

## Software Design

### Scheletul proiectului și interacțiunea dintre funcționalități

Firmware-ul este organizat în trei straturi funcționale care rulează concurent în bucla principală.

Stratul de intrare se execută la fiecare 2 secunde și are rolul de a citi valorile de la senzori și de a actualiza starea sistemului. Pe baza temperaturii citite de DHT22 se calculează culoarea țintă, iar pe baza luminozității citite de BH1750 se calculează intensitatea țintă în modul AUTO. La finalul fiecărei achiziții, display-ul OLED este actualizat cu valorile curente.

Stratul de control al butonului rulează la fiecare iterație a buclei și verifică dacă ISR-ul a semnalat o apăsare. Dacă da, aplică debouncing-ul de 50ms, apoi comută între modurile AUTO și MANUAL sau avansează treapta de luminozitate. De asemenea, monitorizează timeout-ul de 5 secunde după care sistemul revine automat la modul AUTO.

Stratul de ieșire rulează la aproximativ 50fps și este responsabil de actualizarea benzii LED. La fiecare frame, culoarea curentă avansează treptat spre culoarea țintă prin blending, iar intensitatea curentă avansează cu un pas spre intensitatea țintă. Valorile rezultate sunt trimise benzii WS2812B.

Validarea sistemului a fost realizată pe mai multe niveluri. Temperatura și luminozitatea au fost comparate cu valori de referință prin Serial Monitor. Maparea culorii a fost verificată prin printarea valorilor RGB calculate la diverse temperaturi simulate. Butonul a fost testat prin apăsări rapide pentru a confirma că debouncing-ul funcționează corect. Timeout-ul de revenire la AUTO a fost verificat cu cronometru pe telefon, iar tranziția de culoare a fost validată vizual, confirmând absența salturilor la schimbări bruște de temperatură.

## Biblioteci folosite

FastLED - aleasă pentru controlul benzii WS2812B datorită funcției `nblend()` care realizează interpolarea lină între culori fără a fi nevoie de implementarea manuală fiecare canal RGB. De asemenea, `fill_solid()` și `FastLED.setBrightness()` simplifică controlul întregii benzi. `#define FASTLED_ALLOW_INTERRUPTS 0` dezactivează întreruperile în timpul scrierii pe LED-uri, prevenind coruperea semnalului one-wire al WS2812B care e foarte sensibil la timing

DHT (Adafruit) - biblioteca standard pentru protocolul one-wire proprietar al senzorului DHT22

BH1750 - bibliotecă simplă pentru senzorul de luminozitate care comunică pe I2C. `lightMeter.readLightLevel()` returnează direct valoarea în lux, nefiind necesară conversia manuală din valori ADC

Adafruit SSD1306 + Adafruit GFX - set standard de biblioteci pentru display-uri OLED. GFX oferă funcții de desenare text și grafice, SSD1306 gestionează comunicația I2C cu display-ul. Alternativa ar fi fost controlarea display-ului prin registre TWI care ar fi adus complexitate codului

Wire - biblioteca Arduino pentru I2C, utilizată implicit de BH1750 și SSD1306

`avr/interrupt.h` - inclusă pentru configurarea întreruperii hardware PCINT2 prin registre directe

## Elementul de noutate

Proiectele de tip ambient light existente folosesc în general o singură variabilă de intrare (cel mai des temperatura sau lumina, separat). Noutatea proiectului constă prin combinarea a două variabile de mediu - temperatura și luminozitatea, pentru a le putea controla simultan

## Funcționalități din laborator

Lab 0 - GPIO: Butonul este configurat prin registrele DDR și PORT ale ATmega, cu pull-up intern activat

Lab 2 - Întreruperi: Butonul este gestionat printr-un Pin Change Interrupt, evitând polling-ul în bucla principală. Configurarea registrelor, marcarea volatile a flag-ului partajat între ISR și main, și debouncing-ul fără blocare

Lab 6 - I2C: BH1750 și SSD1306 comunică pe magistrala I2C.

## Logica de proiectare a culorii/intensității

La fiecare 2 secunde, senzorii furnizează o nouă culoare și intensitate țintă. În loc să fie aplicate imediat, valorile curente sunt interpolate spre cele țintă la fiecare frame (~50fps). Interpolarea funcționează proporțional cu diferența rămasă - pasul aplicat la fiecare frame este o fracțiune fixă din distanța dintre valoarea curentă și cea țintă, astfel: tranziția este rapidă la început și încetinește natural pe măsură ce valorile se apropie. Acest comportament este obținut prin `nblend()` pentru culoare, care operează independent pe fiecare canal RGB, și printr-un increment/decrement de 1 pas per frame pentru intensitate.

## Calibrarea elementelor de senzorică

DHT22: Senzorul a fost comparat cu temperatura camerei. Intervalele `TEMP_MIN = 24.0°C` și `TEMP_MAX = 31.0°C` au fost alese pentru a acoperi domeniul de temperatură interior tipic vară/iarnă și pentru a produce variație vizibilă de culoare în condiții normale de utilizare.

BH1750: Senzorul returnează direct lux calibrați din fabrică, fără a necesita calibrare suplimentară. Pragurile `LUX_DARK = 10 lux` (lcameră întunecată) și `LUX_BRIGHT = 800 lux` (lumină puternică de zi) au fost determinate prin testarea acestuia în cameră, similar cu senzorul de temperatură

## Optimizări realizate

1. Non-blocking timing: Citirea senzorilor folosește `millis()` - `lastRead >= READ_INTERVAL_MS` în loc de `delay(2000)`, permițând buclei principale să ruleze continuu la 50fps pentru tranziții line și răspuns rapid la buton, indiferent de intervalul de citire al senzorilor.
2. Separarea target/current: Prin delimitarea valorilor dorite de cele afișate fiecare rulează la frecvența proprie fără a se bloca reciproc.
3. `BRIGHTNESS_MAX = 20` în loc de 255: WS2812B la putere maximă pe 12 LED-uri consumă până la

~720mA. Limitând la 20/255, aprox. 8%, consumul scade la ~56mA, astfel fiind compatibil cu alimentarea prin USB fără risc de supraîncărcare și fără a necesita sursă externă dedicată

## Rezultate Obținute

Sistemul funcționează conform specificațiilor inițiale. Temperatura și luminozitatea sunt citite corect și reflectate în timp real pe bandă și pe display. Tranziția de culoare este lină și continuă, fără salturi vizibile între citiri, efectul vizual fiind cel al unei lumini care se schimbă organic. Intensitatea se adaptează automat la lumina ambientală, banda fiind discretă în întuneric și mai prezentă în lumină puternică.

Modul MANUAL funcționează corect - butonul ciclează prin cele 5 trepte de luminozitate, iar sistemul revine automat la modul AUTO după 5 secunde de inactivitate, fără intervenția utilizatorului.

Display-ul afișează în permanență valorile numerice de temperatură și luminozitate, precum și modul curent.

## Concluzii

Prin realizarea acestui proiect am reușit să înțeleg mai bine cum se integrează o serie de periferice simultan. Am realizat că integrarea mai multor componente într-un sistem care trebuie să răspundă constant și natural cere detalii care par mici la prima vedere (cum ar fi modul în care treci de la o culoare la alta), însă pot face diferența dintre un rezultat care își atinge obiectivul și unul care nu.

## Download

Arhivă proiect: [pm\\_proiect.zip](#)

## Bibliografie/Resurse

### Resurse Hardware

[Datasheet DHT22 AM2302](#)

[Datasheet BH1750FVI](#)

[Datasheet SSD1306](#)

[Datasheet WS2812B](#)

## Resurse Software

[Documentație FastLED](#)

[Documentație Adafruit SSD1306](#)

[Documentație Adafruit GFX](#)

[Documentație BH1750 Arduino](#)

[Documentație DHT sensor library](#)

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2026/cezar.zlatea/tiberiu\\_florin.gila](http://ocw.cs.pub.ro/courses/pm/prj2026/cezar.zlatea/tiberiu_florin.gila)



Last update: **2026/05/23 09:38**