

Lampa de birou cu control bluetooth

- Student: Pavel Ioan Cristian
- Grupa: 331CA
- GitHub:

Introducere

Ce face?: Proiectul este o lampa de birou *smart*, capabila sa ajusteze automat intensitatea luminii si sa porneasca/opreasca lumina atunci cand detecteaza miscare in camera; de asemenea, este implementat un mod standard de functionare on/off, cu control de pe telefon.

Scopul: Un dispozitiv inteligent care se adapteaza la modul tau de utilizare; nu trebuie sa faci nimic, doar sa te asezi la birou.

Ideea de start: Nevoia de o sursa de lumina care porneste singura cand te asezi la birou si se stinge singura cand pleci; practic, doar o pozitionezi unde trebuie, iar restul este controlat din partea de software.

Utilitate: Te apuci de lucru la birou cu minim de efort.

Descriere generala

La baza sistemul comuta intre 4 moduri de functionare: Standby (led stins), On (led aprins), Eco (detectie de miscare) si adaptive (ajustare automata in functie de lumina din camera). Schimbarea modului de lucru/a parametrilor de functionare se realizeaza din telefon printr-o conexiune bluetooth sau prin intermediul unui buton.

Interactiunea modulelor:

Senzorul de lumina (Fotorezistorul LDR): Este integrat intr-un circuit cu divizor de tensiune pentru a monitoriza intensitatea luminii. Microcontrolerul folosec modului ADC pentru a esantiona valoarea analogica prin pinul PA0. Valoarea obtinuta este calculata constant, in timp real, si este folosita doar in modul adaptive.

Senzorul de miscare (PIR HC-SR501): Este un trigger pentru modul "Eco". Este conectat la pinul PD3, configurat sa genereze o intrerupere externa (INT1) la detectia miscarii. Astfel, banda LED comuta intre aprins si stins instantaneu, atunci cand e nevoie; se faciliteaza un consum redus de energie.

Controlul LED-urilor (MOSFET IRFZ44N & PWM): Banda LED functioneaza la 12V si este controlata printr-un tranzistor MOSFET. Microcontrolerul genereaza un semnal PWM prin Timerul 1 pe 16 biti, permitand o variatie fina a intensitatii luminoase, fara efect de palpaire, prin modificarea

factorului de umplere.

Modulul Bluetooth (HC-06): Permite controlul de la distanta al lampii prin aplicatie pe telefon. Comunicatia cu microcontrolerul se realizeaza prin protocolul USART, pinii PD0/RX si PD1/TX. Se pot trimite comenzi pentru schimbarea intensitatii LED-urilor sau comutarea intre modurile de functionare.

Control manual (Buton & Buzzer): Butonul fizic este conectat pe pinul PD2 (INT0) si este folosit pentru schimbarea modurilor de functionare fara telefon. Buzzer-ul genereaza beep-uri scurte la confirmarea comenzilor primite, fiind controlat printr-un pin digital GPIO.

Alimentare (LM2596): Intrarea principala de 12V alimenteaza direct banda LED, in timp ce regulatorul LM2596 coboara si stabilizeaza tensiunea la 5V. Acesta din urma alimenteaza microcontrolerul si senzorii, protejand componentele de fluctuatiile de curent.

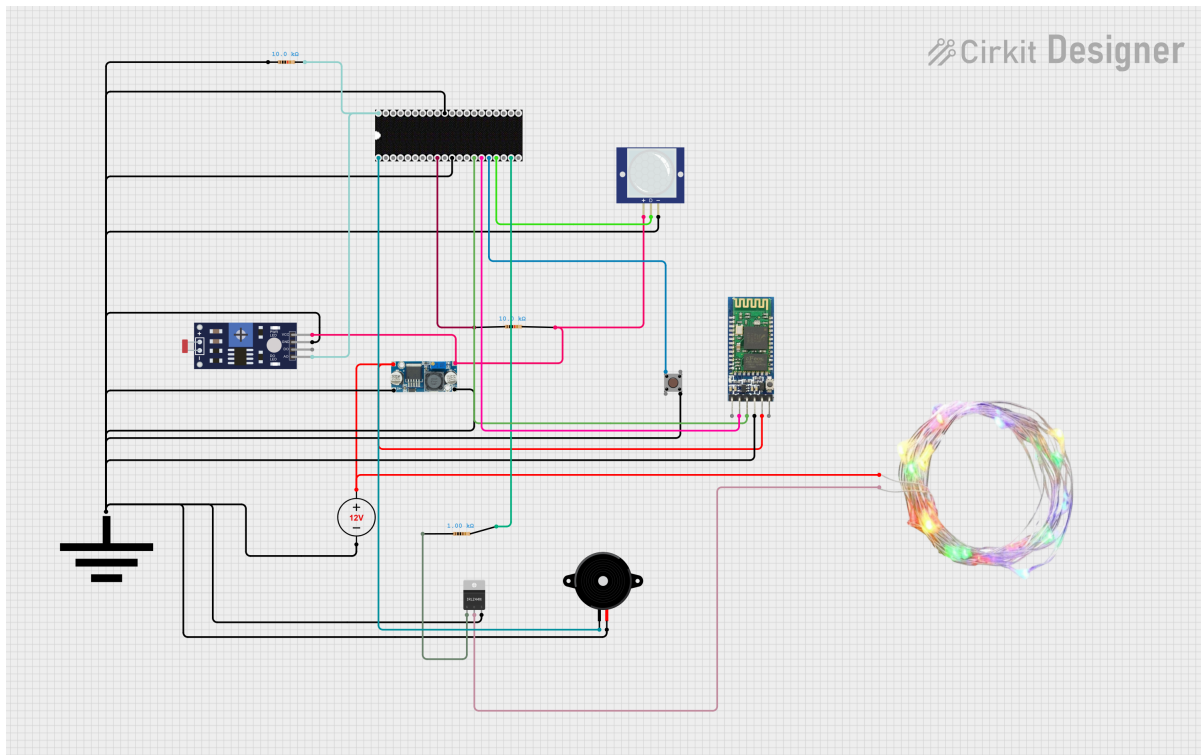
Schema bloc:



Hardware Design

Componenta	Model	Interfață cu MCU	Funcție
Microcontroler	ATMega328P	—	Unitate centrala
Senzor lumina	Fotorezistor LDR	ADC (Pin PA0)	Masurarea intensitatii luminii
Senzor miscare	PIR HC-SR501	GPIO + INT (Pin PD3)	Detectia miscarii
Bluetooth	HC-06	UART (PD0, PD1)	Comunicatie wireless cu aplicatia de telefon
Control de putere	MOSFET IRFZ44N	PWM (Pin PD5)	Comutarea curentului de 12V pentru banda LED
Limitator tensiune	LM2596	—	Limitare 12V → 5V
Iluminare	Banda LED albastra	MOSFET	Sursa de lumina
Feedback audio	Buzzer	GPIO (Pin PB0)	Semnalizare audio la schimbarea modurilor
Control manual	Buton	GPIO + INT (Pin PD2)	Schimbare manuala intre modurile de functionare
Elemente pasive	Rezistente 10kΩ, 1kΩ	—	Rezistente de pull-down, divizor de tensiune si protectie

Schema electrica:



Software Design

1. Mediul de dezvoltare si arhitectura hibrida

Proiectul a fost dezvoltat cu PlatformIO, in limbaj C, folosind si un software multi-tier (python si web) pentru a permite controlul la distanta.

Firmware (C): Controlul perifericelor microcontrolerului (PWM, comunicatie seriala, intreruperi hardware, ADC) + interactionare exclusiva cu registrii chip-ului ATmega328P.

Backend local (Python Flask & PySerial): Rulat pe laptop, serverul actioneaza ca o punte de legatura; expune un API REST cu rute dedicate (/lampa/on, /lampa/off, /lampa/auto) si traduce request-urile HTTP venite din retea in comenzi de tip caractere ASCII ('1', '0', 'A'), pe care le trimite instantaneu prin portul serial (USB sau Bluetooth) catre microcontroler folosind `bt.flush()`.

Frontend si tunneling (Web & SSH): Interfata grafica este o pagina web hostata public prin GitHub Pages. Pentru bypass-ul restrictiilor retelelor se creeaza un tunel (`ssh -R 80:127.0.0.1:5000 nokey@localhost.run`). Serviciul genereaza un link public HTTPS dinamic oferind acces instant de pe datele mobile ale telefonului direct la serverul local Flask.

2. Algoritmul principal si fluxul de date

Logica centrala a sistemului are la baza un selector de moduri prin variabila globala `mod_manual`.

Initializarea: Se configureaza directiile pinilor (I/O), interfata UART la 9600 bps, ADC-ul, temporizatorul pentru PWM si se activeaza intreruperile externe si cele globale. Sistemul emite un beep de confirmare dupa o stabilizare initiala de 2 secunde.

Listening non-blocant (UART): In bucla principala, functia `UART_Receive()` verifica registrul de receive fara a bloca executia codului. Daca de pe telefon se trimite o comanda, variabila `mod_manual` isi

schimba starea, 0 - auto, 1 - on, 2 - off), iar buzzer-ul emite un beep de confirmare.

Logica Aplicatiei:

Mod manual ON (`mod_manual == 1`): forteaza activarea registrului PWM la valoarea maxima (`OCR0B = 255`), aprinzand banda LED prin MOSFET.

Mod Manual OFF (`mod_manual == 2`): dezactiveaza timerul PWM, forteaza pinul PD5 in starea Low (0V) si opreste alimentarea benzii LED.

Modul Automat (`mod_manual == 0`): sistemul citeste valoarea luminii ambientale prin ADC. Daca valoarea LDR scade sub pragul stabilit si flag-ul de miscare este activ se aprinde lampa.

prompt dinamic (JS): link-ul de SSH este dinamic si se schimba la fiecare repornire in CMD, iar codul JavaScript din `index.html` include o logica de fallback. La prima apasare a unui buton, aplicatia deschide un prompt pop-up pe ecranul telefonului cerand introducerea link-ului HTTPS generat; valoarea este salvata local in sesiune.

3. Implementarea modulelor periferice

uart.c / uart.h (comunicatie seriala - Lab 1) - configurarea si gestionarea fluxului de date prin portul serial al microcontrolerului;

interrupts.c / interrupts.h (Intreruperi Externe - Lab 2)

Butonul Fizic (INT0): Pinul PD2 este configurat ca intrare; registrul EICRA este configurat cu bitul ISC01 pentru a declansa intreruperea hardware pe front descrescator; rutina `ISR(INT0_vect)` realizeaza un debounce software rapid de 50ms, dupa care se schimba starea modului.

senzorul PIR (INT1): Pinul PD3 primeste semnal de la senzorul de miscare; este mapat pe intreruperea INT1, configurata prin bitul ISC10 din EICRA. in rutina `ISR(INT1_vect)`, starea pinului PIND este citita direct;

timer.c / timer.h (Generare semnale PWM - Lab 3) - gestioneaza controlul fin si tranzitiile de luminozitate ale benzii LED de mare putere.

Timer 0 (8-bit): configurat sa lucreze in modul fast PWM; pinul PD5 este setat in modul non-inverted; prescalerul este setat la valoarea 64 prin registrul pentru a obtine o frecventa stabila; luminozitatea este controlata direct prin scrierea de valori in registrul de comparare `OCR0` si modificand factorul de umplere.

adc.c / adc.h (Conversie Analog-Digital - Lab 4) - permite citirea valorilor de pe senzorul fotorezistiv (LDR) montat intr-un circuit de divizor de tensiune cu o rezistenta fixa de 10kΩ:

`ADC_Init`: seteaza tensiunea de referinta la AVCC (5V); se configureaza un prescaler de 128 prin registrul `ADCSRA`.

`ADC_Read`: primeste ca parametru canalul (0 pentru pinul PC0); returneaza valoarea convertita pe 10 biti, reprezentand nivelul de luminozitate procesat de algoritmul principal.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Proiectul a fost finalizat, rezultand într-un prototip functional, complet automatizat, capabil sa fie controlat de la distanta.

Sistemul rezolva o problema reala de eficienta energetica si confort. Iluminatul obisnuit ramane activ chiar si cand nu este necesar, sau necesita interactiune fizica constanta. Prin combinarea senzorului fotorezistiv cu cel de miscare se garanteaza faptul ca banda LED consuma energie strict atunci cand cineva este in incapere si lumina naturala este insuficienta. In plus, adaugarea componentei de server web si trecerea prin tunelul SSH a rezolvat problema de accesibilitate IoT in medii cu retele restrictive (cum ar fi lipsa unui IP public sau limitari de tip AP Isolation), permitand controlul prin date mobile.

Desi sistemul indeplineste scopul de proiectare, se pot aduce cateva imbunatatiri pe viitor:

Inlocuirea modulului Bluetooth (HC-06) cu un microcontroler cu Wi-Fi integrat (ex. ESP8266 sau ESP32) pentru a elimina necesitatea unui server cu rol de bridge.

Crearea unui PCB (Printed Circuit Board) dedicat pentru a face un sistem mai compact, care sa se potriveasca cu design-ul anumitor corpuri de lampa.

Bibliografie/Resurse

Resurse Hardware:

- [ATmega328P 8-bit AVR Microcontroller Datasheet](#)
- [ATmega328P Xplained Mini User Guide](#)
- [HC-06 Bluetooth Module Datasheet](#)
- [HC-SR501 PIR Motion Sensor Datasheet](#)
- [Photoresistor \(LDR\)](#)
- [SSH Port Forwarding Documentation](#)


Resurse Software și Tool-uri:

- Cirkit Designer (schema electrica)
- GitHub (cod sursa)

[Export to PDF](#)

Last update:
2026/05/19 pm:prj2026:bianca.popa1106:ioan_cristian.pavel http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/ioan_cristian.pavel
11:51

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/ioan_cristian.pavel 

Last update: **2026/05/19 11:51**