



Universitatea Națională de Știință și
Tehnologie POLITEHNICA
București



FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI
TEHNOLOGIA INFORMAȚIEI

Proiect 2 – Electronică aplicată
Sistem de monitorizare & control al iluminatului într-o
încăpere

Autori: Anton Alexandra-Teodora, Oncioiu Andrei-Daniel

Coordonatori științifici: Ana Neacșu, Mihai Muntean,
Alexandru Guzu, Andrei Dăescu, Dorin Enăchescu

TITULAR DISCIPLINĂ: AS. UNIV. DRD. ING. NEACȘU ANA

Lista figurilor

1. Figura 1: Modalitatea de calcul a rezistențelor alese pentru protecția LED-urilor
2. Figura 2: Modalitatea de calcul a rezistențelor alese în vederea obținerii divizorilor de tensiune
3. Figura 3: Design-ul architectural al sistemului de monitorizare & control al iluminatului dintr-o încăpere
4. Figura 4: Schema electrică de funcționare a sistemului de monitorizare & control al iluminatului într-o încăpere
5. Figura 5: Testarea funcționalității senzorului de intensitate luminoasă din încăpere
6. Figura 6: Organigrama codului

Cuprins

I. Acronime

II. Introducere

I.1. Descrierea proiectului

I.2. Descrierea funcționalității

III. Resurse hardware

IV. Resurse software

V. Implementare hardware

VI. Implementare software

VII. Concluzii & optimizări.

VIII. Bibliografi

I. Acronime

În scopul oferirii clarității, accesibilității, consistenței, respectiv cel al evitării ambiguității, vom include în primul rând toate acronimele utilizate în cadrul proiectului, însoțite de explicații. Astfel, se observă prezența următorilor termeni, în cadrul lucrării de față:

- LCD: Liquid Crystal Display-Display cu Cristale Lichide;
- PCI: Peripheral Component Interconnect-Interconectarea componentelor periferice;
- USB: Universal Serial Bus-Bus Serial Universal;
- LED: Light-emitting diode-Diodă emițătoare de lumină;
- SCL: Serial Clock Line-Linie de ceas serial;
- SDA: Serial Data Line-Linie de date serială;
- IDE: Integrated development environment-Mediu de dezvoltare integrat;
- LDR: Light Dependent Resistor-Rezistor dependent de lumină / foto-rezistor.

II. Introducere

De-a lungul anilor, sistemele automate și-au dovedit superioritatea față de cele manuale, întrucât reușesc cu ușurință să sporească eficiența și durabilitatea în timp, minimalizând utilizarea resurselor în scopul de a conserva energia și, simultan, de a reduce costurile.[1] În cadrul prezentului proiect, ne propunem să realizăm o un circuit capabil să afișeze și să modifice intensitatea luminoasă dintr-o încăpere, oferind utilizatorilor posibilitatea de a ajusta nivelul luminii după propriile preferințe. Pentru a putea ilustra funcționalitatea proiectului, am decis să folosim o scară mai mică, concentrându-ne pe utilizarea unui bec în locul unui sistem electric complex, care se regăsește în instalația dintr-o cameră. Utilizarea microcontrolerul Intel Galileo în realizarea acestui scop ne permite să obținem o controlare precisă și eficientă a luminii, alături de celelalte componente utilizate și a programului implementat.

În primă instanță, foto-rezistorul detectează valoarea intensității luminoase sau nivelul de lumină ambientală, convertind această valoare într-o măsură numerică, ce va fi apoi afișată pe un ecran LCD(Light-emitting diode-Diodă emițătoare de lumină) conectat la circuit. Microcontrolerul procesează continuu aceste date și, în funcție de pragurile stabilite în cadrul algoritmului de funcționare, acesta decide când să aprindă sau să stingă lampa, conectată la sursa de curent. Spre exemplu, așa cum se va vedea ulterior, dacă nivelul de lumină detectat se află sub un anumit prag predefinit, care poate fi modificat din structura algoritmului în funcție de preferințele utilizatorilor, microcontrolerul va transmite informația ce presupune nevoia de aprindere a lămpii. În acest caz, se va aprinde LED-ul (Light-emitting diode-Diodă emițătoare de lumină) galben conectat la circuit. Dacă nivelul intensității este considerat a se afla în limita ambientală, atunci microcontrolerul va menține în continuare lampa în poziția în care se afla inițial, stins sau aprins, fără să îl modifice, aprinzându-se LED-ul de culoare verde. Dacă nu se respectă niciuna dintre condițiile enumerate anterior, atunci se va aprinde LED-ul de culoare roșie, fapt ce sugerează depășirea nivelului setat și ce conduce la decizia de stingere a lămpii. În acest fel, se produce economisirea de energie electrică. Niciunul din 3 LED-uri nu vor fi aprinse în mod simultan.

În general, sistemele automate joacă un rol crucial în ceea ce privește automatizarea locuințelor, a ceea ce înseamnă conceptul de „smart home”, ce reușește să facă viața cotidiană mult mai confortabilă, reușind să faciliteze consumatorilor o varietate de electronice și electrocasnice, de la ventilatoare simple, până la cuptoare. [2]Reușita implementării unui astfel de proiect și utilizarea acestuia poate aduce o multitudine de beneficii în viețile de zi cu zile ale utilizatorilor, pe diverse planuri. Îmbunătățirea calității vieții provine în primul rând din economisirea de energie, atât pe plan personal, din punct de vedere financiar, cât și din perspectiva protejării mediului înconjurător. În același timp, se poate constata o îmbunătățire în ceea ce privește starea de spirit a utilizatorilor, oferită de către nivelul de confort sporit, datorat automatizării procesului de ajustare a luminii în funcție de preferințe, fără să fie nevoie ca utilizatorul să intervină ulterior stabilirii intervalelor, sistemul de iluminare luând singur deciziile pe baza dorințelor și nevoilor acestora.

Deși realizat la o scară mică, proiectul, demonstrându-și eficiența și utilitatea în viața de zi cu zi, poate fi extins și implementat la scară mai mare, în cadrul încăperilor de diverse dimensiuni. Adaptabilitatea acestui sistem automat îl fac să se potrivească în diverse contexte, de la locuințe rezidențiale, la spații comerciale, clădiri de birouri.

III. Resurse hardware

În ceea ce privește alegerea resurselor hardware, documentarea a jucat un rol cheie. Am ales să folosim următoarele componente:

1. Microcontroler Intel Galileo- Microcontroler ușor de utilizat, fiind compatibil cu pinout-ul Arduino și cu mediul de dezvoltare software Arduino. [3] Pe lângă acestea, Galileo oferă anumite beneficii extra, ce ajută dezvoltatorii în vederea realizării diverselor proiecte mai complexe, precum: USB (Universal Serial Bus-Bus Serial Universal) 2.0 client/gazdă, procesor cu frecvență de 400MHz, slot mini PCI-Express(Peripheral Component Interconnect-Interconectarea componentelor periferice), port Ethernet de 100 MB. Toate acestea sunt puse la dispoziție utilizatorilor prin accesarea librăriilor specifice din IDE-ul (Integrated development environment-Mediu de dezvoltare) Arduino.
2. LED-uri: Cunoscute și sub denumirea de diode electroluminiscente, acestea sunt diode semiconductoare de joncțiune p-n special dopată și poartă acest nume datorită faptului că, la polarizarea directă, aceasta emite lumină monocromatică. [4] Durata de viață a acestora este de lungă durată, acestea neavând un filament care să poată să ardă, așa cum se întâmplă în cazul becurilor obișnuite. În cadrul proiectului de față, s-au folosit 3 tipuri de LED-uri, care emit 3 culori, cu semnificații diferite și importante în ceea ce privește interpretarea rezultatelor obținute:
 - 2.1.LED verde- alcătuit din aliaj AlGaP (Aluminiu-Galiu-Fosfor) ne indică faptul că în încăperea în cadrul căreia se realizează măsurătoarea și este conectat circuitul, lumina este favorabilă, drept urmare microcontrolerul este într-o stare de repaus, nu comandă.
 - 2.2.LED galben- format din aliaj GaAsP (Galiu-Arsen-Fosfor) arată faptul că intensitatea luminoasă este mult prea slabă, sub nivelul de prag ales, setat în cadrul configurației software, și, din acest motiv urmează comanda de aprindere a becului, a lămpii utilizate.
 - 2.3.LED roșu- alcătuit din aliaj de tip AlGaAs (Aluminiu-Galiu-Arsen), ne oferă informația că în cadrul încăperii în care se măsoară intensitatea luminoasă, aceasta este peste nivelul de prag setat, este prea puternică, drept urmare becul pe care microcontrolerul îl comandă se va stinge.
3. LCD 16x2- Ecran capabil să afișeze text pe două rânduri, cu câte 16 caractere pe fiecare rând, util în afișările intermediare din cadrul codului, al proiectului, cât și pentru afișările finale. Acesta se folosește de protocolul de comunicare I2C, care utilizează doar două linii, bidirecționale, denumite SDA(Serial Data Line-Linie de date serială) și SCL(Serial Clock Line-Linie de ceas serial). [5] Datele seriale (SDA) facilitează transferul datelor, pe când ceasul serial (SCL) poartă semnalul de ceas, așa cum sugerează și denumirea sa. Protocolul funcționează în 2 moduri: master sau slave, și, indiferent de modul de funcționare, fiecare bit de date transferat pe linia SDA e sincronizat de un impuls de pe linia SCL.

Datele care sunt transmise în această manieră, sub formă de pachete, sunt alcătuite din 9 biți, secvența fiind următoarea:

- Condiția de început- 1 bit;
- Adresa Slave- 8 biți;
- Confirmare- 1 bit.

Condițiile de început și de sfârșit pot fi generate astfel: pentru condiția de start, menținând linia SCL la un nivel înalt și schimbând nivelul de SDA, iar pentru condiția de stop, nivelul liniei SDA ajunge la un nivel scăzut, iar SCL se menține la unul înalt. În ceea ce privește bitul de citire, respectiv de scriere, este important de menționat faptul că un bit de scriere/citire care se află la nivel înalt arată faptul că masterul trimite date slave-ului, în timp ce invers, masterul primește datele de la slave.

4. Modul releu 1 canal, 5V- Modul compatibil cu Arduino, care ajută la preluarea controlului aparatelor care funcționează la tensiune mare. Din punct de vedere tehnic, prezintă următoarele caracteristici:

- Dimensiuni mici, ușor de integrat: 50 x 26 x 18.5 mm
- Tensiune: 5V
- LED-uri care ne indică starea de funcționare a releului
- Un releu cu un canal

Acesta joacă un rol important în ceea ce privește funcționalitatea proiectului din cazul de față, având în vedere faptul că poate fi folosit pentru a controla iluminarea, cu ajutorul microcontrolerului. Fiind de dimensiuni scăzute, este ideal în ceea ce privește adăugarea lui în cadrul proiectului.

5. Sursa de alimentare în comutație AC-DC 220V la 5V, 2A- Utilă în ceea ce privește portabilitatea echipamentului ce urmează să fie alimentat fie de la rețea, fie de la baterie, fără a se crea complicații în ceea ce privește adăugarea unui volum suplimentar de componente, respectiv din punctul de vedere al siguranței utilizatorului, dat fiind faptul că tensiunea de intrare, care este acceptată dacă este cuprinsă în intervalul de 100, până la 240 V în curent alternativ, este transformată într-o tensiune mult mai mică de ieșire, de curent continuu, de 5V.

Această sursă de alimentare este compatibilă cu modelele Arduino și are dimensiuni relativ mici, ceea ce se dovedește a fi în avantajul nostru.

6. 3 rezistori: 10kΩ- Utilizați pentru a proteja curentul care intră în diodele LED ce urmează să fie aprinse în momentul executării condițiilor software. Este legată câte un rezistor de această valoare de fiecare LED, reușind să îndeplinească obiectivul menționat.

Handwritten calculations for LED resistor values:

$V_{CC} = 5V$

$V_{LED-R} = 2V$

$V_{LED-G} = 3V$

$V_{LED-V} = 3V$

$I_{optim-LED} \in [10, 20] mA$ (curs. $I_{optim-LED} = 15mA$)

Formula: $R = (V_{CC} - V_{LED}) / I$

ROȘU: $R = \frac{5-3}{0,015} = 133,33 \Omega$

VERDE + GALBEN: $R = \frac{5-2}{0,015} = 200 \Omega$

Figura 1: Modalitatea de calcul a rezistențelor alese pentru protecția LED-urilor

Cu toate acestea, am ales în cadrul implementării din cazul de față o valoare standardizată, mult mai mare față de valorile calculate, cu obiectivul de a reduce mult curentul prin LED-uri, în așa fel încât să putem asigura o durată de viață mai lungă, o protecție optimă a acestora, care să poată oferi totuși vizibilitate în continuare.

7. 3 rezistori 2kΩ-Utilizați în scopul de a crea, împreună cu foto-rezistorii aleși, divizoare de tensiune capabile s

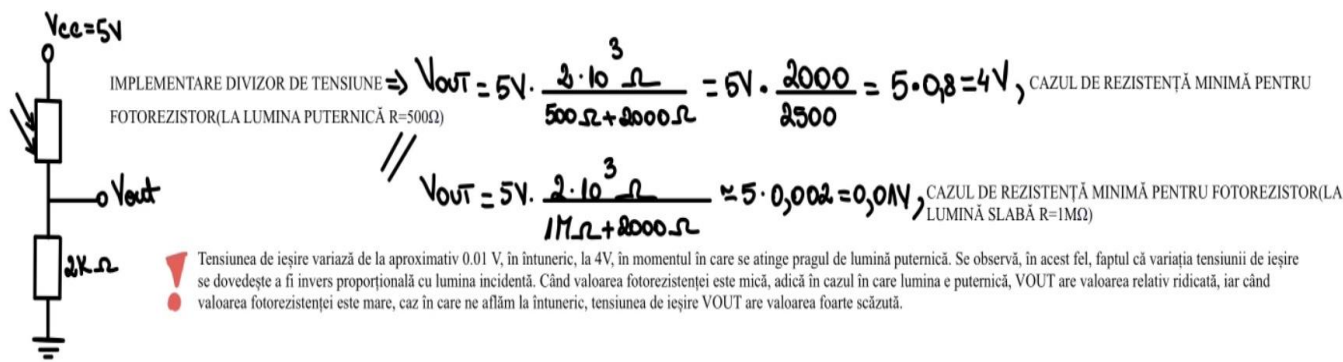


Figura 2: Modalitatea de calcul a rezistențelor alese în vederea obținerii divizorilor de tensiune

În acest fel, oferim circuitului proprietatea de eficiență din punct de vedere energetic, valorile alese pentru realizarea divizorilor nu consumă multă putere din sursa de alimentare acestuia.

8. Foto-rezistor: Reprezintă un dispozitiv semiconductor electronic, pasiv, a cărui rezistență variază invers proporțional cu intensitatea luminoasă. Funcționalitatea acestuia se bazează pe efectul fotoelectric intern, pe creșterea conductivității unui material semiconductor datorită generării de purtători de sarcină suplimentari sub influența radiației luminoase. Când semiconductorul este iluminat, conductivitatea crește proporțional cu numărul de fotoni absorbiți. [6]
9. Cabluri alimentare & de conexiune- Folosite în vederea realizării conexiunilor electrice, s-au utilizat o varietate de cabluri : tată-tată, care prezintă pini la ambele capete, tată-mamă, care prezintă pin într-un capăt și conector la celălalt, precum și cablu de alimentare neizolat, care ne permite realizarea conexiunii între lampă și sursa de alimentare, priză.

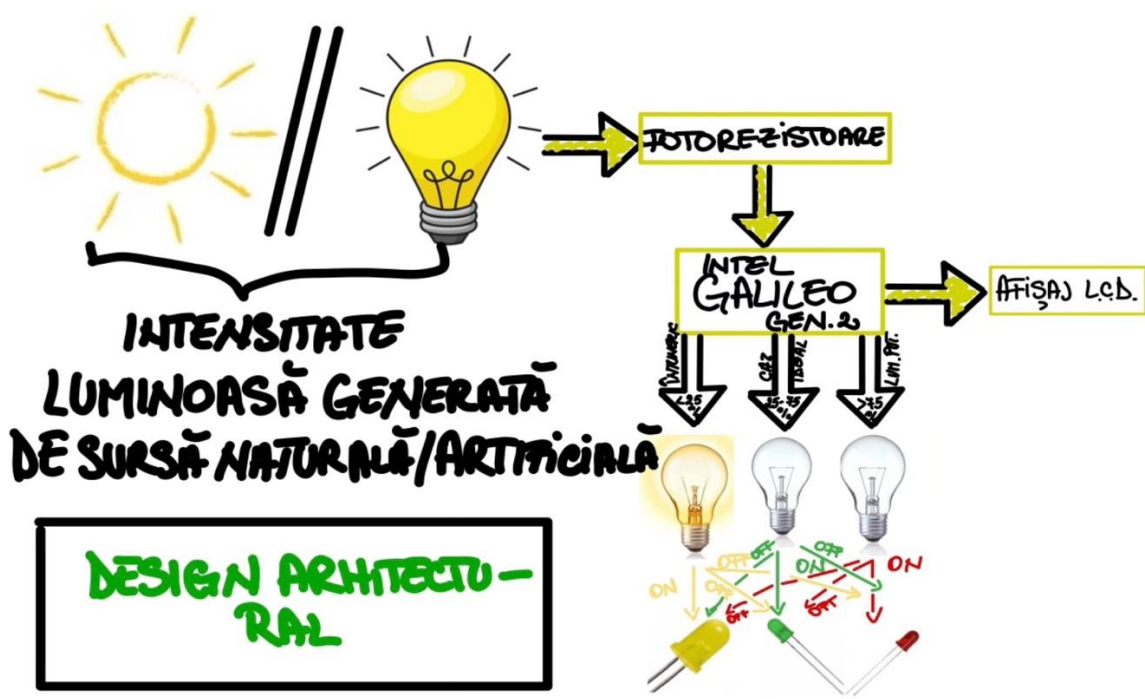


Figura 3: Design-ul architectural al sistemului de monitorizare & control al iluminatului dintr-o încăpere

IV. Resurse software-librării externe si mediul dezvoltare

Resursele software sunt reprezentate de către programe, biblioteci de cod, fișiere de configurare, precum și alte componente digitale ce au ca scop dezvoltarea, gestionarea și implementarea aplicațiilor ce urmează a fi implementate. În cazul de față, IDE-ul (mediul integrat de dezvoltare) utilizat a fost Arduino.

Acesta se dovedește a fi o platformă open-source utilizată în vederea construirii unei multitudini de proiecte electronice. Acest IDE este o aplicație disponibilă pe o varietate de platforme, precum GNU/Linux, mac OS sau Windows și este utilizat în vederea scrierii și încărcării programelor, sau schițelor pe plăci compatibile cu Arduino sau de la alți furnizori, cu ajutorul nucleelor terțe. [7]

Programele Arduino pot fi scrise în orice limbaj de programare care dispune de un compilator capabil să genereze cod mașină binar. Atmel oferă un mediu de dezvoltare pentru microcontrolerele sale, numit AVR Studio, iar mai recent, Atmel Studio.[8][9]

Proiectul Arduino furnizează un mediu integrat de dezvoltare (IDE), o aplicație cross-platform scrisă în Java. Acest mediu își are rădăcinile în mediul de dezvoltare pentru limbajul Processing și în proiectul Wiring. A fost conceput pentru a facilita accesul artiștilor și al celor nefamiliarizați cu dezvoltarea software la programare. IDE-ul include un editor de cod cu funcții precum evidențierea sintaxei, potrivirea acoladelor și spațierea automată, și oferă mecanisme simple de compilare și încărcare a programelor pe placa Arduino, printr-un singur click. Un program scris în IDE pentru Arduino se numește sketch.

Arduino IDE suportă limbajele de programare C și C++, folosind reguli speciale de organizare a codului. IDE-ul include o bibliotecă software numită Wiring, derivată din proiectul Wiring, care furnizează multe proceduri comune de intrare și ieșire. Un sketch tipic Arduino scris în C/C++ este compus din două funcții, care sunt compilate și legate cu un stub al funcției main(), într-un program executabil cu o execuție ciclică:

- setup(): o funcție rulată o singură dată la începutul programului, pentru inițializarea setărilor.
- loop(): o funcție apelată în mod repetat până la oprirea alimentării cu energie a plăcii.

În esență, acest mediu de dezvoltare oferă o modalitate accesibilă tuturor și versatilă în ceea ce privește crearea și programarea diverselor dispozitive și proiecte din domeniul electronicii, fiind utilizat într-o varietate de aplicații, de la configurarea anumitor mașini, motoare la sisteme de monitorizare și automatizare. Datorită naturii open-source, specificațiile schemelor, ale schițelor sunt disponibile oricărui pasionat de acest domeniu, care vrea să se implice sau să dezvolte mai departe.

În același timp, acest avantaj pe care mediul de dezvoltare îl deține aduce cu el și existența librăriilor specifice anumitor componente, care fac utilizarea lor mai ușoară, facilitând timpul utilizatorilor. În cazul de față, am ales utilizarea bibliotecii suplimentare <LiquidCrystal.h>, ce permite comunicarea cu afișaje alfanumerice cu cristale lichide, mai exact cu LCD-uri. [10]

Această bibliotecă se dovedește a fi capabilă să permit unei plăci de tip Arduino sau Genuino să poată controla aceste LCD-uri bazate pe chipset-ul Hitachi HD44780 sau unul compatibil, regăsit în majoritatea dispozitivelor de acest fel.

Biblioteca poate funcționa atât în modul de 4 biți, cât și în cel de 8, este compatibilă cu toate arhitecturile, astfel încât poate fi utilizată pe toate plăcile tip Arduino. Aceasta vine însoțită de o varietate de funcții, mai exact 20, dintre care noi am utilizat un număr de patru funcții în vederea diminuării timpului și dificultății și eficientizării algoritmului: LiquidCrystal(), begin(),setCursor(), print().

În acest fel, am observant următoarele aspect în ceea ce privește funcționalitatea acestora:

- `LiquidCrystal()`: Reprezintă o funcție care creează o variabilă de tipul `LiquidCrystal`. Afișajul poate fi controlat folosindu-ne de 4 sau 8 linii de date. În cazul în care sunt folosite 4 linii, se omit numerele de pin pentru d0, până la d3, fiind lăsate neconectate. De obicei, de la d0 la d7 se notează numerele pinilor Arduino conectați la pinii de date corespunzători LCD-ului. Sintaxa de apel a funcției poate lua mai multe forme, în funcție de nevoile fiecărui utilizator, spre exemplu:
 - ◆ `LiquidCrystal(rs, enable, d4, d5, d6, d7)`
 - ◆ `LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)`
 - ◆ `LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)`
 - ◆ `LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)`,
unde rs este pinul „register select”, care controlează unde în memoria LCD-ului se scriu datele, iar rw este pinul de „read/write”.
- `begin()`: Funcția inițializează interfața cu ecran LCD, specificând dimensiunile: lățimea și înălțimea afișajului, mai exact numărul de coloane, linii și opțional numărul de puncte pe care le are fiecare caracter al LCD-ului. Sintaxa de apel a acesteia poate fi sub forma:
 - ◆ `lcd.begin(cols, rows, charsize)`,
unde „lcd” este o variabilă de tip `LiquidCrystal`, „cols” reprezintă numărul coloanelor afișajului, „rows”, numărul liniilor acestuia, iar „charsize” variabila opțională.
- `setCursor()`: Funcția din acest caz are rolul de a poziționa cursorul LCD, setând locația la care textul ce urmează să fie scris pe ecran va fi afișat. Sintaxa de apel a acestuia este de forma:
 - ◆ `lcd.setCursor(col, row)`,
unde „lcd” este variabilă de tip `LiquidCrystal`, „col” reprezintă coloana la care urmează să fie poziționat cursorul, iar „row” rândul la care să fie poziționat cursorul, ambele fiind inițializate cu 0.
- `print()`: Realizează tipărirea textului pe afișaj, fiind apelat prin sintaxe precum:
 - ◆ `lcd.print(data)`
 - ◆ `lcd.print(data, base)`,
unde „lcd” reprezintă variabilă de tipul `LiquidCrystal`, „data” reprezintă datele care trebuie să fie tipărite, ce poate reprezenta un caracter, un byte/int/long sau un șir de caractere, iar variabila „base” este opțională, reprezentând baza în care s-ar putea tipări numerele și poate fi de tip BIN(pentru baza 2), DEC(pentru baza 10), OCT(pentru baza 8) sau HEX(pentru baza 16).

V. Implementare hardware

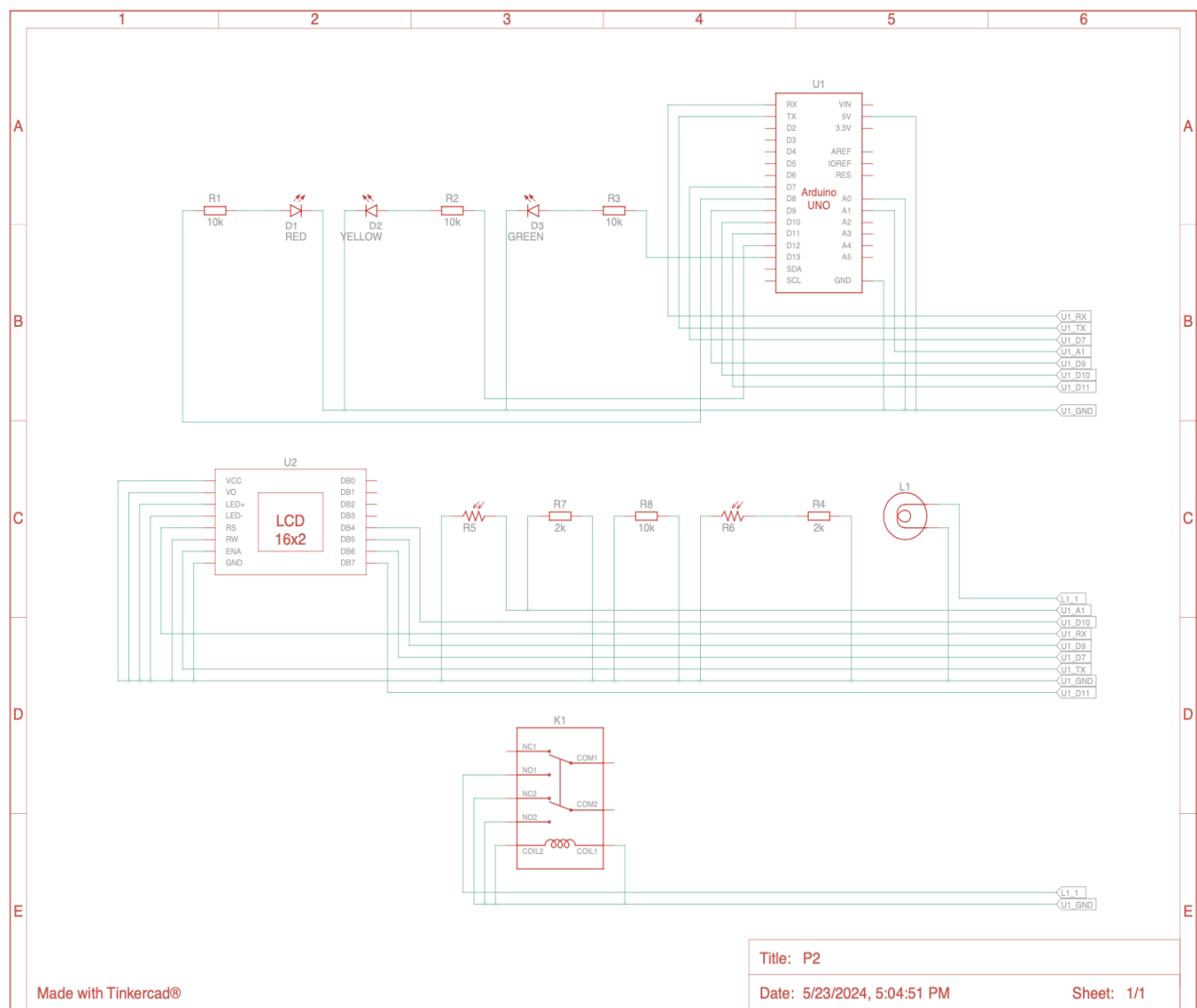


Figura 4: Schema electrică de funcționare a sistemului de monitorizare & control al iluminatului într-o încăpere

Foto-rezistorul reprezintă punctul de plecare al funcționalității proiectului. Acesta măsoară în lăcuși intensitatea luminoasă care se regăsește într-o încăpere, furnizând în acest fel informații esențiale în ceea ce privește tema abordată. Rezistența acestora are o valoare foarte mare în întuneric, aproape de $1\text{M}\Omega$, însă atunci când lumina cade pe LDR(Light Dependent Resistor: foto-rezistor), rezistența scade la câțiva $\text{k}\Omega$ ($10\text{-}20\text{k}\Omega$ la 10 lux , $2\text{-}4\text{k}\Omega$ la 100 lux) sau chiar mai puțin.[11] Acesta reprezintă o componentă pasivă cheie în cazul circuitului de față, fără foto-rezistori nefiind posibilă realizarea proiectului. , cu ajutorul rezistențelor de $2\text{k}\Omega$, se formează un divizor de tensiune, ce ajută la adaptarea, stabilizarea semnalului furnizat de către foto-rezistoare. Semnalul rezultat la ieșirea acestuia este astfel citit în mod analog de către pinii A0 și A1 ai microcontrolerului Intel Galileo. În acest fel, se permite monitorizarea și ajustarea în funcție de nevoile ambientale ce urmează a fi detaliate.

În ceea ce privește alimentarea breadboardului, se conectează pinii de la microcontroller: 5V , pentru a furniza tensiunea necesară funcționării acestuia, respectiv ground (GND), pentru stabilirea referinței de potențial. De la pinul de 5V , un fir este direcționat către releu, conectat la outputul digital 12 , ce are rolul de control al lămpii pe care urmează să o aprindem, stingem sau să o menținem în poziția curentă. Una dintre cele trei decizii cheie menționate anterior este luată în funcție de parametrii A0 și A1, ce reflectă nivelul de iluminare detectat. De asemenea, led-urile sunt conectate prin fire la output-urile digitale 5-7 ale microcontrolerului, permițându-se, în acest fel, controlul lor în funcție de algoritmul de programare specific, astfel: dacă intensitatea luminoasă detectată în încăpere este peste nivelul stabilit în cadrul programului, atunci se va aprinde LED-ul roșu și, ca efect, lampa se va stinge. În cazul în care nivelul intensității luminoase este sub nivelul de prag setat, se va aprinde LED-ul galben, iar lampa se va aprinde. În cazul în care intensitatea luminoasă este la un nivel ambiental favorabil, starea lămpii va rămâne aceeași, fără să se aprindă sau stingă, iar LED-ul care se va aprinde în acest caz este cel de culoare verde.

LED-ul verde este legat la pinul digital 7 considerat output , LED-ul galben la pinul digital 6 considerat output , iar LED-ul roșu la pinul digital 5 .

Suplimentar, schema electrică prezentată în figura anterioară sugerează faptul că se realizează comunicarea cu afișajul LCD, care, cu ajutorul librăriilor, funcțiilor descrise anterior, ne furnizează nouă, utilizatorilor, informații cu privire la nivelul intensității calculate din încăperea în care ne aflăm.

Știind faptul că un foto-rezistor are capacitatea de a varia între valorile de 500Ω și până la $1M\Omega$, în cadrul implementării am ales să creăm scara de valori pe baza formulei: $\frac{A0+A1-150}{150} * 100$, secționând astfel:

- În intervalul 0-25%, intensitatea căpătată este prea mică, de aceea se va aprinde LED-ul galben ce sugerează acest lucru.
- În intervalul 25.1%-75.00%, intensitatea căpătată este la un nivel rezonabil, drept pentru care se va aprinde LED-ul verde.
- Dacă intensitatea depășește valoarea de 75.1%, atunci ne aflăm în cazul în care LED-ul roșu este cel ce urmează să se aprindă.

LCD-ul, în cazul de față, este folosit în acest fel:

- Pinii digitali 8-11 sunt folosiți pentru datele de intrare pe 4 biți;
- Pinii digitali 0-1 sunt utilizați pentru RS, respectiv Enable;
- Alimentarea VCC: VCC, LED+.

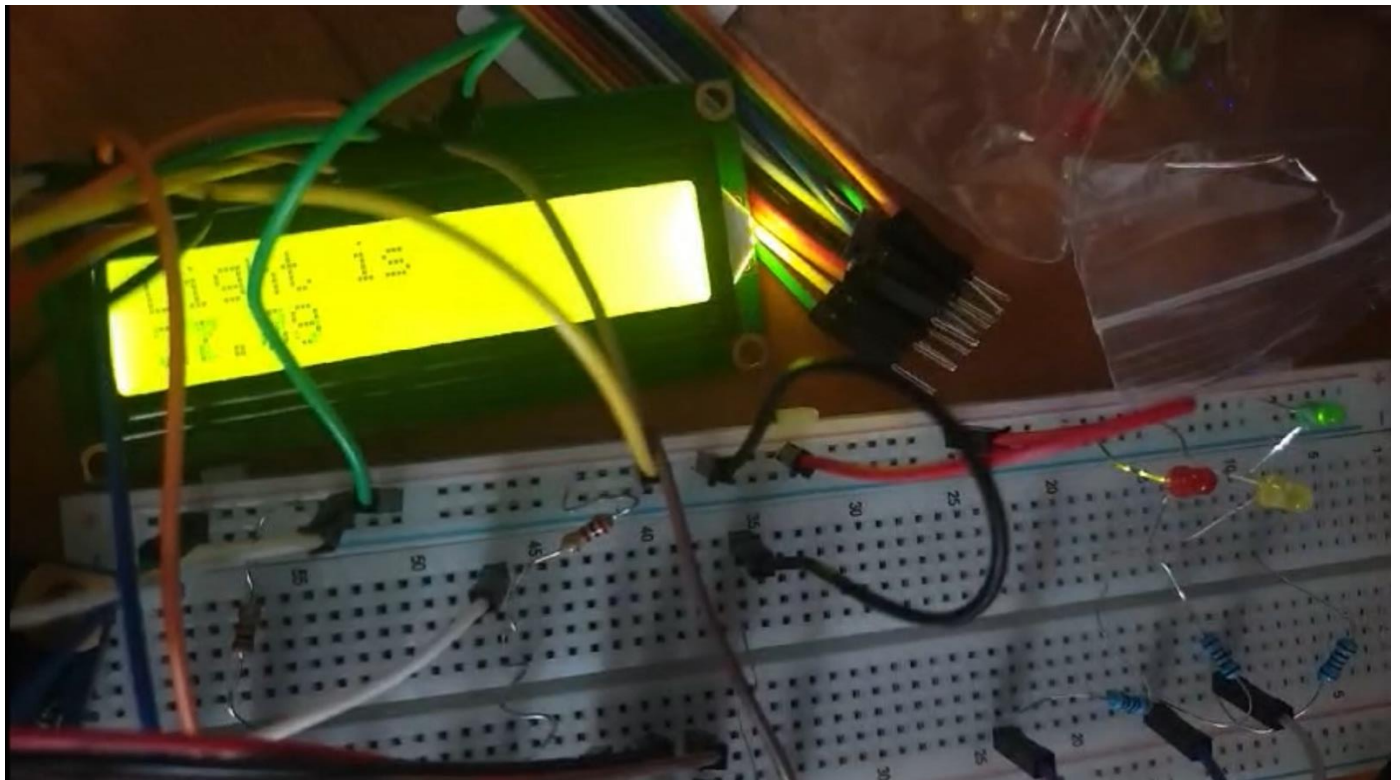


Figura 5: Testarea funcționalității senzorului de intensitate luminoasă din încăperea

VI. Implementare software

În cadrul prezentului capitol, urmează să detaliez codul pe baza căruia proiectul și-a dovedit funcționalitatea. În primul rând, vom începe cu atașarea logicii din spatele acestuia, și anume organigrama, astfel:

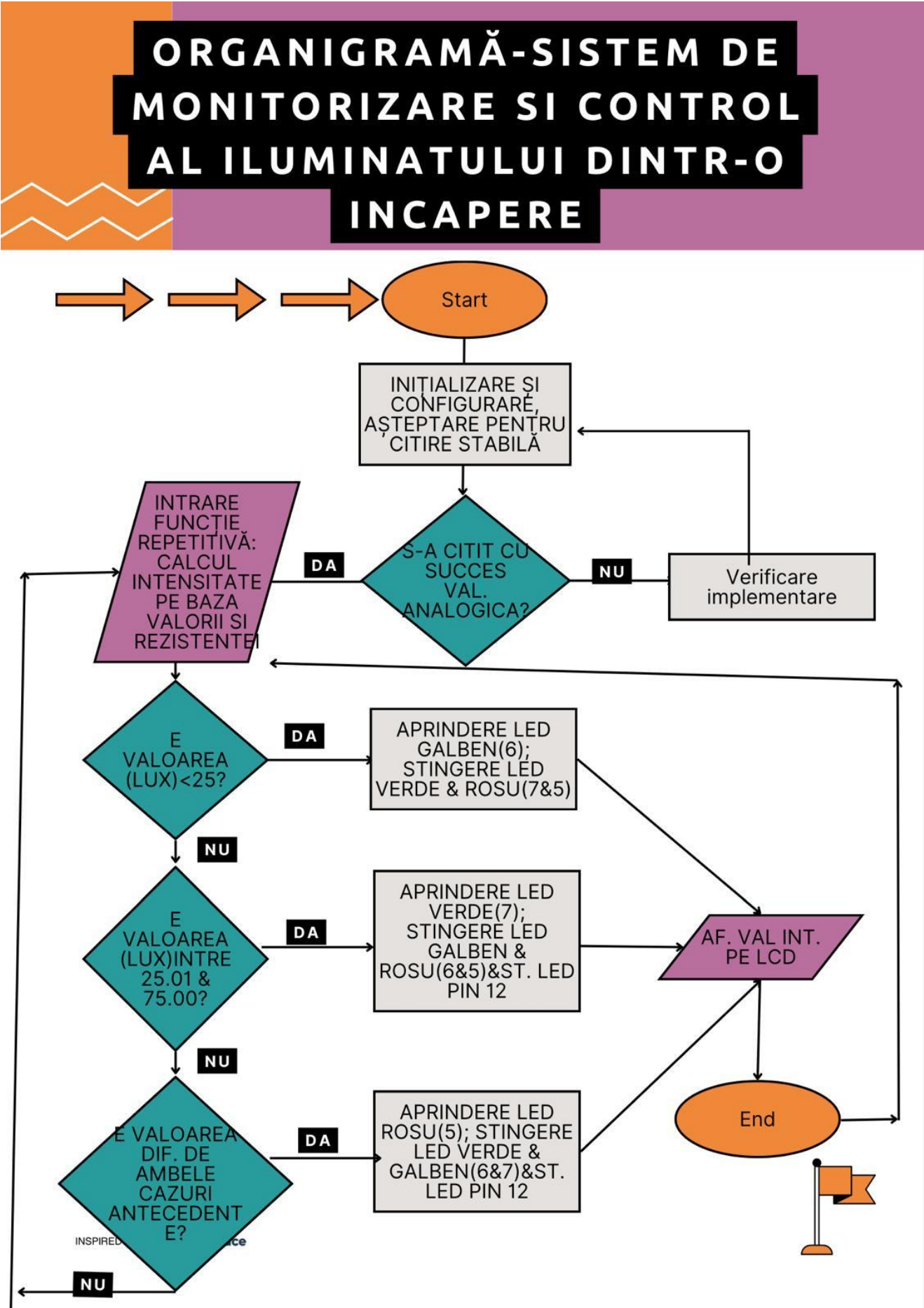


Figura 6: Organigrama codului

Algoritmul de funcționare al proiectului se regăsește sub următoarea formă:

```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(0, 1, 8,9,10,11); /// REGISTER SELECT PIN,ENABLE PIN,D4 PIN,D5
PIN, D6 PIN, D7 PIN

void setup()
{
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(12, OUTPUT);
  lcd.begin(16, 2);
  Serial.begin(9600);

}

void loop()
{

delay(750); //delay of 0.75sec

lcd.setCursor(0, 0); // set the cursor to column 0, line1

  double analogVal = analogRead(A3); //read values from A0
  double analogVal1= analogRead(A5);
  Serial.println(analogVal1);
  delay(100);
  Serial.println(analogVal);

  lcd.print("Light is " );
  lcd.setCursor(0,1);
  lcd.print (((analogVal+analogVal1-150)/140)*100);


  double lux=((analogVal+analogVal1-150)/140)*100;
  if(lux<25.00)
  {digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(5,LOW);
    digitalWrite(12,HIGH);
    delay(100);
  }
  else
  if(lux>=25.01&&75<=lux)
  {digitalWrite(7,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(12,LOW);
```

```

    }
    else
    {digitalWrite(5,HIGH);
      digitalWrite(6,LOW);
      digitalWrite(7,LOW);
      digitalWrite(12,LOW);
    }

    delay(10);
}

```

În prima secțiune a codului, denumită „setup”, vom inițializa configurația necesară în vederea funcționării corecte a circuitului realizat. Începem prin a configura pinii 5, 6, 7, 12, ce vor fi folosiți ulterior, cu scopul de a controla LED-urile, respectiv becul conectat. Cele 3 LED-uri colorate: galben, conectat la pinul 6; roșu, conectat la pinul 5; verde, conectat la pinul 7 ne ajută să vizualizăm intervalul de intensitate luminoasă în care ne aflăm în momentul citirii. Utilizând biblioteca „Liquid Crystal”, reușim să inițializăm ecranul LCD 16x2 descris în cadrul punctelor anterioare. Aceasta este esențială în ceea ce privește comunicarea și interacționarea cu ecranul, oferind, în acest fel, posibilitatea de a afișa informațiile într-un mod eficient și ușor de înțeles pentru utilizatori.

În interiorul funcției repetitive „loop()”, implementăm logica principală a programului. Începem prin a realiza o scurtă așteptare de 0.75 secunde pentru a ne asigura că citirea de la foto-rezistor este stabilă. După aceasta, setăm cursorul LCD-ului pe primul rând, prima coloană, pregătindu-ne să afișăm datele.

În cadrul funcției repetitive „loop()” se realizează o scurtă așteptare, de 0.75 secunde în vederea asigurării unei citiri stabile a foto-rezistorului, după care urmează setarea cursorului pe primul rând, prima coloană a ecranului.

Valoarea citită de la foto-rezistor este de natură analogică și provine de la pinul A3, reflectând intensitatea luminoasă din încăpere. Această valoare analogică este esențială pentru determinarea nivelului de lumină ambientală din cadrul încăperii în care dorim să realizăm monitorizarea și controlul.

Funcția „calcLightIntensity()” are rolul de a converti valoarea analogică citită în unități de măsură standardizate, apoi în procente. Această conversie se realizează pe baza unei formule care ține cont de tensiunea de ieșire a foto-rezistorului și de rezistența acestuia.

În funcție de valoarea procentuală a intensității luminoase calculate, luăm decizii privind controlul LED-urilor. Dacă intensitatea este sub un anumit prag, activăm anumite LED-uri și dezactivăm altele pentru a indica intervalul de lumină ambientală. De exemplu, dacă intensitatea este sub 25.00%, aprindem LED-ul galben, corespunzător pinului 6, indicând un nivel prea scăzut de lumină, și stingem celelalte LED-uri. Dacă intensitatea este între 25.01 și 75.00%, aprindem LED-ul verde, ce corespunde pinului 7 și le vom stinge pe celelalte. În caz contrar, dacă intensitatea nu se încadrează în niciuna dintre aceste două categorii, aprindem LED-ul roșu, conectat la pe pinul 5. În primul caz, se va aprinde becul din încăpere, pe când în celelalte două becul va fi stins.

Codul se dovedește a fi, în acest fel, de complexitate scăzută, oferind simultan o privire de ansamblu asupra modului de utilizare a unui microcontroler Intel Galileo Gen. 2, în scopul măsurătorii, controlului și afișajului intensității luminoase regăsite în cadrul unei încăperi. Astfel, se observă faptul că utilizarea bibliotecilor speciale, precum și configurarea corectă a pinilor asigură funcționarea optimă a circuitul astfel conceput. În final, programul de față reușește să ofere măsurători precise ale intensității luminoase și oferă, în același timp și un feedback vizual, util pentru proiectul din cazul de față, cât și pentru dezvoltarea ulterioară a unei game vaste de aplicații practice.

VII. Concluzii

În vederea redactării punctuale a concluziilor, vom împărți capitolul în trei secțiuni separate, principale, care la rândul lor vor cuprinde subpunctele necesare în vederea evidențierii aspectelor importante ale proiectului, precum și în vederea oferirii unei direcții pentru viitor, ce vizează îmbunătățirea și eficientizarea acestuia.

VII.1. Realizări & beneficii

- Din punctul de vedere al implementării proiectului, am reușit să oferim, în perspectivă personală, o descriere detaliată a acestuia, a funcționalității sale;
- Am reușit să utilizăm în mod eficient resursele hardware și software de care am putut dispune, printre care s-au numărat mediul de dezvoltare Arduino, biblioteca <LiquidCrystal.h>, respectiv foto-rezistori, ecran LCD, LED-uri, microcontroler Intel Galileo, rezistorii.
- În ceea ce privește beneficiile aduse de către punerea în funcționare a ideilor din cadrul prezentului proiect, putem observa economisirea energiei și protejarea mediului înconjurător drept numărându-se printre cele mai importante. În același timp, se poate constata o îmbunătățire per total a calității vieții utilizatorului, prin lipsa stresului de a închide toate circuitele în momentul părăsirii locuinței sau a locului de muncă și ajustarea nivelului de lumină în funcție de propriile gusturi.

VII.2. Rezultate & observații

- Din perspectiva eficienței, ne putem gândi la adaptabilitatea proiectului la diverse condiții, contexte de trai, de la utilitatea în mediul urban, rural, de la locuințe rezidențiale la clădiri de birouri sau spații comerciale, toate acestea pot beneficia de astfel de îmbunătățiri, de automatizări.
- Ar putea exista, pe viitor, anumite extinderi ale prezentului proiect, precum optimizarea algoritmului de control sau expandarea funcționalităților în vederea includerii a mai multor aspecte legate de iluminare, precum afișarea pe telefonul mobil în loc de ecranul LCD al nivelului intensității.

VII.3. Perspective

- În viitor se pot explora posibilitățile de extindere a proiectului, la scară mai mare și în diverse medii, precum și cercetarea și dezvoltarea continuă a diverselor tehnologii de monitorizare și control al iluminatului, în vederea obținerii îmbunătățirii continue.
- Utilizarea perspectivelor, feedback-ului și experienței utilizatorilor în vederea îmbunătățirii și optimizării sistemului în direcția dorită, precum și continuarea colaborării între domeniile electronicii și automatizării ar putea aduce cu sine reale progrese în ceea ce privește proiectul de față.

În concluzie, proiectul nostru a reușit să demonstreze eficiența și utilitatea unei soluții automate în vederea realizării controlului și monitorizării iluminatului dintr-o încăpere. Prin integrarea unui circuit electronic simplu, format din componente hardware și software, am reușit să realizăm un sistem capabil să ajusteze intensitatea luminii în funcție de nivelul ambiental detectat. Utilizând un microcontroler Galileo, LED-uri și un foto-rezistor, am reușit să creăm un sistem versatil și eficient, beneficiile acestui proiect fiind

numeroase și variate. Totodată, adaptabilitatea și extensibilitatea sistemului permit implementarea sa într-o varietate de medii, așa cum am menționat anterior. Prin intermediul acestuia, am reușit să explorăm și să aplicăm concepte din domeniul electronicii și al programării, dezvoltând în acest sens abilități practice și consolidând cunoștințe teoretice dobândite anterior.

În cele din urmă, proiectul nostru reușește să demonstreze potențialul tehnologiei, acela de a aduce îmbunătățiri semnificative în ceea ce înseamnă viața noastră, de zi cu zi, contribuind la crearea unui mediu mai prietenos cu planeta, mai eficient din punct de vedere energetic și, în același timp, mai confortabil pentru noi.

VIII. Bibliografie

- [1] Oditis, I.; Bicevskis, J. The concept of automated process control. Comput. Sci. Inf. 2010, 756, 193–203.
- [2] Chan, M.; Estève, D.; Escriba, C.; Campo, E. A review of smart homes: Present state and future challenges. Comput. Methods Programs Biomed. 2008, 91, 55–81
- [3] <https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/galileo-g2-datasheet.pdf>
- [4] <https://www.electrokits.ro/ce-este-un-led-si-cum-functioneaza/>
- [5] <https://www.geeksforgeeks.org/i2c-communication-protocol/>
- [6] <https://atom.ubbcluj.ro/katalin/Laborok/Felvezeto/Fotoellenallas-uj.pdf>
- [7] Îndrumar laborator Microcontrolere-Arduino, Sabou Sebastian Petru, UTCluj, 2018
- [8] <https://www.megunolink.com/>”
- [9] <https://web.archive.org/web/20120828135304/http://www.engblaze.com/tutorial-using-avr-studio-5-with-arduino-projects/>
- [10] <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>
- [11] <https://www.electroschematics.com/ldr-light-dependent-resistor-photoresistor/>