

PROIECT CODESIGN HARDWARE/SOFTWARE

Sistem de control al becurilor de pe stradă

Realizat de: Iancu Marius Laurențiu
Karașin Kristian
Laeș Adina

Scopul proiectului

Proiectul are ca scop reducerea consumului de energie prin implementarea unui sistem inteligent care va activa becurile doar atunci când trece o persoană sau o mașină într-o anumită zonă. Acest lucru se realizează prin utilizarea tehnologiilor de detectare a prezenței, cum ar fi senzori de mișcare sau camere de monitorizare.

Funcționare

Sistemul funcționează astfel: atunci când senzorii detectează prezența unei persoane sau a unei mașini într-o anumită zonă, un semnal este trimis către sistemul de control al becurilor. Acest semnal activează becurile în acel moment, permițând iluminarea zonei unde se află persoana sau mașina.

Avantaje

Avantajul acestui sistem este că energia electrică este utilizată doar atunci când este necesară, în loc să fie irosită prin menținerea becurilor aprinse într-o zonă fără prezență umană sau vehicule. Astfel, se reduce consumul total de energie și se obține o eficiență mai mare în utilizarea resurselor.

Pe lângă reducerea consumului de energie, acest sistem are și alte beneficii. De exemplu, poate contribui la creșterea siguranței în zonele iluminate, deoarece prezența activată a becurilor poate descuraja activitățile infracționale. De asemenea, poate aduce un plus de confort utilizatorilor, deoarece becurile se vor aprinde automat în momentul în care aceștia trec prin zonele respective, eliminând necesitatea de a căuta și de a porni manual comutatoarele de lumină.

Resurse utilizate

- Placă Arduino UNO R3
- Senzor de lumină (catalex)
- Senzor de mișcare HC-SR04
- LCD 16 x 2 1602A
- LED
- Rezistor
- Fire

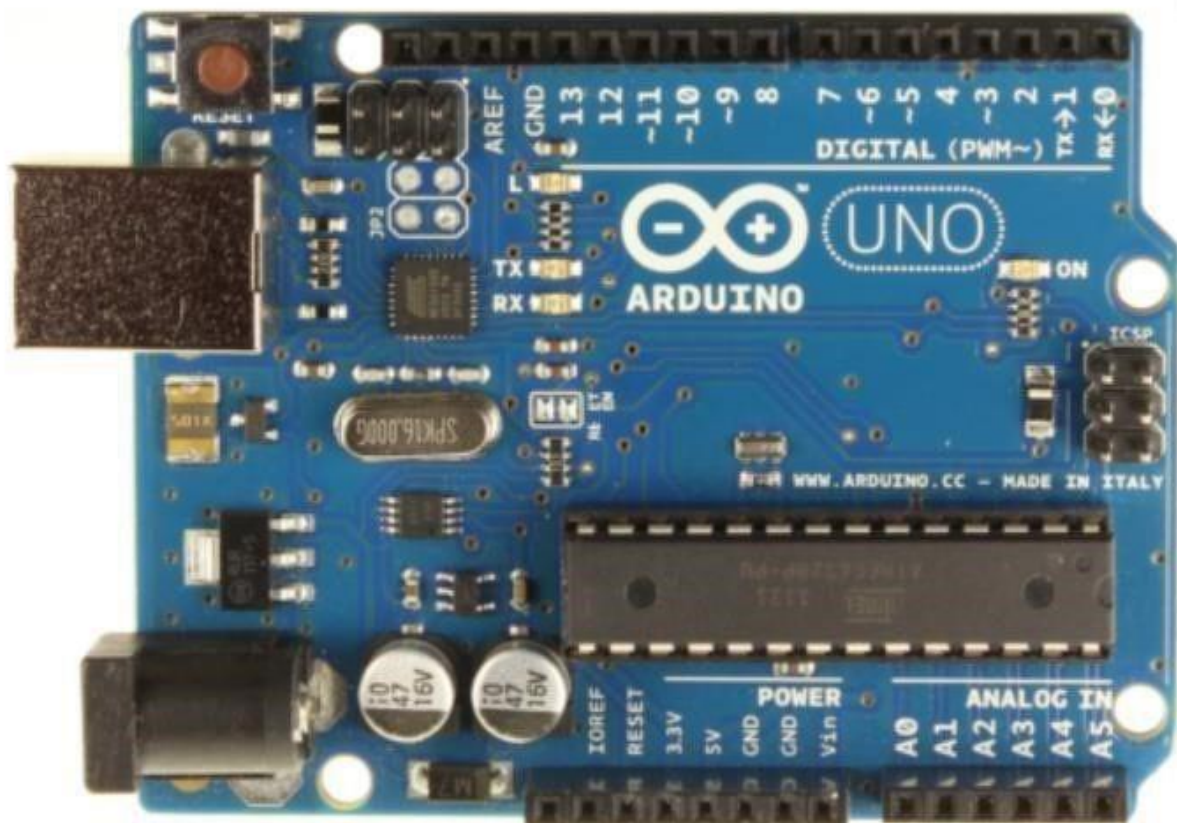
Funcționarea componentelor

1. Placa Arduino Uno R3

Placa de dezvoltare Arduino Uno R3 este echipată cu un microcontroler ATmega328 și are 20 de pini de intrare/ieșire digitală, dintre care 6 pot fi utilizați ca ieșiri PWM (pulse width modulation), iar alți 6 pot fi utilizați ca intrări analogice. Placa dispune de un rezonator de 16 MHz pentru a furniza o sursă de ceas pentru microcontroler. De asemenea, este prevăzută cu o conexiune USB pentru comunicarea cu un computer, o mufă de alimentare și un buton de resetare.

Un aspect distinctiv al plăcii Arduino Uno R3 este că nu utilizează cipul de driver FTDI USBtoserial, care era folosit în versiunile anterioare. În schimb, placa dispune de un microcontroler auxiliar numit ATmega16U2, care este programat să funcționeze ca un convertor USB-to-serial. Acest microcontroler adițional are propriul său bootloader USB, ceea ce facilitează încărcarea programelor pe placa Arduino Uno R3 și comunicarea cu dispozitivele externe prin intermediul portului serial.

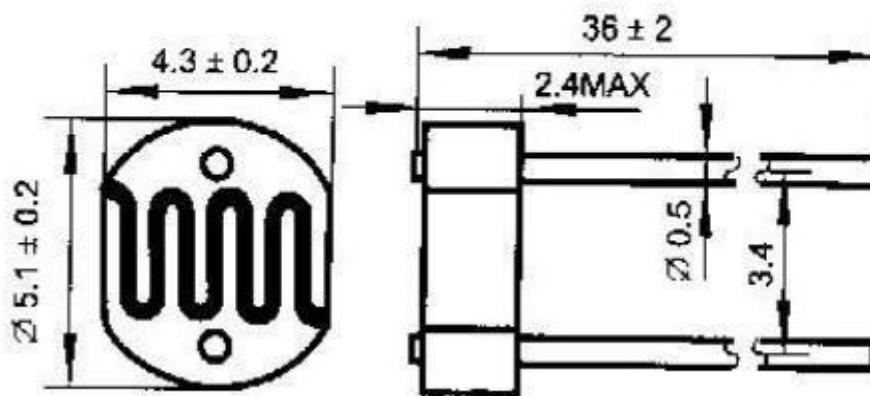
Prin intermediul piniilor de intrare/ieșire digitală și analogică, utilizatorii pot conecta și controla diferite componente electronice, precum senzori, LED-uri și motoare, pentru a crea diverse proiecte interactive. Placa poate fi programată folosind limbajul Arduino și mediul de dezvoltare Arduino IDE, oferind un mod accesibil și ușor de utilizat pentru dezvoltarea proiectelor electronice.



2. Passive Infrared (PIR) sensor

Un senzor de lumină PIR (infraroșu pasiv) este un dispozitiv care utilizează tehnologia infraroșu pasiv pentru a detecta și reacționa la mișcarea obiectelor într-o anumită zonă. Acesta funcționează prin detectarea schimbărilor de radiație infraroșie emisă de obiectele din jur, atunci când acestea se mișcă prin câmpul său de vedere, generând un semnal de ieșire care poate fi utilizat pentru a activa diverse aplicații, cum ar fi sistemele de iluminat automatizat sau sistemele de securitate.

Outline



Light Resistance at 10Lux (at 25°C)	8~20KΩ
Dark Resistance at 0 Lux	1.0MΩ(min)
Gamma value at 100-10Lux	0.7
Power Dissipation(at 25°C)	100mW
Max Voltage (at 25°C)	150V
Spectral Response peak (at 25°C)	540nm
Ambient Temperature Range:	- 30~+70°C

3. Senzor de mișcare HC-SR04

Senzorul HC-SR04 emite ultrasunete și măsoară timpul de propagare al acestora pentru a determina distanța față de un obiect. Această informație poate fi utilizată într-o varietate de aplicații, cum ar fi evitarea obstacolelor, asistența în parcare sau detecția mișcării.



Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

4. Display LCD 16X12 1602A

În esență, un display LCD 16x2 funcționează prin primirea de comenzi și date de la un dispozitiv de control și afișarea acestora pe ecranul cu cristale lichide. Acest tip de display este utilizat într-o gamă largă de aplicații, cum ar fi afișarea mesajelor sau informațiilor în diferite dispozitive electronice.



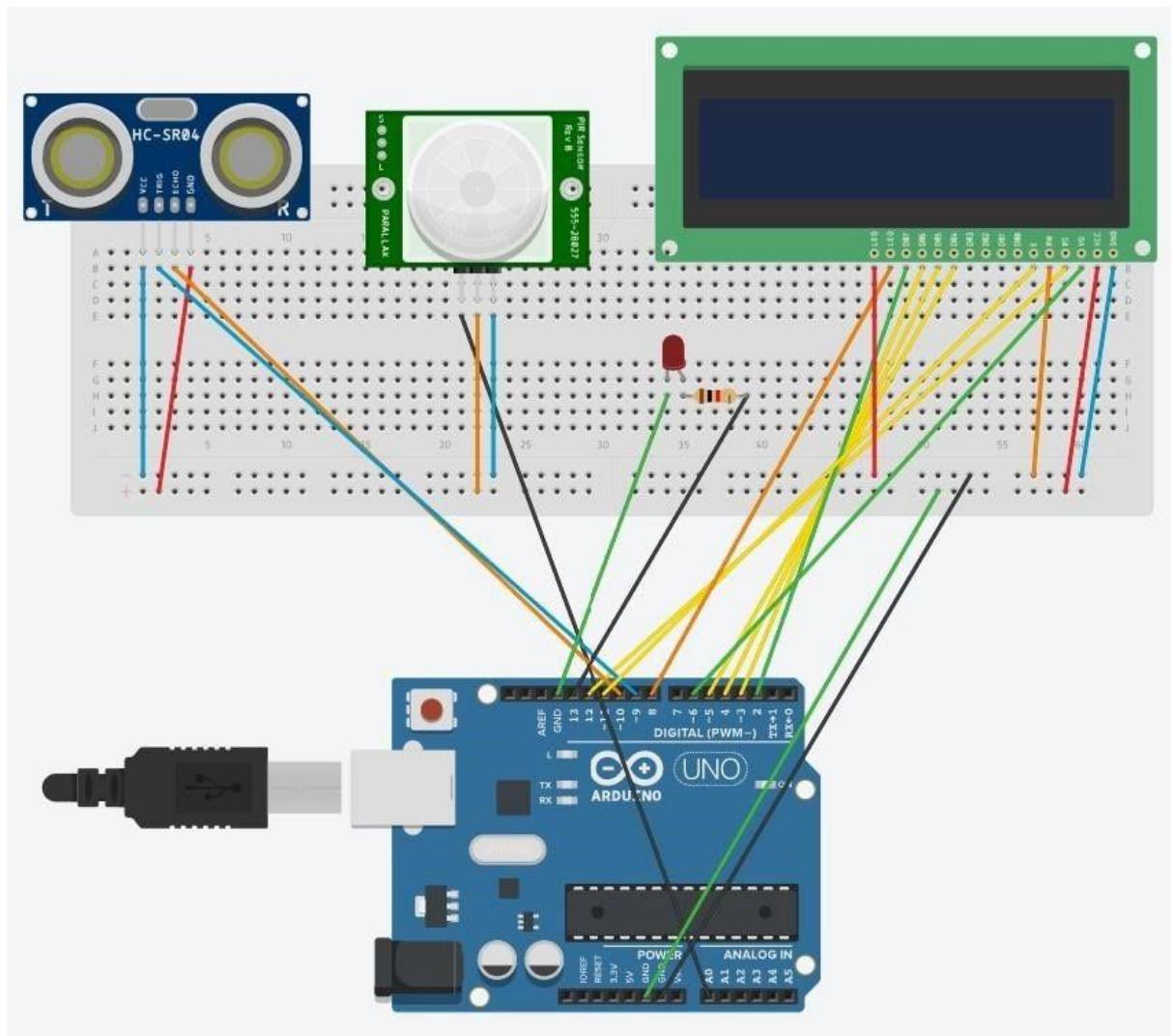
Description

1. Number of Dots : 16 CharacterX2LINE
2. Operating Voltage : 4.8~5.2V
3. Operating Current of LCD : 2.0mA(5.0V)
4. Color of LED Unit : Yellow-Green
5. Current of LED Unit : <100mA
6. Color of LCD unit : Yellow-Green(TN)
7. Operating Temperature: 0~+50°C
8. Storage Temperature: -20~+70°C

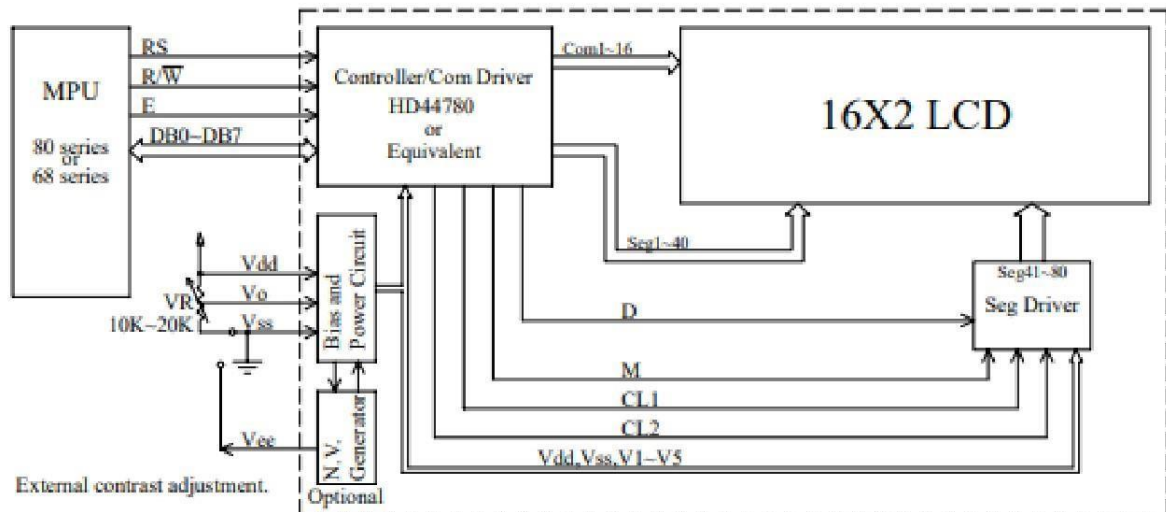
Pinout

1. VSS : GND
2. VDD : Power Supply
3. V0 : Power Supply LCD Driver
4. RS : Register Select-LOW=Instruction,HIGH=Data
5. R/W : Read/Write (LOW=MPU to LCM,HIGH=LCM to MPU)
- 6 E : Enable (R/W=LOW:Data are talking over at falling edge,
R/W=HIGH:Data can be read at E=1)
7. 7 to 14 : DB0 to DB7 Data BUS-Software Selectable 4 or 8bit mode
8. 15 : BLK Cathode of LED Unit
- 9.. 16 : BLA Anode of LED Unit

Schema Hardware



Schema bloc LCD



Cod sursă

```
1  #include <LiquidCrystal.h>
2
3  // Pin configuration for sensors and LED
4  const int PIR_TRIG_PIN = 10;      // Arduino pin connected to PIR sensor's trigger pin
5  const int PIR_ECHO_PIN = 9;       // Arduino pin connected to PIR sensor's echo pin
6  const int LIGHT_SENSOR_PIN = A0;  // Arduino pin connected to light sensor's pin
7  const int LED_PIN = 13;           // Arduino pin connected to LED's pin
8  const int LIGHT_THRESHOLD = 500;  // Light threshold value
9
10 // Initialize the LCD object
11 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
12
13 void setup() {
14     pinMode(PIR_TRIG_PIN, OUTPUT); // Set PIR sensor trigger pin to output mode
15     pinMode(PIR_ECHO_PIN, INPUT);  // Set PIR sensor echo pin to input mode
16     pinMode(LIGHT_SENSOR_PIN, INPUT); // Set light sensor pin to input mode
17     pinMode(LED_PIN, OUTPUT);       // Set LED pin to output mode
18
19     // Setează pinul A ca pin de ieșire
20     analogWrite(6,100);
21
22     // Setează numărul de coloane și rânduri ale LCD-ului
23     lcd.begin(16, 2);
24 }
25
26 void loop() {
27     int lightValue = analogRead(LIGHT_SENSOR_PIN); // Read the input from the light sensor
28     int motionValue = getMotionValue();           // Read the input from the PIR sensor
29
30     if (lightValue >= LIGHT_THRESHOLD || motionValue == LOW) {
31         digitalWrite(LED_PIN, LOW); // Turn off the LED
32         lcd.clear();                // Clear the LCD screen
33     } else {
34         digitalWrite(LED_PIN, HIGH); // Turn on the LED
35         lcd.setCursor(0, 0);         // Set the cursor to the first column, first row
36         lcd.print("DETECTAT");      // Display "DETECTAT"
37     }
38
39     delay(100); // Delay for stability
40 }
41
42 int getMotionValue() {
43     digitalWrite(PIR_TRIG_PIN, LOW);
44     delayMicroseconds(2);
45     digitalWrite(PIR_TRIG_PIN, HIGH);
46     delayMicroseconds(10);
47     digitalWrite(PIR_TRIG_PIN, LOW);
48     return digitalRead(PIR_ECHO_PIN); // Read the input from the PIR sensor
49 }
50 }
```

Bibliografie:

- https://www.electroschemacs.com/wpcontent/uploads/2013/07/HC_SR04-datasheet-version-1.pdf
- <https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/SEN09088.pdf>
- <http://www.datasheet.es/PDF/519148/LCD-1602A-pdf.html>
- <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>