

Sistem automat de numărare a obiectelor pe bandă- Țurcanu Cătălin

Introducere

Acest proiect are ca scop construirea unui sistem simplu, eficient și fiabil care să numere automat obiectele ce trec pe o bandă transportoare, folosind un senzor de obstacole IR. Numărul total este afișat pe un ecran LCD, iar la fiecare obiect detectat se aude un semnal sonor generat de un buzzer. Proiectul salvează automat valoarea contorului în memoria microprocesorului, pentru a păstra datele chiar și după întreruperea alimentării.

Ideea a pornit de la aplicații practice din fabrici și depozite unde este nevoie de un sistem de monitorizare a fluxului de produse fără intervenție umană.

Utilitate:

Practică pentru învățarea folosirii senzorilor, microprocesorului și afișajelor

Aplicabilă în industrie, depozite, educație

Descriere generală

Descriere module: Senzor IR - detectează trecerea obiectului

Arduino Uno - controlează logica de numărare, afișare, memorare și semnalizare

LCD 16x2 - afișează numărul curent de obiecte

Buzzer - emite un sunet la fiecare trecere

Buton - resetează contorul la zero



Hardware Design

Listă componente: 1x Arduino UNO R3

1x Modul senzor IR

1x Display LCD 16x2

1x Buzzer activ

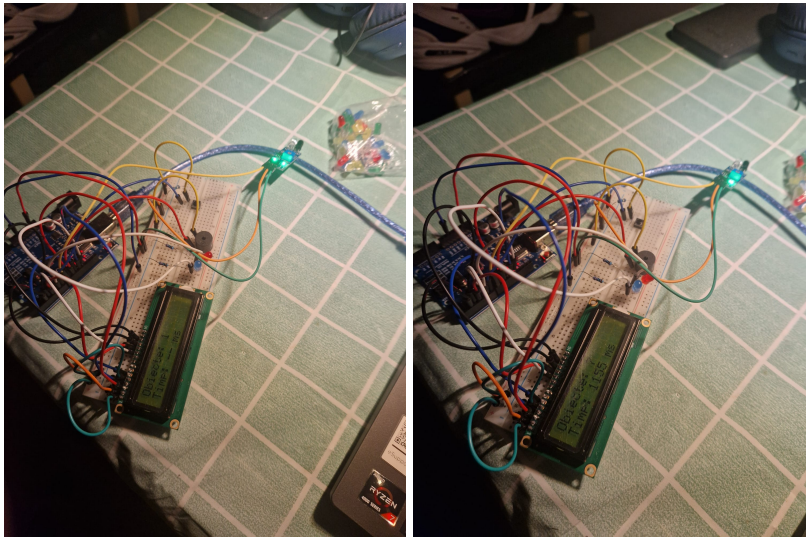
1x Push button

1x Breadboard

Fire jumper

1x Rezistență 10kΩ (pentru buton)

1x Rezistență 220Ω (pentru test LED)



Poze:

Diagrame de semnal: Prezintă semnalul digital DOUT al senzorului:

LOW când detectează obstacol

HIGH în lipsa obstacolului

Software Design

Mediu de dezvoltare Folosim PlatformIO în Visual Studio Code pentru dezvoltare, compilare și upload.

Alternativ, codul poate fi rulat și cu Arduino IDE.

Microcontroller-ul folosit este ATmega328P (placa Arduino UNO).

Limbajul de programare: C++ (framework Arduino).

Librării și surse 3rd-party `#include <LiquidCrystal.h>` - pentru controlul afișajului LCD 16x2 conectat în mod paralel (fără I2C).

`#include <Arduino.h>` - pentru funcții de bază (digitalWrite, millis, etc.), inclusă implicit în PlatformIO.

Algoritmi și structuri implementate 1. Inițializare componente Se definesc pinii pentru LCD, buzzer,

LED-uri, senzor IR și butonul de reset. Se setează tipul fiecărui pin.

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2, ct = 9; const int irPin = 8; const int  
buzzerPin = 10; const int ledPin = 13; const int resetButton = 7;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
void setup() {
```

```
pinMode(irPin, INPUT);  
pinMode(buzzerPin, OUTPUT);  
pinMode(ledPin, OUTPUT);  
pinMode(resetButton, INPUT); // conectat la GND  
analogWrite(ct, 100); // luminozitate ecran  
lcd.begin(16, 2);  
lcd.print("Obiecte: ");
```

} 2. Contorizarea obiectelor detectate Senzorul IR emite semnal LOW când un obiect este prezent. Se folosește o variabilă objectDetected pentru a evita incrementările multiple la o singură trecere.

```
if (irStatus == LOW && !objectDetected) {
```

```
objectDetected = true;  
counter++;
```

```
lcd.clear();  
lcd.print("Obiecte: ");  
lcd.print(counter);
```

```
digitalWrite(buzzerPin, HIGH);  
digitalWrite(ledPin, HIGH);  
delay(100);  
digitalWrite(buzzerPin, LOW);  
digitalWrite(ledPin, LOW);
```

} 3. Debounce pentru buton reset Butonul resetează contorul și afișajul. Se aplică un delay pentru stabilizarea semnalului (debounce simplu software).

```
if (digitalRead(resetButton) == HIGH) {
```

```
counter = 0;  
lcd.clear();  
lcd.print("Obiecte: ");  
lcd.print(counter);  
delay(300);
```

} 4. Detecție de trafic rapid (ALERTĂ) Se memorează ultimele 3 timestamp-uri folosind millis(). Dacă 3 obiecte trec în <1 sec, sistemul emite alertă.

```
lastTimes[0] = lastTimes[1]; lastTimes[1] = lastTimes[2]; lastTimes[2] = millis();
```

```
if (lastTimes[2] - lastTimes[0] <= 1000 && lastTimes[0] != 0) {
```

```
    lcd.clear();  
    lcd.print("ALERTA: PREA");  
    lcd.setCursor(0, 1);  
    lcd.print("MULTE OBIECTE");
```

```
    digitalWrite(buzzerPin, HIGH);  
    digitalWrite(ledPin, HIGH);  
    delay(2000);  
    digitalWrite(buzzerPin, LOW);  
    digitalWrite(ledPin, LOW);
```

```
    counter = 0; // resetăm contorul  
    lcd.clear();  
    lcd.print("Obiecte: ");  
    lcd.print(counter);
```

```
}
```

Rezultate Obținute

Rezultate Obținute: S-a realizat cu succes un sistem funcțional de numărare a obiectelor detectate de un senzor IR.

Afișajul LCD actualizează în timp real numărul obiectelor.

Comportament corect în caz de treceri rapide, sistemul detectând fluxul excesiv și reacționând automat.

Testele au fost realizate atât pe breadboard cât și în PlatformIO, cu program funcțional încărcat cu succes.

S-au folosit 5 piese electronice diferite și 4 concepte din laboratoare: GPIO, PWM, afișaj, debounce, memorie RAM și temporizare.

Concluzii

Proiectul demonstrează aplicarea practică a noțiunilor de PM: programarea microcontrollerului, lucrul cu senzori digitali, afișaj și interacțiune hardware.

Codul este modular, comentat și poate fi ușor extins (ex: numărare bidirecțională, afișare pe serial monitor, stocare etc).

Se poate transforma într-un prototip real de contor pentru bandă transportoare în fabrici sau linii de sortare.

Download

Proiectu [pmproject.zip](#)

Bibliografie/Resurse

LCD Help: <https://www.youtube.com/watch?v=860eErq9c3E>

Sensor Help: <https://www.youtube.com/watch?v=OMZacCLRt9A>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/iotelea/catalin.turcanu>



Last update: **2025/05/26 20:06**