Sistem Intersectie Semaforizata

Introducere

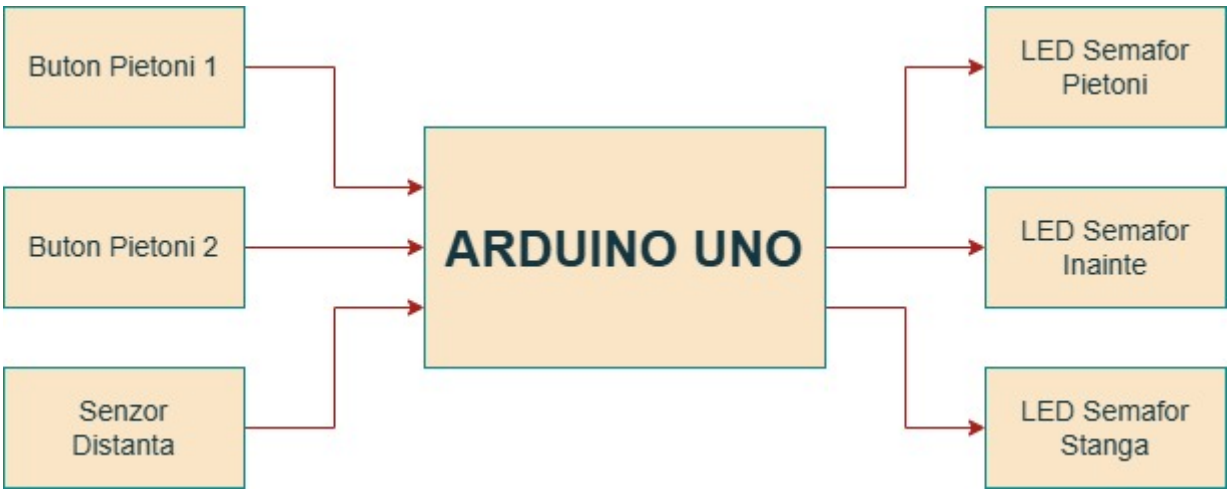
Proiectul prezinta un model pentru o intersectie in T, semaforizata, controlata printr-un senzor de distanta si butoane. Intersectia este formata dintr-un bulevard des circulat, o trecere de pietoni si o strada secundara cu sens unic; soferii care trebuie sa vireze la stanga ca sa intre pe strada secundara trebuie sa astepte ca semaforul pentru directia inainte sa devina rosu.

Pentru a regula aceste intervale, semafoarele pentru directia inainte sunt initial verzi, iar daca un senzor de distanta detecteaza masini care vor sa vireze stanga sau un buton este apasat de catre un pieton, dupa un timp de asteptare semafoarele se vor schimba, permitand celor care asteapta sa treaca. Dupa un timp suficient de lung, sistemul se va intoarce in starea initiala.

Scopul acestui sistem este acela de a regla timpul alocat traversarii intersectiei de catre pietoni si timpul alocat masinilor care circula inainte/la stanga. Ideea de la care am pornit este urmatoarea: daca pietonii si masinile care vireaza la stanga apar mult mai rar decat masinile care merg drept inainte, atunci ar fi util un sistem inteligent care sa gestioneze cand sa permita pietonilor (si masinilor care vireaza la stanga) sa treaca.

Un astfel de sistem ar reduce timpul petrecut la stop pentru cea mai mare parte a soferilor care circula prin intersectie, fara a exclude pietonii si minoritatea soferilor care trebuie sa vireze la stanga.

Descriere generală



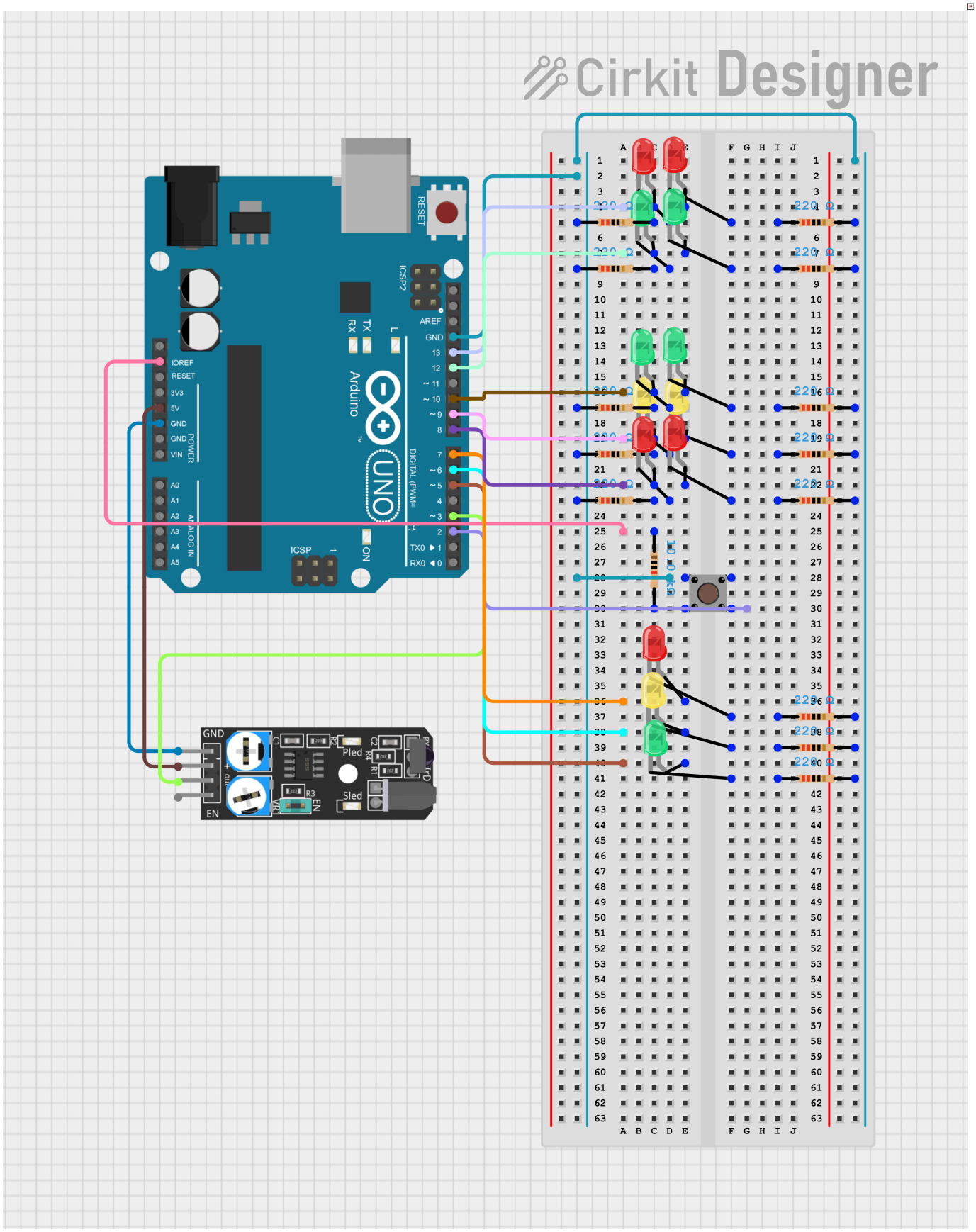
Sistemul trece in mod circular prin mai multe stari care controleaza traficul intersectiei. La inceput, semafoarele pentru stanga si pietoni sunt rosii, iar semafoarele inainte sunt verzi. In momentul in care un pieton apasa pe un buton, dar si in momentul in care o masina care asteapta ca sa vireze la

stanga este detectata de senzorul de distanta, se porneste un timer. Atunci cand timerul ajunge la 0, semaforul inainte va trece printr-o stare intermediara (culoarea galbena), apoi va deveni rosu. Dupa un delay de siguranta, semafoarele pentru pietoni si stanga vor deveni verzi.

Cand semafoarele pentru pietoni si stanga devin verzi, microprocesorul intra intr-o stare in care ignora butoanele si senzorul si porneste un timer de astepare. Cand timerul de asteptare ajunge la 0, semafoarele trec printr-o stare intermediara (pentru pietoni: clipeste rosu, pentru masini: culoarea galbena), apoi devin din nou rosii. Dupa un delay de siguranta, semafoarele pentru inainte devin verzi, iar microprocesorul incepe din nou sa astepte semnale de la senzor si butoane.

Hardware Design

Schema Electrica



BOM

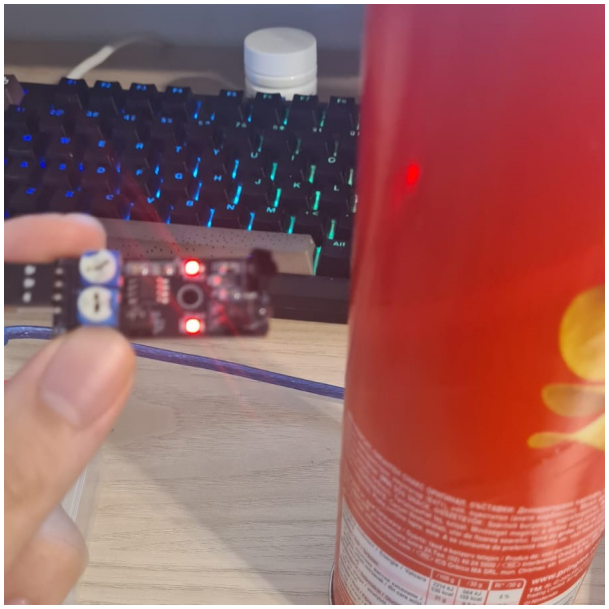
Denumire	Cantitate	Datasheet

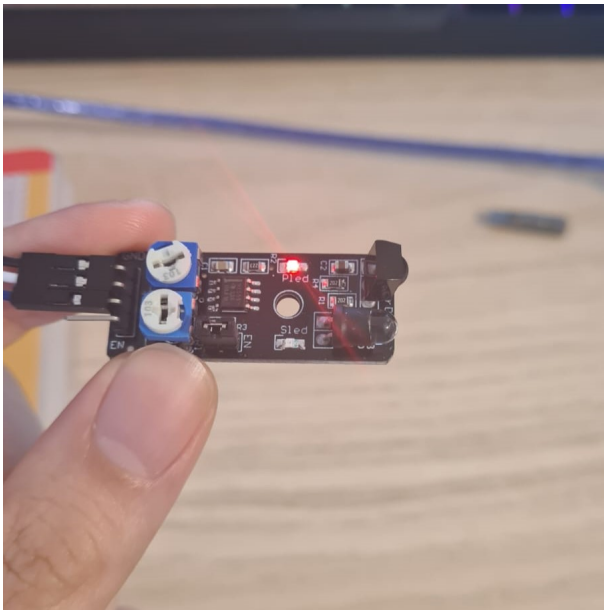
Arduino UNO R3	1	datasheet
Senzor Infrarosu KY-032	1	datasheet
LED Rosu	5	
LED Galben	2	
LED Verde	5	
Rezistor 220 Ohm	13	
Rezistor 10k Ohm	1	
Large Breadboard	1	
PushButton	1	

Descriere componente

- **Arduino Uno R3:** Controller ATMega328P, responsabil cu primirea datelor de la senzor si de aprinderea/stingerea LED-urilor.
- **Senzor Infrarosu KY-032:** (GND-GND, VCC-5V, OUT-D3) Senzor cu infrarosu pentru detectarea obstacolelor, folosit pentru a detecta prezenta unei masini la semafor (distanta reglabila prin intermediul rezistorilor sai variabili; pinul ENABLE nu a fost conectat deoarece sistemul nu dezactiveaza senzorul. Apeleaza o functie de intrerupere atunci cand este inregistrata o valoare LOW.
- **LED-uri colorate:** (pinii D5-10, D12-D13) Becuri LED care reprezinta luminile unor semafoare.
- **Buton:** (pin D2, 5V, GND) Buton pentru pietoni. Apeleaza o functie de intrerupere atunci cand este inregistrat un prag descrescator.

Functionare KY-032





Software Design

- **Mediu de dezvoltare:** Arduino IDE 2.3.6
- Nicio biblioteca third-party
- **Element de noutate:** activarea tranzitiilor folosind ori un senzor, ori un buton (spre deosebire de folosirea numai a unui buton)
- **Functionalitati din laboratoare:** GPIO, intreruperi, timere

States

Sistemul se bazeaza pe multiple stari de functionare prin care cicleaza pe baza unui timer intern (Timer1, prescale 1024, compare mode). Fiecare configuratie de becuri aprinse si stinse are propria sa functie dedicata, functie apelata in interiorul functiei de loop a placutei. Tranzitiile intre stari au loc in cadrul unor intreruperi; la apasarea butonului si/sau detectarea unei masini de catre senzor, timer-ul intern este pornit cu o valoare de timp setata (valorile folosite sunt de 0.5, 1, 2 si 4 secunde, calculate folosind valoarea intern cycle = 16 MHz).

Dupa ce placuta trece din starea initiala (s0) la starea s1 in urma unei intreruperi cauzata de buton sau de senzor, intreruperile de buton si senzor sunt ignorate pana cand sistemul revine in starea initiala. Sistemul va trece prin stari de la 1 la 9 folosindu-se un compare timer interrupt. La fiecare completare a timer-ului, valoarea sa se reseteaza si pragul este modificat in functie de intervalul de timp care se vrea. Odata ce ultimul timer expira (in starea s9), timer-ul este dezactivat, interrupt-urile de buton si senzor sunt reactivate, iar la final se trece inapoi in starea initiala s0.

Scheletul proiectului

- **setup():** contine comenzile de inceput - initializarile de pini, stare initiala si functionarea timer-ului Timer1.
- **loop():** functia apelata la fiecare tick, aprinde sau opreste LED-urile in functie de starea curenta.
- **lights_######():** set de functii care aprind sau opresc LED-uri pentru a obtine o configuratie valida a luminilor intr-o intersectie. Configuratiile valide sunt: complet rosu, verde pentru masini inainte, galben pentru masini inainte, verde pentru masini la stanga si pietoni, si doua pentru galben pentru masini la stanga (semaforul de pietoni clipeste in loc sa foloseasca o a treia culoare).
- **inputInterrupt():** functie asociata pinilor 2 si 3 (buton si senzor), pini folositi de placuta Arduino Uno ca pini de interrupt. Functia este controlata de variabila canTrigger.
- **ISR(TIMER1_COMPA_vect):** functie de interrupt care verifica daca Timer1 a atins pragul setat. In momentul in care este apelata, reseteaza timer-ul si seteaza o noua valoare de atins, apoi trece in starea urmatoare; daca se afla in starea s9, se va dezactiva timer-ul.

Fragmente de cod

```
void lights_forward()
{
  // Forward semaphore
  digitalWrite(fwd_sem_red_pin, LOW);
  digitalWrite(fwd_sem_ylw_pin, LOW);
  digitalWrite(fwd_sem_grn_pin, HIGH);
```

```
  // Left semaphore
  digitalWrite(left_sem_red_pin, HIGH);
  digitalWrite(left_sem_ylw_pin, LOW);
  digitalWrite(left_sem_grn_pin, LOW);

  // Walk semaphore
  digitalWrite(walk_sem_red_pin, HIGH);
  digitalWrite(walk_sem_grn_pin, LOW);
}
```

Exemplu de functie care gestioneaza care LED-uri sunt aprinse; in acest caz, semaforul “inainte” este verde, iar semafoarele “stanga” si “pieton” sunt rosii.

```
ISR(TIMER1_COMPA_vect)
{
  // Set duration based on current state
  switch (runningState) {
    case 1:
      TCNT1 = 0; // Set counter value to 0
      runningState = 2;

      // Set timer value to 2 seconds
      OCR1A = twoSecondCompare;

      break;
      ...
  }
}
```

Secventa de cod din cadrul functiei ISR. La fiecare intrerupere cauzata de Timer1, se trece in starea urmatoare si se reseteaza timer-ul. Pentru a controla timpul dintre schimbarile de culori, OCR1A primeste noi valori (precalculate, constante).

Algoritmi si structuri utilizate

Proiectul foloseste modelul automatelor cu stari si imparte instructiunile in doua categorii: instructiuni de tranzitie intre stari si instructiuni de output pe baza starilor. Prima categorie este compusa din functiile de intreruperi ISR() si inputInterrupt(), care gestioneaza tranzitiile intre starile sistemului pe baza timer-ului intern sau prin folosirea butonului/senzorului. A doua categorie este reprezentata de functia loop() care gestioneaza pinii de output si, in consecinta, semafoarele. Variabila runningState face legatura dintre aceste functii, asigurand aprinderea si stingerea corecta a becurilor LED.

Calibrari

- Pentru a regla distanta maxima a senzorului, am ajustat unul dintre rezistorii reglabili, astfel incat

senzorul sa ignore soseaua, dar sa inregistreze masina care asteapta la semafor.
- Software: am calculat pragurile pentru timer pe baza intervalelor de timp folosite (0.5, 1, 2, 4 secunde) si am salvat valorile ca numere constante.

Optimizari

- Functia de intrerupere apelata de buton si de senzor foloseste un guarding clause pentru a preveni apelarea sa cand sistemul se afla in alta stare decat cea initiala. Acest lucru asigura pastrarea ordinii starilor si prezervarea timer-ului original atunci cand si butonul, si senzorul sunt folosite.
- In loc ca fiecare stare sa aiba operatiile sale in bucla loop(), functionalitatile au fost generalizate si redistribuite intre stari.
- Prin folosirea de intreruperi, bucla principala ramane foarte usoara din punct de vedere al instructiunilor, intrucat tot ce se intampla aici este trimiterea de curent pe pinii asociati LED-urilor.

Rezultate Obţinute

Sistemul inregistreaza atat apasarile de buton, cat si obstacolele din fata senzorului, permitand atat pietonilor, cat si masinilor care vireaza la stanga sa treaca sau sa traverseze. Timpul alocat trecerii pietonilor si a masinilor este reglabil, astfel ca sistemul poate fi usor extins la o scara mai mare.

Concluzii

Acest design este o evolutie a sistemului pe baza de buton pentru pietoni deoarece permite masinilor sa vireze la stanga si in absenta unor trecatori. El poate fi extins si dezvoltat pentru implementari in viata de zi cu zi, pentru a reduce stresul si frustrarea din anumite intersectii.

Proiectul m-a ajutat sa inteleg mai bine functionarea microprocesoarelor, interactiunea dintre acestea si circuitele simple, dar si nivelul de planificare necesar pentru a realiza un astfel de sistem.

Download

badescuandreicristian.rar

Bibliografie/Resurse

Bibliografie - GitHub

https://github.com/Erixxl/PM-Arduino-Intersection

Resurse

https://roboticsbackend.com/arduino-push-button-tutorial/

https://www.instructables.com/Arduino-Timer-Interrupts/

https://arduinomodules.info/ky-032-infrared-obstacle-avoidance-sensor-module/

https://docs.arduino.cc/language-reference/en/functions/external-interrupts/attachInterrupt/

https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

http://irsensor.wizecode.com/

Export to PDF