

Traffic light controller

- Nume: Bogdan-Constantin Rizescu
- Grupa: 332CA

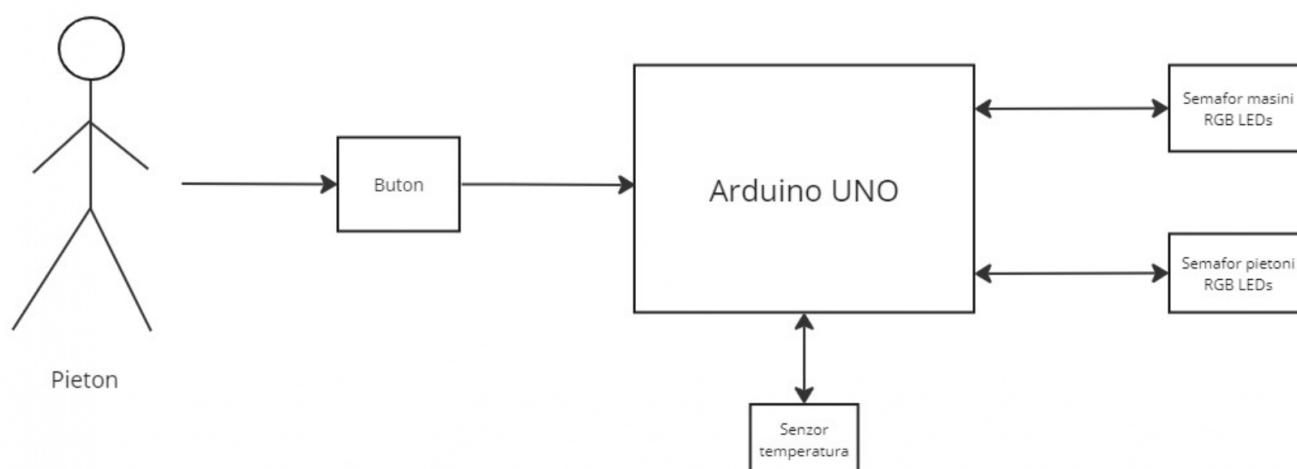
Introducere

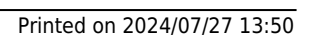
- Scopul proiectului este de a crea un semafor temporizat pentru pietoni si masini
- Temporizarea se va face la apasarea unui buton de catre pieton, culorile semafoarelor inversandu-se dupa apasare
- De asemenea, un senzor analogic de temperatura va masura temperatura ambientala pentru a putea mai apoi micsora temporizarea in cazul in care temperatura scade sub un anumit prag.

Descriere generală

Proiectul presupune implementarea unui semafor temporizat pentru pietoni si unul pentru masini. In momentul in care un pieton apasa butonul, acesta declanseaza o intrerupere in care este tratata schimbarea culorii semafoarelor (atat pentru masini, cat si pentru pietoni). De asemenea, presupune utilizarea unui senzor de temperatura care in cazul in care este prea frig afara, timpul de asteptare pentru pietoni sa fie mai mic. Daca senzorul de temperatura inregistreaza o valoare sub un anumit prag, atunci timpul de asteptare pentru pietoni este micsorat la 2 secunde, altfel va ramane timpul setat in mod obisnuit, de 5 secunde (am setat timpii atat de mici doar pentru a nu astepta prea mult in cadrul testarii).

Schema bloc:





Software Design

Mediul de dezvoltare pe care l-am utilizat a fost **Arduino IDE**. De asemenea, am utilizat <https://webwhiteboard.com/> pentru a desena schema de stari si tranzitii.

Am folosit, de asemenea, cunostinte din urmatoarele laboratoare de PM:

- Laboratorul 2: **Întreruperi** hardware (pentru butonul pe care il apasa pietonul)
- Laboratorul 3: **Timere**. PWM (pentru a temporiza starile in care se afla LED-urile si cand ar trebui sa se faca tranzitia pe o alta stare)
- Laboratorul 4: **ADC** (pentru a citi valoarea senzorului analogic de temperatura si a o transforma din mV in °C)

Design si mod de functionare

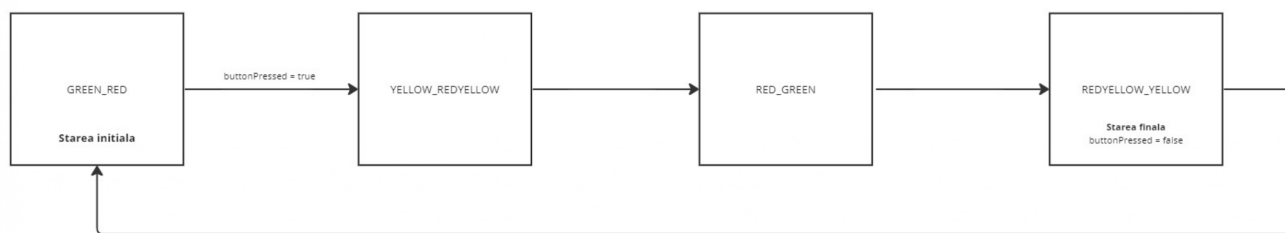
Initializari:

- am initializat 6 variabile pentru fiecare LED in parte, fiecareia corespunzandu-i cate un pin digital de pe placuta Arduino
- am initializat si o variabila pentru senzor cu numarul pinului analogic la care senzorul este conectat (A0)
- utilizez, de asemenea, 2 variabile pentru a seta temporizarea intre starile semaforului (timpul de tranzitie intre toate starile), cat si temporizarea pentru prima schimbare de culori (cat asteapta pietonul sa se schimbe culorile semaforului)
- folosesc enum-ul `LightState` in care imi definesc starile semaforului (`GREEN_RED`, `YELLOW_REDYELLOW`, `RED_GREEN`, `REDYELLOW_YELLOW`)

Descriere functii implementate:

- **setup()** - setez LED-urile in starea lor initiala (verde la masini si rosu la pietoni), atasez intreruperea de butonul caruia ii activez rezistenta de PULLUP si apelez functia de configurare a timer-ului
- **loop()** - se citesc permanent valorile primite de la senzorul de temperatura si se modifica temporizarea initiala pentru pietoni in functie de ce transmite senzorul; de asemenea, cat timp buton este tratat ca fiind fost apasat, functia `changeLights()` este apelata, semaforul schimbându-si starile corespunzator
- **changeLights()** - in functie de starea curenta in care se afla semaforul (a carei valoare este una definita in enum-ul `LightState` mentionat), se schimba starea curenta la o noua stare (luand in calcul si daca a trecut suficient timp pentru a putea trece la o alta stare)
- **configureTimer()** - configureaza `Timer1` conform laboratorului de PM, setand `OCR1A` la o valoare astfel incat sa am o intrerupere o data pe secunda
- **ISR(TIMER1_COMPA_vect)** - adun 1000 la variabila `seconds` (reprezentand 1000ms), functia fiind apelata o data pe secunda
- **buttonInterrupt()** - functia este apelata in momentul in care butonul este apasat, setand variabila `buttonPressed` ca fiind `true`, urmand a fi setata la `false` in momentul in care semaforul ajunge din nou la starea initiala (verde la masini, rosu la pietoni)

Schema de stari si tranzitii:



Rezultate Obținute

Concluzii

A fost un proiect interesant, pentru mine laboratorul de PM fiind prima data cand am interactionat cu Arduino si cu partea de hardware in general. Ma bucur ca am inteles cum se poate construi un proiect software-hardware si ca am acumulat o multime de cunostinte noi despre good/bad practices cand vine vorba de a scrie cod pe o placuta hardware. Imi place ce a iesit in cele din urma, consider ca este un proiect care rezuma bine semestrul acesta la PM si care chiar s-ar putea implementa in realitate, nefiind ceva foarte complicat si devenind chiar util pentru pietoni in conditii de vreme neprielnica.

Download

Arhiva proiect: [traffic_light_controller_rizescu_bogdan-constantin.zip](#)

Jurnal

- 3 mai: alegere tema proiect
- 5 mai: documentatie initiala
- 17 mai: legare si testare componente hardware
- 21 mai: adaugare schema circuit si schema electrica
- 23 mai: update schema circuit si schema electrica (schimbare pin digital buton de pe pinul 12 pe pinul 2 si senzor de pe A1 pe A0)
- 28 mai: update documentatie: adaugare software design, videoclip, concluzii, arhiva proiect, update jurnal, bibliografie

Bibliografie/Resurse

- <https://www.makeuseof.com/tag/arduino-traffic-light-controller/>
- <https://www.electronicwings.com/arduino/lm35-interfacing-with-arduino-uno>
- <https://www.tinkercad.com/>
- <https://webwhiteboard.com/>
- Laboratoare PM

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/vstoica/bogdan.rizescu>



Last update: **2023/05/28 21:42**