

Bariera parcare institutie cu deschidere manunala si inchidere automata

Introducere

Prezentarea pe scurt a proiectului vostru:

- Proiectul consta intr-o mini bariera care se deschide prin intermediul unui buton(asemanator celei de la Politehnica, dupa ce s-a platit taxa) si se inchide automat dupa ce detecteaza ca vehiculul in cauza a trecut sau un anumit timp a trecut si nici un vehicul nu este in dreptul senzorului. Bariera are de asemenea un LED care simbolizeaza daca se permite sau nu trecerea.
- Scopul de a implementa o automatizare eficienta pentru a asigura securitatea institutiilor publice si private.
- Sursa de inspiratie: internet, proiecte anterioare
- Este nevoie ca toate institutiile publice sa aiba o masura de securitate pentru ca angajatii sa aiba unde parca si sa nu fie locurile ocupate de catre persoane straine, lucru mai ales necesar in Bucuresti.

Descriere generală

Proiectul este un micro-controller Arduino, care functioneaza astfel: Pentru deschiderea barierei, trebuie apasat butonul, iar pentru inchidere, este folosit un senzor care detecteaza daca masina a trecut deja de bariera / a trecut deja o limita de timp si nici o masina nu se afla in fata barierei, lucruri care se realizeaza prin intermediul unui servo-motor. Avem un LED, care este aprins pe verde cand bariera este ridicata, galben cand bariera este in ridicare, rosu cand bariera este inchisa.

Schema bloc

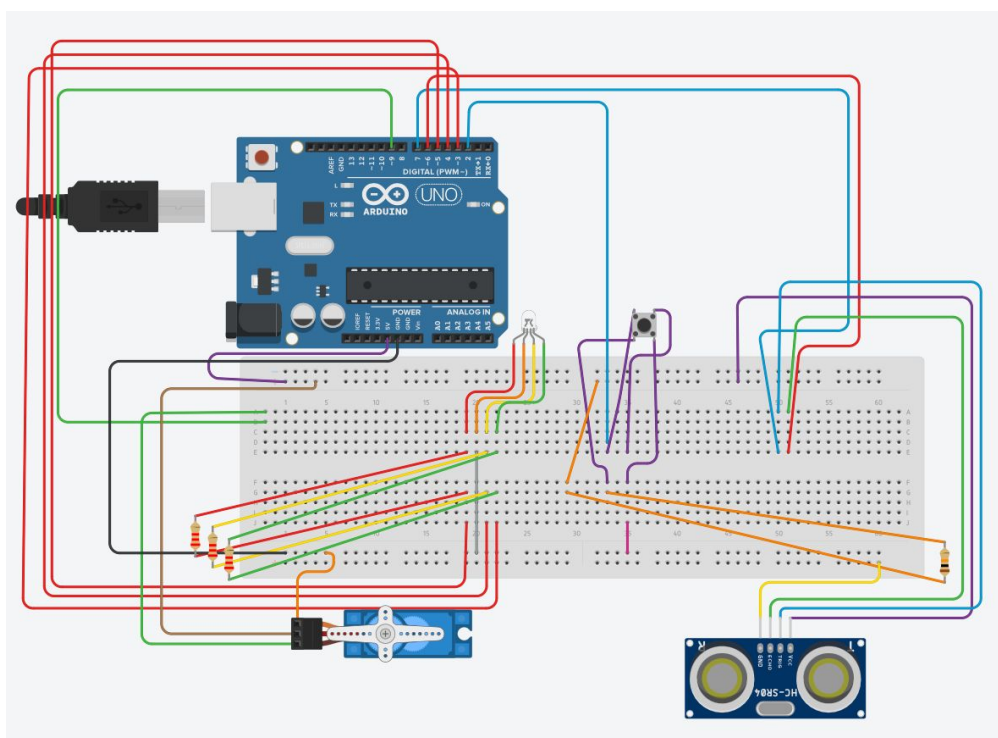


Hardware Design

Lista piese:

- Placa de Dezvoltare Compatibila cu Arduino UNO R3 (ATmega328p + ATmega16u2)
- Kit Breadboard HQ830 cu Fire și Sursă
- Micro Servomotor SG90 90°
- Senzor ultrasonic HC-SR04
- 4 Rezistente
- Butoane
- Led RGB

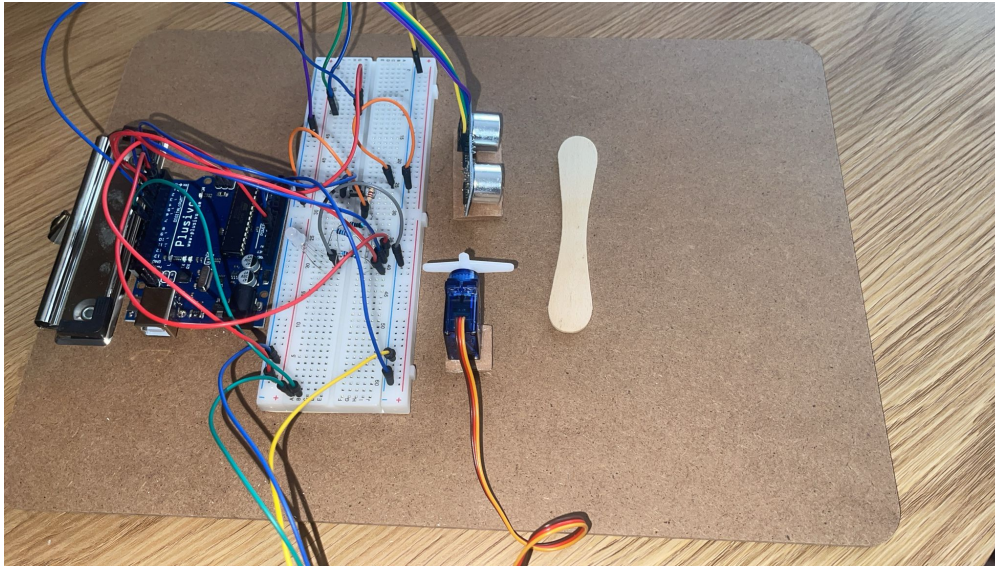
Schema Electrica



Schematic



Hardware Setup



Software Design

Descrierea codului aplicației (firmware):

- Mediul de dezvoltare folosit a fost Arduino IDE + simulari anterioare in TinkerCad pentru a asigura corectitudinea codului inainte de a testa manual pe cablajul realizat de mine.
- Am folosit o singura librarie third party, pentru lucrul cu servomotorul, <servo.h>
- Algoritmi si functii folosite: Logica principala a programului se afla in functiile `setup()` `loop()`. In functia de setup initializez toti pinii de input(butonul, care este input de PULL_UP, la fel si pin-ul pentru echo de la servomotor) si cele de output(led-urile, trigger pin-ul servomotorului). In setup de asemenea setez intreruperea folosita de buton prin ISR si initializez servomotorul. By default, led-ul rosu este pornit, deoarece la inceput bariera este coborata. In functia de loop, la inceput, verific permanent daca s-a activat intreruperea butonului, caz in care, daca bariera este coborata, setez led-ul pe galben prin intermediul GPIO-ului si urc bariera, LED-ul trecand pe verde. Daca aceasta era deja ridicata, resetez timpul la care s-a detectat masina(dupa ce a trecut masina de senzor se asteapta 3 secunde pana la coborarea barierei, se reseteaza acest timp). Daca bariera este ridicata, se verifica daca a trecut masina de senzor, dupa se asteapta 3 secunde, iar dupa trecerea acestora se verifica daca masina mai este in dreptul senzorului sau a trecut. Daca se respecta toate conditiile, LED-ul trece pe galben si bariera este coborata. Functia `checkObjectDetected()` verifica daca un obiect(in cazul nostru, masina RC) se afla in dreptul senzorului. Prima data, se seteaza trigpin-ul pe low pentru 2ms si apoi se citeste rezultatul echoPin. Am folosit urmatoarea formula de calcul a distantei: $dist = duration * 0.034 / 2$. Daca distanta < 10, se verifica daca obiectul se afla persistent in fata pentru 10 semnale de ceas in fata senzorului, caz in care se returneaza 1. Daca distanta este 0(bug), se returneaza 0, iar daca este ≥ 10 se returneaza -1. Pentru setarea culorii led-urilor m-am folosit de functia `setColor()`, in care am facut analogWrite pe pini pentru fiecare culoare.

Rezultate Obținute

Bariera este functionala, demo-ul va fi prezentat la PM fair.

Concluzii

Proiectul a fost foarte interesant, printre cele mai interesante din facultate, am invatat sa lucrez cu Arduino si cum sa fac un cablaj electric corect.

Download

https://github.com/Dave080302/Embedded_barrier

Jurnal

- 5 aprilie: selectare proiect
- 2 mai: finalizare documentatie initiala
- 3 mai: cumparare piese
- 15 mai: finalizare lipire
- 20 mai: mici probleme cu legarea initiala a firelor, mutarea intr-o singura parte a breadboard-ului
- 22 mai: finalizare cod, probleme cu senzorul
- 23 mai: proiect 100% functional

Bibliografie/Resurse

<https://www.arduino.cc/reference/en/libraries/servo/>

<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>

<https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/david_tudor.anghel



Last update: **2024/05/26 21:35**