

Radu-Dumitru STOCHIȚOIU (67017) - Ajustor trafic

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Am ales sa fac un proiect al carui scop este acela de a optimiza durata semnalelor semafoarelor dintr-o intersectie de tip cruce in functie de densitatea de masini care asteapta la fiecare semafor si de eventuale urgente, cum ar fi trecerea unei ambulante sau a unei coloane oficiale. Locuind in Bucuresti de cand sunt la facultate mi-am dat seama ca nu-mi doresc o masina in curand deoarece cozile la fiecare semafor sunt infernale, iar nanosecunda se defineste ca fiind distanta de la culoarea verde pana la primul claxon. Cred ca este util deoarece ar minimiza efortul oamenilor (cu toate ca exista deja aparate care grabesc un anumit semafor - ex. pietoni).

Descriere generală

Schema bloc

Taximetristii au inceput sa mearga cu Uber? V-ati saturat de statul la cozi? Injuraturile nu mai functioneaza? Ati venit in locul potrivit.



Descriere generala

In mod natural, fiecare semafor va avea o cuanta de timp pentru culoarea verde si apoi va deveni galben, respectiv rosu, trecand la urmatorul semafor in sens trigonometric.

Senzorii infrarosu vor detecta numarul de masini care vin (respectiv care pleaca) si daca o anumita directie este mult mai plina decat celelalte, se va prioritiza in pofida celorlalte. De asemenea, cuanta de timp poate deveni mai mare in functie de caz.

Evenimentele neasteptate precum trecerea unei ambulante sau a unei coloane oficiale vor avea rang

critic din punct de vedere al prioritatii. Pe telefon as vrea sa fie, de asemenea, afisate statisticile intersectiei respective pentru ca atunci cand ajungi in ea sa stii la ce sa te astepti si ar fi ideal sa se poata anunta un ambuteiaj, un accident sau orice altceva urgent pentru a inchide acea banda, iar soferii sa fie anuntati din timp. (de asemenea, de pe telefon va urma sa se poata controla numarul de masini existente in fiecare intersectie daca se doreste testarea fara infrarosu)

Hardware Design

Lista de piese

- Placa de baza
- Componente de baza
- Fire
- Rezistente
- $3 * 4 = 12$ LED-uri
- $2 * 4 = 8$ IR Receiver
- $2 * 4 = 8$ IR Transmitter
- (Telefon cu Android/ testarea se va face pe telefonul meu, inclusiv log-uri)

Schema electrica

Schema generala:

Senzor IR:



Semafor din LED-uri:



Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice
- diagrame de semnal
- rezultatele simulării

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuieți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2017:avoinescu:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/avoinescu/traffic-adjustment>



Last update: **2021/04/14 15:07**