Wireless Controlled Car

Date de contact: Alexandru-Gabriel Popescu

Introducere

Proiectul isi propune realizarea unei masinute teleghidate controlate prin wireless. Masinuta va putea merge in spate/fata, lua viraje stanga/dreapta, semnaliza stanga/dreapta (dispunand de cate un led semnalizator atat in fata cat si in spate), aprinde farurile, claxona cu ajutorul unui buzzer pasiv si, pentru realism, va avea posibilitatea de a schimba vitezele.

Am ales acest proiect deoarece consider ca prin realizarea lui imi voi putea testa cunostiintele dobandite, voi putea sa lucrez atat din punct de vedere hardware cat si software si in plus, la final, voi obtine un lucru de care ma voi putea bucura si la care voi lucra in continuare incercand sa aduc imbunatiri.

Descriere generală

Prin intermediul unei aplicatii android, masinuta va primii comenzi precum:

- Claxon
- Semnalizare stanga
- Semnalizare dreapta
- Avarii
- Faruri
- Viraj stanga
- Viraj Dreapta
- Schimbarea vitezei (va fi controlata folosind PWM) inferioara/superioara
- Mers inainte
- Frana
- Mers inapoi

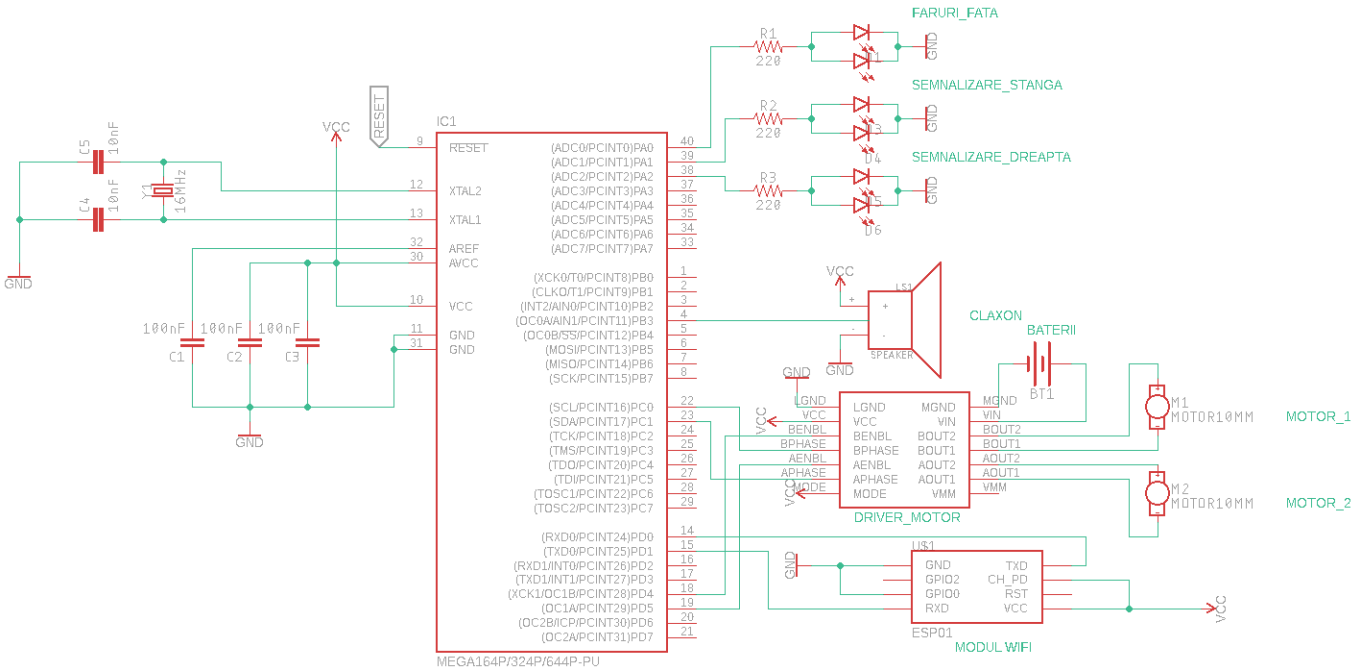
Comenzile vor fi transmise wireless si vor fi receptionate prin intermediul modulului ESP8266 (masinuta asteapta conexiuni, respectiv comenzi de la utilizator). Comanda va fi executata folosind driver-ul motor care va comanda cele doua motoare pentru obtinerea efectului dorit. Pentru a putea afisa adresa IP obtinuta in urma conectarii la un AP o sa folosesc un LCD.

Hardware Design

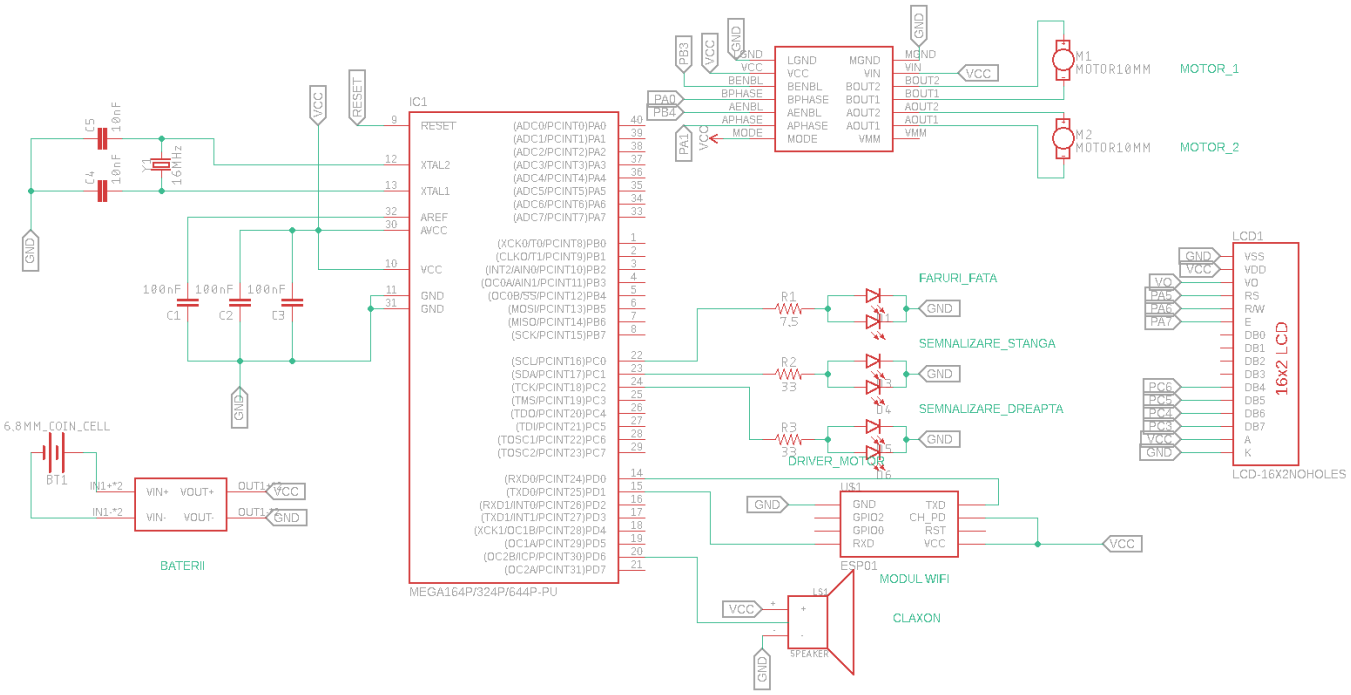
Lista piese:

- ATmega 324
- ESP8266 modul wireless
- DRV8835 driver motor
- 2 x DC motor
- LEDs
- Buzzer
- Sasiu
- LED-uri: 4 galbene si 2 albe
- 2 x Rezistente 33Ω (pentru LED-uri galbene)
- Rezistenta 7.5Ω (pentru LED-uri albe)
- Intrerupator
- Breadboard
- Fire mama-tata (10cm si 20cm)
- Fire tata-tata (10cm)
- Modul DC-DC Step-Down MP1584

Schema electrica initiala:



Schema electrica finala:



Alimentarea placutei:

```
Am folosit 4 baterii care furnizeaza impreuna aproximativ 6V. Bateriile se
conecteaza la sursa coboratoare de tensiune
care a fost setata sa furnizeze la iesire aproximativ 3.3V deoarece atat
modului wireless, modulul LCD cat si driverul motor functioneaza la 3.3V.
```

Software Design

Proiectul a fost realizat folosind urmatoarele fisiere:

- wireless_car.c
- usart.h
- usart.c
- lcd.c
- lcd.h

Aplicatia prin intermediul careia sunt transmise comenzile modulului de wireless se poate gasi atat in App Store cat si pe Google Play: details
id1409018107
Aplicatia permite conectarea la un ip pe un anumit port si transmiterea de mesaje. De asemenea, se pot defini butoane care vor
trimite comenzi predefinite la apasare. Astfel, am definit 12 comenzi care sunt receptionate de modulul wireless apoi receptionate prin USART
de placuta urmand sa fie prelucrate.

Comenzile pe care le recunoaste placuta sunt urmatoarele:

1. “lumini”: aprinde/stinge farurile fata
2. “sem_st”: porneste/opreste semnalizarea stanga
3. “sem_dr”: porneste/opreste semnalizarea dreapta
4. “avarii”: porneste/opreste avariile
5. “claxon”: porneste/opreste claxonul
6. “inainte”: mergi inainte
7. “inapoi”: mergi inapoi
8. “stanga”: porneste/opreste virajul la stanga
9. “dreapta”: porneste/opreste virajul la dreapta
10. “stop”: opreste masinuta
11. “shiftup”: schimba in viteza superioara
12. “shiftdown” schimba in viteza inferioara

Pentru receptia mesajelor pe interfata seriala, am activat intreruperile la receive, salvand intr-un buffer fiecare caracter primit.
In main, intr-o bucla verific la fiecare 100ms daca am primit mesaje de la modulul de wireless caz in care identific comanda si o pun in aplicare.
Fiecare modul folosit are o functie de initializare care seteaza directia pinilor, initializeaza device-ul si un timer(daca este necesar).

Pentru buzzer:

Am folosit timer-ul al doilea, pe 8 biti, al microcontrollerului, folosind un de prescaler de 128 care imi permite obtinerea unui sunet la o
frecventa de ~500Hz ceea ce corespunde unui claxon real.

Pentru semnalizari:

Am folosit timer-ul pe 16 biti al microcontrolerului pentru a putea obtine frecventa de 2Hz care da un aspect real semnaliarii. Am activat intreruperile
Compare Match A si Compare Match B pentru a face toggle la leduri. Bineinteles, acestea nu trebuie sa fie pornite tot timpul, de aceea controlez acest lucru
atribuindu-i lui OCR1A si OCR1B valori mai mari decat TOP-ul cand doresc ca semnalizarile sa fie oprite. Pentru avarii, pur si simplu se activeaza ambele
semnalizari in acelasi timp. Trebuie de avut in vedere alegerea unui timer care sa faca modificarile lui OCR1A/OCR1B pe loc pentru a nu fi buguri.

Pentru faruri:

Farurile sunt contolate de catre un pin GPIO.

Pentru driver-ul de motoare:

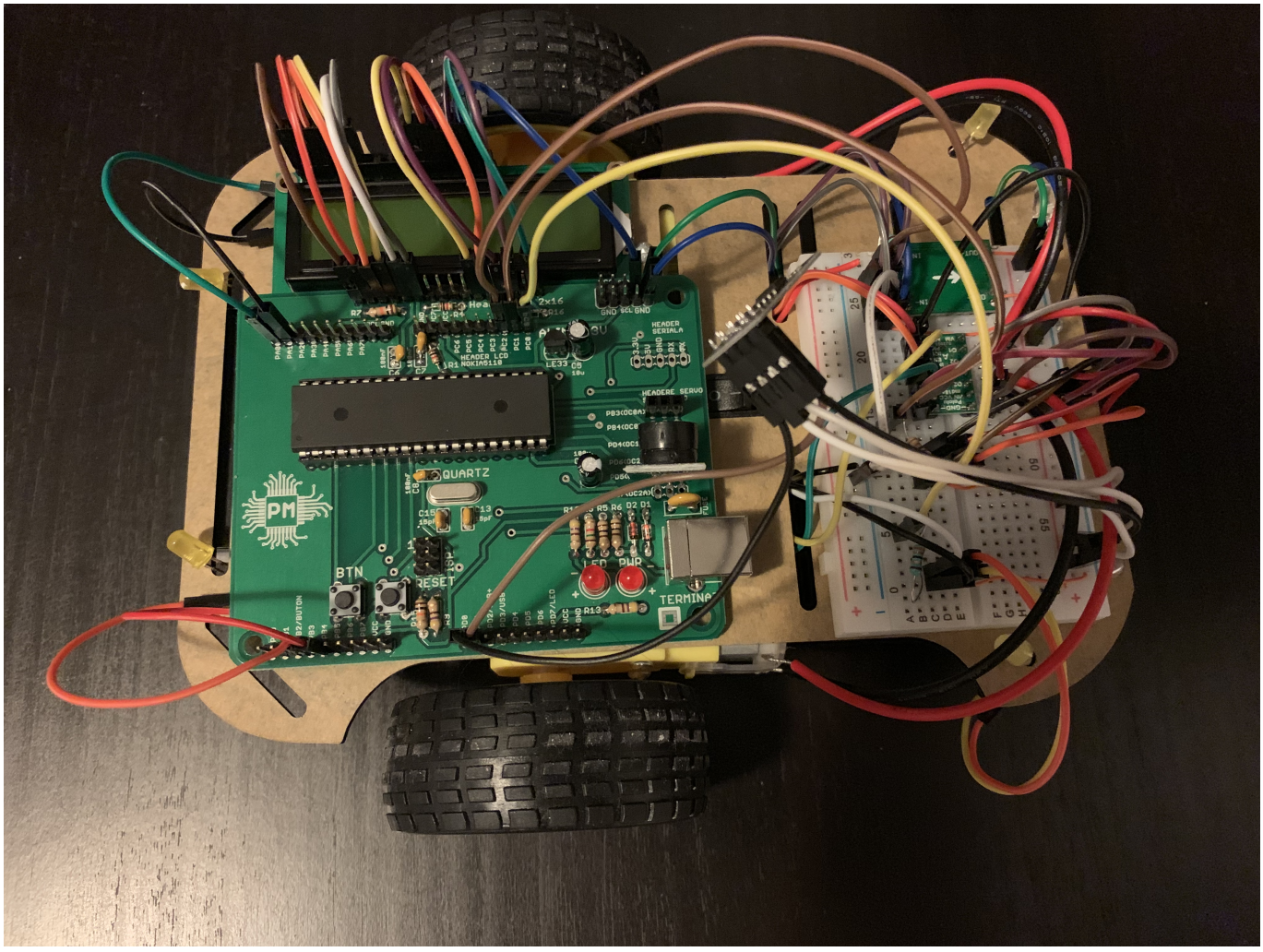
Folosesc primul timer pe 8 biti al microcontrollerului pentru a genera semnal PWM (factorul de umplere reprezinta puterea transmisa motorului) si inca alti
2 pini GPIO pentru a seta directia de deplasare. Atfel, deplasarea inainte/inapoi se face controland valorile celor doi pini: ambii pusi pe HIGH pentru miscare inainte
si ambii LOW pentru miscare inapoi. Am ales sa implementez virajele prin injumatatirea puterii pe roata in directia careia se ia virajul.
Dupa cum am prezentat, se poate schimba viteza de deplasare a masinutei, acest lucru se poate face prin cresterea/scaderea factorului de umplere al semnalului
PWM.

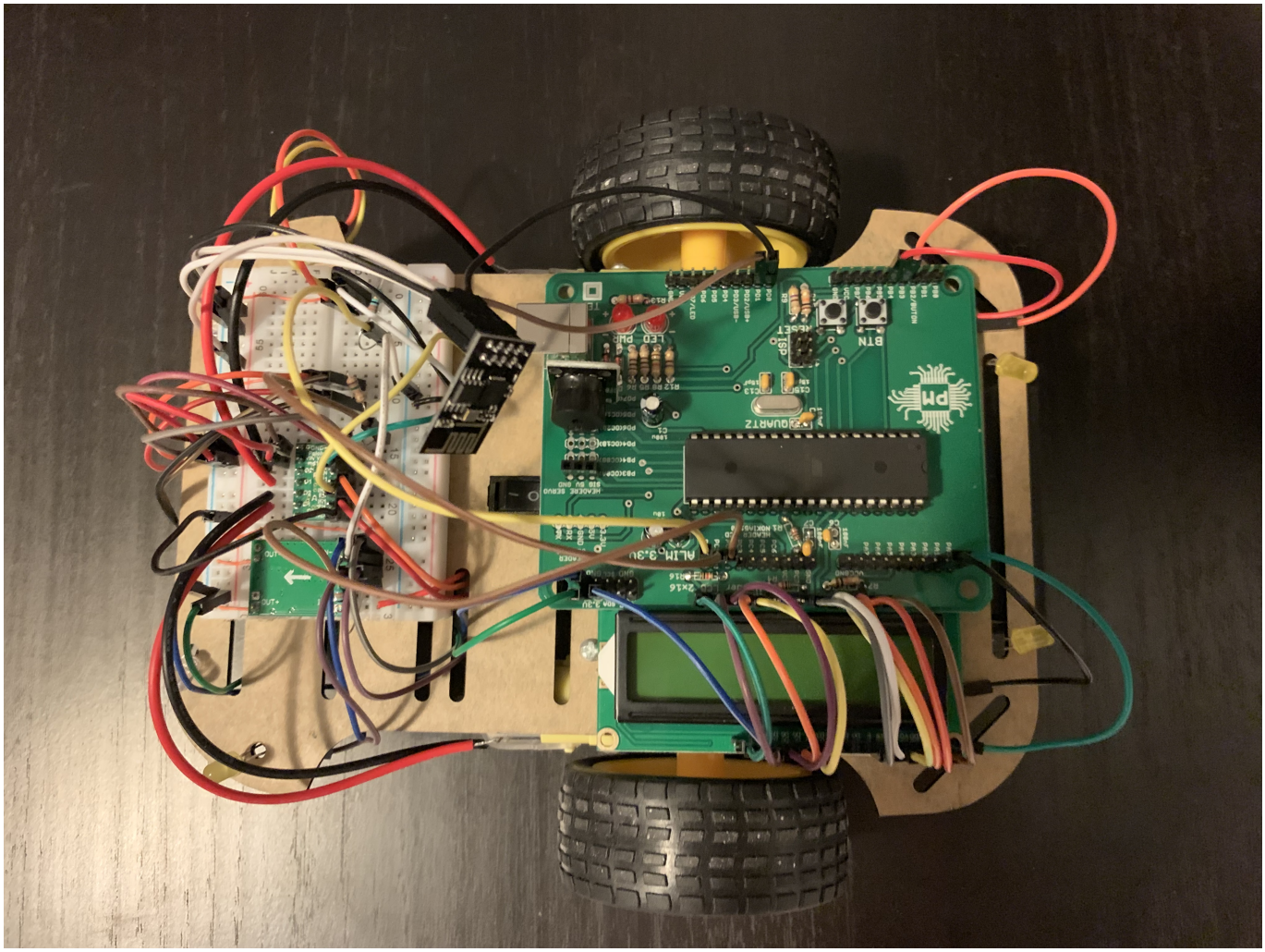
Pentru modului wireles ESP8266:

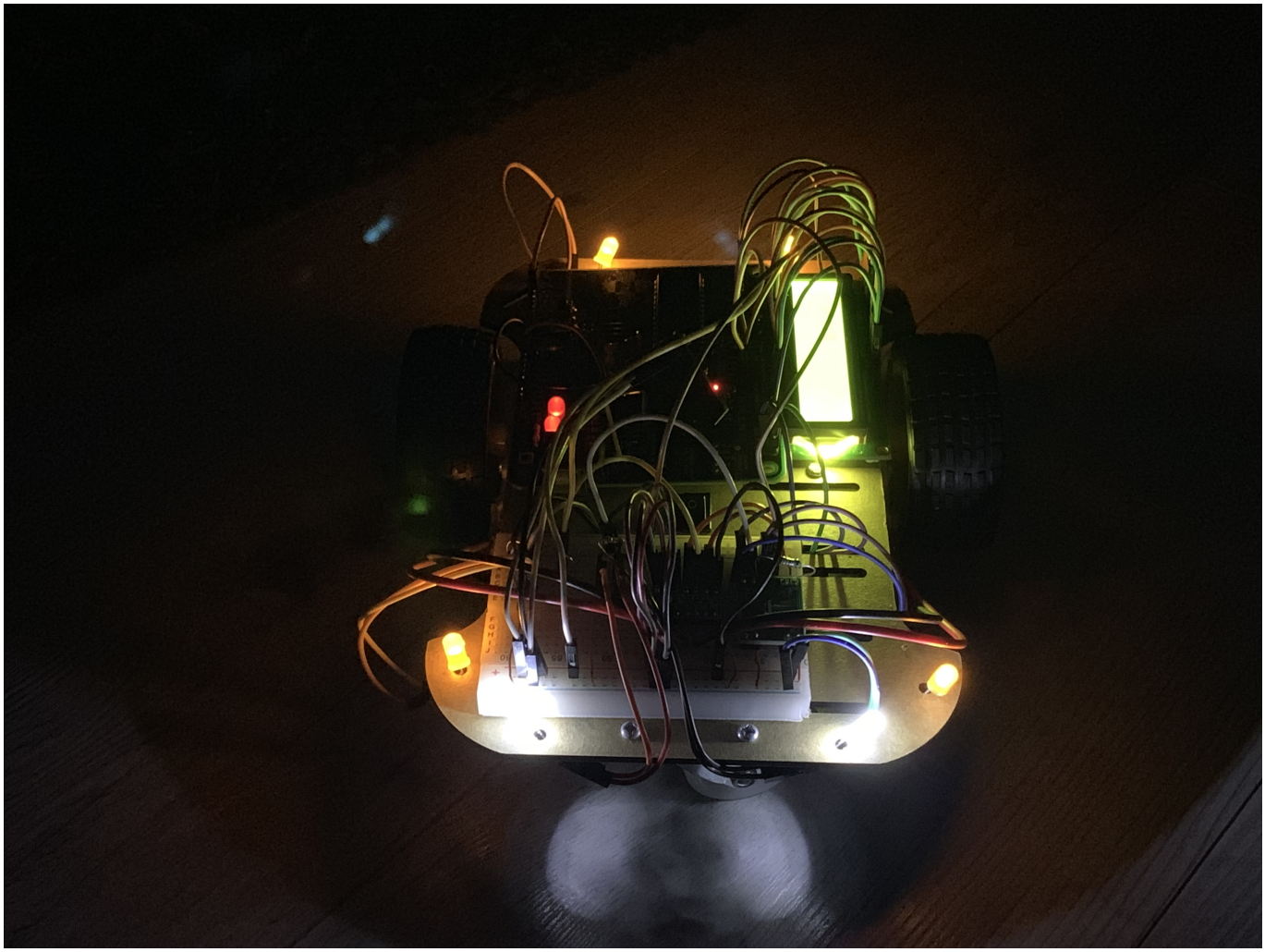
Cu siguranta cea mai grea parte din aceasta tema a fosst configurarea modului prin transmiterea comenzilor pe interfata seriala. Fiecare comanda ce din pasii
de initializare a modului este urmata de un sleep ce permite modulului procesarea comenzii si transmiterea raspunsului inapoi. De asemenea fieacre comanda
este incercata de mai un numar fixat de ori, in cazul in care esueaza. Dupa ce am reusit sa ne conectam la AP dorit afisez pe LCD adresa ip obtinuta.
De asemenea LCD-ul mai este folosit si pentru afisarea comenzilor primite ulterior pentru controlul masinii.

Rezultate Obţinute

Am obtinut o masinuta perfect fuctionala care respecta toate lucrurile pe care mi le-am propus si pe care le-am prezentat in descrierea proiectului.
Probabil componentele puteau fi aranjate mai ordonat pe placuta si pe breadboard dar am incercat gruparea cat mai mult a firelor pentru a nu exista o senzatie
de haos.




BTN
RESET
UART2
PM



Concluzii

- Partea hardware poate fi cel putin la fel de captivanta ca si cea software.
- Acest proiect a fost cel mai complex si mai frumos de pana acum din facultate.
- Alegerea unui proiect in urma caruia voi obtine un lucru de care ma voi putea bucura m-a ambitionat in realizarea lui.
- Modulul de wireless ESP8266 folosit este foarte greu de programat in absenta unui covertor USB la UART si nu prea se gasesc tutoriale pentru programarea sa. Am reusit sa il programez de abia dupa ce am studiat setul de comenzi disponibile si conditiile necesare de configurare pentru conectarea la internet.

Download

Arhiva zip

Jurnal

- 21.04 Adaugare introducere, descriere generala, schema bloc si piese folosite.
- 22.04 Am terminat de lipit placuta si am incarcat bootloader-ul.
- 25.04 Am cumparat componentele necesare.
- 03.05 Realizare schema electrica.
- 08.05 Am lipit modulul STEP-DOWN, ledurile impreuna cu rezistentele si am reusit sa alimentez placuta de la baterii.
- 09.05 Am lipit driverul de motoare si modulul wireless.
- 10.05 Am verificat functionarea driverului de motoare si m-am chinuit, fara succes sa ma conectez la internet.
- 11.05 Am reusit sa ma conectez la internet folosind modulul wireless si am scris partea de soft.
- 14.05 Am cumparat un LCD1602, l-am conectat si l-am facut sa afiseze adresa IP obtinuta + comenzile primite.

Bibliografie/Resurse

4a-esp8266_at_instruction_set_en.pdf
doc8272.pdf
ESP8266
drv8835.pdf
Această pagină în format PDF