



Masinuta Inteligenta

Student : Herman Felician-Nicu

Profesor Indrumator : Andrei Diaconescu

Universitatea Tehnica Cluj-Napoca

Facultatea de Automatica si Calculatoare

Specializare Calculatoare – an 3

Contents

1.Introducere	2
2.Proiectare.....	2
3.Implementare.....	4
4.Echipament Utilizat	6
5.Manual de utilizare.....	6
6.Concluzii.....	7
7.Dezvoltari ulterioare.....	7

1.Introducere

1.1 Context

Încă de mici am avut contact cu tehnologia, cu acele mașinuțe controlabile de la telecomandă de care ne bucuram din toată inima. Cu trecerea anilor lucrurile s-au schimbat și am ajuns de la a cumpăra o mașinuță, la a crea o mașinuță. Lumea este conștientă cât de mare viteză prinde evoluția tehnologiei în fiecare zi, roboții, mașinuțele devenind din ce în ce mai complexe și capabile de realizarea unor lucruri incredibile. Această mașinuță a fost creată fiind capabilă de realizarea unor lucruri destul de interesante care o să fie prezentate în continuare.

1.2 Obiective

Prin conectarea unor cabluri la motorase, a unor senzori la plăcuțe și prin montarea a niste roți întindem spre „aducerea la viață” a unei mașinuțe inteligente.

Aveam în plan construirea mașinuței care să :

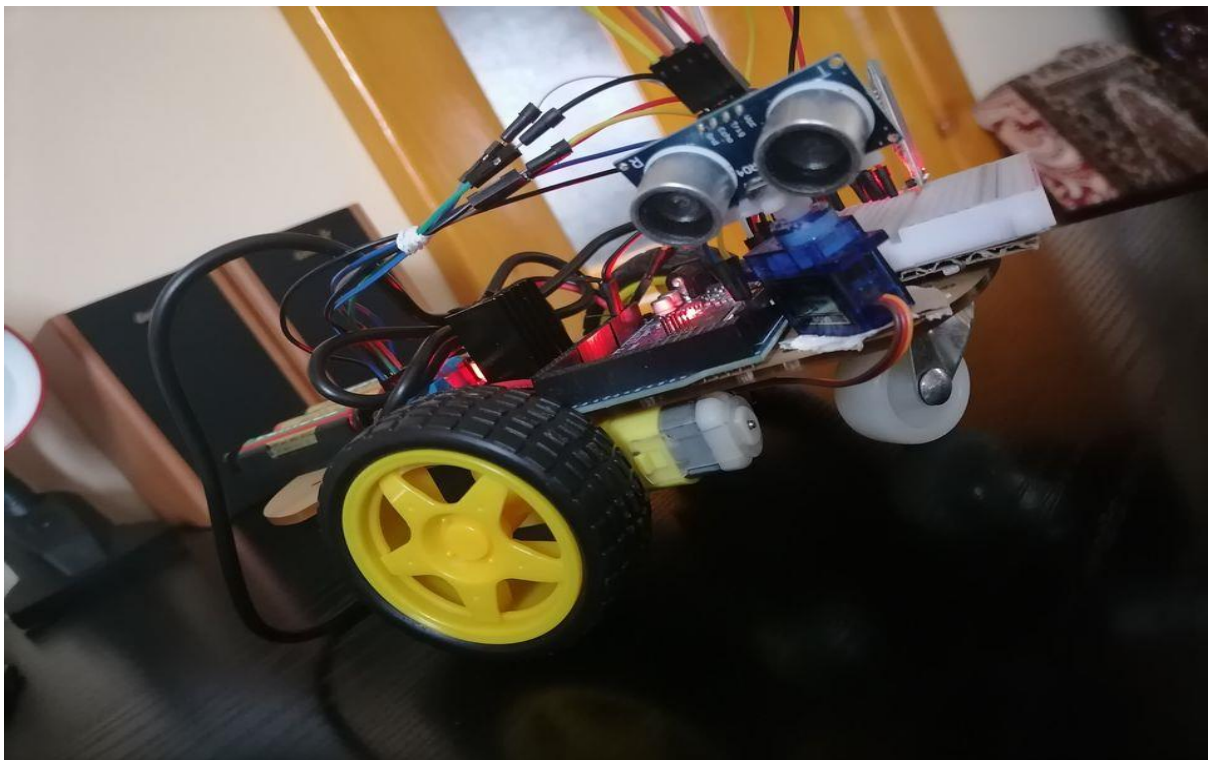
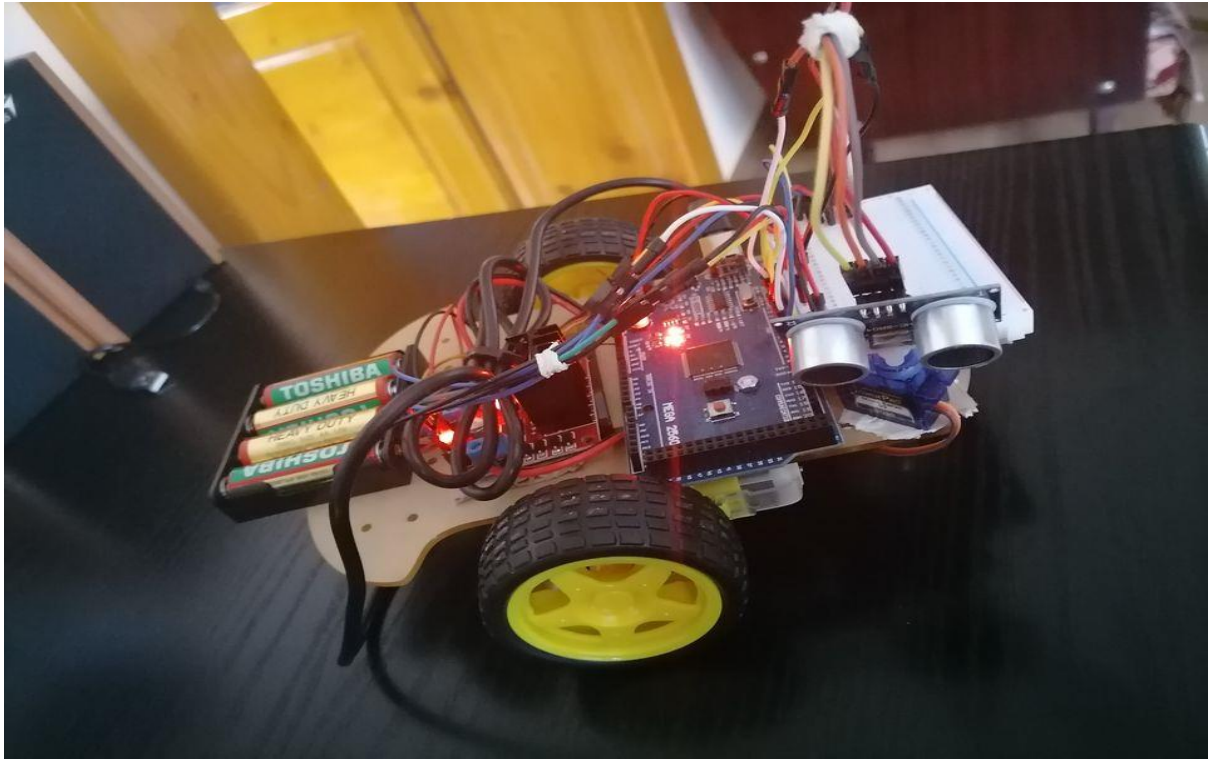
- nu se sinucida
- fie controlată de la butoane folosind o aplicație android
- fie controlată vocal folosind telefonul mobil
- se conecteze la telefon prin bluetooth
- realizeze parcare laterală dacă suprafața locului de parcare este îndeajuns de mare

2.Proiectare

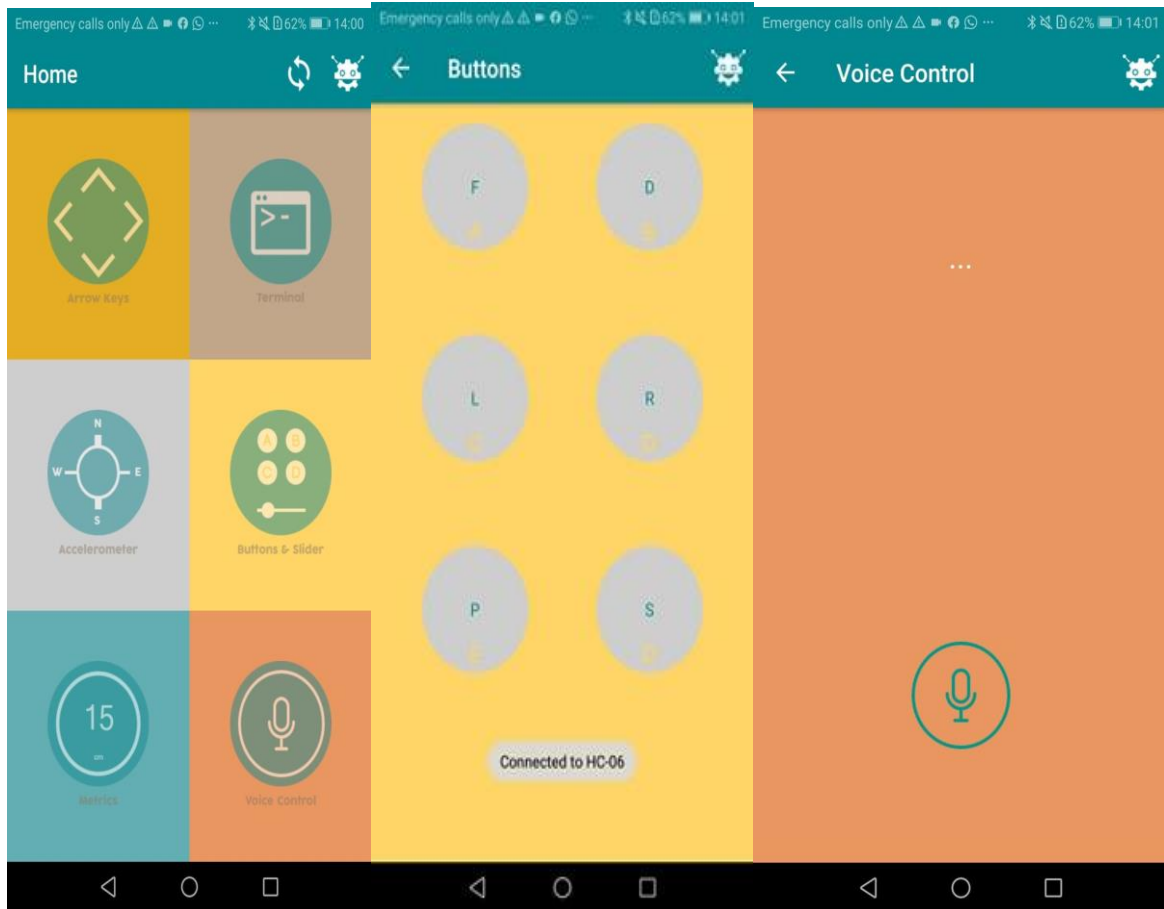
Asamblarea a pornit prin montarea kitului pe care urma să fie adăugată placa mega și alte componente. Proiectarea fizică a fost relativ scurtă, piesele potrivindu-se perfect încoronând o grozavă mașinuță.



Masinuta Finala



Partea de aplicatia android am luat-o de pe MagazinPlay, avand urmatoarea interfata:



Interfata Generala

Pagina Butoanelor

Pagina Voice Control

3.Implementare

Algoritmul din spatele functionalitatii acestei masinute a fost gandit astfel :

Pentru evitarea sinuciderii avem un senzor sonor, pentru verificarea distantei dintre masinuta si orice obstacol.

Transmiterea comenzilor se face prin modulul bluetooth, utilizand Serial1 care transmite de pe aplicatia android catre placuta comezi precum „F” – pentru a merge inainte (atunci cand se apasa butonul F sau cand ii e transmis „forward”) , „L” – pentru a vira la stanga (cand se apasa buton L sau cand ii e transmis „left” prin comanda vocala).

In cazul transmiterii comenzilor folosind voice control, datele o sa ajunga cu putin delay la placuta deoarece aplicatia android functioneaza folosind google translate(pentru recunoasterea cuvintelor), dupa receptarea corecta a cuvintelor(in cele mai multe cazuri sunt necesare mai multe incercari) aplicatia recunoaste comanda respectiva ca fiind una custom si abia apoi transmite datele catre placuta, care proceseaza si executa o anumita operatiune in functie de ce primeste pe Serial1.

Aceasta logica de functionare este implementata in functia „moveManagement” care primeste ca parametrii comanda, in forma de un caracter, motorasele, viteza si limita distantei la care sa se opreasca in cazul unui obstacol. Motorasele sunt pornite prin functia „StartMotor” si sunt oprite prin „delayStopped”.

Partea de parcare laterala se gaseste in „parkRight” unde se calculeaza paramterii parcarii si se aproba sau interzice parcare, in functie de datele preluate de la senzor. Senzor care citeste distanta in functia „readDistanceSensor” si returneaza distanta de la senzor catre cel mai apropiat obiect de pe directia curenta. Directia spre care priveste senzorul este modificata de catre motorasul servo, prin functia „playWithServo” care asigura capabilitatea masinutei de a executa absolut orice miscare ceruta de utilizator.

Functia StartMotor()

Daca viteza primita ca parametru este 0, motorasul se opreste, altfel daca primim comanda de a merge inainte, o sa pornim motorasele in fata, altfel in spate.

Functia delayStopped()

Opreste functionalitatea motoraselor.

Functia moveMangement()

Daca se primeste comanda F de la aplicatie, se verifica limita ca fiind mai mare de 30, daca e asa motorasul servo primeste „inainte” si masinuta porneste, altfel se opreste. In acelasi mod o sa functioneze si in cazul comenzilor R(dreapta), L(stanga). In cazul comenzii de mers inapoi, aceasta nu e constransa de nimic. Comanda de parcare apeleaza functia parkRight() care o sa fie descrisa ulterior.

Functia parkRight()

Pentru a realiza parcare masinutei intre 2 obiecte(parcare laterala) am gandit astfel: In primul rand o sa verificam distanta de la obiectul de langa noi(cel mai apropiat), o memoram, pe urma incercand sa calculam distanta dintre cele 2 obiecte. Acest lucru l-am realizat astfel : am verificat pe rand distanta in 3 puncte, la difetenta de 20 de grade(folosind motorul servo), luand minimul dintre ele ca fiind punctul final al celui de-al doilea obiect. Distanța dintre ele am obintut-o prin efectuarea diferentei dintre distantele celor 2 obiecte. In cazul in care distanta este cuprinsa intre 35 si 90, se efectueaza parcare, altfel masinuta o sa stea pe loc.

Functia playWithServo()

Am implementat aceasta functie pentru motorasul servo pe care am montat un senzor de distanta, aceasta combinatie senzor-motoras furnizand toate distantele de care avem nevoie pentru o buna functionalitate a acestei masinute.

In final am apelat in functia Loop() functia moveManagement() care manageriaza miscarea in toate cazurile.

4.Echipament Utilizat

Pentru aducerea „la viața” a acestei masinute am folosit următoarele lucruri :

- kit sasiu care contine : - 1 x Carcasa vehicul
2 x Motoare electrice
2 x Cauciucuri
2 x Codificatoare de viteza
2 x Elemente de fixare
1 x Roata universala
1 x Surub
1 x Cutie de baterii (bateriile nu sunt incluse)
Toate suruburile si piulitele necesare
- placuta Mega 2560
- punte H (L298N)
- breadbord
- modul bluetooth(HC06)
- senzor ultrasonic(distanta) (HC-SR04)
- servomotor (SG90)
- fire
- baterie externa(pentru alimentarea placutei)
- baterii de 1.5V (pentru motorase)

5.Manual de utilizare

Pentru inceput, se conecteaza placuta la bateria externa, urmand conectarea celor 4 baterii pentru motorase. Apoi, utilizatorul intra pe aplicatia android, se conecteaza la masinuta prin bluetooth si incepe transmiterea comenzilor. Din pagina butoanelor se pot apasa următoarele butoane :

- Butonul A (F) – transmite forward(inainte)
- Butonul B (D) – transmite back (inapoi)
- Butonul C (L) – transmite left(stanga)
- Butonul D (R) – transmite right(dreapta)
- Butonul E (P) –transmite park(parcheaza)
- Butonul F(S) – transmite stop(oprire)

In cazul comenzii vocale, doar se apasa pe butonul cu microfon si se transmite comanda vocala.

Comenzile inregistrate sunt : -forward(transmite F – inainte)

-back (transmite D – inapoi)

-right(transmite R – dreapta)

-left(transmite L – stanga)

-stop(transmite S – stop)

-park(transmite P – parcheaza)

6.Concluzii

În final, se poate spune că mașinuța funcționează foarte bine, nu are bug-uri nerezolvate sau erori de funcționare, cel puțin nu am descoperit nimic de genul deocamdată. Au existat probleme pe parcurs la realizarea alimetării cu energie electrică a plăcuței și a motoraselor, fiind insuficiente cele 4 baterii mici. Apoi, după ce montajul fizic a fost realizat, la implementare au fost mici probleme la evitarea sinuciderii (am pus distanță fiind 30, deoarece mereu exista un delay la citirea distanței motiv pentru care dacă distanța era mai mică, mașinuța apuca să se lovească de perete). Mai apoi, în cazul parării, determinarea distanței până la obiectul îndepărtat funcționa bine doar în unele cazuri, când obiectul era exact pe linie cu unghiul motorasului servo, motiv pentru care am ales să îmbunătățesc asta prin introducerea unui artificiu, calcularea unui minim dintre distanțele „zonale”. Pentru mersul înapoi, nu am avut cum implementa evitarea sinuciderii, doar având un nou senzor sonor, lucru care se poate dezvolta ulterior.

Modul în care funcționează mașinuța este vizibil în următoarele videoclipuri :

<https://www.youtube.com/watch?v=hRVIRDJmLvg&t=6s>

<https://www.youtube.com/watch?v=64NNoo73TT8>

7.Dezvoltari ulterioare

Că și dezvoltări ulterioare mașinuța poate să fie îmbunătățită astfel :

- adaugarea unui senzor în spate pentru a evita sinuciderea în absolut toate cazurile
- verificarea obiectelor din față prin mișcarea continuă a senzorului (dacă obiectul nu este exact pe unghiul senzorului, există șanse să se lovească de obiecte din lateral)
- implementarea unei ieșiri din parcare, după ce aceasta a fost realizată cu succes
- adaugarea unei carcase pentru îmbunătățirea aspectului