

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Line follower

propusă de

Ioana Valentina Muscă

Sesiunea: *februarie, 2019*

Coordonator științific

Lect. Dr. Cosmin Vârlan

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI

FACULTATEA DE INFORMATICĂ

Line follower

Ioana Valentina Muscă

Sesiunea: *februarie, 2019*

Coordonator științific

Lect. Dr. Cosmin Vârlan

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP,

absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de

..... specializarea, promoția

....., declar pe propria răspundere, cunoscând consecințele falsului în

declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elaborată sub îndrumarea dl. / d-na

_____, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul “Line follower”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, data

Absolvent Prenume Nume

(semnătura în original)

Cuprins

Introducere	6
Motivația lucrării	6
Obiectivul lucrării	7
Sumar	7
Contribuții	7
1.Componente hardware	8
1.1 Arhitectura hardware	8
1.1.1 Arduino	9
1.1.2 Senzor ultrasonic	11
1.1.3 Senzori infraroșu	12
1.1.4 Motoare.....	13
1.1.5 Punți H.....	14
1.1.6 Modul wireless	15
2.Implementarea soluției	17
2.1 Arhitectura software	17
2.2 Controlul robotului	18
2.2.1 Detecția obstacolelor	18
2.2.2 Controlul motoarelor	20
2.3.3 Urmărirea traseului (Line follower)	21
2.3 Controlul prin telecomandă	23
2.4 Funcția loop()	24
3.Concluzii	25
4.Bibliografie	26

5. Anexe	27
Anexa 1	27
Anexa 2	28

Introducere

Motivația lucrării

Industria automotive s-a dezvoltat considerabil în ultimii ani, a pus accentul din ce în ce mai mult pe tehnologiile de asistență a șoferului. Aceste tehnologii au menirea de a strânge și prelucra informații din mediul exterior: existența unor obstacole, distanța dintre autovehicule sau a pietonilor. În unele țări există chiar autoturisme ce se pot parca din telecomandă, aceasta fiind chiar cheia mașinii dar până în prezent chiar și cu aceste noi sisteme pentru autovehicule încă nu ne putem baza pe acestea să se conducă singure în orice circumstanțe.

Jaguar Land Rover a făcut un pas important în dezvoltarea mașinilor autonome, britanicii prezentând recent aplicația care îți permite să conduci autoturismul cu ajutorul telefonului mobil.

Aplicația își propune să transforme telefonul mobil într-un fel de telecomandă avansată pentru autoturism, însă care să-ți permită mult mai multe decât să închizi și să deschizi portierele.

Smartphone-ul ar fi practic un volan care va funcționa și din exteriorul mașinii, cei de la Jaguar Land Rover dorind să implementeze inclusiv o comandă vocală.

Această nouă tehnologie îi va veni în ajutor șoferului pentru a parca în locuri strâmte, pentru a scoate autoturismul dintr-o parcare în care a fost flancată de alte mașini sau pentru a depăși zone cu teren accidentat

Prin aplicația respectivă, șoferul controlează viteza și direcția automobilului. Sistemul poate vira, întoarce, accelera și frâna pentru a realiza orice tip de mișcare, chiar și o întoarcere la 180 de grade.

Noua tehnologie are totuși câteva limitări: în cazul utilizării acesteia, viteza maximă nu va depăși 6.5 kilometri pe oră, iar telefonul mobil trebuie să fie la maximum 10 metri distanță de mașină.

Dacă vine vorba despre condițiile meteo, inginerii britanici garantează că aplicația va funcționa indiferent dacă va ninge sau va ploua torențial.

În acest moment vorbim doar despre un prototip aflat în faza de testare, însă producția în serie a modelelor dotate cu acest sistem ar putea începe în 2020. [1]

Obiectivul lucrării

Lucrarea *Line follower* încearcă dezvoltarea unei soluții viabile pentru parcurgerea unui traseu bine stabilit, detecția de obstacole și controlarea acestuia printr-o telecomandă. Procesarea algoritmului și prototipizarea conceptului este făcută de o placă de dezvoltare de tip Arduino. Motoarele, senzorii infraroșii, senzorul ultrasonic și șasiul reprezintă modulele input-output ale robotului, care împreună fac tot ansamblul prezentat.

Sumar

Cele trei capitole ale lucrării de licență sunt:

- Componente hardware
- Implementarea soluției
- Evoluția și rezultatele soluției

În primul capitol, *Componente hardware*, este prezentată schema hardware a robotului, descrierea caracteristicilor, modul de funcționare și argumentarea alegerii resurselor folosite în aplicație.

În continuare, capitolul 2, *Implementarea soluției*, este descris scheletul aplicației alcătuit din blocuri de funcții menite să pună în mișcare *Line follower-ul*. Acest capitol este împărțit în două subcapitole: - Detecția și recunoașterea traseului;
- Controlul prin telecomandă.

În ultimul capitol este descrisă evoluția aplicației însoțită de rezultatele acesteia în urma dezvoltării prototipului de *Line follower*.

Contribuții

Contribuția la această aplicație a constat în adaptarea și optimizarea componentelor hardware folosite. Având în vedere că această aplicație nu a fost dezvoltată pentru a putea fi rulată pe un calculator, ci pe plăci de dezvoltare Arduino ce au puterea de calcul mică. Arhitectura software a aplicației a trecut prin mai multe etape până a ajuns la soluția finală și optimizată.

O altă contribuție constă în folosirea unei imprimante 3D, ce a avut ca scop adaptarea și tipărirea de componente, în special la poziționarea senzorilor pe scheletul robotului dar totodată o mai bună asamblare a pieselor.

1. Componente hardware

1.1. Arhitectura hardware

Din punct de vedere hardware, aplicația Line follower este împărțită după microcontrolere în două module:

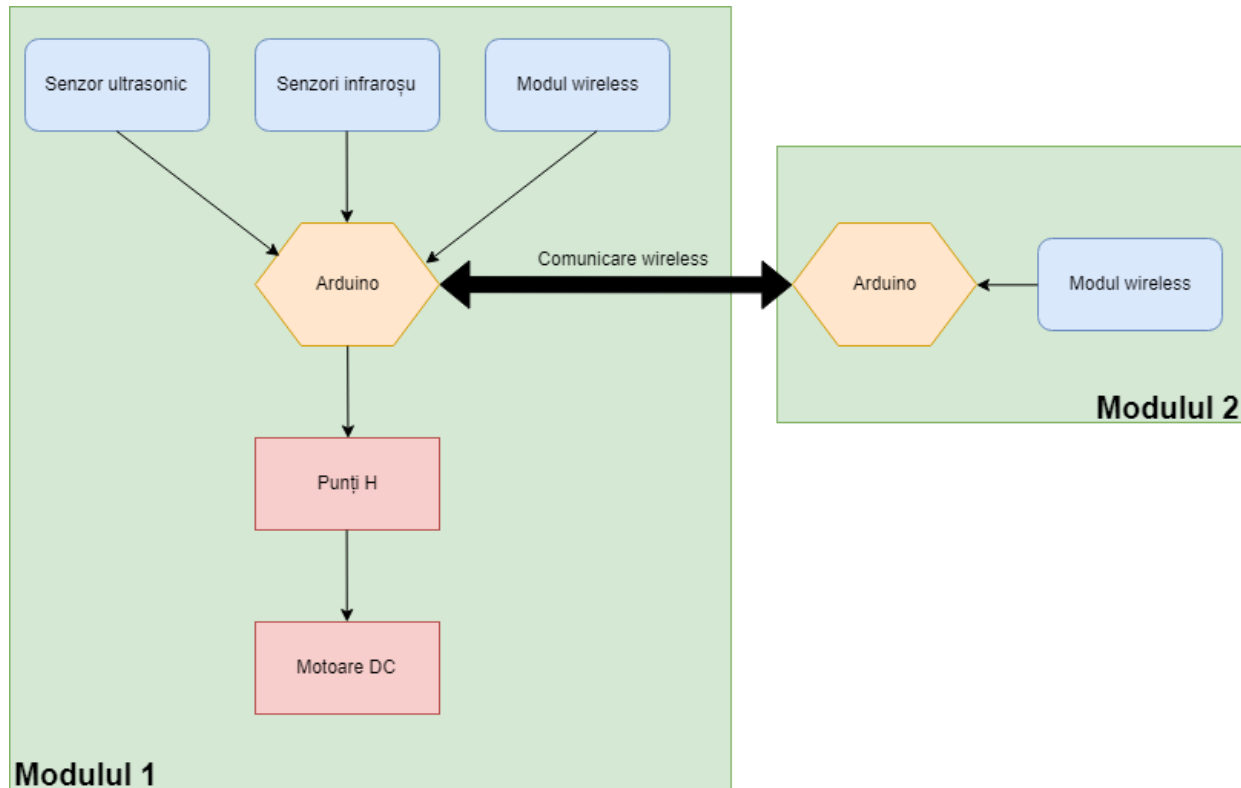


Fig. 1-Arhitectura hardware a aplicației

Senzorii infraroșu și senzorul ultrasonic au ca rol colectarea de date din mediul exterior, acestea fiind primite și procesate de către Arduino. Punțile H. în funcție de informațiile primite de la modulele input transmit comenzi mai departe motoarelor. A doua plăcuță Arduino și modulul wireless formează un modul separat de robot, comunicarea dintre acestea două este wireless, modulul doi are rol de telecomandă asupra modulului unu.

În Anexa 1 și Anexa 2 se poate regăsi schemele generale de conectare a componentelor robotului.

1.1.1.Arduino

Am ales plăcuțele de dezvoltare să fie Arduino deoarece acestea sunt platforme de prototipare open-source, bazate pe software și hardware flexibil și simplu de folosit.

Ele sunt capabile de a prelua date din mediul înconjurător printr-o serie de senzori și de a efectua acțiuni asupra mediului prin intermediul luminilor, motoarelor, servomotoare, și alte tipuri de dispozitive mecanice. Procesorul este capabil să ruleze cod scris într-un limbaj de programare care este foarte similar cu limbajul C++.

Specificații Arduino UNO :

- Microcontroler: ATmega328
- Tensiune de lucru: 5V
- Tensiune de intrare (recomandat): 7-12V
- Tensiune de intrare (limita): 6-20V
- Pini digitali: 14 (6 PWM output)
- Pini analogici: 6
- Curent per pin I/O: 40 mA
- Curent 3.3V: 50 mA
- Memorie Flash: 32 KB (ATmega328) 0.5 KB pentru bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz [2]



Fig.2-Arduino UNO

Specificații Arduino Mega:

- Tensiune de funcționare: 5V;
- Tensiune de alimentare Jack: 7V - 12V;
- Pini I/O: 54;
- Pini PWM: 15 (din cei de I/O);
- Pini analogici: 16;
- 4 x UART;
- Memorie flash: 256KB, din care 8KB ocupați de bootloader;
- Frecvență de funcționare: 16MHz.
- Dimensiuni: 5.3cm x 10cm. [3]

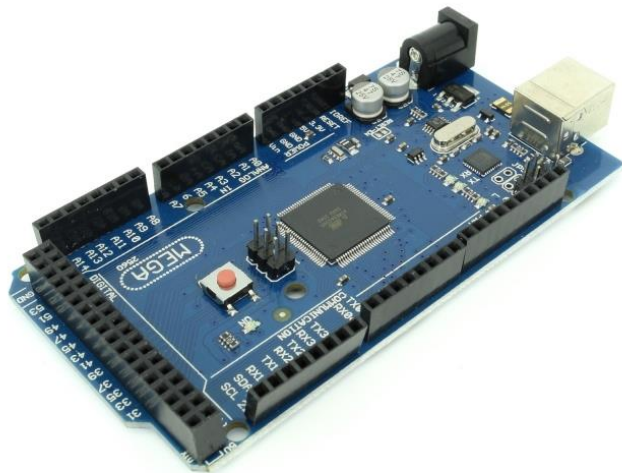


Fig.3-Arduino MEGA

1.1.2. Senzor ultrasonic



Fig.4-Senzor ultrasonic HC-SR04⁵



Fig. 5-Mod funcționare HC-SR04⁵

Senzorul ultrasonic HC-SR04 este unul dintre cei mai utilizați senzori pentru aflarea distanței. În special folosit pentru proiectele cu plăci de dezvoltare Arduino, are avantaje față de senzorii analogici, necesitând doar pini I/O digitali și are imunitate mai mare la zgomotul din jur. Senzorul emite ultrasunete la o frecvență de 40000Hz care circulă prin aer, iar dacă întâlnește un obstacol, acesta se va întoarce înapoi spre modul, astfel, luând în considerare viteza sunetului se poate calcula distanța până la obiect.

Specificații :

- Tensiune de alimentare: 5V
- Curent consumat: 15mA
- Distanță de funcționare: 2cm - 4m
- Unghi de măsurare: 15 grade
- Eroare de doar 3mm
- Durată semnal input: 10us
- Dimensiuni: 45mm x 20mm x 15mm [4]

1.1.3. Senzori infraroșu

Specificații :

- Tensiune de alimentare: 3V - 5V;
- Distanță sesizare obstacol: 2cm - 30cm;
- Unghi observare obstacol: 35°;
- Output digital;
- Comparator LM393;
- Tensiune de referință reglabilă.

Senzorul de obstacole se bazează pe reflexia radiației IR de către obstacol. Radiația IR este emisă de către un LED și este recepționată de către un fototranzistor. Modulul poate fi folosit și pentru a detecta linia albă sau neagră de către roboți. Output-ul senzorului este digital. Distanța de detecție poate fi reglată dintr-un potențiometru.

Modulul conține două led-uri indicatoare, unul pentru alimentare și celălalt pentru detectarea obstacolului.

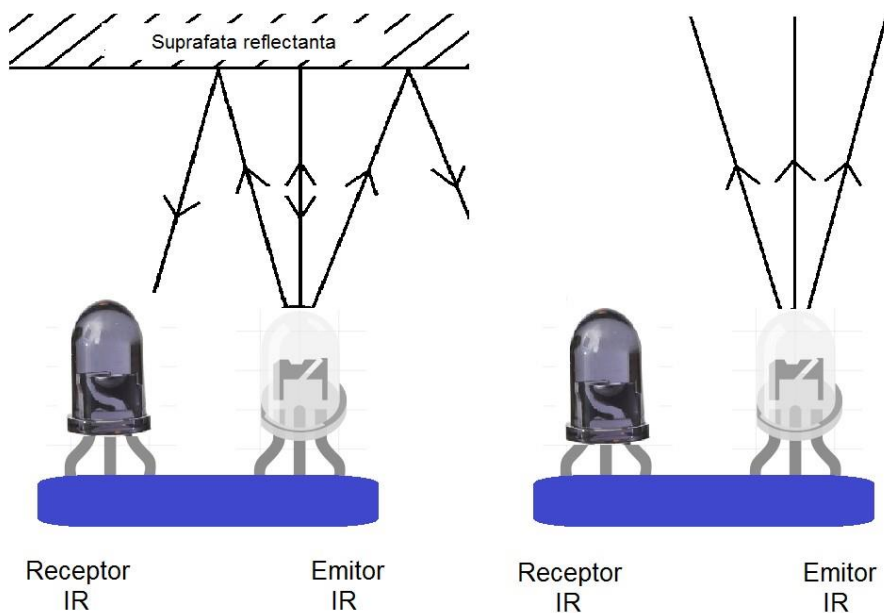


Fig.6-Mod funcționare senzor infraroșu

Modulul se alimentează la o tensiune de 3-5 V și are un unghi de observare a obstacolelor de 35°. Cu ajutorul potențiometrului putem modifica nivelul de referință, astfel încât să ajustăm sensibilitatea, deci, distanța la care modulul detectează obstacole. Comparatorul furnizează la ieșire 1 logic atunci când nu detectează obstacole și 0 logic atunci când întâlnește un obstacol.). [5]

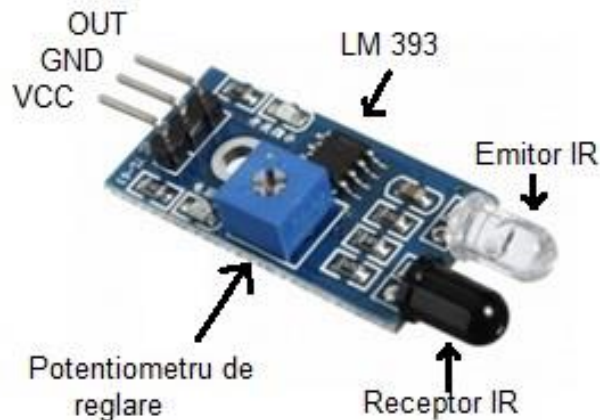


Fig.7-Senzor infraroșu

1.1.4. Motoare

Cele 4 motoare HC02-48 selectate pentru deplasarea robotului sunt standard DC, cu o tensiune de alimentare 3-6V și reductor 1:48 și au fost alese în mare parte datorită costului redus.



Fig.8-Motor DC

Acestea se pot comanda prin intermediul a doi conectori : dacă se aplică o diferență de tensiune, motorul se învârtă în direcția stabilită de sensul curentului.

La o tensiune de comandă de 5V, motorul poate atinge o viteză de aproximativ 10000 rpm fără reductor (respectiv 208 rpm cu reductor) și un cuplu de 0.8 kg/cm. Viteza poate varia prin intermediul tensiunii de comandă. Din prisma faptului că acestora le lipsesc ieșirile analogice (unde

în mod normal tensiunea ar putea fi reglată) reglarea tensiunii de ieșire se face cu ajutorul unui semnal *PWM* (*Pulse-width modulation*).

PWM-ul se realizează prin alternarea rapidă a tensiunii de ieșire prin *ON/OFF*, aceste perioade în care semnalul este pe *ON* raportat la perioada semnalului poartă numele de duty cycle și reprezintă modul în care tensiunea de ieșire poate fi „variata”.

1.1.5. Punți H

Specificații :

Tensiune motoare: 5V - 35V;

Tensiune circuite logice: 5V;

Curent motoare: 2A (MAX);

Curent logica: 36mA;

Frecvență maximă pwm: 40kHz.

Dimensiuni: 43 x 43 x 27 mm.

Driver-ul conține și un limitator de tensiune liniar, astfel că atunci când tensiunea de alimentare a motoarelor este >7V, nu este nevoie să alimentăm separat partea de logică.

Driver-ul este unul dual, putând să controleze două motoare. El poate fi folosit și pentru motoare pas cu pas.

Chiar dacă are dimensiuni mai mari, este util prin faptul că beneficiază de un radiator destul de mare și disipă o cantitate mare de căldură.

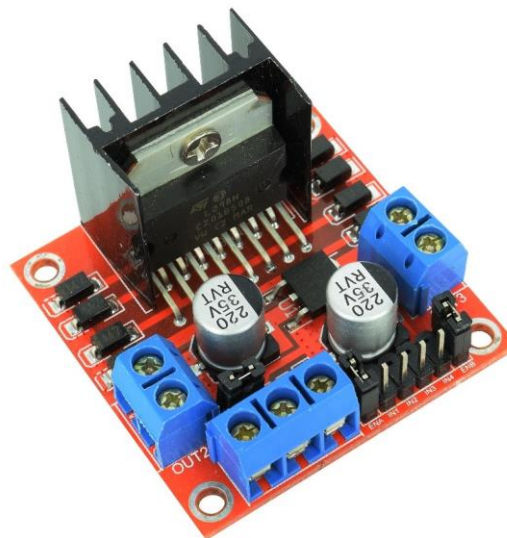


Fig.9 Driver LN298N

Conexiuni :

- **VCC** – se conectează la sursă de putere pentru alimentarea motoarelor.
- **GND** – se conectează la masă
- **5V**: poate fi folosit ca ieșire constantă de 5V
- **Out 1, Out 2**: ieșiri de putere pentru motorul A
- **Out 3, Out 4**: ieșiri de putere pentru motorul B
- **In1, In2**: intrări logice de comandă pentru Out1, Out2
- **In3, In4**: intrări logice de comandă pentru Out3, Out4

- **EnA, En B:** Linii de *enable* pentru grupurile Out1, Out2 respectiv Out3, Out4. Doar când tensiunea de comandă pe aceste inputuri este ON, outputurile de putere sunt comandate conform In1, In2, respectiv In3, In4.

Pentru a controla motoarele DC cu driverul L298N ne asigurăm că pinii EnA și EnB nu sunt conectați la 5V. Pe aceștia îi vom conecta la pinii PWM 3 și 10 pentru a controla turația motoarelor. Pentru a controla sensul motoarelor: pe In1 scriem HIGH, iar pe In2 scriem LOW și motorul va merge înainte. Pentru a inversa sensul scriem LOW, respectiv HIGH pe In1, In2. Similar pentru motorul B. De asemenea, GND-ul driverului trebuie conectat cu GND-ul plăcuței Arduino pentru a funcționa.

Deoarece acest driver este conceput pentru 2 motoare iar robotul prezentat în această lucrare conține 4 motoare, am legat motoarele două câte două iar mișcările executate de acesta sunt asemenea șenilelor unui tanc. [6]

1.1.6. Modul wireless

Optimus Digital

Specificații :

- Tensiune de alimentare: 1.9 - 3.6V;
- Consum curent de 11.3mA la emisie cu o putere de 0 dBm;
- Consum curent de 13.5mA la recepția datelor cu 2 Mbps;
- Consum curent de 26uA în modul standby;
- Consum curent de 900nA în modul power down;
- Viteza de 250kbps, 1Mbps sau 2Mbps;
- Frecvență de funcționare de 2.4 GHz;
- Există pini toleranți la 5V;
- Interfață de comunicație SPI;
- Poate trimite date în câmp deschis până la 80m.

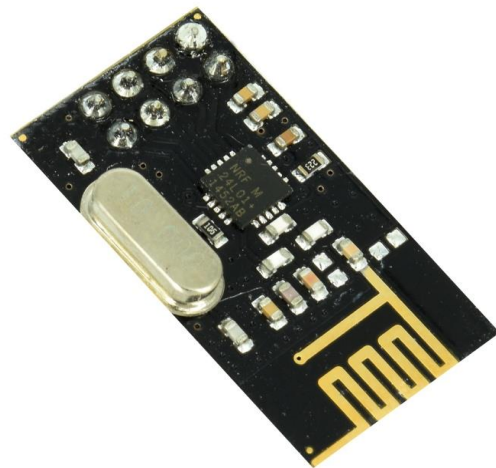


Fig.10 Modul Transceiver nRF24L01 (2.4 GHz)

Modul wireless bazat pe circuitul integrat nRF24L01 care operează în banda ISM standard de 2.4 GHz, capabil de viteze de până la 2 Mbps. Acest modul dispune de intrări care tolerează tensiuni de până la 5 V, fiind ușor de interfațat cu plăci de dezvoltare populare precum Arduino.

Modulul se alimentează la tensiuni de 3.3 V și comunică prin interfața SPI. Poate trimite date în spațiu liber până la o distanță de 80 m. Modulul este foarte popular, fiind folosit deseori în cadrul proiectelor dezvoltate la nivel hobby, dar și în aplicații industriale.

Modulul este ideal pentru construirea de periferice wireless, telecomenzi fără fir, aplicații de automatizări pentru casa inteligentă, RFID activ, aplicații VOIP și jucării wireless.

Am ales acest modul pentru că poate avea mai multe canale de comunicație peste aceeași frecvență, , poate transmite pe 125 de canale diferite (de la 0 la 124) cu o lățime de bandă de 1 MHz. [7]

2. Implementarea soluției

2.1. Arhitectura software

Aplicația *Line follower* este compusă din două module principale, asemenea arhitecturii hardware:

- I. Controlul robotului
- II. Controlul prin telecomandă

Primul capitol se ocupă cu comandarea motoarelor în funcție de traseu sau a obstacolelor. Controlul efectiv al robotului are la bază punțile H și motoarele, modulul primește date din mediul exterior de la senzorii infraroșu și senzorul ultrasonic. Informațiile preluate de la senzori și procesate au ca rezultat mișcarea robotului. Schema modulului 1 este prezentată la pagina 6.

Cel de al doilea modul, are ca scop transmiterea de comenzi de la telecomandă spre robot. Din punct de vedere hardware, acesta corespunde Modulului 2, ilustrat la pagina 6.

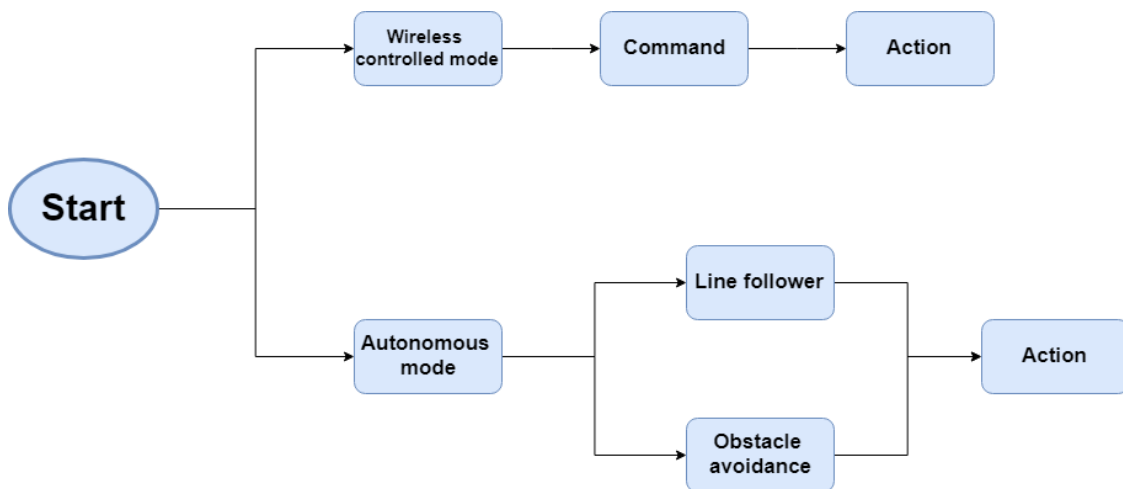


Fig. 11-Arhitectura software

În următoarele subcapitole voi descrie modul de funcționare a fiecărui modul, metodele abordate și rezultate obținute.

2.2. Controlul robotului

Acest modul a fost dezvoltat utilizând mediul de programare Arduino IDE. Aplicația este structurată în mai multe sub module, precum:

- a) Detecția obstacolelor
- b) Controlul motoarelor
- c) Urmărirea traseului (*Line follower*)

2.2.1 Detecția obstacolelor

Detectarea obstacolelor din partea frontală a robotului se face cu ajutorul unui senzor ultrasonic a cărui funcționare este descrisă în capitolul 1.2.1.

Asignarea pinilor și distanța minimă dintre un obstacol și robot este următoarea:

```
//ultrasonic distance sensor
#define trigPin          A5
#define echoPin          A4
#define brake            A0
#define minDistance      30
#define distanceFormula  0.017
```

Orice obiect situat la o distanță mai mare decât minDistance nu sunt considerate obstacole. O undă ultrasonică este generată pentru o perioadă de 10 μ s setând pinul trigPin pe *High*. Această undă va parcurge distanța până la primul obiect întâlnit cu o viteză egală cu viteza de propagare a sunetului în aer (340 m/s).

```
//set trigPin on HIGH for 10 microseconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(delay10);
//set trigPin on LOW
digitalWrite(trigPin, LOW);
//read echoPin
duration = pulseIn(echoPin, HIGH);
//calculate the distance to object
distance = (duration * distanceFormula);
//if the distance read from the sensor is greater than the distance set as minimum there is no obstacle
```

Determinarea duratei de timp este făcută de funcția *pulseIn()*. Funcția citește durata în microsecunde (μs) a unui puls de un anumit nivel logic (*HIGH* sau *LOW*), pe un anumit pin.

Dacă distanța calculată este mai mică decât pragul determinat de *minDistance*, atunci flagul obstacle este setat TRUE, altfel îi este asignată valoarea FALSE.

```
if (distance > minDistance) {
  //turn brake led off
  //display clear road on Serial Monitor
  obstacle = false;
  digitalWrite(brake, LOW);
  Serial.print("clear road");
}
else {
  //if there is an obstacle
  //turn brake led on
  //display obstacle detected on Serial Monitor
  obstacle = true;
  digitalWrite(brake, HIGH);
  Serial.print("obstacle detected ");
}
delay(delay5);
//return if there is an obstacle detected or not
return obstacle;
```

De exemplu, pentru un obiect aflat la o distanță de 10 cm față de senzor, timpul necesar undei pentru a ajunge la obiect și de a se întoarce este de 294 μs .

$$v = 340 \text{ m/s} \Rightarrow v = 0,034 \text{ cm/} \mu s \text{ (1)}$$

$$t = d / v \Rightarrow t = 10 \text{ cm} / 0,034 \text{ cm/} \mu s \Rightarrow t = 294 \mu s \text{ (2)}$$

$$d = t * v \Rightarrow d = t / 2 * 0,034 \text{ cm/} \mu s \text{ (3)}$$

Din relația (3) se deduce că durata de timp este împărțită la doi. Acest lucru se explică prin prisma faptului că durata este măsurată ca fiind intervalul de timp necesar undei să ajungă de la senzor la obiect și înapoi, de la obiect la senzor.

2.2.2. Controlul motoarelor

Motoarele sunt controlate de driverul de motoare L298N, descris în subcapitolul 1.2.4.

Pentru acestea au fost create următoarele funcții:

- void moveStraight()
- void steerLeft()
- void steerRight()

Pentru a se putea deplasa înainte robotul, toate motoarele trebuie să se învârtă cu aceeași viteză și direcție:

```
//move straight, left and right motors at the same slow speed
void moveStraight() {
    digitalWrite(leftInput1, LOW);
    digitalWrite(leftInput2, HIGH);
    digitalWrite(rightInput1, LOW);
    digitalWrite(rightInput2, HIGH);
    analogWrite(enableA, minSpeed);
    analogWrite(enableB, minSpeed);

    Serial.print("moveStraight with Speed: ");
    Serial.println("50");
    delay(delay10);
}
```

Cu ajutorul funcției *digitalWrite()* se stabilește sensul de rotație pentru cele patru motoare. Fiecărui motor îi sunt asignate doua ieșiri logice, unul pentru fiecare sens de rotație, iar motoarele sunt conectate două câte două asemenea unor șenile de tanc. Astfel, pentru a se învârti înainte se setează toți pinii corespunzători pe HIGH, iar cei corespunzători sensului de rotație opus pe LOW.

Prin apelul funcției *analogWrite()* se va trimite semnalul analogic către puntea H, care va comanda motoarele cu o viteză corespunzătoare valorii transmise. Motoarele vor fi în repaus dacă se va trimite valoarea 0, iar pentru valoarea de 255, motoarele se vor roti la viteza maximă, respectiv 208 rpm.

Pentru funcțiile *steerLeft()* și *steerRight()* motoarele de pe partea dreaptă se vor roti într-un anumit sens, iar motoarele de pe partea stângă se vor roti în sens opus. Pentru ca robotul să

se deplaseze spre dreapta funcția *steerRight()* va inversa valorile pentru pinii ce sunt conectați la intrările logice de pe parte dreaptă. Astfel, motoarele de pe partea dreapta vor merge în sens invers iar cele de pe partea stângă înainte, obținându-se virajul la dreapta a robotului. Pentru funcția *steerLeft()* se procedează analog, inversându-se sensurile motoarelor din stânga și dreapta.

2.2.3. Urmărirea traseului (*Line follower*)

Pentru acest obiectiv s-au folosit 4 senzori infraroșu, așa cum am descris în subcapitolul 1.2.2. Distanța de detecție poate fi ajustată prin potențiometrul în sensul acelor de ceasornic crește distanța de detecție; pentru a reduce distanța de detecție se rotește potențiometrul în sensul invers acelor de ceasornic. Acești senzori generează 2 valori de output, 1 dacă senzorul este deasupra liniei sau 0 dacă nu.

Robotul poate întâmpina următoarele cazuri pe traseu, în funcție de poziționarea senzorilor față de linie:

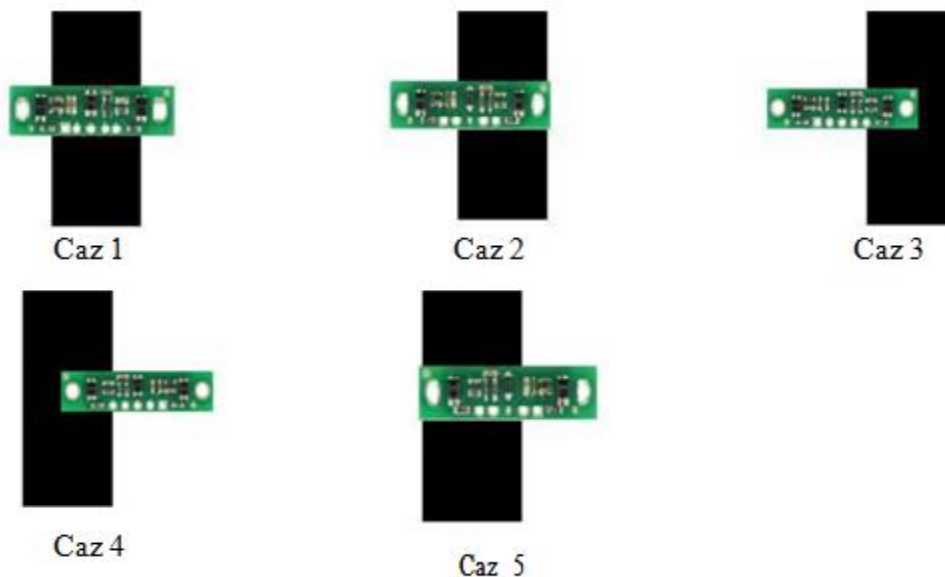


Fig.12 -Cazuri posibile(*Line follower*)

Caz 1: Cei doi senzori din mijloc se află pe suprafața liniei negre, iar robotul este situat corect și poate înainta.

Caz 2: Doi dintre senzori sunt deasupra marcajului, respectiv cei din dreapta. În acest caz robotul este încă pe traseu și poate înainta.

Caz 3: Doar senzorul din dreapta exterior este pe linie, robotul se abate de la traseu. În acest caz robotul trebuie să vireze la dreapta pentru a se redresa pe traseu.

Caz 4: Această situație este similară cu cea de la cazul 3 numai că pe partea stângă, pentru a putea continua traseul robotul trebuie să vireze la stânga.

Caz 5: Această situație este similară cu cea de la cazul 2, iar robotul își poate continua traseul.

Dacă robotul întâlnește un obstacol pe parcursul traseului toate cazurile de mai sus nu se mai iau în calcul iar acesta se oprește și aprinde led-ul ce indică întâlnirea unui obstacol.

```
leftOutsideS = digitalRead(leftOutsideSensor); //read leftOutside sensor
delay(infraredDelayTime);
leftInsideS = digitalRead(leftInsideSensor); //read leftInside sensor
delay(infraredDelayTime);
rightInsideS = digitalRead(rightInsideSensor); //read rightInside sensor
delay(infraredDelayTime);
rightOutsideS = digitalRead(rightOutsideSensor); //read rightOutside sensor
delay(infraredDelayTime);
```

Cu ajutorul funcției `digitalRead()` citim valorile de output generate de senzori.

O serie de rezultate pentru date de test pot fi observate în tabelul următor:

Senzor 1	Senzor 2	Senzor 3	Senzor 4	Decizie
0	1	1	0	moveStraight()
0	0	1	1	moveStraight()
1	1	0	0	moveStraight()
1	0	0	0	steerLeft()
0	0	0	1	steerRight()
0	1	1	1	moveStraight()
1	1	1	0	moveStraight()

2.3. Controlul prin telecomandă

Pe lângă parcurgerea unui traseu prestabilit pe parcursul căruia robotul întâlnește obstacole, aplicația oferă utilizatorului posibilitatea de a putea controla robotul. Acesta iese de pe modul Line Follower și execută comenzile date de către utilizator. S-a ales un modul wireless pentru această funcție, descrierea modului de funcționare și specificațiile tehnice se regăsește în subcapitolul 1.1.6.

Robotul este controlat de o telecomandă formată dintr-o plăcuță de dezvoltare Arduino, un joystick și un transceiver. După ce robotul trece de pe modul autonom pe cel manual, utilizatorul poate începe să transmită comenzi. Cu ajutorul transceiver-ului, acestea vor fi transmise, citite și procesate de aplicație.



Fig.13 –Mod funcționare transceiver

2.4. Funcția *loop()*

În funcția *loop()* sunt folosite toate funcționalitățile descrise mai sus, structura putând fi observată în figura de mai jos. Robotul poate funcționa în două moduri: modul line follower și cel cu ajutorul comenzilor transmise de către utilizator prin wireless. Utilizatorul poate controla robotul prin *joystick-ul* atașat pe telecomandă. În modul line follower robotul urmează traseul pe care se află și se oprește în caz că întâlnește un obstacol pe drum.

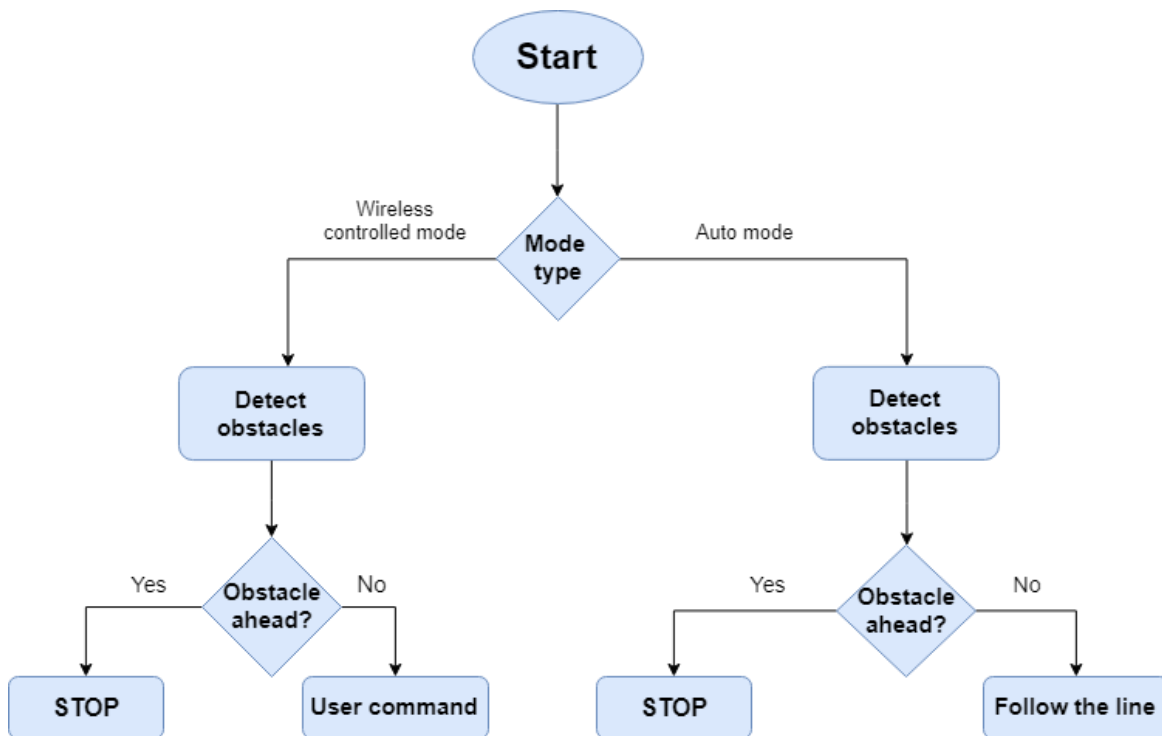


Fig.14 - Flowchart pentru funcția *loop()*

3. Concluzii

Scopul lucrării *Line follower* este să arate o posibilă soluție pentru conceperea unei aplicații capabile să primească date din mediul exterior, să le proceseze și să ia hotărâri în concordanță cu situația în care se află. În timpul procesului de dezvoltare am întâmpinat probleme, atât de natură software cât și de natură hardware. La construcția robotului am avut probleme la asamblarea componentelor, defecțiuni ale acestora, alimentarea lor și limitări de natură tehnică. După integrarea celor două module, a urmat sincronizarea și testarea robotului ca un ansamblu, urmând ca acestea să fie calibrate pentru a putea funcționa la parametrii normali.

Per ansamblu, aplicația își îndeplinește toate cerințele impuse. Robotul este capabil să urmeze un traseu format de o linie de culoare neagră și să oprească la întâmpnarea unui obstacol sau să funcționeze răspunzând unei serii de comenzi transmise de utilizator.

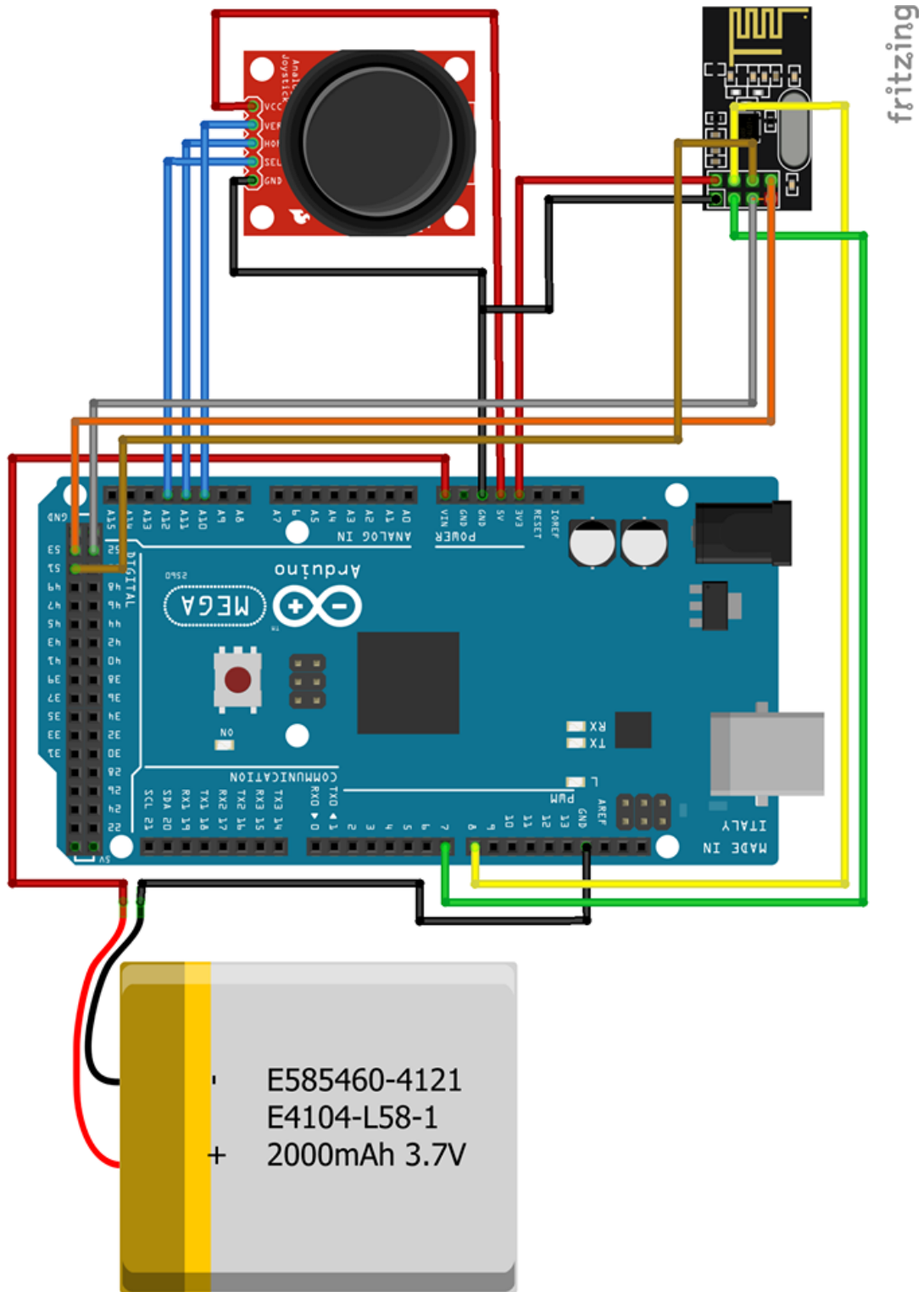
Pentru viitor, aplicația poate fi îmbunătățită hardware cu piese de o calitate mai bună care să primească datele din mediul exterior mult mai rapid și să ia decizia cea mai bună în acel context. La nivel software, o posibilă îmbunătățire este dezvoltarea deciziilor de direcție pe care le poate lua, crescând acuratețea și complexitatea sistemului.

Bibliografie

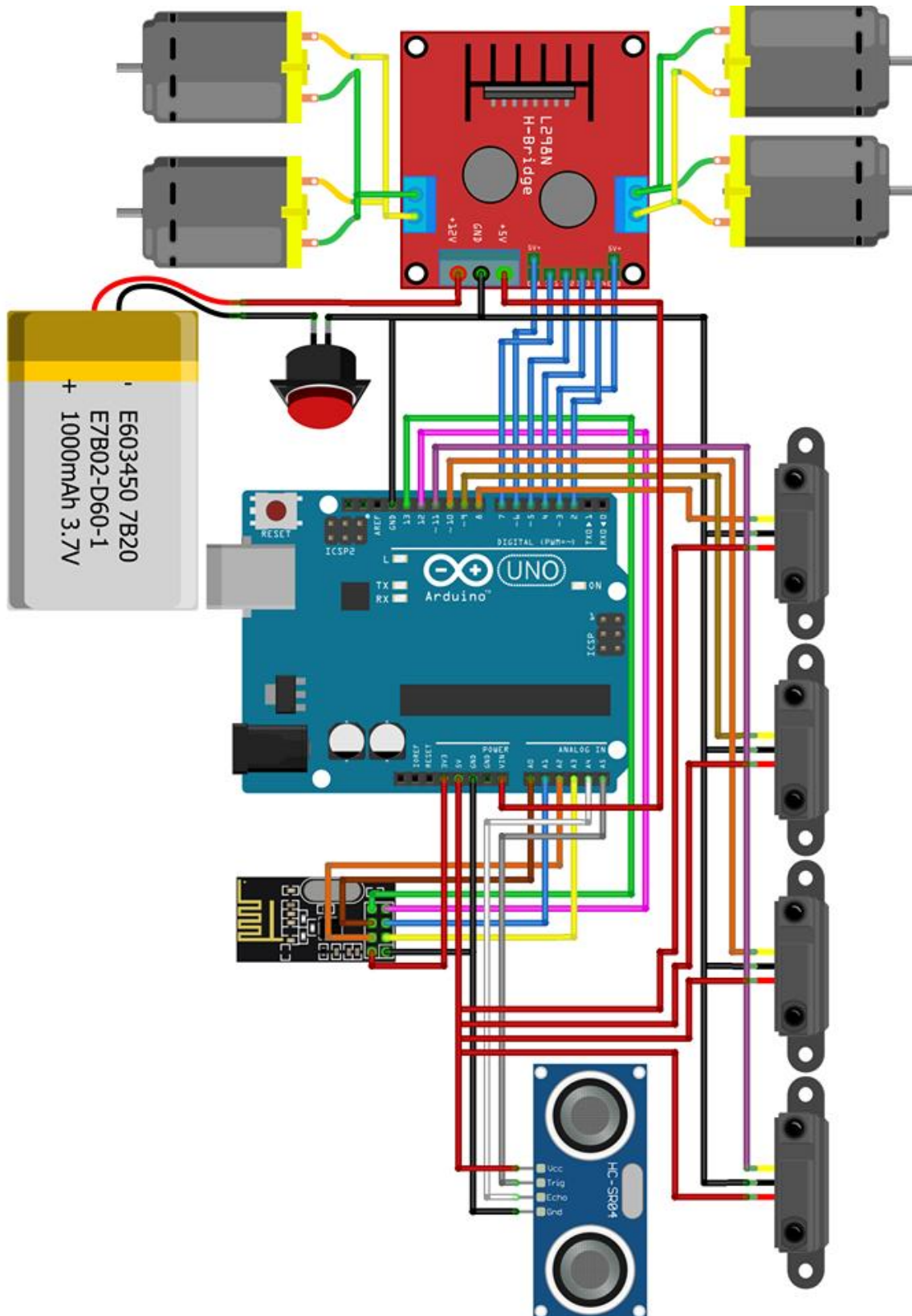
- [1] <https://www.4tuning.ro/inovatii-auto/masina-care-poate-fi-condusa-de-la-distanta-cu-ajutorul-telefonului-mobil-26633.html?fbclid=IwAR0GnFZVHm-XNs2rPk3Ek24wNUk2LR8qeuGCElRaDreZ7iK5Afi2jripxTo>
- [2] https://www.robofun.ro/arduino_uno_v3
- [3] <https://www.optimusdigital.ro/ro/compatibile-cu-arduino-mega/1687-clona-arduino-mega-2560-r3-atmega2560-atmega16u2.html>
- [4] <https://cleste.ro/senzor-ultrasonic-hc-sr04.html>
- [5] <https://www.optimusdigital.ro/ro/senzori-senzori-optici/163-senzor-infrarosu-de-obstacole.html>
- [6] https://www.optimusdigital.ro/ro/drivere-de-motoare-cu-perii/145-driver-de-motoare-dual-l298n.html?gclid=CjwKCAiAv9riBRANEiwA9Dqv1d_tE5Oz9DsBA3yms_Om1oIoQ6KOT08_t06EU2vtDT9SeWRhQWm9OBoCirUQAvD_BwE
- [7] https://www.optimusdigital.ro/ro/ism-24-ghz/48-modul-tranceiver-nrf24l01-24-ghz.html?gclid=CjwKCAiAv9riBRANEiwA9Dqv1VfuSKQxhWquevYHnigL7nf4KzQQowzw01bWjnAofHwaOeJa9tKgVxoCc1EQAvD_BwE

Anexe

Anexa 1



Anexa 2



fritzing