**Minirobot cu detectie obstacole**

Lucrarea prezentata isi propune punerea in vedere a unor algoritmi de masurare,prelucrare si cuantificare a distantei relative fata de un obiect care este considerat obstacol. Pentru a evidentia practic utilitatea si importanta implementarii unor astfel de algoritmi in industria moderna, vom proiecta un minirobot care va avea in dotare trei module de baza. Aceste module vor fi denumite in continuare Modulul de Procesare, Modulul de Control si Modulul de Achizitie de date.

Modul in care minirobot-ul va actiona va depinde de datele pe care le va achizitiona modulul de achizitie, si anume o distanta intre minirobot si obstacol. In prima faza aceste date vor fi citite si introduse in Modulul de procesare, unde vom stabili exact ce rutina trebuie sa inceapa minirobot-ul nostru.

Scopul robotului este acela de a demonstra importanta integrarii unui algoritm de detectie de obstacole in industria moderna (in special cea automotive) , unde o astfel de functionalitate ar putea inbunatati considerabil siguranta pe drumurile publice. Desigur, acesta este un exemplu izolat pentru ca implementarea poate fi facuta si pe masini automate,roboti industriali etc. , unde un obstacol ar putea pune probleme considerabile desfasurarii activitatii in conditii de siguranta.

In cazul de fata, minirobot-ul proiectat cu scop demonstrativ va avea urmatoare rutina la intalnirea unui obstacol de dimensiuni medii:   
 - Robotul va avertiza intalnirea unui obstacol pe direactia inainte cu ajutorul a patru led-uri rosii (care se vor aprinde si stinge intermitent atata timp cat obstacolul este prezent).  
 - Va activa un Buzzer Activ care va semnaliza sonor intalnirea unui obstacol (va fi actionat intermitent , exact ca led-urile , atata timp cat obstacolul este prezent).   
 - In timp ce semnalele de avertizare sunt emise, robotul va opri orice fel de activitate (fie ea de deplasare,verificare etc) si va ramane pe loc atata timp cat obstacolul este prezent.

Derularea proiectului va avea urmatorul curs:

- Proiectarea digitala a hardware-ului necesar inglobarii celor trei module precizate anterior  
 - Implementarea hardware-ului proiectat anterior pentru teste si optimizari  
 - Implementarea software-ului necesar pentru comunicarea corecta intre cele trei module . Optimizarea modului de achizite de date pentru procesarea distantelor si a comportamentului robotului in cazurile necesare.  
 - Proiectarea unei platforme mecanice pentru robot care sa includa monturile aferente pentru modulele precizate anterior, urmand a fi imprimata 3D cu ajutorul tehnologiei Fused Filament Fabrication (FFF).

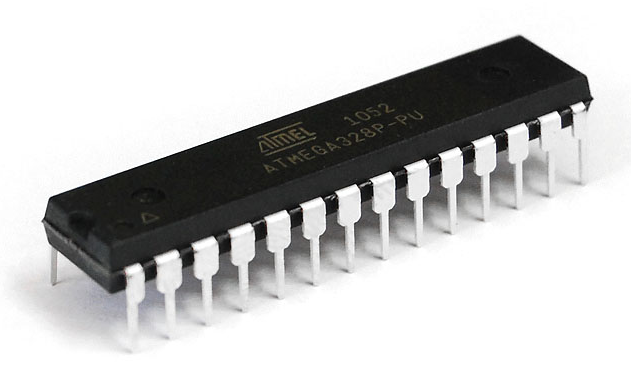
**CAPITOLUL 1**

**Hardware**

In scopul obtinerii unui concept functional,rapid si eficient au fost alese urmatoarele componente hardware:

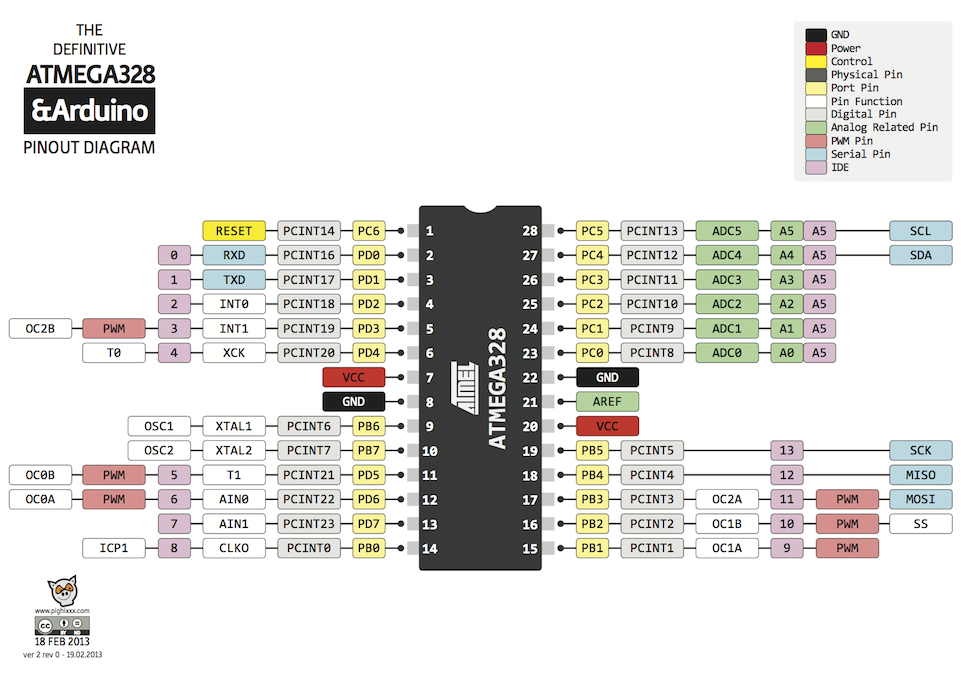
**Capitolul 1.1 - Microcontroller**

O platforma de dezvoltare foarte populara in ziua de azi este platforma Arduino, oferind o gama foarte variata de proiecte DIY,informatii si ghiduri cu ajutorul carora putem aduce un concept la realitate. In cazul proiectului prezentat anterior vom utiliza un Microcontroller ATmega328P prezent intr-o placa de dezvoltare Arduino Uno.



Microcontroller-ul ales ne ofera un pachet de 14 pini digitali I/O dintre care 6 pot fi utilizati in regim PWM, 6 pini analog I/O pe care ii putem utiliza pentru achizitia de date din exterior iar mai apoi sa livram rutina de executie conforma cu rezultatele procesarii. ATmega328P este un microcontroller pe 8-bit, avand o memorie Flash de 32KB

Un avantaj important in proiectul prezentat este utilizarea unei arhitecturi avansate RISC, ea oferind un set puternic de instructiuni (131 instr) , majoritatea putand fi executate intr-un singur ciclu de ceas ( foarte important pentru achizitia rapida de date si executarea rutinei alese in urma procesarii ).

 Sursa: [pighix](http://www.pighixxx.com/test/portfolio-items/atmega328/)

In materie de memorie, microcontroller-ul ATmega328P prezinta si o memorie EEPROM (1KB) ce poate fi folosita accesarii/modificarii unor variabile ce au scop de calibrare,optimizare. Acestea pot fi accesate cu usurinta printr-un adaptor USB-UART.

Avantajele utilizarii memoriei EEPROM este ca are un numar de 10 ori mai mare de scrieri/stergeri fata de memoria FLASH. Astfel putem creste fiabilitatea produsului proiectat.

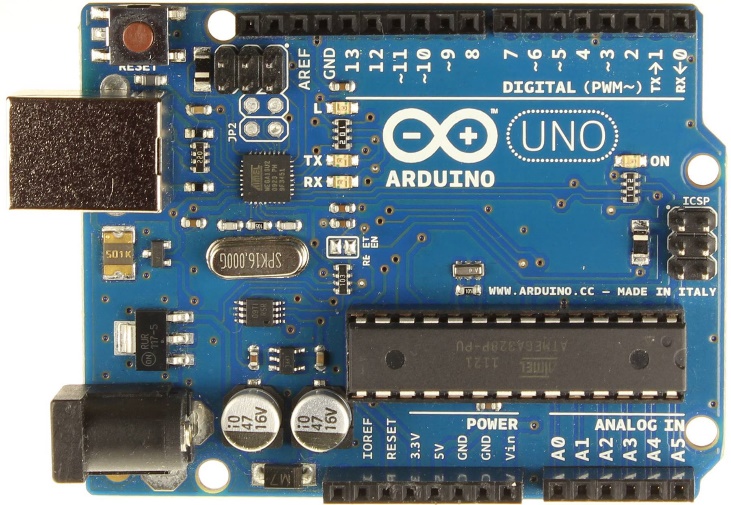
Pachetul in care se regaseste microprocesorul ATmega328P pe placa de dezvoltare Arduino Uno este 28-PIN-PDIP. Acesta lucreaza la tensiuni cuprinse intre 1.8V si 5.5V **[1]**

O concluzionare a avantajelor utilizarii microcontroller-ului ATmega328P si cateva din specificatiile acestuia se regasesc in tabelul urmator:

|  |  |
| --- | --- |
| Advanced RISC Architecture | - 131 Powerful Instructions  - 32 x 8 General Purpose Working Registers  - Fully Static Operation  - Up to 20 MIPS Throughput at 20MHz  - On-chip 2-cycle Multiplier |
| High Endurance Non-volatile Memory Segments | - 32KBytes of In-System Self-Programmable Flash program memory  - 1KBytes EEPROM  - 2KBytes Internal SRAM  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM |
| Peripheral Features | - Two 8-bit Timer  - One 16-bit Timer  - Real Time Counter with Separate Oscillator  - Six PWM Channels  - 8-channel 10-bit ADC  - 6-channel 10-bit ADC  - Programmable Serial USART  - Master/Slave SPI Serial Interface |
| Special Microcontroller Features | - Power-on Reset and Programmable Brown-out Detection  - Internal Calibrated Oscillator  - External and Internal Interrupt Sources |
| Operating Voltage | 1.8 - 5.5V |
| Temperature Range | -40°C to 85°C |
| Speed Grade | - 0 - 4MHz@1.8 - 5.5V,  - 0 - 10MHz@2.7 - 5.5.V,  - 0 - 20MHz @ 4.5 - 5.5V |
| Power Consumption at 1MHz, 1.8V, 25°C | - Active Mode: 0.2mA  - Power-down Mode: 0.1µA - Power-save Mode: 0.75µA |

Tabel 1.1.1 - **[2]**

**Capitolul 1.2 - Platforma de dezvoltare**



Sursa: [Arduino.cc](https://www.arduino.cc/en/main/arduinoBoardUno) & [Fritzing.com](http://fritzing.org/projects/by-tag/arduino%20uno/)

Placa de dezvoltare Arduino Uno contine toate componentele necesare unei utilizari usoare a microcontroller-ului ATmega328P. Aceasta dispune de o interfata USB-UART (RS2322,PL2322,CH340G) pentru a face posibila programarea microcontroller-ului de pe un PC sau laptop prin intermediul unei conexiuni USB.

Pe aceasta placa de dezvoltare, ATmega328P lucreaza la 16MHz datorita prezentei unui cristal cu cuartz extern, de 16MHz

Utilizarea unei placi de dezvoltare precum Arduino Uno ne permite o interfatare facila cu microcontroller-ul prin intermediul programului Arduino IDE, unde IDE reprezinta “Integrated Development Environment” .   
Acest program ne permite o vizualizare facila a codului,o monitorizare prin intermediul monitorului serial modul cum sunt achizitionate/procesate/cuantificate datele la iesirea senorilor pe care ii vom conecta.

Platforma de dezvoltare dispune si de un regulator de tensiune (AMS1117-5V) care ne permite alimentarea externa de la un convertor AC-DC sau de la o baterie, cu precizarea ca tensiunea de alimentare trebuie sa se incadreze in intervalul 7-12V. **[3]**

Acesta este un avantaj pentru conceptul nostru, avand in vedere ca trebuie sa fie un dispozitiv autonom care nu trebuie sa prezinte fire de alimentare externe, ci doar un modul capabil sa detecteze si sa anunte prezenta unui obstacol.

Programarea placii Arduino Uno se face cu ajutorul unui bootloader pre-programat, care permite incarcarea codului fara a utiliza o interfata de programare externa (USB-UART) . Comunicarea se face folosind protocolul STK500. Evitarea bootloader-ului se poate face folosind pinii ICSP (In-Circuit Serial Programming) cu ajutorul header-ului de pini prezenti pe Arduino Uno, insa pentru aplicatia descrisa anterior nu va fi nevoie sa folosim ICSP.

Un avantaj in utilizarea placilor de dezvoltare este siguranta acestora, spre exemplu pe Arduino Uno regasim o siguranta resetabila,care protejeaza portul USB al PCului sau laptopului conectat la el, intrerupand conexiunea daca un curent mai mare de 500mA este solicitat din portul USB si ramane intrerupta pana cand solicitarea dispare. **[4]**

O concluzionare a avantajelor utilizarii placii de dezvoltare Arduino Uno si cateva din specificatiile acesteia se regasesc in tabelul urmator:

|  |  |
| --- | --- |
| Microcontroller | ATmega328P |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7V |
| Input Voltage (limit) | 12V |
| Digital I/O Pins | 14 |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20mA |
| DC Current for 3.3V Pin | 50mA |
| Flash Memory | 32KB – 0.5KB used by bootloader |
| SRAM | 2KB |
| EEPROM | 1KB |
| Clock Speed | 16MHz |
| LED\_BUILTIN | 13 |
| Length | 68.6mm |
| Width | 53.4mm |
| Weight | 25g |

Tabel 1.2.1 - **[5]** Sursa: [Arduino.cc](https://www.arduino.cc/en/main/arduinoBoardUno)

In concluzie, placa de dezvoltare Arduino UNO va integra intr-un singur modul urmatoarele functii:

- Citirea de date de la iesirea senzorilor de distanta

- Interpretarea acestor date intr-o maniera cunoscuta procesorului (utilizand ADC la nevoie)

- Procesarea si selectarea rutinei potrivite pentru datele citite anterior

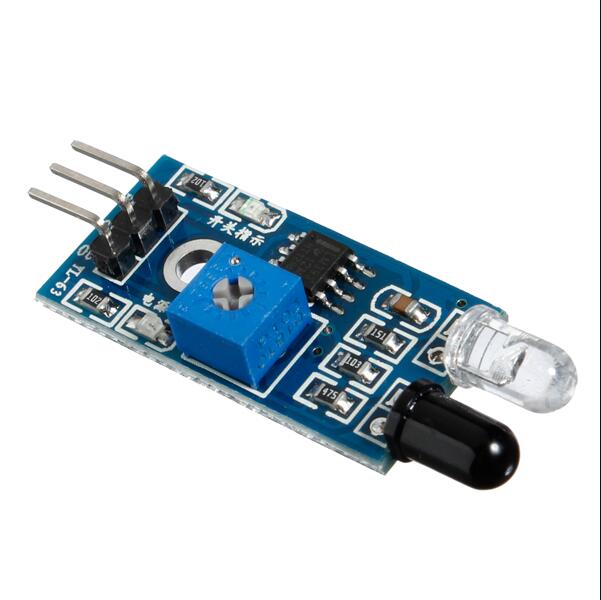
- Executia rutinei corespunzatoare

- Scrierea pe pinii de iesire ATmega328P a informatiei pentru motoare

- Avertizarea luminoasa si sonora (in functie de rutina aleasa)

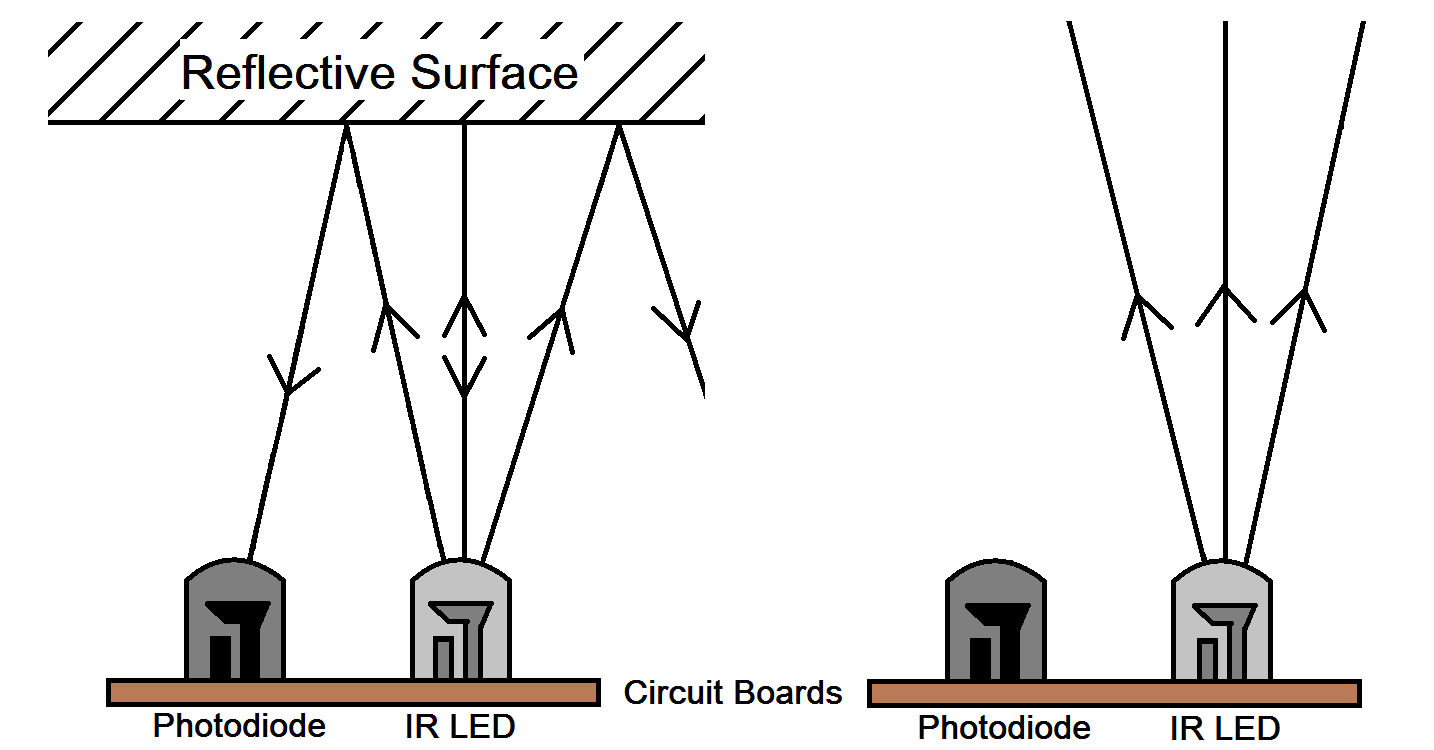
- Schimbarea rutinei in cazul in care obstacolul nu mai este prezent.

**Capitolul 1.3 - Senzori distanta**



In scopul detectarii obstacolelor vom avea in vedere utilizarea unor senzori de proximitate cu tehnologia IR (InfraRed). In imaginile anterioare avem doua modele des intalnite in industrie (mai ales modelul din dreapta, Sharp GP2Y0A21YK ). Evident, calitatea masuratorilor determina si pretul de fabricatie al senzorilor. Din aceste considerente vom folosi modelul mai ieftin, luand in considerare faptul ca acesta este un concept ce se poate scala in functie de necesitati.

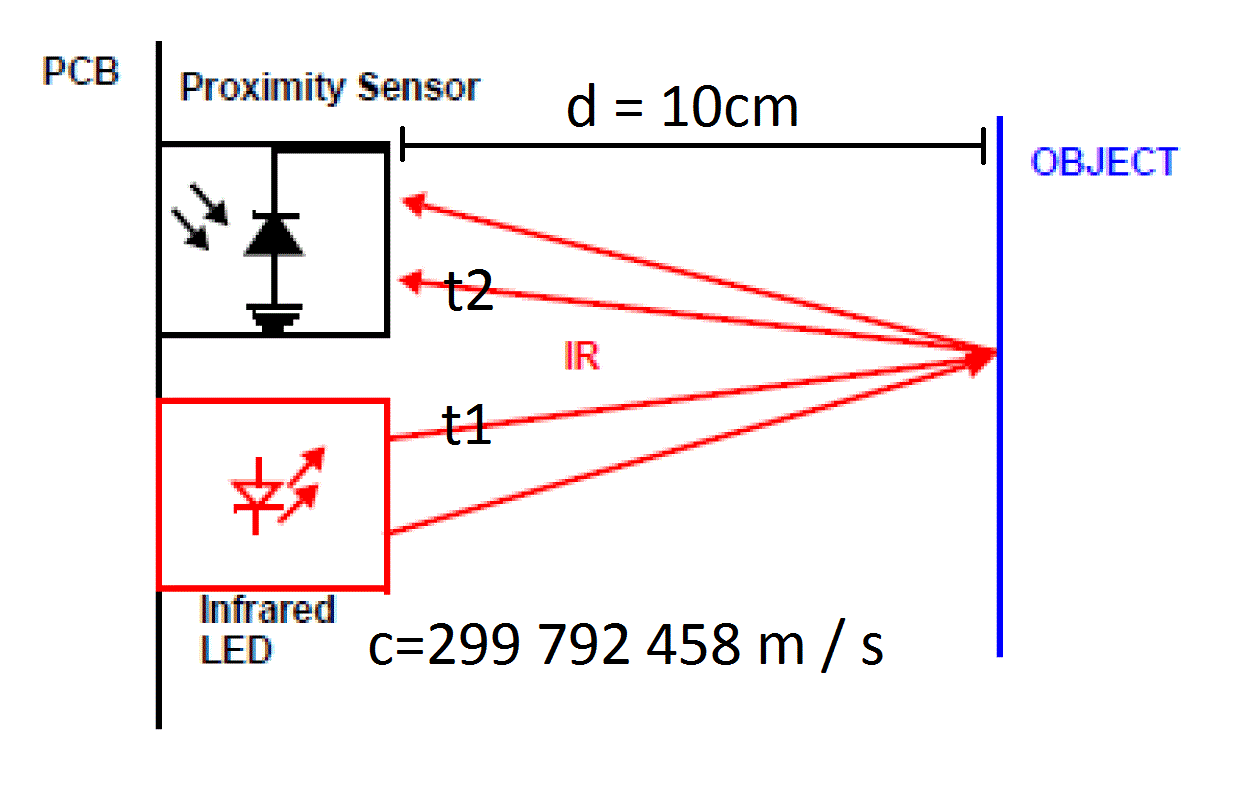
Utilizarea senzorilor de proximitate IR aduce un avantaj enorm la capitolul Timp de Masura, folosind fasciculele de unde infrarosii ca instrument de achizitie de date. Datele se prelucreaza la nivelul placii break-out,care include si o conversie Analog-Digitala a semnalului vazut la iesirea fototranzistor-ului.



Sursa: [Maxembedded.com](http://maxembedded.com/2013/08/how-to-build-an-ir-sensor/)

Pentru a efectua o achzitie de date (masuratoare de distanta) , led-ul IR transmite o serie de fascicule luminoase cu lungime de unda in intervalul 800-1400nm (inafara domeniului perceptibil al ochiului uman) si asteapta reflexia (partiala sau totala) a acestor fascicule, masurand si timpul in care acestea traverseaza mediul de propagare. Cunoscand viteza cu care fasciculul IR se deplaseaza in mediul de propagare si timpul in care acesta s-a intor si a fost citit de fototransistor, se determina distanta dintre sistemul de masura si un obiect oarecare,de dimensiuni perceptibile de catre senzorul anterior mentionat. **[6]**

Sa presupunem un obiect aflat la distanta d=10cm de senzorul ales. Senzorul nostru emite o serie de fascicule luminoase in domeniul IR cu viteza c = 299 792 458 m / s

  
 Sursa: [Wikipedia](https://en.wikipedia.org/wiki/Proximity_sensor)

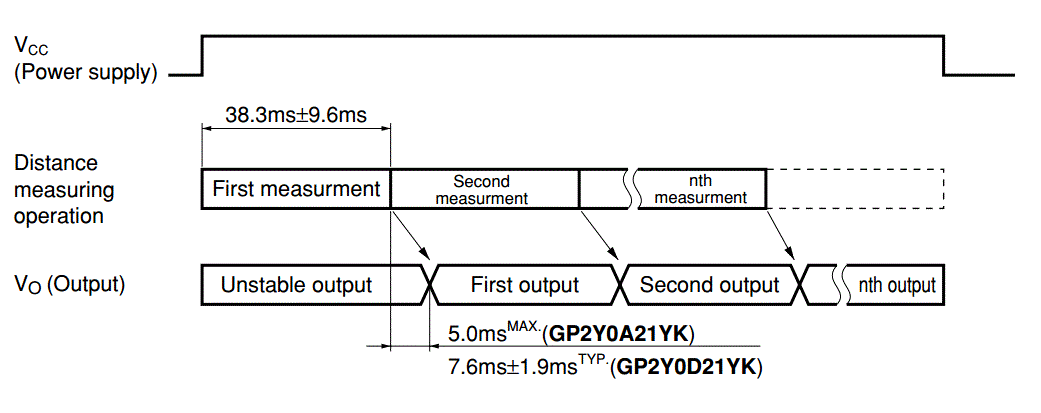
Pentru ca fasciculul luminos sa ajunga la fototransistor, acesta trebuie sa strabata mediul de propagare, sa loveasca obiectul, sa se reflecte in spre fototransistor, parcurgand din nou mediul de propagare. In acest caz, timpul de propagare pana la obiect este t1 iar timpul de propagare pana la fototransistor dupa reflexia fasciculului este t2. Modulul va masura acest timp de propagare ca fiind **t1+t2** si il va diviza cu 2, pentru a determina timpul in care fasciculul luminos ajunge la suprafata obiectului.

Cunoscand c,d putem spune : , unde

Astfel rezulta un timp de propagare

Deci

Putem observa ca o singura masuratoare dureaza aproximativ , ceea ce ofera un timp foarte bun pentru situatile de urgenta in care se poate gasi conceptul pe care propunem sa il aducem la realitate. Desigur, acest calcul este pur teoretic iar el poate fi folosit ca un factor de calibrare al senzorului nostru, folosindu-ne de acest timp de propagare ca o unitate etalon pentru o distanta de 10cm. **[7]**

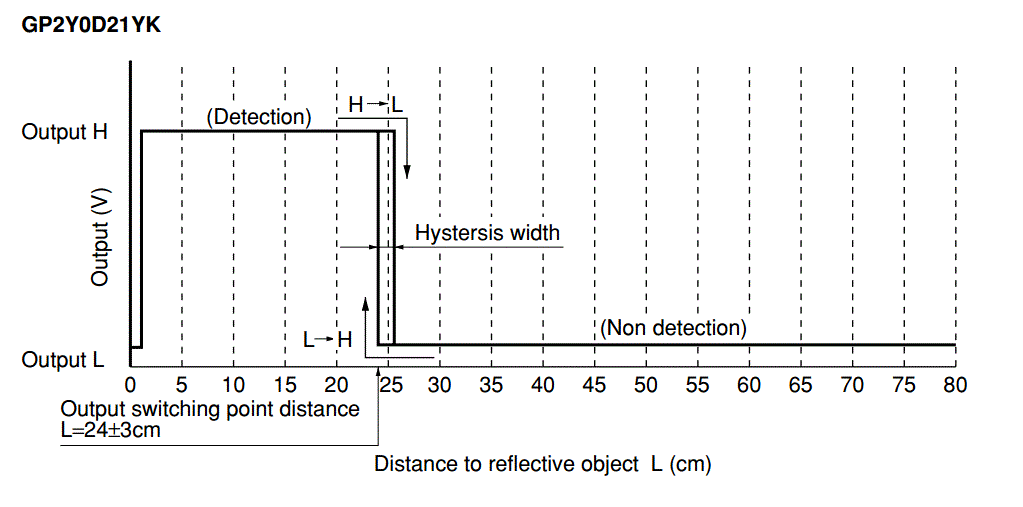
  
 Sursa: [Sharp Datasheet](file:///C:\Users\nitut\AppData\Roaming\Microsoft\Word\Datasheet\GP2Y0A21YK.pdf)

Mai mult decat atat, in foaia de catalog a senzorului Sharp GP2Y0A21YK , putem observa ca acesta raspunde cu o distanta in 5ms de la achizitia de date, ceea ce inseamna ca in diferenta de timp intre cele aproximativ 34ns/masuratoare si cele 5 ms dupa care se returneaza o valoare, modulul nostru face mai multe masuratori si le mediaza, pentru a reduce semnificativ erorile de masura ce pot aparea din cauza factorilor externi.

Considerand aceste aspecte, putem observa ca modulul executa un numar de

Avand in vedere limitarile de procesare , acest numar nu poate fi atins din considerente tehnologice, insa numarul de masuratori efectuat este suficient de mare in asa fel incat eroarea de masura sa fie in limita a 2,5% pentru intervalul 0-80cm . **[8]**

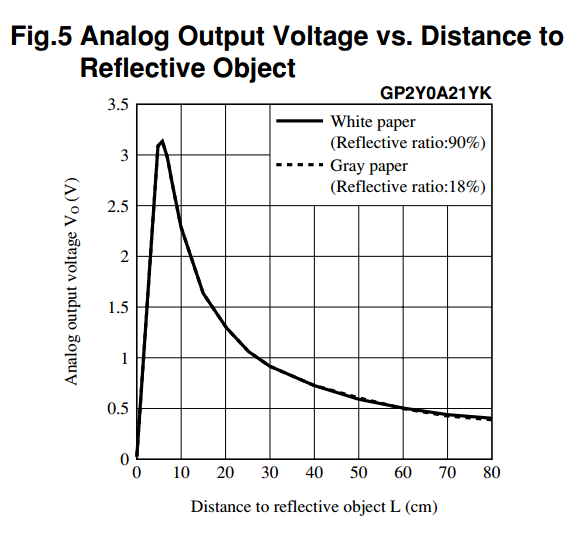
Mai jos putem observa caracteristica Output(Distance) a senzorului Sharp GP2Y0A21YK

  
 Sursa: [Sharp Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\GP2Y0A21YK.pdf)

Singurul impediment in utilizarea acestui senzor in plan teoretic ar fi existenta unui corp care indeplineste proprietatile (teoretice) de corp negru sau corp absolut negru. Acesta reprezinta in fizica un model teoretic pentru corpurile radiante de energie electromagnetica. Acest concept se defineste cu ajutorul unui corp care absoarbe integral orice forma de energie a radiatiei incidente, fara sa reflecte,refracte sau sa transmita mai departe aceasta energie.

In schimb, acesta poate emite radiatii iar spectrul electromagnetic al radiatiei emise depinde numai de temperatura sa absoluta. In natura acest corp nu poate fi regasit, deoarece reprezinta un concept teoretic pentru abstractizarea echilibrului energetic dintre materie si radiatie, distributia spectrala a radiatiei lui putand fi regasita in natura.

Avand in vedere faptul ca acest corp negru nu va fi regasit in sfera de aplicatii ale proiectului nostru, singurul impediment (real) ramas ar fi reflectibilitatea obiectelor ce pot fi considerate obstacole. Insa, urmarind foaia de catalog a senzorilor mentionati anterior ( Sharp GP2Y0A21YK ) , putem observa ca acestea percep fara probleme fascicule luminoase transmise catre un corp cu reflectibilitate mica (10-15%). **[9]**



Sursa: [Sharp Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\GP2Y0A21YK.pdf)

**Capitolul 1.4 - Motor DC**

****

Sursa: [optimusdigital.ro](http://www.optimusdigital.ro)  
  
 Pentru ca robotul nostru sa prinda viata,avem nevoie de un element de transmisie. In acest caz vom folosi doua motoare DC cu perii carora le-a fost atasat un reductor de viteza si o roata cu cauciuc. Motivul pentru care vom utiliza un reductor de viteza este ca aceste motoare nu ofera in general un cuplu foarte bun,insa ofera o gama variata de viteze disponibile in functie de tensiunea aplicata acestora.   
Astfel vom pierde din viteza totala a ansamblului mobil in scopul obtinerii unui cuplu mai bun si totodata a unui raspuns mai prompt la actionarea digitala a acestor motoare.

Aceste motoare sunt des intalnite in industria jucariilor RC, fiind cea mai ieftina varianta de motoare DC care poate servi unui grup mare de aplicatii precum Masini Teleghidate RC, Barci RC, Elicoptere RC etc. Un alt avantaj semnificativ in utilizarea acestui tip de motoare este eficienta si fiabilitatea lor. Avand un consum de numai 70mA in rotire fara sarcina, aceste motoare ofera o durata de viata mai lunga jucariilor ce sunt alimentate de la baterii sau acumualtori. **[10]**

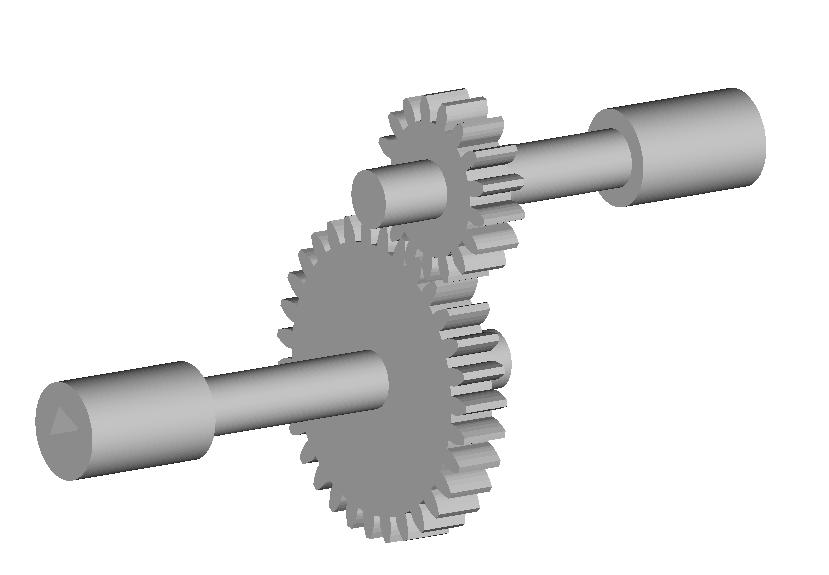
In proiectul nostru vom folosi aceste motoare ca sursa de tractiune. Acestea vor fi plasate in partea din spate a ansamblului , imitand intr-o oarecare masura un autoturism cu tractiune spate. Alegerea pozitionarii lor a fost facuta in scopul testarii ulterioarelor modificari ale robotului, intrucat acesta poate oferi posibilitatea ocolirii obstacolelor intalnite .  
Aceste functionalitati sunt la nivel de concept si nu se vor regasi in prima faza a proiectului final, ramanand pentru cercetari ulterioare.

Dimensiunea rotilor este de 65mm. Aceasta a fost aleasa tinand cont de micile denivelari de pot aparea pe o suprafata oarecare de testare, oferind robotului o gama mai larga de zone in care sa isi poata desfasura activitatea fara a fi impiedicat de eventualele imperfectiuni ale drumului,podelei,mesei etc. **[11]**



Sursa: [Pololu](https://www.pololu.com/product/1117)

Factorul de reductie pentru motorul ales este de 1:48, ceea ce inseamna ca pentru o rotatie a axului pe care este prinsa roata, motorul trebuie sa se roteasca de 48 ori.

Un bun exemplu de reductor se regaseste in imaginea alaturata, unde numarul dintilor rotitei mici este de 3 ori mai mic decat numarul dintilor rotitei mari, rezultand astfel o reductie de 1:3.

In cazul in care motorul ar fi conectat la rotita mica, acesta ar fi fost nevoit sa faca trei rotatii complete pentru ca rotita mare sa faca o rotatie completa.   
  
In schimb, daca el ar fi legat la rotita mare, acesta ar avea nevoie doar de o treime dintr-o rotatie pentru ca rotita mica sa efectueze o rotatie completa.   
Sursa: [brighthubengineering.com](http://www.brighthubengineering.com/machine-design/47267-what-is-a-reduction-gear/)

Acest concept poate fi folosit in diverse aplicatii,inclusiv cele cu roboti. Folosindu-ne de acest reductor, putem transforma toata energia pe care motorul o transfera in ansamblul de reductie in (rotinduse cu viteza mare) in cuplu. **[12]**

Ansamblul ales de noi (motor si reductor cu roata) ofera un cuplu de 0.8 kg \* cm, ceea ce este mai mult decat avem nevoie pentru proiectul nostru. In schimb, acest cuplu ne poate oferi informatii despre greutatea maxima pe care robotul o poate avea.

Un caz foarte fericit este acela in care motoarele au nevoie de foarte putin cuplu pentru a misca robotul din loc. In acest caz consumul de energie (curentul pe care motoarele il solicita pentru a muta robotul din loc) este mai mic, lucru pe care il cautam pentru a creste eficienta ansamblului si durata de viata a bateriei ce il va alimenta.   
 Folosind doua motoare de tractiune, acestea vor avea insumat 1.6 kg\*cm cuplu, deci telul nostru va fi sa pastram greutatea ansamblului sub 1 kg. In acest fel ne vom asigura ca eficienta celor doua motoare va ramane in pragul superior.

Din foaia de catalog a producatorului am putut extrage si valoarea variabilei RPM (Rotations per Minute) pentru diverse valori ale tensiunii de alimentare. Mai mult decat atat, am extras si un consum estimat pentru aceste valori ale tensiunii de alimentare. In acest fel vom putea estima un consum de energie in functie de modul in care robotul se va comporta pe terenul de test , rezultand ca mai apoi un timp mediu de acitivitate .

|  |  |  |
| --- | --- | --- |
| **Tensiune [V]** | **Viteza [rpm]** | **Curent [mA]** |
| 3 | 125 | 60 |
| 3.5 | 142 | 70 |
| 4 | 158 | 80 |
| 4.5 | 184 | 90 |
| 5 | 200 | 100 |
| 5.5 | 215 | 110 |
| 6 | 230 | 120 |

Tabelul 1.4.1 – **[13]** -Sursa: [optimusdigital.ro](file:///C:\Users\nitut\AppData\Roaming\Microsoft\Word\optimusdigital.ro)

Putem observa insa si gama de tensiuni in care motorul ales lucreaza, si anume intre 3V si 6V . Aceste tensiuni vor putea fi oferite de catre modulul de control ales de noi, deci motorul va fi utilizat pe toata plaja de tensiuni specificata de producator.

Tot din foaia de catalog am extras temperaturile de lucru ale motoarelor alese, si anume intre -10°C ~ +60°C . Tinand cont de mediul in care se va regasi robotul nostru, aceste temperaturi sunt mai mult decat permisive.  
Un singur lucru trebuie luat in calcul si anume acela ca orice cantitate de energie disipata sub efect Joule din apropierea motoarelor le poate afecta atat pe ele, cat si reductorul (care este construit din plastic).

In concluzie temperatura motorului si a mediului inconjurator lui trebuie sa fie constanta si de preferat in imediata apropiere a temperaturii mediului ambiant.   
Pentru a avea sub control acest aspect vom asigura un flux de aer constant in carcasa robotului folosind un ventilator (Fan,Cooler) conectat la un pin 5V al placii de dezvoltare Arduino Uno. [14]

**Capitolul 1.5 - Adafruit Motor Shield**

****

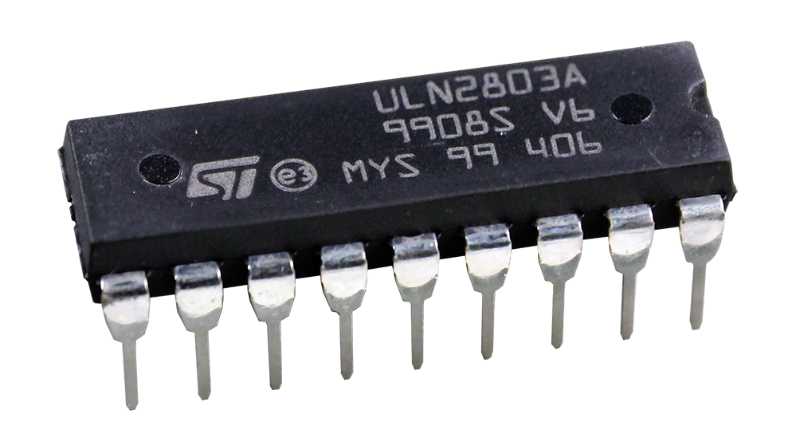
Sursa: [Adafruit](https://www.adafruit.com/product/81)

Pentru a facilita conectarea si utilizarea celor doua motoare de transmisie vom avea nevoie de un shield de control pentru motoare. In acest caz, Adafruit a pus la punct un shield pentru placa de dezvoltare Arduino Uno , care se conecteaza dupa modelul “backpack” , adica pinii acestui modul se introduc in conectorii originali ai Arduino Uno.

Am ales acest modul pentru simplitatea efectuarii conexiunilor, dispunand de 2 conectoare bloc cu suruburi. Aceste conectoare ofera o siguranta sporita conexiunilor in detrimentul celor pe baza mufelor Dupont (mufele des intalnite in calculatoare,senzori,ventilatoare etc) de 2.54mm. Mai mult decat atat, face ca asamblarea/dezasamblarea robotului sa fie mai simpla si mai rapida,fara sa existe posibilitatea deteriorarii contactelor dintre motor si modul de control.

Modulul Adafruit Motor Shield contine o sumedenie de functionalitati de control pentru diverse motoare electrice. Printre acestea, se enumera:

**- 2 conectori pentru motoare Servo Hobby (5V)** – conectori cu traseu catre timer-ul de rezolutie mare al placii de dezvoltare Arduino Uno ( rezulta miscare foarte fina,fara intreruperi )   
 - **4 conectori pentru motoare DC (bi-directionale)** – cu selectie de viteza pe 8 biti, rezultand o rezolutie de 0.5% pentru controlul DC.  
 - **2 conectori pentru motoare Stepper (unipolare si bipolare)** – care suporta motoare cu o bobina, doua bobine sau cu microstepping controlat digital

Pentru a putea controla aceste tipuri foarte diferite de motoare, modulul Adafruit Motor Shield are 4 Punti H (H-Bridge) de tipul L293D care ofera pana la 600mA pe fiecare punte, incluzand un curent maxim de varf de 1.2A pentru perioade scurte de timp. **[14]**  
Mai mult decat atat, chip-ul L293D include si un sistem de thermal-shutdown care protejeaza atat driver-ul cat si modulul de control de efectele supra-incalzirii componentelor. 

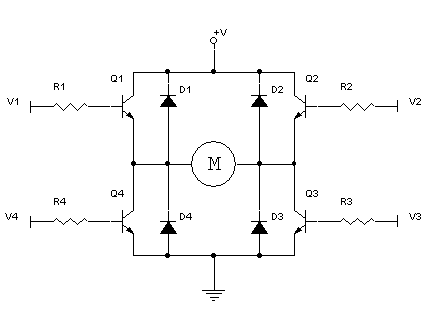
Inca o functionalitate utila a modulului este aceea ca toate motoarele conectate vor fi nealimentate pe perioada de pornire a modului, gratie rezistentelor de pull-down.

Alimentarea modulului este unul din lucrurile cele mai importante pentru

modelul selectat, si anume faptul ca suporta o tensiune de intrare intre 4.5V si 24V . Acest lucru ne ofera posibilitatea utilizarii atat a motoarelor de 5V (cele de hobby) cat si a motoarelor de 12V sau 24V, care evident ofera mai mult cuplu si mai multa putere. In acest sens conceptul se poate scala cu usurinta si la un nivel inalt, fiind usor de integrat in diverse aparate cu control automat.

Motivul pentru care am ales o punte H pentru controlul motoarelor de tractiune este acela ca aceasta punte ne ofera posibilitatea de a controla motorul in ambele sensuri folosind doar 2 fire de conexiune, si anume cele care sunt legate la bornele motorului.

Conceptul de H-Bridge se poate explica cu usurinta folosind 4 tranzistori conectati ca in imaginea de mai jos:



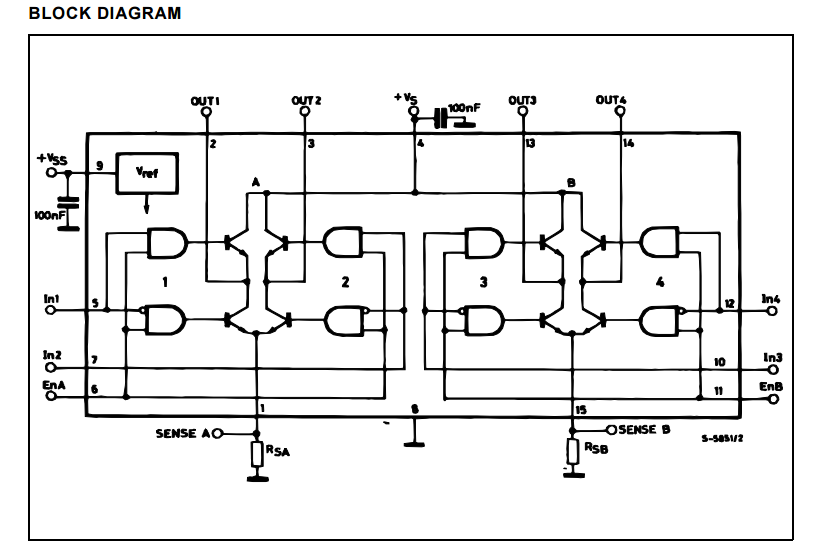
Sursa: [Modularcircuits](http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/)

Dupa cum se poate observa, daca dorim sa rotim motorul in sensul acelor de ceasornic va trebui sa activam tranzistorii Q1 si Q3, trecand cei doi pini V1 si V3 in modul High (acest lucru se face cu usurinta folosind placa de dezvoltare Arduino Uno) . In acest moment tranzistorii Q1 si Q3 sunt activi ,iar curentul circula de la V+ la GND prin Q1, M si Q3, actionand motorul in directia dorita.

De asemenea, daca dorim sa actionam motorul in sens invers acelor de ceasornic, vom avea nevoie sa actional tranzistoarele Q2 si Q4. Pentru ca acestea sa se deschida, trebuie sa trecem pinii V2 si V4 in HIGH, folosind pinii digitali ai placii de dezvoltare Arduino Uno.

Se poate observa ca aceste tranzistoare au rol de switch (intrerupator) si singurul lucru de care trebuie sa tinem cont este ca tensiunea Baza-Emitor sa fie destul de mare (aprox. >0.7V) pentru ca tranzistorul sa permita trecerea curentului dinspre Colector catre Emitor. **[15]**

Aceasta functionalitate se regaseste si in interiorul puntii H L293D, insa aceasta este redusa la nivel de porti logice pentru a accelera procesul de schimbare de directie sau de actionare intr-o anumita directie.



Sursa: [L293d Datasheet](file:///C:\Users\nitut\AppData\Roaming\Microsoft\Word\Datasheet\L298_H_Bridge.pdf)

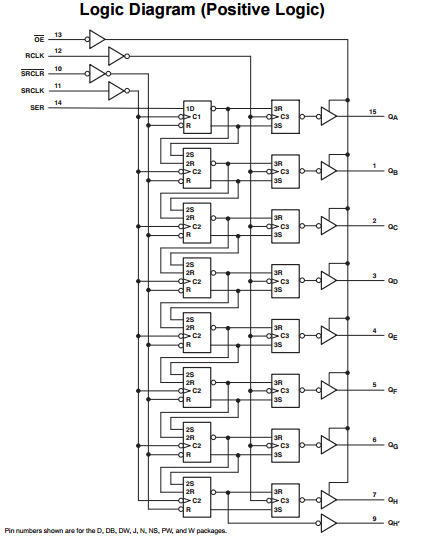
In imagine este reprezentat acelasi principiu, insa se regasesc 2 punti H pentru control. Asta faciliteaza conectarea a 2 motoare in acelasi timp, ce pot fi controlate fie simultan,identic fie separat. In cazul nostru vom avea doua motoare ce vor fi conectate simultan,identic, deoarece robotul nu va avea posibilitatea de a lua viraje.

In privinta consumului de energie, singurul impediment ar fi acela ca si in lipsa sarcinii, modulul Adafruit Motor Shield poate consuma pana la 30mA, ceea ce poate sa scada durata de viata a bateriei care alimenteaza robotul. **[16]**

In concluzie modulul ales ofera un numar mare de functionalitati intr-un pachet foarte mic, facand utilizarea lui foarte avantajoasa pentru un robot de dimensiuni mici. Modul de atasare “backpack” pentru placa de dezvoltare Arduino Uno elimina dificultatea intelegerii ansamblului de catre un utilizator fara experienta si cel mai important factor este acela ca elimina posibilitatea de aparitie a erorilor de conectare.   
Aceste eror pot duce des la distrugerea modulelor,regulatoarelor de tensiune sau a circuitelor integrate cu rol important in proiectul nostru. Singurele conexiuni externe modulului vor fi acelea pentru partea de avertizare precum led-uri, buzzer-ul activ etc.

Modulul Adafruit Motor Shield include si un registru de shiftare pe 8 biti pentru tranzitia “8-Bit Serial-In , Parallel-Out” . Acesta se regaseste in chip-ul 74HC595N produs de Texas Instruments.



 Acest chip este folosit pentru a putea controla simultan mai multe iesiri catre motoare, scriind prin intermediul “Serial-In” 8 valori ce urmeaza a fi transmise in paralel, catre 8 destinatii diferite. Acest lucru permite atasarea motoarelor Stepper si DC si controlarea lor folosind mai putini pini ai Arduino Uno.

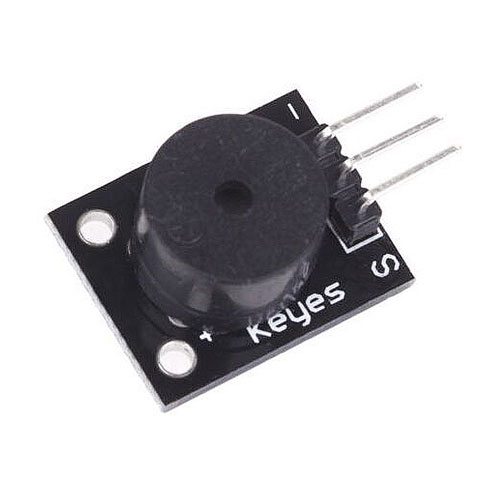
In schema alaturata se poate observa modul in care sunt conectate iesirile si intrarile chip-ului 595.

Pinii regasiti in schema reprezinta urmatoarele:

SER – Serial Input  
 SRCLK – Serial Clock Input  
 SRCLR – Serial Clear Input  
 OE – Output Enable  
 VCC – Alimentare  
 GND – Ground  
 QA – QH – Output Pins

Sursa: [76HC595 Datasheet](file:///C:\Users\nitut\AppData\Roaming\Microsoft\Word\Datasheet\sn74hc595.pdf) **[17]**

**Capitolul 1.6 - Buzzer Activ**

  
 Sursa: [ArduinoModules](http://arduinomodules.info/ky-012-active-buzzer-module/)

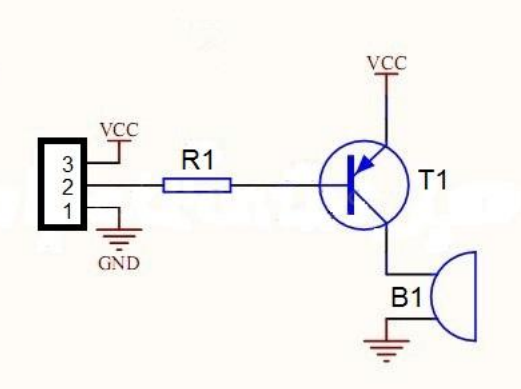
Cunoscut si sub numele de Buzzer Electromagnetic, acesta este des intalnit in sistemele de alarma, sistemele de avertizare, module auto etc. Buzzer-ele pot fi de mai multe tipuri mecanice, electromagnetice sau piezoelectrice. Noi vom alege pentru acest proiect versiunea electromagnetica, fiind una din cele mai simple variante din punct de vedere al controlului .

Varianta pe care am decis sa o folosim in proiectul actual include si un circuit imprimat si cateva elemente ce vor usura utilizarea si controlul modulului. In acest senz placa pe care este lipit buzzer-ul electromagnetic include:

- buzzer  
 - 8550 Triode   
 - base resistor

In acest fel, putem controla fara problemele modulul utilizand orice pin de intrare/iesire al microcontroller-ului sau a placii de dezvoltare Arduino Uno, pentru ca modulul foloseste un semnal High pentru a activa un tranzistor (in regim de dioda) ce alimenteaza buzzer-ul. **[18]**

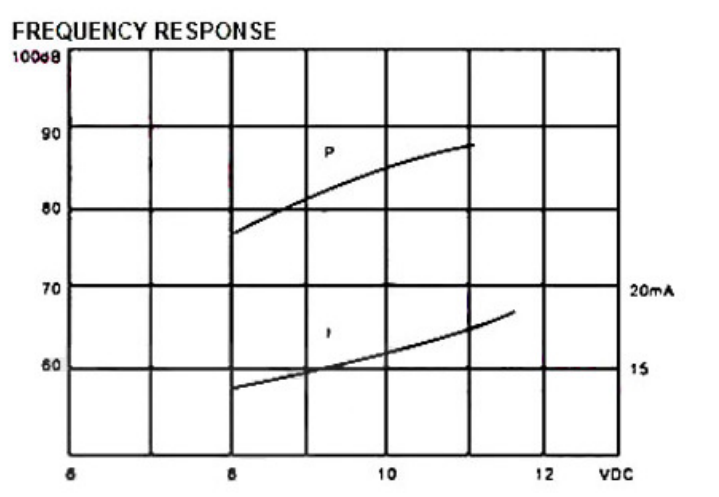
Circuitul de conectare este urmatorul



Sursa: [Openimpulse](https://www.openimpulse.com/blog/products-page/product-category/active-buzzer-module-and-cable/)

Modulul Buzzer ales de noi lucreaza la o valoare a tensiunii nominale intre 4-8V. La aceasta tensiune consumul maxim specificat in foaia de catalog este de 30mA, un consum destul de mare pentru aplicatia pe care o dorim, insa ne vom baza pe faptul ca nu va fi alimentat constant deci consumul va fi doar pe perioade scurte de timp.

Motivul pentru care am ales aceasta metoda de avertizare este ca acest buzzer este cel mai zgomotos modul electronic ce se poate controla cu usurinta,fara amplificatoare audio, ajungand chiar la 85dB (Minim, la o distanta de 10cm fata de buzzer) . Astfel, momentul in care robotul detecteaza un obstacol si actioneaza rutina de avertizare, acest buzzer poate fi auzit cu usurinta si de la distante mai mari. **[19]**

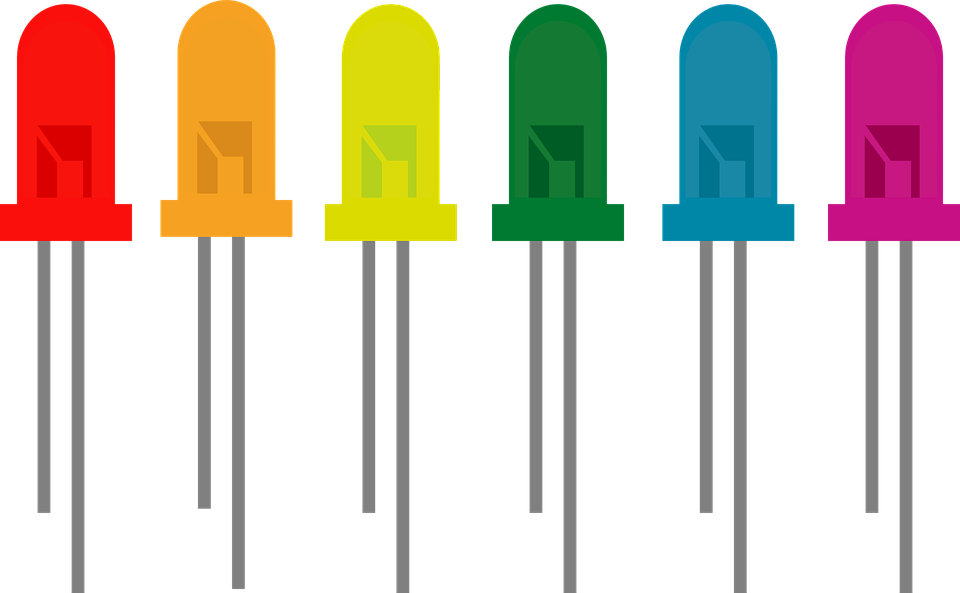


Sursa: [Active Buzzer Datasheet](file:///C:\Users\nitut\AppData\Roaming\Microsoft\Word\Datasheet\ef532_ps.pdf)

Dupa cum specifica si foaia de catalog, frecventa de rezonanta a buzzer-ului este in intervalul 2000-2600 Hz , cea mai probabila valoare fiind 2300Hz

Astfel putem deduce ca buzzer-ul se poate controla atat digital , in cazul in care el este activ cat si analogic (sau PWM) , in cazul in care el este pasiv, aplicand-ui un semnal PWM cu frecventa apropiata de frecventa de rezonanta. Indepartarea de acest punct denumit frecventa de rezonanta va determina o scadere in intensitate (volum) a sunetului pe care buzzer-ul il emite. **[20]**

**Capitolul 1.7 – Dioda LED**

****

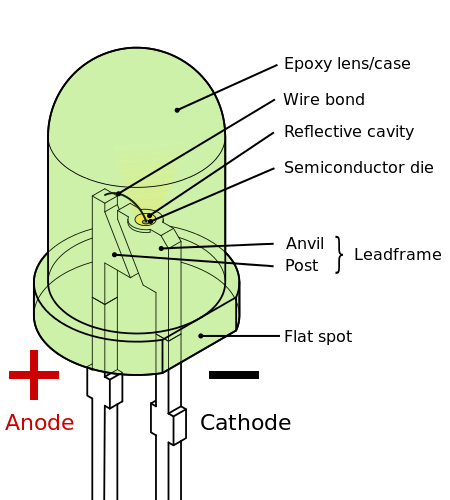
Cea mai eficienta metoda de avertizare vizuala este prin intermediul unei diode LED. In cazul nostru vom folosi 4 diode led de culoare rosie pentru a semnaliza prezenta unui obstacol in calea robotului nostru. Aceste diode LED vor lumina intermitent atata timp cat obstacolul va fi prezent in fata robotului.

Acronimul LED provine din engleza, din denumirea “Light Emitting Diode” care inseamna dioda emitatoare de lumina. Avand in vedere ca un led este practic o dioda, numele acesteia se justifica cu usurinta din functionalitatea componentei. Modul de functionare al diodei LED este urmatorul: La aplicarea unei diferente de potential mai mare decat tensiunea minima de activare, electronii din interiorul capsulei acrilice a led-ului se recombina cu golurile. In urma acestui proces energie sub forma de fotoni este emisa. Aceasta emisie de energie sub forma de fotoni se numeste electroluminescenta.

Aceste diode LED pot avea diferite culori, pornind de la culorile de baza Rosu, Albastru si Verde pana la culori mai complexe precum Portocaliu,Mov,Galben etc. Culoarea ledului depinde de cantitatea de energie emanata sub forma de fotoni si de banda energetica a semiconductorului utilizat in fabricarea diodei. Aceste benzi pot fi atat in domeniul vizibil, cat si in domeniul infrarosu sau ultraviolet.

Diodele LED au aparut pe piata electronicii in 1962, primele diode fiind cele infrarosu, avand intensitate luminoasa foarte slaba. Acestea sunt des intalnite chiar si in zilele noastre in majoritatea telecomenzilor sau instrumentelor de control la distanta. Diodele LED moderne sunt mult mai puternice, oferind intensitate luminoasa mare cu consum de energie redus. **[21]**

Cele mai semnificative progrese in domeniul electronicii la capitolul Diode LED a aparut in urma utilizarii LED-urilor pentru indicatoare, inlocuind becurile incandescente de dimensiuni reduse cu un led compact,fiabil si cu consum redus de energie la o tensiune usor de livrat. Mai mult decat atat, aparitia lor in afisaje cum ar fi afisajul de tip “7 Segmente” a condus la aparitia ceasurilor digitale, afisajelor compacte etc. Practic, se poate spune ca dioda LED a fost una dintre cele mai importante descoperiri din lumea electronicii moderne.



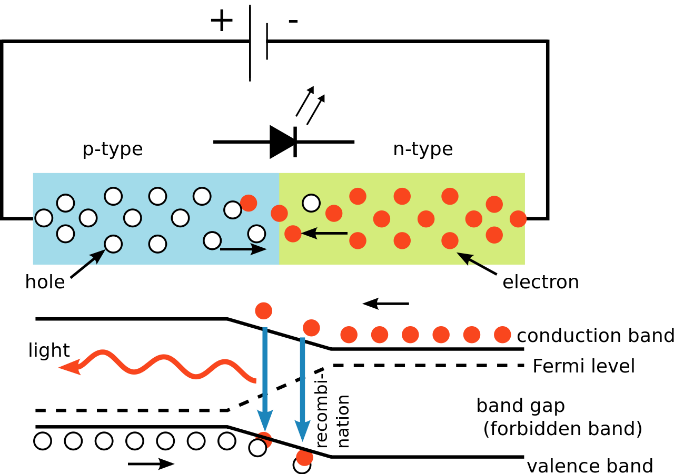
Dupa cum se poate vedea in imaginea alaturata, dioda LED este incastrata intr-o lentila din rasina epoxidica. Aceasta rasina protejeaza ledul dar ofera si o focalizare a fasciculului luminos emis de semiconductorul plasat pe suportul din metal.

Acesta dispune si de o cavitate reflectiva care are rolul de a focaliza fasciculele catre lentila din rasina epoxidica.  
  
 Tot pe carcasa lui, led-ul prezinta si o latura plata care serveste de obicei ca referinta pentru catodul diodei LED. Astfel, majoritatea utilizatorilor recunosc polaritatea diodei cu ajutorul acestei suprafete plate. **[22]**

Cele doua fire acre ies din carcasa din rasina epoxidica sunt coductori metalici. Pin-ul marcat ca pozitiv este Anod-ul, iar cel marcat negativ este Catod-ul. Intre acesti doi conductori se aplica o tensiune constanta , a carei valoare depinde de culoarea,banda de lungime de unda si caracteristicele interne ale semiconductorului.

De regula, o dioda LED obisnuita are tensiuni de alimentare cuprinse intre 2.5 si 3.6V. Aceste tensiuni sunt semnificativ mai mari decat tensiunile de polarizare ale unei diode normale, deoarece energia necesara pentru a emite fascicule luminoase este semnificativ mai mare .

Chip-ul care reprezinta inima diodei LED este un semiconductor dopat cu impuritati pentru a crea o jonctiune P-N asemeni unei diode obisnuite. In aceasta situatie , curentul va circula dinspre jonctiunea P (Anod) catre jonctiunea N (Catod) si nu in directie inversa. In urma acestui proces electronii traverseaza jonctiunea si la intalnirea cu un gol, acesta cade intr-o stare energetica de nivel mic si emana (radiaza) energie sub forma de fotoni.

In imaginea alaturata se poate observa rutina de tranzitie a electronilor catre goluri si modul cum acestia emit (radiaza) energie sub forma de fotoni.

Sursa: [Wikipedia-LED](https://en.wikipedia.org/wiki/Light-emitting_diode)

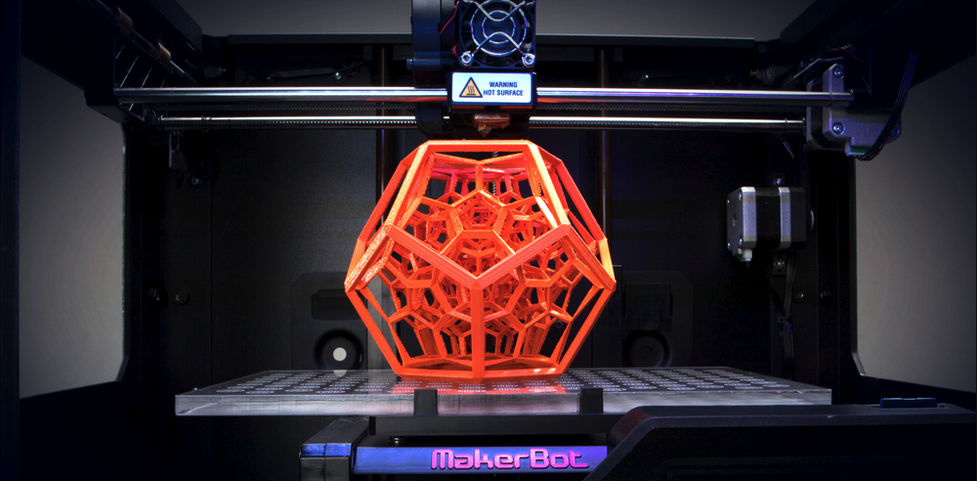
In functie de marimea benzii de energie a semiconductorului ales, dioda LED poate emite diferite culori. Aceste culori evident sunt catalogate dupa eficienta, deoarece obtinerea mai multor culori din spectrul vizibil si atingand o anumita lungime de unda echivalenta cu culoarea aleasa, avem nevoie de cantitati de energie diferite. In acest caz, anumite diode au nevoie de o putere livrata mai mica sau mai mare, in functie de eficienta lungimii de unda a diodei led. Aceste caracteristici se pot regasi in tabelul urmator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Culoare** | **Lungime de unda (nm)** | **Coeficient de eficienta** | **Eficienta (**[**lm**](https://en.wikipedia.org/wiki/Lumen_(unit))**/**[**W**](https://en.wikipedia.org/wiki/Watt)**)** |
|  | [Rosu](https://en.wikipedia.org/wiki/Red) | 620 < *λ* < 645 | 0.39 | 72 |
|  | [Portocaliu](https://en.wikipedia.org/wiki/Red-orange) | 610 < *λ* < 620 | 0.29 | 98 |
|  | [Verde](https://en.wikipedia.org/wiki/Green) | 520 < *λ* < 550 | 0.15 | 93 |
|  | [Albastru](https://en.wikipedia.org/wiki/Cyan) Deschis | 490 < *λ* < 520 | 0.26 | 75 |
|  | [Albastru](https://en.wikipedia.org/wiki/Blue) | 460 < *λ* < 490 | 0.35 | 37 |

Sursa: [radio-electronics.com](http://www.radio-electronics.com/info/data/semicond/leds-light-emitting-diodes/characteristics.php)

Se poate observa cu usurinta faptul ca diodele LED cu lungime de unda intre 610 si 620 nm au o eficienta foarte mare. Avand in vedere ca portocaliul este o culoare cu intensitate luminoasa ridicata, si eficienta unei diode LED de culoare portocalie va avea o eficienta mare. In general aceste culori sunt reprezentate in temperaturi in grade Kelvin, culoarea care ofera eficienta maxima fiind in jurul temperaturii de 6500K. **[23]**

**Capitolul 2 – Proiectare 3D – Sasiu MiniRobot detectie obstacole**

  
 Sursa: [Makerbot.com](http://www.makerbot.com)

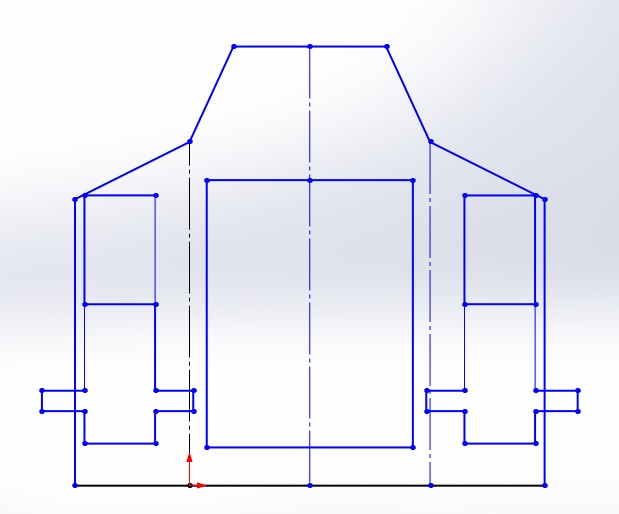
In vederea realizarii practice a robotului, am decis ca instrumentul potrivit pentru fabricarea unei platforme pentru robotul prezentat ar fi o imprimanta 3D. Am ales acest mod de fabricare (FFF) – Fused Filament Fabrication – deoarece este cel mai accesibil pe piata imprimarii 3D si ofera o calitate suficient de buna pentru cerintele de precizie pe care le are proiectul prezentat.

Pentru a avea o imagine de ansamblu asupra robotului , am efectuat masuratorile componentelor alese in lista de materiale (BOM) precizata anterior. Astfel am transpus in programul de proiectare CAD aceste dimensiuni si am creat ansamble cu aceste referinte de dimensiuni.

Programul folosit in vederea proiectarii sasiului robotului este Solidworks 2016. Acesta ofera o sumedenie de functii ajutatoare pentru proiectarea 3D, unele din acestea find “Extrude Boss Base” , “Extrude Cut” etc. Aceste functii iau ca “argument” niste schite denumite in programul respectiv “Sketch” care reprezinta dimensiunea in milimetrii a reperului ce urmeaza a fi creat.

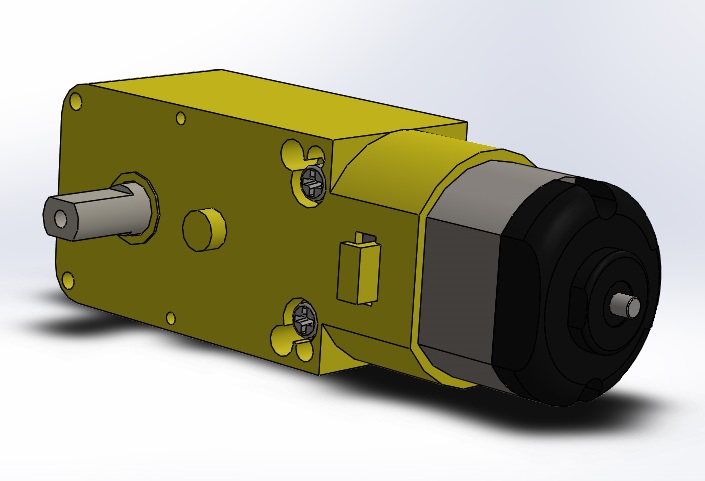
Mai mult decat atat, alegerea programului a fost facuta si in urma unui studiu cu privire la informatile disponibile pe internet. Astfel , Solidworks 2016 s-a dovedid a fi unul dintre cele mai bine documentate programe de proiectare 3D, oferind posibilitatea simularii tuturor imbinarilor robotului pe care trebuie sa il construim. Disponibilitatea documentatiei si a mai multor ghiduri de utilizare au facut utilizarea acestui program mult mai usoara, avand practic posibilitatea de a urmari pas cu pas schitarea si asamblarea partilor componente (denumite in Solidworks 2016 – Part ) .   
 Imprimantele 3D primesc numai fisiere in format litografic, cum ar fi fisiere de tip STL (STereoLithography) , OBJ (Object) . Aceste fisiere contin informatii atat despre natura axelor X,Y a obiectului, cat si despre axa Z.

**Capitolul 2.1 – Proiectare 3D – Solidworks 2016**

 In imaginea alaturata se poate observa o schitare 2D a proiectului robotului nostru. In partile laterale se pot observa cele doua motoare de tractiune, bazate pe motoarele DC130.  
In centrul ansamblului se poate vedea un dreptunghi ce reprezinta placa de dezvoltare Arduino Uno R3. Aceasta va fi prinsa pe platforma cu ajutorul unor suruburi de fixare si a piulitelor de tip Metric 3 ( M3 ).

In vederea montarii acestei placi de dezvoltare, vor fi adaugate 4 gauri la pozitionari specifice placii, pentru a acomoda cele 4 suruburi M3 precizate anterior.

Avand in vedere ca motoarele 130DC au o montura la un unghi de 90 de grade, acestea vor avea nevoie de un suport sub forma de perete pentru montaj.   
Precum se observa si in imagine, cele doua axe ale motoarelor ies din aria platformei, pentru a acomoda doua roti de tractiune si anvelope de cauciuc. Acestea trebuie sa se regaseasca la o distanta de minim 3 milimetrii fata de platforma pentru a reduce la zero riscul de atingere a platformei in timpul functionarii. Astfel, am luat in considerare aceste distante pentru o montura optima a motorului pe platforma ce urmeaza a fi imprimata.

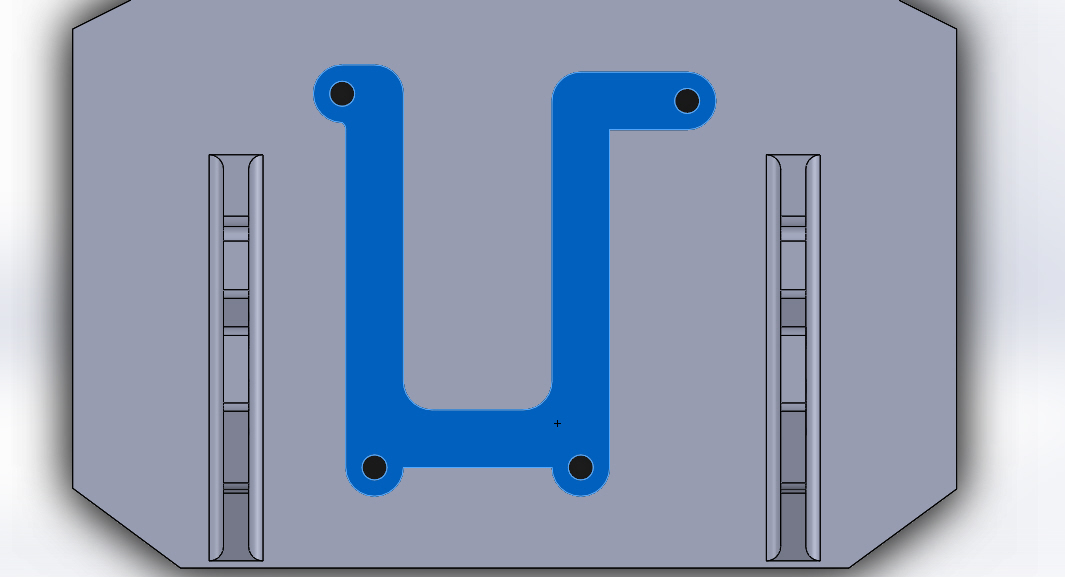


Datorita comunitatii foarte numerice in domeniul imprimarii si modelarii 3D, am avut acces la modele 3D pentru piesele pe care le vom folosi in procesul de asamblare. Spre exemplu, motorul DC130 ce se regaseste in imaginea alaturata. Acest model a oferit atat dimensiuni precise pentru piesa in sine (distanta intre gaurile de montaj, pin-ul de centrare,dimensiunile exterioare ale ansamblului etc) cat si o idee mai mult decat teoretica asupra aspectului final al robotului. In acest sens putem lucra si la partea de design fara sa pierdem din Sursa: [GrabCad.com](http://www.grabcad.com)  
vedere functionalitatea adevarata a produsului   
ce urmeaza a fi realizat. Consider un avantaj enorm acest fapt, tinand cont ca in majoritatea cazurilor imaginea exterioara este cel mai atractiv punct al unui produs, iar daca aceasta nu da dovada de excelenta, fiabilitatea sau punctele cheie ale produsului nu ajung a fi vizualizate.

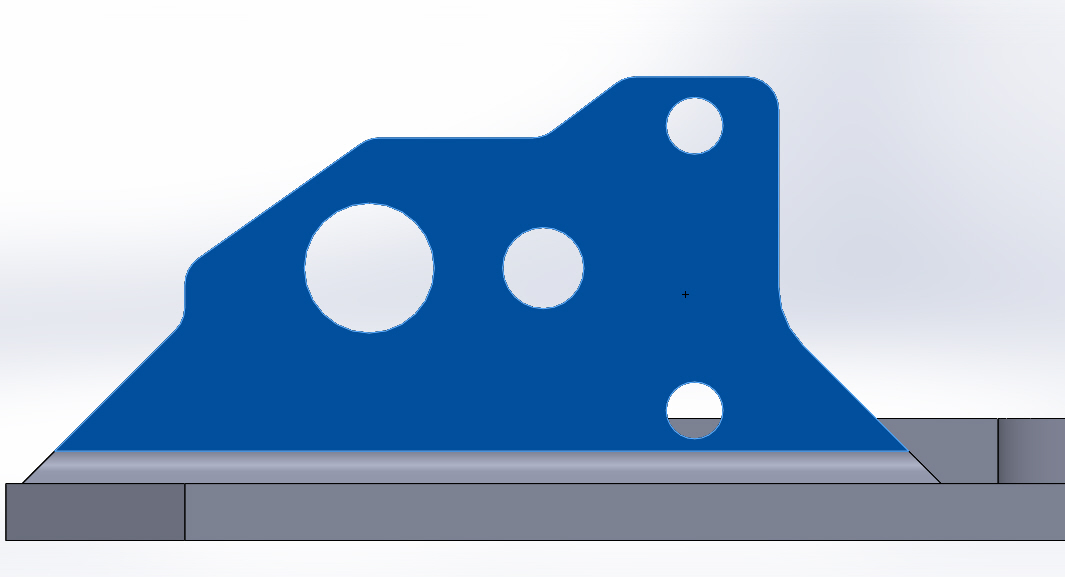
Avand aceste dimensiuni, am folosit functia “Extrude Boss Base” din Solidworks 2016 si am creat platforma pe care se vor monta piesele . Tot aici am folosit functia “Extrude Cut” pentru a crea gaurile necesare montarii placii de dezvoltare Arduino Uno R3. Dimensiunile gaurilor placii de dezvoltare au fost extrase tot dintr-un model 3D oferit de comunitatea open-source a imprimarii 3D.

Sursa: [GrabCad.com](http://www.grabcad.com)

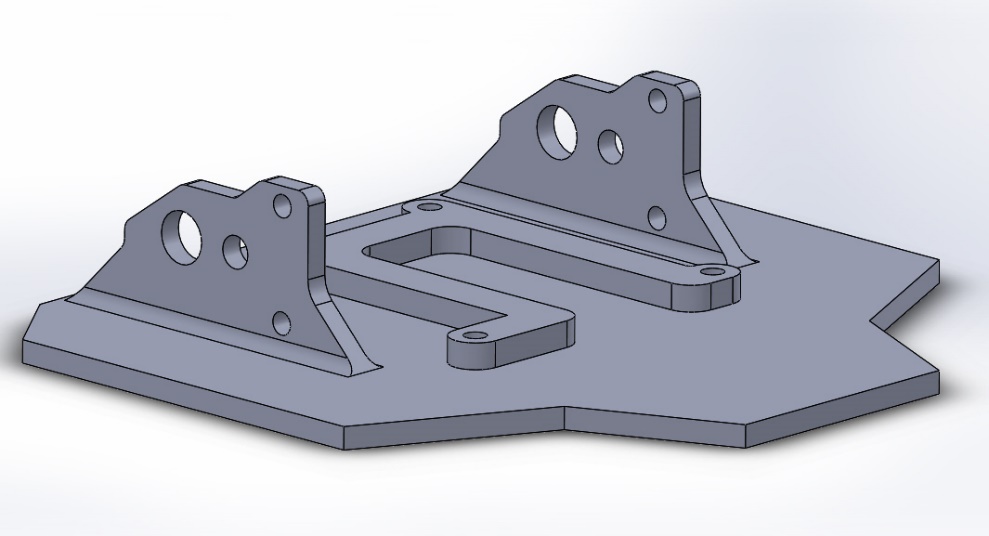
Pentru a permite aerului sa circule pe sub placa de dezvoltare Arduino UNO R3, am decis ca adaugarea unui suport mai inalt pentru electronica este necesar. Folosind aceeasi tehnica precum pana acum, si anume creerea unui Sketch cu dimensiunile necesare apoi utilizarea functiei “Extrude Boss Base” , am creat acest suport pentru placa de dezvoltare.



In lateralele acestui suport se pot observa si cei doi suporti la 90 de grade pentru motoarele DC de tractiune. Acestea au fost create folosind aceeasi rutina, insa au fost adaugate niste repere de sustinere pentru a creste fiabilitatea ansamblului, cum ar fi cele doua prelungiri din lateralele peretelui. Totodata, au fost adaugate doua prelungiri rotunjite la baza peretelui. Acestea sunt numite “Fillets” in programul Solidworks 2016 si sunt folosite pentru a sparge unghiurile de 90 de grade in curbe de raza specificata de noi.

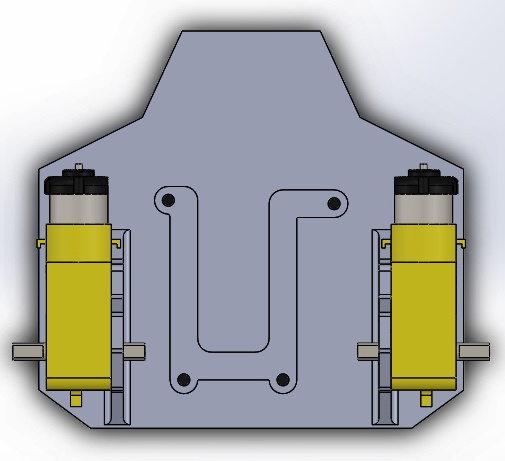


Avand toate aceste repere realizate, singurele monturi ce mai trebuie adaugate ar fi cele pentru senzorii de proximitate IR. Acestia au o singura gaura de montaj, deci singura problema pe care o vom intampina va fi de simetrie fata de axa centrala a robotului. Insa, folosind functia “Mirror” din programul Solidworks 2016 am efectuat cu succes oglidirea reperului de montaj pentru senzorul de proximitate IR.   
Mai mult decat atat, comunitatea open-source de imprimare 3d are in biblioteca de piese chiar si senzorul pe care il vom folosi. Astfel putem vizualiza si mai bine imaginea de ansamblu a robotului pe care il proiectam, adaugand toate componentele necesare direct in programul de pe calculator, Soldiworks 2016.

 Aceasta simulare ne va oferi informatii si despre dimensiuniel finale ale robotului, greutatea si fiabilitatea lui in urma efectuarii testelor din cadrul “Simulation”. Aceste teste nu sunt necesare iar efectuarea lor este pur informativa.

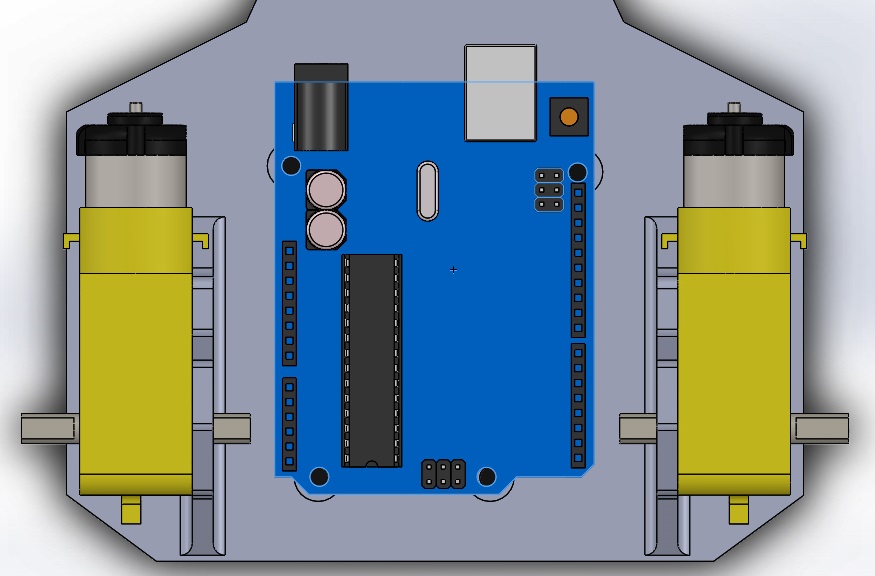
In imagine se observa platforma completa pentru robotul proiectat.

Avand toate componentele necesare sub forma unor modele 3D, putem incepe asamblarea virtuala a proiectului , adaugand rand pe rand cate o componenta si mentionand constrangerile de montaj pentru fiecare dintre acestea.   
 Pentru o exemplificare mai usoara, vom prezenta un ghid de asamblare digital cu poze explicite pentru fiecare pas important. Astfel, procesul devine repetitiv chiar si pentru cei care citesc pentru prima data acest document.



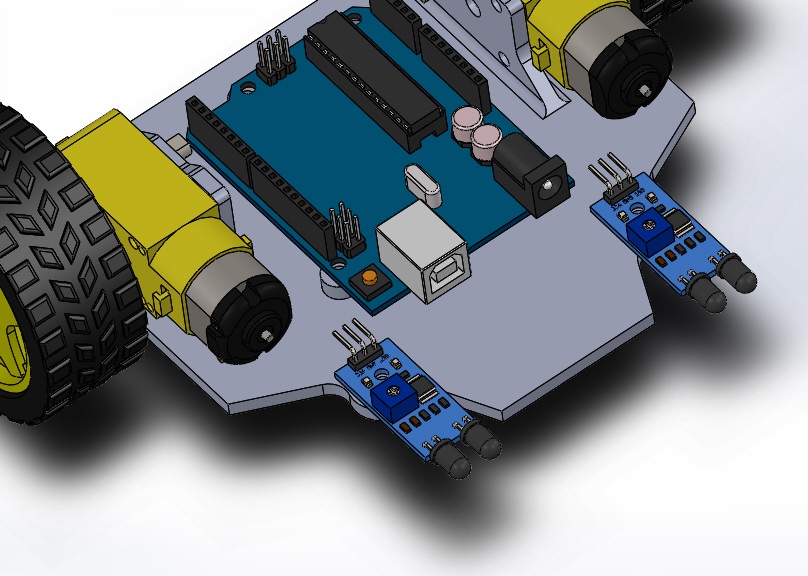
Pasul 1:

Primele doua piese care se vor monta sunt cele doua motoare de transmisie. Acestea se prind cu ajutorul a 2 suruburi M3x30mm (suruburi metric 3 cu lungimea filetului de 30 mm) . Dupa adaugarea suruburilor si insurubarea in plastic in gaurile aferente, se adauga doua piulite M3 pentru a definitiva fixarea celor doua motoare.

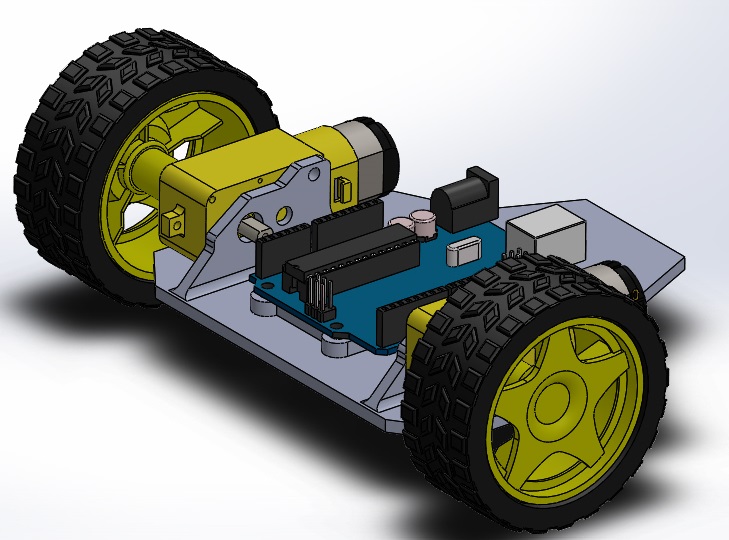


Pasul 2:

Se adauga cele 4 suruburi cu cap ingropat M3x15 (metric 3 cu lungimea filetului de 15 mm) din partea de jos a platformei pana cand trec prin gaurile de montare a placii de dezvoltare Arduino UNO R3. Se adauga 4 piulite M3 pentru definitivarea monturii placii de dezvoltare

 Pasul 3

Se adauga cele doua suruburi cu cap ingropat M3x15 (metric 3 cu lungimea filetului de 15 mm) din partea de jos a platformei, dupa care se pozitioneaza cei doi senzori intr-o pozitie favorabila detectiei de obstacole. Se adauga cele doua piulite M3 pentru definitivarea pozitionarii celor doi senzori de proximitate IR.

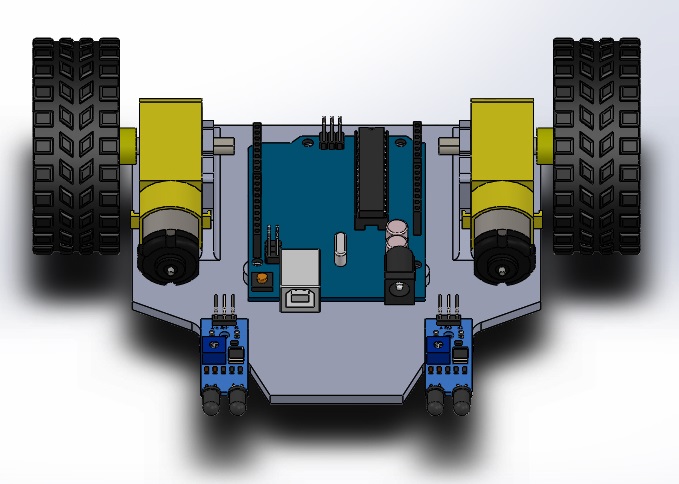


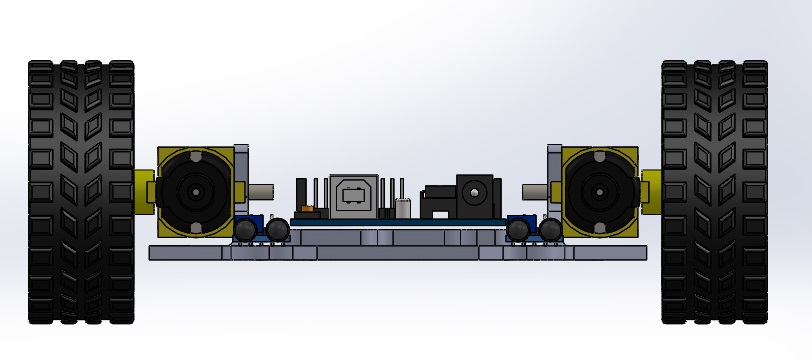
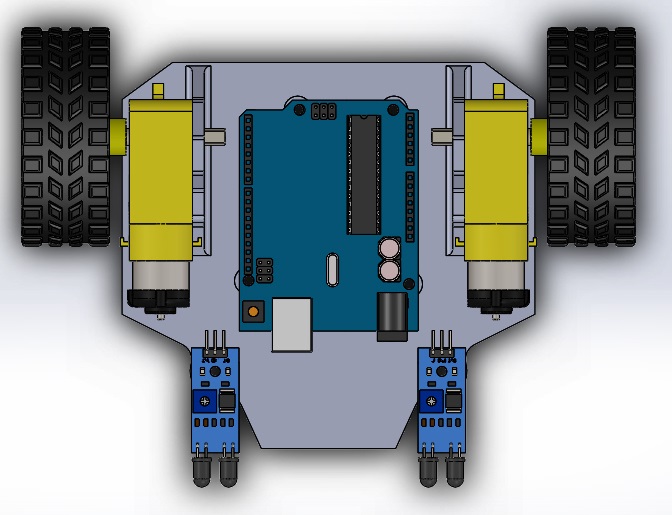
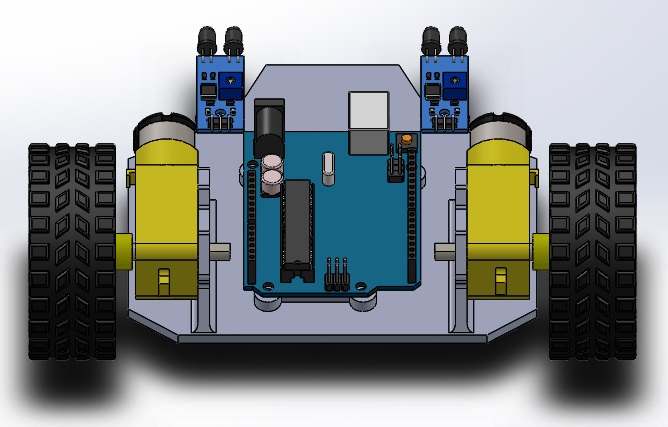
Pasul 4

Se adauga cele doua roti pentru motoarele de tractiune. Acestea se preseaza pe axul motorului. Axul motorului are doua suprafete drepte pentru a evita alunecarea rotii pe ax. Aceste roti nu au nevoie de montaj cu suruburi deoarece vor fi tinute pe pozitie de forta de frecare dintre piesele de plastic (Ax si suport roata). Ulterior se adauga cele doua anvelope (cauciucuri) .

Nu in ultimul rand se adauga Shield-ul Adafruit Motor Shield si se efectueaza conexiunile conform schemei ce va fi prezentata in Capitolul 3 – Proiectare Hardware.

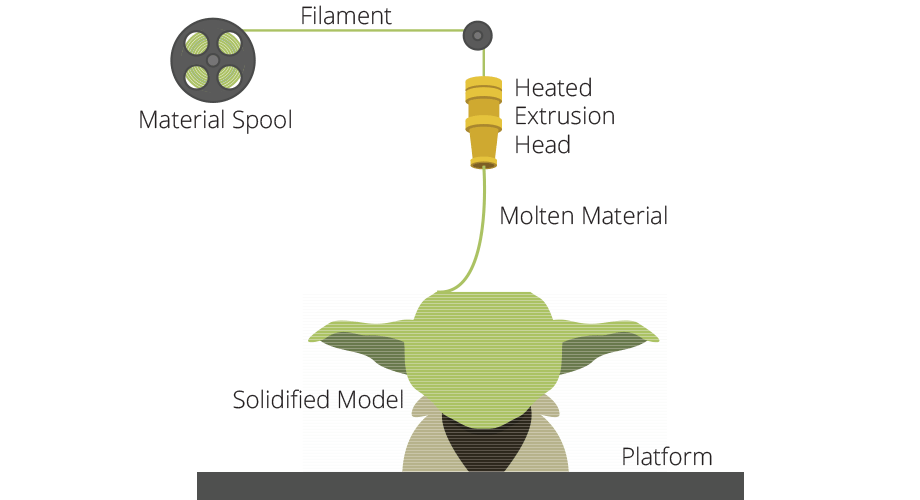
In partea din fata a robotului se va adauga ulterior o roata de echilibru ce se poate roti in orice directie, aceasta are doar rol de a mentine robotul la o inaltime constanta,egala de-a lungul platformei, si nu are rol in tractiunea sau virarea lui. O alternativa simpla ar fi o piesa de mobilier des intalnita la mesele de cafea, si anume o roata cu mai multe grade de libertate.

Acestea fiind spuse, robotul proiectat si simulat pe calculator ar trebui sa aiba urmatorul aspect:



**Capitolul 2.2 – Imprimarea 3D – UNDA DeltaPRO**

Odata finalizat procesul de proiectare 3D a venit vremea imprimarii pieselor necesare robotului prezentat. In vederea acestui proces a fost utilizata o imprimanta 3D UNDA DeltaPro. Aceasta difera prin modul de miscare de majoritatea imprimantelor disponibile, acestea folosind un sistem de miscare cartezian, iar UNDA DeltaPro folosind un sistem DELTA ROBOT.

 Serviciile de imprimare 3D au fost din partea UNDA Tech.

Sursa:

Procesul de imprimare cu ajutorul modelului FFF ( Fused Filament Fabrication ) consta in incalzirea unui material care vine sub forma de filament bobinat (de regula in cantitate de 1kg) pana cand acesta ajunge intr-o stare semilichida. In acest moment el este impins de un Extruder care are la baza o rotita dintata care musca partial din filament. Aceasta rotita este invartita de un motor pas cu pas Nema 17. Practic, rotita musca din filament si il impinge printr-un tub de teflon care are rolul de a ghida filamentul pana in zona incalzita.

Zona incalzita este constituita dintr-o duza de extrudare si un bloc de incalzire. Acest bloc are o rezistenta tip cartus ceramica. Aceasta rezistenta poate ajunge pana la temperaturi de 300 grade Celsius, determinand astfel si limita superioara a materialelor plastice ce se pot topi. Langa aceasta rezistenta se regaseste un termistor NTC (Negative Temperature Coefficent) de 100KOhm, care masoara temperatura curenta a hotend-ului .

Aceasta temperatura depinde materialul folosit, spre exempu exista materiale care necesita temperaturi de 190°C, dar si materiale care necesita temperaturi de peste 250°C.

Un tabel cu temperaturile aferente mai multor materiale se regaseste mai jos.

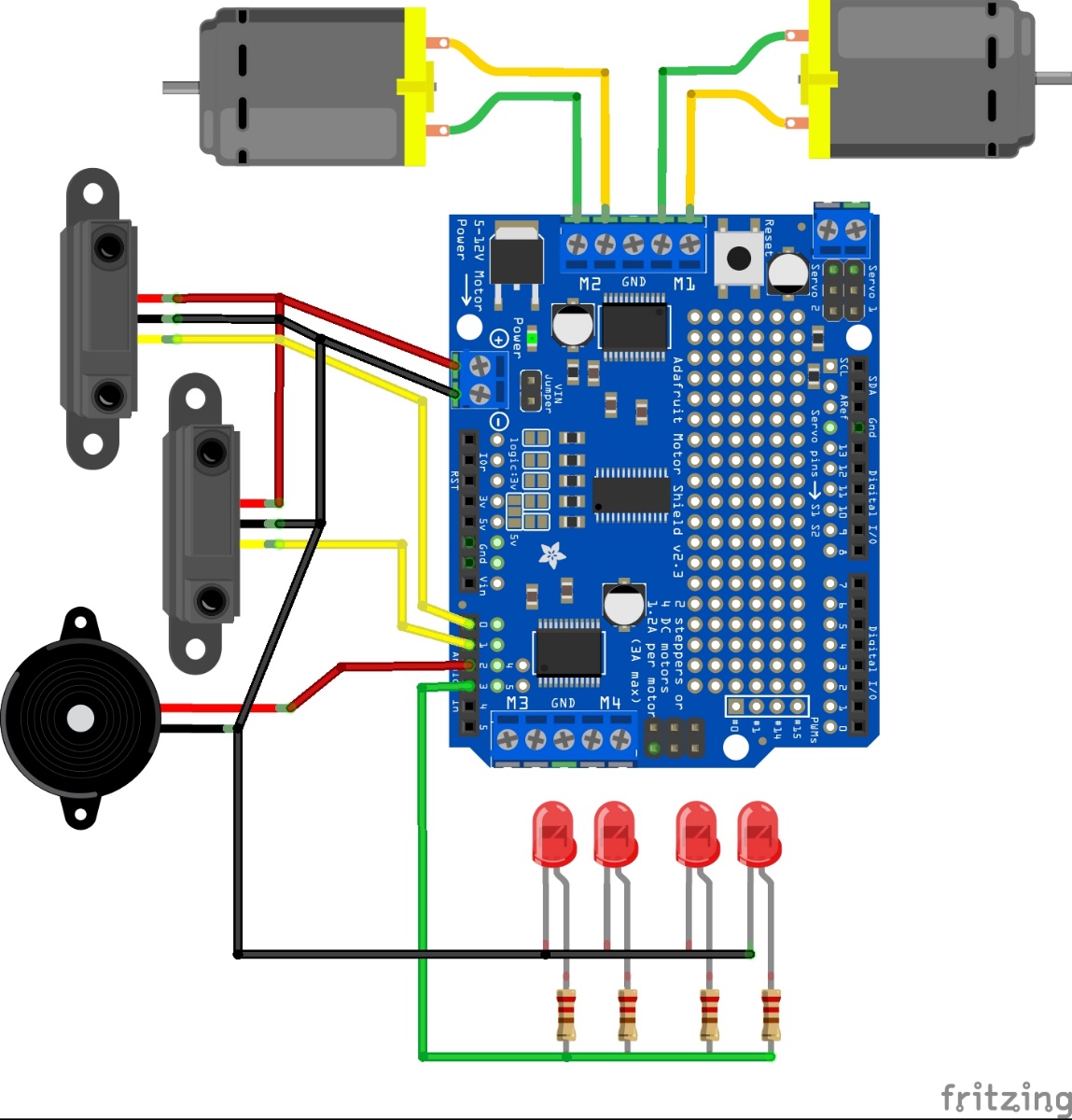
|  |  |  |  |
| --- | --- | --- | --- |
| **MATERIAL** | **TEMP HOTEND** | **TEMP PAT** | **OBSERVATII** |
| PLA | 185-210 | 0-60 | T<200 Mat  T>200 Lucios |
| ABS | 230-250 | 90-120 | T<235 Mat  T>235 Lucios |
| ABS+ | 220-250 | 77-110 | T<230 Mat  T>230 Lucios |
| PETG | 220-260 | 60-80 | T<235 Mat  T>235 Transparent |
| TPU | 195-230 | 65-75 | T<210 Rigid T>210 Flexibil |
| HIPS | 230-250 | 95-115 | A nu se dezlipi inainte de 25 |
| Laywood PLA | 160-240 | 0-55 | T<200 bright wood  T>200 dark wood |
| CarbonFiber PLA | 190-230 | 40-65 | Minim 0.5mm Nozzle Inox |

Tabelul 2-1 : Sursa: UNDA Troubleshooting Guide PDF

Pentru modelul proiectat anterior materialul ideal este PETG, acesta este acelasi material ca cel regasit in sticlele de plastic de tip PET (de unde este si denumirea). Este un plastic rezistent,flexibil si cu un punct de topire destul de ridicat. Acesta este un avantaj major avand in vedere ca robotul proiectat are si motoare (care se pot incalzi daca sunt utilizate la putere maxima pentru o perioada mai lunga de timp) . Un alt avantaj al acestui material ar fi calitatea produsului finit, fiind un material care se imprima cu usurinta si ofera un aspect foarte placut produsului finit.

**Capitolul 3 – Proiectare Hardware**

Pentru a da viata pieselor mecanice si electronice care constituie robotul proiectat, trebuie sa efectuam o seria de legaturi electrice intre aceste module pentru a le stabili functionalitatea. In acest capitol vom parcurge procesul de determinare a legaturilor optime pentru ca robotul proiectat sa aiba functionalitatile propuse.

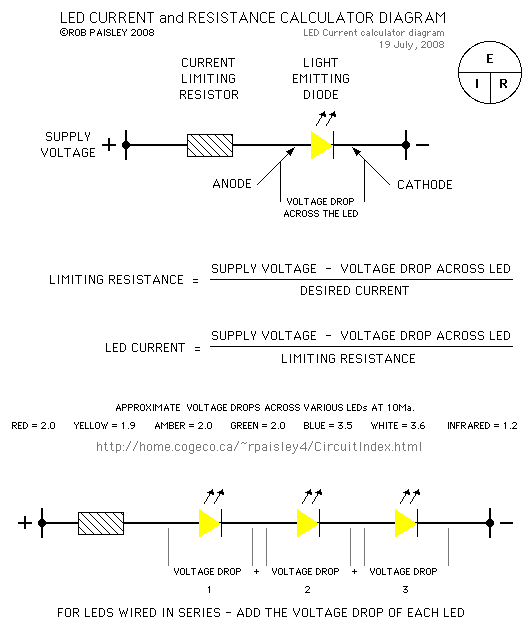


Pentru a realiza o schema grafica a proiectului descris anterior, vom utiliza programul Fritzing. Fritzing este un program gratuit, open-source, care ofera posibilitatea realizarii schemelor electrice intr-o maniera grafica. Spre exemplu, acesta include o sumedenie de librarii care contin desene grafice 2D pentru componente electrice des intalnite in domeniul electronicii. Aceste componente au definite si conexiunile (porturile) externe, intrucat sunt create cu ajutorul programelor de grafica vectoriala si predefinite ca si conexiuni externe.

Un avantaj considerabil in utilizarea acestui program este faptul ca ne putem face o idee despre cum vom pozitiona componentele electronice pe platforma robotului. Acestea adesea sunt numeroase si cu dimensiuni foarte variate si devine greu sa le luam in considerare in procesul de proiectare. Mai mult decat atat, putem observa si complexitatea conexiunilor, numarul de cabluri de conexiune necesare etc. Utilizarea acestui program ofera oricarui entuziasmat de electronica posibilitatea de a-si organiza mult mai bine procesul de proiectare si realizare a unui produs, trecand prin toate etapele necesare.

Avand in vedere ca singurii pini disponibili pe placa de dezvoltare Arduino Uno R3 sunt cei 6 pini analogici, vom folosi 4 dintre acestia pentru preluarea de date din exterior si livrarea rutinei de atentionare prin intermediul diodelor LED si a buzzer-ului activ.  
 - Astfel, achizitia de date se face pe pinii A0,A1 prin intermediul senzorilor de proximitate IR. Dat fiind faptul ca senzorii de proximitate au un output digital, va trebui ca acesti pini care sunt la baza pini de tip analog sa fie declarati ca pini de input digital. Acest lucru se rezolva cu usurinta in antetul proiectului folosind “pinMode(nrPin,modPin)” iar apoi citirea cu “digitalRead(nrPin)” . Oricand unul dintre acesti doi senzori de proximitate IR are iesirea LOGIC HIGH, robotul nostru trebuie sa intre in rutina de oprire si avertizare. Astfel va trebui sa implementam in conditie modificarea oricarui pin dintre A0 sau A1 din LOGIC LOW in LOGIC HIGH.

- Pentru atentionarea sonora se va utiliza pinul A2. La acest pin se va conecta terminalul pozitiv al buzzer-ului activ. Pentru a activa avertizarea sonora este suficient ca pinul A2 sa devina logic HIGH pentru o perioada determinata de timp , iar acesta va emite avertizari sonore de intensitate constanta pe perioada determinata. Pentru a face avertizarea mai eficienta, vom implementa o functie ce va apela repetat avertizarea sonora, formand un sunet intermitent ce va fi mai usor de observat. Daca intensitatea sunetului produs de buzzer-ul ales este prea mare, putem folosi pinul A2 ca pin analog si putem implementa un PWM care sa reduca volumul buzzer-ului la valoarea dorita.

- Pentru atentionarea luminoasa se va utiliza pinul A3. La acest pin se va conecta magistrala cu terminale pozitive (Anod) prin intermediul unor rezistoare de 150Ohm.   
Acestea vor fi legate pe fiecare led in parte si au rolul de a limita curentul ce trece prin dioda la o valoare specifica culorii diodei.

In cazul de fata, avand un led de culoare ROSIE, putem regasi in foaia de catalog faptul ca diodele LED rosii au o cadere de tensiune de 2V pe dioda si suporta un curent maxim de 25mA.   
  
In acest caz, noi livram 5V de la pinul A3 al placii de dezvoltare Arduino Uno R3, si trebuie sa alegem o rezistenta de valoare convenabila pentru a limita curentul ce trece prin dioda.   
Sursa: [Ledcalc](http://ledcalc.com/)

Daca aceste rezistente de limitare de curent nu ar fi prezente, diodele led ar putea fi afectate atat pe loc cat si pe perioada mai lunga(in cazul in care rezista unui soc de curent).

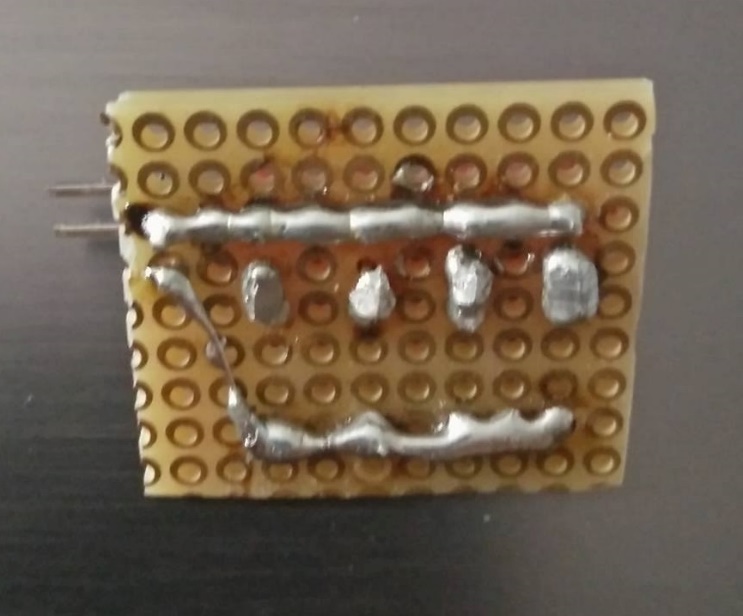
Pentru a prelungi durata de viata a acestora si pentru a le utiliza in conformitate cu specificatile producatorului, am efectuat urmatorul calcul:

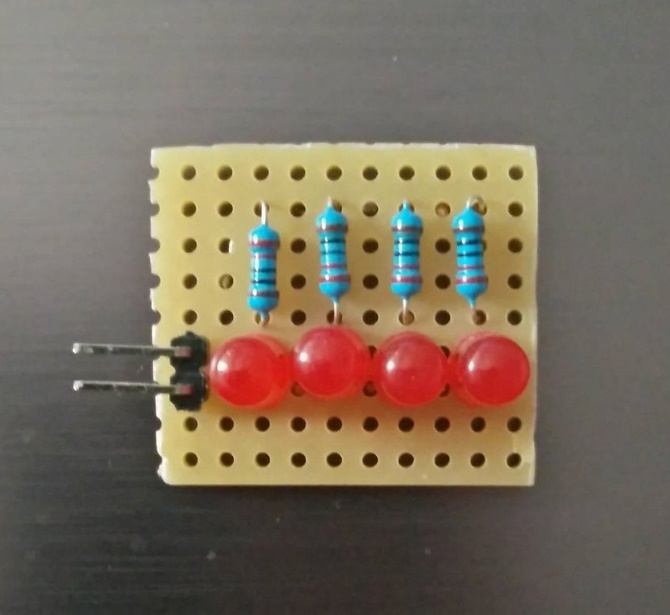
Cunoscand Vcc = 5V , Vled = 2V si curentul maxim Imax = 25mA putem deduce

Totusi, pentru siguranta si stabilitate vom alege un rezistor de 150Ohm. In acest caz curentul cu siguranta nu va depasi valoarea de 25mA si dioda led va lucra in parametri extinzand cat de mult cu putinta durata de viata a ei.

In acest sens in momentul in care vom scrie pe pinul A3 o valoare de 5V, cele 4 leduri se vor aprinde la intensitate maxima. Pentru a acompania avertizarea sonora, aceste leduri vor fi aprinse intermitent exact in aceeasi maniera in care va fi activat si buzzer-ul.

Pentru a ingloba intr-un singur modul rezistentele si diodele led necesare, am folosit o placa de tip protoboard care este prevazuta cu gauri de 1mm si cu distanta de 2.54mm. Aceste placi se folosesc in general pentru prototiparea modulelor ce urmeaza a fi fabricate pe masini CNC sau prin depunere chimica.

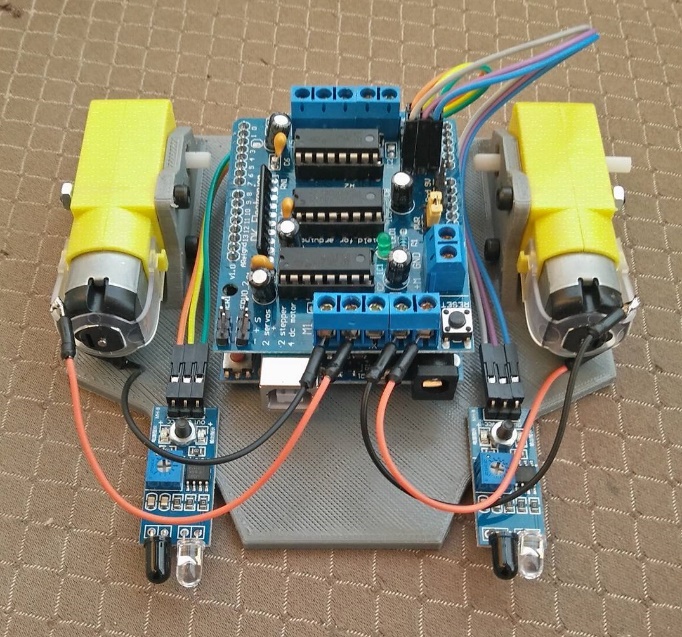
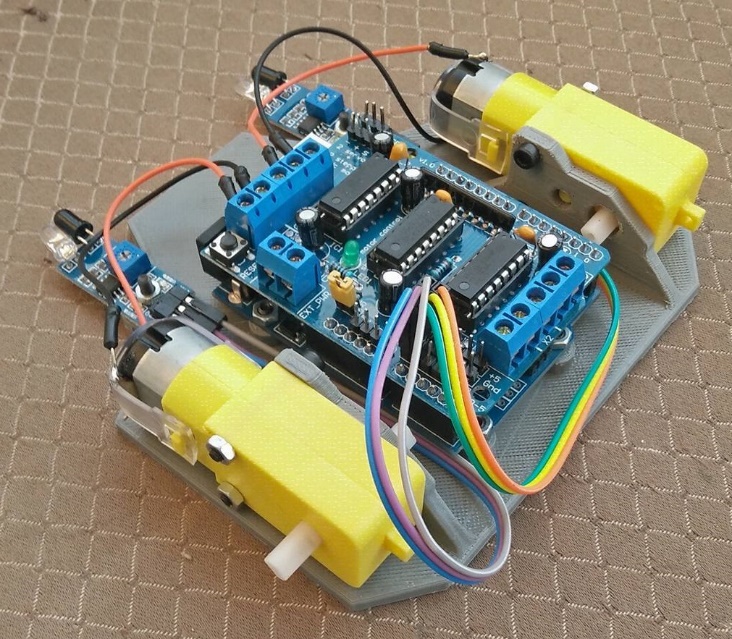
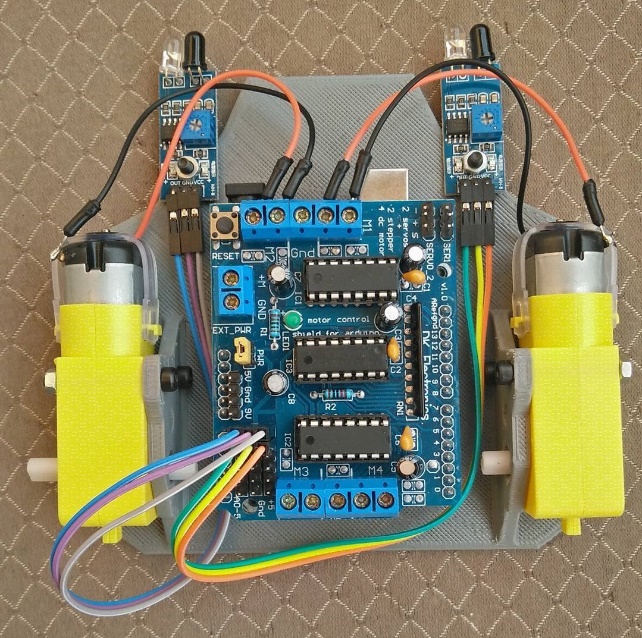
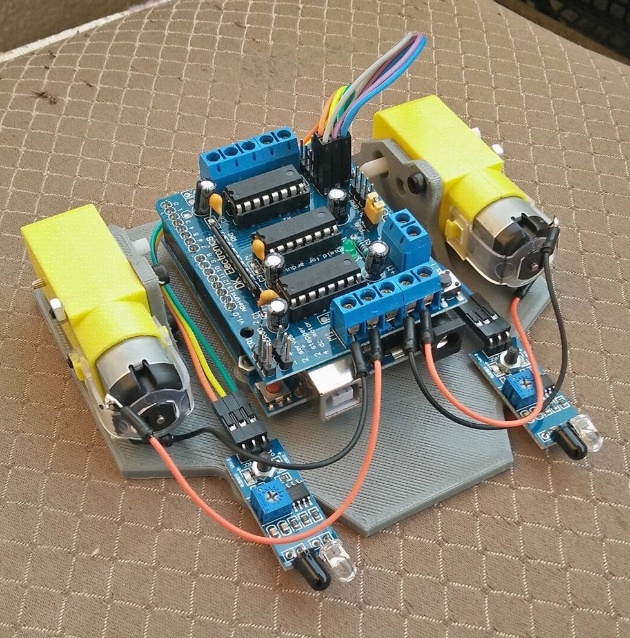




Toate conexiunile au fost efectuate cu ajutorul cablurilor tip panglica. Aceste cabluri au la capete conectori Dupont si vin in mai multe modele: Mama-Tata, Mama-Mama si Tata-Tata. In cazul de fata am folosit cabluri mama-tata si mama-mama pentru a conecta motoarele,senzorii,diodele led si buzzer-ul activ. Pentru conexiunile care au necesitat lipire cu statia de lipit s-a folosit acelasi tip de cablu care a fost dezizolat si decapat inainte de lipire. Lipirea s-a facut cu fludor fara plumb la temperatura de 340°C.

Alimentarea modulului se va face cu ajutorul unei baterii de 9V. Aceasta baterie va alimenta atat placa de dezvoltare Arduino Uno R3 cat si shield-ul de control pentru motoare Adafruit Motor Shield. Avand in vedere ca tensiunea nominala de lucru pentru placa de dezvoltare este 5V, este necesara o conversie la aceasta tensiune. Partea buna insa este ca platforma de dezvoltare Arduino Uno R3 are integrat un regulator de tensiune AMS1117-5V care permite o gama larga de tensiuni de intrare, intre 6 si 12V pentru un randament multumitor.

Modulul de control Adafruit Motor Shield are o alimentare separata, insa avand in vedere ca acesta poate lucra cu tensiuni de intrare intre 8V si 16V, bateria va servi cu succes alimentarea chip-urilor de control si anume a L298D.

****

**Capitolul 3 – Proiectare Software**

Pentru a oferi inteligenta robotului proiectat anterior, avem nevoie de o serie de linii de cod care sa stabileasca in ce circumstante se regaseste robotul. In acest sens vom folosi un IDE (Integrated development environment) precum Arduino IDE.   
  
Arduino IDE este un software scris in limbajul de programare Java. Acesta a fost inspirat din predecesorii lui, Processing sau Wiring.  
  
 Mai mult, Arduino IDE ofera aproape toate functionalitatile unui IDE pentru C/C++ cum ar fi indentare automata,sugestii de scriere in timp real etc. Motivul pentru care acest IDE este foarte utilizat la nivel international este ca ofera un pachet de librarii foarte diversificat, ideal pentru incepatorii care nu au avut acces la modurile de adresare specifice microcontrollerelor AVR. Astfel, acesta include un set de instructiuni bine documentate si foarte usor de retinut. **[24]**

In vederea realizarii software-ului necesar pentru robot-ul proiectat anterior, vom avea nevoie de :

- o functie care citeste si verifica distanta pana la obstacol si decide daca obstacolul este prea aproape de robot  
 - o functie care efectueaza o rutina de atentionare pentru cele doua metode de atentionare – dioda LED si buzzer-ul activ  
 - variabile globale in care vom stoca diferiti parametri.

Libraria AF\_DCMotor.h oferita de Adafruit include functii ce usureaza utilizarea shield-ului Adafruit Motor Shield. In acest sens vom crea doua obiecte leftMotor si rightMotor care vor fi initializate pe conectorii M1 si M2 . Acesti conectori (M1,M2) ofera posibilitatea utilizarii unei frecvente PWM de pana la 64KHz. Utilizarea unei frecvente mari a PWM-ului va face motoarele sa lucreze mai discret. In schimb, utilizarea PWM-ului la 64KHz va atrage cu sine un consum mai mare de energie. Insa, avand in vedere ca aceste motoare nu au un consum semnificativ iar alimentarea va fi facuta de la o baterie de 9V, problema consumului de curent nu va fi un impediment vizibil.  
 Libraria ofera totusi posibilitatea utilizarii mai multor frecvente de PWM pentru motoarele conectate. Astfel, terminalele pe care am decis sa le utilziam (M1,M2) ofera posibilitatea alegerii unor frecvente de 64KHz , 8KHz , 4KHz , 2KHz si 1KHz . Cea mai mare frecventa va oferi o functionare foarte tacuta insa cu un consum de energie mai mare si un cuplu redus . Pentru frecvente mai mici motoarele vor deveni mai galagioase iar cuplul acestora va creste . Pentru aplicatia de fata am decis ca robot-ul nu are nevoie de un cuplu semnificativ si vom utilzia o frecventa PWM de 64KHz .

Un avantaj vizibil al utilizarii librarie AF\_DCMotor.h este ca aceasta prezinta doua functii foarte importante in utilizarea motoarelor DC. Una dintre acestea se poate apela in void setup() si se numeste motor.setSpeed(Speed) unde “motor” este numele obiectului definit ca motor iar “Speed” este o valoare intre 0 si 255 unde 0 reprezinta oprit iar 255 reprezinta viteza maxima cu care motorul poate roti axul exterior. Mai mult decat atat, o functie denumita motor.run(Direction) ofera posibilitatea controlarii motorului in ambele sensuri prin folosirea argumentului FORWARD sau BACKWARD. Totusi ,exista si posibilitatea de a opri motorul folosind argumentul RELEASE. Acesta este exact modul de lucru pe care il urmarim. Robotul va functiona cu motor.run(FORWARD) iar la intalnirea obstacolului se va apela functia motor.run(RELEASE).

**In continuare vom vizualiza codul sursa pentru robotul cu detectie de obstacole**

In primul rand vom include libraria AFMotor.h pe pusa la dispozitie de catre Adafruit pentru utilizarea modulului pentru motoare Adafruit Motor Shield

#include <AFMotor.h>

In continuare vom crea cele doua obiecte leftMotor si rightMotor care vor abstractiza cele doua motoare de tractiune pentru robotul nostru. Aici se observa ca argumentele sunt “1” si “2” pentru M1 si M2 pe shield-ul Adafruit iar frecventa PWM este setata la 64KHz pentru un rulaj fin si cu nivel de zgomot mai mic.

AF\_DCMotor leftMotor(1, MOTOR12\_64KHZ);

AF\_DCMotor rightMotor(2, MOTOR12\_64KHZ);

Vom crea si initializa variabilele aferente pinilor placii de dezvoltare Arduino Uno R3 , denumind sugestiv senzorii (left,right) , buzzer-ul activ si led-urile ce sunt legate in paralel. Tot aici se defineste o variabila in care vom stoca prezenta obstacolelor. Avand in vedere ca senzorii de proximitate IR sunt “Active Low” (adica la intalnirea unui obstacol senzorii transmit pe terminalul Signal un semnal LOGIC LOW) vom initializa aceasta variabila cu HIGH, adica practic ca nu exista obstacol.

int leftSensorPin = A0;

int rightSensorPin = A1;

int buzzerPin = A2;

int ledPin = A3;

bool isObstacle = HIGH;

Pentru a detecta obstacolele vom crea o functie de tip void() . Aceasta functie nu va returna nimic la terminare insa va modifica variabila globala isObstacle in cazul intalnirii unui obstacol. Astfel vom putea apela functia dupa care vom verifica starea variabilei isObstacle. In functie vom efectua doua masuratori. Acestea se vor face cu ajutorul functiei digitalRead(PIN) ce provine din libraria standard Arduino IDE.   
Aceasta functie verifica starea logica a pin-ului dat ca argument, returnand 1 pentru Logic HIGH sau 0 pentru Logic LOW. Aceste stari vor fi salvate in variabilele left si right pentru a putea fi prelucrate ulterior.   
 Se observa ca am adaugat un delay de 10ms (milisecunde) pentru a oferi senzorilor de proximitate IR suficient timp de initializare a rutinei si pentru ca undele IR emise de unul din senzori sa nu interfere cu celalalt.

void detectObstacle()

{

int left = digitalRead(leftSensorPin);  
 delay(10);  
 int right = digitalRead(rightSensorPin);  
 delay(10);

Odata salvate starile celor doi senzori, acestea se pot prelucra pentru a decide ce rutina sa abordeze robotul. Se va verifica cu ajutorul unui IF daca fie unul dintre senzori sau celalalt a detectat obstacol, iar daca aceasta conditie este indeplinita scriem in Serial Monitor un mesaj (foarte util pentru debugging) si modificam valoarea isObstacle in Logic LOW. In caz contrar, suprascriem isObstacle cu Logic HIGH iar programul va continua masuratorile pe pinii leftSensor si rightSensor

if( left == LOW || right == LOW )

{

Serial.println("S-a detectat obstacol!!!");  
 isObstacle = LOW;  
 delay(100);

}

else

isObstacle = HIGH;

}

Pentru a avea control si asupra motoarelor, vom implementa o functie de oprire a motoarelor de tractiune. Aceasta va folosi metoda motor.run(DIRECTION) implementata de Adafruit in libraria AFMotor.h . Aceasta metoda primeste ca argument directia in care motorul sa se invarta, si anume “FORWARD” , “BACKWARD” sau “RELEASE”. Cea din urma fiind ce ne intereseaza, si anume oprirea motorului.

void motorStop(){

leftMotor.run(RELEASE);  
 rightMotor.run(RELEASE);

}

Dupa implementarea functiei de oprire vom implementa functia de deplasare pe directia inainte. Aceasta va fi implementata asemeni functiei de oprire cu exceptia argumentului metodei motor.run(DIRECTION) , care va fi setat pe FORWARD.

void motorRun()

{

leftMotor.run(FORWARD);  
 rightMotor.run(FORWARD);

}

In scopul avertizarii mediului inconjurator la intalnirea unui obstacol vom implementa o functie ce efectueaza o serie de atentionari sonore si vizuale. Pentru inceput vom aprinde intermitent diodele led rosii o data la 200ms (milisecunde) . In aceeasi maniera in care efectuam modificari asupra terminalului la care sunt legate diodele LED modificam si terminalul la care este legat buzer-ul Activ.   
 Aceste modificari se fac cu ajutorul functiei analogWrite(pin,val) si digitalWrite(pin,state) . Pentru functia analogWrite(pin,val) se va alege pin-ul corespunzator buzzer-ului activ si o valoare cuprinsa intre 0-255 unde 0 reprezinta oprit iar 255 reprezinta activ iar pentru digitalWrite(pin,state) se va alege pin-ul corespunzator diodelor led si HIGH (1) sau LOW (0) pentru a aprinde/stinge diodele led.   
 Delay-ul de 200ms a fost adaugat cu scopul de a crea o intermitenta in rutina de avertizare.

void showAvertizare()

{

Serial.println("Initiez rutina de avertizare!!!");

analogWrite(buzzerPin,255);  
 digitalWrite(ledPin,HIGH);  
 delay(200);

analogWrite(buzzerPin,0);  
 digitalWrite(ledPin,LOW);  
 delay(200);

analogWrite(buzzerPin,255);  
 digitalWrite(ledPin,HIGH);  
 delay(200);

analogWrite(buzzerPin,0);  
 digitalWrite(ledPin,LOW);  
 delay(200);

}

Odata finalizata implementarea si initializarea variabilelor si functilor necesare, vom adauga codul necesar pentru void setup(). In aceasta zona vom initializa monitorul serial (Serial Monitor ) cu un baudrate de 9600. Acest baudrate poate lua valori de la 9600 pana la 250000, insa pentru un proiect care nu comunica foarte mult pe serial, o valoare de 9600 este suficienta. In continuare vom afisa un mesaj specific robotului (o rutina de intializare,un titlu,numele robotului,numele creatorului etc).

void setup() {

Serial.begin(9600);  
 delay(500);

Serial.println("Mini-robot cu detectie obstacole");  
 delay(500);

Serial.println("Licenta 2017");  
 delay(500);

Serial.println("Ioan-Marian Manta");  
 delay(500);

In libraria pusa la dispozitie de Adafruit se regaseste o metoda pentru stabilirea vitezei motoarelor conectate la Adafruit Motor Shield. Aceasta este motor.setSpeed(VAL) unde VAL poate lua valori intre 0-255 unde 0 reprezinta oprit si 255 reprezinta viteza maxima. Pentru proiectul actual vom alege 255 deoarece robotul raspunde foarte rapid la aparitia unui obstacol, chiar si cu toate delay-urile adaugate in procesul de achizitie de date.

leftMotor.setSpeed(255);  
 rightMotor.setSpeed(255);

In continuare vom definit tipul de pin pentru fiecare conexiune in parte. Acest lucru se face cu ajutorul functiei pinMode(pin,tip) implementata in libraria standard Arduino IDE. Aceasta ia ca argument numarul pinului si tipul acestuia (INPUT/OUTPUT). Pentru rutina de atentionare vom avea nevoie de OUTPUT iar pentru achizitia de date vom avea nevoie de INPUT. Astfel vom defini ledPin si buzzerPin ca OUTPUT iar leftSensorPin si rightSensorPin ca INPUT.

pinMode(ledPin,OUTPUT);  
 pinMode(buzzerPin,OUTPUT);  
 pinMode(leftSensorPin,INPUT);  
 pinMode(rightSensorPin,INPUT);

O rutina de initializare/pornire a robotului a fost creata tot cu ajutorul functiilor digitalWrite(pin,state). Diodele LED se vor aprinde si stinge de 3 ori inainte ca robotul sa inceapa achizitia de date.

digitalWrite(ledPin,HIGH);  
 delay(100);

digitalWrite(ledPin,LOW);  
 delay(100);

digitalWrite(ledPin,HIGH);  
 delay(100);

digitalWrite(ledPin,LOW);  
 delay(100);

digitalWrite(ledPin,HIGH);  
 delay(100);

digitalWrite(ledPin,LOW);  
 delay(100);

}

In continuare urmeaza functia void loop() . Aceasta este functia care va fi rulata mereu de catre robotul nostru. In acest loop vor fi plasate toate functile create anterior.

void loop() {

Pentru inceput apelam functia motorRun() pentru a porni motoarele cu directia FORWARD

motorRun();

Vom apela functia detectObstacle() care ne va modifica variabila isObstacle . In acest sens vom putea verifica daca exista sau nu obstacol in calea robotului.

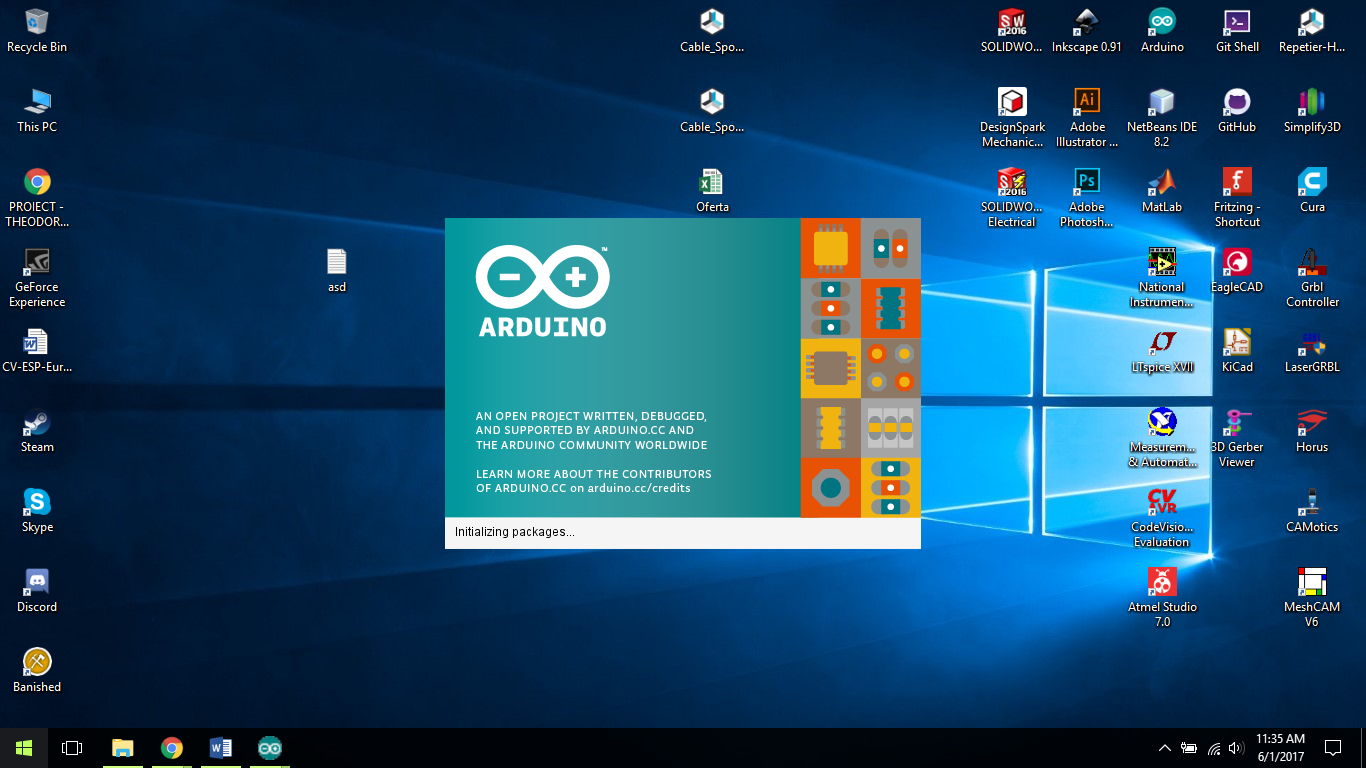
detectObstacle();

Vom verifica cu ajutorul unui IF daca exista obstacol, iar in cazul in care a fost detectat unul, vom afisa un mesaj pe monitorul serial iar apoi vom opri motoarele si vom initia rutina de avertizare. Pentru aceste actiuni se vor folosi functile definite anterior motorStop() si showAvertizare().

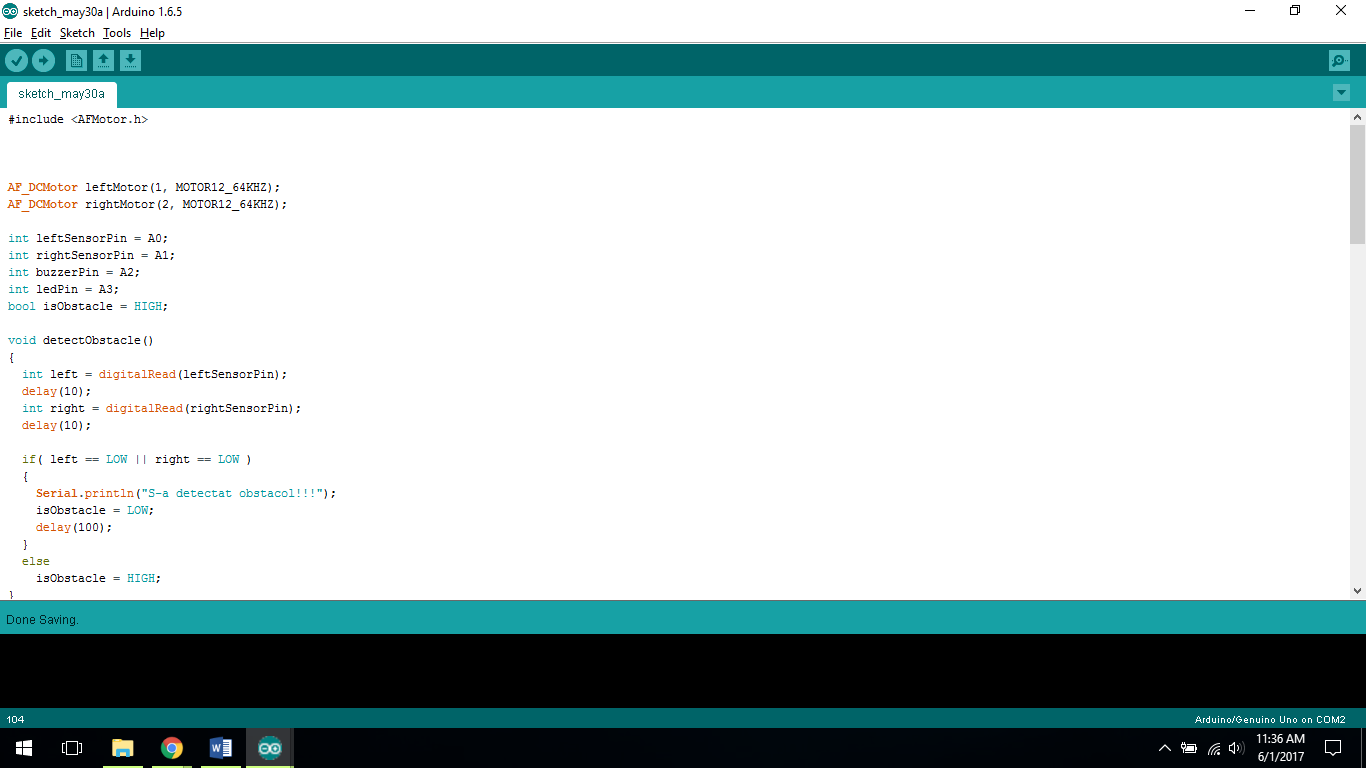
if( isObstacle == LOW )

{  
 Serial.println("S-a detectat obstacol!!!");  
 motorStop();  
 showAvertizare();  
 delay(100);  
 }  
}

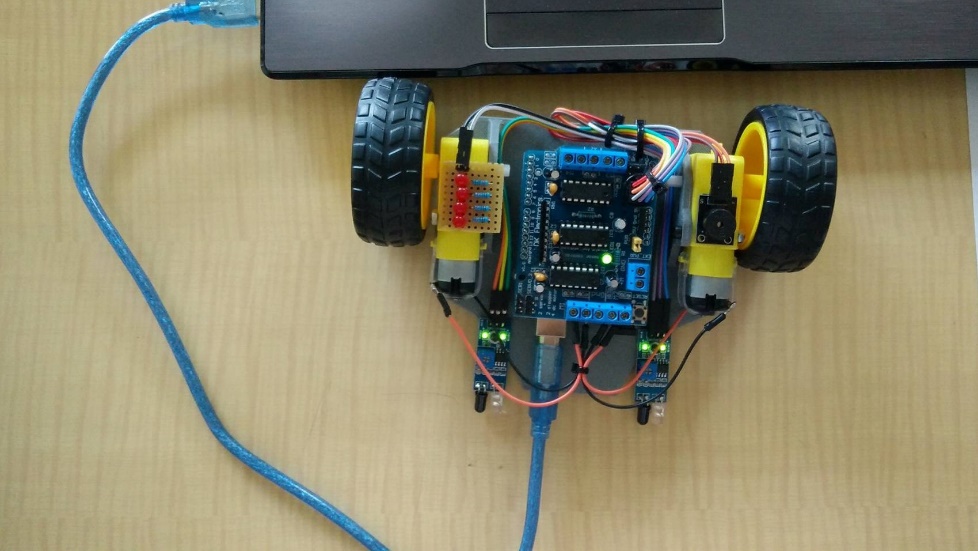
Odata finalizat procesul de programare, se va incarca codul pe placa de dezvoltare Arduino UNO R3 urmarind instructiunile:

1. 

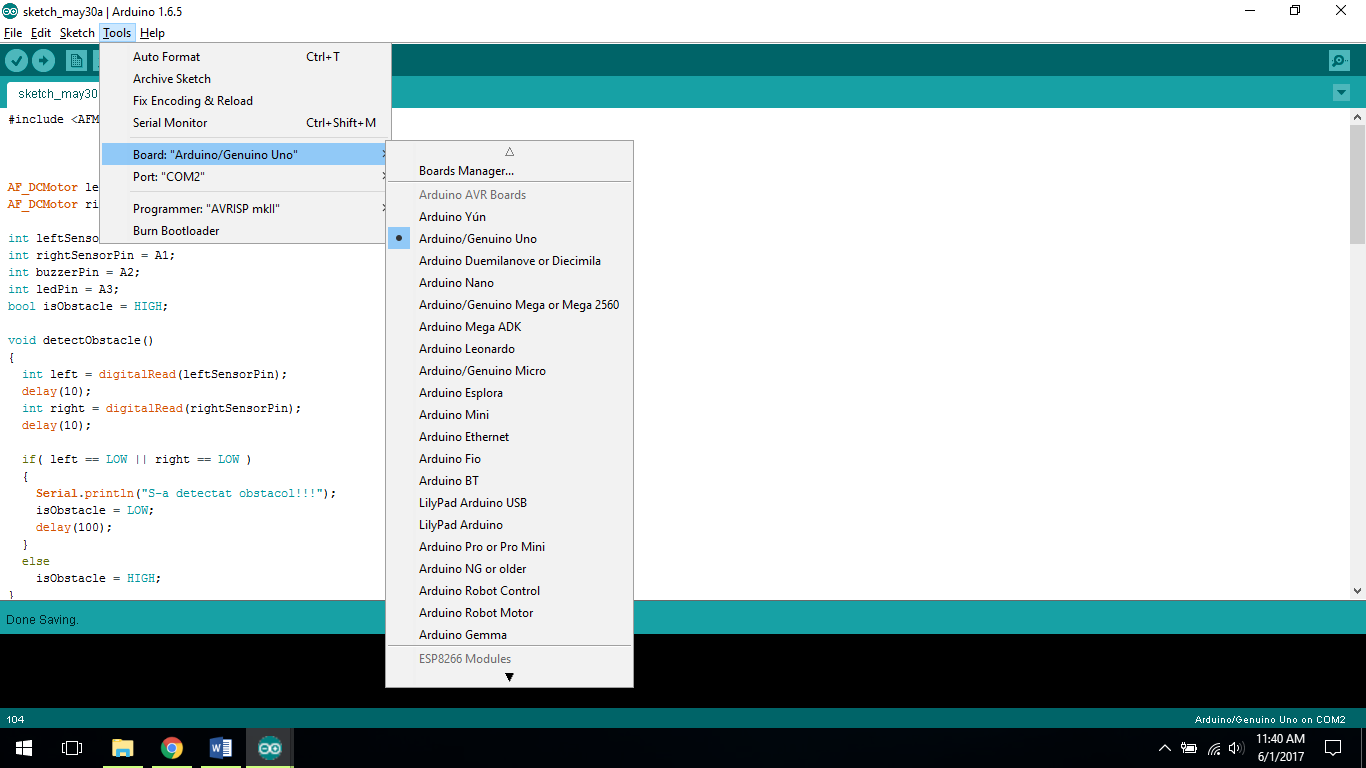
2. Se deschide codul scris anterior



3. Se conecteaza placa de dezvoltare Arduino UNO R3 la PC cu un cablu USB 2.0

****

4. Se alege din Tools>Board>Arduino UNO si Tools>Com Port>Com2

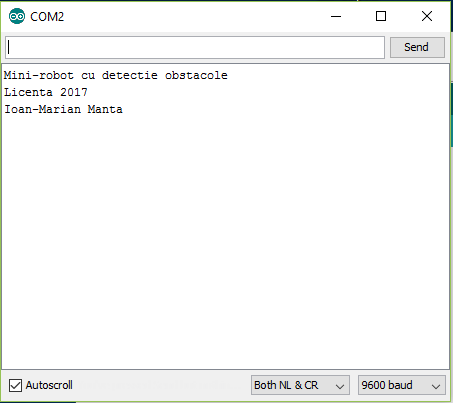


5. Se apasa butonul Upload si se asteapta incarcarea software-ului pe placa Arduino Uno



In continuare vom putea vizualiza citirile robotului, se actioneaza meniul Tools > Serial Monitor si se selecteaza din dreapta jos BAUDRATE 9600. Se bifeaza AutoScroll.

La initializare robotul va afisa:



In cazul in care se observa niciun obstacol, robotul va afisa in serial monitor aceste mesaje:



**Capitolul 4 – Concluzii si observatii**

Ajungand la finalul procesului de proiectare,prototipare si programare a minirobotului cu detectie de obstacole, am tras cateva concluzii si am notat cateva observatii relevante care si-au facut simtita prezenta pe parcursul procesului descris anterior.

**Prototiparea 3D** – Un avantaj semnificativ in domeniul acesta il prezinta prototiparea 3D. Avand acces la un astfel de serviciu de imprimare 3D atrage cu sine posibilitatea reducerii costului de productie a unui produs. Ca in cazul de fata, prototipul simulat pe calculator a fost facut in conformitate cu toate dimensiunile specificate de producatorii componentelor hardware. Insa, tolerantele sunt adesea greu de gasit pentru aceste dimensiuni si de cele mai multe ori sunt inexistente. Prototiparea 3D cu ajutorul imprimarii 3D mi-a oferit posibilitatea de a verifica toate dimensiunile cu un cost minim. Daca s-ar fi pus problema realizarii la scara larga a acestui produs, costurile numai pentru prototip ar fi fost semnificativ mai mari decat costul unui prototip imprimat 3D. Consider ca acesta este un avantaj in industria electronicii si ca inginerii ce au acces la un astfel de serviciu sunt si vor fi avantajati pe plan monetar. Mai mult decat atat, cunoasterea unui program de modelare 3D sau prototipare 3D constituie un avantaj in realizarea oricarui proiect.

**Prototiparea Hardware** – Un alt avantaj semnificativ se regaseste in regiunea hardware. In domeniul public (open-source) se regasesc foarte multe informatii,librarii sau software-uri capabile sa simuleze legaturile si conexiunile unui astfel de proiect. Spre exemplu programul gratuit din domeniul public Fritzing mi-a oferit posibilitatea de asimula toate legaturile necesare asamblarii hardware a robotului. Astfel organizarea firelor si a conexiunilor devine mult mai simpla si mult mai usor de verificat in caz de erori de proiectare. Fiind chiar in domeniul public, acesta este accesibil atat utilizatorilor profesionisti cat si a celor entuziasmati de electronica DIY. In concluzie prezenta acestor programe aduce doar beneficii domeniului prezent.

**Arduino IDE** – Programarea unui microcontroller precum ATmega328P (microcontroller-ul regasit in platforma de dezvoltare Arduino UNO R3) poate fi dificila pentru utilizatorii mai putin experimentati. Aceasta necesita un programator (convertor USB-UART), prezenta unui BOOTLOADER specific pe microcontroller si cateva conexiuni specifice microcontroller-ului. Folosirea unui IDE precum Arduino IDE ofera posibilitatea oricarui entuziast de a programa un microcontroller prin intermediul placii de dezvoltare. Astfel, nu numai ca face mult mai usoara programarea acestuia , insa si limiteaza complexitatea codului oferind foarte multe functii ajutatoare precum analogRead (care ar necesita citire dintr-un ADC al microcontroller-ului, care la randul ei presupune modificarea unor registre specifice etc), digitalRead dar si digitalWrite,analogWrite,pinMode etc.

**Inbunatatiri posibile** – Unul din marile dezavantaje ale utilizarii senzorilor de proximitate IR este ca acestia nu au un filtru IR atasat. In acest caz exista posibilitatea ca in conditii de lumina naturala puternica senzorii sa considere ca au detectat un obstacol, deoarece dioda care verifica existenta undelor IR le vede pe cele din mediul natural. Acest lucru se poate rezolva cu usurinta folosind un filtru IR. Dar, avand in vedere ca robotul proiectat are ca scop utilizarea in interior, aceasta problema poate fi considerata nesemnificativa.

- O inbunatatire posibila ar fi implementarea unui algoritm software de ocolire a obstacolului intalnit. Acesta poate fi efectuat prin rotirea celor doua motoare in sensuri opuse pentru o rotatie de 90° apoi deplasare inainte si din nou rotatie la 90°. Insa, pentru aceasta implementare ar fi recomandat sa adaugam si senzori de proximitate IR pe lateralele robotului, pentru ca el sa stie si cand a trecut de obstacol, ajustandu-si singur distanta de ocolire.

- O alta inbunatatire ar putea fi utilizarea unor senzori de proximitate Sharp. Acesti senzori sunt de regula mai scumpi insa precizia si distanta de masura a lor este vizibil mai mare. Singurul dezavantaj al acestora ar fi faptul ca au nevoie de un traductor, insa se poate folosi cu usurinta si intrarea analogica a placii de dezvoltare Arduino UNO R3 si stabilita o valoare intre 0-255 la care obstacolul sa fie considerat valid.

- In materie de aspect, robotul ar putea primi o carcasa exterioara care sa ascunda electronica si conexiunile acesteia, oferind si o protectie impotriva agentilor externi ce ar putea pune in pericol integritatea ansamblului.

**BIBLIOGRAFIE**

**[1] – Wikipedia/ ATmega328P Information -** [Wikipedia/ATmega328P](https://en.wikipedia.org/wiki/ATmega328) **[2] – Atmega328P Datasheet** - [ATmega328P Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf)  
**[3] – Arduino Boards – Uno -** [Arduino.cc](https://www.arduino.cc/en/main/arduinoBoardUno)  
**[4] – Carte Arduino Pentru incepatori – limba romana – Robofun**   
**[5] – Arduino Boards – Uno** **-** [Arduino.cc](https://www.arduino.cc/en/main/arduinoBoardUno)  
**[6] – Proximity Sensors Explained -** [Maxembedded.com](http://maxembedded.com/2013/08/how-to-build-an-ir-sensor/) **[7] – Infrared LED -** [Wikipedia](https://en.wikipedia.org/wiki/Proximity_sensor) **[8] – Sharp sensor Datasheet -** [Sharp Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\GP2Y0A21YK.pdf) **[9] – Corp negru -** [Wikipedia](https://ro.wikipedia.org/wiki/Corp_absolut_negru) **si Fizica II. Curs universitar -** [**Alexandrina Nenciu**](http://www.librarie.net/cautare-rezultate.php?au=32122)**,**[**Ruxandra Atasiei**](http://www.librarie.net/cautare-rezultate.php?au=32123)**– Editura Matrixrom  
[10] –** **Motor DC 130 RC -** [optimusdigital.ro](file:///C:\Users\nitut\AppData\Roaming\Microsoft\Word\optimusdigital.ro) **[11] – Pololu 130DC Motor -** [Pololu](https://www.pololu.com/product/1117) **[12] – Reduction Gears Explained -** [brighthubengineering.com](http://www.brighthubengineering.com/machine-design/47267-what-is-a-reduction-gear/) **[13] – 130DC Motor Datasheet -** [130 DC Motor Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\fa_130ra.pdf) **[14] – Adafruit Motor Shield -** [Adafruit](https://www.adafruit.com/product/81) **[15] – H-Bridge DC Motor Explained -** [Modularcircuits](http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/) **[16] – L293d Datasheet -** [L293d Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\L298_H_Bridge.pdf) **[17] – 76HC5959 Datasheet -** [76HC595 Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\sn74hc595.pdf) **[18] – Active Buzzer Explained -** [Openimpulse](https://www.openimpulse.com/blog/products-page/product-category/active-buzzer-module-and-cable/) **[19] – Active Buzzer Datasheet -** [Active Buzzer Datasheet](file:///D:\UNDA\Tutoriale%20&%20DIY\Minirobot%20detectie%20obstacole\Datasheet\ef532_ps.pdf) **[20] – Active Buzzer Pros/Cons -** [ArduinoModules](http://arduinomodules.info/ky-012-active-buzzer-module/) **si** [Engineersgarage](https://www.engineersgarage.com/insight/how-piezo-buzzer-works)  
**[21] – LED Diode -** [Wikipedia - LED Diode](https://en.wikipedia.org/wiki/Light-emitting_diode)  
**[22] – Introduction to LED Diodes –** [electronics-tutorials.com](http://www.electronics-tutorials.ws/diode/diode_8.html) **[23] – LED Diode Wavelength by Colors –** [radio-electronics.com](http://www.radio-electronics.com/info/data/semicond/leds-light-emitting-diodes/characteristics.php)  
**[24] – Arduino IDE Wiki -** [Arduino IDE](https://en.wikipedia.org/wiki/Arduino)