

Maria - Cristina CHIȚU (66962) - Obstacle Avoiding Robot

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

Introducere

Prezentarea pe scurt a proiectului:

Ce face?

Proiectul consta in realizarea unui robotel care va putea sa se plimbe singur printr-o incapere, fiind capabil sa evite obstacolele pe care le intampina.

Care este scopul lui?

Robotul este un dispozitiv inteligent care poate detecta si evita obstacolele aflate pe traiectoria sa de deplasare. Initial, robotul porneste sa se deplaseze in directia in care acesta a fost orientat, urmand sa isi schimbe traiectoria in momentul in care a sesizat un obstacol in drum. Va face dreapta si isi va continua drumul. Ledul verde se va aprinde cand va merge in fata, iar cel rosu cand va da cu spatele.

Care a fost ideea de la care am pornit?

Ideea de a construi acest proiect a pornit de la amintirea masinutelor teleghidate din copilarie si dorinta ca acestea sa fie capabile sa se deplaseze fara a provoca diverse accidente. Mi se pare un proiect foarte interesant, din care pot invata o multime de lucruri noi si aplica ceea ce am studiat in cadrul laboratorului.

Utilitatea proiectului

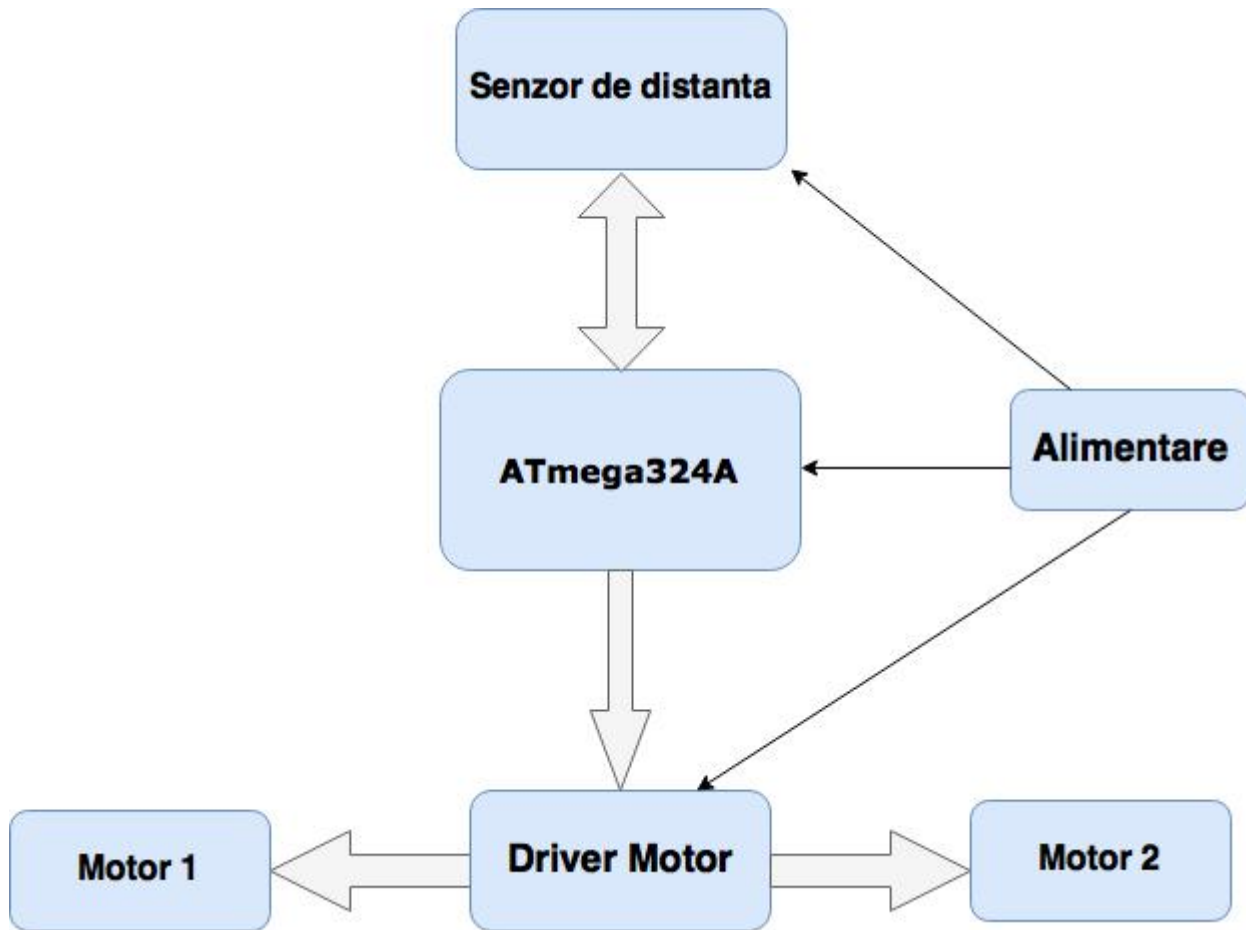
Proiectul poate fi util pentru construirea unor jucarii inteligente.

Dupa implementarea capabilitatii de cartografiere si optimizare a traseului parcurs, poate fi folosit:

- pentru construirea hartii unei incaperi
- pentru obtinerea unei ustensile automate de curatare, prin montarea pe un aspirator

Descriere generală

Schemă bloc



Miscarea robotului este comandata de microcontroler prin intermediul driver-ului de motoare, pe baza algoritmului implementat in memoria interna si a informatiilor primite de la senzorul de distanta. Driver-ul de motoare L298 este utilizat pentru controlul motoarelor de curent continuu prin comanda primita de la microcontroler. Senzorul de distanta detecteaza prezenta unor obiecte aflate la o distanta cuprinsa intre 10 cm si 80 cm. Toate modulele hardware necesita energie electrica furnizata de blocul de alimentare.

Robotul porneste in directia in care a fost positionat initial accelerand pana la viteza de croaziera. In momentul in care senzorul de distanta sesizeaza un obstacol aflat la o distanta de 50 de cm reduce viteza. Cand ajunge la 10 cm, se opreste si schimba directia cu 90 de grade stanga sau dreapta si continua deplasarea.

Hardware Design

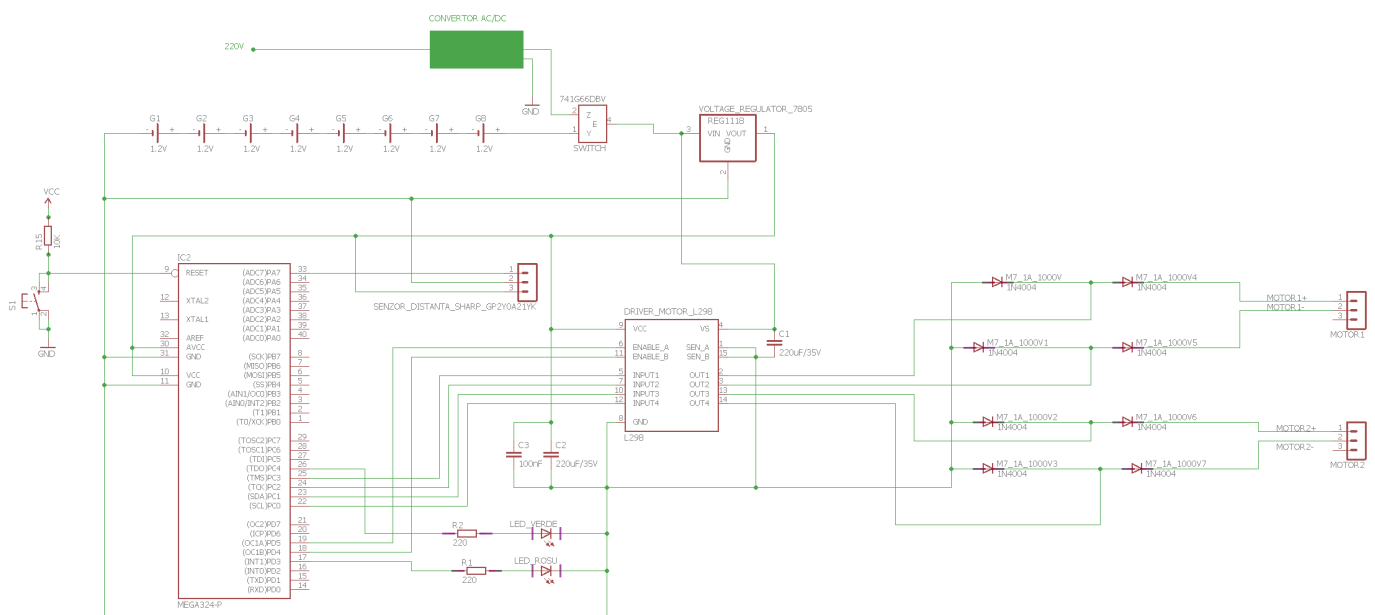
Lista de piese necesara in realizarea proiectului:

Nume	Cantitate	Distribuitor
Microcontroler ATmega324A	1	-
sensor distanta SHARP GP2Y0A21YK	1	RoboFun
driver motoare L298	1	OptimusDigital
motoare de curent continuu (MG-6-120)	2	RoboFun
convertor AC/DC	1	-
acumulatori	8	Emag

regulator de tensiune 7805	1	RoboFun
sasiu	1	RoboFun
LED rosu	1	OptimusDigital
LED verde	1	OptimusDigital
rezistentă 220	2	OptimusDigital
mufa mama alimentare DC	1	OptimusDigital
support acumulatori	2	RoboFun
cablu panglica 6 fire	1	RoboFun
fire conexiune mama-mama	10	RoboFun
mufa conectare baterie 9V	2	RoboFun
mufa mama 3 pini	1	RoboFun
regleta de pini tata	10	RoboFun
regleta de pini mama	3	RoboFun
comutator	1	OptimusDigital
incarcator acumulatori	1	-
roti motoare	2	RoboFun
o roata mobila	1	RoboFun

Pe langa acestea, se adauga componentele de baza ce au fost montate pe placa.

Schema electrica



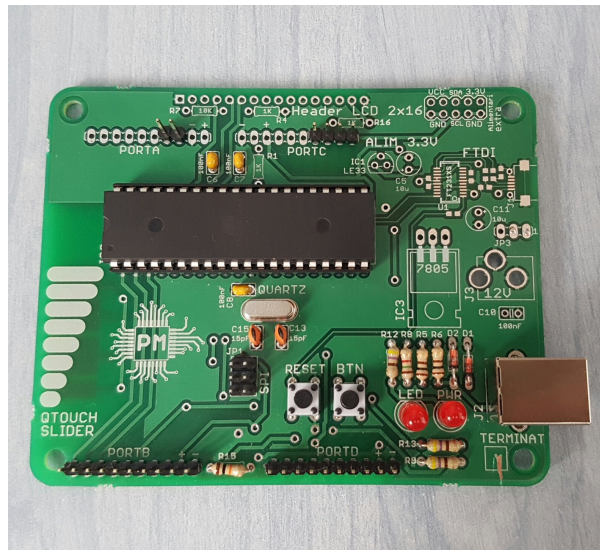
Explicatii suplimentare referitoare la schema:

- Intrarile ENABLE_A si ENABLE_B ale driver-ului de motor L298 trebuie sa fie conectate pe portul D, la unii dintre pinii PD7:4 (pentru a putea folosi "PWM mode timer function"). Cum pe PD7 avem user LED, iar pe PD6 user switch, ne raman disponibili pinii 5 si 4.
- Cele 4 intrari (1:4) din L298 trebuie comandate din 4 pini de iesire de la microcontroler. I-am ales pe primii 4 de pe portul C.
- L298 are nevoie de 2 tensiuni: una dintre ele tipic 5V, iar cealalta maxim 46V.

Tensiunea de 5V necesara si pentru alimentarea microcontrolerului se obtine din regulatorul de tensiune 7805 care are nevoie la intrare de o tensiune de minim 7V. Folosind acumulatori de 1.2V, aveam varianta de 6 sau 8 bucati. Am ales varianta cu 8.

- Switch-ul S3 permite intrerupere alimentarii la modulul cu driver-ul de motor.
- Am ales senzorul de distanta tip SHARP GP2Y0A21YK care permite masurarea distantei pana la un obstacol. Citirea informatiei referitoare la distanta este o marime analogica pe care o transmit microcontrolerului pe unul dintre pinii portului A, intrari de convertor analog-numeric.
- Simbolul de GND adaugat suplimentar pe schema nu suprascrie nimic, este pus doar pentru a se urmari circuitul cu usurinta.

Placa de baza a fost realizata si arata cf. figurii de mai jos:



Software Design

Mediul de dezvoltare

- Programmers Notepad [WinAvr] (avand posibilitatea sa compilez direct cu Tools→Make all)
- BootloaderHID

Biblioteci

- avr/io.h
- util/delay.h

Surse

- Exista in singur fisier: first_attempt_robot.c care implementeaza miscarea robotului.
- Makefile care este folosit pentru generarea fisierului .hex. S-a folosit avr-g++ pentru compilare si avr-objcopy pentru linkare.

Algoritm implementat

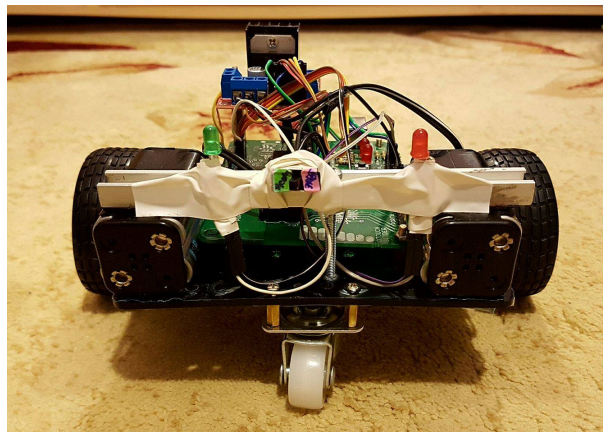
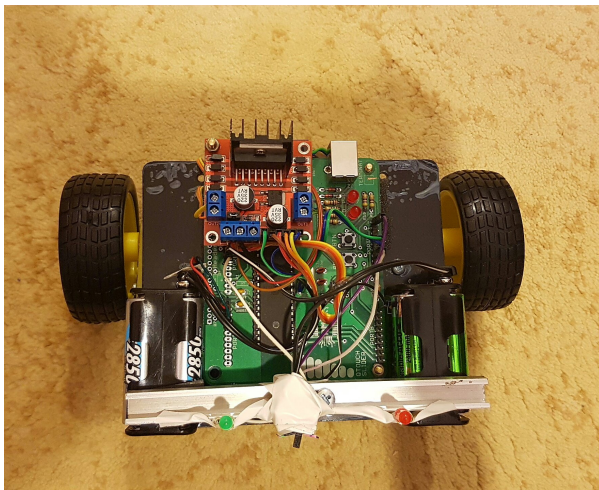
1. Se initializeaza ADC
2. Se initializeaza senzorul
3. Se initializeaza timer-ul
4. Se initializeaza motoarele
5. daca distanta preluata de la senzor \geq critical_DISTANCE
 1. astept putin
 2. verific daca distanța preluata de la senzor $<$ critical_DISTANCE
 - daca da \Rightarrow au fost doar zgomote interceptate, nu exista obstacol in fata \Rightarrow cresc viteza + merg inainte + aprinde LED verde
 3. altfel, aprind LED rosu + opresc motoare
 4. merg in spate putin
 5. orientez robotul catre dreapta
 6. revin la prima etapa de la pasul 5
6. altfel, cresc viteza + merg inainte + aprinde LED verde
7. revin la prima etapa de la pasul 5

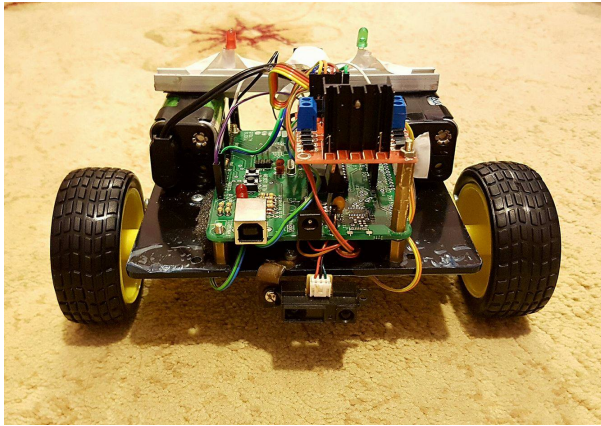
Rezultate Obținute

Robotelul merge in fata, in momentul in care detecteaza un obstacol, da cu spatele pentru a avea loc sa se roteasca, se orienteaza catre dreapta continuand sa mearga inainte pana cand intalneste din nou un obstacol.

Robotelul poate fi alimentat atata de la priza cat si din acumulatori, optiunea prin care dorim sa fie alimentat alegandu-se prin intermediul comutatorului montat.

Robotelul care se poate plimba si evita obstacolele este gata si arata cf. pozelor de mai jos:





Mai jos se afla in video ce demonstreaza faptul ca robotelul ocoleste obstacolele cf. algoritmului implementat.

The [Adobe Flash Plugin](#) is needed to display this content.


https://www.youtube.com/watch?v=AWbfj_cBoro

Concluzii

Consider ca acest proiect a fost una dintre cele mai interesante teme pe care le-am avut de facut in cadrul facultatii. Mi-a facut placere sa lucrez. A fost primul meu proiect hardware. Am invatat o multime de lucruri noi, acum fiind capabila sa realizez diferite scheme electrice, sa ma descurc cu datasheet-uri si sa leg componentele intre ele, cum sa le programez astfel incat sa functioneze, dar si cum sa fac debugging atunci cand ceva nu merge corespunzator.

Robotul functioneaza bine dat fiind sensorul ales. Este capabil sa se plimbe intr-o camera fara sa se loveasca de obstacole. Stie sa le detecteze, sa se intoarca putin in spate (pentru a nu se lovi cu roata stanga atunci cand ia curba) si sa schimbe directia de miscare (face dreapta si continua sa mearga in fata).

O imbunatatire a acestui proiect ar consta in alegerea unui alt tip de senzor sau sa folosesc mai multi senzori de la care sa se preia simultan date si sa se realizeze diverse comparatii intre ele astfel incat sa se stabileasca directia de miscare a robotului. Mergand pe una dintre aceste variante robotul ar fi capabil sa evite mai multe tipuri de obstacole, in momentul de fata acesta fiind capabil sa le detecteze doar pe cele care se afla in raza de acoperire a senzorului.

In concluzie sunt foarte multumita si mandra de produsul final obtinut. 

Download

Schema actualizata a proiectului se gaseste aici:
[chitu_maria_robot_schema_done.zip](#)

Fisierul sursa impreuna cu Makefile si README se gasesc in arhiva urmatoare:

[chitu_maria_robot_obstacole_pm.zip](#)

Schema electrica veche a proiectului:

[scheme_robot_obstacole_chitumaria.zip](#)

Jurnal

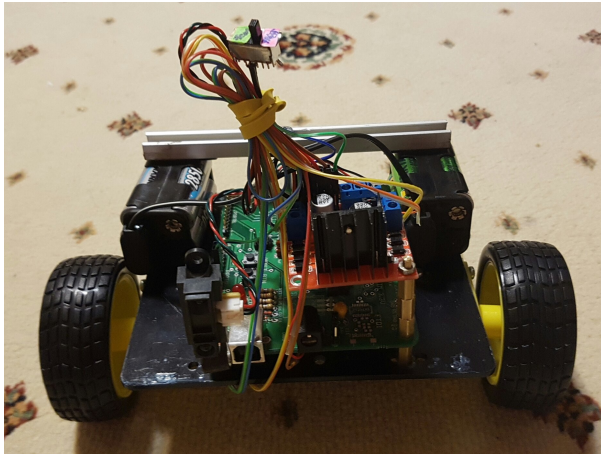
1. Placa de baza este gata si acum poate fi programata.
2. Am realizat schema electrica.
3. Am testat functionalitatea motoarelor. Am conectat unul dintre motoare direct la alimentarea de la priza si functioneaza corespunzator. Am conectat cele doua motoare la L298, iar pe acesta l-am conectat la priza ajung sa fie alimentat corect. Urmeaza sa scriu un program si sa il incarc pe placuta pentru a se invarti motoarele.
4. Am terminat de lipit restul componentelor de care am nevoie pe placa de baza. Am conectat componentele externe de placa prin firele necesare. Am asamblat intregul robot. Am prins placa de baza de sasiu impreuna cu driver-ul de motoare. Am conectat placa la prize. Pana aici totul bine.
5. Am scris in cod minimal pentru a testa acum daca pot porni motoarele. In momentul de fata robotelul meu stie sa mearga in fata pentru o anumita perioada de timp. Momentan acesta este inca alimentat de la priza.
6. Robotul merge in fata pana intr-un punct, apoi se intoarce cu spatele in pozitia initiala.
7. Am reusit sa controlez viteza robotelului. Acesta porneste cu o viteza si incetinesc pana cand viteza este zero, urmand apoi sa reporneasca pana la viteza maxima si sa scada inca o data pana se opreste definitiv.
8. Am reusit sa controlez motoarele astfel incat sa poata sa faca curba de 90 de grade la stanga sau la dreapta dupa care sa mearga drept inainte.

Robotul se poate deplasa in cele 4 directii.

The [Adobe Flash Plugin](#) is needed to display this content.

<https://www.youtube.com/watch?v=O3ZwEFPy5wc>

9. Am conectat senzorul de distanta la micro-controler.
10. Am montat acumulatorii pe robotel si functioneaza sa se deplaseze alimentandu-se de la ei. Acum va trebui sa schimb aranjarea componentelor pe sasiu deoarece nu este echilibrat si se rastoarna cand se deplaseaza.
11. Robotelul este echilibrat. Am adaugat un comutator pentru a putea alege daca dorim ca acesta sa fie alimentat de la priza sau din acumulatori. La momentul de fata, robotelul arata cf. fotografiei de mai jos:



12. Am reusit sa fac senzorul de distanta sa sesize obstacolele din fata lui. Am avut foarte multe probleme sa realizez acest lucru din cauza faptului ca pe laptop aveam o versiune mai noua a compilatorului decat cea de la facultatea.

13. Am reusit sa fac robotul sa se opreasca in momentul sesizarii unui obstacol urmand ca acesta sa isi schimbe orientarea si sa continue drumul. Din cauza senzorului folosit si a razei mici de acoperire pe care o are, robotul meu va putea detecta doar obstacolele care se afla fix in fata senzorului, la inaltimea si pe directia acestuia.

14. Am adaugat 2 LED-uri (unul rosu si unul verde). Cel rosu va fi aprins in momentul in care robotul meu da cu spatele, iar cel verde va sta aprins in timp ce robotul merge in fata.

15. Am rearanjat toate firele, le-am prins pe cat mai multe dintre ele pe sub placa de baza pentru un aspect cat mai dragut. Am izolat LED-urile si intrerupatorul si le-am prins de bara din spate pentru a sta fix. Robotul arata acum cf. pozelor postate in sectiunea anterioara.

16. Robotul functioneaza cf. asteptarilor. Stie sa se plimbe print-o incapere si sa evite obstacolele.

17. Schema electrica a fost modificata, s-au adaugat cele doua leduri care se aprind in functie de directia de miscare a robotului.

18. Se finalizeaza documentatia.

Bibliografie/Resurse

Resurse Hardware

Datasheet:

- [ATmega324A](#)
- [L298](#)
- [SHARP GP2Y0A21YK](#)
- [POSITIVE-VOLTAGE REGULATOR 7805](#)
- GEAR MOTOR MG-6-120 : <https://store.comet.bg/download-file.php?id=6844>
- Codul culorilor rezistente : <http://www.bobtech.ro/tutoriale/componente-electronice/45-codculori-calculator-online>

Resurse Software

- [laborator introductiv](#)
- [laborator Convertorul analog-digital](#)
- Tutorial despre cum functioneaza motoarele, cum se stabileste sensul de rotire
- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/amusat/1103>



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