

Ioana-Roxana RADU (66826) - MinicarController

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

Introducere

Proiectul constă în realizarea unei telecomenzi ce va transmite date prin Bluetooth. În funcție de mișcare/inclinație se vor transmite anumite comenzi prin Bluetooth, aceste comenzi putând fi mai departe recepțate de către o mașinuță. De asemenea, la apăsarea unor butoane se pot transmite comenzi ce determină aprinderea farurilor/declanșare sunet claxon. Scopul final al proiectului este comunicarea cu o mașinuță care primește comenzi prin Bluetooth.

Am ales acest proiect pentru ca mi se pare interesant să poți controla o mașinuță în funcție de mișcarea/inclinația telecomenzii. Voi stabili un protocol de comunicație împreună cu un coleg care va implementa o [mașinuță ce primește comenzi prin Bluetooth](#).

Descriere generală

Microcontrollerul va comunica cu modulul de Bluetooth prin interfața USART. Se vor primi date de la giroscop/accelerometru/butoane și în funcție de acestea se vor transmite mai departe comenzi: mers în fata/spate/stânga/dreapta/declanșare claxon/aprindere faruri. Pentru început, se vor transmite date utilizând informațiile de la accelerometru. Ca și componenta hardware voi folosi un giroscop cu accelerometru și în funcție de timpul rămas voi încerca să realizez o prelucrare a datelor de la giroscop și transmiterea unor comenzi mașinuței care să îi determine o anumită mișcare (de exemplu schimbarea direcției de mers).

Cum funcționează?

În funcție de inclinație față/spate, stânga/dreapta se vor transmite comenzi care să determine mișcarea mașinuței în mod corespunzător. De asemenea se vor aprinde ledurile corespunzătoare pentru a indica direcția transmisă. În plus, la apăsarea butonului se va declanșa claxonul (sunetul se va menține cât timp butonul este apăsat și este oprit când butonul este eliberat). Aprinderea/stingerea farurilor se va face în funcție de rotația telecomenzii pe axa Ox.

Schemă bloc



Hardware Design

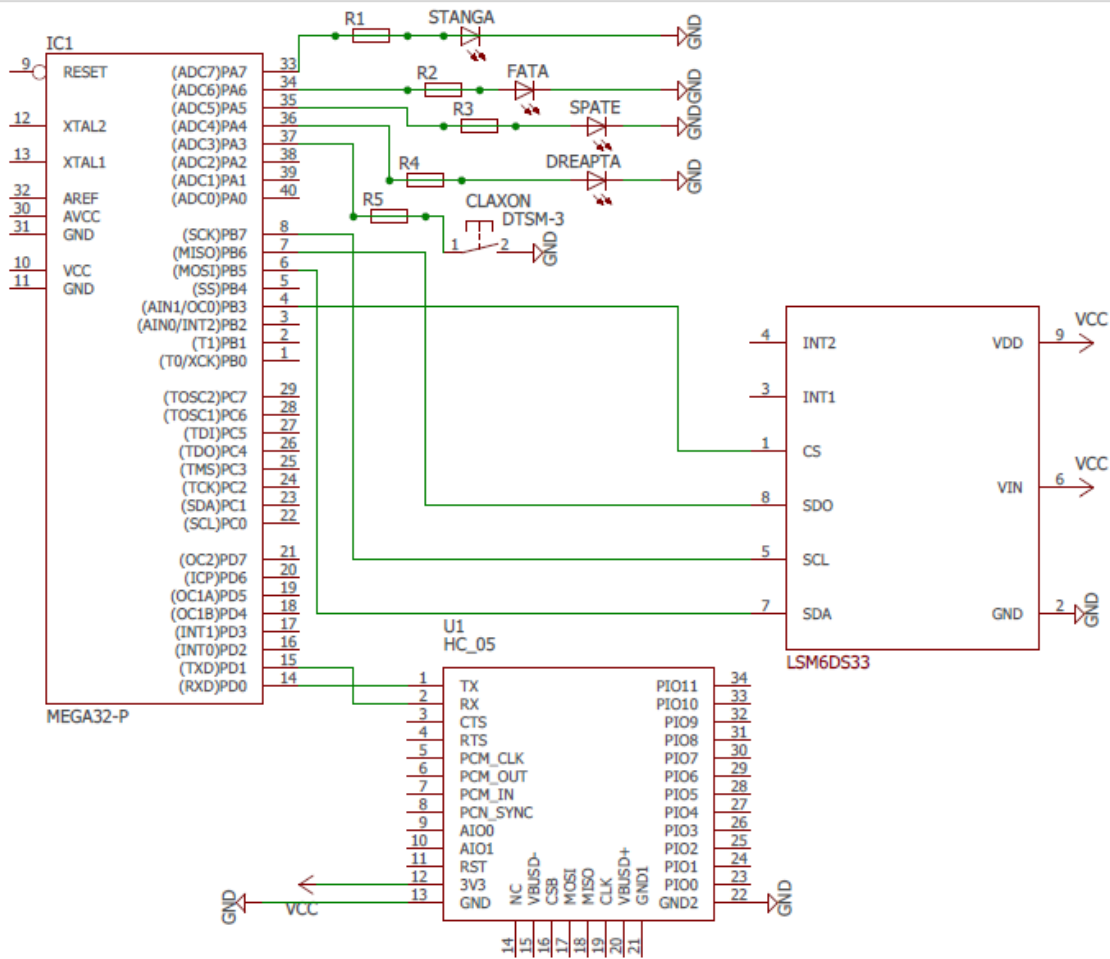
Lista de piese	Preț (aproximativ)
Placa de baza cu componentele aferente	32 lei
Giroscop cu accelerometru (Pololu LSM6DS33)	68 lei
Modul Bluetooth (HC-05)	27 lei
4 leduri	2 lei
Buton	1 leu
Fire mama-mama	10 lei
Regulator tensiune 5V - 3.3V (LE33)	2 lei
O cutie din plastic	6 lei
Rezistențe pentru leduri	extra în pachetul de bază
Total	148 lei

Majoritatea componentelor le-am cumpărat de la [Optimus Digital](#)

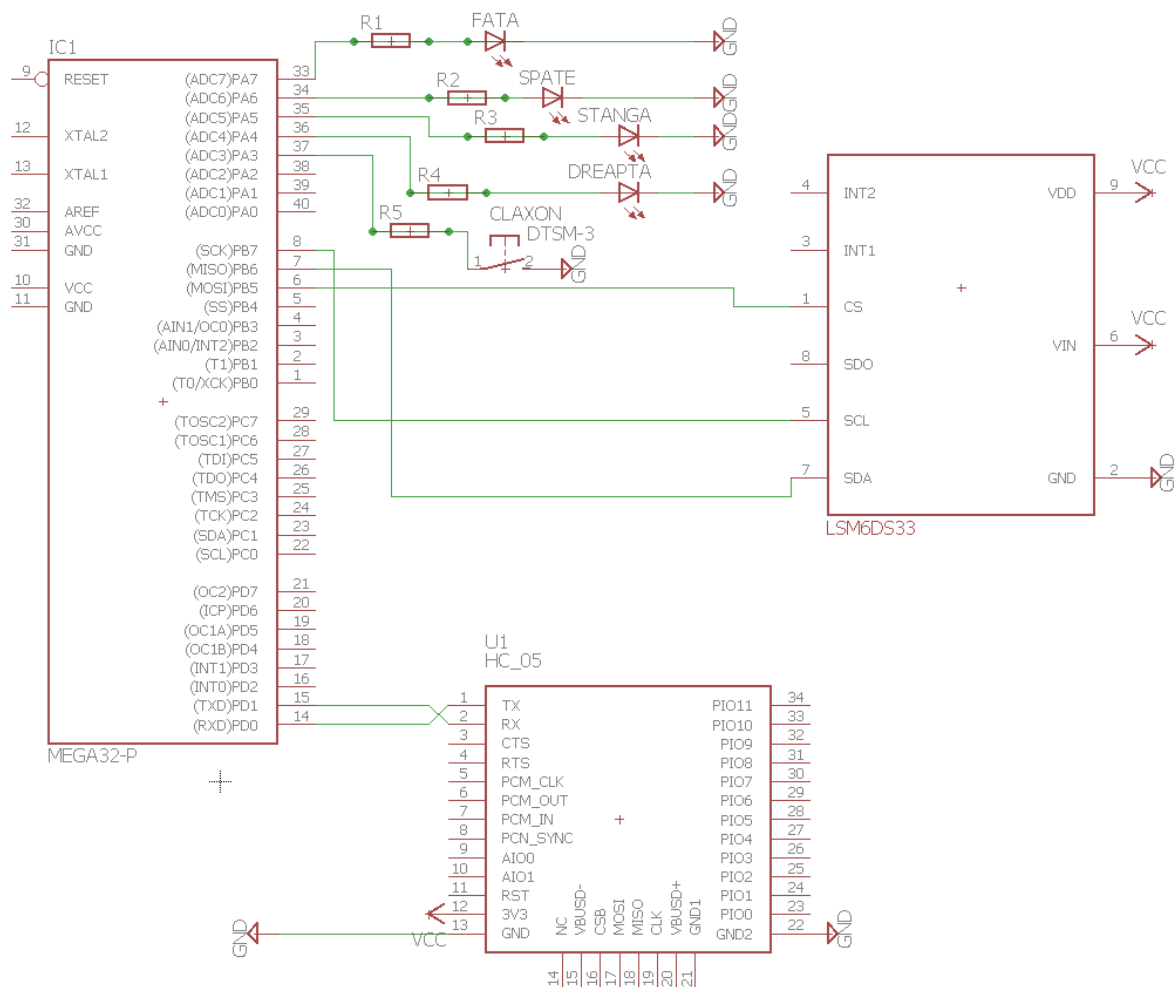
Schema electrica

[\[Versiunea Finală\]](#)

Am făcut câteva modificări la schema electrică. Am legat câteva fire diferit de cum mi-am propus inițial



[Versiunea Inițială]



[Versiunea1] [telecomanda.sch](#)

[Versiunea2] [telecomanda_schema_electricav2.sch](#)

Software Design

Mediu de dezvoltare

Pentru dezvoltarea software am utilizat Programmer's Notepad și WinAVR.

Interfațarea cu hardware-ul

- **Giroscop/Accelerometru:** Pentru a înțelege cum se primesc datele am studiat [datasheet-ul](#). Comunicația cu acest dispozitiv se realizează prin SPI, urmărind protocolul descris în datasheet la capitolul aferent (capitolul 6.2). De asemenea, am folosit ca model biblioteca pentru Arduino disponibilă pe [github](#). Pentru configurațiile inițiale necesare a trebuit să scriu regiștrii corespunzătorii activării accelerometrului și giroscopului. Similar, la citire am preluat date din

regiștrii corespunzători.

- **Modulul de Bluetooth:** deoarece a fost necesar doar să trimit date, i-am conectat doar cu RX-ul la placă și astfel am trimis date prin USART, similar cu laboratorul. El funcționează în modul master și doar transmite date.
- **Butonul și ledurile:** pin-ul corespunzător este setat ca intrare, respectiv ieșire.

Structura codului

Pentru a implementa funcțiile specifice pentru fiecare dintre USART, SPI și LSM6 exista fișiere separate. Mai exista un fișier ce conține implementarea "main".

- pentru USART fișierele conțin funcții necesare comunicării cu seriala: inițializare, trimitere caracter/string. (nu a fost necesară recepția)
- pentru SPI fișierele conțin funcții pentru transmisia unui octet prin SPI.
- pentru giroscop fișierele conțin funcții de citire/scriere a unui registru din giroscop, de inițializare și de citire a datelor atât de la accelerometru cât și de la giroscop.


Interpretarea datelor/Transmiterea comenzilor

În funcțiile de date primite de la accelerometru se determina orientarea.

1. Dacă s-a detectat înclinație stânga/dreapta se trimite comandă corespunzătoare virajului corespunzător și se aprinde ledul corespunzător.
2. Dacă s-a detectat înclinație față/spate se trimite comandă corespunzătoare mersului înainte/înapoi și se aprinde ledul corespunzător.
3. Dacă s-a detectat rotația dispozitivului/telecomenzii pe orizontală se trimite comandă corespunzătoare schimbării stării farurilor (toggle).

Pentru a intercepta claxonul am activat întreruperea corespunzătoare butonului. În rutina de tratare a întreruperii, se determină starea butonului. Dacă a fost apăsat și acum este liber se trimite comandă corespunzătoare opririi. Altfel, se trimite semnal de declanșare a claxonului. (Am adăugat și un delay deoarece la o singură apăsare rutina de tratare a întreruperii era apelată de 4-5 ori, deși ar fi trebuit o singură dată).

Cum am interpretat datele?

M-a ajutat modulul de Bluetooth. Am folosit o aplicație Android pe telefon pentru a mă conecta la modulul de Bluetooth și a primi date. (ce trimiteam prin USART de pe placă primeam pe telefon ). Astfel, am stabilit valorile de referință pentru accelerometru, precum și un prag care dacă este depășit se iau în considerare datele primite și pe baza acestora se iau decizii de transmitere a comenzilor corespunzătoare. Pentru a elimina "șocurile" care apar inevitabil deoarece dispozitivul este ținut în mână și pentru a nu transmite comenzi incorecte realizez 2000 de citiri de la accelerometru, la interval de 5 us și după analiza valorilor obținute se ia decizia. Similar, datele de la

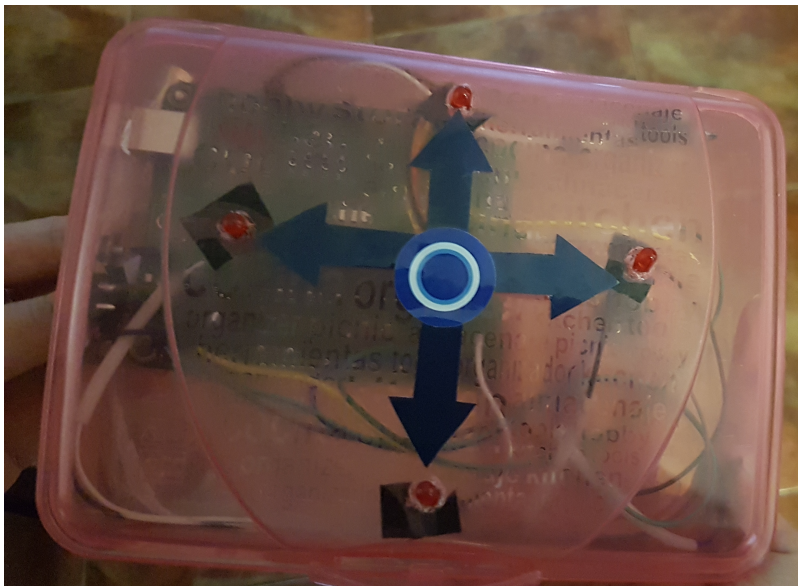
giroscop le citesc la o secundă (am folosit un timer), deoarece la o singură rotație pe orizontală giroscopul va furniza de mai multe ori valori care depășesc pragul stabilit.

Cum am stabilit codificarea comenzilor?

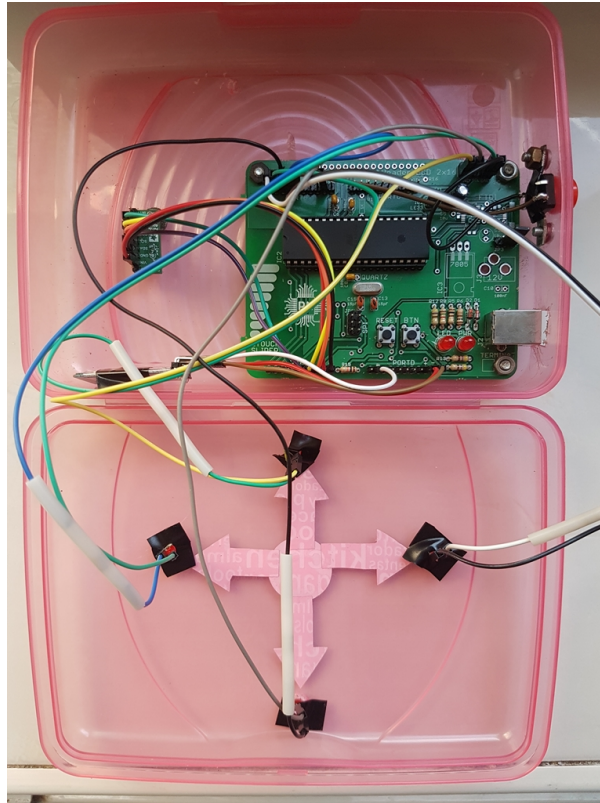
După cum am precizat inițial, protocolul de comunicație (modul de codificare a comenzilor) l-am stabilit împreună cu un coleg care și-a implementat [proiectul](#) astfel încât să primească comenzi de la Bluetooth (tot prin USART) și să le proceseze pentru a i le trimite mai departe mașinuței.

Rezultate Obținute

Am obținut o “telecomandă” `<fc #FF00FF>roz</fc>` care controlează o mașinuță și indică direcția/sensul de deplasare prin aprinderea ledurilor corespunzătoare.







Concluzii

Pentru mine a fost primul proiect care a presupus interacțiune atât cu partea hardware cât și cu cea software. Mi-a fost greu la început să înțeleg protocolul de comunicație prin SPI cu accelerometrul, să îl configurez și să îi citesc datele, dar cu ajutorul resurselor disponibile (datasheet-ul) m-am descurcat. Faptul că am început să lucrez din timp la proiect, urmărind milestone-urile, m-a ajutat mult. Mă bucur că am reușit să realizez ce mi-am propus și sunt foarte mulțumită de rezultatele obținute.

Download

Arhiva ce conține codul sursă este disponibilă [aici](#)

Jurnal

- **[27.04]** Am terminat de lipit piesele pe placă.
- **[Săptămâna 12: 10-12 mai]** Am început partea de hardware și puțin software. Nu am avut însă modulul de Bluetooth HC-05.
- **[Săptămâna 13]** Am continuat implementarea software și definitivat partea hardware (toate componentele conectate la placă)
- **[21.05]** Am făcut câteva modificări la schema electrică. Noua versiune este cea finală.
- **[22.05]** Am finalizat și partea software.
- **[23.05-25.05]** Testarea funcționalității.

Bibliografie/Resurse

Resurse Software

- [Biblioteca LSM6DS33 pentru Arduino](#)
- Laboratoarele 0, 1, 2, 4

Resurse Hardware

- [Datasheet LSM6DS33](#)
- [Application Note LSM6DS33](#)
- [Datasheet ATmega324A](#)
- Documentația în format [PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2017/tvisan/telecomanda>



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