

# George-Sebastian PÎRTOACĂ (66847) - Fast and Furious

Autorul poate fi contactat la adresa: [gpirtoaca@gmail.com](mailto:gpirtoaca@gmail.com)

## Introducere

Proiectul are ca scop realizarea unei masinute cu telecomanda ce detecteaza obstacolele frontale si nu permite accidente in acest sens. De asemenea, masinuta va dispune si de un claxon pentru atentionarea diferitelor situatii in trafic 🚗. Controlul masinii se va face prin inclinarea telecomenzii in anumite unghiuri si pe anumite directii (alese in mod natural, pentru un control cat mai facil). De asemenea, restul comenzilor catre masinuta (precum claxon) se vor da tot cu ajutorul unor butoane plasate pe telecomanda. Comunicatia intre telecomanda si masinuta se va realiza prin tehnologia Bluetooth. Ideea acestui proiect provine din seria de filme de actiune [The Fast and the Furious](#) a carei ultima pelicula abia a fost lansata in cinematografe. Proiectul este util pentru a invata cum se interfeateaza diferitele componente pe care le voi utiliza (precum modulul de Bluetooth) cu un microcontroler dar si pentru a-mi cultiva pasiunea pentru masini 🚗

## Descriere generală

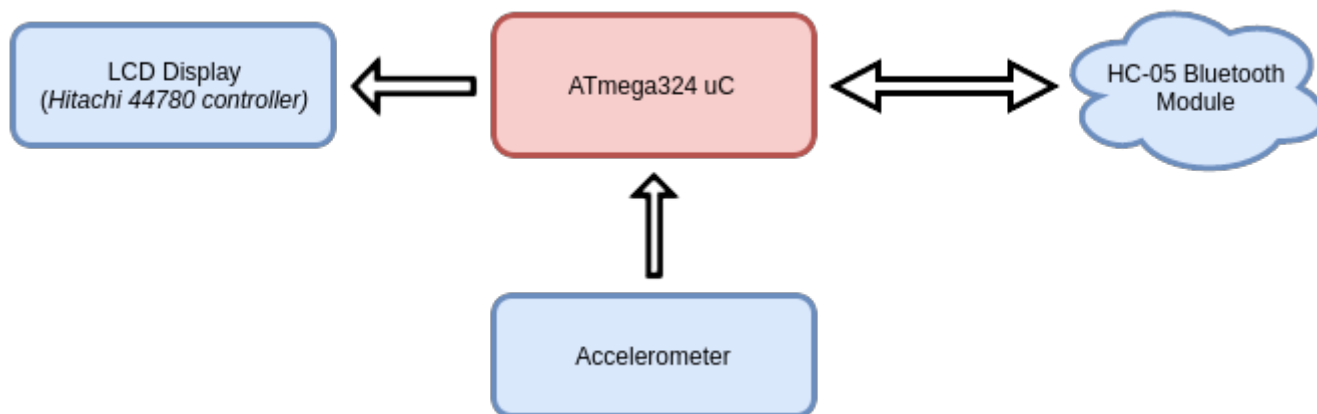


Fig. 1: Schema bloc pentru telecomanda

Comunicatia intre microcontroler si modulul de bluetooth se face prin interfata seriala USART. Modulul de Bluetooth lucreaza in modul master si transmite comenzi catre modulul de Bluetooth de pe masinuta (care, implicit, va fi slave). Periodic, uC citește datele de la accelerometru, folosind pini conectati la convertorul analogic-digital, pentru a sesiza eventualele schimbari si pentru a transmite comenzile corespunzatoare catre microcontrolerul de pe masinuta. Microcontrolerul va lua decizii pe baza a doua unghiuri pe care acesta le poate masura (unghiul din planul XoY nu poate fi masurat doar

pe baza accelerometrului, dar doua unghiuri sunt suficiente). Pentru mai multe detalii se pot consulta: [Unghiurile lui Euler](#) si [Axele unui avion](#). Unul dintre aceste unghiuri va da acceleratia masinii iar celalat directia. Date importante vor fi afisate pe LCD (in functie si de timpul extra de care dispun pentru implementarea proiectului pot sa adaug anumite feature-uri bonus precum: masurarea vitezei masinii, muzica la bord).

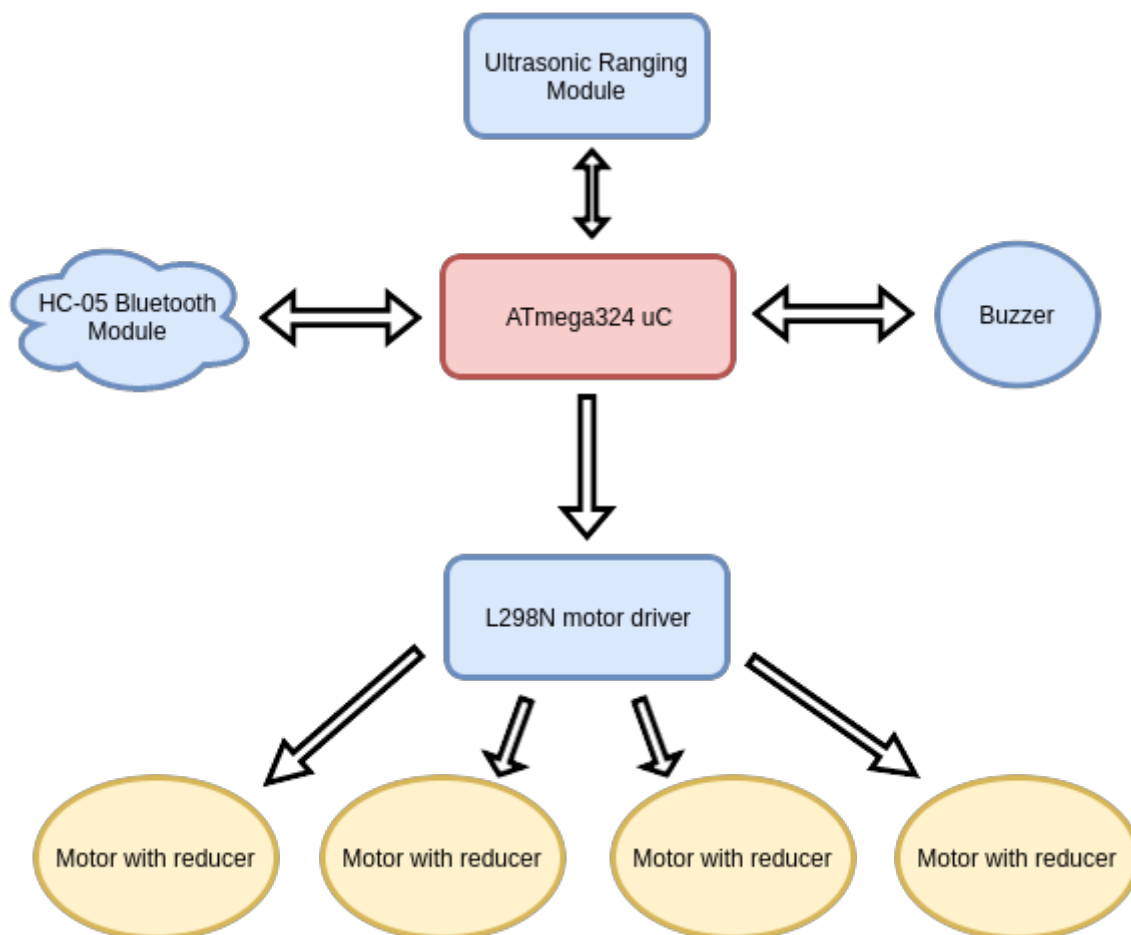


Fig. 2: Schema bloc pentru masina

Masinuta va avea 4 motoare (2 fata si 2 spate) - 4x4 🤪 iar directia se va modifica prin schimbarea vitezei de rotatie a rotilor de pe o parte sau alta a masinii. Pentru a asigura curentul necesar consumat de cele 4 motoare am ales sa folosesc driverul L298N. Acesta trebuie alimentat la o tensiune mai mare sau egala cu 9V (eu am ales sa folosesc 12V) si poate fi controlat folosind PWM pentru a modifica turatia motoarelor. Modulul de Bluetooth lucreaza in mod slave si asteapta date de la telecomanda. Microcontrolerul discuta cu un sensor de distanta (care va fi amplasat frontal) pentru a putea evita eventualele coliziuni ale masinutei (cand se detecteaza obiecte la o anumita distanta, aproximativ 20cm, masinuta nu va mai inainta). Buzzer-ul este folosit pentru implementarea claxonului.

## Hardware Design

Lista de piese:

Nume	Distribuitor
Placa de baza PM 2017	Echipa PM 🤖
Modul Bluetooth HC-05	<a href="#">Optimus Digital</a>
Motor cu reductor, cuplu 0.8 kg * cm	<a href="#">Optimus Digital</a>
Buzzer	<a href="#">Optimus Digital</a>
Driver motoare L298N	<a href="#">Optimus Digital</a>
Senzor ultrasonic HC-SR04	<a href="#">Optimus Digital</a>
LCD 16x2	<a href="#">Robofun</a>
Sasiu si roti	<a href="#">Optimus Digital</a>
Fire mama-mama	<a href="#">Optimus Digital</a>
Fire mama-tata	<a href="#">Optimus Digital</a>
Rezistente	<a href="#">Optimus Digital</a>
Condensatoare	<a href="#">Optimus Digital</a>
Acumulatori	<a href="#">Optimus Digital</a>
Modul Accelerometru Digital MMA8452	<a href="#">Optimus Digital</a>

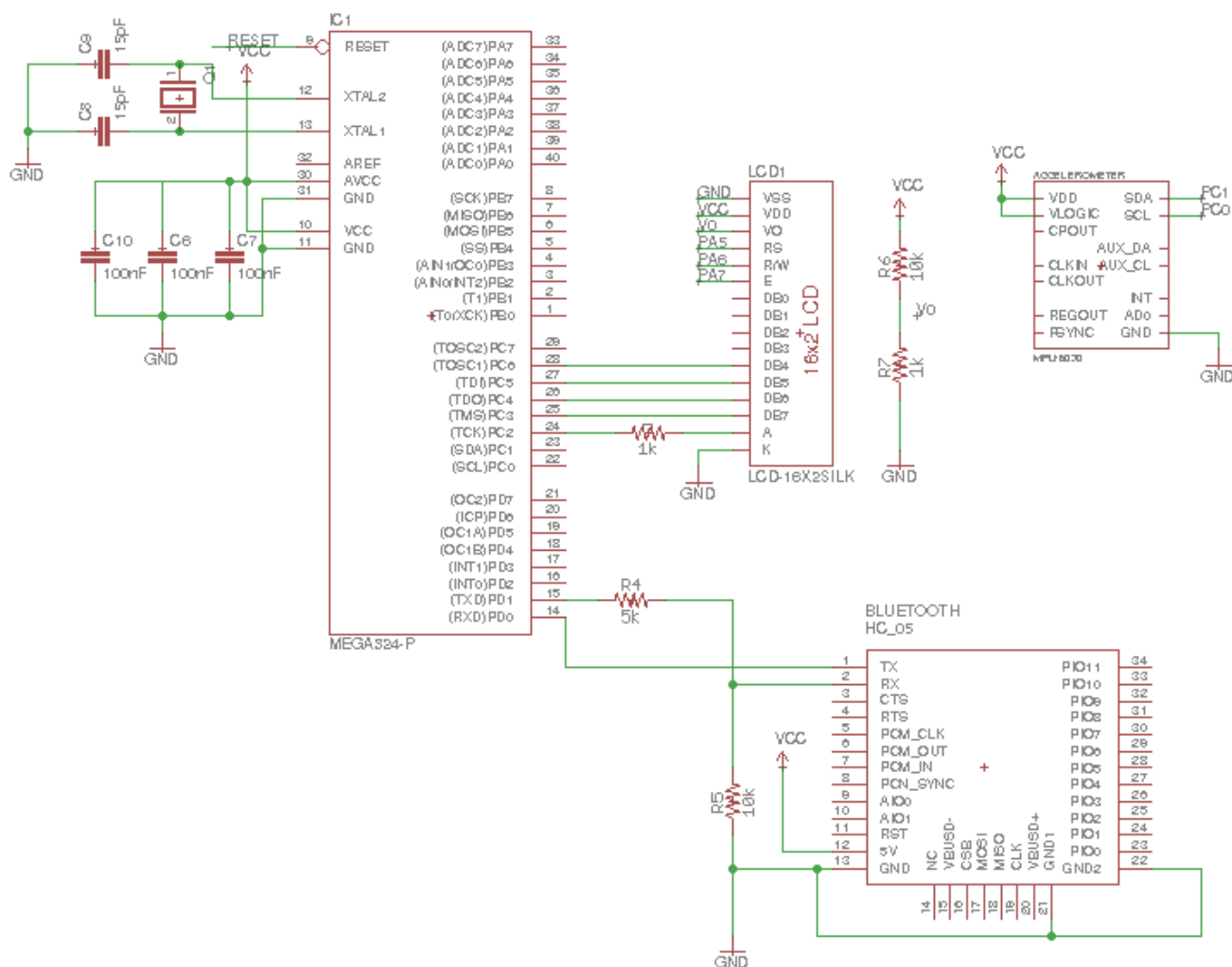


Fig. 3: Schema electrica pentru telecomanda

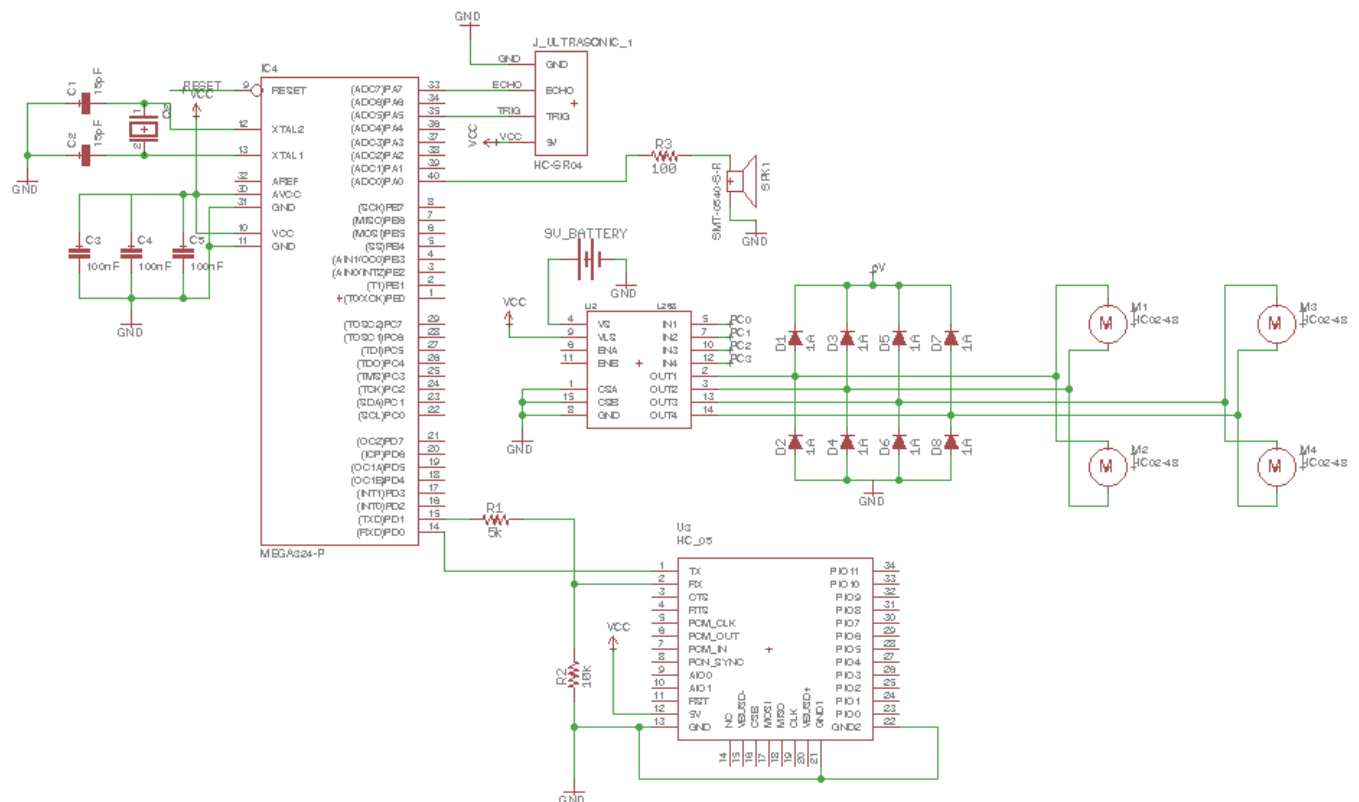


Fig. 4: Schema electrica pentru masina

## Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2017:avoinescu:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

- Documentația în format [PDF](#)

From:

<http://cs.curs.pub.ro/wiki/pm/> - **PM Wiki**

Permanent link:

[http://cs.curs.pub.ro/wiki/pm/prj2017/anitu/masinuta\\_pirtoaca\\_george\\_sebastian\\_335cb\\_2017](http://cs.curs.pub.ro/wiki/pm/prj2017/anitu/masinuta_pirtoaca_george_sebastian_335cb_2017)

Last update: **2017/05/13 23:41**