

# Speeding Car

## Introducere

Proiectul consta intr-un speedometru ce va intoarce viteza actuala si poate fi setata o anumita viteza pe care daca o depaseste va opri motorul unei masini si va afisa un mesaj de eroare pe display timp de 5 secunde. Acesta il voi monta pe o masina ce functioneaza din keyoad si poate merge doar in fata. Acesta are rolul de a fi montat pe masini ca un semnal de alarma in cazul depasiri vitezei.

## Descriere generală

Pe display cu ajutorul modului gps vom afisa viteza. Acestea vor fi montate pe masina astfel incat viteza afisata sa fie aceea a masinutei. Vom putea pune o limita de viteza astfel pe display va aparea si limita si viteza actuala. Daca limita este depasita atunci se va afisa pe display un mesaj de avertizare si masina va fi oprita 5 secunde. Voi folosi un keypad cu ajutorul caruia se va selecta modul de limitare al vitezei, superior sau inferior, apoi se va introduce viteza care nu se vrea depasita sau scazuta. Totodata cu ajutorul sau poti selecta ce unitate de masura vrei cum ar fi km/h, m/s etc si poti porni si deplasarea masini

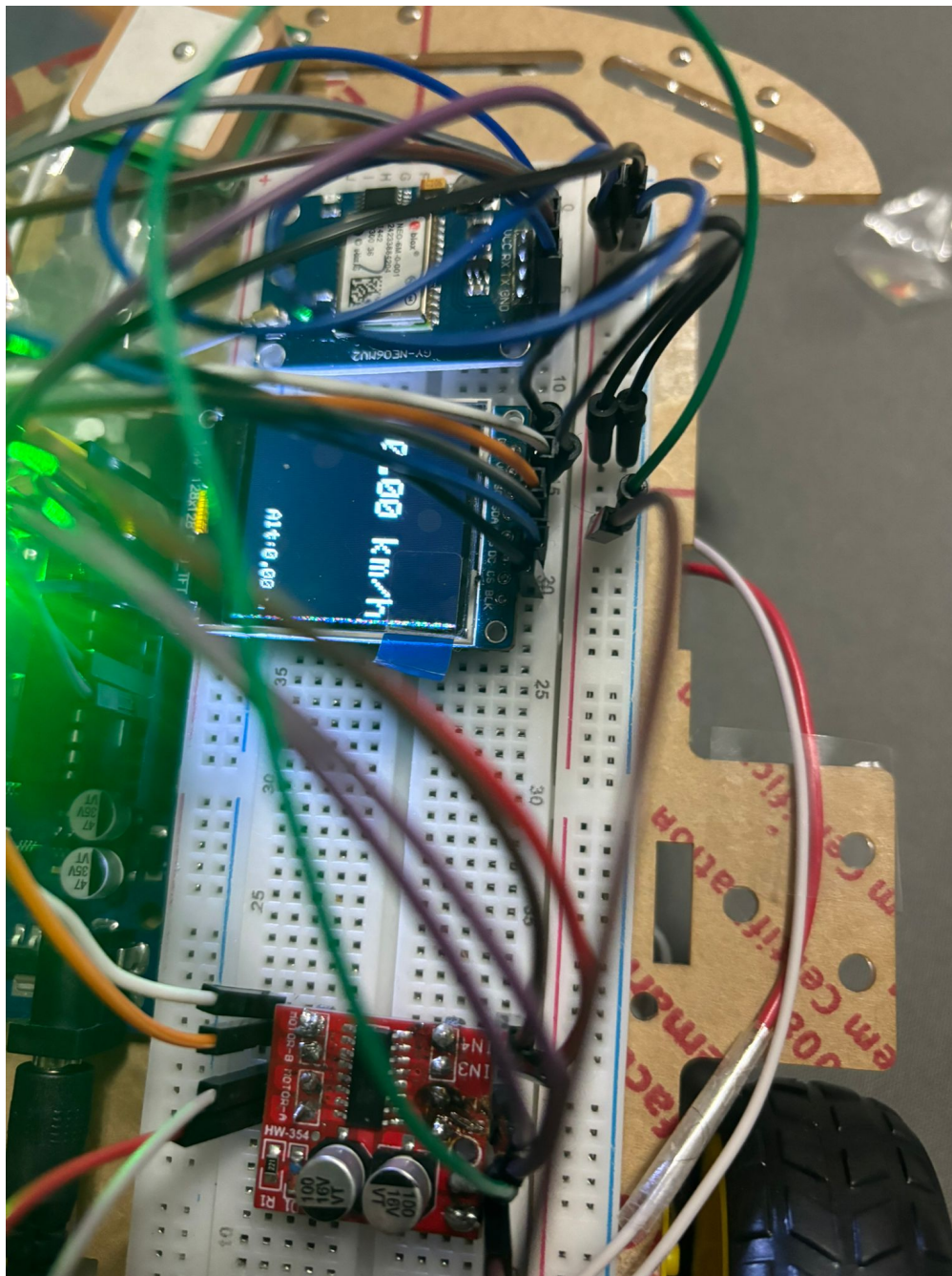


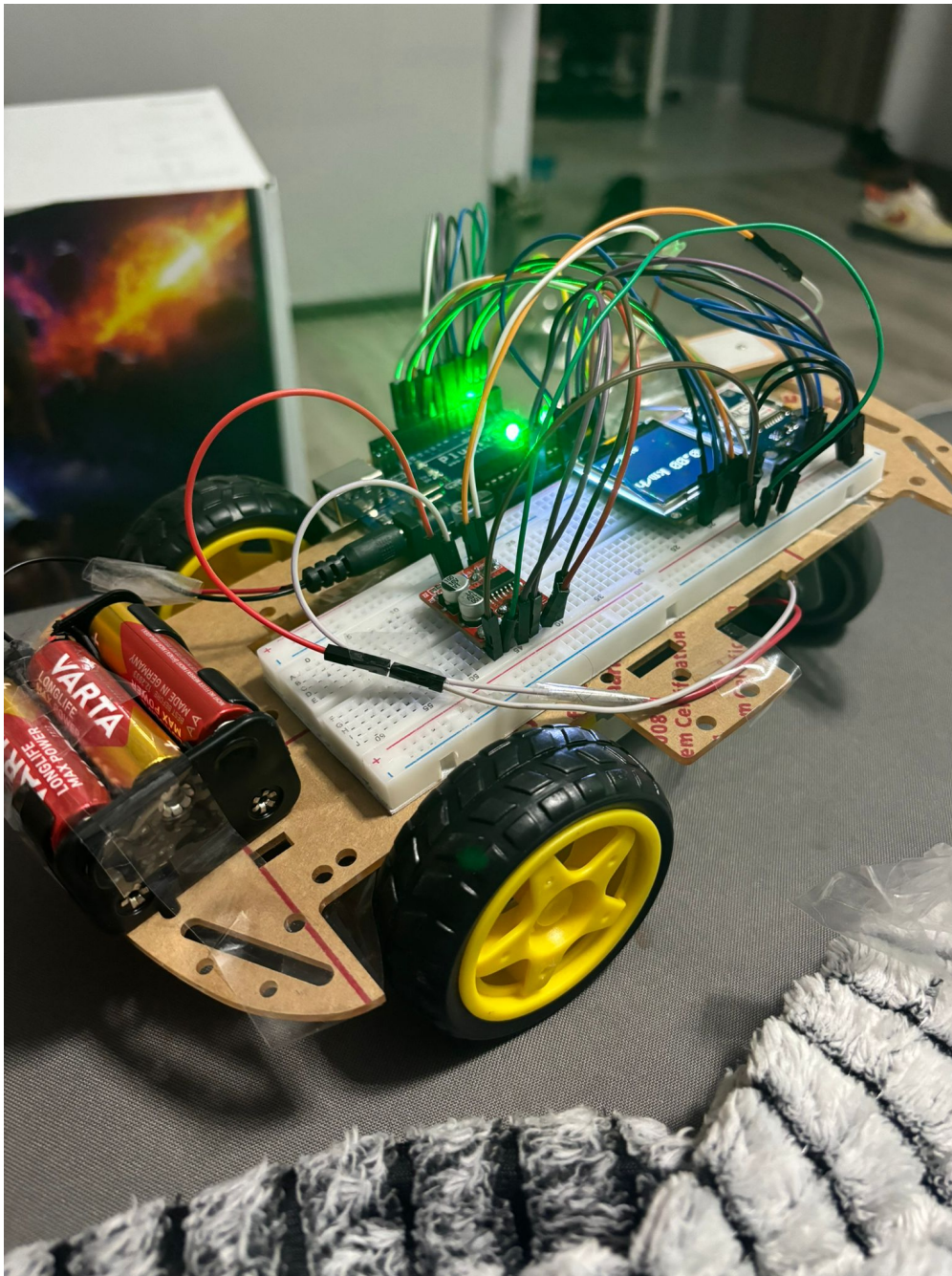
## Hardware Design

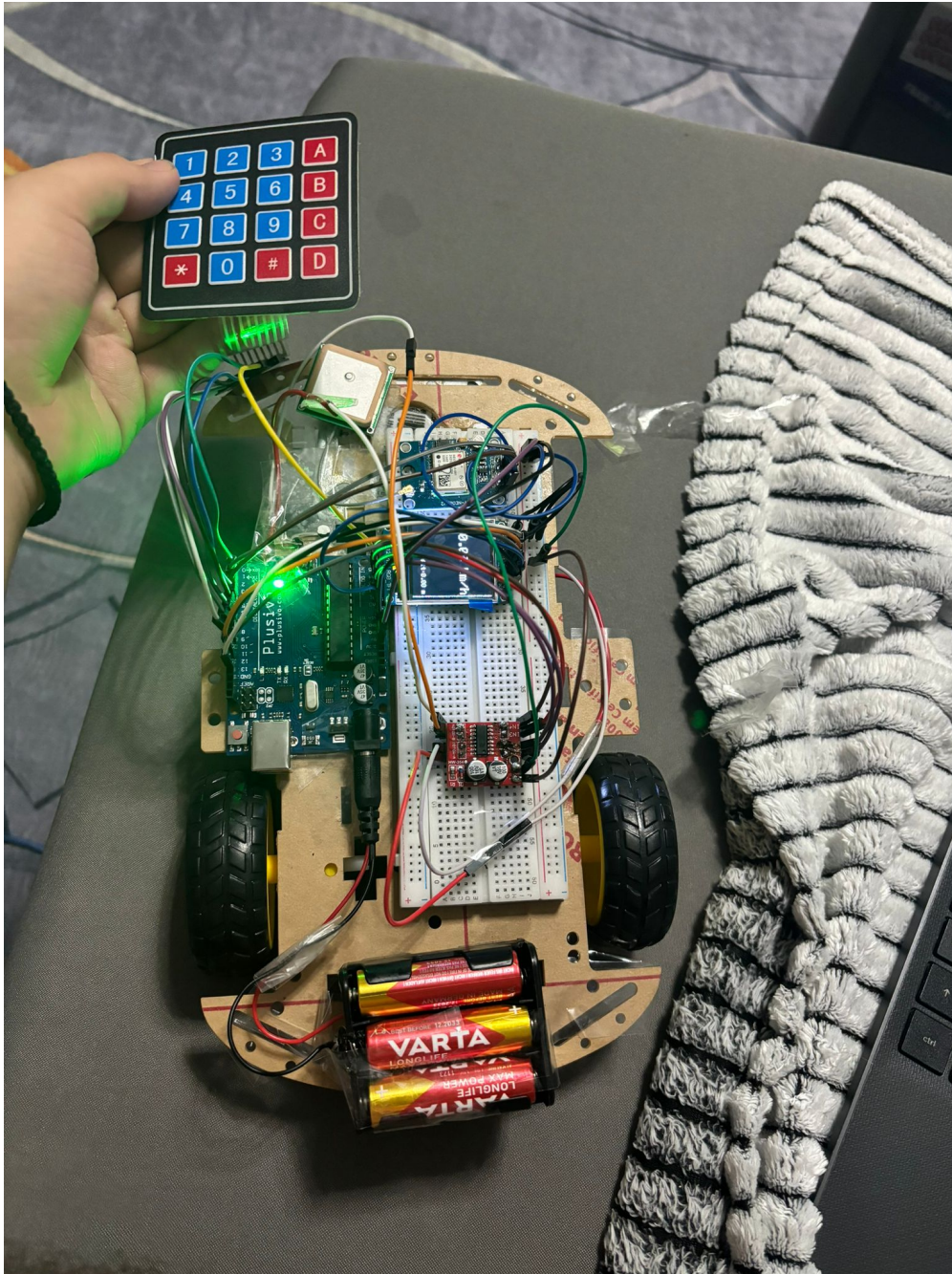
Lista piese:

- arduino uno
- doua motoare
- un drive modul Mx1508
- Gps-module
- modul lcd ST7735
- keypad
- breadboard









## Software Design

Mediu de dezvoltare: Arduin Ide

Biblioteci suplimentare pe care le-am folosit:

- TinyGPSPlus
- SoftwareSerial
- Adafruit\_GFX
- Adafruit\_ST7735
- Keypad

Setup:

- initializez displayul, pini de in al modului si modulul Gps

```
void setup() {
```

```
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
Serial.begin(9600);
ss.begin(9600);
tft.initR(INITR_BLACKTAB);
tft.fillScreen(ST7735_BLACK);
tft.setTextColor(ST7735_WHITE);
}
```

Loop:

- Se verifica ce tasta este apasata daca este apasata tasta pentru pornirea motorului, pentru a accesa meniul de selectie a limitari vitezei sau cel de unitati. Se verifica daca limitarea vitezei este pusa. Prin apasarea tastei 1 se sterge limitarea. Daca este selectata limitarea inferioara apare mesajul de a creste viteza pana o depaseste. Se verifica daca este pusa o limita superiora daca cumva s-a depasit limita vitezei si daca da se opresc motoarele timp de 5 sec. Se citesc datele primite de la gps. Cu ajutorul bibliotecii intoarcem altitudine si speedul in km/h m/s.

```
void loop() {
```

```
char key = keypad.getKey();
unsigned long currentMillis = millis();
if(key)
{
  if(key == '1' && tip != 0 && !speedmode && !option && mod == 0){
    tft.fillRect(0, 110, 70, 10, ST7735_BLACK);
    tft.fillRect(0, 100, 70, 10, ST7735_BLACK);
    tft.fillScreen(ST7735_BLACK);
    prevspeed = -1;
    prevAlt = -1;
    tip = 0;
    speeddisplay = true;
  }
  if(key == '*')
  {
    pinstate = !pinstate;
  }
  if(key == '#')
  {
    speeddisplay = !speeddisplay;
    if(speeddisplay){
      option = false;
    }else{
      option = true;
    }
  }
}
```

```
    mod = 0;
    speedmode = false;
    tft.fillScreen(ST7735_BLACK);
    prevspeed = -1;
    prevAlt = -1;
}
}
while (ss.available() > 0)
{
    gps.encode(ss.read());
}
float currspeed = 0;
if(unit == 1){
    currspeed = gps.speed.isValid() ? gps.speed.kmph() : 0.0;
}else{
    currspeed = gps.speed.isValid() ? gps.speed.mps() : 0.0;
}
float currAlt = gps.altitude.isValid() ? gps.altitude.meters() : 0.0;
if(speeddisplay){
    if (currspeed != prevspeed) {
        tft.fillRect(20, 40, 128, 20, ST7735_BLACK);
        tft.setTextSize(2);
        tft.setCursor(20, 40);
        tft.print(currspeed);
        if(unit == 1){
            tft.print(" km/h");
        }else {
            tft.print(" m/s");
        }
        prevspeed = currspeed;
    }
}
```

```
if (currAlt != prevAlt) {
    tft.fillRect(70, 110, 128, 10, ST7735_BLACK);
    tft.setTextSize(1);
    tft.setCursor(65, 110);
    tft.print("Alt:");
    tft.print(currAlt);
    tft.print(" m");
    prevAlt = currAlt;
}
if(tip != 0){
    tft.setTextSize(1);
    tft.setCursor(5, 110);
    tft.setTextColor(ST7735_WHITE);
    tft.print(limitspeed);
    if(unit == 1){
        tft.print(" km/h");
    }else {
        tft.print(" m/s");
    }
}
```

```
tft.setCursor(5, 100);
tft.print("Limit:");
}
}
if(speedmode){
tft.setTextSize(1);
tft.setCursor(5, 50);
tft.print("Introdu viteza");
tft.setCursor(10, 70);
tft.print(limitspeed);
if (key && key >= '0' && key <= '9') {
int number = key - '0';
limitspeed = limitspeed * 10 + number;
tft.fillRect(10, 70, 50, 10, ST7735_BLACK);
tft.setCursor(10, 70);
tft.print(limitspeed);
}
}
if (tip == 2 && limitspeed > gps.speed.kmph() && !option && !speedmode) {
tft.fillRect(ST7735_RED);
tft.setTextSize(1);
tft.setCursor(20, 50);
tft.setTextColor(ST7735_WHITE);
tft.print("Pick up the pace");
speeddisplay = false;
}
if (tip == 2 && limitspeed < gps.speed.kmph() && !option && !speedmode){
tft.fillRect(ST7735_BLACK);
speeddisplay = true;
}
if(mod == 1){
tft.setTextSize(1);
tft.setCursor(5, 50);
tft.print("Alege tipul limitarii");
tft.setCursor(10, 70);
tft.print("1-superior");
tft.setCursor(10, 90);
tft.print("2-inferior");
if(key == '1'){
tip = 1;
tft.fillRect(ST7735_BLACK);
mod = 0;
Serial.print(F("intra"));
speedmode = true;
delay(200);
}
if(key == '2'){
tip = 2;
tft.fillRect(ST7735_BLACK);
mod = 0;
speedmode = true;
}
```

```
    delay(200);
  }
}
if(mod == 2){
  tft.setTextSize(1);
  tft.setCursor(10, 50);
  tft.print("Alege unitatea de masura");
  tft.setCursor(10, 70);
  tft.print("1-km/h");
  tft.setCursor(10, 90);
  tft.print("2-m/s");
  if(key == '1'){
    unit = 1;
    tft.fillScreen(ST7735_BLACK);
    mod = 0;
    speeddisplay = true;
    prevspeed = -1;
    delay(200);
  }
  if(key == '2'){
    unit = 2;
    tft.fillScreen(ST7735_BLACK);
    mod = 0;
    prevspeed = -1;
    speeddisplay = true;
    delay(200);
  }
}
if(option) {
  tft.setTextSize(1);
  tft.setCursor(10, 50);
  tft.print("Alege optiunea");
  tft.setCursor(10, 70);
  tft.print("1-limitare");
  tft.setCursor(10, 90);
  tft.print("2-unitate");
  if(key == '1'){
    mod = 1;
    tft.fillScreen(ST7735_BLACK);
    option = false;
    limitspeed = 0;
    tip = 0;
    delay(200);
  }
  if(key == '2'){
    mod = 2;
    tft.fillScreen(ST7735_BLACK);
    option = false;
    delay(200);
  }
}
```

```
}  
  
if(pinstate)  
{  
  if (tip == 1 && limitspeed < gps.speed.kmph()) {  
    stop();  
    tft.fillScreen(ST7735_RED);  
    tft.setTextSize(1);  
    tft.setCursor(20, 50);  
    tft.setTextColor(ST7735_WHITE);  
    tft.print("LIMIT EXCEEDED");  
    if (currentMillis - previousMillis >= interval) {  
      tip = 0;  
      limitspeed = 0;  
      prevspeed = -1;  
      prevAlt = -1;  
      tft.fillScreen(ST7735_BLACK);  
    }  
  } else {  
    moveForward();  
  }  
}else {  
  stop();  
}  
  
}
```

Move Forward:

- Schimba starea motoarelor pentru a putea merge in fata;

```
void moveForward() {
```

```
  digitalWrite(in1, LOW);  
  digitalWrite(in2, HIGH);  
  digitalWrite(in3, HIGH);  
  digitalWrite(in4, LOW);  
}
```

Stop:

- Opreste motoarele de tot;

```
void stop() {
```

```
  digitalWrite(in1, LOW);  
  digitalWrite(in2, LOW);  
  digitalWrite(in3, LOW);  
  digitalWrite(in4, LOW);  
}
```

## Rezultate Obținute

Rezultatele obtinute au fost cele asteptate gps-ul fiind testat merge pentru viteze mici cum ar fi cele ale masinutei ,dar la viteze destul de mari prinde un anumit delay.Overall Masina functioneaza exact cum ar trebui gps-ul functionand pentru masina,meniul functioneaza pentru a adauga constrangerile necesare vitezei si pentru a schimba viteza masinii.

<https://www.youtube.com/watch?v=6-NOmEXkooM>

<https://www.youtube.com/shorts/MEMOLnyzuo8>

## Jurnal

- 9 mai: au venit piesele
- 12 mai:am inceput implementarea hardware si am amplasat pini pieselor
- 19 mai:am testat unele functionalitatii ale pieselor si am continuat cu implementarea software
- 25 mai:am verificat toata functionalitatea proiectului

## Concluzii

Am inceput sa prind o afectiune pentru partea de hardware dupa acest proiect.Mi s-a parut foarte interesant sa creezi un altfel de obiect ce poate avea si o folosinta.Planuiesc sa mai adaug asupra proiectului meu inca cateva functii pe viitor,cum ar fi viteza motoarelor sa fie constant la viteza limita data de user.Am invatat ca este nevoie de anumite sacrificii pentru a duce un proiect la bun sfarsit.Din cauza numarului limitat al pinilor am fost nevoit sa renunt la led-uri si buzzer,as fi complicat destul de mult programul fiind nevoit cumva ori sa scad numarul de pini folositi sau sa cresc numarul de pini.Deja am sacrificat coloana a 4 a keypadului de care oricum nu aveam nevoie.Cel mai greu a fost testarea din punct de vedere al gps-ului fiind destul de pretentios cu locatia sa fiind nevoie sa fie departe de blocuri inalte si de preferat afara pentru a putea functiona.In general a fost o experienta placuta de a face un proiect si mai mult ca sigur voi continua sa dezvolt proiectul meu in viitor.

## Download

[radunicolae2.zip](#)

## Bibliografie/Resurse

- <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>
- <https://www.mantech.co.za/datasheets/products/GY-NEO6MV2.pdf>
- <https://www.electroschematics.com/mx1508-motor-driver-module-quick-start/>
- <https://www.elecrow.com/1-44-inch-tft-display-ips-spi-hd-65k-full-color-lcd-module-st7735s-128-128.html>
- <https://www.optimusdigital.ro/>
- <https://cleste.ro/>

[Export to PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2024/amocanu/radu\\_nicolae2103](http://ocw.cs.pub.ro/courses/pm/prj2024/amocanu/radu_nicolae2103)



Last update: **2024/05/27 14:34**