

# DariusCell

Autori:

- [Radu Nichita](#)
- [Cristian Olaru](#)

## Introducere

Am ales să realizăm acest proiect ca să trecem la PM cu notă bună :) (acuma sperăm și să ne iasă).

## Descriere generală

### Motivație

- participarea la concursul [Perpetuum](#), unde am realizat un automat de dat pagini
- dorința de a realiza un robot, ce poate evita obstacolele și să fie controlat de pe telefon. (acesta poate fi apoi îmbunătățit pentru a filma și a monitoriza locuința de la distanță).

### Structură proiect

- în primă fază, vrem să facem un robot care poate fi controlat printr-un joystick.
- în a doua fază, vrem ca robotul să fie capabil să evite obstacolele folosind un senzor.
- în a treia fază, vrem să schimbăm modul de controlare al robotului, prin adăugarea unui modul Bluetooth și realizarea interacțiunii cu o aplicație ( [Bluetooth Terminal](#)).

## Hardware Design

### Componente

- [Placă de dezvoltare compatibilă cu Arduino UNO \(ATmega328p și CH340\)](#)
- [Kit Robot cu 2 Motoare](#)
- [Modul cu Driver de Motoare Dual L298N Rosu](#)
- [Modul Bluetooth Master Slave HC-05 cu Adaptor \(compatibil 3.3 V si 5 V\)](#)
- [Fire Colorate Tată - Tată](#)
- [Fire Colorate Mamă - Mamă](#)
- [Fire Colorate Mamă - Tată](#)
- [Sensor Ultrasonic HC-SR04](#)
- 2 \* [Suport Baterii cu Capac](#)
- și ce s-a mai găsit pe Optimus Digital ...

## Asamblarea robotului

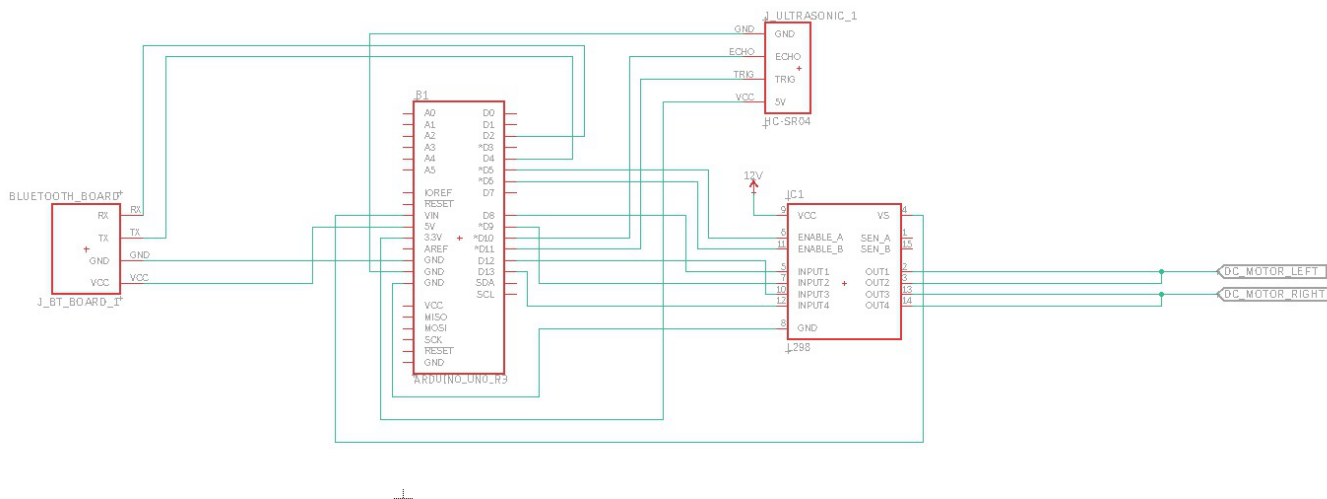
- lipire fire de motoare folosind un pistol de lipit (orice Letcon e bun)
- prima componentă fixată pe șasiu au fost motoarele, la care s-au atașat roțile.
- urmează montarea roții pivotante a robotului.
- conectarea a doi suportți, a câte 4 baterii de 1,5V fiecare, în serie (pentru a asigura tensiunea necesară de 12V, pentru driverul de motoare L298N, urmată de fixarea (cu șuruburi) pe șasiu.
- montare driver de motoare deasupra celor doi suportți (care au capac, pentru a asigura izolarea electrică)
- fixare Arduino pe șasiu și conectare pini între plăcuța și driverul de motoare.
- montare suport plastic în fața robotului. (pentru design + modul Bluetooth în spate și senzor ultrasonic în fața, pentru a da impresia de robot [Wall-E](#))
- fixare modul + senzor + conectarea pinilor cu pini de Arduino.

## Schema bloc



## Schema electrică

Pentru schema electrică am folosit [Eagle](#) și am atașat schema la secțiunea de Resurse



# Software Design

Dezvoltarea codului s-a realizat folosind [Arduino IDE](#) și am bibliotecă externă `SoftwareSerial` pentru a realiza conexiunea cu modulul Bluetooth.

Pași funcționare:

- Utilizatorul deschide aplicația Bluetooth Terminal.
- Având funcția de Bluetooth activată pe telefonul mobil, el se va conecta la dispozitivul HC-05.
- El va putea da următoarele comenzi robotului:
  - w = mișcare înainte
  - a = rotire stânga, urmată de oprire (pentru a simula o rotire pe loc)
  - d = rotire dreapta, urmată de oprire (pentru a simula o rotire pe loc)
  - s = mișcare înapoi, urmată de oprire
  - q = oprirea robotului
- De ce toate operațiile în afară de w necesită oprire? Pentru că partea frontală a robotului are montat un senzor ultrasonic, capabil să detecteze obstacolele din față. El se va opri automat când va ajunge la o distanță apropiată de acestea, fără a se ciocni de ele.

```
#include <SoftwareSerial.h>

#define DELTA_SPEED 10 // increase / decrease in speed;
#define MIN_SPEED_ALLOWED 120
#define BALANCE_DIFFERENCE 17

#define SPEED 127
#define SPEED_TURNING 100

#define UP_DIRECTION 'w'
#define DOWN_DIRECTION 's'
```

```
#define STOP_DIRECTION 'q'
#define LEFT_DIRECTION 'a'
#define RIGHT_DIRECTION 'd'

int In3 = 8; // left wheel - int3
int In4 = 9; // left wheel - int4
int In2 = 12; // right wheel - int2
int In1 = 13; // right wheel - int1
int ENA = 5; // left wheel
int ENB = 6; // right wheel
int robot_speed = 0;

bool goesForward = false;
bool moving = false;
int rx_pin = 4; // pin 4 as RX
int tx_pin = 2; // pin 2 as TX

SoftwareSerial BTSerial(tx_pin, rx_pin);

char bt_data;
String arduino_data;

void setup()
{

  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(ENA, OUTPUT);

  pinMode(In3, OUTPUT);
  pinMode(In4, OUTPUT);
  pinMode(ENB, OUTPUT);

  digitalWrite(In1, LOW);
  digitalWrite(In2, HIGH);

  digitalWrite(In3, LOW);
  digitalWrite(In4, HIGH);

  analogWrite(ENA, robot_speed);
  analogWrite(ENB, robot_speed);

  BTSerial.begin(9600);
  Serial.begin(9600);
  delay(1000);
}
```

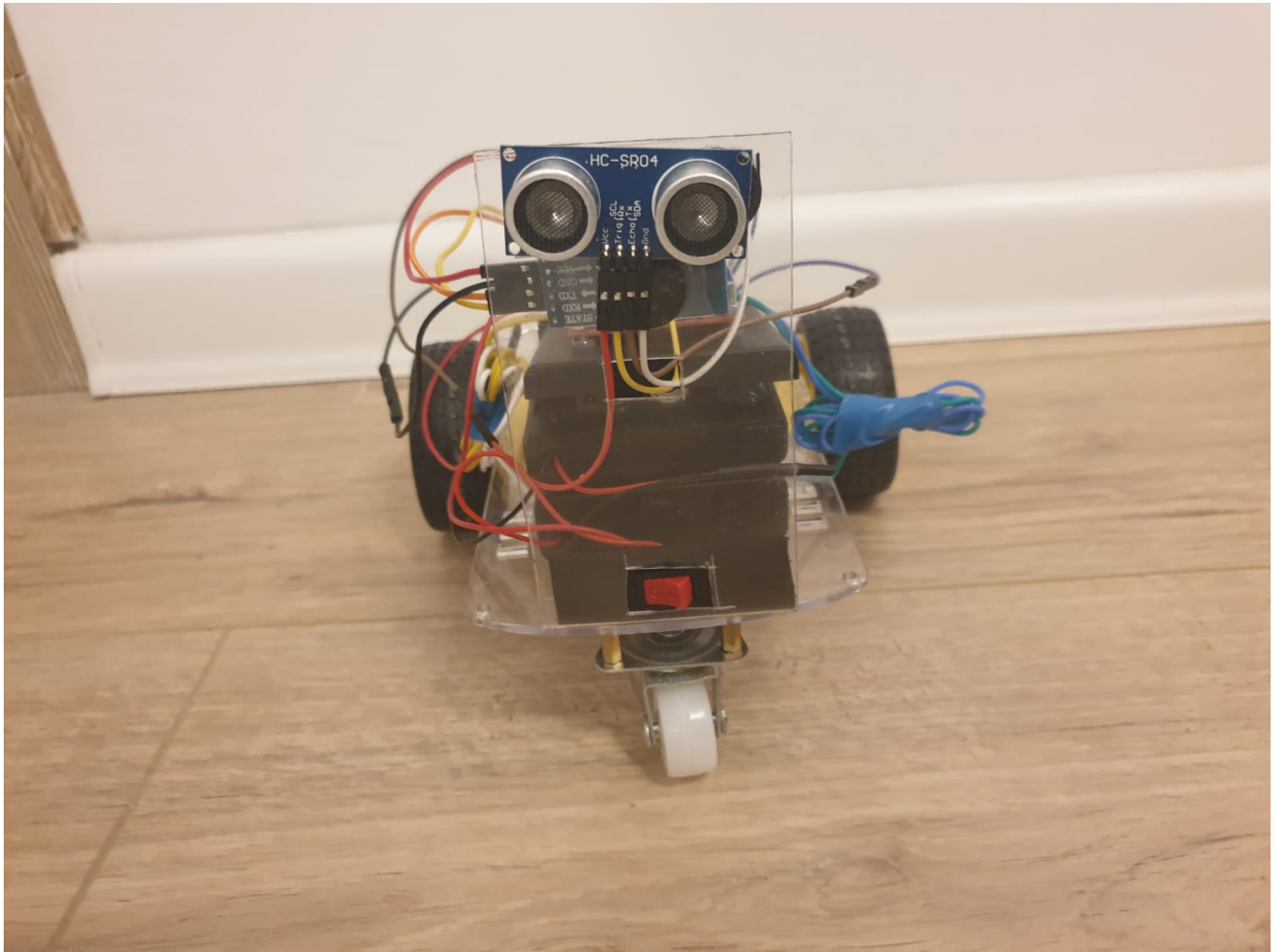
```
void moveForward() {  
    if(!goesForward || !moving) {  
        moveStop();  
        goesForward = true;  
        moving = true;  
        digitalWrite(In1, LOW);  
        digitalWrite(In2, HIGH);  
  
        digitalWrite(In3, LOW);  
        digitalWrite(In4, HIGH);  
  
        for (; robot_speed < SPEED; robot_speed +=2) {  
            analogWrite(ENA, robot_speed + BALANCE_DIFFERENCE);  
            analogWrite(ENB, robot_speed);  
            delay(5);  
        }  
    }  
}  
  
void moveStop() {  
    for (; robot_speed > 0; robot_speed -=2) {  
        analogWrite(ENA, robot_speed + BALANCE_DIFFERENCE);  
        analogWrite(ENB, robot_speed);  
        delay(5);  
    }  
    robot_speed = 0;  
    analogWrite(ENA, robot_speed);  
    analogWrite(ENB, robot_speed);  
    moving = false;  
}  
  
void moveBackward() {  
    if(goesForward || !moving) {  
        moveStop();  
        moving = true;  
        goesForward=false;  
        digitalWrite(In1, HIGH);  
        digitalWrite(In2, LOW);  
        digitalWrite(In3, HIGH);  
        digitalWrite(In4, LOW);  
        for (; robot_speed <SPEED; robot_speed +=2) {  
            analogWrite(ENA, robot_speed + BALANCE_DIFFERENCE);  
            analogWrite(ENB, robot_speed);  
            delay(5);  
        }  
    }  
}  
  
void turnLeft() {
```

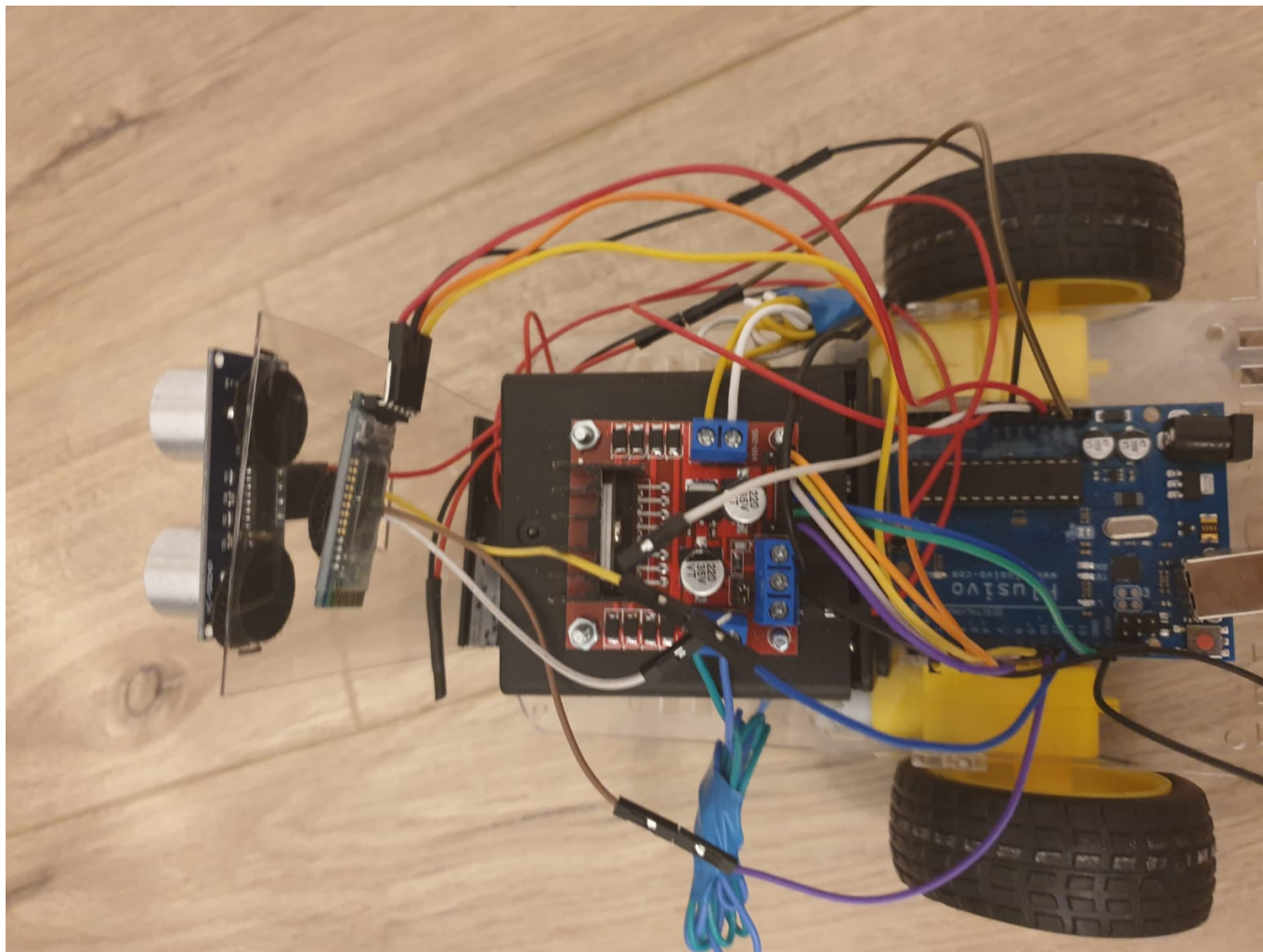
```
if (moving) {
  analogWrite(ENA, 0);
  analogWrite(ENB, SPEED);
  delay(300);
  robot_speed = SPEED;
  for (; robot_speed > 0; robot_speed -=2) {
    analogWrite(ENB, robot_speed);
    delay(5);
  }
  analogWrite(ENA, 0);
  analogWrite(ENB, 0);
  moving = false;
}

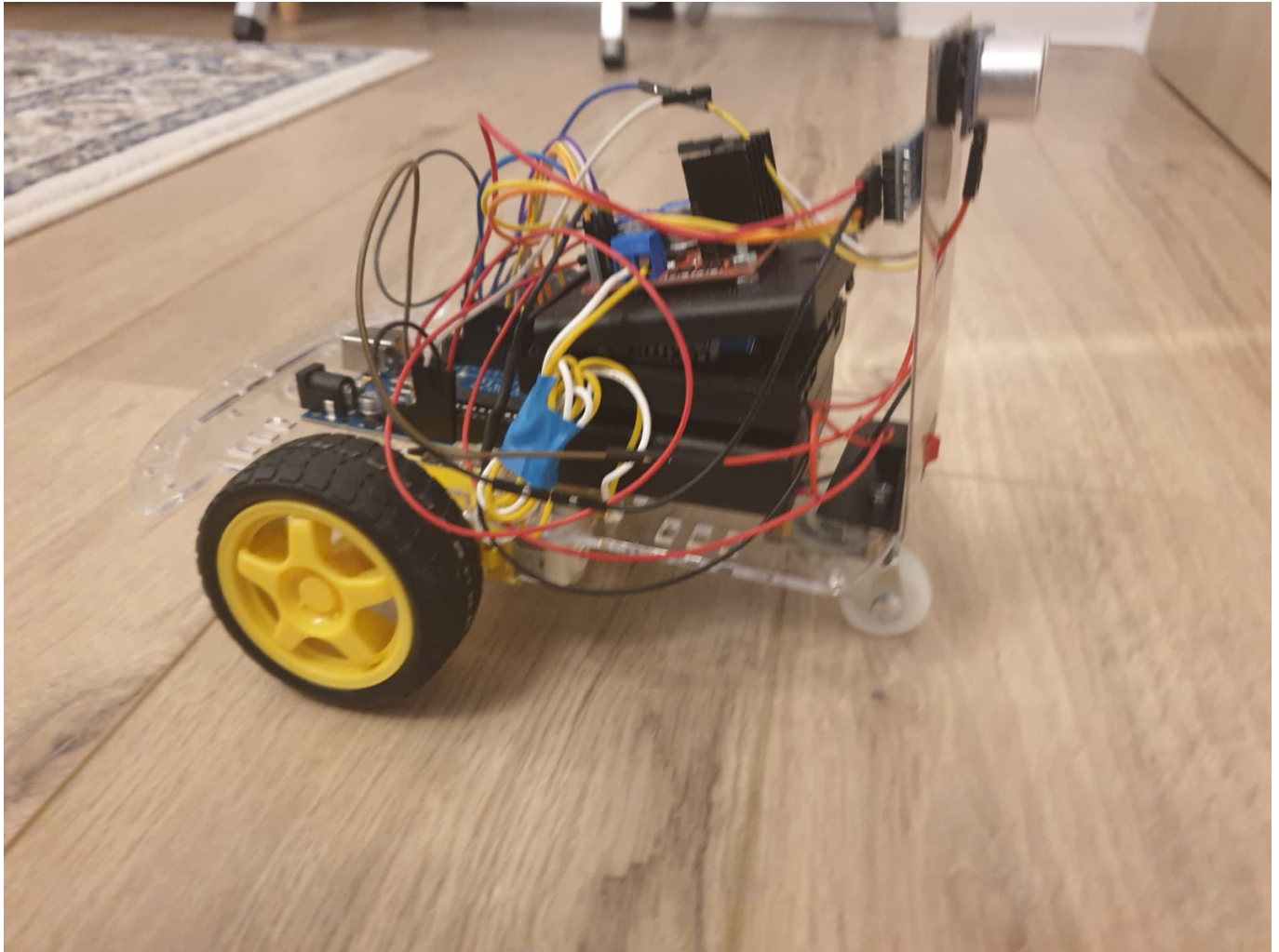
void turnRight() {
  if (moving) {
    analogWrite(ENA, SPEED);
    analogWrite(ENB, 0);
    delay(300);
    robot_speed = SPEED;
    for (; robot_speed > 0; robot_speed -=2) {
      analogWrite(ENA, robot_speed);
      delay(5);
    }
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    moving = false;
  }
}

void loop() {
  if (BTSerial.available()) {
    bt_data = BTSerial.read();
    if (bt_data == UP_DIRECTION) {
      moveForward();
    } else if (bt_data == DOWN_DIRECTION) {
      moveBackward();
    } else if (bt_data == STOP_DIRECTION) {
      moveStop();
    } else if (bt_data == LEFT_DIRECTION) {
      turnLeft();
    } else if (bt_data == RIGHT_DIRECTION) {
      turnRight();
    }
  }
}
```

# Rezultate Obținute







## Concluzie

Acest proiect a fost primul realizat fizic și a combinat multe noțiuni de la materiile de hardware și software. A fost interesant că am putut alege piesele și am avut control deplin asupra proiectului. Partea de asamblare a robotului în sine a fost cea mai challenging, necesitând câteva zile de lipit, aranjat și legat piese. La final, putem zice că a fost o experiență inedită, în care am aplicat majoritatea noțiunilor învățate în cursurile și laboratoarele de PM.

## Jurnal

**02.06.2021** - terminăm de scris documentația pentru proiect. DariusCell is now complete. (+ temele la SO).

**01.06.2021** - ziua Copilului.

**26.05.2021** - "Baterii alimentate pe bază de consum puternic de alcool. A nu se lăsa la îndemâna copiilor!"

**26.05.2021** - Am conectat senzorul ultrasonic și am făcut testele și videoclipul de prezentare a lui DariusCell.

**25.05.2021** - Legare VCC la ground și invers, circuitul ideal.

**24.05.2021** - Am reușit să punem Bluetooth și am comandat un nou senzor ultrasonic, cel actual fiind defect.

**23.05.2021** - Terminare Tema4SO + am reușit să învartim roțile din robotului din cod.

**22.05.2021** - Radu: Cristi, iese fum ... Cristi: Termină cu glumele proaste. (3 seconds later Cristi ia șurubelnița și zvârle bateriile din suport pe jos). Radu: Ia o bere!

**21.05.2021** - mergem să lipim robotul fără să luăm piesele. Bonus, făceam alimentarea la 6V în loc de 12V, cât era necesar pt L298N. GG!

**30.04.2021** - adăugare schemă electrică

**25.04.2021** - DariusCell was born.

## Resurse

- Documentația proiectului în format [pdf](#)
- Un scurt videoclip cu controlarea robotului se poate găsi [aici](#)
- Schema electrică [schema\\_electrica2021.sch](#)

## Documentație

- [L298N Datasheet](#)
- [HC-05 Bluetooth module Datasheet](#)
- [Ultrasonic Module HC - SR04](#)
- [PWM](#)
- [Tutorial Youtube](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2021/abasoc/dariuscell>

Last update: **2021/06/03 14:54**



