

## Laborator 9 - Utilizare motoare si servomotoare. Robotul experimental.

Această lucrare de laborator prezintă utilizarea motoarelor de curent continuu (DC), și utilizarea servo-motoarelor.

### 1. Motoare DC



Figura 1. Motor DC cu reductor de turatie 1:48

Motoarele de curent continuu (DC motors) clasice convertesc energia electrică în lucru mecanic. Viteza de rotație a unui motor este proporțională cu tensiunea de alimentare de la bornele acestuia, iar direcția de rotație depinde de polaritate (conectarea celor 2 fire de alimentare ale motorului la +Vcc și Gnd, sau vice-versa). Motoarele au cutie de viteze (reductor de turatie) cu raport de 1:48, ceea ce înseamnă că pentru o rotație completă a axului extern se efectuează de fapt 48 de rotații ale motorului electric. Folosirea unui reductor are avantajul că mărește forța de acționare, cu costul vitezei.

Datorită faptului că motoarele necesită o intensitate a curentului semnificativă pentru a produce mișcare, ele nu pot fi conectate direct la ieșirile (pinii) unui microcontroller. Se impune separarea semnalelor de comandă de circuitul de putere, și acest lucru se realizează prin folosirea punților H ("H bridges"). Punțile H sunt circuite care conțin 4 comutatoare (de obicei tranzistori), numerotate S1, S2, S3 și S4 (figura 2).

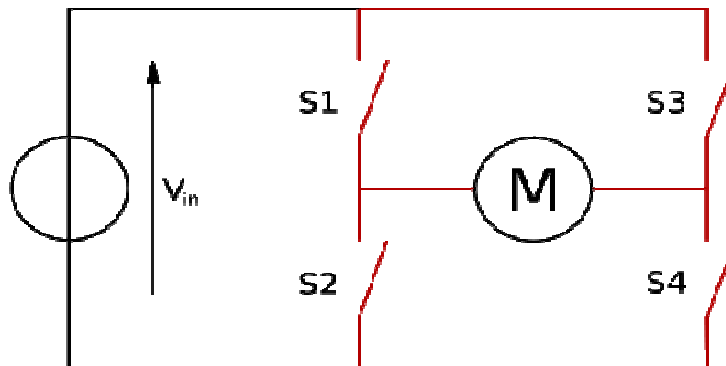


Figura 2. Punte H: S1-S4 sunt comutatoarele, iar M reprezintă motorul.

Denumirea de punte „H” vine de la aspectul schemei din figura de mai sus. Porțile din stânga sus (S1) și dreapta jos (S4) sunt de obicei conectate la un semnal de control comun (“A”), în timp ce porțile din dreapta sus (S3) și stânga jos (S2) sunt conectate la un al semnal de control comun, (“B”). Semnalele A și B sunt exclusive, activarea unuia cauzând rotația motorului într-un anume sens. Activarea ambelor semnale în același timp va scurtcircuita sursa de alimentare.

Cele două stări permise ale comutatoarelor unei punți H sunt ilustrate mai jos în figura 3:

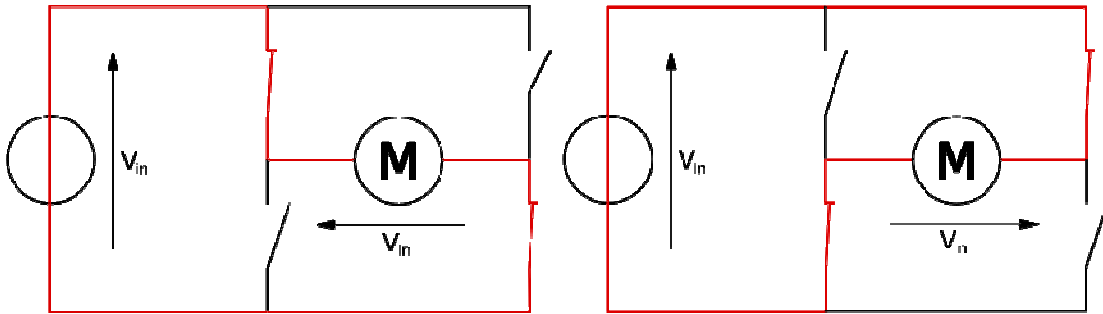


Figura 3. Stările posibile ale comutatoarelor.

Prin deschiderea comutatoarelor S1 și S4 motorul se va roti într-o direcție, iar dacă vom deschide comutatoarele S2 și S3 motorul se va roti în direcția opusă.

Pentru această lucrare de laborator vom folosi puntea duală L298N Dual H-Bridge, care este capabilă să acționeze două motoare DC în același timp.

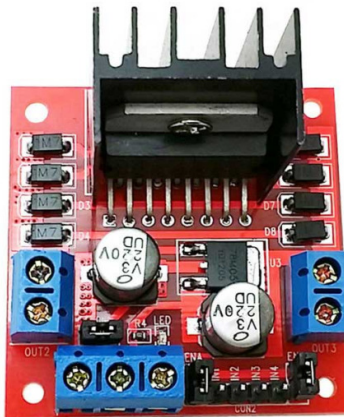


Figura 4. L298N Dual H-Bridge.

Specificațiile circuitului:

- Tensiunea de alimentare pentru acționarea motoarelor (pinul marcat +12V) Vs: 5~35V; Dacă dorim să alimentăm din aceeași sursă și placa Arduino, trebuie să atașăm o tensiune între 7 și 35V, pentru a permite regulatorului integrat să genereze tensiunea de 5 V pe pinul +5V.
- Curent maxim pentru circuitul de alimentare motoare: 2A

- Tensiune pentru alimentarea circuitelor logice (pinul marcat +5V)  $V_{ss}$ : 5 - 7V (poate fi conectat la Arduino + 5V, pentru alimentarea acestuia)
- Curent maxim pentru circuitul logic 36mA
- Nivele ale semnalelor de control: logic 0,  $-0.3 \leq V_{in} \leq 1.5V$ , logic 1,  $2.3V \leq V_{in} \leq V_{ss}$
- Putere maximă: 20W

Schema bloc este prezentată în figura 5:

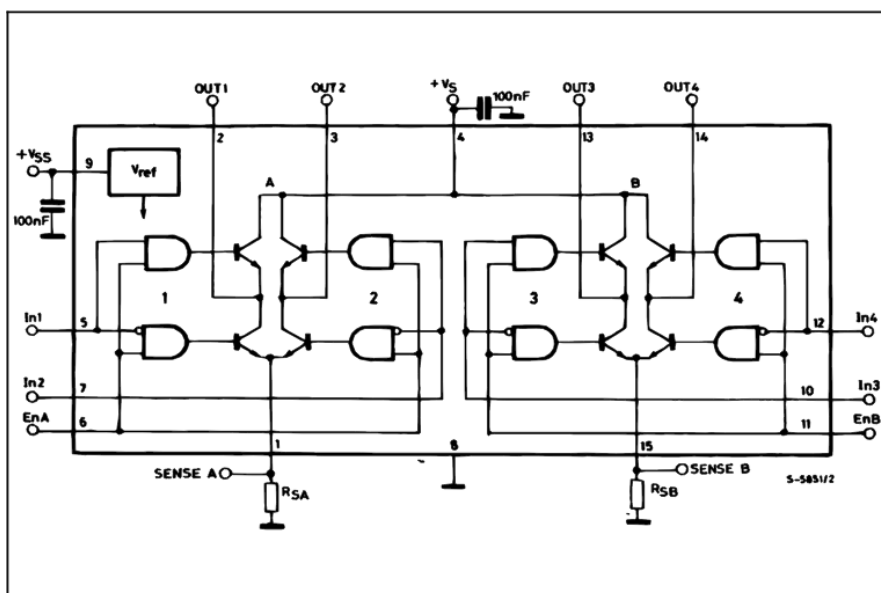


Figura 5. Schema bloc a circuitului driver L298N

Fiecare motor are trei pini de control. Astfel, primul motor este controlat de pinii EnA, In1, si In2, iar motorul al doilea de pinii EnB, In3 si In4. Pinii En sunt conectați la nivelul logic 1 prin jumperi, deci prin program vom controla doar pinii In. Sunt disponibile următoarele combinații:

In1	In2	Efect
0	0	Motor 1 oprit (frână)
0	1	Motor 1 pornit – înainte
1	0	Motor 1 pornit – înapoi
1	1	Motor 1 oprit (frână)

In3	In4	Efect
0	0	Motor 2 oprit (frână)
0	1	Motor 2 pornit – înainte
1	0	Motor 2 pornit – înapoi
1	1	Motor 2 oprit (frână)

Motoarele pot fi acționate în același timp. Nu schimbați direcția de rotație a motorului fără a-l opri înainte pentru câteva milisecunde.

Turația unui motor este dată de tensiunea aplicată acestuia. Deoarece din microcontroller putem să generăm doar semnale de 0 sau 5 V, pentru varierea turației unui motor puteți folosi un semnal de tip PWM, aplicat pe pinii In1, In2, In3 sau In4.

## 2. Servo-motoare

Spre deosebire de motoarele DC, care produc rotație continuă cât timp sunt conectate la o sursă de tensiune, motoarele servo sunt folosite pentru a obține rotații parțiale, stabile și controlate, pentru efectuarea unor operații cu amplitudine mică dar cu precizie ridicată: acționare mecanism de închidere-deschidere, poziționare senzori, efectuarea unor gesturi, etc.



Figura 6. Motor Servo.

Motoarele servo au 3 fire, iar culoarea acestora variază în funcție de producător. Culoarea roșie desemnează de obicei Vcc (5V), în timp ce GND este de obicei negru sau maro. Pe lângă aceste două fire de alimentare, există un al treilea, firul de comandă, care este de obicei galben, portocaliu sau alb. Figura 7 ilustrează diferite tipuri de scheme de culori.

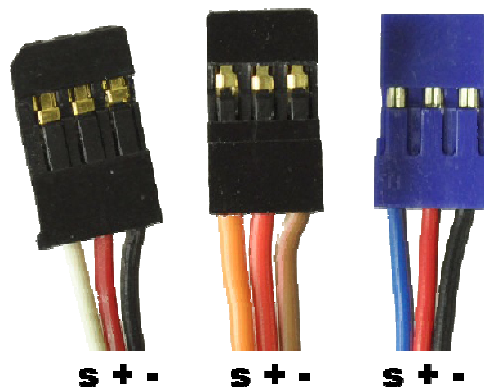


Figura 7. Tipuri de scheme de culori folosite la servo-motoare.

Motorul servo nu va executa (de obicei!) o rotație completă, ci va devia de la poziția de echilibru cu un unghi controlat de tensiunea aplicată pinului de semnal. Folosind un semnal PWM pe acest pin, vom avea control asupra unghiului de rotație al motorului.

Cel mai simplu mod de a controla motoarele de tip servo este prin folosirea bibliotecii Servo. Folosirea acestei biblioteci permite controlarea a până la 48 de motoare pe placa Arduino Mega. Dacă se folosesc mai mult de 12 motoare, biblioteca va dezactiva funcția PWM pe pinii 11 și 12. La Arduino Uno, această bibliotecă va dezactiva PWM pe pinii 9 și 10, indiferent de câte motoare se folosesc.

Metodele puse la dispoziție de biblioteca Servo sunt prezentate mai jos:

**servo.attach(pin) / servo.attach(pin, min, max)** – atașează obiectul Servo la pini

- servo: un obiect instanță a clasei Servo
- pin: numărul pinului digital unde va fi atașat semnalul pentru motorul Servo
- min (opțional): lățimea pulsului, în microsecunde, corespunzătoare unghiului minim (0 grade) al motorului servo (implicit 544)
- max (opțional): lățimea pulsului, în microsecunde, corespunzătoare unghiului maxim (180 grade) al motorului servo (implicit 2400)

**servo.detach()** – detașează obiectul de tip Servo de la pin.

**boolean val servo.attached()** – verifica dacă obiectul de tip Servo este atașat unui pin.

Returnează adevărat sau fals.

**servo.write (angle)** – scrie o valoare (0 .. 180) către servo, controlând mișcarea:

- Pentru Servo standard ⇒ setează unghiul axului [grade] cauzând motorul să se orienteze în direcția specificată.
- Servo cu rotație continuă ⇒ configurează viteza de rotație (0: viteza maxima intr-o directie; 180: viteza maxima in direcția opusă; ≈ 90: oprit)

**int val servo.read()** – citește unghiul curent al motorului servo, configurat la ultimul apel al metodei **write()**.

### 3. Robotul experimental

Pentru desfășurarea activităților de proiect, au fost asamblați roboți experimentali, având următoarele componente:

1. Placă microcontroller compatibilă Arduino Uno
2. Driver motoare L298N Dual H-Bridge
3. 2x Motor DC
4. 1 Motor Servo
5. Carcasă baterii 4xAA (R6)
6. 2 roți conectate la motoare, 1 roată suport
7. Suport plexiglas
8. Două plăci de prototipizare (Breadboard)

## 9. 1 senzor sonar (neconectat)

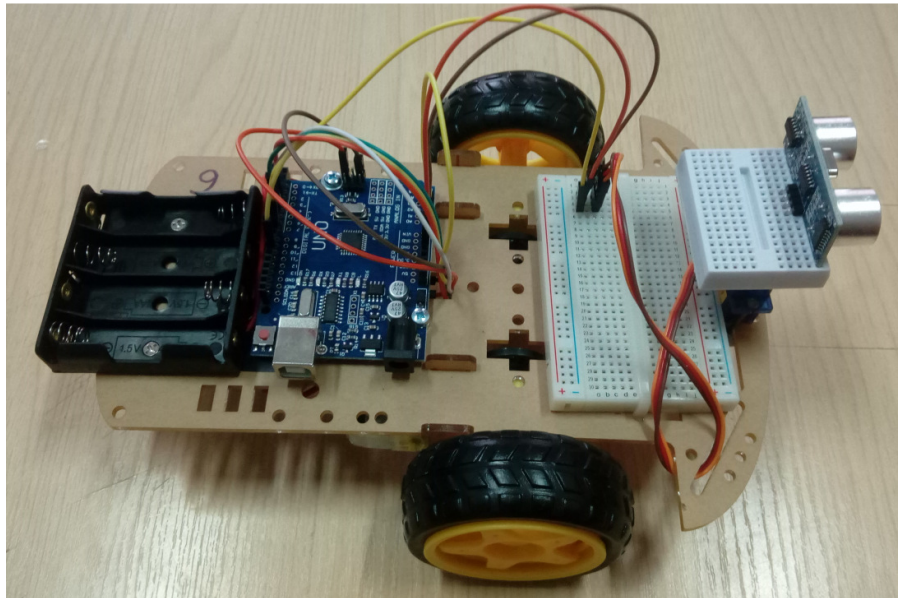


Figura 8. Robotul experimental – vedere de sus – elementele de control.

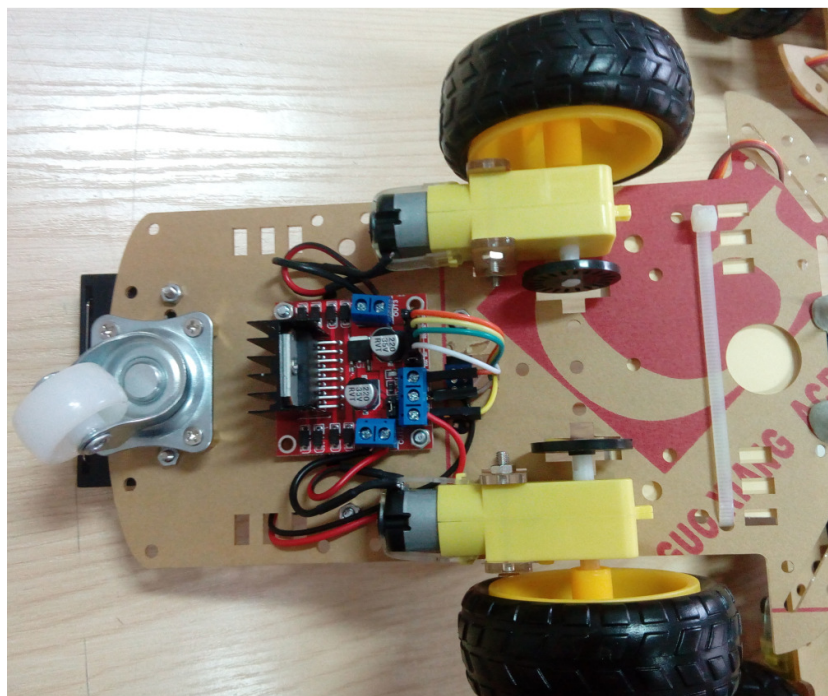


Figura 9. Robotul experimental – vedere de jos – circuitul de putere.

Schema conexiunilor dintre elementele robotului este prezentată în figura 10:

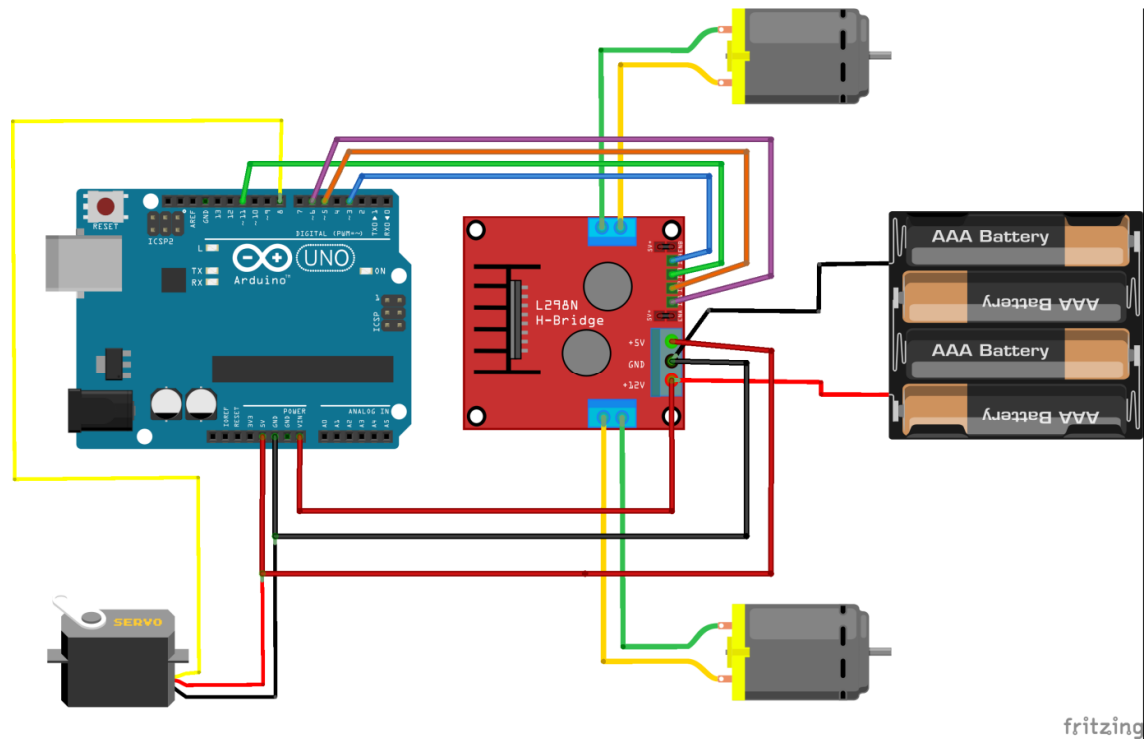


Figura 10. Schema electrică a robotului.

Câteva caracteristici importante:

- Alimentarea motoarelor se poate face de la baterii (soluție neoptimă, conform specificațiilor L298N, dar care oferă mobilitate), sau de la o sursă externă conectată la mufa jack a plăcii Arduino. Pinul Vin al Arduino este conectat la mufa jack, și este de asemenea conectat la pinul +12V al L298N.
- Nu folosiți sursa externă împreună cu bateriile, pentru că ele vor fi expuse tensiunii acestei surse !
- Ieșirea +5V a L298N este conectată la alimentarea +5V de pe placa Arduino. Pentru economisirea pinilor, a fost utilizat conectorul ICSP.
- Pini de control ai motoarelor au fost conectați la pinii 3 și 11 (Motor 1), și 5 și 6 (Motor 2), pini capabili de generare semnal PWM.
- Controlul motorului Servo este conectat la pinul 8.

Elemente de precauție:

- Nu lăsați robotul să pornească motoarele cât timp este conectat cu cablul de programare la PC. În programul exemplu, care va fi prezentat mai jos, există un timp de avertizare (LED-ul de pe pinul 13 va pâlpâi la început mai rar, apoi mai des, înainte de pornirea motoarelor). Programați robotul, apoi deconectați cablul USB, și conectați sursa externă sau baterii.
- Verificați polaritatea sursei externe de alimentare (electrodul pozitiv să fie în centrul mufei), și tensiunea acesteia (recomandare: 7.5V).
- Nu lăsați robotul să pornească motoarele cât timp este cu roțile pe masă! Se va mișca brusc, și poate cauza rupere de cabluri sau poate cădea de pe masă. Țineți robotul în

mână, sau montați un suport care să țină roțile depărtate de masă. Când doriți să testați modul de deplasare al robotului, puneți-l pe podea și lăsați spațiu suficient în jurul lui.

#### 4. Program exemplu

Următorul cod este deja programat în microcontrollerul robotului. El va testa toate motoarele, după execuția codului de avertizare.

```
#include <Servo.h>

// Pinii motor 1
#define mpin00 5
#define mpin01 6

// Pinii motor 2
#define mpin10 3
#define mpin11 11

Servo srv;

void setup() {
    // configurarea pinilor motor ca iesire, initial valoare 0
    digitalWrite(mpin00, 0);
    digitalWrite(mpin01, 0);
    digitalWrite(mpin10, 0);
    digitalWrite(mpin11, 0);

    pinMode (mpin00, OUTPUT);
    pinMode (mpin01, OUTPUT);
    pinMode (mpin10, OUTPUT);
    pinMode (mpin11, OUTPUT);

    // pin LED
    pinMode(13, OUTPUT);
}

// Functie pentru controlul unui motor
// Intrare: pinii m1 si m2, directia si viteza
void StartMotor (int m1, int m2, int forward, int speed)
{

    if (speed==0) // oprire
    {
        digitalWrite(m1, 0);
```



```

    digitalWrite(m2, 0);
}
else
{
    if (forward)
    {
        digitalWrite(m2, 0);
        analogWrite (m1, speed); // folosire PWM
    }
    else
    {
        digitalWrite(m1, 0);
        analogWrite(m2, speed);
    }
}
}

// Functie de siguranta
// Executa oprire motoare, urmata de delay
void delayStopped(int ms)
{
    StartMotor (mpin00, mpin01, 0, 0);
    StartMotor (mpin10, mpin11, 0, 0);
    delay(ms);
}

// Utilizare servo
// Pozitionare in trei unghiuri
// La final, ramane in mijloc (90 grade)
void playWithServo(int pin)
{
    srv.attach(pin);
    srv.write(0);
    delay(1000);
    srv.write(180);
    delay(1000);
    srv.write(90);
    delay(1000);
    srv.detach();
}

void loop() {

```

```

// Cod avertizare
// Blink lent
for (int i=0; i<10; i++)
{
    digitalWrite(13, 1);
    delay(200);
    digitalWrite(13, 0);
    delay(200);
}

// Blink rapid. Scoateti cablul USB!!!!
for (int i=0; i<10; i++)
{
    digitalWrite(13, 1);
    delay(100);
    digitalWrite(13, 0);
    delay(100);
}

digitalWrite(13, 1);

// Pornirea motorului Servo
playWithServo(8);

// Acum se pornesc motoarele DC
StartMotor (mpin00, mpin01, 0, 128);
StartMotor (mpin10, mpin11, 0, 128);

delay (500); // Cat timp e motorul pornit
delayStopped(500); // Cat timp e oprit

StartMotor (mpin00, mpin01, 1, 128);
StartMotor (mpin10, mpin11, 1, 128);

delay (500);
delayStopped(500);

StartMotor (mpin00, mpin01, 0, 128);
StartMotor (mpin10, mpin11, 1, 128);

delay (500);
delayStopped(500);

StartMotor (mpin00, mpin01, 1, 128);

```

```
StartMotor (mpin10, mpin11, 0, 128);  
  
delay (500);  
delayStopped(500);  
}
```

### **Desfășurarea activităților de proiect:**

1. Fiecare echipă de studenți (1 sau 2) va prelua un robot. Cadrul didactic va scrie în fișa de prezență numărul robotului. În săptămânile următoare, studenții vor folosi aceeași roboți, cu excepția cazurilor de forță majoră.
2. Se vor verifica conexiunile, să corespundă cu documentația. Se va rula programul de test, și se va urmări ca toate motoarele să funcționeze corect (fiecare motor trebuie să efectueze rotații în ambele sensuri). Dacă există defecte, se vor identifica și se va încerca remedierea lor.

### **Punctul 2 este obligatoriu pentru fiecare ședință de laborator!**

3. Echipele vor adăuga funcționalități robotului, prin modificarea programului și prin adăugarea de componente suplimentare. Evitați modificările permanente, lăsați roboții așa cum i-ați găsit.
4. Fiecare echipă trebuie să își păstreze codul program confidențial. Este responsabilitatea fiecărui student să se asigure că nu rămâne cod pe calculator, și că realizările sunt salvate la loc sigur.
5. În ultima ședință de laborator, se va evalua proiectul din punct de vedere al funcționalității, al complexității și al originalității. Studenții vor preda și o scurtă documentație (5-10 pagini).

### **Fiecare componentă defectată se va înlocui de către persoana vinovată !**

### **Idei pentru funcții suplimentare (se pot cumula):**

- Implementarea unui control precis al motoarelor pentru a asigura o deplasare în linie dreaptă. Implică măsurarea turației și reglarea pulsului PWM pentru a asigura turație egală celor două motoare.
- Detecția și ocolirea obstacolelor. Implică utilizarea senzorului sonar împreună cu motorul servo pentru a determina distanța și unghiul până la obstacole sau spre spațiul liber.
- Crearea unei hărți a mediului înconjurător. Implică detecția obstacolelor și evidența mișcării proprii, pentru a poziționa obstacolele pe hartă.
- Utilizarea memoriei EEPROM pentru a memora un traseu parcurs anterior.
- Control la distanță, folosind telefonul mobil. Implică utilizarea modulului WiFi.
- Comunicare între roboți, prin gesturi.
- Urmărire robot leader.

Este posibil ca funcțiile suplimentare să necesite achiziționarea unor piese. Dacă aceste piese sunt achiziționate de către studenți, ele rămân în proprietatea acestora. Se recomandă achiziționarea individuală a cel puțin 4 baterii AA alcaline.

Recomandare: puteți folosi avantajele plăcilor Mega fără a desface plăcile UNO de pe robot. Conectați plăcile prin I2C, și folosiți placa UNO pentru acționarea motoarelor, iar placa Mega pentru celelalte funcții. Conectați împreună pinii 5V și GND ai celor două plăci, pentru alimentare simultană.

**Referințe suplimentare:**

1. Senzor fotoelectric pentru utilizarea împreună cu roata perforată, pentru măsurarea turației

[https://ardushop.ro/ro/home/146-senzor-de-intrerupere-infrarosu.html?search\\_query=fotoelectric&results=1](https://ardushop.ro/ro/home/146-senzor-de-intrerupere-infrarosu.html?search_query=fotoelectric&results=1)

sau

<https://www.robofun.ro/senzor-ir-break-beam-led-3mm>

2. Senzorul de tip sonar pentru determinarea distanțelor

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

3. Utilizarea senzorilor de distanță Sharp

[http://www.robotshop.com/letsmakerobots/files/IR\\_SHARP.doc](http://www.robotshop.com/letsmakerobots/files/IR_SHARP.doc)

4. Utilizarea memoriei EEPROM integrată AVR

<https://www.arduino.cc/en/Reference/EEPROM>