

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

## **Comanda motoarelor de curent continuu folosind Arduino si driver-ul de motoare shield L298**

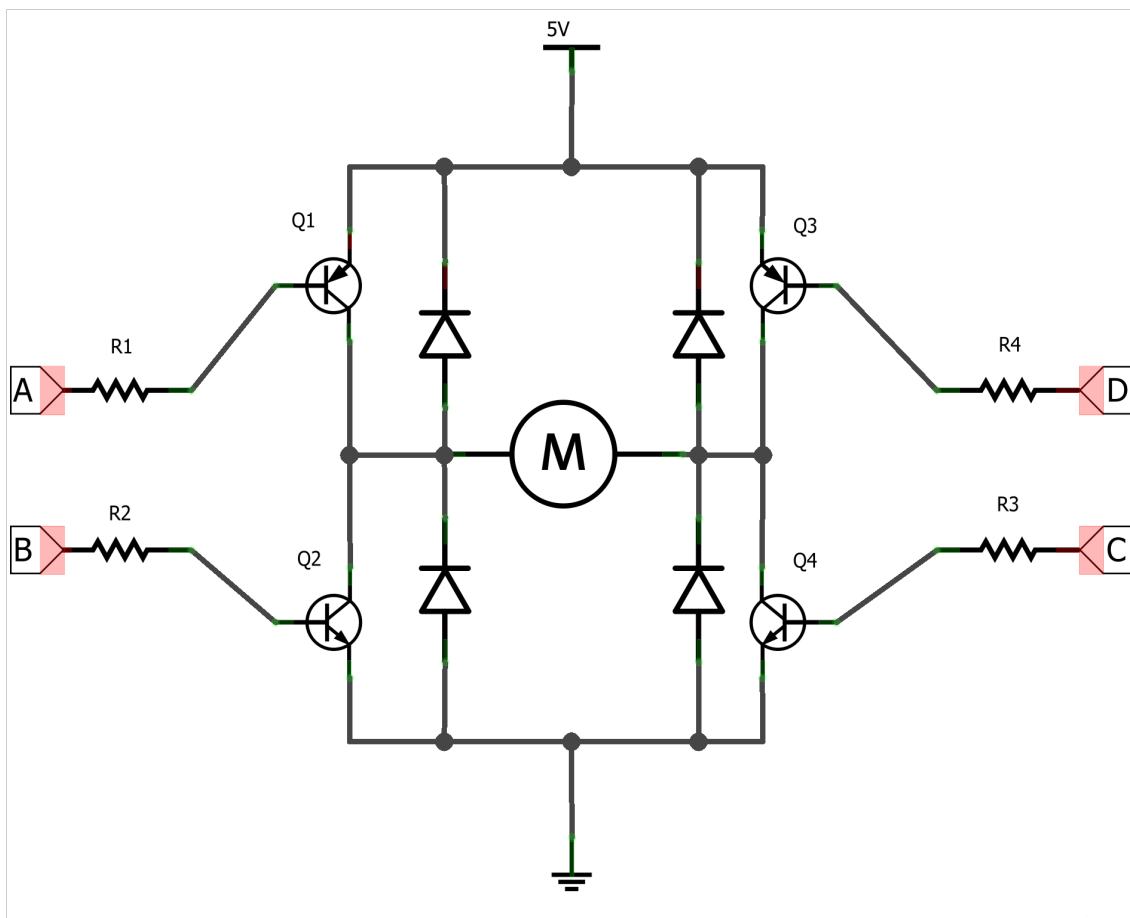
Driver-ul de motoare L298 se utilizeaza pentru controlul motoarelor de curent continuu. Arduino este capabil sa scoata pe porturile lui o putere foarte mica, total insuficienta pentru a invarti un motor. Daca vom conecta un motor electric direct la un port Arduino, cel mai probabil vom obtine arderea procesorului din placa Arduino.

Ca sa nu se intample acest lucru, avem nevoie de un amplificator de putere, care sa ia putere din sursa de alimentare (baterie, de exemplu), si sa o transmita motoarelor asa cum ii spune Arduino. Acest amplificator poarta numele generic de "driver de motoare". Exista o multitudine de drivere de motoare, diferenta majora intre ele fiind cat de multa putere pot conduce (cat de puternice pot fi motoarele pe care le pot controla).

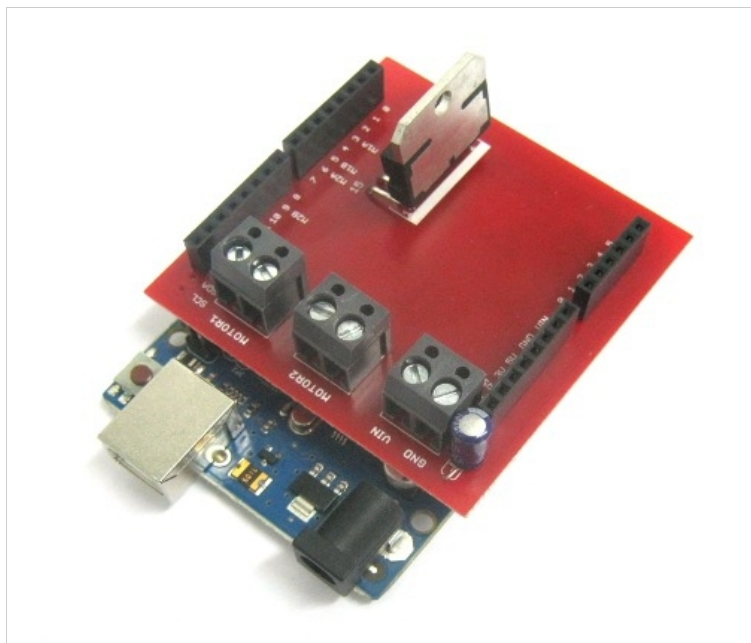
Driver-ul din aceasta sectiune este bazat pe integratul L298, fiind un driver de nivel mediu din punct de vedere al puterii conduse. Poate controla motoare care necesita cel mult 2 Amperi (daca nu-ti e clar ce sunt aia "Amperi", vezi tutorialul "Analogie electricitate - curgerea fluidelor").

Ca sa intelegi mai bine lucrurile, in cele ce urmeaza vom prezenta simplificat modul de functionare interna al integratului L298. Nu este necesar sa intelegi neaparat cum functioneaza in interior ca sa il poti utiliza, dar ajuta. Integratul L298 este format din foarte multe tranzistoare conectate in cascada. Pentru simplificare, vom considera doar patru tranzistoare, conectate ca mai jos.

In schema de mai jos este o punte H alcatuita din tranzistoare bipolare. Motorul este simbolizat cu M si diodele au rol de protectie. Daca aplici tensiune pe punctele A,B,C,D conform tabelului de adevar, vei obtine cele 4 situatii. Daca nu aplici nici o tensiune pe bazele tranzistoarelor atunci motorul ramane in starea in care era inainte, adica nu se va roti. Punctele A,B,C,D se conecteaza la porturile platformei Arduino.



A	B	C	D	Rezultat obtinut
1	0	1	0	Motorul se roteste in sensul acelor de ceasornic
0	1	0	1	Motorul se roteste in sensul invers acelor de ceasornic
1	0	0	1	Motorul franeaza
0	1	1	0	Motorul franeaza



Pe shield sunt prezenti 3 conectori cu surub unde la MOTOR1 si MOTOR2 se conecteaza cele doua motoare iar la conectorul marcat cu Vin si Gnd se conecteaza sursa de alimentare externa pentru motoare. Pinii de comanda ai driverului sunt conectati prin shield la pinii 3, 5, 6 si 9 de pe platforma Arduino.

Codul sursa este prezentat mai jos.

```

int MOTOR2_PIN1 = 3;
int MOTOR2_PIN2 = 5;
int MOTOR1_PIN1 = 6;
int MOTOR1_PIN2 = 9;

void setup() {
    pinMode(MOTOR1_PIN1, OUTPUT);
    pinMode(MOTOR1_PIN2, OUTPUT);
    pinMode(MOTOR2_PIN1, OUTPUT);
    pinMode(MOTOR2_PIN2, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    go(255,-255);
    delay(1000);
    go(-255,-255);
    delay(1000);
    go(-255,255);
    delay(1000);
    go(255,255);
    delay(1000);
}

void go(int speedLeft, int speedRight) {
    if (speedLeft > 0) {
        analogWrite(MOTOR1_PIN1, speedLeft);
        analogWrite(MOTOR1_PIN2, 0);
    }
    else {
        analogWrite(MOTOR1_PIN1, 0);
        analogWrite(MOTOR1_PIN2, -speedLeft);
    }

    if (speedRight > 0) {
        analogWrite(MOTOR2_PIN1, speedRight);
        analogWrite(MOTOR2_PIN2, 0);
    }
    else {
        analogWrite(MOTOR2_PIN1, 0);
        analogWrite(MOTOR2_PIN2, -speedRight);
    }
}

```

Funcția `setup()` setează pinii 3, 5, 6 și 9 ca fiind ieșire. În interiorul buclei `loop()` se execută funcția `go()` care determină viteza cu care se vor roti motoarele dar și sensurile de rotație.

Îți recomand să nu modifici conținutul funcției `go()` (cel puțin deocamdată). Ca să poți folosi shield-ul, tot ce ai de făcut este să apelezi funcția `go()` în `loop()` cu cei doi parametri, `vitezaMotorStanga` respectiv `vitezaMotorDreapta`. Valorile posibile pentru acești doi parametri sunt între -255 și 255.

În general, este important ca motoarele să fie alimentate de la o sursă de putere altă decât cea care alimentează placa Arduino. Shield-ul ține cont de această recomandare, astfel încât așa cum spuneam și mai sus, este necesar să îl alimentezi prin pinii VIN și GND de la o sursă de putere externă (cutie cu baterii sau acumulator). Motoarele consumă foarte mult curent, mult mai mult decât poate furniza portul USB. Driverul poate fi alimentat la o tensiune mult mai mare decât 5V. În plus, motoarele introduc zgomot în circuit atunci când își schimbă sensul sau dacă nu sunt decuplate corect cu condensatori. Dacă alimentezi motoarele extern, atunci toate riscurile de mai sus sunt eliminate.