Cum se controleaza sensul de rotatie al unui motor prin Wi-Fi

In acest tutorial vei descoperi cum se poate controla sensul de rotatie al unui motor de curent continuu, utilizand un Arduino WiFi Shield si un Driver de motoare L298. Tutorialul poate fi un punct de start foarte bun atunci cand vrei sa realizezi un robot pe care sa il comanzi de la distanta sau vrei sa controlezi alte dispozitive prin WiFi.

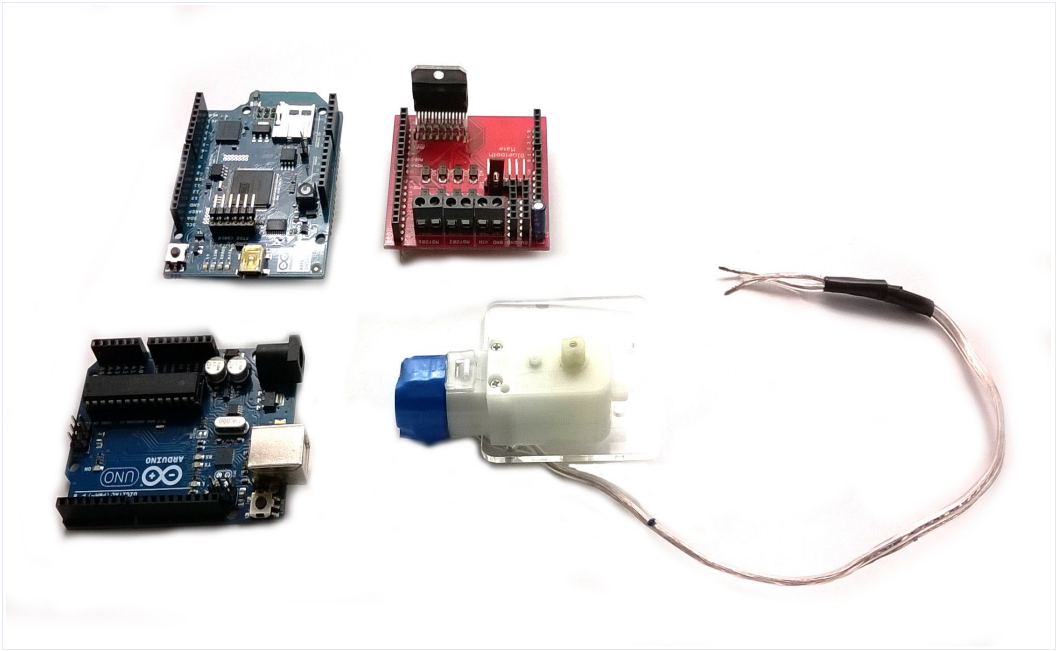
Comanda motorului se va realiza prin reteaua locala, mai exact prin protocolul UDP. Te vei folosi de tastatura calculatorului pentru a transmite 3 comenzi catre placa Arduino. Comenzile sau tastele apasate vor fi preluate printr-o aplicatie scrisa in Processing.

Vei avea nevoie de urmatoarele componente:

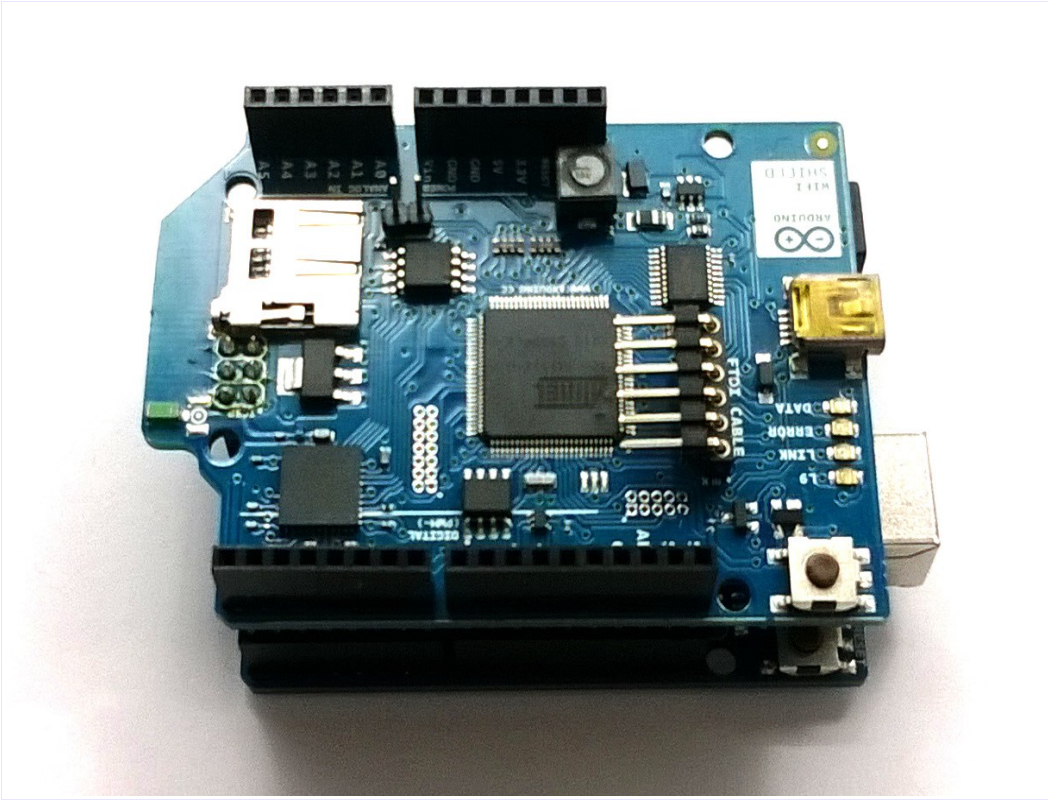
- O placa Arduino - http://www.robofun.ro/arduino
- Arduino Wifi Shield - http://www.robofun.ro/arduino_wifi_shield
- Driver Motoare L298 versiunea 2:
  http://www.robofun.ro/mecanice/driver/shield-motoare-l298-v2
- Un motor compatibil cu Driver-ul L298:
  http://www.robofun.ro/mecanice/motoare
- Alimentator extern Arduino 9V @ 1A:
  http://www.robofun.ro/surse_de_alimentare/alimentatoare/alimentator-extern-arduino-9V

Cum se asambleaza ?

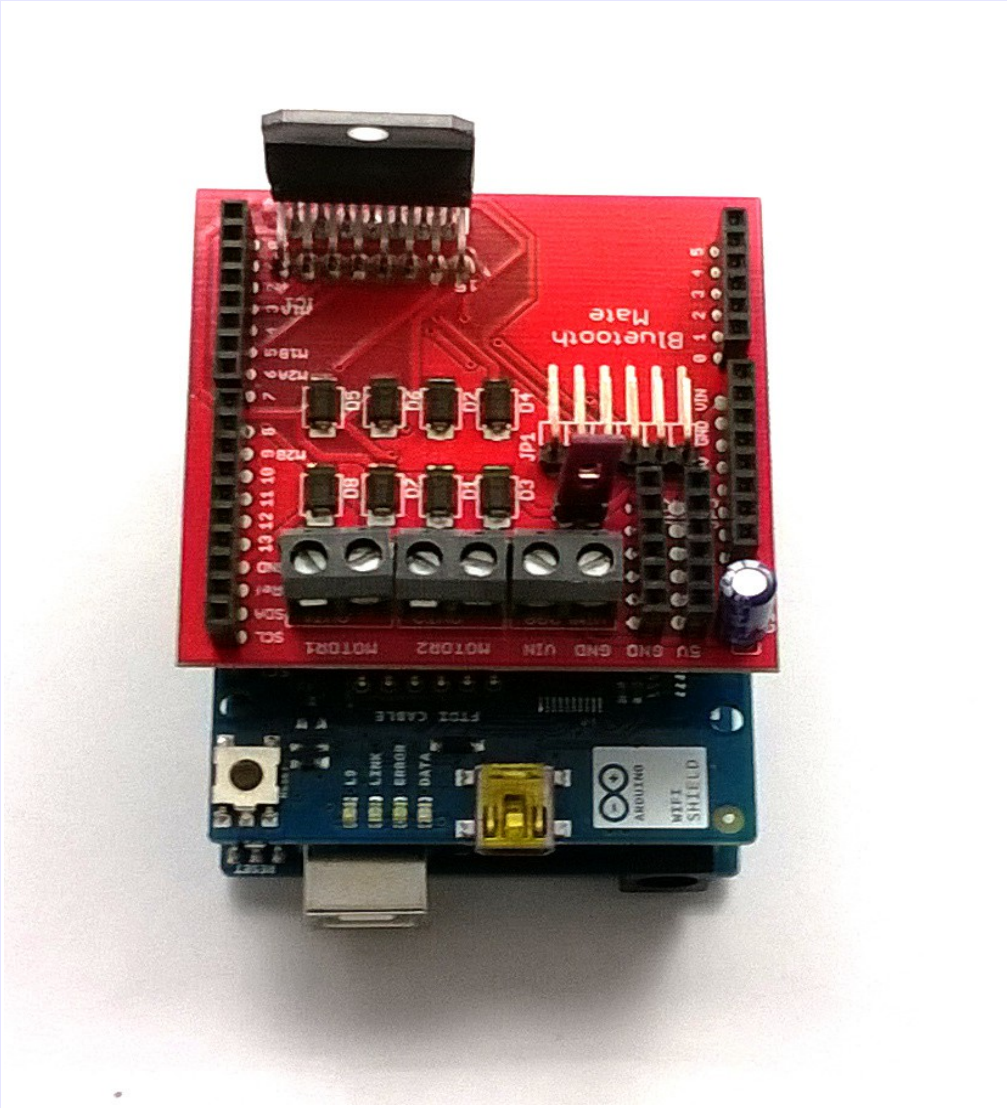
Asamblarea celor 3 placi este foarte simpla:



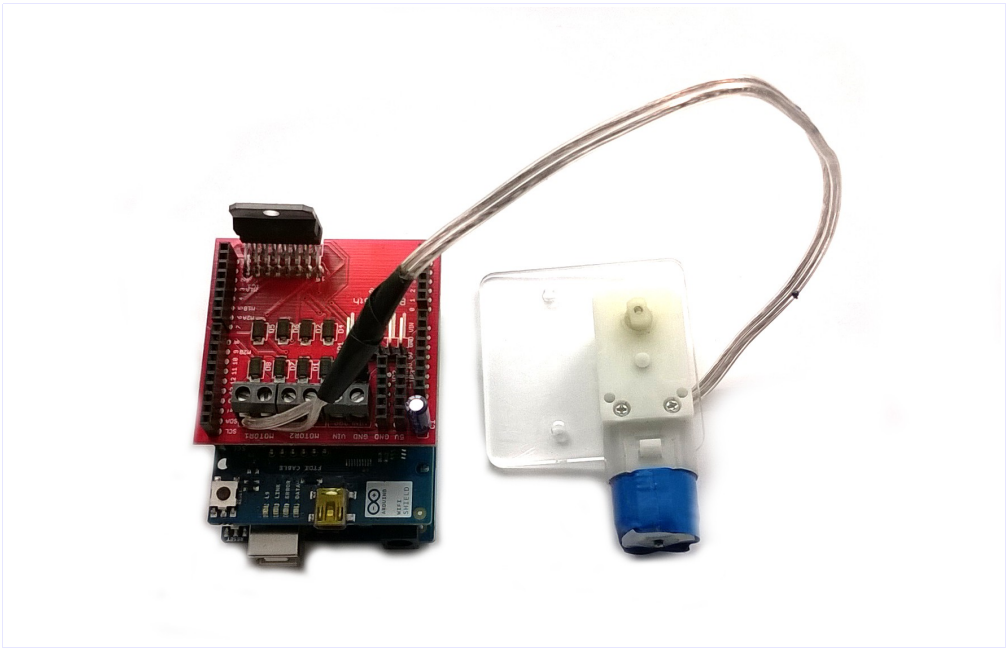
1. Shield-ul Arduino Wifi se infige in placa Arduino.



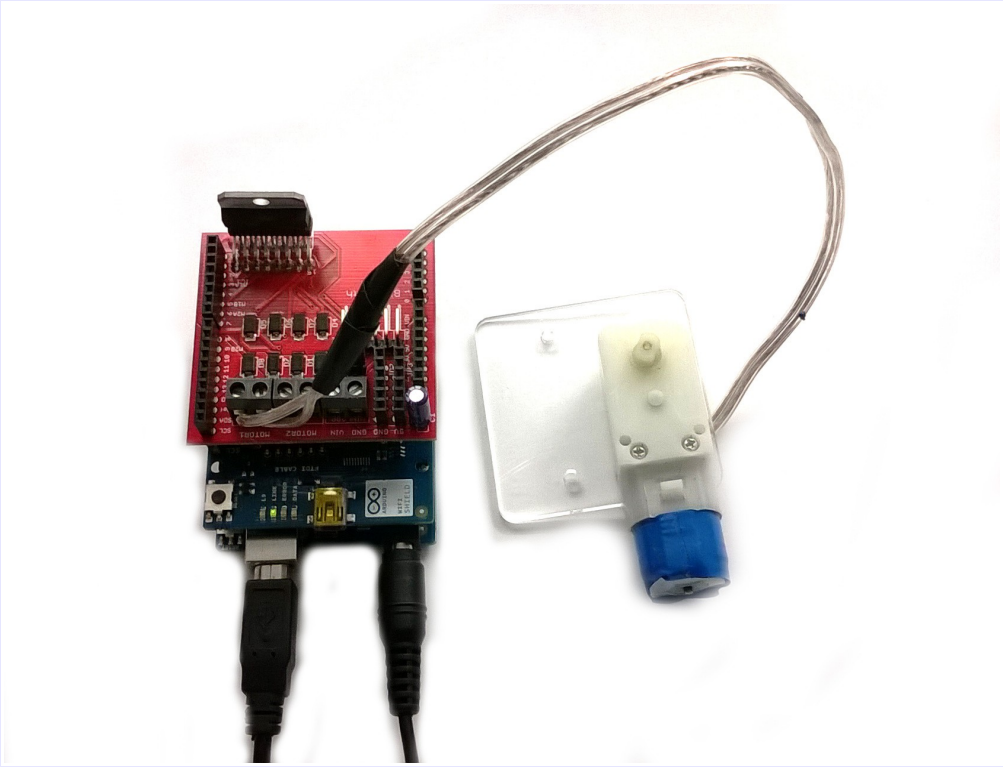
2. Shield-ul L298 se infige in pinii shield-ului WiFi.


Bluetooth
Mate
MOTOR1
MOTOR2
VIN
GND
5V
FTDI CABLE
ARDUINO
WIFI
SHIELD

3. Cele 2 fire ale motorului de curent continuu se infig in conectorul MOTOR1 si se fixeaza ferm prin strangerea suruburilor.



4. Se conecteaza cablul USB si alimentatorul extern de 9V.



Codul sursa Arduino.

Urmatorul pas este sa incarci in placa Arduino sketch-ul de mai jos. Dupa ce ai copiat codul in mediul Arduino, urmeaza sa modifici cateva linii de cod si anume:

```
char ssid[] = "default"; //  your network SSID (name)
char pass[] = "password";    // your network password (use for WPA,
or use as key for WEP)
```

Valoarea „default“ reprezinta SSID-ul routerului wireless. Modifica aceasta valoare conform routerului tau.

Valoarea password reprezinta parola routerului wireless. Modifica aceasta valoare conform parolei routerului tau.

```
#include <SPI.h>
#include <WiFi.h>
#include <WiFiUdp.h>

int status = WL_IDLE_STATUS;
char ssid[] = "default"; //  your network SSID (name)
char pass[] = "password";    // your network password (use for WPA,
or use as key for WEP)
int keyIndex = 0;            // your network key Index number
(needed only for WEP)

unsigned int localPort = 2390;      // local port to listen on

char packetBuffer[255]; //buffer to hold incoming packet

WiFiUDP Udp;

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



void setup(){
  pinMode(MOTOR1_PIN1, OUTPUT);
  pinMode(MOTOR1_PIN2, OUTPUT);
  pinMode(MOTOR2_PIN1, OUTPUT);
  pinMode(MOTOR2_PIN2, OUTPUT);
```

```
  go(255,-255);
  delay(2000);
  go(-255,255);
  delay(2000);
  go(0,0);
  Serial.begin(9600);
  //UDP Configuration
  // check for the presence of the shield:
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while(true);
  }

  // attempt to connect to Wifi network:
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open
or WEP network:
    status = WiFi.begin(ssid,pass);

    // wait 10 seconds for connection:
    delay(10000);
  }
  Serial.println("Connected to wifi");
  printWifiStatus();
```

```
  Serial.println("\nStarting connection to server...");
  // if you get a connection, report back via serial:
  Udp.begin(localPort);
}

void loop() {
  // if there's data available, read a packet
  int packetSize = Udp.parsePacket();
  if(packetSize)
  {
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remoteIp = Udp.remoteIP();
    Serial.print(remoteIp);
    Serial.print(", port ");
    Serial.println(Udp.remotePort());

    // read the packet into packetBufffer
    int len = Udp.read(packetBuffer,255);
    if (strcmp(packetBuffer,"1") == 0) {
      go(255,-255);
    } else if (strcmp(packetBuffer,"2") == 0) {
      go(-255,255);
    } else if (strcmp(packetBuffer,"3") == 0) {
      go(0,0);
    }
    if (len >0) packetBuffer[len]=0;
    Serial.println("Contents:");
    Serial.println(packetBuffer);

  }
}

void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
```

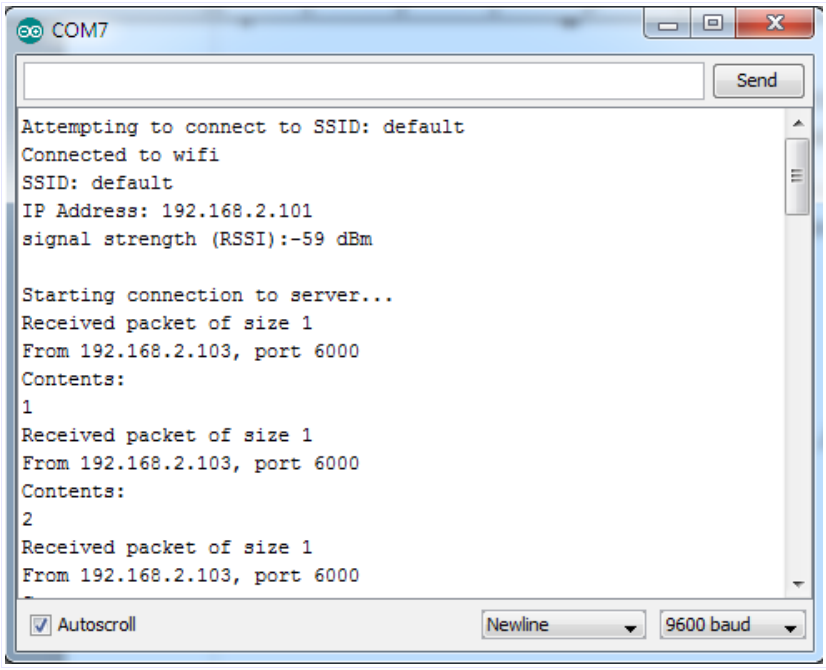
```
  Serial.println(" dBm");
}

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);
  }
}
```

Aplicatia Processing.

Urmatorul pas este sa descarci, sa instalezi si sa pornesti mediul Processing de la adresa: http://processing.org/ . Codul sursa care comanda motorul de curent continuu este listat mai jos.

Dupa ce ai copiat codul sursa, modifica urmatoarea linie cu IP-ul placii Arduino. Acesta il vei afla din Monitorul Serial ca in imaginea de mai jos:

```
String ip = "192.168.2.101"; // remote ip address
```


COM7
Send
Attempting to connect to SSID: default
Connected to wifi
SSID: default
IP Address: 192.168.2.101
signal strength (RSSI):-59 dBm
Starting connection to server...
Received packet of size 1
From 192.168.2.103, port 6000
Contents:
1
Received packet of size 1
From 192.168.2.103, port 6000
Contents:
2
Received packet of size 1
From 192.168.2.103, port 6000
Autoscroll
Newline
9600 baud

```
import hypermedia.net.*;

UDP udp;  // define the UDP object

void setup() {
  udp = new UDP( this, 6000 );  // create a new datagram connection
on port 6000
  //udp.log( true );     // <-- printout the connection activity
  udp.listen( true );           // and wait for incoming message
}

void draw()
{
}

void keyPressed() {
  String ip = "192.168.2.101"; // remote ip address
  int port = 2390; // destination port

  if (key == CODED) {
    if (keyCode == UP) {
      udp.send("1", ip, port);
    }
    else if (keyCode == DOWN) {
      udp.send("2", ip, port);
    }
  }
  else if (key == BACKSPACE) {
    udp.send("3", ip, port);
  }
}

void receive( byte[] data ) {       // <-- default handler
  //void receive( byte[] data, String ip, int port ) {  // <--
extended handler

  for (int i=0; i < data.length; i++)
    print(char(data[i]));
  println();
}
```

Concluzie.

To ce trebuie sa faci acum este sa pornesti aplicatia in Processing si sa tastezi UP, DOWN si Backspace. Apasand tasta UP motorul se va roti intr-un sens, apasand tasta DOWN motorul se va roti in sens opus si apasand tasta Backspace motorul se va opri.