

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.

WiFi Car



Realizarea unei mașini teleghidate este o provocare plăcută pentru orice pasionat de electronică indiferent dacă este începător sau veteran. Dacă soluțiile clasice RC bazate pe frecvențe subgigahertz sau bazate pe protocolul bluetooth sunt exemplificate într-o multitudine de proiecte varianta de comandă radio prin protocol WiFi este puțin mai problematică din cauza consumului și costurilor mari pentru componenta de comunicație. În cadrul lecției de față vom prezenta o soluție simplă de realizarea a unei mașini controlate prin WiFi bazându-ne pe următoarele componente:

1. Un șasiu FlexyBot cu două motoare:



<https://www.robofun.ro/kit-roboti/flexy-bot-2-motoare>

2. O placă de dezvoltare Arduino Yun ce va asigura atât partea de logică de comandă cât și partea de comunicație WiFi:



https://www.robofun.ro/wireless/wireless-wifi/arduino_yun

<https://www.robofun.ro/forum/>

3. Un shield driver de motoare bazat pe circuitul L298:



<https://www.robofun.ro/shields/shield-motoare-l298-v2>

Asamblarea celor trei componente este simplă și este prezentată detaliat în pagina produsului FlexyBot. Nu există deosebiri majore între placa Arduino Uno și placa Arduino Yun în această fază. Nu alimentați ansamblul până nu parcurgeți următorii pași ai lecției (Configurarea inițială a plăcii Arduino Yun și Alimentarea WiFi Car).

Configurarea inițială a plăcii Arduino Yun

Configurarea inițială a plăcii Arduino Yun este prezentată în materialul:

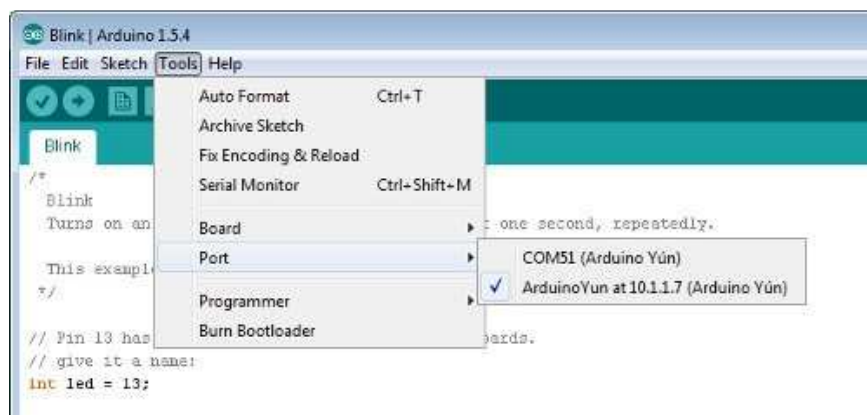
Start with Arduino Yun

<http://www.arduino.org/learning/getting-started/getting-started-with-arduino-yun>

În funcție de modul în care doriți să interacționați cu WiFi Car există două posibile configurații WiFi:

- a) Să lăsați placa Arduino Yun configurată ca WiFi AP – configurație utilă dacă doriți să accesați WiFi Car și în afara acoperirii unei rețele WiFi cunoscute, vă veți conecta cu un dispozitiv de comandă la AP-ul WiFi Car.
- b) Să configurați placa Arduino Yun să se conecteze la o rețea WiFi – această configurație are avantajul de a oferi posibilă conectivitate Internet plăcii Arduino Yun dar este dependentă de aria de acoperire a rețelei WiFi prin care se va face comanda la distanță.

Indiferent de varianta aleasă, utilizarea plăcii Arduino Yun oferă marele avantaj al posibilității programării la distanță a WiFi Car – nu trebuie să conectați, deconectați, reconectați cablul USB la fiecare nouă testare de program. Totul se poate face la distanță prin selectarea portului OTA (Over-The-Air).



Alimentarea WiFi Car

Pentru alimentarea WiFi Car se vor folosi două elemente distincte: o cutie de baterii ce va alimenta motoarele prin intermediul shield-ului și o baterie USB pentru a alimenta placa Arduino Yun. În cadrul testelor s-a utilizat o cutie cu 6 baterii AA și un USB Battery Pack de 2000mAh.



https://www.robofun.ro/surse_de_alimentare/baterii/cutie-6-baterii-cablu

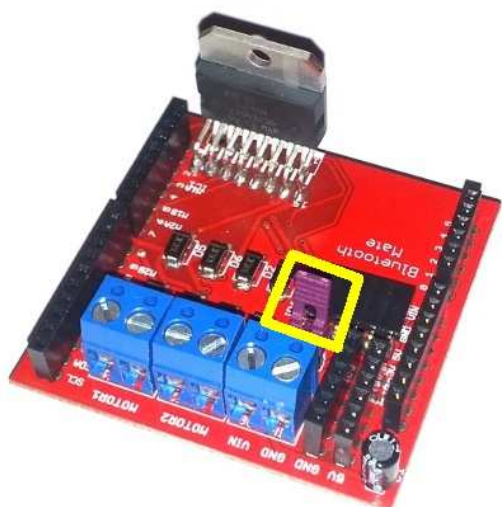


https://www.robofun.ro/surse_de_alimentare/acumulatori/usb-battery-pack-2200-mah-capacity-5v-1a-output

<https://www.robofun.ro/forum/>

Cele două surse de alimentare distincte oferă o mai mare stabilitate plăcii de comandă Arduino Yun neproducând fluctuații ale tensiunii de alimentare la manevrele de pornire / oprire a motoarelor. În plus de asta, placa Arduino Yun nu are regulator de tensiune integrat (cum are placa Arduino Uno) și din acest motiv trebuie alimentată cu o tensiune de fix 5V.

ATENȚIE!!! Conectarea jumperului de pe shield-ul driver de motoare ce permite alimentarea plăcii de dezvoltare din aceeași sursă ca motoarele va conduce la arderea imediată a plăcii Arduino Yun deoarece aceasta nu acceptă pe pinul Vin altă tensiune decât fix 5V.



NU!!!

Programarea WiFi Car

Programul ce va rula pe microcontrolerul ATmega32U4 al plăcii Arduino Yun va utiliza biblioteca software Bridge pentru a crea legătura dintre comanda motoarelor și comenzile web pe care le vom trimite prin conexiunea WiFi.

```
#include <Bridge.h>
#include <BridgeServer.h>
#include <BridgeClient.h>
```

```
BridgeServer server;
```

Configurarea și comanda motoarelor este similară cu utilizarea shield-ului în conjuncție cu o conexiune bluetooth sau un algoritm local de tip evitare de obstacole sau urmărire de linie.

```
#define MOTOR2_PIN1 3
#define MOTOR2_PIN2 5
#define MOTOR1_PIN1 6
#define MOTOR1_PIN2 9

void setup() {
    Bridge.begin();
    pinMode(MOTOR1_PIN1, OUTPUT);
    pinMode(MOTOR1_PIN2, OUTPUT);
    pinMode(MOTOR2_PIN1, OUTPUT);
    pinMode(MOTOR2_PIN2, OUTPUT);
    server.listenOnLocalhost();
    server.begin();
}

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);
    }
}

```

În cadrul secțiunii *loop()* se vor prelua, prin intermediul bibliotecii Bridge, conexiunile de comandă și se vor trimite către procedura *process()* pentru execuție.

```

void loop() {
    BridgeClient client = server.accept();
    if (client) {
        process(client);
        client.stop();
    }
    delay(50);
}

```

```

void process(BridgeClient client) {
    String command = client.readStringUntil('/');
    command.trim();
    if(command == "forward") {
        go(150,150);
        client.println("Forward");
        client.println("Vs: 150 Vd: 150");
    }
    if(command == "left") {
        go(100,150);
        client.println("Left");
        client.println("Vs: 100 Vd: 150");
    }
    if(command == "stop") {

```

```

        go(0,0);
        client.println("Stop");
    }
    if(command == "right") {
        go(150,100);
        client.println("Right");
        client.println("Vs: 150 Vd: 100");
    }
    if(command == "back") {
        go(-150,-150);
        client.println("Backward");
        client.println("Vs: -150 Vd: -150");
    }
}

```

După încărcarea programului se vor putea accesa comenzile WiFi Car de pe orice sistem ce deține un client web (browser) și este conectat la aceeași subrețea ca și placa Arduino Yun accesând următoarele URL-uri:

- Pentru mers înainte
http://ip_placa_arduino_yun/arduino/forward/
- Pentru virare stânga
http://ip_placa_arduino_yun/arduino/left/
- Pentru virare dreapta
http://ip_placa_arduino_yun/arduino/right/
- Pentru oprire
http://ip_placa_arduino_yun/arduino/stop/
- Pentru mers înapoi
http://ip_placa_arduino_yun/arduino/back/

Bineînțeles, partea de comandă poate fi îmbunătățită prin introducerea unor comenzi de accelerare, frânare sau accentuare/încetinire a virării.

Realizarea unei interfețe centralizate de comandă

Având în vedere faptul că placa Arduino Yun rulează implicit un server web (uhttp) se poate realiza o pagină html care să centralizeze accesul la URL-urile de comandă. Există două variante de copiere și localizare la nivelul sistemului de fișiere OpenWRT a fișierului html:

- Utilizarea unui card microSD și includerea fișierului html în proiectul Arduino. Copierea fișierului html se va face automat la încărcarea programului. Această variantă este descrisă în următorul material:

Arduino Yún: Intro to web server

https://create.arduino.cc/projecthub/Arduino_Scuola/arduino-yun-intro-to-web-server-6caf93

- Crearea manuală a fișierului html în linie de comandă. Aceasta este varianta propusă datorită simplității.

Se va crea un fișier *wifcar.html* în directorul */www* cu următorul conținut:

```
<html>
<head>
<style>
a.button {-webkit-appearance: button;
-moz-appearance: button;
appearance: button;
height:200px;
line-height:200px;
text-align:center;
text-decoration: none;
font-size: 50px;
color: initial;}
</style>
</head>
<body>
<a href="/arduino/forward/" class="button"
```

<https://www.robofun.ro/forum/>

```
        style="width:100%; ">FORWARDS</a>
<br />
<a href="/arduino/left/" class="button"
        style="width:35%; ">LEFT</a>
<a href="/arduino/stop/" class="button"
        style="width:30%; ">STOP</a>
<a href="/arduino/right/" class="button"
        style="width:35%; ">RIGHT</a>
<br />
<a href="/arduino/back/" class="button"
        style="width:100%; ">BACKWARDS</a>
<br />
</body>
</html>
```

Se va verifica dacă în fișierul */etc/config/uhttp* variabila *option home* are valoarea */www*:

```
# Server document root
option home                /www
```

Se va accesa URL-ul:

http://ip_placa_arduino_yun/wificar.html

Din păcate latența de comunicație a protocoalelor TCP/IP, mai ales al protocolului HTTP, este destul de mare. Nu vă așteptați să aveți aceeași viteză de răspuns ca în cazul unei mașini teleghidate RC obișnuite.

FORWARDS

LEFT STOP RIGHT

BACKWARDS