

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND

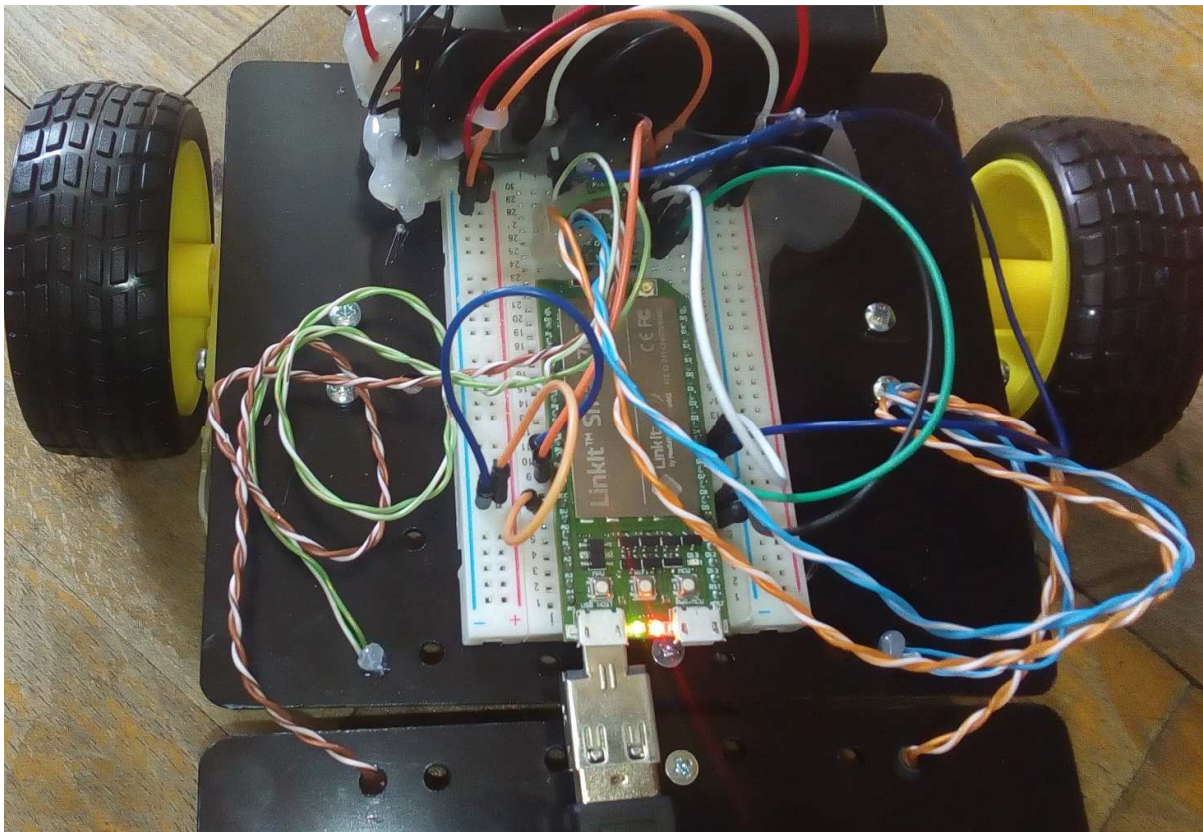


Codul sursa din acest document este licentiat

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 Reloaded (partea I)

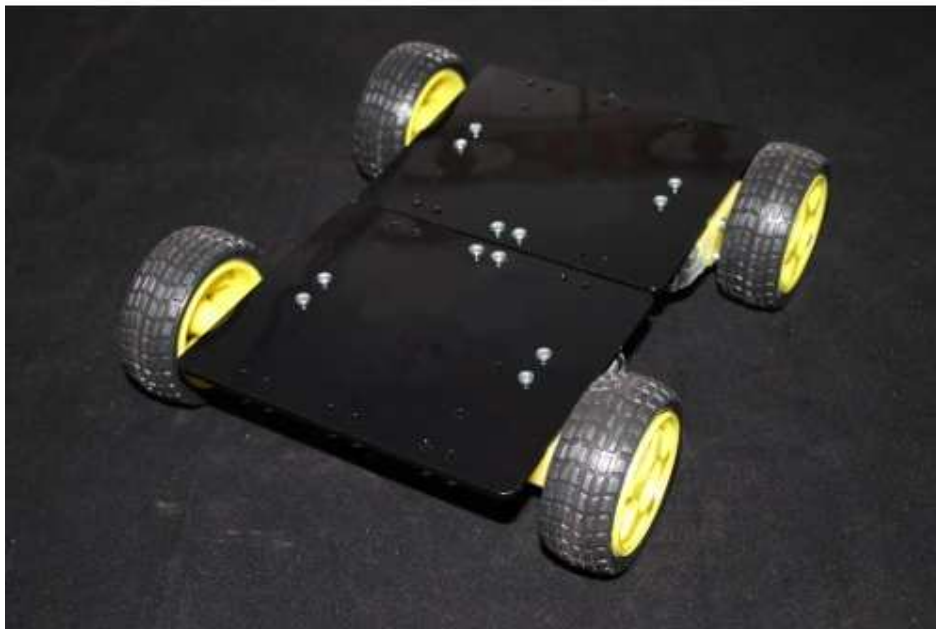


Așa cum am prezentat și în lecția precedentă (WiFi Car), realizarea unei platforme robotice comandate la distanță prin intermediul unei conexiuni WiFi se poate realiza foarte simplu prin intermediul unei plăci de dezvoltare mixte (microprocesor + microcontroler) – Arduino Yun sau echivalentă. În cadrul lecției de față vom relua aceeași provocare dar vom aduce o serie de îmbunătățiri atât la nivel hardware cât și la nivel funcțional.

Din punct de vedere hardware ne propunem să utilizăm o placă de dezvoltare similară cu placa Arduino Yun dar de dimensiuni mult mai mici, în plus nu vom mai utiliza un șasiu cu 2 motoare ci unul cu 4 motoare pentru o stabilitate crescută și o manevrabilitate mult mai bună. Din punct de vedere funcțional vom adăuga posibilitatea de control prin intermediul unei aplicații mobile și transmiterea de imagini live direct de pe platforma robotică (vom adăuga o cameră ce va transmite de pe WiFi Car).



Șasiul utilizat pentru implementare este asemănător cu cel folosit anterior dar dispune de 4 roți și 4 motoare oferind, în ciuda unui consum mai mare, o stabilitate foarte bună și o posibilitate de control al direcției superioară variantei cu doar 2 motoare:



<https://www.robofun.ro/kit-roboti/flexybot-4-motoare>

Placa de dezvoltare utilizată este o placă MediaTek LinkIt Smart 7688 Duo, placă cu aceleași facilități funcționale și performanțe ca și placa de dezvoltare de Arduin Yun dar la un preț și o dimensiune mult mai mici:



<https://www.robofun.ro/wireless/wireless-wifi/linkit-smart-7688-duo>

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

Pentru controlul motoarelor vom utiliza un driver dual de curent Pololu DRV8835 ce permite o logică de comandă între 2V și 7V și o tensiune de comandă de până la 11V și 1.2A per canal (potrivit pentru mașina noastră ce va avea o logică de comandă de 3.3V și o tensiune de alimentare a motoarelor de 7.4V și un curent de 2x300mA per canal):



https://www.robofun.ro/mecanice/driver/driver_dc/drv8835-dual-motor-driver

Alimentarea mașinii se va face prin intermediul unui acumulator LiPo de 7.4V și capacitate 1800mAh dar dacă se dorește o autonomie mai mare se poate înlocui varianta indicată cu un acumulator de capacitate mai mare. Atenție!!! Utilizarea acumulatorilor LiPo necesită anumite precauții specifice (pericol de explozie / incendiu). Pentru începători este recomandată utilizarea unei cutii de 6 baterii AA – la fel ca și în lecția precedentă – conectarea este identică atât pentru acumulatorul LiPo cât și pentru cutia de baterii.



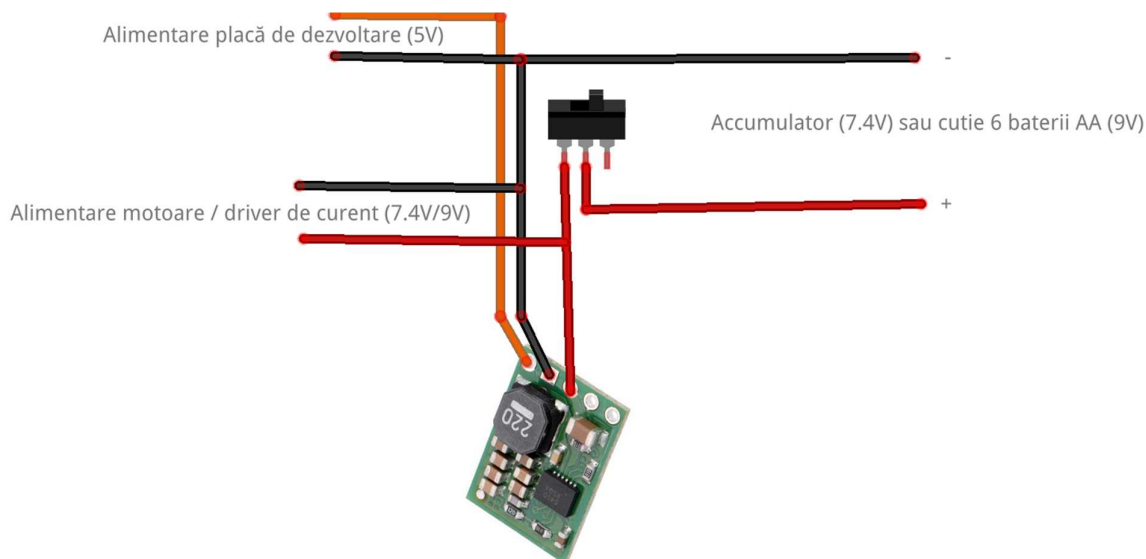
https://www.robofun.ro/surse_de_alimentare/acumulatori/acumulator-lipo-gens-1800mah

Tensiunea oferită de acumulator va alimenta în mod direct motoarele prin intermediul driverului de curent dar pentru alimentarea plăcii de dezvoltare, deoarece nu există regulator de tensiune pe placă, este nevoie de un regulator step-down de 5V extern, de exemplu:

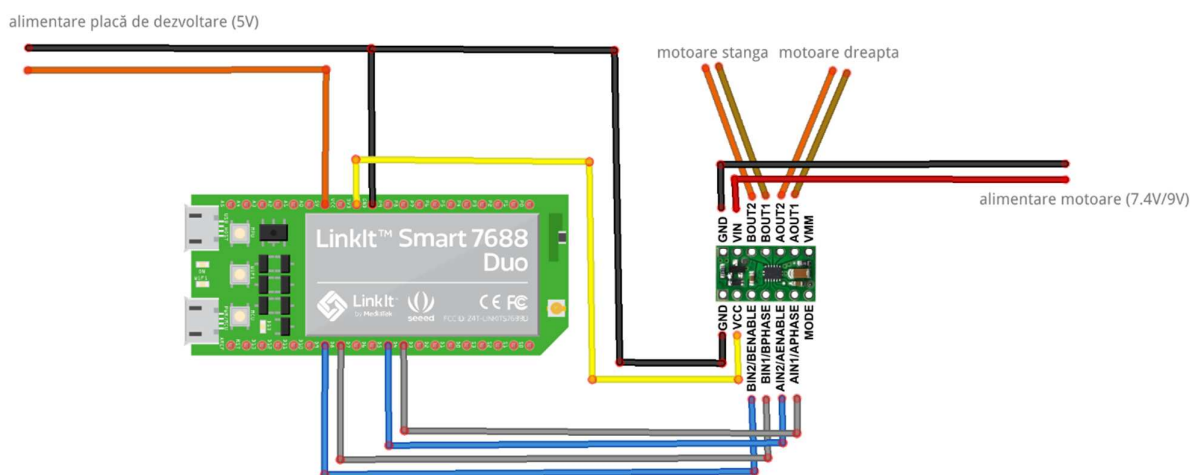


https://www.robofun.ro/surse_de_alimentare/regulator_step_down/pololu-5v-1a-step-down-voltage-regulator-d24v10f5

Astfel, sistemul de alimentare al mașinii are următoarea schemă (este inclus și un întrerupător pentru oprirea alimentării, este util dar opțional):



chema de interconectare a componentelor de comandă este următoarea:



După cum se poate vedea din schema de interconectare, placa de dezvoltare se va alimenta la 5V (ieșirea regulatorului de tensiune) iar motoarele (prin intermediul driverului de curent) la 7.4V/9V. Placa de dezvoltare dispune totuși de un regulator de tensiune intern ce coboară tensiunea de la 5V la 3.3V, ieșirea de 3.3V va fi utilizată pentru a indica driverului de curent nivelul tensiunii de comandă (spre deosebire de placa Arduino Yun care avea logica pinilor pe 5V, placa LinkIt Smart 7688 Duo are logica pinilor pe 3.3V). La ieșirea driverului de curent se vor conecta câte două motoare pe fiecare canal (cele din dreapta pe un canal, cele din stânga pe celălalt canal).

Comanda individuală a celor patru motoare nu este necesară deoarece controlul direcției se poate face foarte bine și prin comanda simultană a motoarelor de pe aceeași parte. Placa va comanda motoarele de pe partea stângă prin intermediul pinilor digitali 5 și 6 iar motoarele de pe partea dreaptă prin intermediul pinilor digitali 9 și 10 (toți cei patru pini au facilitare de ieșire PWM). Dacă, la punerea în funcțiune a mașinii, se observă anomalii de tipul: motoarele de pe aceeași parte se rotesc în sens invers sau în loc ca mașina să meargă înainte se rotește înseamnă că motoarele au fost conectate încrucișat și trebuie modificată conectarea acestora.

Noua platformă WiFi Car (Reloaded) poate fi programată în mod similar cu vechea platformă WiFi Car (placa LinkIt Smart 7688 Duo are aproape aceeași arhitectură cu placa Arduino Yun și rulează tot sistemul de operare OpenWRT) – singura modificare necesară este schimbarea definițiilor de conectare a pinilor de comandă.

Nu vom continua dezvoltarea programului din lecția precedentă ci vom propune o nouă abordare în comanda la distanță a WiFi Car. Noul sistem software va avea două componente: o componentă ce va rula pe microprocesorul plăcii (sub OpenWRT) și se va ocupa de retransmiterea traficului de rețea către portul serial al microcontrolerului și o componentă ce va rula pe microcontrolerul ATmega32U4 și se va ocupa cu comanda motoarelor conform mesajelor primite pe serială (similar cu funcționarea unei mașini comandate prin bluetooth).

Programul ce va rula pe microcontrolerul ATmega32U4 este următorul (programarea plăcii se poate face, la fel ca și în cazul plăcii Arduino Yun, prin WiFi – Over The Air):

```
#define MOTOR2_PIN1 10
#define MOTOR2_PIN2 9
#define MOTOR1_PIN1 6
#define MOTOR1_PIN2 5

int vs = 0;
int vd = 0;

void setup()
{
    pinMode(MOTOR1_PIN1, OUTPUT);
```



```

pinMode(MOTOR1_PIN2, OUTPUT);
pinMode(MOTOR2_PIN1, OUTPUT);
pinMode(MOTOR2_PIN2, OUTPUT);
Serial1.begin(9600);
}

void loop()
{
    if (Serial1.available()) {
        char c = (char)Serial1.read();
        switch (c) {
            case 'i':
                if (vs<245) vs=vs+10;
                if (vd<245) vd=vd+10;
                Serial1.print("Viteza: ");
                Serial1.print(vs);
                Serial1.print("/");
                Serial1.println(vd);
                break;
            case 'b':
                if (vs>-245) vs=vs-10;
                if (vd>-245) vd=vd-10;
                Serial1.print("Viteza: ");
                Serial1.print(vs);
                Serial1.print("/");
                Serial1.println(vd);
                break;
            case 's':
                if (vs<245) vs=vs+10;
                if (vd>-245) vd=vd-10;
                Serial1.print("Viteza: ");
                Serial1.print(vs);

```

```

        Serial1.print("/");
        Serial1.println(vd);
        break;
    case 'd':
        if (vs>-245) vs=vs-10;
        if (vd<245) vd=vd+10;
        Serial1.print("Viteza: ");
        Serial1.print(vs);
        Serial1.print("/");
        Serial1.println(vd);
        break;
    case 'x':
        vs=0;
        vd=0;
        Serial1.print("Viteza: ");
        Serial1.print(vs);
        Serial1.print("/");
        Serial1.println(vd);
        break;
    }
}

Serial1.flush();
go(vs,vd);
delay(50);
}

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);
}
}

```

Se observă că programul preia de pe portul serial Serial1 (portul serial ce face legătura între microprocesor și microcontroler) comenzile de înainte (i), dreapta (d), stânga (s), înapoi (b) și stop (x) și le transmite către motoare. Comenzile constau în modificarea (incrementarea sau decrementarea cu 10) celor două viteze (a motoarelor de pe partea dreapta și a motoarelor de pe partea stângă).

Înainte de a implementa cea de a doua componentă software a sistemului, componenta de legătură între conexiunea WiFi și comunicația serială dintre cele două circuite programabile ale plăcii de dezvoltare, trebuie să configurăm placa LinkIt Smart 7688 Duo să poată acceseze Internet prin intermediul unei rețele WiFi locale – configurația inițială a plăcii – similară cu partea de configurare inițială din lecția precedentă. Pentru configurarea de bază / inițială a plăcii de dezvoltare LinkIt Smart 7688 Duo se poate consulta următorul material:

LinkIt Smart 7688 Duo - Seeed Wiki

http://wiki.seeed.cc/LinkIt_Smart_7688_Duo/

Testarea configurației următoare a fost realizată pe o placă LinkIt Smart 7688 Duo cu firmware 0.9.4.

Componenta software ce va rula pe microprocesorul sistemului sub OpenWRT este aplicația **ser2net** ce trebuie instalată:

```
opkg update
opkg install ser2net
```

Pentru pornirea automată a aplicației la restarea sistemului vom crea fișierul */etc/init.d/ser2net* cu următorul conținut:

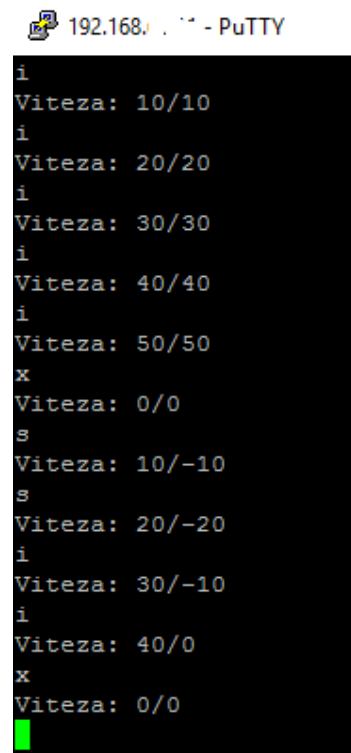
```
#!/bin/sh /etc/rc.common
START=10
STOP=15
start() {
ser2net
}
stop() {
killall ser2net
}
```

După care vom activa pornirea aplicației și vom porni efectiv aplicația:

```
chmod +x /etc/init.d/ser2net
/etc/init.d/ser2net enable
/etc/init.d/ser2net start
```

Pentru a transmite comenzile către mașină vom utiliza o aplicație client telnet (putty de exemplu) și ne vom conecta la adresa IP locală a plăcii de dezvoltare pe portul 2001. În consola telnet vom trimite comenzile descrise anterior (caracterele i, s, d, b, x) – captură de ecran alăturată.

În partea a doua a lecției vom vedea cum putem realiza comanda prin intermediul unei aplicații mobile și, bineînțeles, cum putem adăuga sistemului nostru o cameră video.



```
192.168.1.1 - PuTTY
i
Viteza: 10/10
i
Viteza: 20/20
i
Viteza: 30/30
i
Viteza: 40/40
i
Viteza: 50/50
x
Viteza: 0/0
s
Viteza: 10/-10
s
Viteza: 20/-20
i
Viteza: 30/-10
i
Viteza: 40/0
x
Viteza: 0/0
```