


Masinuta controlata prin Bluetooth

Introducere

Proiectul va consta intr-o masinuta controlata de la distanta, prin intermediul unei conexiuni Bluetooth.

Masinuta este dotata cu o camera, astfel incat utilizatorul vede pe ecranul aplicatiei atat butoanele de control ale masinii cat si imaginea real-time cu perspectiva masinutei. 

Informatii student:

- Boianiu Victor-Miron
- 332 CD
- Indrumator: Razvan Virtan

Descriere generală

Utilizatorul va interactiona cu masinuta prin aplicatia mobile, care transmite comenzile masinutei prin Bluetooth. Masinuta transmite inapoi utilizatorului un stream video.



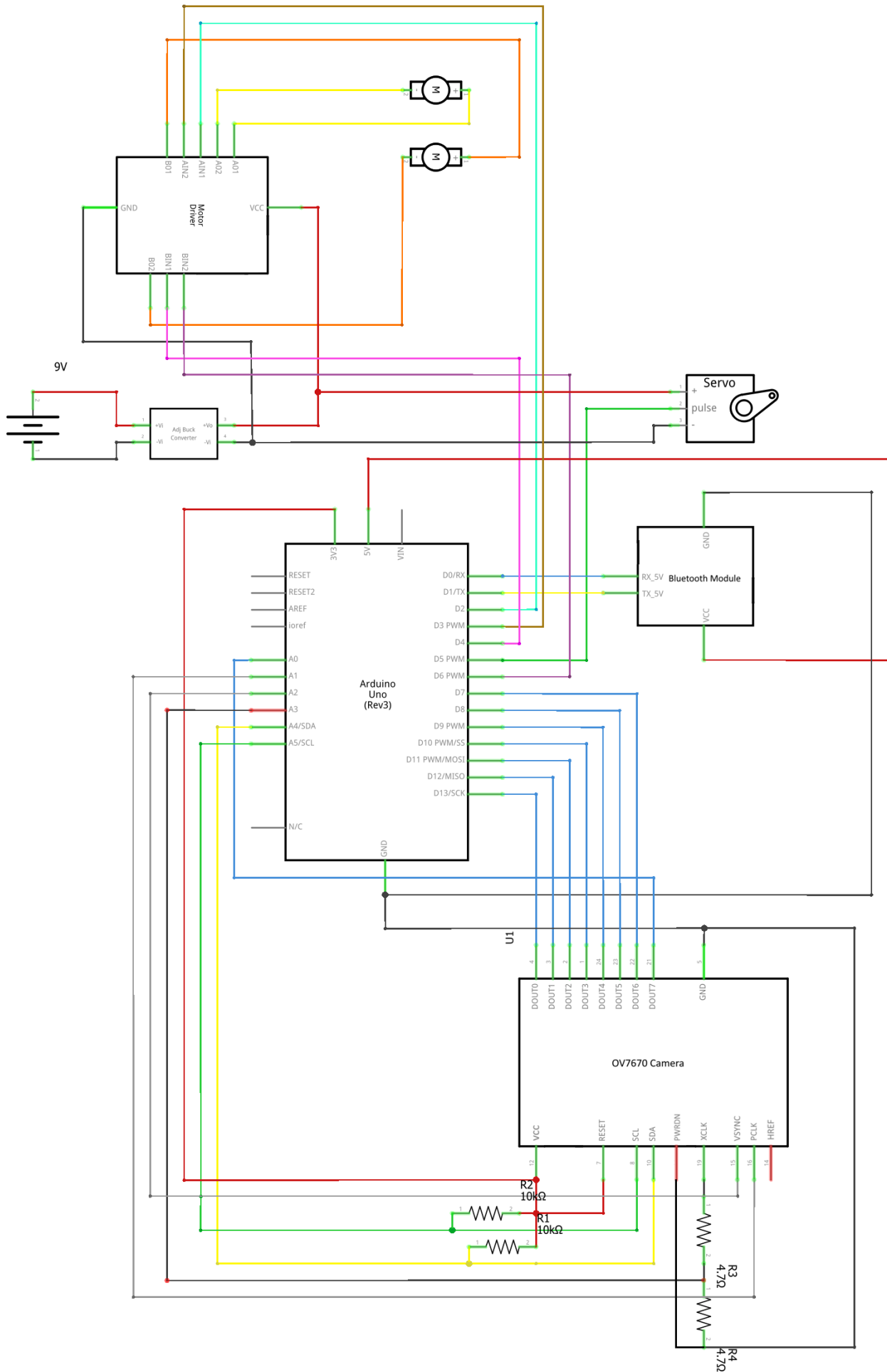
Hardware Design

Lista de piese

- 2 motorase DC;
- 1 camera OV 7670;
- 1 modul bluetooth HC-05;
- 1 driver de motoare dual L9110S;
- 1 micro servomotor 90 grade;
- 2 baterii de 9V;
- 1 sursă DC-DC coborâtoare XL4015 de 5 A;
- diversi rezistori.

Diagrame de conectare

Componentele proiectului vor fi conectate conform diagramei:



Note importante:

- Modulul HC-05 nu este legat tot la sursa externa de tensiune pentru ca acesta era restartat in momentul pornirii motoarelor;
- Rezistorii care duc de la Arduino la intrarile de la HC-05 si OV7670 sunt acolo intrucat aceste module au nivele logice de 3.3, in sens invers nefiind necesari.

Modificari importante facute ulterior:

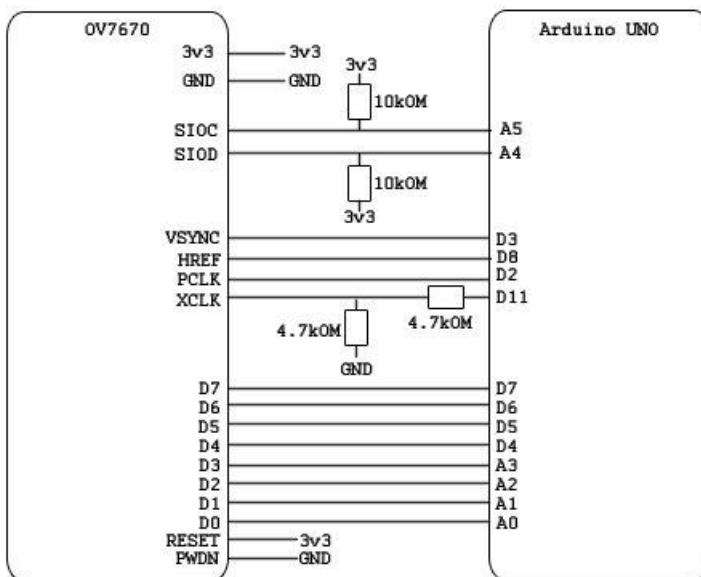
- Fiindca motorasele merg in acelasi sens si simultan, Input A de la acestea este controlat de acelasi pin. Asemenea pentru Input B.

Resurse utilizate pentru legaturi

Pentru conectarea camerei am urmarit atat diagrama de mai jos, cat si resursele urmatoare:

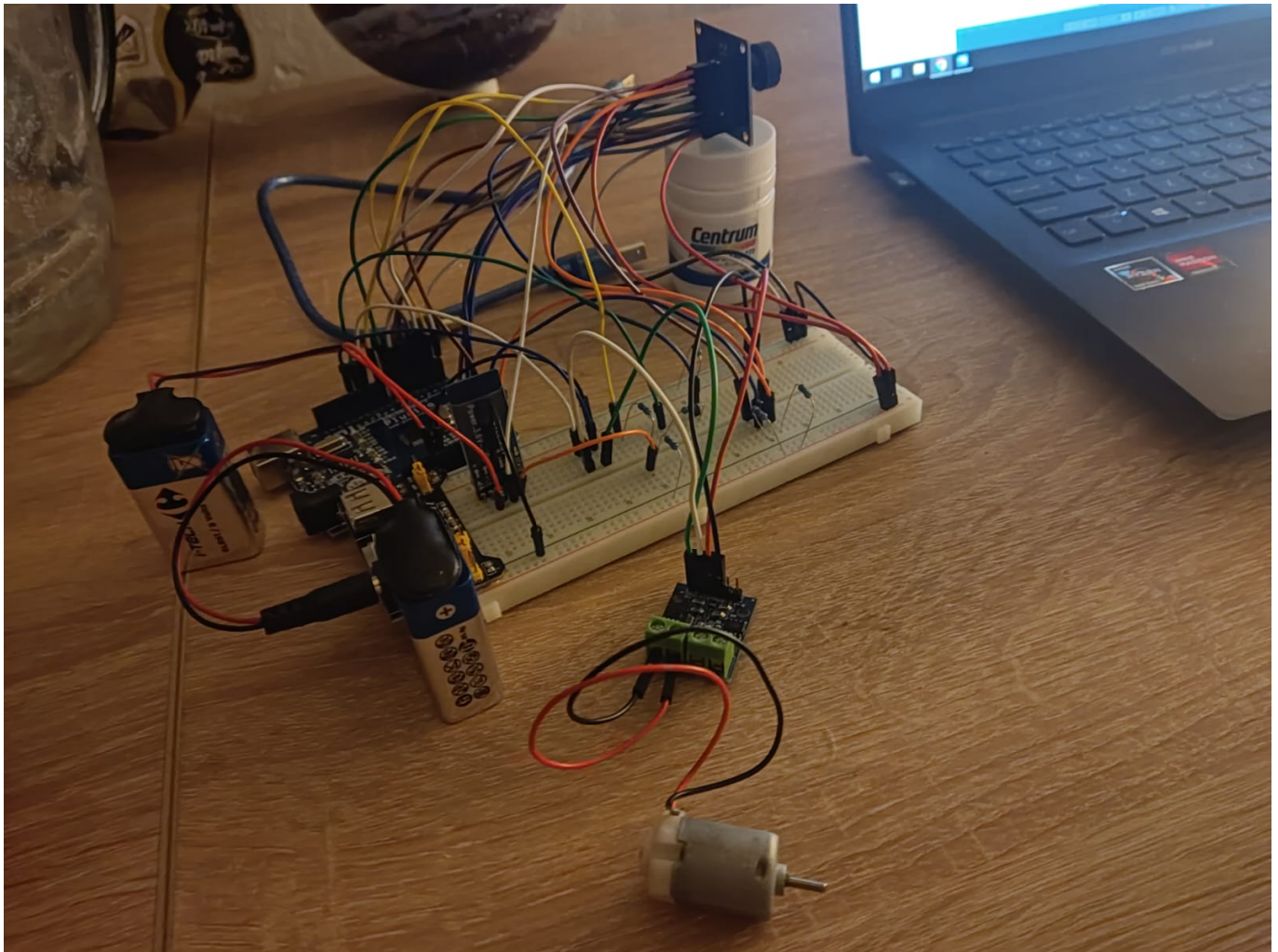
[Arduino Forum - OV7670](#)

[Youtube - OV7670 Camera Module with Arduino](#)



Stadiul actual hardware

Pentru ca partea de software este in lucru, momentan nu am montat si celelalt motor DC si servomotorul, urmand a fi montate cand software-ul este scris pentru celelalte componente.



Software Design

Link pentru repo-ul de Git: [GitHub](#)

Am decis sa fac un program de control pentru PC in loc de telefon, pentru usurarea debug-ului si pentru viteza mai mare de scriere.

Arduino

Pentru interfatarea cu camera am plecat de la [sursa aceasta](#), pastrand doar partea de cod de Arduino din tutorial, dar facand modificari majore.

- camera este acum configurata pe QQVGA, deci cu o rezolutie de 160×120 ;
- camera are formatul culorii setat pe YUV (4:2:2), dar numai componenta Y este trimisa, rezultatul final fiind o imagine monocroma;
- prescaler-ul camerei este setat la 26 pentru ca microprocesorul sa aiba timp sa trimita pixelii prin Bluetooth si sa raspunda la comenzile primite;
- conexiunea seriala (prin care interfateaza placa de dezvoltare cu modulul de Bluetooth HC-05) are baud rate-ul de 230.4K (nu am reusit marirea valorii, pastrand comunicarea stabila);

- modulul HC-05 a fost configurat in modul AT astfel incat sa aiba acelasi baud rate cu microprocesorul de la Arduino UNO, am modificat numele dispozitivului si parola de conectare.

Intrucat informatiile vizuale trebuie sa fie trimise in continuu si niciun byte primit de la camera nu trebuie sa fie ratat, am ales momentele in care tratez input-ul primit de la PC conform diagramelor de timing (vezi fig. 4, 5, 6 de [aici](#)), incercand diferite valori pentru prescaler-ul camerei pentru a asigura functionarea stabila a dispozitivului.

Nu am utilizat biblioteci suplimentare, acestea adaugand overhead, fiind astfel necesara programarea direct pe registre a microprocesorului Atmega328P pentru a evita overhead-ul si pentru a face diverse setari (de exemplu, initializarea clock-ului extern pentru camera pe un pin de PWM, analogWrite() modificand doar duty cycle-ul pe pin, nu si frecventa de pe el).

Fun fact: o parte din cod este preluata din [acest](#) driver [oficial](#) de Linux.

PC - Java

Pentru preluarea datelor de la modulul de Bluetooth, am utilizat biblioteca [BlueCove](#), prin care primesc valoarea componentei Y a fiecarui pixel, valoare care devine R, G, B pentru acesta, fiind suficienta reprezentarea monocromatica a imaginii.

Interfata grafica este creata folosind un frame de Swing peste care pun un Canvas, in care pictez manual fiecare pixel primit.

Apasarea unei taste lanseaza un event ce trimite prin Bluetooth codul ASCII al tastei apasate, dispozitivul fiind cel ce decide apoi ce actiune sa ia in functie de tasta. Desi poate ar fi fost corect sa trimit un cod corespunzator comenzii in loc de codul ASCII (pentru eventualitatea schimbarii tastelor de pe PC), codul ar fi aratat identic in rest, doar valoarea trimisa ar fi fost alta. De asemenea, comportamentul curent face sa fie usoara testarea functionalitatilor printr-un terminal.

Desi camera este setata pe format de culoare YUV, aceasta trimite doar componenta Y, fiindca am considerat ca este suficienta reprezentarea monocromatica a imaginii capturate si permitand procesarea input-ului de la utilizator in ciclul in care ar fi fost receptat si transmis byte-ul cu celelalte componente.

Prima imagine capturata de OV7670

M-am gandit ca ar fi fun sa includ si ceea ce am reusit sa capturez la prima conectare reusita a camerei: un mouse.



Nota importanta:

- Am coborat drastic rezolutia imaginii de cand a fost facuta captura asta.
- De asemenea, camera nu era focalizata, se vede surprinzator de bine la rezolutie maxima.

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/rvirtan/victor.boiangiu>



Last update: **2024/05/22 02:58**