

# Dan-Cristian SPORICI (67058) - Pendulum POV

Autorul poate fi contactat la adresa: **Login pentru adresa**

## Introducere

Proiectul constă în dezvoltarea unui dispozitiv de afisare (nu neapărat numai text), bazat pe fenomenul de **Persistence of Vision**, folosind o bară de leduri RGB - însă fără a face rotații complete ale acesteia. Scopul este unul pur educational, având utilitate redusă.

Am pornit de la structura originală (*cookie-cutter*) și am ajuns la această versiune în încercarea de a elimina o parte din problemele tehnice precum:

- necesitatea rotirii unei întregi placi (îngreunarea ansamblului)
- alimentarea separată pentru placă aflată în mișcare
- utilizarea unui motor cu nr. mare de RPM

## Descriere generală

În loc să se execute o întreagă rotație, bară de leduri se poate deplasa stânga-dreapta doar într-o zonă restrânsă unde este necesată afisarea (**90°** sau **180°**), permitând astfel utilizarea unei singure surse de tensiune și a unui simplu stepper.

Toate componente sunt coordonate de **uC**, fiind legate la pinii de ieșire ai acestuia.

În vederea afisării, **uC** va trebui să sincronizeze următoarele acțiuni:

- generația de outputuri pentru driverul de comandă al stepper-ului astfel încât să rotească bară de leduri
- generația unor semnale de comandă în funcție de care chipul de pe bară să stie ce leduri să aprindă și cu ce culori

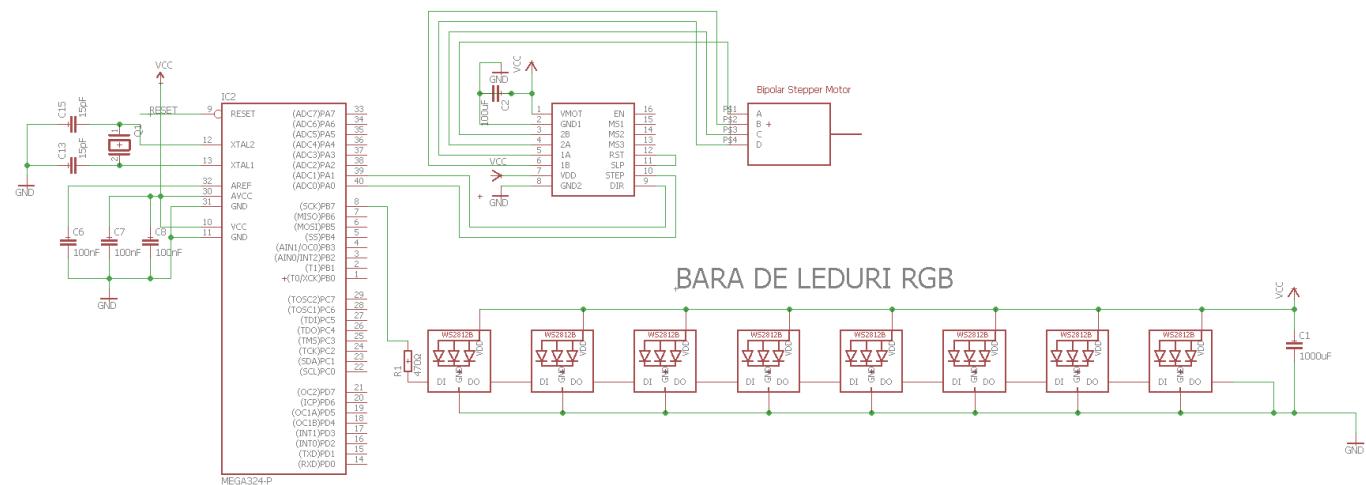


## Hardware Design

Componentă	Magazin
<Placa de la PM>	NULL

Bara de 8 LED-uri RGB WS2812 (5V)	Optimus Digital
Driver ULN2003 (5V) (incompatibil cu noul stepper)	Optimus Digital
Stepper 28BYJ-48 (5V) (prea slab / lent)	Optimus Digital
Driver Pololu A4988	Optimus Digital
Stepper Nema-17 42BYGHW609	Optimus Digital
Condensator Electrolitic 1000uF	Optimus Digital
Condensator Electrolitic 100uF	Optimus Digital

### A 2-a varianta a schemei:



# Software Design

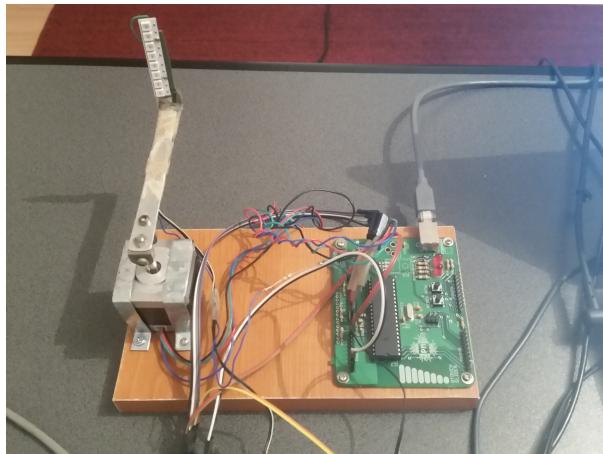
Partea de soft a fost scrisa in **Programmers's Notepad** si compilata cu **avr-g++**. In mare, se bazeaza pe un timer care genereaza intreruperi la un anumit interval de timp - iar rutina de tratare implica trimitera semnalului de **step** catre driver, ajustarea valorii din **OCR1A** si un apel catre functia ce se ocupa de configurarea ledurilor corespunzatoare pasului curent (in cazul de fata, aceeasi linie este afisata pt 2 pasi consecutivi).

Modificarea valorii de top din **OCR1A** e facuta cu scopul de a accelera/incetini stepperul la inceput, respectiv final de cursa (in ideea de a trimite mai des sau mai rar semnalul de step).

Asta se face după o funcție ce depinde de patratul numărului de pași efectuați până la momentul curent.

# Rezultate Obtinute

Functioneaza, insa n-am reusit sa obtin intocmai viteza dorita de la stepper (cel mai probabil mai trebuie modificati parametrii din functia dupa care se fac accelerarea si incetinirea) - prin urmare, afiseaza binisor text dar imaginile cu culori mai complexe sau animatii nu sunt la fel de clare.



Small demo: [https://www.youtube.com/watch?v=oiu\\_-\\_DVWuw](https://www.youtube.com/watch?v=oiu_-_DVWuw)

## Concluzii

Proiect destul de interesant, m-a ajutat sa inteleag mai bine cum functioneaza un POV si cum sa comand un stepper; totodata m-a lamurit de ce nu foloseste lumea astfel de motoare la proiectele astea...

## Download

Schema: [ds\\_pendulum\\_pov\\_sch\\_v2.sch](#)

Sursa: [pendulum\\_pov\\_src.zip](#)

## Jurnal

## Bibliografie/Resurse

- Documentația în format [PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2017/tvisan/ds\\_pendulum\\_pov](http://ocw.cs.pub.ro/courses/pm/prj2017/tvisan/ds_pendulum_pov)

Last update: **2021/04/14 15:07**

