

# Alina-Irina PÎRVAN (78709) - Osciloscop digital

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

## Introducere

Proiectul constă în implementarea unui osciloscop digital simplu. Cu ajutorul unui osciloscop putem vizualiza semnale electrice în funcție de timp.

Am divizat proiectul în 3 părți: atenuarea datelor de intrare, convertirea acestora și afișarea semnalului. Pentru atenuarea semnalului de intrare folosesc un circuit extern, format din AO-uri. Mai departe, convertesc datele din analogic în digital cu ajutorul unui ADC, pe care apoi le trimit la un PC prin USART și afișez semnalul.

## Descriere generală



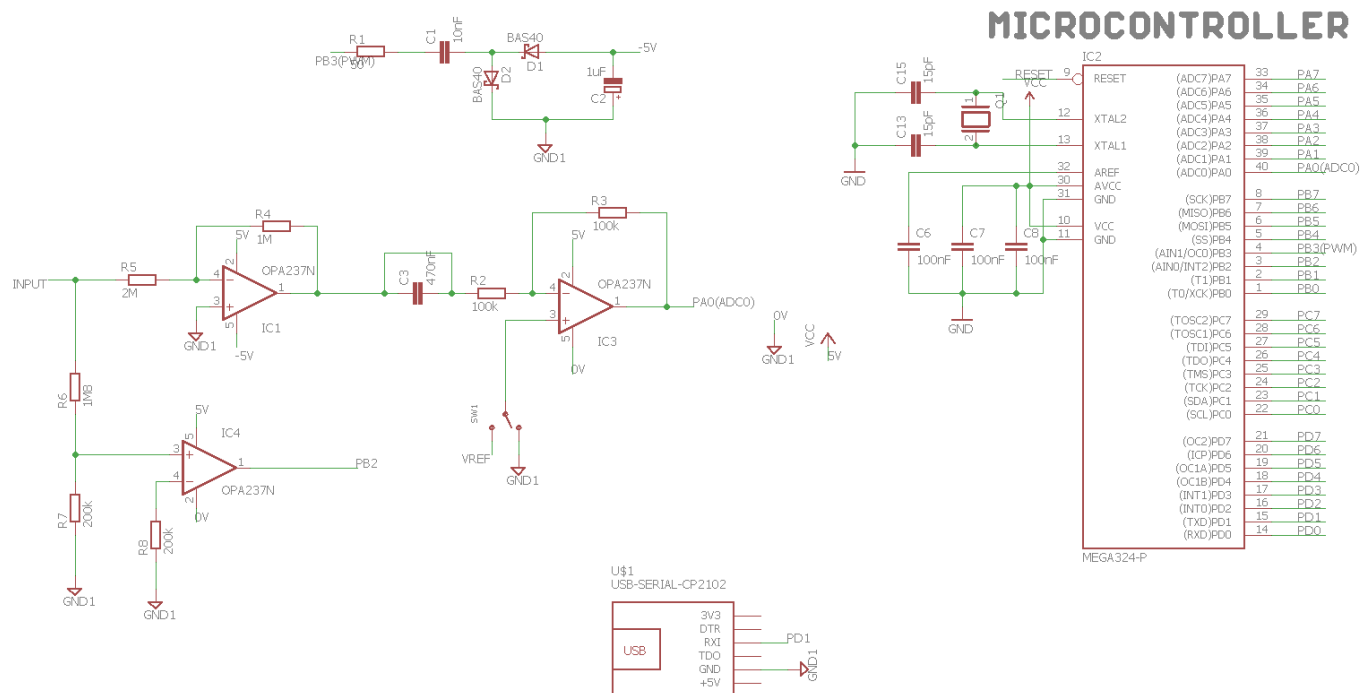
- **Input:** semnal analogic cu o amplitudine maximă de 10V vârf-la-vârf.
- **Atenuare input:** aduce semnalul la jumătate din amplitudinea sa și îl centrează în 2,5V.
- **Afișare semnal:** În urma conversiei datelor cu ADC, acestea vor fi transmise unui PC prin serială și vor fi afișate.

## Hardware Design

Listă de piese:

- placa cu piesele de baza
- convertor USB la USART
- 2 integrate cu 2 AO-uri: MCP6002
- analog switch(integrat cu 4 switchuri): DG441DJZ
- 10 rezistente
- 3 condensatoare

## Schema electrică



## Software Design

Ca și mediu de dezvoltare am ales Atmel Studio, iar ca și limbaj C++. Am folosit biblioteca de USART din laborator la care am adăugat funcții de printare a unui float sau double dintr-o altă bibliotecă de USART. În implementarea mea folosesc timer0, usart0 si canalul 0 de pe ADC.

Funcții implementate:

- void USART0\_print\_double() - funcție din usart.h care trimite un double
- convert\_from\_ADC\_to\_USART() - transformă valoarea citită de ADC înapoi în volți
- timer0\_init() - fast pwm, prescaler 1
- comparator\_init() - activare întrerupere externă pe PB2 (INT2)
- ADC\_init() - canal 0, AVCC cu condensator extern la AREF

Folosesc timer0 pentru a genera pwm la o frecvență mare. Acest pwm trece printr-un mic circuit și are ca output -5V, pe care îi folosesc la alimentarea primului amplificator operațional.

Funcția comparator\_init() îmi activează întreruperea externă de pe pinul PB2. Am nevoie de comparator pentru a distinge când semnalul este pozitiv sau negativ. Iar în funcție de asta, eu închid sau deschid cele 2 switch-uri conectate la alimentarea negativă a celui de-al doilea amplificator. Astfel, dacă semnalul e pozitiv, voi avea  $V_- = \text{GND}$ , iar dacă semnalul e negativ voi avea  $V_- = 1.25\text{V}$ .

Pentru restaurarea semnalului folosesc convert\_from\_ADC\_to\_USART() folosesc procesul invers: iau valoarea de la ADC o înmulțesc cu AVCC și o împart la 1024. Cu ajutorul acestei valori calculez semnalul de intrare în funcție de starea switch-urilor (închis/ deschis).

Flow normal al lucrurilor este următorul: inițializez USART, ADC, comparatorul și timer0. Activez

Înteruperile globale, iar în bucla while dau start conversie, aștept până când am ceva pe ADC, rețin valoarea, o convertesc înapoi în volți și o trimit pe serială.

Pentru afișarea semnalului folosesc "Serial Plotter" de la Arduino.

## Rezultate Obținute

Am reușit să obțin un osciloscop ce este capabil să măsoare semnal sinus, dreptunghiular și rampă. Acesta poate măsura și afișa semnale de frecvență maximă 100Hz, iar amplitudinea maximă 10V vârf-la-vârf.

## Demo:

<https://www.youtube.com/watch?v=etSXRH-eEvM>

## Concluzii

Este prima dată când am ocazia să realizez un proiect de la zero. Fiind obișnuită cu temele de la celelalte materii care includeau doar partea software, mi s-a părut interesant și o provocare să realizez atât partea software, cât și hardware. Ce mi se pare mai greu este partea de debug. Am stat cu multimetrul și osciloscopul pe pini pentru a vedea cum, unde, când, de ce se schimbă semnalul, de ce nu merge ceva.

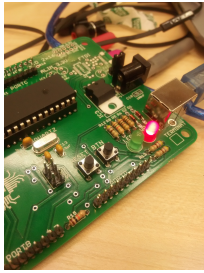
Însă, la final, văzând rezultatul muncii, pot spune că îți dă o oarecare satisfacție.

## Download

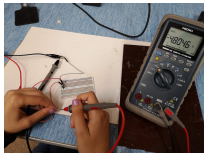
[osciloscoppm\\_alinairinapirvan\\_333ca.zip](#)

## Jurnal

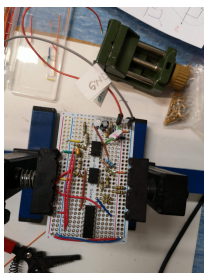
- 20 aprilie: Alegerea proiectului + creare pagină
- 26 aprilie: Lipit plăcuță + bootloader



- 10 mai: circuit ce soate -5V pentru a alimenta opamp-ul



- 11 mai: aproape gata hardware-ul



- 12 mai: finalizare hardware
- 18 mai: inceput partea software
- 22 mai: finalizare proiect software si hardware



- 23 mai: terminat de scris wiki

## Bibliografie/Resurse

### Resurse Software

- Laboratoare: <http://cs.curs.pub.ro/wiki/pm/>

### Resurse Hardware

- Amplificator operational (MCP6002): <http://ww1.microchip.com/downloads/en/DeviceDoc/21733j.pdf>
- Analog switch (DG441DJZ):  
<https://www.intersil.com/content/dam/Intersil/documents/dg44/dg441-42.pdf>
- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/cpetrisor/175939>



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