

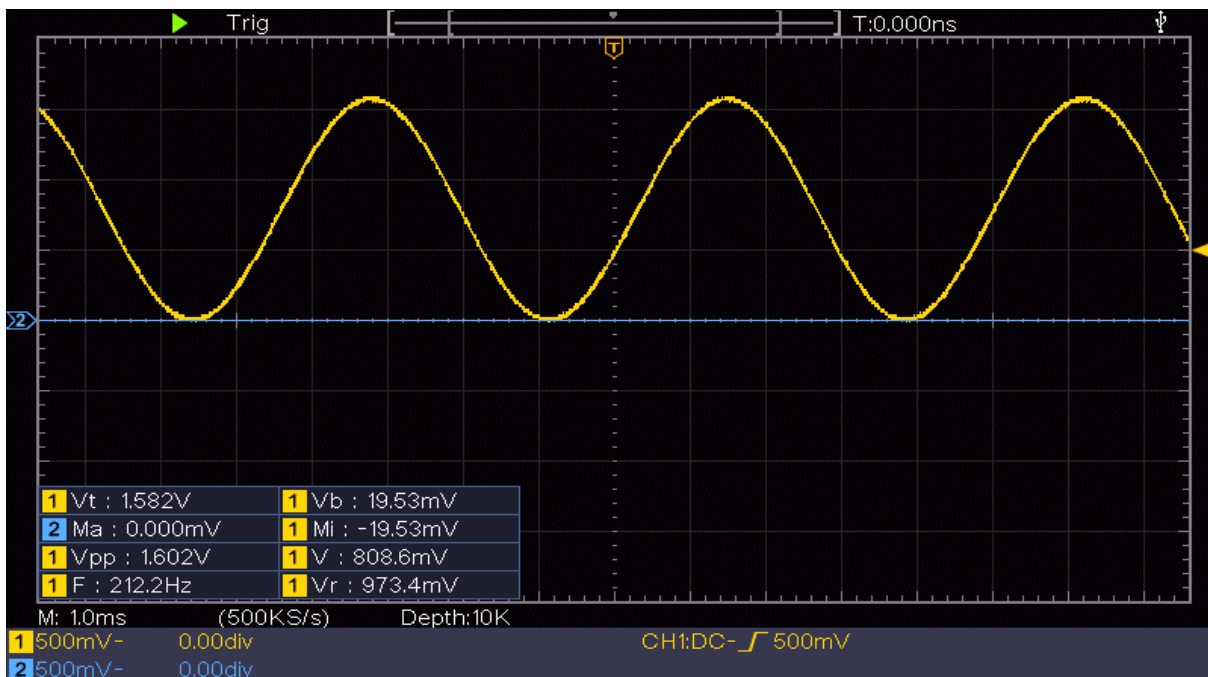
# PROCESARE DE SEMNALE DIGITALE CU AJUTORUL MICROCONTROLOARELOR

## I. Introducere

În cadrul acestei lucrări practice ne propunem implementarea și analiza unor metode de procesare a semnalelor electrice analogice sinusoidale (deci în curent alternativ) folosind microcontroloare și circuite de filtrare. Pentru realizarea acestuia ne-am folosit și de convertoare analog-digitale, convertoare digital-analogice, transformata Fourier, atât directă, cât și inversă, filtre RC fizice, dar și filtre software.

## II. Descriere

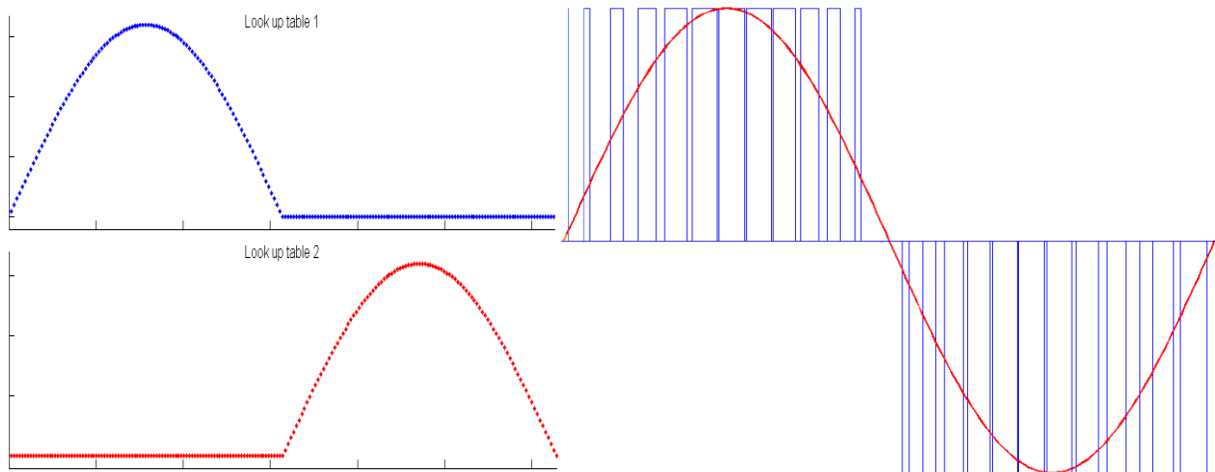
Un convertor digital-analog (DAC) este un dispozitiv electronic ce primește o valoare numerică (din software) și produce o anumită tensiune electrică raportată la acea valoare (în hardware). Discretizând un semnal sinusoidal cu o frecvență aleasă, punctele acestuia pot fi pasate unui DAC pentru a crea semnalul, de exemplu:



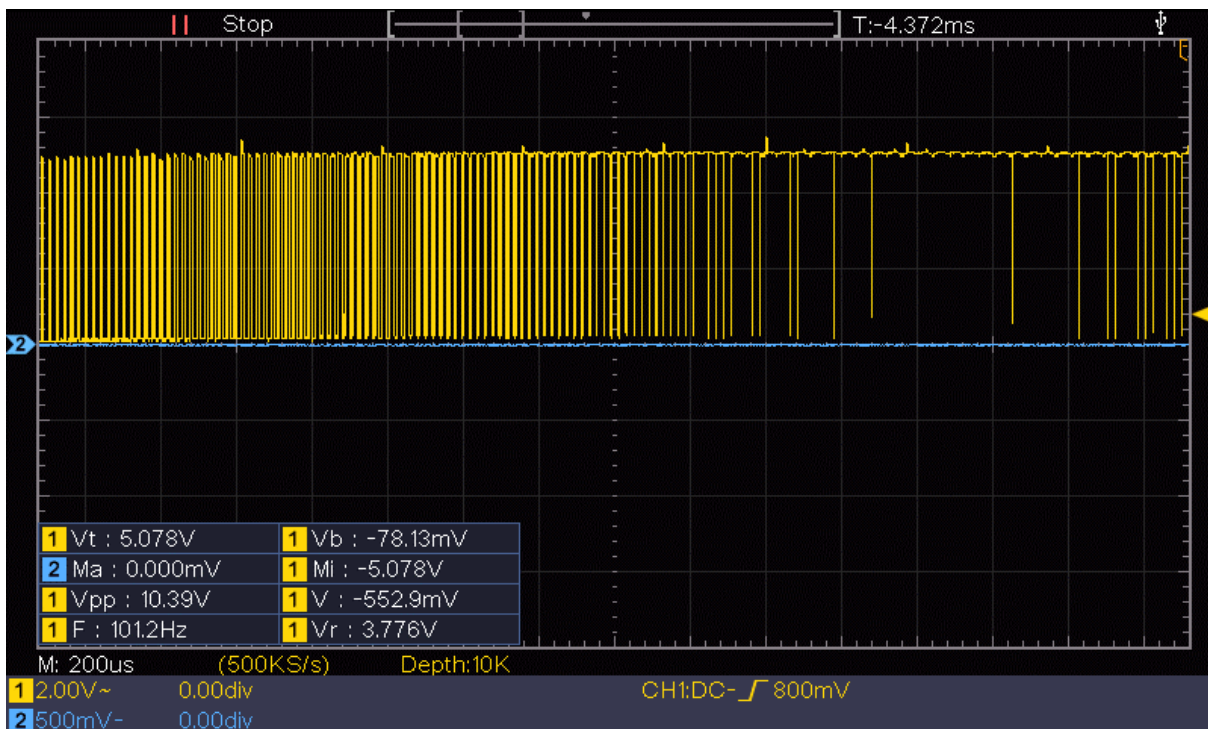
În imaginea precedentă, obținută de la osciloscop, canalul 1 (galben) reprezintă acest semnal generat de DAC, iar canalul 2 (albastru) este potențialul de masă (0 volți). Cu o precizie a DAC-ului suficient de bună, observăm că obținem o sinusoidă foarte netedă. Totuși, aceste componente (din microcontroloare) funcționează doar cu tensiuni pozitive, între 0 și o tensiune de referință. Apar aici două probleme, care fac scopul lucrării noastre: ne dorim să transmitem informație printr-un semnal sinusoidal analogic, generat de un microcontrolor, care să varieze între un potențial pozitiv și unul negativ, egale în modul (curent alternativ), și, mai important, să putem genera, procesa și recupera acest semnal fără a avea la dispoziție convertoare digital-analogice sau alte circuite analogice, precum circuite oscilatoare și punte H.

### III. Implementare

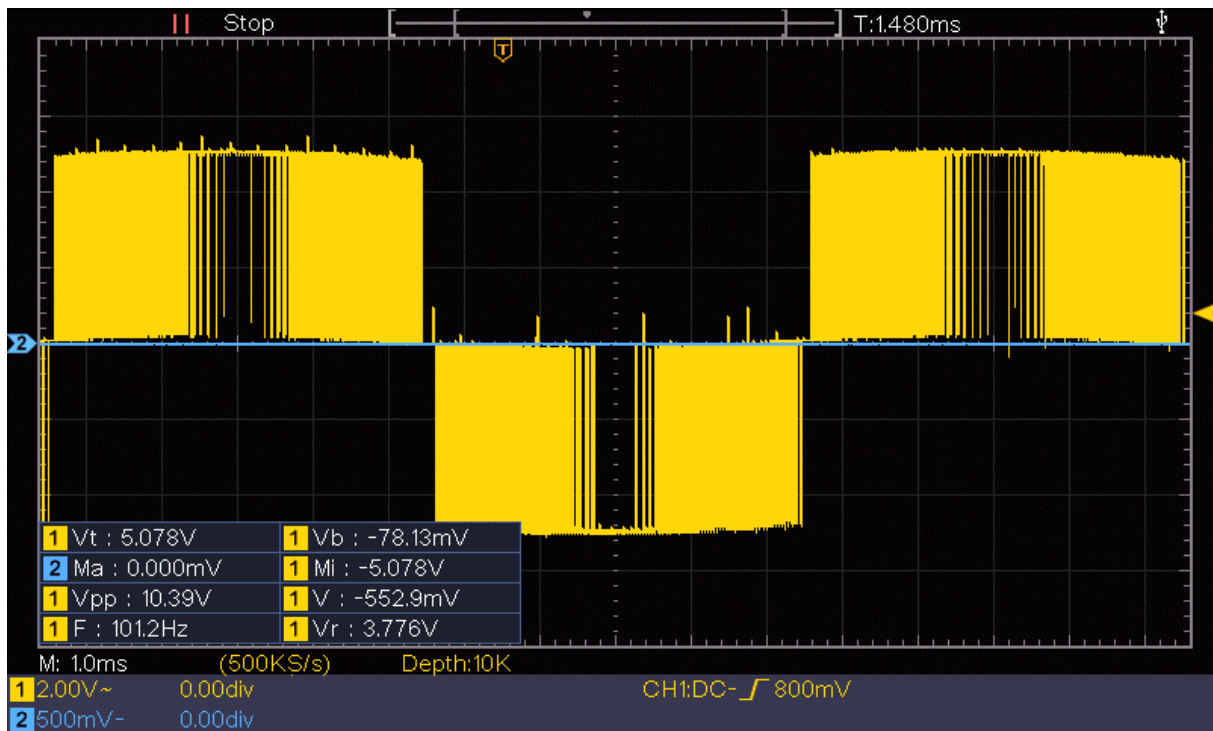
Microcontroloarele lucrează numai digital, cu două valori fixe de tensiune: 0 logic / masă / GND și 1 logic / tensiunea de alimentare /  $V_{CC}$ . Totuși, acestea au de obicei implementată o tehnologie care ne permite o formă de scriere analogică: modularea prin lățime (pulse width modulation – PWM). Aceasta presupune generarea unui semnal dreptunghiular prin alternarea valorii de ieșire între 0 și 1 logic, la o frecvență fixată, putând fi modificat factorul de umplere, însemnând procentajul de timp din perioadă în care valoarea este 1 logic. Pentru ca semnalul să fie generat corect, adică să nu sufere întârzieri cauzate de execuția altor instrucțiuni pe microprocesor, ieșirea PWM este realizată cu ajutorul mecanismului de întreruperi.



Având un microcontrolor cu capabilitatea de ieșire PWM, și, bineînțeles, cu mecanismul de întreruperi, putem genera un astfel de semnal, dar într-un mod mai puțin obișnuit, variind factorul de umplere foarte repede, reușind să creeze câte un singur dreptunghi cu acel factor. Astfel, pentru a reprezenta punctele de pe o sinusoidă, atunci când factorul de umplere este maxim (100%) avem vârful parabolei, când este minim (0%) avem punctul 0, iar toate celelalte puncte intermediare din sinus vor avea procentaje corespunzătoare.



Am ales generarea unui astfel de semnal cu frecvența de aproximativ 100 Hz (+/- eroare de precizie), având câte 512 puncte pentru o perioadă de sinus. Pentru a obține o formă de curent alternativ din acest semnal în curent continuu este necesară inversarea parabolilor sinusului din două în două, așa că vom crea două semnale, pe pini și fire diferite, primul având frontul pozitiv al sinusului și 0 în rest, iar celălalt fiind opusul acestuia (0, urmat de frontul pozitiv). Folosindu-ne de un al treilea fir, reprezentând punctul de masă (GND), când le scădem pe cele două, raportându-ne la al treilea, obținem un semnal ce oscilează între  $V_{CC}$  și  $-V_{CC}$  (în cazul nostru,  $V_{CC} = 5V$ ).



Un fenomen foarte interesant, fiind și baza acestui experiment, este că un astfel de semnal, trecut printr-un filtru trece-jos fizic, adică printr-un circuit de filtrare, ales corespunzător, va produce o sinusoidă. Până să trecem la rezultatele obținute din filtrare este necesară o descriere a diverselor tipuri de circuite de filtrare.

## IV. Filtre (trece-jos)

### A. Din punct de vedere al pasivității

#### 1. Pasive

Filtrele pasive sunt compuse din elemente de circuit pasive (rezistoare, bobine și condensatori). Neavând un multimetru capabil să măsoare inductanțe, am implementat doar circuite cu rezistoare și condensatori (RC).

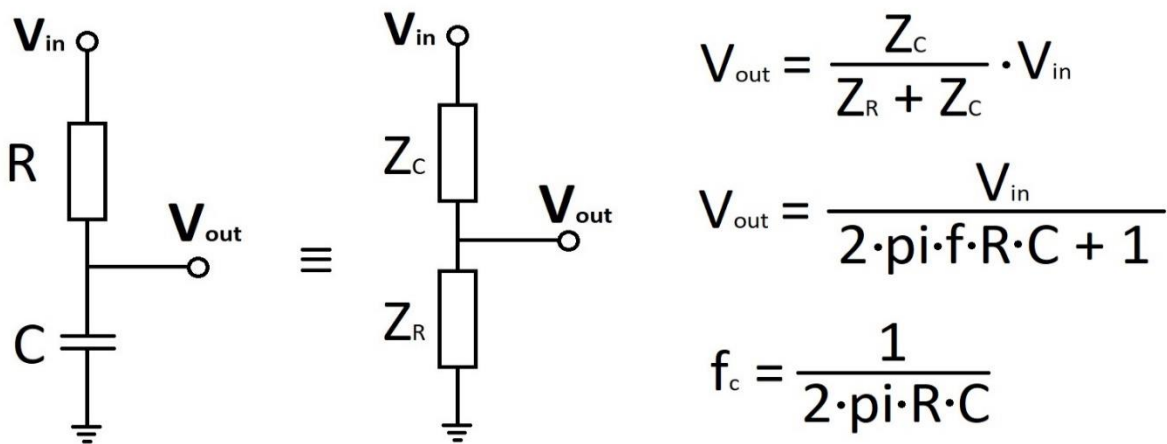
#### 2. Active

Filtrele active sunt dispozitive mai complexe, ce necesită energie suplimentară, pentru alimentare, și sunt compuse din amplificatoare operaționale, tranzistori, plus elemente pasive.

## B. Din punct de vedere al numărului de elemente pasive

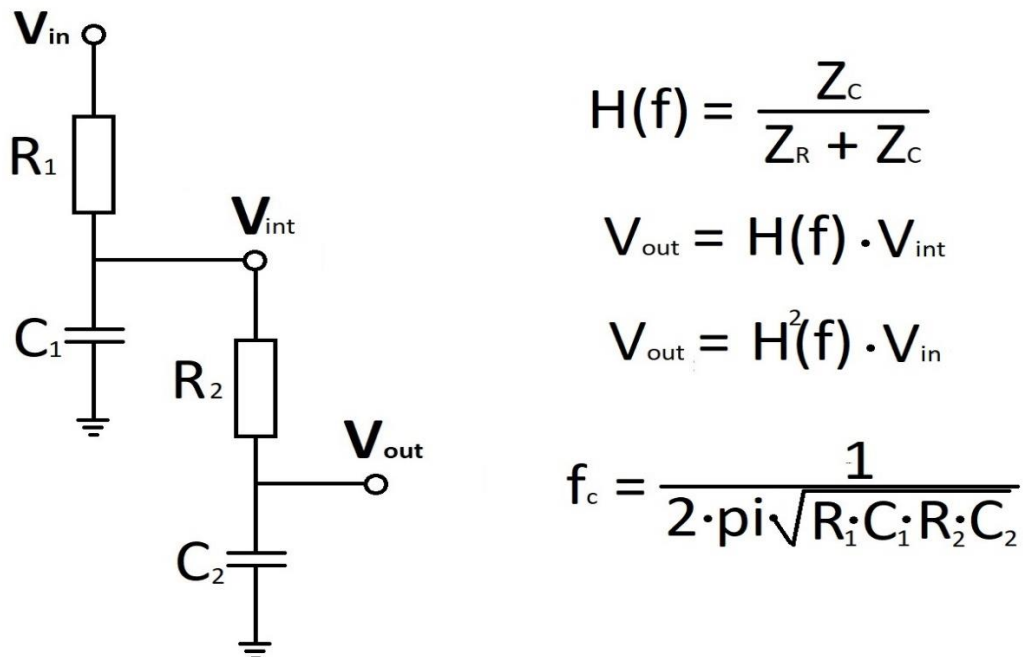
1. **De ordin inferior / întâi:** care conțin unul sau două elemente RLC
2. **De ordin superior:** compuse din mai multe elemente pasive (în plus față de cele active), înălțuite prin cascadă (practic, mai multe circuite de ordinul întâi în serie)

Analizăm, întâi, cel mai simplu filtru trece-jos, pasiv, RC, de ordinul întâi:



Schema cu impedențe este utilă pentru a privi circuitul ca pe un divizor de tensiune în curent alternativ și a deduce ușor funcția de transfer. Ne dorim să treacă semnalul sinusoidal de 100 Hz, deci frecvența de tăiere a filtrului dorit ar trebui să fie puțin mai mare.

Circuitele de filtrare pasive de ordin superior sunt construite prin înălțuirea în cascadă a mai multor filtre de ordin inferior.

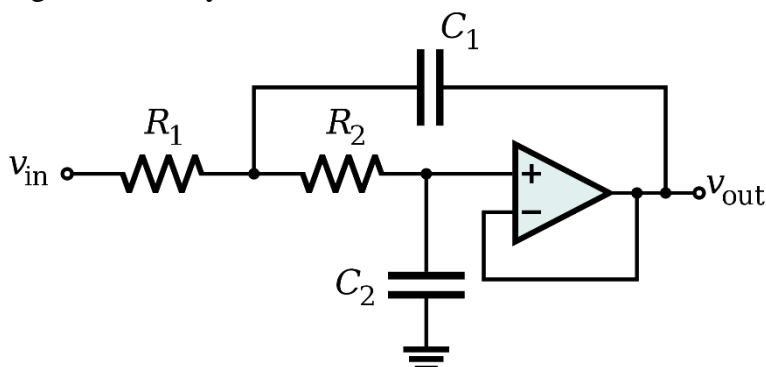


Folosind aceleași rezistoare și condensatori, ecuația finală devine identică cu cea de la filtrele de ordin inferior, și, prin analogie, pentru ordine și mai mare putem deduce la fel. Ceea ce am observat și în experiment este că în cazul menționat anterior, un filtru de ordin superior nu ne aduce niciun fel de avantaj față de cel mai simplu filtru. Totuși, va aduce un dezavantaj major, fiind divizoare de tensiune înălțuite, circuitul va reduce semnificativ amplitudinea



semnalului, iar dacă frecvența sa ar crește (sau doar am alege o alta mai mare), amplitudinea s-ar diminua și mai mult (pentru că funcția de transfer în curent alternativ depinde de frecvență.). Aceste circuite sunt utile atunci când rezistoarele și condensatorii din niveluri diferite au valori diferite, pentru că permit combinarea lor în diverse feluri, până ajungem la un rezultat satisfăcător.

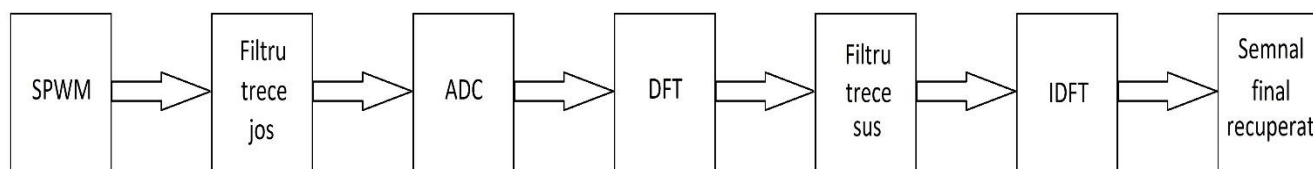
În afara filtrelor pasive de ordin 1, 2 și 3, am încercat și implementarea unui filtru activ, format dintr-un amplificator operațional, două rezistoare și două condensatoare, respectând topologia Sallen-Key.



Pentru acest circuit ecuațiile funcției de transfer și a frecvenței de tăiere sunt mult diferite față de cele de la filtrele anterioare, și mult mai complexe, deoarece, având un amplificator operațional, acesta are o funcționalitate internă complet diferită, pe bază de tranzistori ce comută. Bineînțeles, poate apărea și aici aceeași problemă a alegerii valorilor elementelor de circuit, mai ales în cazul de bază, când se poate reduce la primul filtru studiat, dar trebuie luat în calcul și că circuitele integrate cu amplificatoare sunt considerabil mai scumpe decât simple elemente RC, și, dacă nu observăm un câștig important în filtrarea obținută cu acestea, am alege soluțiile mai simple. O mențiune importantă este că filtrele active nu sunt adecvate pentru aplicația noastră, deoarece nu se comportă bine la primirea de semnale SPWM, deoarece ele trebuie alimentate cu o tensiune pozitivă și un potențial de masă (0) și pot fi folosite pentru acest tip de aplicație doar dacă semnalul este întâi măcar parțial filtrat, deci ar necesita două circuite de filtrare, în timp ce același rezultat poate fi obținut cu un singur filtru pasiv.

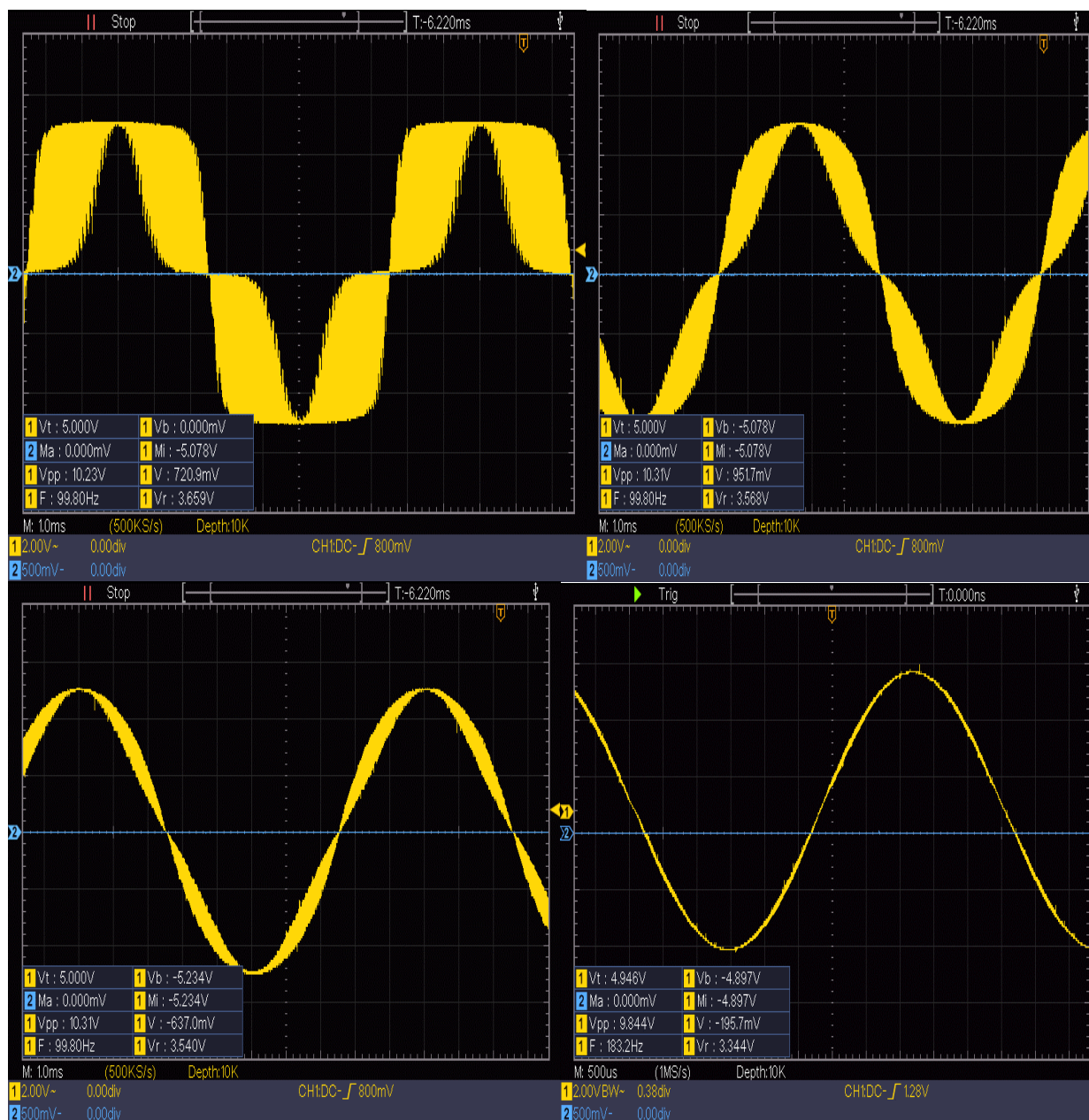
## V. Testare

Prezentăm întâi schema de lucru a experimentului:



Primul pas a fost, deci, generarea semnalului SPWM inițial, pe care l-am realizat cu ajutorul întreruperilor dintr-un microcontrolor. Al doilea pas reprezintă alegerea și crearea unui circuit de filtrare trece-jos eficient pentru semnalul de mai devreme. După cum am observat și prezentat, este suficient un filtru pasiv de ordinul întâi, fiind cel mai simplu și cel mai puțin costisitor, dar și potrivit pentru această aplicație, contând numai alegerea corectă a valorilor elementelor RC.

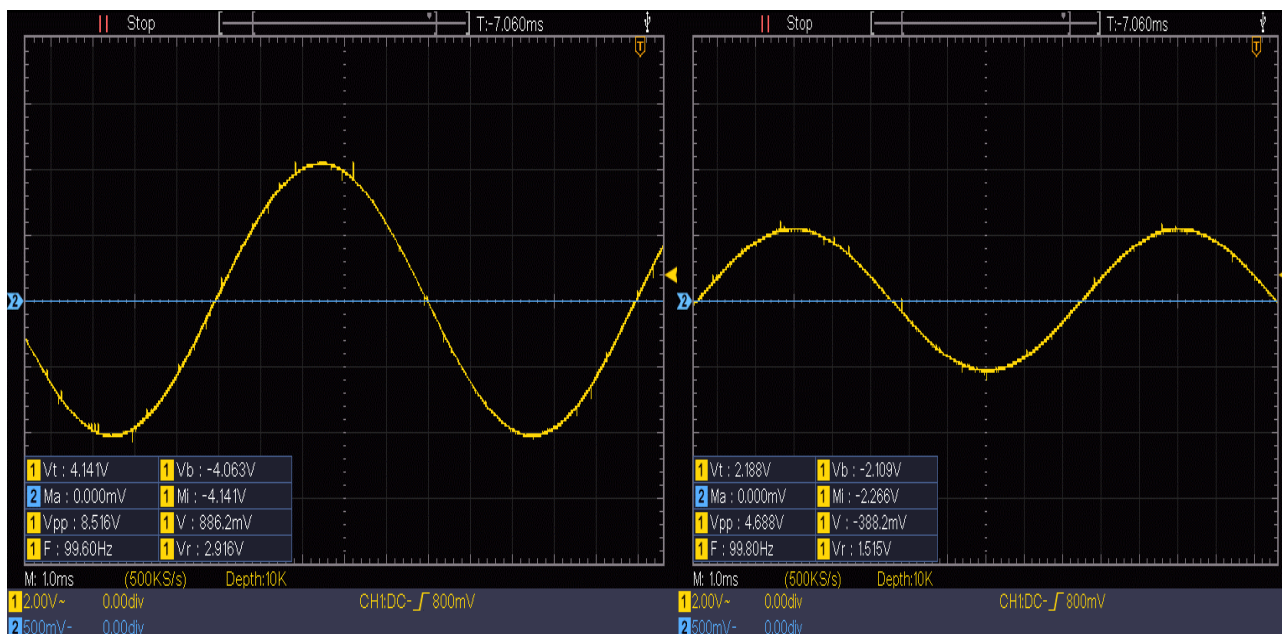
Alegând un condensator cu capacitatea de 22.4 nF și un rezistor cu rezistența de aproximativ 50 K $\Omega$ , obținem un filtru cu frecvența de tăiere de aproximativ 142 Hz, suficient pentru semnalul dorit, de 100 Hz. Pentru a observa gradul de filtrare obținut prin diverse valori RC, vom păstra condensatorul, dar vom înlocui rezistorul cu un potențiometru de 50 K $\Omega$ , pe care îl vom varia:



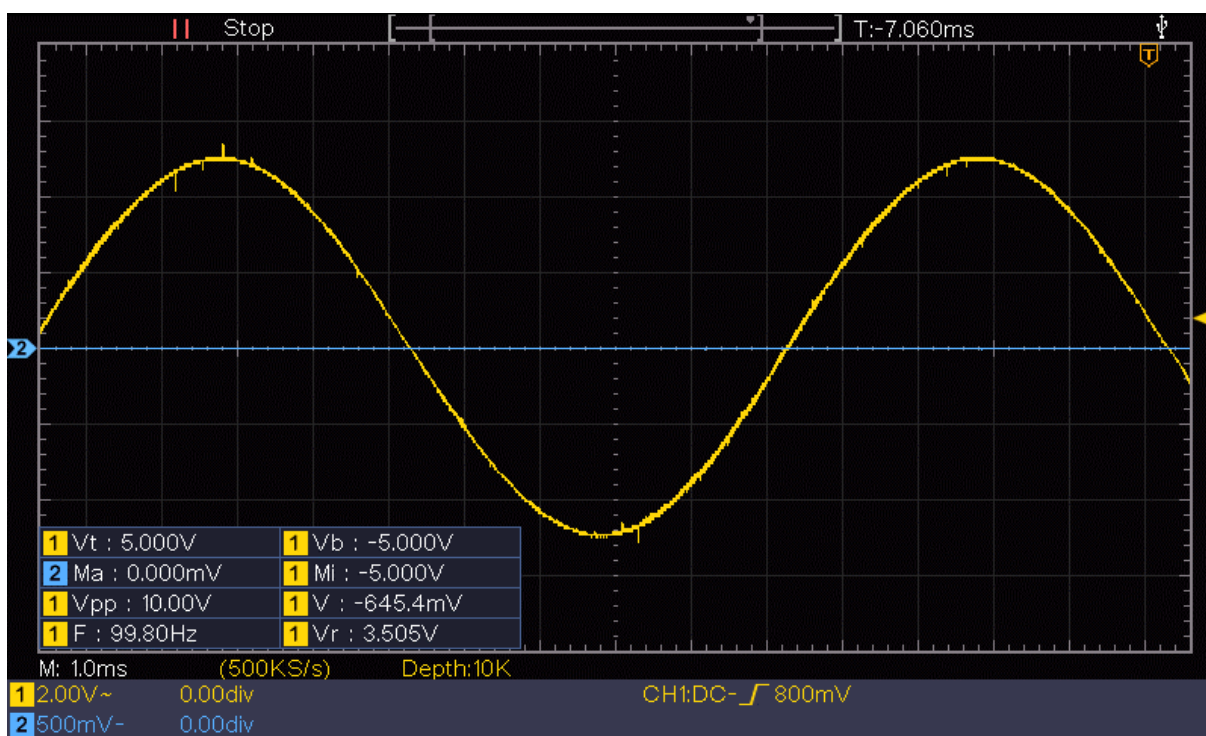
Un lucru cunoscut în domeniul ingineriei sistemelor încorporate este folosirea condensatorilor pentru filtrarea diverselor semnale, precum cel provenit de la un cristal de cuarț, un circuit oscilator sau chiar pentru tensiunea de alimentare. Analizând schema celui mai simplu filtru RC, dacă înlocuim rezistorul cu un fir (ce are o rezistență mică, dar nenulă) și condensatorul cu unul cu o capacitate suficient de mare, putem obține aceeași frecvență de tăiere. Chiar și în acest experiment, un condensator suficient de mare poate oferi, singur, o filtrare suficientă.

Totuși, luând de această dată în calcul schema cu impedanțe, având deja cu circuit ce asigură o filtrare bună pentru această aplicație, introducând un condensator mai mare, acesta nu va duce la o filtrare mai bună, ci va adăuga impedanță divizorului de tensiune, care va duce la diminuarea amplitudinii semnalului sinusoidal:

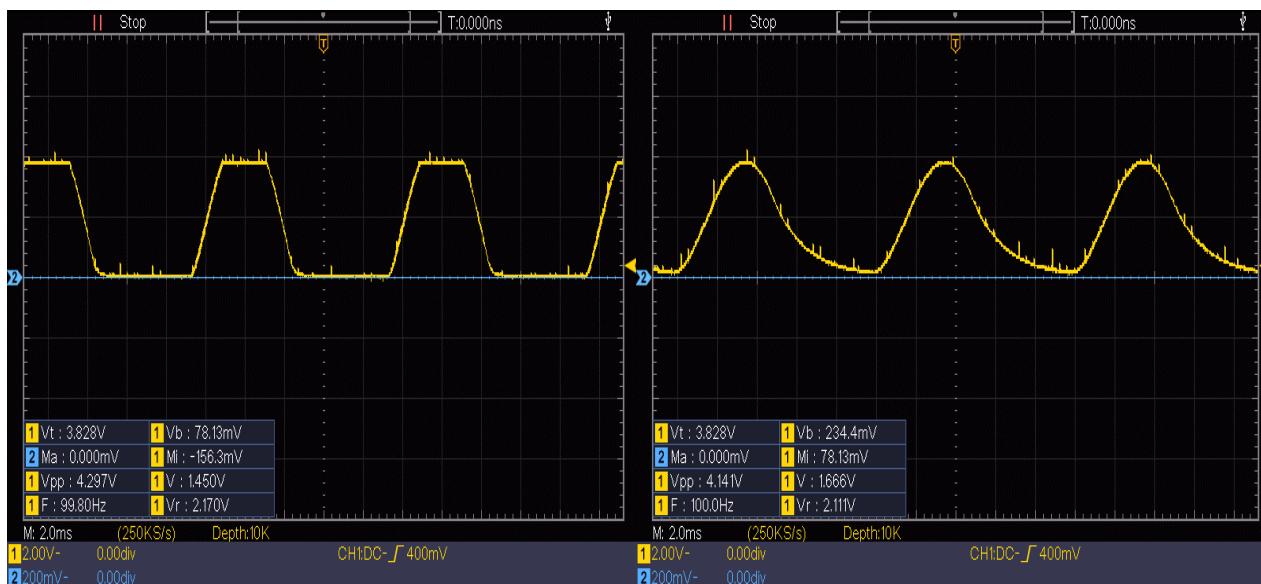




De-a lungul întregului circuit, zgomotul în amplitudine va fi inevitabil: aceasta poate suferi fluctuații (formând țepi) sau se poate diminua (dacă este conectat încă un element consumator). Cu un rezistor de 47.8 K $\Omega$  și un condensator de 22.4 nF am obținut cea mai bună filtrare, cu frecvența de tăiere de aproximativ 140 Hz, fără diminuarea amplitudinii:



După cum am menționat, un filtru activ Sallen-Key nu este adecvat pentru această aplicație, dar nu ezităm să prezentăm rezultatele obținute, mai ales pentru că am observat și un fenomen interesant: alegând pentru unul dintre rezistoare o valoare mai mică (cam la jumătate față de primul), semnalul filtrat devine trapezoidal și destul de stabil. Dacă mărim acea rezistență, obținem un semnal sinusoidal, dar distorsionat:



Având acum un semnal filtrat și destul de neted, vom folosi un convertor analog-digital de pe un alt microcontrolor, cel care va recepționa informația transmisă, pentru a recupera acest semnal. Având în vedere specificațiile microcontrolorului ales și operațiile pe care trebuie să le execute, putem ajunge la o frecvență de eșantionare la citire de aproximativ 4 kHz, însemnând câte 40 de puncte pentru o perioadă a sinusului. Această rată a eșantionării s-a dovedit a fi destul de bună, impedimentul principal venind din precizia ADC-ului, instabilitatea semnalului filtrat și a zgomotului apărut pe parcurs (o parte din zgomot era datorat și sondelor osciloscopului, care acționau ca un consumator). Ar mai exista o problemă: microcontroloarele funcționează numai cu tensiuni pozitive (curent continuu), deci fronturile negative ale sinusoidei nu pot fi citite de ADC. O soluție ar fi folosirea unei punți rectificatoare, ce să pozitiveze fronturile negative, dar, mult mai simplu, ar fi să recuperăm numai jumătate din semnal, adică fronturile deja pozitive. Deoarece sinusoida este simetrică față de abscisă, putem deplasa și inversa semnalul, apoi, scăzându-l din cel inițial, obținem o sinusoidă completă.

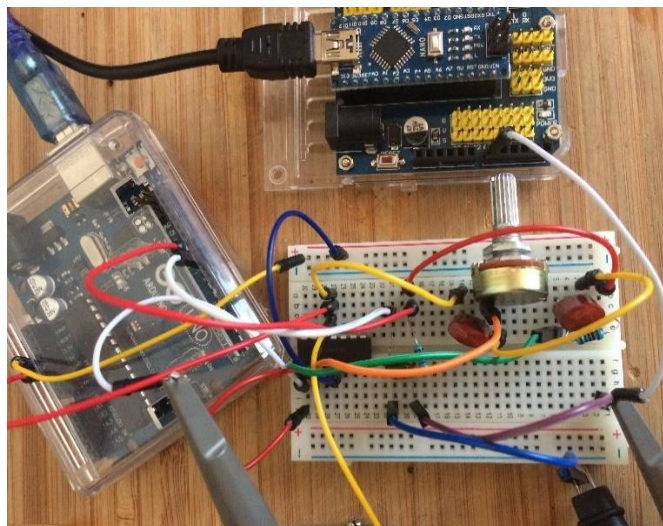
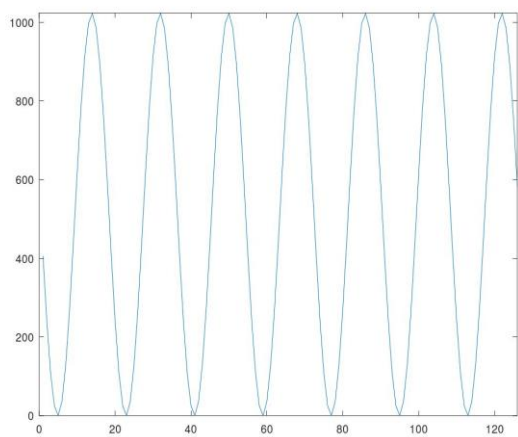
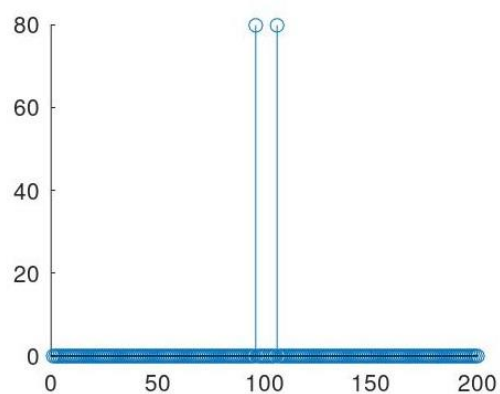
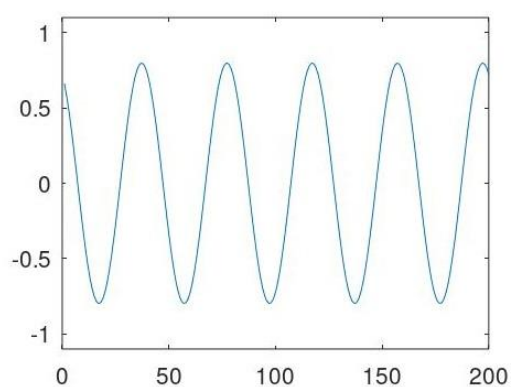
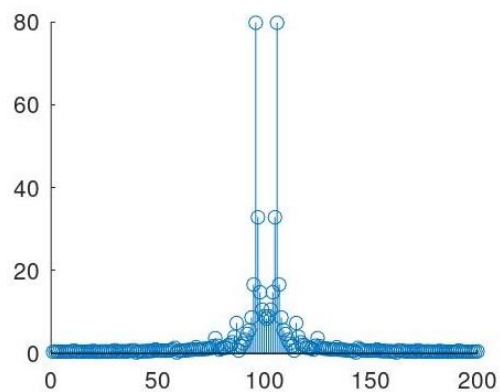
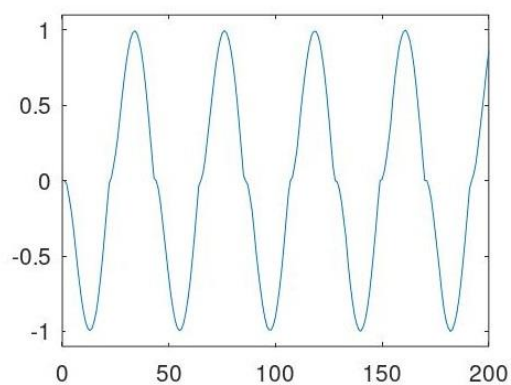
Având acest semnal citit și reconstituit, aplicăm transformata Fourier discretă (DFT) pe microcontrolor pentru a-i determina coeficienții corespunzători frecvențelor. Cum ne propusesem generarea unei sinusoide cu o frecvență fixă, ar trebui să existe numai doi coeficienți nenuli (și egali), corespunzători frecvenței așteptate și a opusei sale. În practică, din pricina zgomotului și a erorilor de precizie, apar multe frecvențe parazite mici, pe care le vom elimina cu un filtru trece-sus din software, păstrând numai acei doi coeficienți.

După ce am realizat și filtrarea din software, trecem înapoi în domeniu timp cu ajutorul transformatei Fourier inverse (IDFT), având numai coeficienții doriți, și reconstruim semnalul propus inițial, cu adevărat sinusoidal și fără zgomot, apoi îl normalizăm, pentru a avea valorile într-un interval dorit.

Pentru a putea observa, interpreta și prelucra rezultatele obținute de microcontrolor, ne-am folosit de posibilitatea de a-l conecta la consola serială prin USB la un laptop, pe care să afișăm conținutul șirurilor de numere. Cu acestea am putut realiza grafice în Octave, atât din simulări, cât și din rezultatele reale.



## VI. Rezultate



## VII. Surse de inspirație

- <https://github.com/Irev-Dev/Arduino-Atmel-sPWM>
- <https://ee-diary.blogspot.com/2020/10/generate-sine-wave-using-arduino-dac.html>
- [https://www.electronics-tutorials.ws/filter/filter\\_2.html](https://www.electronics-tutorials.ws/filter/filter_2.html)
- <https://tunahanvatansever.com/index.php/2018/06/07/analyze-of-passive-rlc-low-pass-filter/>
- <https://busylog.net/arduino-timer-interrupt-isr-example/>