UNIVERSITATEA POLITEHNICA BUCUREȘTI

FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

Proiect 2 – Electronic˘a aplicat˘a

FRECVENȚMETRU

*Autori:* Ana Neacșu, Mihai Muntean, Alexandru Guzu, Andrei Da˘escu, Dorin Ena˘chescu

*Studenți:* Ghioca Iuliana, Stanca Maria-Alexandra

An universitar: 2023–2024

# Cuprins

1. Introducere ................................................................................................................ 5
2. Resurse Hardware ......................................................................................................6
3. Resurse Software .......................................................................................................9
4. Implementare Hardware ......................................................................................... 12
5. Implementare Software .......................................................................................... 17
6. Concluzii ................................................................................................................ 21
7. Bibliografie ............................................................................................................ 22

# Lista figurilor și lista tabelelor

# -Fig. II.1 – Schema logică a plăcii Intel Galileo

# -Fig. III.1 – Fereastra principală dedicată scrierii codului

# -Fig. III.2 – Zona dedicată debugging-ului

# -Fig. III.3 – Zona dedicată instrumentelor -Fig III.4 – Zona dedicată uneltelor suplimentare -Fig IV.1 – Retea RC de tip Wien -Fig IV.2– Schema buclei de reacție negativă -Fig IV.3 – Schema electrică/analogică -Fig IV.4 – Configurație push-pull -Fig IV.5 – Schema electrică a LCD 1602 -Fig V.1 – Schema bloc a circuitului -Fig V.2 – Lcd cu pinii -Fig V.3 – Reprezentarea perioadelor pe semnalul de ieșire

# Lista acronimelor

ADC=Analog to Digital Converter

GND=Ground

GPIO=General Purpose Input/Output

I2C=Inter Integrated Circuit

IDE=Integrated Development Environment

LCD = Liquid Crystal Display

PWM= Pulse Width Modulation

SPI = Serial Peripheral Interface

TDP= Thermal Design Power

UART=Universal Asynchronous Receiver/Transmitter

USB=Universal Serial Bus

VCC=Voltage Common Collector

# Introducere

În lumea modernă a electronicii și telecomunicațiilor, măsurarea precisă a frecvenței semnalelor electrice este esențială. Un instrument cheie utilizat în acest scop este **frecvențmetrul**, cunoscut și sub denumirea de contor de frecvență. Frecvențmetrul este un dispozitiv electronic care măsoară frecvența unui semnal electric, exprimată de obicei în hertzi (Hz), indicând astfel numărul de cicluri pe secundă ale semnalului.

Acest aparat este indispensabil în diverse aplicații, cum ar fi proiectarea și testarea echipamentelor de telecomunicații, monitorizarea semnalelor de radio și televiziune, și calibrarea echipamentelor audio. În laboratoarele de cercetare și dezvoltare, precum și în industria producătoare de echipamente electronice, frecvențmetrele sunt utilizate pentru a asigura funcționarea corectă și performantă a circuitelor și sistemelor.

Proiectul nostru se concentrează pe realizarea unui frecvențmetru care poate măsura și afișa frecvența generată de un oscilator tip punte Wien. Acest oscilator, ajustabil printr-un potențiometru, va genera semnale a căror frecvență va fi precis afișată pe un ecran LCD conectat la un microcontroler. În plus, semnalul de ieșire al oscilatorului va fi redat printr-un difuzor, oferind atât o reprezentare digitală, cât și una auditivă a frecvenței.

Oscilatoarele armonice sunt circuite electronice care generează semnale de formă sinusoidală v(t)=Vsin(omega\*t), numite oscilații. Frecvența semnalului depinde de caracteristicile circuitului.

Reţeaua Wien are comportarea unui filtru trece-bandă centrat în jurul unei frecvenţe centrale şi este compusă din 2 rezistenţe şi 2 condensatoare. Trebuie făcută observaţia că în funcţie de topolgia lor reţelele RC pot fi “atacate” şi “citite” în tensiune sau curent. În acest caz expresia frecvenţei de oscilaţie devine:fosc=1/2  RC.

Scopul principal al acestui proiect este de a dezvolta un frecvențmetru funcțional și ușor de utilizat, care să fie capabil să măsoare cu acuratețe frecvența semnalelor generate de un oscilator tip punte Wien. De asemenea, prin realizarea acestuia, avem în vedere alegerea și dimensionarea corectă a componentelor necesare pentru proiectarea dispozitivului. Acest proiect își propune să atingă următoarele obiective :

1. Construirea unui Oscilator Stabil: Realizarea unui oscilator tip punte Wien care să genereze semnal sinusoidal stabil și precis, cu frecvența ajustabilă prin intermediul unui potențiometru.

2. Măsurarea Frecvenței: Implementarea unui sistem de măsurare a frecvenței semnalului generat de oscilator, utilizând un microcontroler pentru procesarea datelor și calcularea frecvenței.

3. Afișarea Frecvenței: Dezvoltarea unei interfețe de afișare pe un ecran LCD, care să prezinte în mod clar și exact valoarea frecvenței măsurate.

4. Redarea Semnalului: Conectarea unui difuzor pentru redarea auditivă a semnalului de ieșire al oscilatorului, permițând utilizatorilor să audă frecvența semnalului.

5. Documentarea Procesului: Realizarea unei documentații detaliate care să includă schema electrică, instrucțiuni de cablare, pseudocod pentru logica de funcționare și codul sursă complet compatibil cu placa Intel Galileo Gen 2.

Acest proiect nu doar că va oferi o înțelegere profundă a conceptelor fundamentale ale electronicii și măsurării frecvenței, dar va și dezvolta abilități practice esențiale pentru orice inginer din domeniul electronicii.

# RESURSE HARDWARE

# Microcontrolerul pe care l-am folosit pentru realizarea acestui proiect este placa de dezvoltare Intel Galileo Gen.2. Placa Intel Galileo Gen.2 este o placă de dezvoltare hardware bazată pe arhitectura Intel Quark SoC X1000, care combină puterea unui procesor x86 cu funcționalități de conectivitate și control digital. Aceasta a fost creată pentru a oferi un mediu de prototipare și dezvoltare rapidă a sistemelor integrate și a aplicațiilor Internet of Things.

# În ceea ce privește componentele necesare, am utilizat o placă PCB (perfboard) pentru a conecta toate componentele. Printre acestea se numără ecranul LCD pentru afișaj, difuzorul pentru redarea semnalului audio, potențiometrele pentru ajustarea frecvenței oscilatorului, amplificator operațional de tipul .... precum și rezistențe și condensatoare utilizate pentru realizarea punții Wien. Pentru a controla semnalele digitale și analogice, s-au creat conexiuni la pinii corespunzători utilizand placa Galileo Gen 2, ce reprezintă microcontrolerul central.

# 1. Placa de dezvoltare Intel Galileo Gen 2

# Placa Intel Galileo Gen.2 se remarcă ca un instrument de dezvoltare fiind usor de implementat, bazându-se pe arhitectura Intel Quark SoC X1000. Acesta îmbină puterea unui procesor x86 cu capacități extinse de conectivitate și control digital, fiind proiectat pentru a facilita prototiparea rapidă și dezvoltarea sistemelor integrate și a aplicațiilor Internet of Things (IoT).

# Punctele sale forte includ un procesor Intel Quark X1000, caracterizat de un set de instrucțiuni pe 32 de biți, cu o litografie de 32nm și o dimensiune compactă de 15x15mm, oferind un TDP (Thermal Design Power) de 2.2W și o frecvență de lucru de 400MHz. Această putere este completată de o memorie RAM DDR3 de 256MB, cu o lățime de bandă de 2.5GB/s și interfață pe 32 de biți, asigurând o performanță fluidă și receptivitate în aplicații complexe.

# În ceea ce privește stocarea și conectivitatea, placa dispune de o memorie Flash integrată de 8MB, ideală pentru stocarea codului de program și a datelor necesare. Pentru comunicare și interacțiune cu dispozitivele periferice sau alte echipamente, sunt disponibile 3 porturi USB 2.0 și un port Ethernet 10/100Mbps.

# Pentru extensibilitate și interoperabilitate, placa oferă o gamă variată de interfețe, inclusiv porturi GPIO, SPI, I2C și UART, facilitând conectarea și controlul diferitelor senzori și componente externe.

# Pini digitali (0-13), inclusiv 6 PWM (pinii 3, 5, 6, 9, 10, 11) și 6 pini analogici (A0÷A5), permit o varietate de interacțiuni cu mediul exterior, asigurând flexibilitate și adaptabilitate. În plus, portul UART oferă o soluție de comunicare serială programabilă, esențială în interacțiunea cu alte dispozitive și echipamente. De asemenea, reprezintă și un port de viteză programabil – pinii digitali (RX) și 1(TX).

# 

# Placa Intel Galileo Gen.2 este compatibilă cu mediul de dezvoltare Arduino, ce permite să fie programată și utilizată cu biblioteci și instrumente de dezvoltare Arduino.

# 

# Fig. II.1 – Schema logică a plăcii Intel Galileo

# PCB (perfboard)

# Perfboard-ul este o placă izolatoare cu orificii regulate, acoperite cu straturi subțiri de cupru, utilizată în construcția și testarea circuitelor electronice. Orificiile sunt dispuse într-o rețea de puncte uniformă, permițând montarea și conexiunea componentelor electronice prin intermediul lipirii acestora în orificii. Acest lucru facilitează crearea rapidă și flexibilă a prototipurilor de circuite, fără a fi necesară fabricarea unui cablaj imprimat dedicat.

# Display LCD

# Display-ul LCD este un ecran utilizat în dispozitive electronice, caracterizat prin utilizarea cristalelor lichide pentru afișarea imaginilor și textelor. Structura sa constă din cristale lichide între două plăci de sticlă polarizată, controlate de tensiuni electrice pentru a modifica lumina polarizată și a crea imagini. Avantajele sale includ calitatea superioară a imaginii, consumul redus de energie și dimensiunile compacte, fiind utilizat într-o varietate de aplicații, de la telefoane mobile și computere până la electrocasnice și instrumente de măsură.

# Am utilizat modulul LCD 1602 ce afișează 2 linii a cate 16 caractere, fiind un model compatibil cu modulul I2C.

1. Difuzor

Am utilizat un difuzor electrodinamic clasic(RCF de 16V și 3W) pentru voce și frecvențe înalte.

1. Rezistențe si condensatoare

Au fost utilizate pentru realizarea schemei oscilatorului de punte Wien: rezistențe de valoare 100k, ajustabile pentru fine-tuning al amplitudinii; 2 condensatoare de 10nF, 470uF pentru a amplifica semnalul la ieșire.De asemenea s-au folosit si rezistențe SMT pentru compatibilitate automata și pentru a face placa să devină mai compactă și ușoară.

1. Amplificator operational LM324

În proiectul nostru, LM324 joacă rolul unui amplificator operațional quad. În contextul specific al unui oscilator cu punte Wien, LM324 este esențial pentru generarea unui semnal sinusoidal stabil.

1. Pontețiometru dublu de tip B50K

Este folosit pentru a ajusta frecvența de oscilație. În această configurație, cele două secțiuni ale potențiometrului dublu vor înlocui rezistențele fixe din puntea Wien, permițând ajustarea frecvenței printr-o singură rotație a potențiometrului.Astfel, când ajustăm potențiometrul, ambele rezistențe variază simultan, schimbând frecvența de oscilație.

S-a folosit de asemenea pe LCD un potențiometru de reglaj pentru contrast.

1. Diode de tip 1N4148

Sunt folosite pentru a stabiliza amplitudinea semnalului de ieșire. Ele limitează amplitudinea semnalului atunci când acesta depășește un anumit prag.

1. 2 tranzistoare bipolare BC556 PNP

Utilizate ca tranzistoare de semnal în configurație push-pull pentru alternanțe negative.

# RESURSE SOFTWARE

# Arduino IDE este un software gratuit și open-source utilizat pentru dezvoltarea de proiecte bazate pe plăcile de dezvoltare Arduino. Acest mediu de dezvoltare oferă o platformă centralizată pentru scrierea, compilarea și încărcarea codului pe plăcile Arduino, fiind compatibil cu placa Intel Galileo Gen 2.

# Una dintre caracteristicile deosebite ale mediului de dezvoltare Arduino este mediul de programare software, care facilitează procesul de scriere și testare a codului. Editorul de cod simplu și intuitiv face ca programarea să fie ușor de înțeles și de implementat. Programatorii pot utiliza limbajul de programare bazat pe C/C++, familiar și puternic, și beneficiază de o bibliotecă extinsă de funcții predefinite. Aceasta acoperă o gamă largă de funcționalități, cum ar fi controlul LED-urilor, citirea și procesarea datelor de la senzori, comunicarea prin protocoale de comunicație precum I2C sau UART, și multe altele. Prin intermediul acestei abordări, programarea dispozitivelor Arduino devine accesibilă oricui.

# Versatilitatea este o altă caracteristică cheie a mediului de dezvoltare Arduino. Plăcile Arduino pot fi utilizate într-o gamă largă de proiecte, de la aplicații simple până la proiecte mai complexe, cum ar fi construirea de roboți autonomi sau dezvoltarea de sisteme de monitorizare și control automat. De asemenea, Arduino permite conectarea și interacțiunea cu o varietate de senzori, module și dispozitive externe. Această flexibilitate deschide oportunități nelimitate pentru proiectarea aplicațiilor și permite dezvoltatorilor să creeze soluții personalizate și inovatoare în funcție de nevoile lor. Un alt aspect important al mediului de dezvoltare Arduino este comunitatea sa activă și implicată. Există o comunitate vastă de utilizatori Arduino care oferă suport tehnic, împărtășesc exemple de proiecte, soluții la probleme, și stimulează schimbul de cunoștințe și idei.

# Datorită simplității și accesibilității sale, mediul de dezvoltare Arduino este potrivit atât pentru prototipare rapidă, cât și pentru proiecte mai elaborate. El încurajează inovația și creativitatea în domeniul electronicelor și automatizării, oferind posibilitatea de a da viață ideilor și de a construi proiecte cu impact.

# Pentru implementarea codului, conectarea plăcii, scrierea și rularea codului pe aceasta am utilizat Arduino IDE 2.3.2, fiind mai stabil cu placa Intel Galileo Gen.2 decât versiunile recente.

# IDE-ul pus la dispoziție de Arduino conține:

# • O fereastră principală unde se scrie codul, folosind limbajul C/C++ ;

# A screenshot of a computer program Description automatically generated

**Fig. III.1** – Fereastra principală dedicată scrierii codului

Codul Arduino este structurat în mod obișnuit în următoarele părți principale:

1. Declarațiile și inițializările globale: În această secțiune, se declară variabilele globale, constantele și obiectele utilizate în program. Aici pot fi incluse și bibliotecile externe necesare pentru diverse funcționalități.

• High/Low – sunt utilizate pentru a reprezenta nivelurile de tensiune asociate cu starea unui pin digital. `High` reprezintă nivelul de tensiune de 5V, care este considerat “1” logic sau “ON”,`Low` reprezintă nivelul de tensiune 0V, care este considerat “0” logic sau “OFF”.

• Input/Output – sunt utilizate pentru a seta pinii de intrare/ieșire.

2. Funcția setup(): Aceasta este funcția care rulează o singură dată la începutul programului, după ce placa este alimentată sau resetată. Este utilizată pentru inițializarea setărilor, cum ar fi configurarea pinilor de intrare și ieșire, inițializarea comunicației seriale sau a altor protocoale de comunicație.

3. Funcția loop(): Aceasta este funcția principală care rulează în mod repetat, formând un ciclu continuu. Aici se scrie logica programului, cum ar fi citirea senzorilor, controlul dispozitivelor sau gestionarea comunicației.

4.Funcții suplimentare: În afară de funcțiile setup() și loop(), pot fi definite funcții suplimentare pentru a structura mai bine codul și a facilita reutilizarea acestuia. Aceste funcții sunt de obicei scrise pentru a îndeplini sarcini specifice și sunt apelate din interiorul funcțiilor setup() sau loop(), cum ar fi instrucțiuni de decizie (if, else, switch), instrucțiuni de buclă (for, while), instrucțiuni de întrerupere (interrupts).

5. Funcții personalizate: În codul Arduino, poți defini și crea funcții personalizate pentru a organiza mai bine codul și pentru a realiza sarcini specifice. Aceste funcții pot fi apelate din funcția loop()sau din alte funcții și pot include orice secvență de cod:

• pinMode()– stabilește rolul unui pin (intrare/ieșire);

• digitalWrite()– setează nivelul de tensiune al pinului digital;

• digitalRead()– preia valoarea intrării pinului digital;

• analogRead()– preia valoarea intrării analogice;

• delay() – oprește programul pentru timpul specificat în milisecunde dat ca parametru;

Prin respectarea acestei structuri, codul Arduino devine organizat și ușor de înțeles, facilitând dezvoltarea și întreținerea proiectelor.

• O zonă de debugging, respectiv de comunicare a erorilor de compilare:

A black background with white text

Description automatically generated

**Fig. III.2** – Zona dedicată debugging-ului

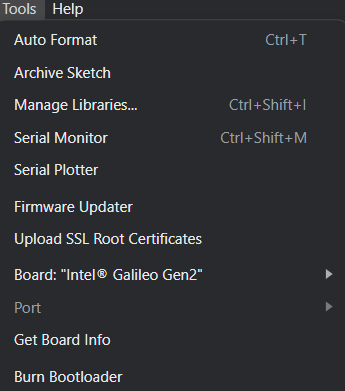
• Zona de instrumente, cu ajutorul cărora se compilează, încarcă codul și ajută la comunicarea cu

microcontrolerul, prin intermediul porturilor seriale :



**Fig. III.3** – Zona dedicată instrumentelor

• O serie de unelte care ajută la afișarea anumitor date



**Fig. III.4** – Zona dedicată uneltelor suplimentare

# IMPLEMENTARE HARDWARE

# Atât placa de dezvoltare, cât și circuitul sunt alimentate cu o tensiune de 5V.

# Se utilizează o schemă de amplificator având o rețea de reacție pozitivă și o rețea de reacție negativă, la acestea se adaugă un circuit de stabilizare a oscilațiilor (AGC) realizat cu un tranzistor TEC-J.

# 

# Oscilatorul armonic este un circuit electronic care generează un semnal sinusoidal, pe baza energiei furnizate de sursă de alimentare fără a avea nevoie de un semnal la intrare.

# 

# Rețeaua Wien este o rețea RC selectivă cu funcție de transfer de tip trece-bandă, folosim o astfel de rețea deoarece este ușor de implementat și permite reglarea frecvenței de oscilație prin utilizarea unui potențiometru dublu în loc de cele două rezistoare.

# A diagram of electrical circuits Description automatically generated Fig IV.1-Retea RC de tip Wien

# Condiția de oscilație

# Amplificatorul cu reacție pozitivă devine oscilator dacă fără a avea un semnal aplicat la intrare se obține semnal la ieșire, ceea ce echivalează cu conditia:

# Ar=X2/Xi → ∞

# Astfel se deduce conditia lui Barkhausen pentru amorsarea oscilațiilor:

# βA=1

# Din această condiție se determina amplificarea minima necesara producerii oscilatiilor.

# Pentru a determina frecventa de oscilatie scriem urmatoarea conditie de faza:

# φA + φβ = 2k𝜋

# In majoritatea situatiilor practice, amplificarea A este un numar real, deci defazajul φA = 0 sau φA = 𝜋. In acest caz frecventa de oscilatie este determinata numai de reteaua de reactie.

# Calcularea rețelei Wien

# Factorul de transfer al retelei Wien are valoarea β=1/3, stiind conditia pentru amorsarea oscilatiilor Aβ=1 rezulta ca amplificarea circuitului trebuie sa fie egala cu 3.

# Bucla de reactie negativa este cea care stabileste la o valoarea fixa amplificarea in tensiune, valoarea amplificarii este data doar de rezistoarele RG si RF dupa urmatoarea formula:

# A=1+RF/RG

# Pentru indeplinirea conditiei lui Barkhausen rezulta ca RF trebuie sa aiba o valoare dubla fata de RG astfel incat amplificarea in tensiune sa fie cel putin 3.

# Daca fixam valoarea lui RG=10kΩ rezulta ca RF trebuie sa aiba o valoarea de aproximativ 20kΩ, in realitate valoarea acestuia trebuie sa fie putin mai mare decat valoarea calculata teoretic pentru a se realiza amorsarea oscilatiilor.

# A diagram of a circuit Description automatically generated

**Fig IV.2**-Schema buclei de reacție negativă

Frecventa de oscilatie se calculeaza dupa formula:

f=1/(2𝜋RC)

In general pentru calcul incercam sa dam condensatoarelor o valoare fixa si sa dimensionam rezistorul R astfel incat sa rezulte frecventa dorita.

In locul unui rezistor de valoarea fixa se poate pune un potentiometru dublu pentru a face variabila frecventa de oscilatie.

Calculam elementele retelei pentru valoarea dorita a frecventei de oscilatie, stabilim valoarea condensatorilor C3=C6=10nF.

Astfel putem scrie urmatoarea formula pentru rezistori:

R=1/(2𝜋Cf)

Se scrie formula de doua ori, pentru Fmax=6kHz si Fmin = 300Hz,

Rezulta astfel Rmin=3.3k si Rmax=53k daca folosim condensatori de 10nF.

Astfel rezulta ca vom folosi cate un rezistor de 3.3kΩ pentru R35 si R36, se vor utiliza capsule 0805 pentru toate componentele pasive din circuit. Pentru potentiometre se vor utiliza valori de 50k.

# 

# Fig IV.3-Schema electrică/analogică

# 

# Funcționarea circuitului

# Fara o stabilizare, amplitudinea oscilatiilor creste necontrolat pana cand amplificatorul ajunge la saturatie iar semnalul devine puternic distorsionat. Din acest motiv am implementat un circuit care realizeaza controlul automat al amplitudinii de oscilatie, am folosit o topologie clasica de limitator realizat cu diode de siliciu care vor intra in conductie la aprox 0.6V si astfel amplitudinea generata de oscilator va fi de circa 0.5V – 0.6V.

# Se folosesc doua diode legate in paralel dar cu polaritate inversata pentru ca circuitul de limitare a amplitudinii sa functioneze pentru ambele alternante ale semnalului sinusoidal.

# Amplitudinea la care va intra in conductie circuitul de limitare este aproximativ:

# Amax= VBE (dar VBE variaza in functie de curentul prin diode)

# Atunci cand tensiunea de la anodul diodei este suficient de mare pentru ca aceasta sa intre in conductie, va incepe sa circule curent catre intrare inversoare a amplificatorului iar amplificarea acestuia va scadea rezultand in micsorarea amplitudinii oscilatiilor dioda apare in paralel cu R19.

# Pentru ca amplificatorul ce sta la baza oscilatorului sa poata reproduca ambele alternante ale semnalului sinusoidal, trebuie ca reteaua de reactie negativa si reteaua Wien sa fie legate la un punct de masa virtuala. Punctul de masa virtuala va avea o valoare egala cu jumatate din tensiunea de alimentare.

# VG=Vcc/2

# Potentialul VG este stabilit prin intermediul unui divizor rezistiv alcatuit din R39 si R40 care au valori egale impreuna cu un amplificator operational cu care vor realiza un punct de masa virtuala (VGND). Se poate alege aproape orice valoare pentru R39 si R40 daca asigura un curent de cativa uA pentru polarizarea intrarii amplificatorului operational.

# Pentru a se indeplini conditia de amorsare a oscilatiilor trebuie ca initial amplificarea in tensiune sa fie mai mare de 3, astfel am ales R19=22k si R31=10k.

# Av=1+R19/R31

# Etajul amplificator in curent (Q si Q10), numit mai adesea etaj final, are rolul de a realiza o adaptare intre etajele de curent mic si impedanta redusa a sarcinii. Dorim ca acest etaj sa amplifice doar in curent iar amplificarea in tensiune sa fie unitara.

# Pentru a indeplini conditia amplificarii in tensiune vom utiliza o conexiune de tip colector comun utilizand o pereche de tranzistoare bipolare complementare.

# Cele doua tranzistoare complementare sunt utilizate intr-o conexiune push-pull in care vor functiona alternatic. Tranzistorul NPN pentru alternante pozitive si cel PNP pentru cele negative. Aceasta configuratie push-pull este specifica amplificatoarelor in clasa B.

# A diagram of a circuit Description automatically generated

# Fig IV.4-Configurație push-pull

# Acest tip de etaj are avantajul randamentului bun si consumului nul de curent in absenta semnalului la intrare, in schimb acesta poate prezenta distorsiuni de trecere prin zero a semnalului daca tranzistoarele nu sunt foarte bine imperecheate.

# Amplificarea in tensiune a etajului final este dat de raportul

# Av=1+R37/R38

# Condensatorul C6 reduce amplificarea la frecvente mari, din afara spectrului util pentru a preveni aparitia de oscilatii parasite.

# Condensatorul C7 elimină componenta continuă prezentă la ieșirea etajului final astfel încât la bornele difuzorului să ajungă doar componenta de curent alternativ. Valoarea lui se alege cât mai mare posibil astfel încât să prezinte o reactanță foarte mică la cea mai mică frecvență pe care dorim să o redăm.

# Pentru a genera un semnal digital pe care îl putem măsura foarte ușor cu microcontrolerul astfel încât să stabilim frecvența acestuia, avem nevoie de un comparator care să transforme semnalul analogic de nivel mic într un semnal dreptunghiular cu amplitudinea de 4-5 V. Astfel am utilizat un comparator cu histerezis care va elimina posibilitatea apariției unor pulsul false prin faptul că există o diferență între pragurile Vhigh si Vlow. Rezistoarele R41 si R42 stabilesc un histerezis de 0.5V care va fi sufficient pentru a crea un semnal dreptunghiular stabil. Amplitudinea semnalului digital este redusa cu ajutorul unui divisor rezistiv format de R43 si R44, acesta limiteaza amplitudinea la circa 4V pentru a fi siguri ca nu se poate deteriora pinul digital.

# A diagram of a display Description automatically generated

# Fig IV.5-Schema electrică a LCD 1602

# Denumirile pinilor pt implementare

# pin 1: VSS;

# pin 2: VDD (+5V);

# pin 3: Vo - contrast LCD (între VSS și VDD );

# pin 4: EN;

# pin 5: VSS;

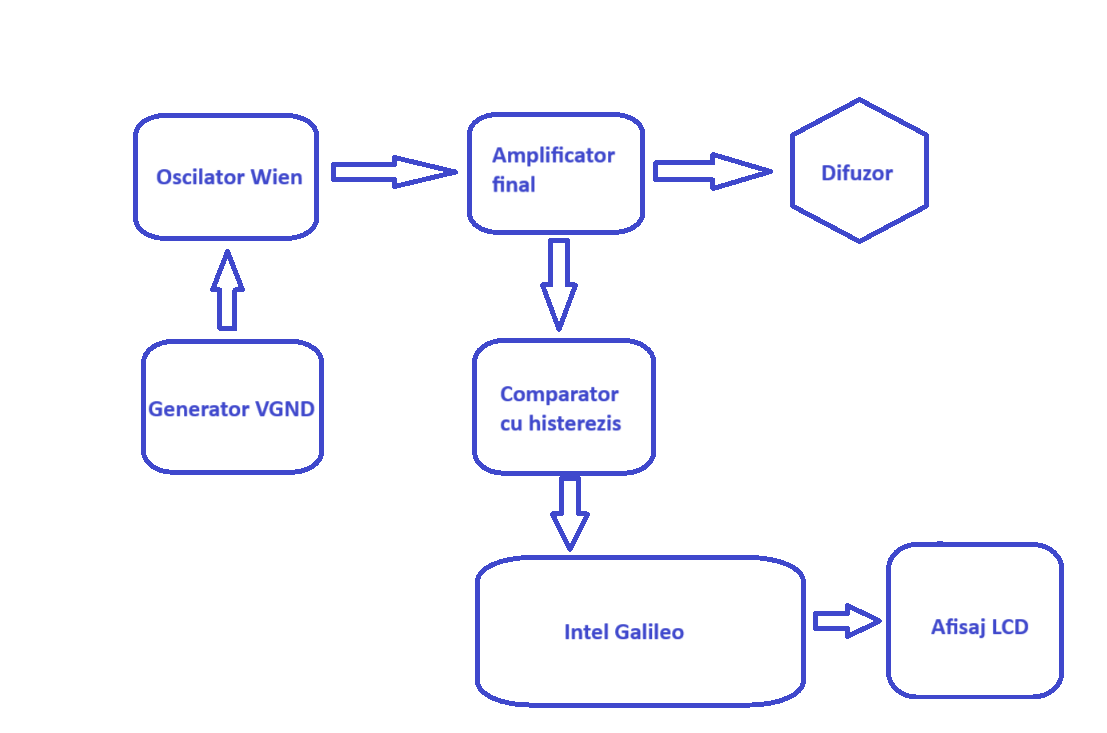
# pin 6: RS;

# pin 11 - 14: D4 - D7;

# pin 15: VCC

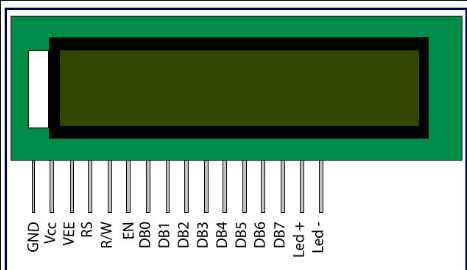
# pin 16: GND

# IMPLEMENTARE SOFTWARE



**Fig V.1**-Schema bloc a circuitului

Afisarea mesajelor catre utilizator se face cu ajutorul unui ecran cu cristale lichide cu o dimeniuni de 2x16 caractere alfanumerice. Astfel ecranul este organizat pe doua randuri cu cate 16 caractere si este controlat prin intermediul a 3 pini de comanda (RS, RW, EN) si 4 pini de date (D4,D5,D6,D7). Microcontrolerul realizeaza afisarea prin trimiterea unor semnale de sincronizare si a unui cod corespondent pentru fiecare caracter de afisat, codul urmand sa fie decodat de circuitul de control din afisaj (HD44780).



**Fig V.2**-LCD cu pinii

Alimentarea partii logice a ecranului si a iluminarii acestuia sunt ambele realizate din tensiunea de 5V comuna sistemului. Reglajul contrastului se realizeaza manual din potentiometrul prin aplicarea unui potential de tensiune continua pe un pinspecial al afisajului (V0).

Aparatul de masura a frecventei si perioadei a fost realizat pe baza unui microcontroler (partea electronica comuna) la care au fost adaugate cateva componente externe: afisaj, circuit analogic de prelucrare, alimentare, etc

Citirea impulsurilor de la generatorul de semnal se face utilizand un pin de digital pe care se va masura durata semnalului masurat, mai exact se masoara Ton si se deduce perioada ca fiind dublul lui Ton pentru ca factorul de umplere este 50%.

A diagram of a period

Description automatically generated

**Fig V.3****-**Reprezentarea perioadelor pe semnalul de ieșire

Ca si functionare generala, microcontrolerul va masura durata unui impuls de tip HIGH, adica logic 1, la fiecare 100ms. Dupa care calculeaza frecvenţă ca inversul perioadei, de aici va rezulta exact frecventa exprimata in Hz impatind numarul de impulsuri la perioada cunoscuta.

Biblioteca **LiquidCrystal** este inclusă pentru a controla un display LCD, iar obiectul lcd este creat folosind pinii specificați pentru conexiunea la display. Urmează definirea pinului de intrare pentru măsurarea frecvenței, contorii pentru perioadele de afișare și măsurare, numărul de impulsuri măsurate, valoarea calculată a frecvenței, perioada semnalului măsurat și o variabilă pentru a memora unitatea de măsură folosită (Hz/kHz).

Funcția **setup** efectuează configurările inițiale care se execută o singură dată. Se inițializează LCD-ul cu 2 rânduri și 16 coloane, se setează cursorul pe primul rând al LCD-ului și se afișează un mesaj inițial, apoi se setează cursorul pe al doilea rând și se afișează un alt mesaj. Comunicarea serială este inițializată, iar pe portul serial se afișează un mesaj de inițializare.

Funcția **loop** reprezintă o buclă infinită. În această funcție se măsoară perioada semnalului (T = 2 \* durata impulsului HIGH) și, dacă au trecut 0.5 secunde de la ultima afișare, se afișează frecvența pe portul serial și unitatea de măsură corespunzătoare (Hz sau kHz). LCD-ul este curățat, se setează cursorul pe primul rând și se afișează titlul "Frecventmetru", iar pe al doilea rând se afișează eticheta "F:" urmată de valoarea frecvenței măsurate și unitatea de măsură corespunzătoare (Hz sau kHz).

Dacă au trecut 0.1 secunde de la ultima măsurare, frecvența este calculată în funcție de perioada măsurată. Dacă perioada este zero, frecvența este setată la zero. Dacă frecvența depășește 1000 Hz, aceasta este convertită în kHz și unitatea de măsură este actualizată.

Valoarea numerica calculata pentru frecventa se converteste intr-o variabila de tip double deoarece frecventa poate ajunge la valori mari si dorim sa evitam depasirea memoriei alocate.

Afisarea se face utilizand functia lcd.print() ce va printa valoarea masurata urmata de unitatea de masura in functie de numarul stocat in variabila „unitate\_masura”

Afisarea se va face utilizand un ecran cu cristale lichide de dimensiune 2x16 caractere, afisajul utilizeaza controlerul HD44780 pentru care avem biblioteca integrata in compilator. Afisajul se alimenteaza din tensiunea de 5V a sistemului, iar pentru comanda afisajului se utilizeaza 7 biti conectati la microcontroler.

**Codul implementat**

#include <LiquidCrystal.h>

LiquidCrystal lcd(3, 4, 5, 6, 7, 8);

const int input\_digital = 2;

unsigned long timp\_afisare;

unsigned long timp\_masurare;

unsigned long impulsuri;

double frecventa;

float perioada;

int unitate\_masura = 1;

void setup()

{

lcd.begin(16, 2);

lcd.setCursor(0, 0);

lcd.print("Frecventmetru");

lcd.setCursor(0, 1);

lcd.print("Proiect 2 ETTI");

Serial.begin(9600);

Serial.println("Frecventmetru");

Serial.println("Proiect 2 ETTI");

}

void loop()

{

perioada = 2 \* pulseIn(input\_digital, HIGH);

if (millis() - timp\_afisare > 500) {

Serial.print("Frecventa:");

Serial.print(frecventa);

if (unitate\_masura == 1)Serial.println("Hz");

else if (unitate\_masura == 2)Serial.println("kHz");

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Frecventmetru");

lcd.setCursor(0, 1);

lcd.print("F:");

lcd.print(frecventa);

if (unitate\_masura == 1)lcd.print("Hz");

else if (unitate\_masura == 2)lcd.print("kHz");

timp\_afisare = millis();

}

if (micros() - timp\_masurare >= 100000) {

if (perioada)frecventa = 1000000 / perioada;

else frecventa = 0;

if (frecventa > 1000) {

frecventa = frecventa / 1000;

unitate\_masura = 2;

}

else unitate\_masura = 1;

timp\_masurare = micros();

} //if timp

} //loop

# CONCLUZII

# În urma realizării acestui proiect, am reușit să dezvoltăm un frecvențmetru funcțional și ușor de utilizat, capabil să măsoare cu acuratețe frecvența semnalelor generate de un oscilator tip punte Wien. Principalele concluzii obținute sunt următoarele:

# -Stabilitatea oscilatorului Wien: Oscilatorul de tip punte Wien s-a dovedit a fi stabil și capabil să genereze un semnal sinusoidal precis, cu frecvența ajustabilă prin intermediul unui potențiometru. Acest lucru a fost esențial pentru a asigura măsurători precise ale frecvenței.

# -Măsurarea exactă a frecvenței: Implementarea sistemului de măsurare a frecvenței, utilizând un microcontroler, a permis calcularea precisă a frecvenței semnalului generat de oscilator. Algoritmul de măsurare a fost optimizat pentru a oferi rezultate rapide și exacte.

# -Afișarea clară a rezultatelor: Interfața de afișare pe ecranul LCD a fost dezvoltată astfel încât să prezinte în mod clar și exact valoarea frecvenței măsurate. Aceasta a îmbunătățit semnificativ utilizabilitatea dispozitivului, oferind utilizatorilor informații ușor de interpretat.

# -Redarea auditivă a semnalului: Conectarea unui difuzor pentru redarea auditivă a semnalului de ieșire al oscilatorului a permis utilizatorilor să audă frecvența semnalului, oferind astfel o reprezentare suplimentară a acestuia.

# Îmbunătățiri și probleme întâmpinate: În timpul implementării, au apărut diverse probleme legate de stabilitatea semnalului și interferențe. Acestea au fost rezolvate prin ajustarea componentelor și optimizarea circuitului. În viitor, s-ar putea îmbunătăți proiectul prin integrarea unor filtre suplimentare pentru a elimina zgomotul și prin utilizarea unor componente de calitate superioară.

# 

# BIBLIOGRAFIE

# D. Dobrescu, Analiza circuitelor electronice de la funcție către dispozitiv, Ed. Printech, 2004

# G. Brezeanu, F. Drăghici, Circuite electronice fundamentale, Ed. Niculescu, București, 2013

# <https://sites.google.com/site/bazeleelectronicii/home/oscilatoare/4-oscilatorul-punte-wien>

# <https://docs.arduino.cc/retired/getting-started-guides/IntelGalileoGen2/>

# <https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/galileo-g2-datasheet.pdf>

1. [Wien Bridge Oscillator (using op-amp) Explained (youtube.com)](https://www.youtube.com/watch?v=gbUXbaxvX94)
2. <https://docs.arduino.cc/tutorials/zero/simple-audio-frequency-meter/>
3. [http://https//lastminuteengineers.com/arduino-1602-character-lcd-tutorial/](http://https/lastminuteengineers.com/arduino-1602-character-lcd-tutorial/)