**Принцип дії схеми**

Неперервний вхідний сигнал х (t) поступає в аналого-цифровий перетворювач (АЦП), керований синхронізатором і імпульсами від генератора, що задає частоту дискретизації. У момент подачі синхронізуючої імпульсу на виході АЦП виникає сигнал, що відображає результат вимірювання моментального значення вхідного коливання у вигляді двійкового числа з фіксованим кількома розрядів. В залежності від особливостей побудови пристрою цього числу відповідає або послідовність коротких імпульсів (передача в послідовному коді), або з сукупністю  
рівнях напруги на сигнальних шинах окремих розрядів (передача в паралельний код). Перетворений таким чином сигнал поступає в основний блок пристрою, так званий цифровий процесор, що складається з арифметичного пристрою та пристрою пам'яті. Арифметичне пристрій виконує ряд цифр операцій, таких як множення, додавання і переміщення в часі на заданий номер інтервалів дискретизації. У пристрої пам'яті може зберігатися деяка кількість попередніх відліків вхідного і вихідного сигналів, які необхідні для виконання операцій обробки. Цифровий процесор перетворює що поступають у нього числа відповідно до заданого алгоритму фільтрації і створює на виході послідовність двійкових чисел, що представляють вихідний сигнал.

**Аналіз електричної принципової схеми стабілізатор електричних**

**сигналів**

Мікросхеми DA1, DA2 та конденсатори С1-С4 виробляють стабільну напругу 5 В, що використовується для живлення цифрових мікросхем DD1 і DD2. Конденсатори С6-С8 додатково фільтрують напругу живлення мікросхем DD1 та DD2.

# Для стабілізування електричних сигналів використовується МК фірми Atmel ATmega 328p TQFP. Конденсатор С5 використовується в якості захисту від неправильного підключення виводів до мікроконтролеру. На вхід МК (pin 23) подається сигнал котрий потрібно стабілізувати. На вході 23 стоїть АЦП, який перетворює аналоговий сигнал в цифровий. Після АЦП сигнал відправляється на внутрішній таймер МК, за рахунок таймеру буде відбуватися аналіз сигналу по ключовим точкам згідно розрядності таймера (12bit). Після аналізу сигнал відправляється до арифметичного пристрою. Де арифметичний пристрій виконує функцію лінійне перетворення Фур'є . Після цього значення стабілізованого сигналу відправляються на 27 та 28 МК для його подальшого перетворення в аналоговий сигнал відправляється з 27 та 28 виходу МК до зовнішнього ЦАП DD2. Сполучаються МК та ЦАП завдяки шині і2с. Резистори R1-R2 підтягують шину даних SDA та шину тактування SCL до Vdd. Конденсатори С10-С11 байпасні конденсатори увімкненні паралельно лінії Vdd. Конденсатор С9 на виході ЦАПу виконує ті ж самі функції, що й конденсатор С5.

Програмування мікроконтролеру виконується завдяки зовнішнього USB-TTL програматора. Його потрібно підключити до виводів Х2:1 та Х2:2

**ТОПОЛОГІЧНЕ ПРОЕКТУВАННЯ ПРИЛАДУ**

**Трасування друкованої плати та постановка задачі на трасування**

Зіставляючи схему електричну принципову і компонуючи елементи, можна чітко визначитися з місцем розташування елементів на платі. Для досягнення високої якості трасування був зроблений конструкторсько-технологічнпе обчислення.

При трасуванні сполучень необхідно виконувати головні вимоги ДСТУ 10317-79, ДСТУ 2.41778.

Спершу на площину друкованої плати паралельно її сторонам наноситься координатна сітка. Лівий нижній кут є базою, його приймаємо за початок координат для друкованої плати. Основний крок координатної сітки 1.25 мм. Фокуси отворів і контактних площадок поміщають у вузлах координатної сітки. Береться округла форма контактних площадок для збільшення надійності при експлуатації виробу.

Найпростішим прийомом трасування є пряме розведення. У цьому випадку траси протягують по найкоротшому маршруту, що зв'язує крапки. Траси проходять побіля з уже прокладеними трасами, огинаючи їх.

Метод має наступні недоліки: низька ефективність у складних схемах; надмірна заплутаність отриманого малюнка друкованого монтажу; наявність великої кількості перехідних отворів; значне збільшення сумарних довжин зв'язків. Таким чином цей метод трасування рекомендується застосовувати для простих схем.

У даному випадку для скорочення довжини провідників їх розміщають у взаємно перпендикулярних площинах.

Трасування плати здійснялося за допомоги програми PCBLayout, яка входить до пакету програм DipTrace.

Спершу обираємо радіоелементи з розділу «Компоненти» та вручну розставляємо їх на уявній платі.

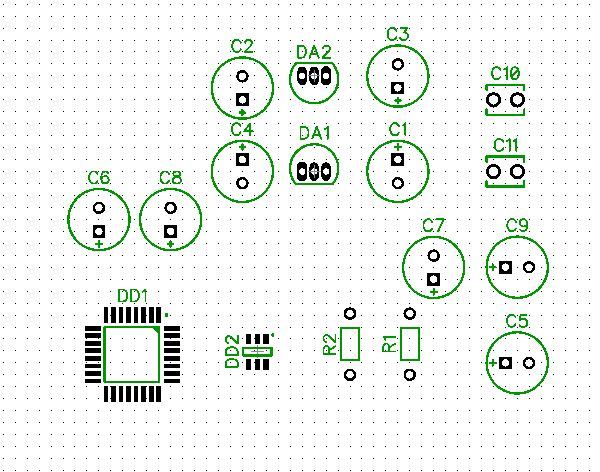


Рисунок 1.Розташування ЕРЕ на платі

У наступних діях можна використати автотрасування, але це приведе до того, що плата буде занадто великою, при тому, що схема електрична принципова є достатньо простою . У панелі інструментів вибирається пункт Ручне Трасування і у вікні справа, вибираються параметри провідника.

Тому що на схемі присутні провідники землі і живлення, ширина вибирається наступним засобом. Натисненням двійчастим кліком ЛКМ, і у вікні, що з'явилося, пишеться та ширина яка розраховувалася в попередньому розділі.

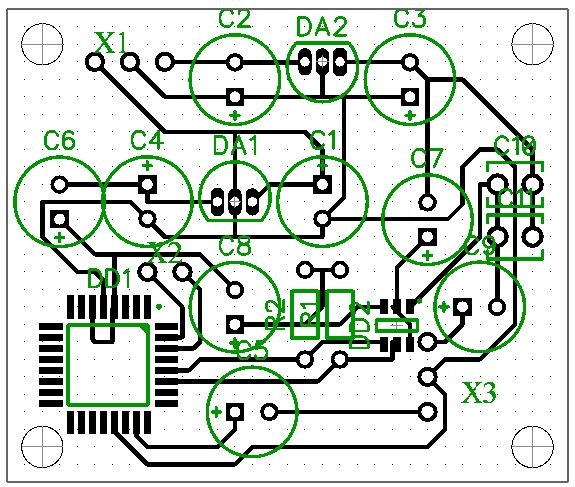


Рисунок 2. Розведення плати

Після розведення плати варто зняти шар із зображенням елементів тому що в даному випадку, ми робимо трасування. Для цього у вікні справа вибираємо Шари-> Верхнія шовкографія .

Перевірка помилок трасування Змінити дизайн запускається після трасування і показує всі потенціальні помилки, якщо вони є (червоні і сині кола). Перебуває перевірка похибок у панелі інструментів.

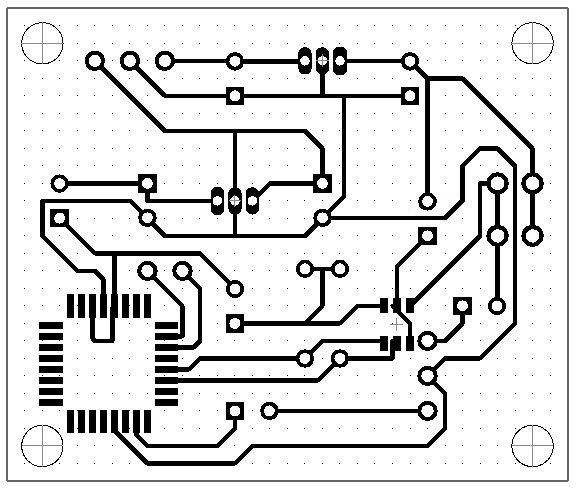


Рисунок 3. Трасування плати без шару елементів

**Розміщення ЕРЕ на платі**

З розстановкою радіоелементів все набагато простіше, виходячи з файлу трасування, можна убрати шар провідників і одержати передбачуваний результат. Для цього потрібно у вікні, що знаходиться з правого боку, властивостей шарів, убрати галочку "Верхній шар".

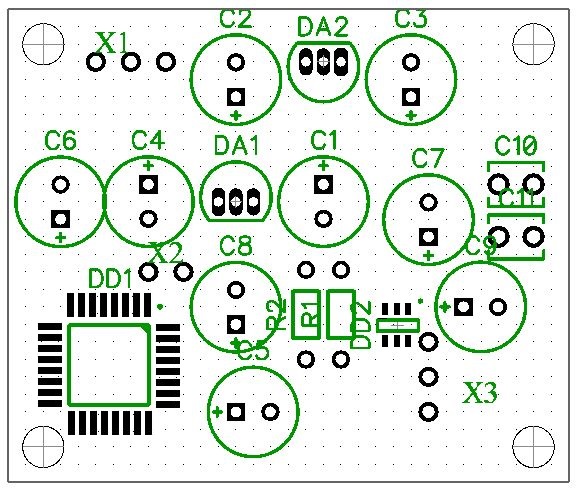
****

Рисунок 4 - Розміщення ЕРЕ на платі

**Друкована плата**

Друко́вана пла́та, — пластина, виконана з [діелектрика](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D0%BA%D0%B8) ([склотекстоліт](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D0%BB%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%BB%D1%96%D1%82), [текстоліт](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%BB%D1%96%D1%82), [гетинакс](https://uk.wikipedia.org/wiki/%D0%93%D0%B5%D1%82%D0%B8%D0%BD%D0%B0%D0%BA%D1%81), [ситал](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%82%D0%B0%D0%BB%D0%B8) тощо), на якій або/і всередині якої [сформований хоча б один шар](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B4%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%85_%D0%BF%D0%BB%D0%B0%D1%82) з провідними доріжками. На друковану плату монтуються [електронні компоненти](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D1%96_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B8), які з'єднуються своїми виводами з елементами провідного рисунка [паянням](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%8F%D0%BD%D0%BD%D1%8F), або, значно рідше, [зварюванням](https://uk.wikipedia.org/wiki/%D0%97%D0%B2%D0%B0%D1%80%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), у результаті чого складається [електронний модуль](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8C) — змонтована друкована плата.

**Розміщення компонентів поверхневого монтажу на друкованій платі**

Поверхневий монтаж ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) surface mount technology, SMT) — технологія виготовлення [електронних пристроїв](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%B8%D1%81%D1%82%D1%80%D1%96%D0%B9), в якій [компоненти](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D1%96_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B8) встановлюються безпосередньо на поверхню [друкованої плати](https://uk.wikipedia.org/wiki/%D0%94%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D0%BF%D0%BB%D0%B0%D1%82%D0%B0). Компоненти для поверхневого монтажу називаються SMD ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) surface mount device). Цей метод виготовлення друкованих вузлів значною мірою замінив технологію [наскрізного монтажу](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BA%D1%80%D1%96%D0%B7%D0%BD%D0%B8%D0%B9_%D0%BC%D0%BE%D0%BD%D1%82%D0%B0%D0%B6), в якому вивідні ко мпоненти монтуються на друкованій платі за допомогою отворів у ній.

Компоненти для поверхневого монтажу зазвичай мають менші розміри, ніж їх аналоги у виводних корпусах. Електронна промисловість має низку стандартних форм і типорозмірів SMD компонентів.

Типова послідовність операцій в технології поверхневого монтажу включає:

- Нанесення [паяльної пасти](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%8F%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BF%D0%B0%D1%81%D1%82%D0%B0) на контактні площадки (дозування в одиничному і дрібносерійному виробництві, трафаретний друк в серійному і масовому виробництві)

- Установка компонентів

- Групове паяння методом оплавлення пасти у печі (переважно методом конвекції, а також інфрачервоним нагріванням або в паровій фазі)

- Очищення плати від [флюсу](https://uk.wikipedia.org/wiki/%D0%A4%D0%BB%D1%8E%D1%81) (в залежності від його активності) і нанесення захисних покриттів.

Одним з найважливіших технологічних матеріалів, що застосовуються при поверхневому монтажі, є [паяльна паста](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%8F%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BF%D0%B0%D1%81%D1%82%D0%B0), що являє собою суміш порошкоподібного припою з органічними наповнювачами, до яких входить флюс. Окрім забезпечення процесу [паяння](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%8F%D0%BD%D0%BD%D1%8F) [припоєм](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BF%D1%96%D0%B9) і підготовки поверхонь, паяльна паста також виконує функцію фіксування компонентів до паяння за рахунок [в'язкості](https://uk.wikipedia.org/wiki/%D0%92%27%D1%8F%D0%B7%D0%BA%D1%96%D1%81%D1%82%D1%8C) і склеювальних властивостей.

При паянні методом поверхневого монтажу дуже важливо забезпечити правильний [температурний графік](https://uk.wikipedia.org/w/index.php?title=%D0%A2%D0%B5%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D1%83%D1%80%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%84%D1%96%D0%BB%D1%8C_%D0%BF%D0%B0%D1%8F%D0%BD%D0%BD%D1%8F&action=edit&redlink=1) в часі ([термопрофіль](https://uk.wikipedia.org/w/index.php?title=%D0%A2%D0%B5%D1%80%D0%BC%D0%BE%D0%BF%D1%80%D0%BE%D1%84%D1%96%D0%BB%D1%8C&action=edit&redlink=1" \o "Термопрофіль (ще не написана))), щоб уникнути [термоударів](https://uk.wikipedia.org/w/index.php?title=%D0%A2%D0%B5%D1%80%D0%BC%D0%BE%D1%83%D0%B4%D0%B0%D1%80&action=edit&redlink=1), забезпечити добру активацію [флюсу](https://uk.wikipedia.org/wiki/%D0%A4%D0%BB%D1%8E%D1%81) і змочування поверхні [припоєм](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BF%D1%96%D0%B9).

**МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ ПАРАМЕТРІВ**

**Моделювання в Proteus**

Моделювання проводилося в програмі Proteus 8 Pro

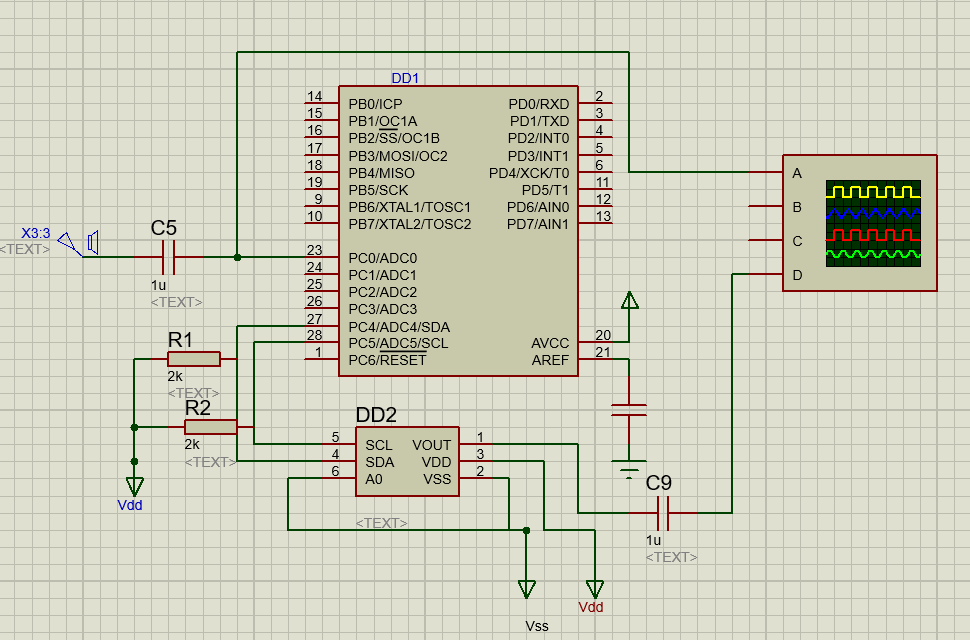


Рисунок 5. Моделювання електричних параметрів

**Перетворення Фур'є** — [інтегральне перетворення](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F) однієї комплекснозначної функції дійсної змінної на іншу. Тісно пов'язане з [перетворенням Лапласа](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_%D0%9B%D0%B0%D0%BF%D0%BB%D0%B0%D1%81%D0%B0) та аналогічне розкладу у [ряд Фур'є](https://uk.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A4%D1%83%D1%80%27%D1%94) для неперіодичних функцій. Це перетворення розкладає дану функцію на [осциляторні функції](https://uk.wikipedia.org/wiki/%D0%93%D0%B0%D1%80%D0%BC%D0%BE%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%BA%D0%BE%D0%BB%D0%B8%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Використовується для того, щоб розрахувати спектр частот для сигналів змінних у часі (таких як [мова](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0) або [електрична напруга](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0_%D0%BD%D0%B0%D0%BF%D1%80%D1%83%D0%B3%D0%B0)).

Перетворення Фур'є [функції](https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) f ( t ) {\displaystyle f(t)\,} математично визначається як [комплексна](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81%D0%BD%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) функція F ( ω ) {\displaystyle F(\omega )\,} , яка задається [інтегралом](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB)

H:\Диплом\Fur'e.JPG

Якщо використовувати перетворення Фур'є ми отримаємо

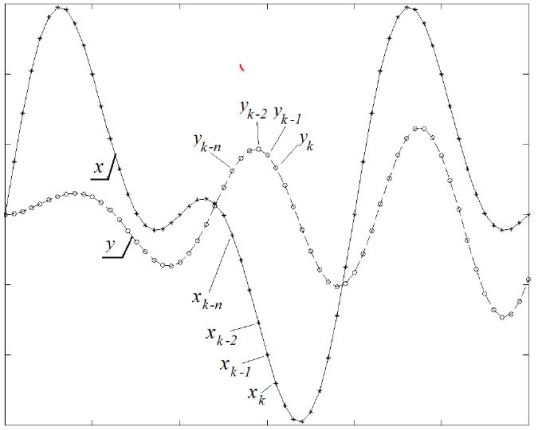


Рисунок 7. Графік перетворення Фур'є

,де Х – вхідний сигнал, а У- вихідний сигнал.

**Лінійність**

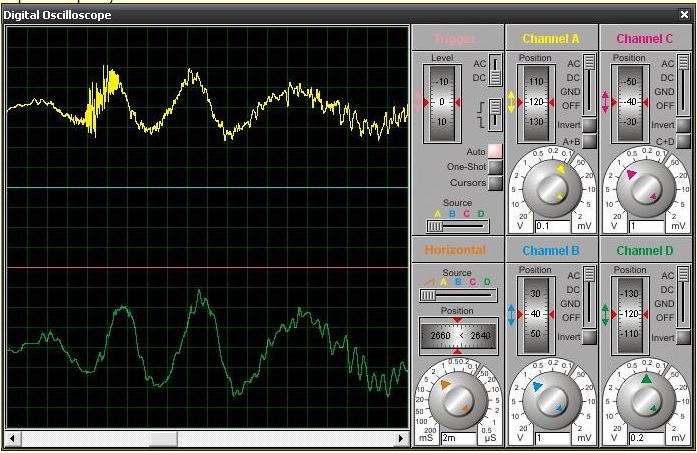
Для довільних комплексних чисел a {\displaystyle a} α та b {\displaystyle b}b, якщо

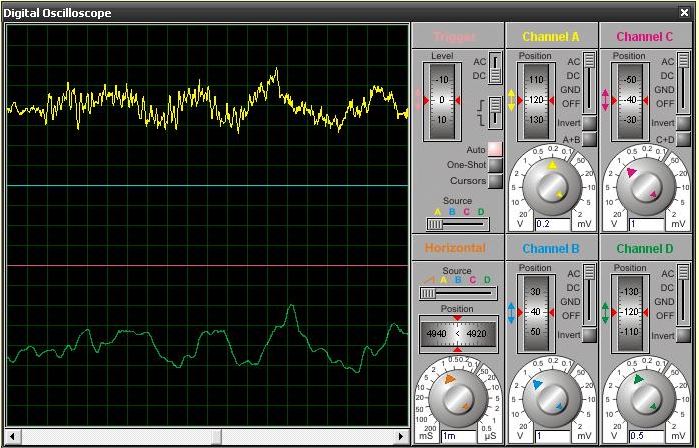
H:\Диплом\Lin.JPG

Метою цього розділу є моделювання електричних параметрів стабілізатора електричних сигналів. Даний стабілізатора електричних сигналів, схему якого показано на рисунку 15, являє собою цифровий (фільтр що програмується )фільтр.

На вхід осцилографа подається сигнал котрий непройшов стабілізацію (вхід А), та сигнал котрий пройшов стабілізацію (вхід D). На осцилографі жовтий колір це вхід А, а зелений - вхід D

Алгоритм цього лінійного перетворення Фур'є виходить вписується в 500 тактів мікроконтролера. Так, код програми може виконнвати 280 тисяч операцій для 128-точкового (обчислювальна кладність Nfft²), тобто 35 мс на 8 МГц. Частота мікроконтролеру 16 МГц та дільник на АЦП = 32. Це приблизно 38 кГц дискретизація... А преривання робиться через кожні 416 тактів мікроконтроллера.

Роботу цього стабылызатора електричних сигналыв можна побачити на наступних рисунках. 



**Код та його робота**

.include "m8def.inc"

.def zeroreg = r2;

.def count\_k = r3;

.def count\_n = r4;

.def accum\_a\_l = r5;

.def accum\_a\_h = r6;

.def accum\_b\_l = r7;

.def accum\_b\_h = r8;

.def tmpX = r9;

.def signal = r16; r16...r23 only!

.def coeff = r17; r16...r23 only!

.def tmp = r18;

ldi tmp, high(RAMEND);

out SPH, tmp;

ldi tmp, low(RAMEND);

out SPL, tmp; установка стека

ldi r27, $01;

clr r26; X - sin/cos(nk), $0100 = 256

clr r29;

ldi r28, $60; Y - s(n), $0060 = 96

ldi tmp, 16

clr count\_k;

ldi r31, high(input\_signal\*2);

ldi r30, low(input\_signal\*2);

loop\_test:

lpm tmpX, Z+;

st Y+, tmpX;

inc count\_k;

cpse count\_k, tmp;

rjmp loop\_test;

clr count\_k;

ldi r31, high(coeff\_sinus\*2);

ldi r30, low(coeff\_sinus\*2);

loop\_prepare:

lpm tmpX, Z+;

st X+, tmpX;

inc count\_k;

cpse count\_k, tmp;

rjmp loop\_prepare;

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sin\_cos\_transform: ;початок (основної) підпрограми

clr count\_k; ініциалізація лічильника гармоник

clr r31;

ldi r30, $B0; Z - a(k) & b(k). $00B0 = 176

loop\_k:

;{ loop\_k

inc count\_k; інкремент лічильника k для отримання 1-ї гармоніки

ldi r27, $01;

clr r26; X - sin/cos(nk), $0100 = 256

clr r29;

ldi r28, $60; Y - s(n), $0060 = 96

clr accum\_a\_l;

clr accum\_a\_h;

clr accum\_b\_l;

clr accum\_b\_h;

clr count\_n;

loop\_n:

;{{ loop\_n

ld signal, Y+; Завантажити посилання сигналу з пост-інкрементом

ld coeff, X;? Завантажити відлік синуса

mov tmpX, r26; Зберегти наступну адресу

mulsu coeff, signal; s(n)\*sin(nk)

adc r1, zeroreg;

clr r0;

sbrc r1,7

com r0;

add accum\_a\_l, r1;

adc accum\_a\_h, r0;

ldi tmp, 4; %%%%%%%%%%%%%%%%%%%%

add r26, tmp; Ми додаємо Nfft/4 щоб отримати косинус

andi r26, 0b00001111; маска щоб не вілазити за кордон таблиці

ld coeff, X;? заваантаження відліку косинуса

mov r26, tmpX; відновлюєм наступ. адрес синуса

add r26, count\_k;? інкремент відповідний номеру гармоніки

andi r26, 0b00001111; маска щоб не вілазити за кордон таблиці

mulsu coeff, signal; s(n)\*cos(nk)

adc r1, zeroreg;

clr r0;

sbrc r1,7

com r0;

add accum\_b\_l, r1;

adc accum\_b\_h, r0;

inc count\_n;

ldi tmp, 15;15 Nfft-1 %%%%%%%%%%%%%%%%%%%%%%%%%%%

cpse count\_n, tmp;

rjmp loop\_n;

;}} loop\_n

rol accum\_a\_l;

rol accum\_a\_h;

rol accum\_a\_l;

rol accum\_a\_h;

rol accum\_a\_l;

rol accum\_a\_h;

rol accum\_a\_l;

rol accum\_a\_h;

rol accum\_a\_l;

rol accum\_a\_h;

st Z+, accum\_a\_h;

rol accum\_b\_l;

rol accum\_b\_h;

rol accum\_b\_l;

rol accum\_b\_h;

rol accum\_b\_l;

rol accum\_b\_h;

rol accum\_b\_l;

rol accum\_b\_h;

rol accum\_b\_l;

rol accum\_b\_h;

st Z+, accum\_b\_h;

ldi tmp, 8;8 Nfft/2 %%%%%%%%%%%%%%%%%%%%%%%%%%%

cpse count\_k, tmp;

rjmp loop\_k;

;} loop\_k

;END of sin-cos-transform

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

nop;

nop;

.cseg

input\_signal:

;.db 128, 176, 218, 245, 255, 245, 218, 176, 128, 79, 37, 10, 0, 10, 37, 79;sinus\_1 0...255

.db 255, 0, 255, 0, 255, 0, 255, 0, 255, 0, 255, 0, 255, 0, 255, 0;conus\_8 0...255

coeff\_sinus:

.db 0, 49, 90, 117, 127, 117, 90, 49, 0, -49, -90, -117, -127, -117, -90, -49;127\*sinus

Спершу завантажується до ОЗУ таблиця синусів та їх коефіцієнти. Потім вибираються регистри вводу, виводу сигналу та регістри акумулятора.

Z - a(k) & b(k) – лінійне перетворення Фур'є. Проводяться визначення кута сигналу. Після цього визначається середня точка між сусідніми точками(їх кількість 128). Після обробки сигналу, він відправляється до зовнішнього ЦАП.