

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «СИСТЕМА ЦИФРОВОГО ГЕНЕРУВАННЯ АНАЛОГОВИХ
СИГНАЛІВ»

Виконав: студент 2 курсу, групи 1КІ-21м
спеціальності 123 Комп'ютерна інженерія

Дідур І.В.
Керівник д.т.н., проф., проф. каф. ОТ

Азаров О.Д.
Опонент к.т.н., доц., зав.каф. МБІС

Карпінець В.В.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.
« 11 » 12 2023 р.

Вінниця 2023

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
обчислювальної техніки
проф., д.т.н. О. Д. Азаров
« 26 » вересня 2023 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Дідуру Ігору Вячеславовичу

1 Тема роботи «Система цифрового генерування аналогових сигналів», керівник роботи Азаров Олексій Дмитрович, д.т.н., професор, затверджені наказом вищого навчального закладу від 18.09.223 року № 247.

2 Строк подання студентом роботи 9.12.2023 р.

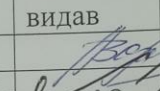
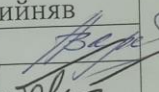
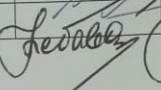
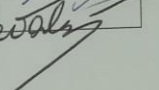
3 Вихідні дані до роботи: мікроконтролери сімейства STM32 із ядром Cortex-M4F фірми STMicroelectronics та цифровий синтезатор AD9850 фірми Analog Devices, Inc, відлагоджувальні комплекси – STM32F4 Discovery та STM32F429I Disco.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, огляд та аналіз систем цифрового генерування аналогових сигналів, проектування апаратного забезпечення системи, проектування програмного забезпечення системи, дослідження спроектованої системи, висновки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
 Код програми для генератора на STM32F4 Discovery, Код програми для генератора на AD9850 та STM32F4 Discovery, Код програми для генератора AD9850, Код програми для генератора AD9850, Структура схема системи цифрового генерування аналогових сигналів

6 Консультанти розділів роботи представлено в табл. 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	Азаров О.Д., д.т.н., проф. каф. ОТ		
5	Небава М.І., к.е.н., проф. каф. ЕП і ВМ		


7 Дата видачі завдання 19.09.2023 р.

8 Календарний план наведено в табл. 2.

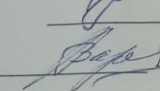
Таблиця 2 — Календарний план

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Прим.
1	Постановка мети та задач роботи	21.10.23	век.
2	Огляд та аналіз систем цифрового генерування аналогових сигналів	25.10-30.10.23	век.
3	Проектування апаратного забезпечення системи	31.10-08.11.23	век.
4	Складання структурної схеми системи	09.11-15.11.23	век.
5	Вибір інструментів розробки	16.11-20.11.23	век.
6	Проектування програмного забезпечення системи	21.11-25.11.23	век.
7	Використання інструментів розробки	26.11-31.11.23	век.
8	Дослідження спроектованої системи	01.12-04.12.23	век.
9	Розрахунок економічної частини роботи	01.12-04.12.23	век.
10	Оформлення пояснювальної записки та ілюстративного матеріалу	05.12.23	век.
11	Аналіз виконання роботи, висновки, додатки		век.
12	Перевірка якості виконання магістерської роботи та усунення недоліків		век.

Студент

 Дідур І.В.

Керівник роботи

 Азаров О.Д.

АНОТАЦІЯ

УДК 004.9

Дідур І.В.

Система цифрового генерування аналогових сигналів. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2023, 110 с.

На укр.мові. Бібліогр.: 19 назв, рис. 61, табл. 10.

Дана магістерська кваліфікаційна робота присвячена створенню системи цифрового генерування аналогових сигналів.

Генератор є важливою ланкою в багатьох електронних пристроях, чи це тестове, лабораторне обладнання, медичні або радіоелектронні системи. Крім технічних характеристик, важливими для користувача залишаються такі параметри, як зручність застосування, компактність і низька вартість генераторів.

Ключові слова: генератор аналогових сигналів, комп'ютер, структурна схема, мікроконтролер, цифровий синтезатор, програмне забезпечення, апаратне забезпечення.

ANNOTATION

Didyr I.V.

System of digital generation of analog signals. Master's qualification route in the specialty 123 — computer engineering, educational program computer engineering. Vinnitsa, VTNU, 2023, 110 p.

In the Ukr. leng. Libr. name 19, figure 61, table 10.

This master's thesis is devoted to the creation of a system of digital generation of analog signals.

A generator is an important link in many electronic devices, whether it is test, laboratory equipment, medical or radio electronic systems. In addition to technical characteristics, such parameters as ease of use, compactness and low cost of generators remain important for the user.

Key words: analog signal generator, computer, structural diagram, microcontroller, digital synthesizer, software, hardware.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД ТА АНАЛІЗ СИСТЕМ ЦИФРОВОГО ГЕНЕРУВАННЯ АНАЛОГОВИХ СИГНАЛІВ	10
1.1 Системи цифрового генерування аналогових сигналів.....	10
1.1.1 Прямий аналоговий синтез (DAS).....	11
1.1.2 Непрямий синтез на основі фазового автопідстроювання (PLL).....	12
1.1.3 Прямий цифровий синтез (DDS).....	13
1.2 Типи генераторів сигналів довільної форми.....	15
1.2.1 Генератори сигналів довільної форми та стандартних функцій (AFG).....	15
1.2.2 Генератори сигналів довільної форми (AWG).....	18
1.2 Огляд існуючих генераторів сигналу довільної форми.....	20
2 ПРОЕКТУВАННЯ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	27
2.1 Структурна схема	27
2.2 Вибір мікроконтролера.....	30
2.2.1 Опис плати STM32F4 Discovery.....	31
2.2.2 Опис плати STM32F429I Disco.....	32
2.3 Цифровий синтезатор частоти.....	34
2.4 Перетворювач напруги.....	36
2.5 Фільтр низьких частот.....	38
2.6 Підсилювач.....	45
3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	47
3.1 Логіка роботи системи.....	47
3.2 Використання інструментів розробки.....	51
3.2.1 Середовище розробки Coocox CoIDE.....	52
3.2.2 STM32 ST-LINK Utility.....	57

					<i>08-54.МКР.006.00.000 ПЗ</i>		
		<i>№ докум.</i>	<i>Підпис</i>				
<i>Розробив</i>		<i>Дідур І.В.</i>		Система цифрового генерування аналогових сигналів <i>Пояснювальна записка</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Азаров О.Д.</i>			6		
<i>Рецензент</i>		<i>Карпінець В.В.</i>			ВНТУ, гр. 1КІ-22м		
<i>Н. Контроль</i>		<i>Швець С. І.</i>					
<i>Затверджую</i>		<i>Азаров О. Д.</i>					

4 ДОСЛІДЖЕННЯ СПРОЕКТОВАНОЇ СИСТЕМИ.....	61
4.1 Генерація сигналів із використанням STM32F4 Discovery.....	61
4.2 Тестування цифрового синтезатора AD9850.....	62
4.3 Підключення цифрового синтезатора AD9850 до STM32F4 Discovery.....	66
4.4 Функціональний DDS генератор на STM32F429I Disco.....	71
4.5 Функціональний DDS генератор на STM32F429I Disco та AD9850.....	74
4.6 Функціональний DDS генератор на STM32F429I Disco та AD9850.....	74
5 ЕКОНОМІЧНА ЧАСТИНА	77
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	77
5.2 Прогнозування витрат на виконання науково-дослідної роботи	80
5.3 Розрахунок економічної ефективності науково-технічної розробки	86
ВИСНОВКИ.....	92
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	93
ДОДАТОК А Технічне завдання.....	88
ДОДАТОК Б Код програми для генератора на STM32F4 Discovery.....	99
ДОДАТОК В Код програми для генератора на AD9850 та STM32F4 Discovery,	108
ДОДАТОК Г Код програми для генератора AD9850 до STM32F4 Discovery....	112
ДОДАТОК Д. Код програми для генератора AD9850 до STM32F429I Disco.....	114
ДОДАТОК Е Структура схема системи цифрового генерування аналогових сигналів	117
ДОДАТОК Ж Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	118

						08-54.МКР.006.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ВСТУП

Генератор є важливою ланкою в багатьох електронних пристроях, чи це тестове, лабораторне обладнання, медичні [1, 2] або радіоелектронні системи. Крім технічних характеристик, важливими для користувача залишаються такі параметри, як зручність застосування, компактність і низька вартість генераторів. На жаль, пристрої, які пропонують більшість виробників, є дорогими. У зв'язку з цим залишається актуальною проблема створення бюджетного макета генератора, здатного замінити більш дорогі чи застарілі моделі [2].

Виробляється безліч інтегральних мікросхем, що дозволяють генерувати сигнали різної форми. Ці мікросхеми відрізняються за способом синтезу сигналів [3] та технічними характеристиками. За методом синтезу сигналів вони поділяються на прямі аналогові синтезатори, непрямі синтезатори на основі фазового автопідстроювання частоти, прямі цифрові синтезатори та гібридні синтезатори. В даний час найбільшою популярністю користуються синтезатори на основі прямого цифрового синтезу (Direct Digital Synthesis, або DDS) завдяки перевагам, пов'язаним з широкою роздільною здатністю за частотою, високою якістю сигналу, можливістю цифрового керування, низькою вартістю, малим розміром корпусів та енергоспоживанням [2].

Для налаштування та керування цифровим синтезатором необхідний мікроконтролер (МК). В даний час найбільшою популярністю серед розробників електронних пристроїв різного рівня стали використовувати мікроконтролери STM32.

Сімейство 32-розрядних флеш-мікроконтролерів STM32 на базі процесора Arm Cortex-M розроблено, щоб запропонувати користувачам нові ступені свободи. Він пропонує продукти, що поєднують дуже високу продуктивність, можливість обробки цифрових сигналів у реальному часі, роботу з низьким енергоспоживанням, а також можливість підключення, зберігаючи при цьому повну інтеграцію і простоту розробки. [3,8]. Більшість моделей платформи програмується через USB-порт, який дозволяє підключати плату безпосередньо до

комп'ютера. Пристрої на базі STM32 можуть працювати самостійно, або спільно з програмним забезпеченням комп'ютера. Як засоби управління макетом може бути використане як периферійне обладнання (дисплеї, кнопки, енкодери), так і персональний комп'ютер. А тому розробка бюджетного генераторів на базі сучасних мікроконтролерів є **актуальною**, а запропоновані рішення здатні замінити більш дорогі чи застарілі моделі

Метою магістерської кваліфікаційної роботи є вдосконалення системи цифрового генерування аналогових сигналів.

Задачі дослідження:

— виконати огляд та аналіз існуючих систем цифрового генерування аналогових сигналів;

— розробити структурну, функціональну та електричну принципову схеми.

— обрати компонентну базу системи, що розробляється;

— розробити програмне забезпечення для управління мікропроцесором.

— запропонувати варіанти вдосконалення та модифікації генераторів аналогових сигналів;

— провести дослідження та моделювання варіантів цифрового генерування аналогових сигналів, розглянути їх основні переваги та недоліки;

— провести економічний розрахунок системи, що розробляється.

Об'єкт дослідження — процеси, що протікають в системах цифрового генерування аналогових сигналів.

Предмет дослідження — методи та засоби побудови систем цифрового генерування аналогових сигналів.

Наукова новизна — удосконалено систему цифрового генерування аналогових сигналів та реалізовано декілька модифікацій апаратних і програмних засобів.

Практичне значення одержаних результатів:

— розроблено структурну схему та підібрано компоненти для системи, що розробляється;

— розроблено програму для цифрового генерування аналогового сигналу та

гнучке управління системою.

— запропоновано можливі варіанти модифікації програми керування, а також розглянуто апаратні реалізації, їх основні переваги та недоліки.

Апробацію результатів наукової роботи було проведено на науковій конференції «LIII Науково–технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2024)», доповідь на тему: Система цифрового генерування аналогових сигналів” .

1 ОГЛЯД ТА АНАЛІЗ СИСТЕМ ЦИФРОВОГО ГЕНЕРУВАННЯ АНАЛОГОВИХ СИГНАЛІВ

1.1 Класифікація методів синтезу частот

Ще кілька років тому прямі цифрові синтезатори частоти (DDS) були рідкістю і мали обмежену сферу застосування, їх широке використання стримувалося складністю реалізації, а також недостатньо широким діапазоном робочих частот. DDS вже не сприймаються розробниками як складні, незрозумілі та дорогі пристрої.

Синтезатор частот — це електронний пристрій, здатний формувати з опорної частоти на виході необхідну частоту або набір частот, згідно з керуючими сигналами.

Основні параметри, що характеризують якість синтезатора частоти:

- чистота спектра вихідного сигналу (рівень шуму та рівень побічних компонентів);
- діапазон перебудови (смуга частот вихідного сигналу);
- швидкість перебудови;
- частотний дозвіл;
- кількість частот, що генеруються;
- гнучкість (можливість здійснення різних видів модуляції);
- нерозривність фази вихідного сигналу під час перебудови.

1.1.1 Прямий аналоговий синтез (DAS)

Розглянемо структурну схему прямого аналогового синтезатора (DAS), яка зображена на рисунку 1.1. Через те, що у ньому відсутній процес корекції помилки, даний метод називають прямим. Якість вихідного сигналу залежить від якості опорного сигналу. Фазовий шум такого синтезатора є досить низьким внаслідок прямого синтезу. Перебудова частотою може бути дуже швидкою.

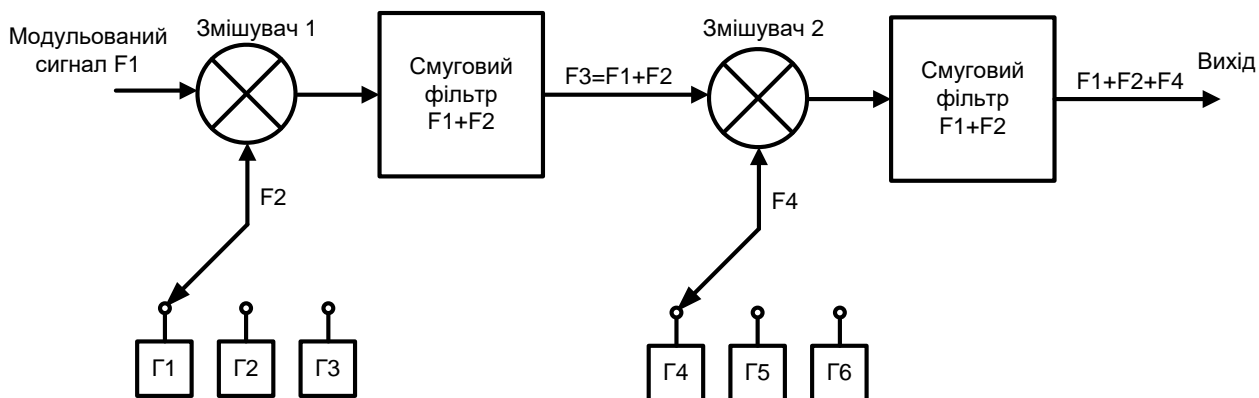


Рисунок 1.1 — Прямий аналоговий синтез

Однією з важливих особливостей DAS-синтезатора на основі змішувача/фільтра є можливість повернутися на будь-яку частоту і продовжувати працювати в тій же фазі, якби переходу не було взагалі. Цей ефект називають «фазовою пам'яттю».

Банк опорних генераторів, що перемикається, використовують для перебудови по частоті. Використовуючи ділянки частоти, що мають структуру змішувач/фільтр/ділянка, можна зменшити кількість необхідних опорних генераторів, хоча і в цьому випадку можливості перебудови залишаться більш ніж скромними.

1.1.2 Непрямий синтез на основі фазового автопідстроювання (PLL)

Розглянемо непрямий синтез частоти на основі фазового автопідстроювання (PLL) у відповідності до рисунку 1.2.

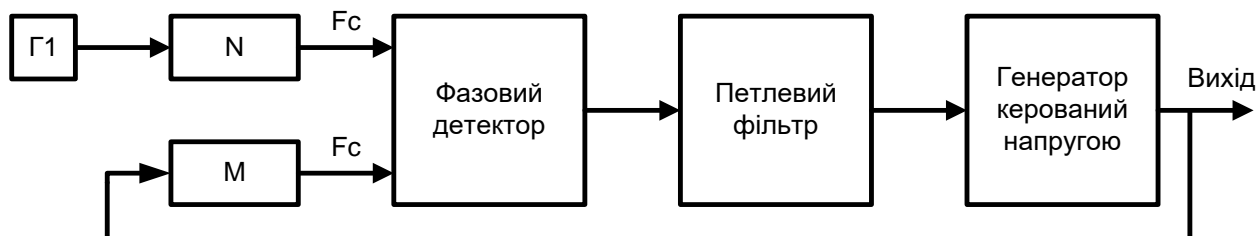


Рисунок 1.2 — Непрямий синтез на основі фазового автопідстроювання

Цей метод синтезу використовує принцип порівняння фази вихідного сигналу та частоти, яка формується генератором керованим напругою (VCO), з сигналом опорного генератора. Виявлення помилки забезпечує фазовий детектор (ФД), який працює на певній частоті FC , яка називається частотою порівняння. Оскільки дільники частоти мають цілі коефіцієнти поділу, крок сітки такого синтезатора визначає частота порівняння.

Вихідна частота визначається за формулою:

$$FOUT = FC \times M = \left(\frac{FCLK}{N}\right) \times M = FCLK \times \left(\frac{N}{M}\right),$$

де $FOUT$ — Вихідна частота;

FC — частота порівняння;

N — коефіцієнт поділу опорної частоти;

M — коефіцієнт поділу вихідної частоти.

PLL-синтезатор множить опорну частоту N/M разів. Коефіцієнти N і M можуть задаватися мікроконтролером, хоча на практиці число N при перебудові змінюють рідко, так як це тягне зміну частоти порівняння і вимагає зміни параметрів петльового фільтра.

Фазовий датчик є джерелом додаткових фазових шумів. Спроба отримати мінімальний крок перебудови частоти змушує працювати на нижчій частоті порівняння, що вимагає зниження частоти зрізу петлевого фільтра. І це ще більше збільшує фазові шуми. Швидку перебудову частоти у такому синтезаторі забезпечити також дуже складно.

Для отримання малого кроку перебудови частоти іноді об'єднують в одному синтезаторі кілька петель PLL. Однак багатопетльовий PLL-синтезатор є дуже дорогим та громіздким пристроєм, що стримує його широке застосування.

1.1.3 Прямий цифровий синтез (DDS)

Прямий цифровий синтез (DDS) — це відносно новий метод синтезу частоти,

який з'явився на початку 70-х років XX століття. Всі описані методи синтезу доступні розробникам вже багато часу, але лише останнім часом DDS приділяється пильна увага. Поява дешевих мікросхем з DDS та зручних засобів розробки робить їх сьогодні привабливими для різних сфер застосування.

DDS унікальні своєю цифровою визначеністю, що генерується ними сигнал синтезується з властивою цифровим системам точністю. Частота, амплітуда і фаза сигналу в будь-який момент часу точно відомі і підконтрольні. DDS практично не схильні до температурного дрейфу і старіння. Високі технічні характеристики спричинили те, що останнім часом DDS витісняють звичайні аналогові синтезатори частот.

Основні переваги DDS:

— дуже висока роздільна здатність за частотою і фазою, управління якими здійснюється в цифровому вигляді;

— екстремально швидкий перехід на іншу частоту (або фазу), перебудова частотою без розриву фази, без викидів та інших аномалій, пов'язаних з часом встановлення;

— архітектура, заснована на DDS, зважаючи на дуже малий крок перебудови по частоті, виключає необхідність застосування точного підстроювання опорної частоти, а також забезпечує можливість параметричної температурної компенсації;

— цифровий інтерфейс дозволяє легко реалізувати мікроконтролерне керування.

Частотна роздільна здатність DDS становить соті і навіть тисячні частки герца при вихідній частоті близько десятків мегагерц. Таке дозвіл недосяжно інших методів синтезу. Іншою характерною рисою DDS є дуже висока швидкість переходу в іншу частоту. Для DDS швидкість перебудови обмежена практично лише швидкодією цифрового керуючого інтерфейсу. Більше того, всі перебудови по частоті DDS відбуваються без розриву фази вихідного сигналу. Оскільки вихідний сигнал синтезується у цифровому вигляді, дуже просто можна здійснити модуляцію різних видів.

Параметри синтезатора частоти є дуже важливими для апаратури зв'язку.

Будучи серцем системи налаштування, синтезатор переважно визначає споживчі властивості конкретного апарату. Як з технічної, так і економічної сторони DDS задовольняє більшості критеріїв ідеального синтезатора частоти: простий, високоінтегрований, з малими габаритами. Сучасні DDS використовують субмікронну CMOS-технологію, тривольтову логіку, мініатюрні корпуси. Одночасно постійно знижуються ціни на них. Усе це робить DDS дуже перспективними пристроями [9, 10].

1.2 Типи генераторів сигналів довільної форми

1.2.1 Генератори сигналів довільної форми та стандартних функцій (AFG)

Генератори сигналів довільної форми можна розділити на генератори сигналів довільної форми та стандартних функцій (AFG) та генератори сигналів довільної форми (AWG).

Генератор AFG призначений для вирішення широкого спектру завдань і на сьогоднішній день генератори цього типу є найпоширенішими. Зазвичай такий прилад пропонує менше можливостей зміни сигналу, ніж аналогічний AWG, але має чудову стабільність і швидкий відгук на зміну частоти. Якщо досліджуваному пристрою необхідна синусоїда та меандр (не кажучи вже про інші сигнали) і можливість майже миттєвого перемикання між двома частотами, вирішити це завдання можна за допомогою AFG. Іншою перевагою AFG є низька ціна, що робить його дуже привабливим для додатків, що не потребують гнучкості AWG.

AFG і AWG мають спільні риси, хоча по конструкції AFG є більш спеціалізованим приладом. AFG має унікальні переваги: він створює стабільні сигнали стандартних функцій, зокрема, синусоїди і меандри, що широко застосовуються, що володіють високою точністю і швидко перебудовуються по частоті. Швидка перебудова означає можливість швидкого та чистого переходу з однієї частоти до іншої.

Більшість AFG пропонує набір наступних широко використовуваних сигналів та функцій:

- синусоїда;
- меандр;
- трикутник;
- свіпування;
- імпульс;
- лінійне наростання;
- модуляція;
- гаверсінус.

І хоча АWG теж можуть генерувати всі ці сигнали, сучасні АFG забезпечують покращене керування фазовими, частотними та амплітудними характеристиками вихідного сигналу.

Крім того, багато АFG дозволяють модулювати сигнал внутрішнім або зовнішнім джерелом, що дуже важливо для деяких типів тестування на відповідність стандартам. Старі моделі АFG для створення вихідних сигналів використовували аналогові генератори, що задають, з подальшою обробкою сигналу.

Останні моделі АFG використовують технологію прямого цифрового синтезу (DDS) тактової частоти, з якої вибірки сигналу виймаються з пам'яті.

Особливістю технології DDS є синтез вихідних сигналів та всіх частот, що використовуються всередині приладу, лише з однієї тактової частоти. На рисунку 1.3 показано спрощену архітектуру АFG, побудованого на основі DDS.

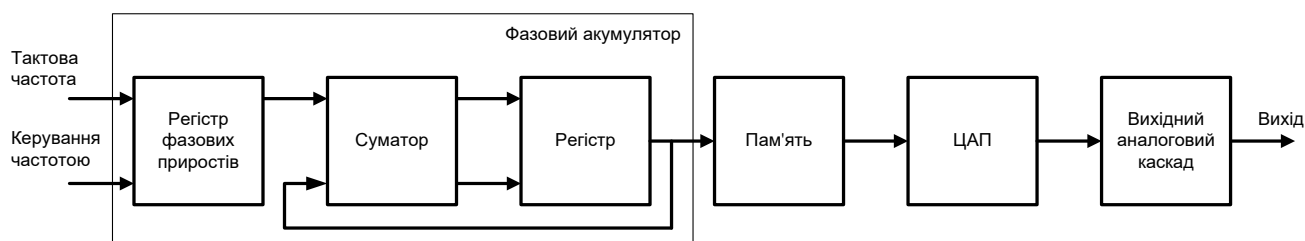


Рисунок 1.3 — Схема спрощеної архітектури АFG

Регістр фазових збільшень у фазовому акумуляторі приймає інструкції від

контролера частоти, що показують, на скільки повинна змінюватися фаза вихідного сигналу в кожному наступному періоді. У сучасних продуктивних AFG роздільна здатність по фазі може досягати 2^{-30} , що приблизно дорівнює $1/1000000000$.

Вихід фазового акумулятора використовується як тактова частота для пам'яті сигналів. Робота цього приладу дуже нагадує роботу AWG, за винятком, що пам'ять сигналів містить зазвичай лише кілька базових сигналів, таких як синус і меандр.

Аналоговий вихідний ланцюг зазвичай є фіксованим фільтром нижніх частот, який забезпечує надходження на вихід тільки запрограмованої корисної частоти (пригнічуючи проникнення тактової частоти).

Щоб краще зрозуміти, як фазовий акумулятор створює потрібну частоту, уявіть собі, що контролер посилає значення "1" у 30-бітовий регістр фазових прирощень. Вихідний регістр фазових прирощень фазового акумулятора змінюватиме фазу на $360/2^{30}$ градусів у кожному періоді, оскільки повний період вихідного сигналу відповідає фазі 360 градусів. Отже, значення "1" регістра фазових прирощень відповідає мінімальній частоті вихідного сигналу і вимагає 2^{30} прирощень для створення одного періоду.

Генератор працюватиме на цій частоті до надходження нового значення в регістр фазових прирощень. Значення більше "1" швидше проходилимуть повний період 360 градусів, створюючи вищу вихідну частоту (деякі AFG використовують інший підхід: вони підвищують вихідну частоту, пропускаючи деякі вибірки, прискорюючи, тим самим, зчитування вмісту пам'яті).

Єдине, що змінюється, це значення фази, що надходить від контролера частоти. Головну тактову частоту міняти не треба. Крім того, це дозволяє розпочинати генерацію сигналу з будь-якої точки періоду.

Припустимо, нам треба отримати синусоїду, яка починається з максимуму позитивної напівхвилі. Елементарна математика каже, що це максимум відповідає фазі 90 градусів. . Отже:

$$2^{30} \text{ прирощень} = 360^\circ;$$

$$i \ 90^\circ = 360^\circ / 4;$$

$$\text{отже, } 90^\circ = 2^{30}/4$$

Коли фазовий акумулятор отримує еквівалентне значення ($2^{30} / 4$), він змушує пам'ять сигналів почати генерацію з точки, що містить позитивний максимум синусоїдального сигналу.

У пам'яті готових сигналів типового AFG зберігається кілька стандартних сигналів. Зазвичай, найчастіше застосування знаходять синусоїдальні сигнали та меандри. Сигнали довільної форми зберігаються в області пам'яті, яка доступна для перепрограмування користувачем. Ці сигнали можна визначати з тією ж гнучкістю, як і традиційних AWG. Однак архітектура DDS не підтримує сегментування пам'яті та послідовне виведення сигналів. Такими розширеними можливостями мають лише високопродуктивні AWG.

Архітектура DDS забезпечує виключно високу швидкість перебудови частоти, спрощуючи програмування частотних та фазових змін, що корисно для тестування пристроїв, які використовують частотну модуляцію, наприклад, компонентів радіостанцій та супутникових систем. І якщо вам вистачає частотного діапазону AFG, то такий генератор ідеально підходить для тестування ЧМ та телефонних технологій зі стрибкоподібною перебудовою частоти, таких як GSM.

І, хоча AFG не має можливості створення практично будь-яких форм сигналів, як це робить AWG, він може відтворювати більшість широко поширених сигналів, що використовуються в лабораторіях, ремонтних центрах та конструкторських відділах. Крім того, він забезпечує чудову швидкість перебудови частоти. І, що важливо, AFG найчастіше є найекономічнішим рішенням.

1.2.2 Генератори сигналів довільної форми (AWG)

Генератор сигналів довільної форми (AWG) може створити сигнал будь-якої форми. При цьому для створення потрібного сигналу можна використовувати безліч методів — від математичної формули до «рисунок» сигналу.

По суті, AWG являє собою складну систему відтворення, яка створює сигнали на основі збережених цифрових даних, що описують рівні напруги сигналу змінного струму, що постійно змінюються. Блок-схема цього приладу проста.

Якщо звернутися до більш звичних термінів, AWG можна уявити як плеєр

компакт-дисків, який зчитує збережені дані в реальному часі (у AWG — з власної пам'яті сигналу; в CD плеєрі — з диска).

Обидва ці пристрої видають на вихід аналоговий сигнал. Для розуміння принципу роботи AWG необхідно спочатку засвоїти загальну концепцію цифрової дискретизації.

Суть цифрової дискретизації повністю описується її назвою: вона визначає сигнал за допомогою дискретних вибірок, або точок даних, що являють собою послідовність вимірних напруг уздовж графіка сигналу.

Ці вибірки можна визначити, реально вимірюючи сигнал, наприклад, осцилографом, або використовуючи графічні чи математичні методи. На рисунку 1.4 (ліворуч) показано серію вибірок.

Усі вибірки отримані через рівні інтервали часу, хоча у вигляді кривої може здатися, що інтервали не рівні. У AWG значення вибірок зберігаються у двійковій формі в швидкому Оперативному Запам'ятовуванні (ОЗУ).



Рисунок 1.4 — Серія вибірок, що представляють синусоїду (ліворуч) та реконструйована синусоїда (праворуч)

Використовуючи збережену інформацію, сигнал можна у будь-який час реконструювати, зчитуючи значення з пам'яті та пропускаючи їх через цифро-аналоговий перетворювач (ЦАП). Результат показаний малюнку 1.4(праворуч). Зверніть увагу, що вихідний ланцюг AWG містить фільтр, який об'єднує окремі точки в чистий безперервний вихідний сигнал. У результаті досліджуваний пристрій не «бачить» окремих точок сигналу, а сприймає його як безперервну аналогову хвилю.

Спрощена блок-схема AWG, що реалізує описані функції, показано на рисунку 1.5.

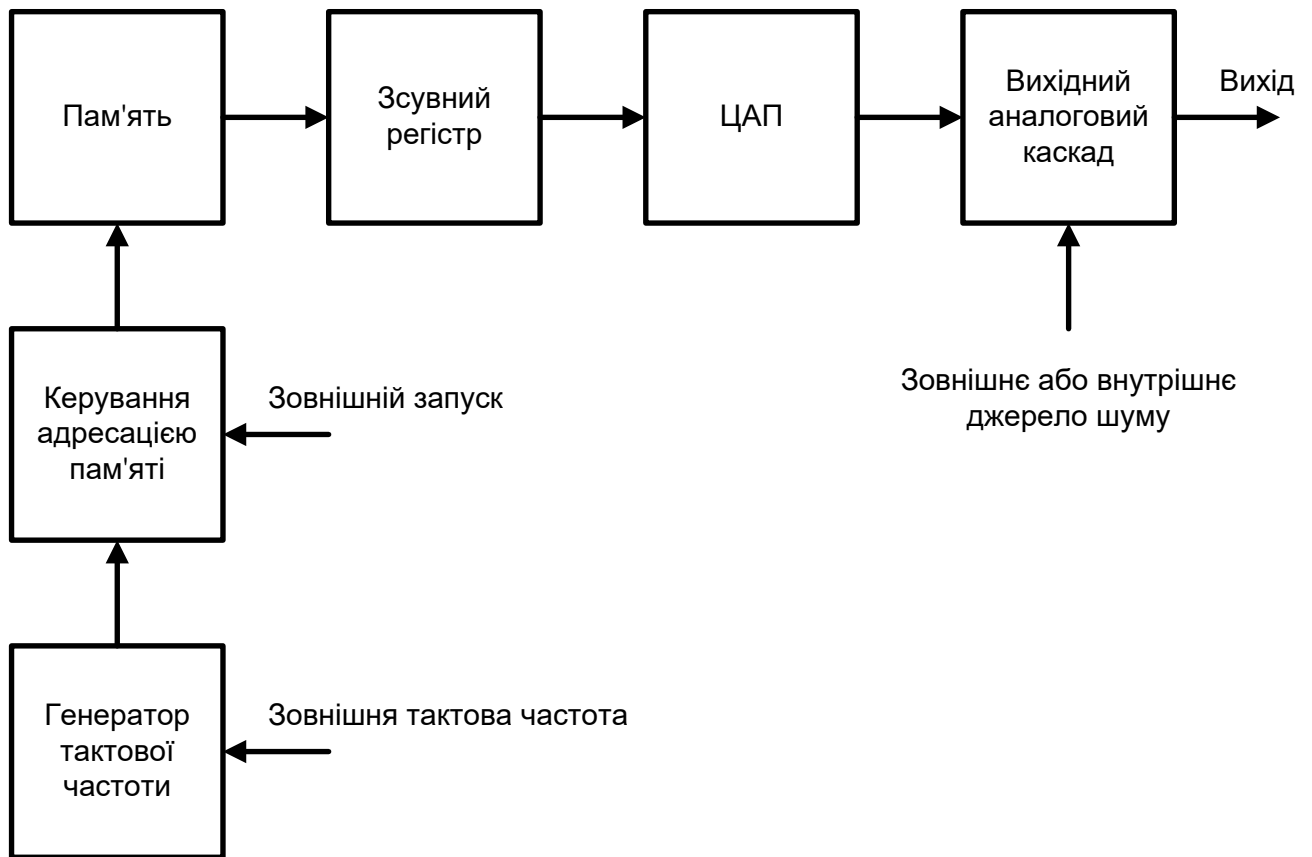


Рисунок 1.5 — Спрощена архітектура генератора сигналів довільної форми

AWG пропонує гнучкість, що практично не досяжна іншими приладами. Маючи можливість відтворювати сигнал будь-якої мислимої форми, AWG може застосовуватися практично скрізь, від моделювання роботи антиблокувальної системи гальм автомобіля до тестування бездротових мереж у граничних режимах[1, 2].

1.2 Огляд існуючих генераторів сигналу довільної форми

Універсальні генератори сигналів Rigol DG5352 (рис. 1.6) використовують прямий цифровий синтез (технології DDS), який забезпечує стабільні, точні та «ідеальні» синусоїдальні сигнали з мінімальними спотвореннями. Генератори сигналів Rigol DG5352 прості у використанні: мають користувацькі інтерфейси та зручне управління, оскільки є візуальні інструкції. Вбудовані функції модуля AM, FM, PM і FSK значно розширюють діапазон використання генератора.

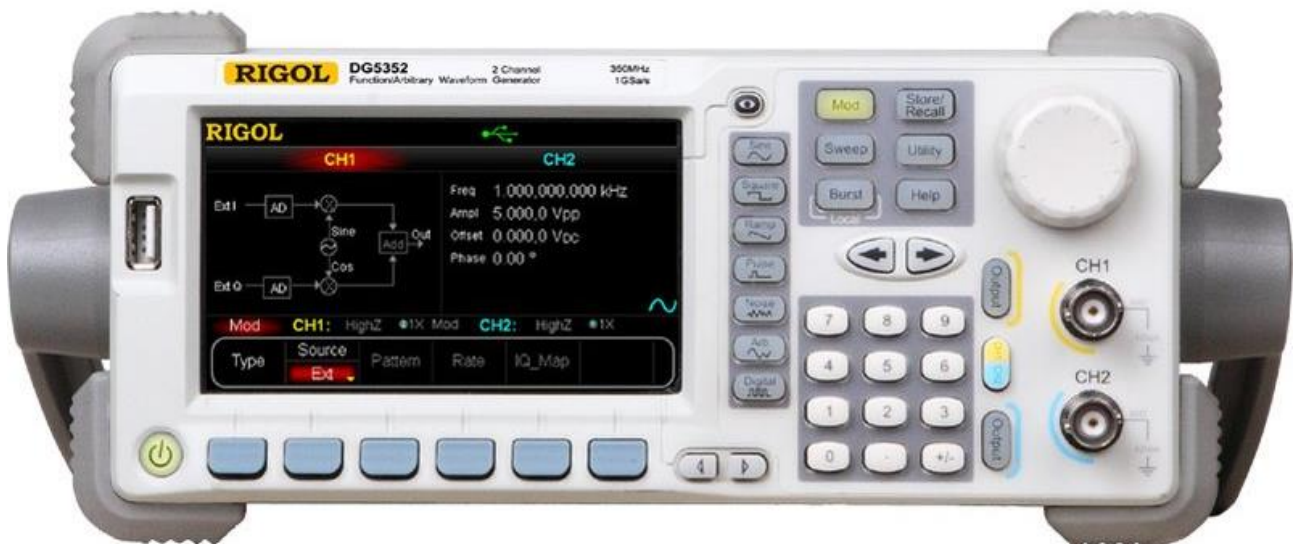


Рисунок 1.6 — Зовнішній вигляд універсального генератора Rigol DG5352

Звукові генератори Rigol DG5352 можуть імітувати сигнали датчиків, імітувати електронні схеми, це дозволяє використовувати звук для встановлення та тестування різного обладнання, а також для обслуговування та досліджень.

Основні можливості та характеристики цього генератора:

- РК-дисплей має діагональ 4,3 дюйма;
- максимальний синусоїдальний сигнал Basomad становить 100 МГц;
- вибір частоти 1 Гбіт/с;
- роздільна здатність 14 біт;
- можливість підключення цифрових модулів на 16+2 канали, зовнішнього підсилювача потужності та підтримки частоти трафіку (в комплект не входить);

— технології прямого цифрового синтезу: вихідні сигнали точні та стабільні

Має 14 стандартних сигналів:

- синус;
- імпульс;
- прямокутниковий;
- пилкоподібний;
- шум;
- синуси серця (раковина);

- зростання за експоненціальним законом (експоненціальний ріст);
- експоненціальне спадання;
- кардіотонічний (ЕКГ);
- функції Гауса;
- гаверсинус;
- функції Лоренца;
- двотональний;
- постійний струм (dc).

PROTEK 9305 одна з наймасовіших і найвідоміших серій генераторів сигналів довільної форми — Protek 9305 (рис. 1.7).

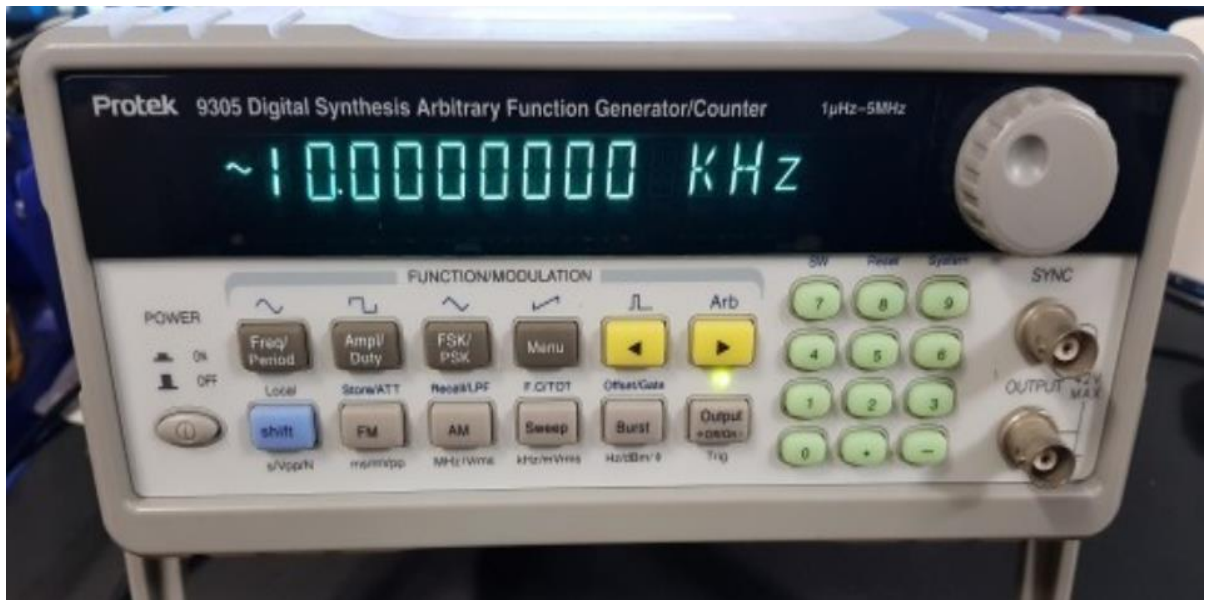


Рисунок 1.7 — Зовнішній вигляд генератора сигналів довільної форми Protek 9305

Основні можливості та характеристики цього генератора:

- формування сигналу шляхом прямого синтезу (DDS);
- діапазон частот основних типів сигналів від 100 до 5 МГц;
- роздільна здатність по вертикалі 12 бітів, частота дискретизації 200 МГц;
- пам'ять форми сигналу 4096 точок;
- висока точність установки коефіцієнта заповнення – до 1/1000;
- режими модуляції сигналу: АМ, ФМ, ЧС, ІМ;

- висока точність і роздільна здатність установки ЧС сигналів;
- плавне регулювання фази у сигналах із високочастотним заповненням;
- роздільна установка частот старту та зупинки в режимі колювання частоти;
- довільне встановлення глибини АМ (від 1 до 120%);
- 10 каналів пам'яті для сигналів, що синтезуються;
- 27 типів стандартних вихідних сигналів (плюс довільно задаються сигнали);
- вбудований частотомір із частотою вимірювання до 100 МГц;
- RS232C інтерфейс, GPIB інтерфейс (опція).

Інші генератори сигналів довільної форми Protek відрізняються тільки більш високою максимальною частотою, що генерується:

- 9310 — до 10 МГц;
- 9320- До 20 МГц;
- 9340- До 40 МГц;
- 9380- До 80 МГц;
- 93120- До 120 МГц.

АНР-4010 Серія генераторів довільної форми АНР, подібна до генераторів фірми Protek.



Рисунок 1.8 — Зовнішній вигляд генератора сигналів довільної форми АНР-4010

Розглянемо коротко їх характеристики:

— форма вихідного сигналу: синус, прямокутник, імпульс, трикутник, сходовий, пилкоподібний, довільний, кардіограма, $\sin(x)/x$, шум, експонента, логарифм тощо;

— частота вихідного сигналу 10 мкГц...10 МГц;

— амплітуда вихідного сигналу 1 мВп-п... 10 Вп-п (50 Ом, $f \leq 20$ МГц), 100 мкВп-п...3 Вп-п (50 Ом, $f > 20$ МГц), 2 мВп-п .. 20 Вп-п (1 МОм, $f \leq 20$ МГц), 200 мВп-п ... 6 Вп-п (1 МОм, $f > 20$ МГц);

— габаритні розміри 240×90×300 мм;

— вага 3 кг.

Інші моделі відрізняються максимальною частотою генерації синусоїдальної напруги:

— АНР_4020 20 МГц;

— АНР_4040 40 МГц;

— АНР_4060 60 МГц;

— АНР_4080 80 МГц;

— АНР_4120 120 МГц.

Існує також аналогічна серія генераторів: ГСС-05, ГСС-05/1, ГСС-10, ГСС-10/1, ГСС-20, ГСС-20/1, ГСС-40, ГСС-40/1, ГСС-80, ГСС-80/1, ГСС-120, ГСС-120/1.

Як неважко зрозуміти, цифри вказують на верхню межу частоти в мегагерцах. Генератори з цифрою 1 під рисою мають підвищену стабільність частоти.

На рис. 1.9 показаний зовнішній вигляд генератора ГСС-80.



Рисунок 1.9 — Зовнішній вигляд генератора сигналів довільної форми ГСС-80

Прилад Hantek DDS-3005 має автоматичне калібрування. На рисунку 1.9 показано зовнішній вигляд генератора Hantek DDS-3005. Генератор Hantek DDS-3005 може створювати стандартні та спеціальні електромагнітні сигнали різних видів: прямокутні, синусоїди, імпульсні, пилкоподібні, тощо.

Універсальність дозволяє застосовувати пристрій під час налагодження, перевірки та налаштування різних каналів зв'язку та високоточних радіоелектронних приладів. Вирізняється вкрай низьким рівнем спотворення, високоточними результатами роботи завдяки дуже низькому рівню похибки та оптимальним співвідношенням ціни та якості. Може працювати з різними пристроями.

Генератор сигналів Hantek DDS-3005 може працювати як підключеним до ПК, і автономно. Є вбудований двоканальний частотомір діапазон якого сягає 2.7 ГГц.

Оскільки пристрій має USB-інтерфейс, для його роботи не потрібний блок живлення або батареї. Користувач може створювати параметри сигналу, малювати

точку за допомогою комп'ютерної миші.

Декілька генераторів серії можна підключати до одного ПК та використовувати в мультисканальному режимі. Одиначний генератор DDS-3005 може працювати як модуль інших пристроїв, видаючи сигнал із встановленими користувачем параметрами частотою до 5 МГц.



Рисунок 1.10 — Генератор сигналів довільної форми Hantek DDS-3005

2 ПРОЕКТУВАННЯ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Структурна схема

Прямий цифровий синтез (Direct Digital Synthesis, або DDS) — один із найпоширеніших методів синтезу частот, заснований на використанні цифрової обробки даних для отримання на виході сигналу необхідної частоти та фази сигналу фіксованої опорної частоти.

Генерований сигнал при цьому має властиву цифрову систему точності. Частота, амплітуда та фаза сигналу у будь-який момент часу відомі та легко регулюються. Генератори DDS практично не схильні до температурного дрейфу і старіння. Єдиним елементом, який має властиву аналогову схему нестабільність, є ЦАП. Високі технічні характеристики спричинили те, що останнім часом DDS-генератори витісняють як звичайні синтезатори частот, і аналогові функціональні генератори [9, 10].

Основні переваги методу DDS:

- висока роздільна здатність за частотою і фазою, управління якими здійснюється в цифровому вигляді;
- швидкий перехід в іншу частоту (час переходу становить частки мікросекунд). Швидкість перебудови обмежена практично лише швидкодією цифрового керуючого інтерфейсу;
- через малий крок перебудови по частоті не потрібно точне підстроювання опорної частоти, замість неї легко ввести код поправки;
- оскільки вихідний сигнал синтезується у цифровому вигляді, то обмежень у виборі форми сигналу практично немає;
- низька вартість, мале енергоспоживання та компактність DDS синтезаторів.

Метод DDS має деякі обмеження:

- максимальна вихідна частота теоретично не може бути вищою половини тактової (на практиці вона в 5–6 разів менша);
- побічні складові вихідного сигналу можуть бути помітнішими порівняно

з синтезаторами інших типів. Спектральна чистота вихідного сигналу DDS залежить від якості ЦАП;

— споживана DDS-генераторами потужність живлення практично прямо пропорційна тактовій частоті і може досягати 100 мВт, при більших тактових частотах DDS можуть виявитися непридатними для пристроїв із батарейним живленням.

Виходячи з того, як побудовані класичні рішення генераторів (див. рис 2.1), варто врахувати існуючі рішення та запропонувати вдосконалення. Принцип роботи синтезатора DDS [2, 9] полягає в наступному: в накопичувачі фази (НФ) формується послідовність кодів вихідного сигналу, що лінійно змінюється в часі.

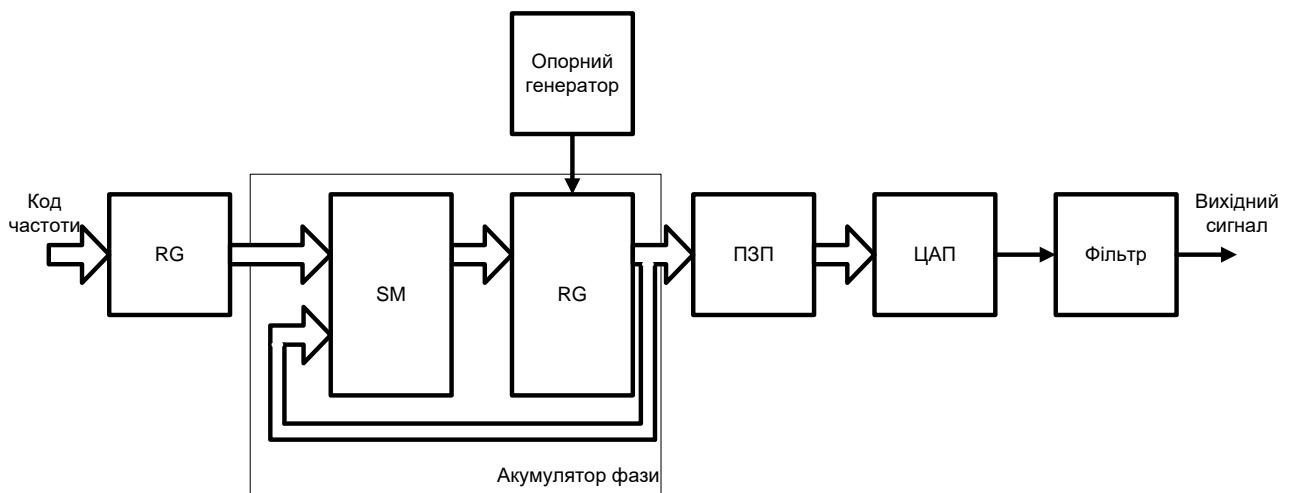


Рисунок 2.1 — Структура прямого цифрового синтезатора частоти

Як накопичувач фази використовується накопичуючий суматор (SM). Він являє собою регістр (RG), що перезавантажується в кожному такті роботи пристрою величиною, що дорівнює його старому вмісту, плюс деяка постійна добавка (код частоти). Вміст регістру RG лінійно збільшується в часі, а збільшення залежить від величини постійної добавки. Нагромаджуючий суматор, що використовується для формування коду фази, називають акумулятором фази. Акумулятор фази працює з періодичними переповненнями, що відповідає періодичній зміні функції і тактується по частоті f_{clk} від генератора (G). Частота переповнення акумулятора дорівнює вихідній частоті, і визначається за формулою:

$f_{out} = K * f_0/2^n$, де f_{out} — вихідна частота, f_0 — тактова частота, K — код частоти, n — розрядність акумулятора фази.

Зміни в накопичувачі фази відбуваються під впливом тактової частоти f_0 , що виробляється опорним генератором (ОГ), та коду частоти, яка генерується синтезатором. Значення амплітуди сигналу, що відповідають поточній фазі сигналу, вибираються з перекодувальної таблиці (Look Up Table), що розміщується в ПЗП. Вибір значення sin проводиться за адресою в ПЗП із потрібною фазою. Значення з комірок пам'яті надходять на цифро-аналоговий перетворювач (ЦАП), де виходить «ступінчастий» синусоїдальний сигнал. Для згладжування вихідного сигналу на виході ЦАП розташовується нижній фільтр частот (ФНЧ).

В якості вдосконалення запропоновано структурну схему (рисунок 2.2) в якій замінити окремі вузли: реєстри, накопичуючий суматор, постійний запам'ятовуючий пристрій, опорний генератор на єдиний блок на базі мікроконтролерної системи, а також покласти на нього функції формувача коду фази, контролю поточної фази, значень для АЦП і функцій керування всією системою.

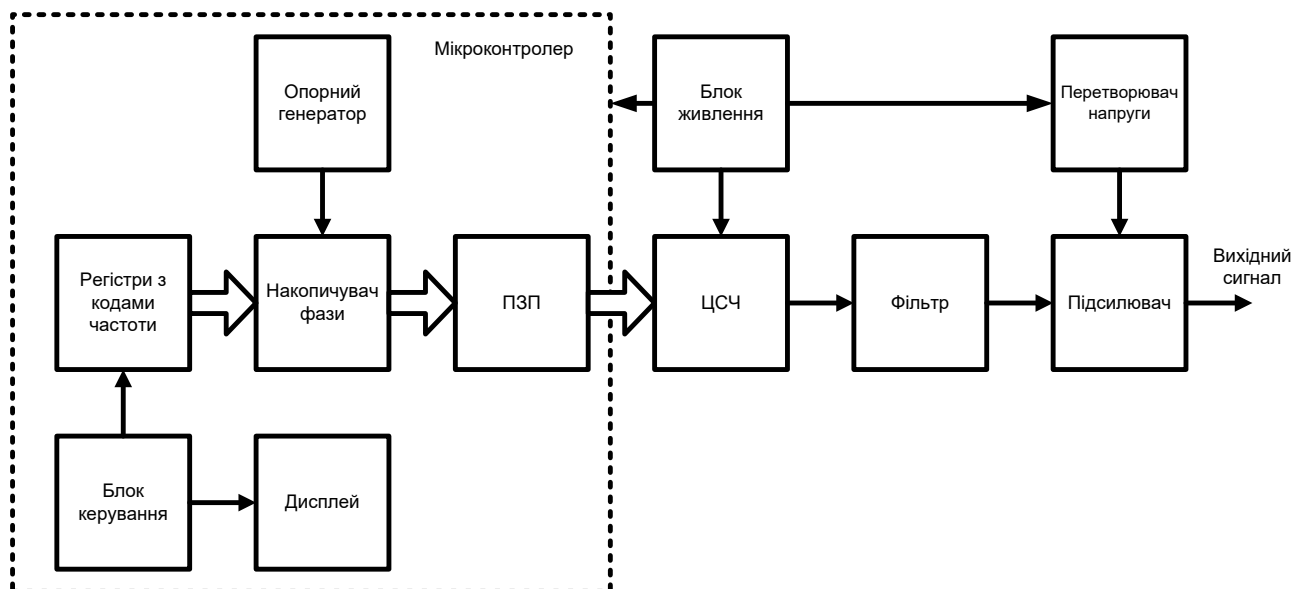


Рисунок 2.2 — Структурна схема системи цифрового генерування аналогових сигналів

ЦАП замінити на сучасний високоточний швидкодіючий цифровий синтезатор частоти (ЦСЧ) з цифровим керуванням із можливістю послідовної та

паралельної передачі даних. Додати підсилювач вихідного сигналу для збільшення сили виходу. Такий підхід дозволить покращити характеристики генерування, підвищити точність, гнучкість управління, можливість подальшого вдосконалення та інтегрування в різні галузі діяльності.

2.2 Вибір мікроконтролера

STM32 — сімейство МК, побудованих з використанням 32-розрядного ядра Cortex різних версій.

МК даного типу побудовані на гарвардській архітектурі та мають 3-ступінчастий конвеєр, який мінімізує час виконання команд. Вони розроблені для побудови систем з максимальною енергоефективністю та мають кілька режимів керування енергоспоживанням. Вони використовуються внутрішні інтерфейси пам'яті шириною більше, ніж середня довжина інструкції. Це мінімізує кількість доступів до шини пам'яті, а відтак і споживання електроенергії, пов'язане з операціями з шини та читанням енергонезалежної пам'яті. Технологія безперервної обробки переривань за винятком внутрішніх операцій над стеком (tail chaining) скорочує час реакції на переривання і виключає зайві операції зі стеком [3-8].

Периферійний вузол може бути поділений на два блоки. Перший блок призначений для виконання завдань, що не вимагають участі процесора, таких як прості завдання передачі даних, керування аналоговими входами або виконання функцій, прив'язаних до синхросигналів. Зв'язок із зовнішнім світом здійснюється за допомогою портів введення/виведення МК. Другий блок призначений для налаштування та управління периферією, за допомогою програми через регістри, з'єднані з внутрішньою шиною, що поділяється з іншими ресурсами МК.

У цій роботі використовується найбільш популярні та поширені версії сімейства МК STM32 – STM32F4 Discovery (рисунок 2.3) та STM32F429I Disco (рисунок 2.4).

2.2.1 Опис плати STM32F4 Discovery

Основним елементом плати STM32F4 Discovery є мікроконтролер

STM32F407VGT6 із ядром Cortex-M4F тактовою частотою 168 МГц. Загальна кількість Flash-пам'яті, доступної для зберігання програми, що завантажується 1 Мб, а також 192 кб ОЗП в корпусі LQFP-100, що надає в цілому 100 контактів. Живлення плати здійснюється через USB або від зовнішнього джерела живлення 5 В. Для налагодження та програмування МК є відладчик ST-Link/V2.



Рисунок 2.3 — Зовнішній вигляд плати STM32F4 Discovery

Також плата STM32F4 Discovery оснащена:

- датчиком руху ST MEMS LIS302DL та виходами цифрового акселерометра по трьох осях;
- звуковим ЦАП CS43L22;
- датчиком звуку ST MEMS MP45DT02;
- двома кнопками;
- вісьмома світлодіодами: — LD1 (червоний/зелений) для USB-підключення, LD2 (червоний) для живлення 3.3 В, чотири користувальницькі

світлодіоди: LD3 (помаранчевий), LD4 (зелений), LD5 (червоний), LD6 (синій) та два світлодіода для USB On-The-Go — LD7 (зелений) та LD8 (червоний).

Мікроконтролер STM32F407VGT6 у свою чергу містить:

- п'ять 16-розрядних портів введення/виведення загального призначення;
- чотирнадцять таймерів;
- три АЦП, розрядність яких становить 12 біт;
- шість приймачів-передавачів, чотири з яких є синхронно-асинхронними (USART) та два асинхронними (UART);
- два контролера DMA, які забезпечують загалом 16 потоків, кожен з яких може забезпечувати до 8 каналів (запитів).

Таким чином, налагоджувальна плата оснащена великою кількістю периферії.

2.2.2 Опис плати STM32F429I Disco

Налагоджувальна плата STM32F429I Disco надає більше можливостей, оскільки містить кольоровий графічний РК-дисплей [11].

Плата STM32F429I Disco має:

- мікроконтролер STM32F429ZIT6 з ядром Cortex-M4 тактовою частотою 180 МГц, 2 Мб Flash-пам'яті, 256 кб ОЗП у корпусі LQFP-144;
- живлення плати через USB або від зовнішнього джерела живлення 5 В;
- відлагоджувач ST-Link/V2 для налагодження та програмування МК;
- послідовний звуковий інтерфейс SAI;
- контролер зовнішньої пам'яті Flexible Memory Controller (FMC);
- інтегрований контролер РК-дисплей та графічний прискорювач Chrom-ART (DMA2D);
- дві кнопки (для програмування користувачем та для перезапуску).
- з) чотири світлодіоди: LD1 (червоний/зелений) для USB-підключення, LD2 (червоний) для живлення 3.3 В, два користувальницькі світлодіоди: LD3 (зелений), LD4 (червоний).



Рисунок 2.4 — Зовнішній вигляд плати STM32F429I Disco

Найбільшу увагу при роботі з платою приділено графічному РК-дисплею (типу TFT) з роздільною здатністю 240×320 пікселів. У свою чергу, контролер дисплея LTDC забезпечує 24-бітний RGB-інтерфейс і всі необхідні сигнали для взаємодії з РК-панелями з роздільною здатністю до 640×480. Він має два екранні шари з буферами FIFO (64×32 біт), до восьми форматів кольору на кожен шар та таблицю кольорів (Color Look-Up Table) з підтримкою до 256 24-бітових кольорів на кожен шар [11].

2.3 Цифровий синтезатор частоти

У роботі використовувався цифровий синтезатор частоти AD9850. Його

основні технічні характеристики [12, 13] представлені у таблиці 2.1.

Таблиця 2.1 — Основні технічні характеристики AD9850

Частота тактового генератора	від 1 до 125 МГц
Робочий діапазон частот	від 1 Гц до 40 МГц
Види сигналів	Прямокутний, sin
Максимальний струм споживання (при $f_{in} = 125$ МГц)	96 мА
Кількість розрядів ЦАП	10
Максимальний вихідний струм ЦАП (при $R_{set} = 3.9$ к)	10.24 мА
Максимальна інтегральна нелінійність ЦАП	1 МЗР
Мінімальна напруга високого рівня на виході	4.8 В
Максимальна напруга низького рівня на виході	0.4 В

Призначення контактів [13] модуля DDS синтезатора (рисунок 2.5) наведено у таблиці 2.5.

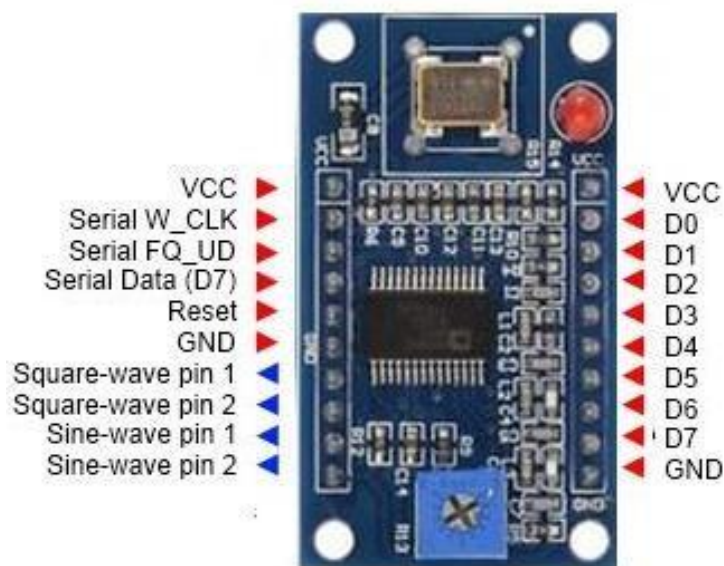


Рисунок 2.5 — Зовнішній вигляд синтезатора AD9850

Таблиця 2.2 — Призначення контактів синтезатора

VCC	Живлення 3.3 або 5 В
-----	----------------------

GND	Заземлення
W_CLK	Синхронізація паралельних чи послідовних сигналів частоти (фази, керування)
FQ_UD	Оновлення частоти
DATA (D7)	Послідовне завантаження даних
RESET	Скидання значень
D0-D7	Восьмибітний порт для паралельного введення даних
Square Wave	Прямокутні вихідні коливання
Sine Wave	Синусоїдальний вихідний сигнал

Завантаження даних модуль може здійснюватися за допомогою послідовного або паралельного інтерфейсу.

При використанні послідовної передачі [12, 13] в мікросхемі (рисунок 2.6), на вхід DATA (D7) подається слово довжиною 40 біт. Кожен біт супроводжується позитивним імпульсом контакту синхронізації W_CLK. Після завантаження слова, що управляє, з подачею позитивного імпульсу на вхід FQ_UD відбувається оновлення параметрів генерації.

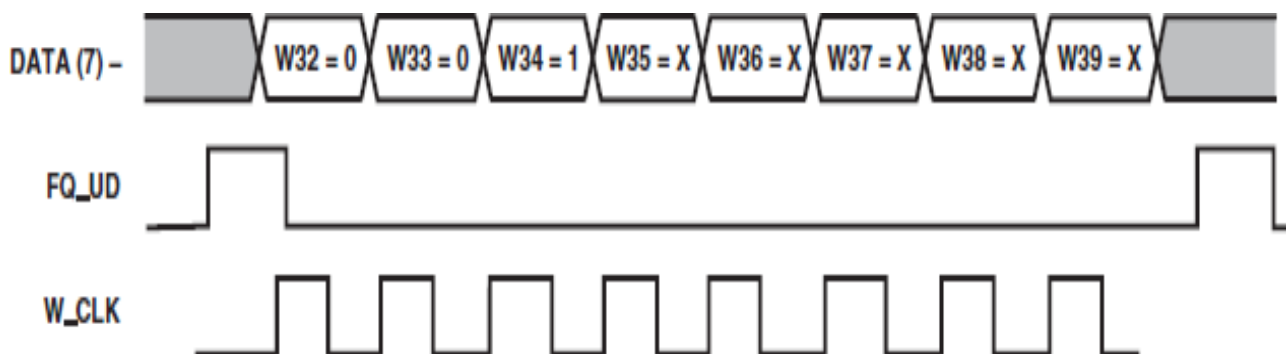


Рисунок 2.6 — Часова діаграма під час послідовної передачі даних

При використанні паралельного інтерфейсу [12, 13], лініями D0-D7 модуль передаються п'ять управляючих слів по вісім біт (рисунок 2.7). Кожне слово супроводжується сигналом синхронізації W_CLK. Оновлення даних відбувається з настанням імпульсу на вхід FQ_UD.

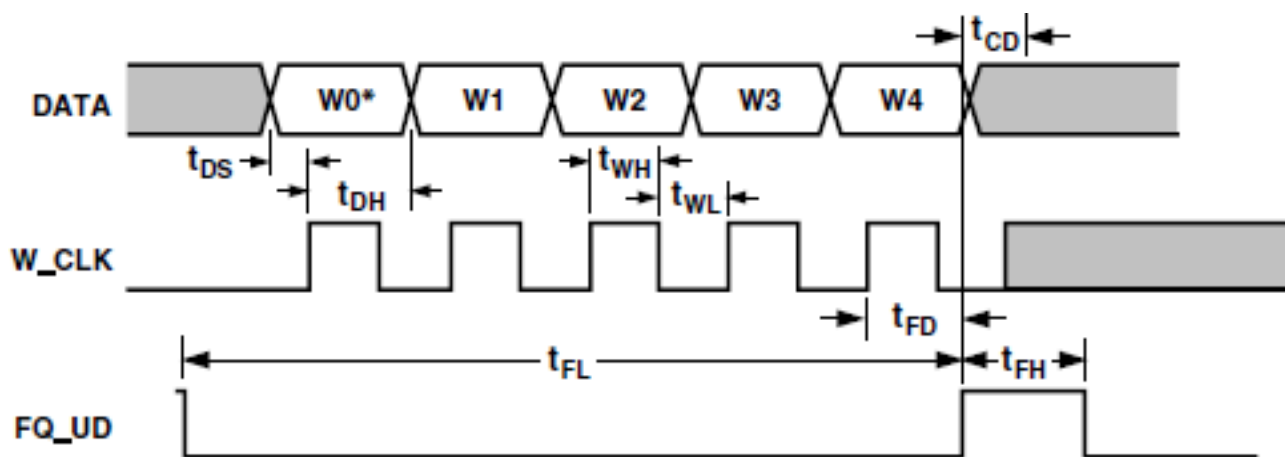


Рисунок 2.7 — Часова діаграма під час паралельної передачі даних

Паралельний інтерфейс забезпечує швидшу передачу даних, але задіює більше управляючих контактів. Насправді частіше використовується варіант з послідовним інтерфейсом, як зручніший для програмної та апаратної реалізації.

2.4 Перетворювач напруги

MAX743 — двоканальний, імпульсний стабілізатор (ІС) (від +5 до ± 15 В або ± 12 В) (рисунок 2.8). Інтегральний DC/DC перетворювач напруги MAX743 має всі активні ланцюги, необхідні для побудови малогабаритних, двоканальних джерел живлення. Реалізоване, з використанням стандартних дроселів з двома виводами, а не з трансформаторами, схемне рішення дозволяє, незалежно стабілізувати обидві вихідні напруги, з похибкою в межах $\pm 4\%$, в діапазоні допустимих вхідних напруг, температури і струмів навантаження.

ІС MAX743 типово забезпечує ефективність перетворення від 75% до 82%, для більшості значень навантаження. ІС функціонує як струмовий зворотний зв'язок на частоті 200 кГц, тобто, можливе спільне використання малогабаритних, легких зовнішніх елементів. Також це дозволяє реалізувати просту фільтрацію вихідних пульсацій і шумів.

ІС MAX743 має підвищену надійність, завдяки вбудованим силовим транзисторам та монолітній конструкції. Функція термовідключення запобігає перегріву, а обмеження струму всередині кожного циклу перетворення захищає силові транзистори. До інших функцій відносяться система блокування запуску при

нестачі вхідної напруги та програмований м'який старт [9]. Схема підключення показано на рисунку 2.8.

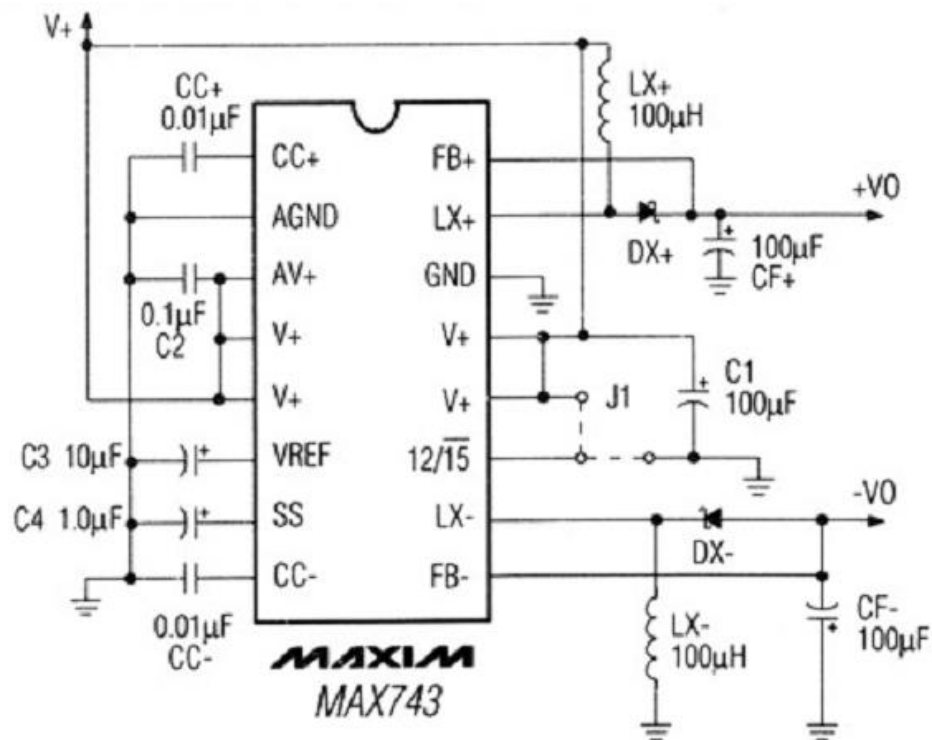


Рисунок 2.8 — Типова схема включення MAX743

Відмінні особливості:

- здатність навантаження: ± 100 мА, або ± 125 мА;
- похибка вихідної напруги $\pm 4\%$ у діапазоні температур, вхідних напруг та навантаження;
- типова ефективність перетворення 82%;
- низький рівень шумів, зворотний по струму;
- вбудована система обмеження струму;
- система термозахисту з режимом Shutdown;
- система захисту від запуску при зниженій напрузі та м'який старт;
- логічне управління перемиканням режимів ± 12 В та ± 15 В;

Області застосування:

- модульна заміна DC/DC перетворювачів;
- системи розподілу енергії;

- комп'ютерна периферія;
- портативні вимірювальні прилади.

Імпульсний перетворювач MAX743 обраний і використовується через те, що дозволяє отримати двополярну напругу $\pm 15\text{В}$. Цей перетворювач здійснює підвищення до $\pm 15\text{В}$. Він є універсальним, який дозволяє отримувати $\pm 15\text{В}$, або $\pm 12\text{В}$. Вибір одержуваних на виході напруги здійснюється завдяки виводу 11, якщо цей вивід з'єднати на загальний провід, то вихідна напруга буде генеруватися в діапазоні $\pm 15\text{В}$ як на схемі, якщо з'єднати на $+5\text{В}$ живлення, тобто подати логічну одиницю, то у нас відбудеться генерація діапазон $\pm 12\text{В}$.

2.5 Фільтр низьких частот

ФНЧ потрібний для згладжування вихідного сигналу, який має «ступінчастий» вигляд.

ФНЧ з апроксимацією прямокутної амплітудно-частотної характеристики (АЧХ) за Баттервортом характеризується максимально плоскою і монотонно спадаючою АЧХ.

Аналітична залежність коефіцієнта передачі даного фільтра $K(F)$ має вигляд:

$$K(F) = \frac{1}{\sqrt{(1 + F^{2n})}}$$

де $F = f / f_c$ — нормована частота,

f — частота,

f_c — частота зрізу,

n — порядок фільтра.

Графіки $K(F)$ для $n = 1, 2, 3$ представлені на рис. 2.9.

З графіків видно, що чим вищий порядок фільтра n , тим більш плоскою стає АЧХ у смузі пропускання ($F \leq 1$) і крутішою за її межами ($F > 1$), наближаючись до прямокутної.

На нульовій частоті коефіцієнт передачі дорівнює 1. На частоті зрізу ($F = 1$) коефіцієнт передачі, як це впливає з формули дорівнює:

$$K(1) = \frac{1}{\sqrt{2}} = 0.707$$

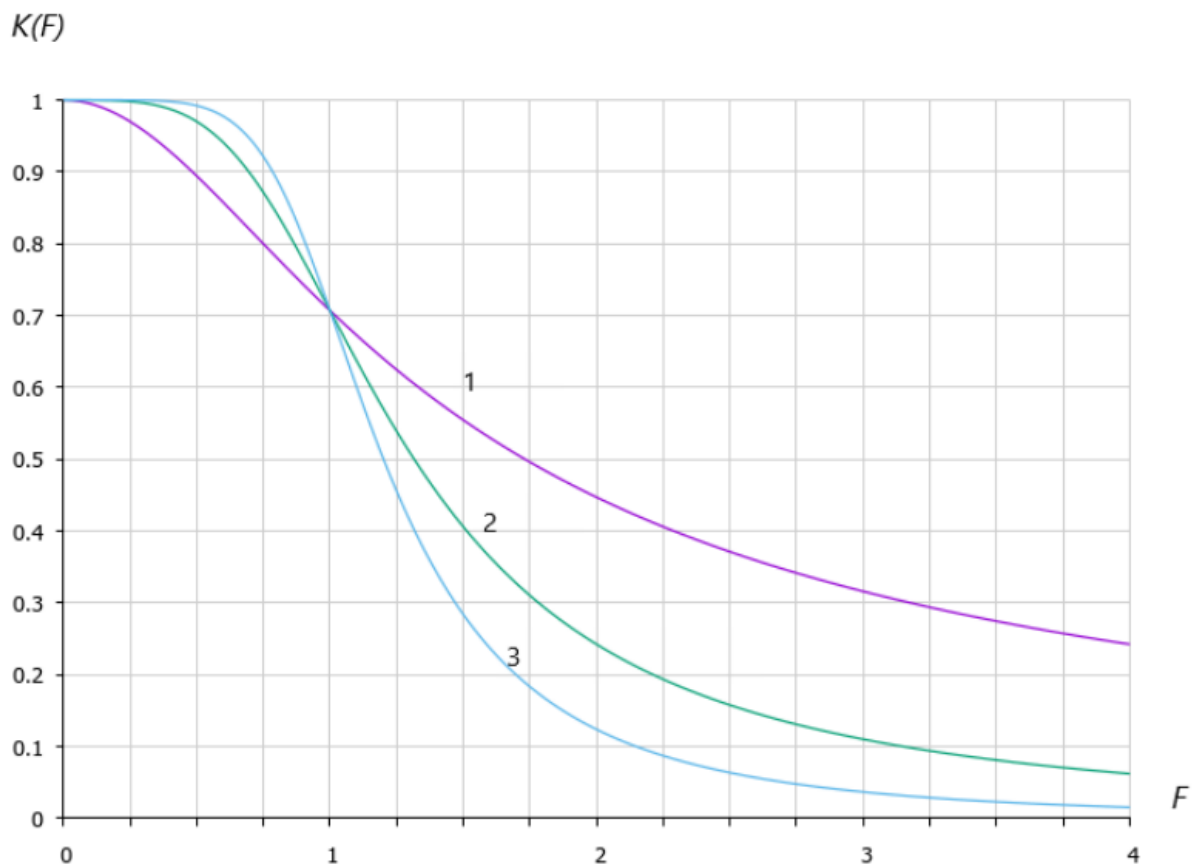


Рисунок 2.9 — Графіки залежності $K(F)$

На практиці АЧХ прийнято виражати в децибелах через десятковий логарифм:

$$K_{db}(F) = 20 \lg K(F).$$

Графіки $K_{db}(F)$ для $n = 1, 2, 3$ представлені на рис. 2.10.

На нульовій частоті коефіцієнт передачі дорівнює 0 дБ. На частоті зрізу ($F = 1$) коефіцієнт передачі $K_{db}(1)$ дорівнює -3 дБ.

Спад АЧХ області досить віддаленої від частоти зрізу ($F > 2$) становить залежно від порядку фільтра:

- $n = 1$, спад 6 дБ на октаву;
- $n = 2$, спад 12 дБ на октаву;

— $n = 3$, спад 18 дБ на октаву.

У разі швидкість спаду для ФНЧ Баттерворта $n \cdot 6$ дБ/октава.

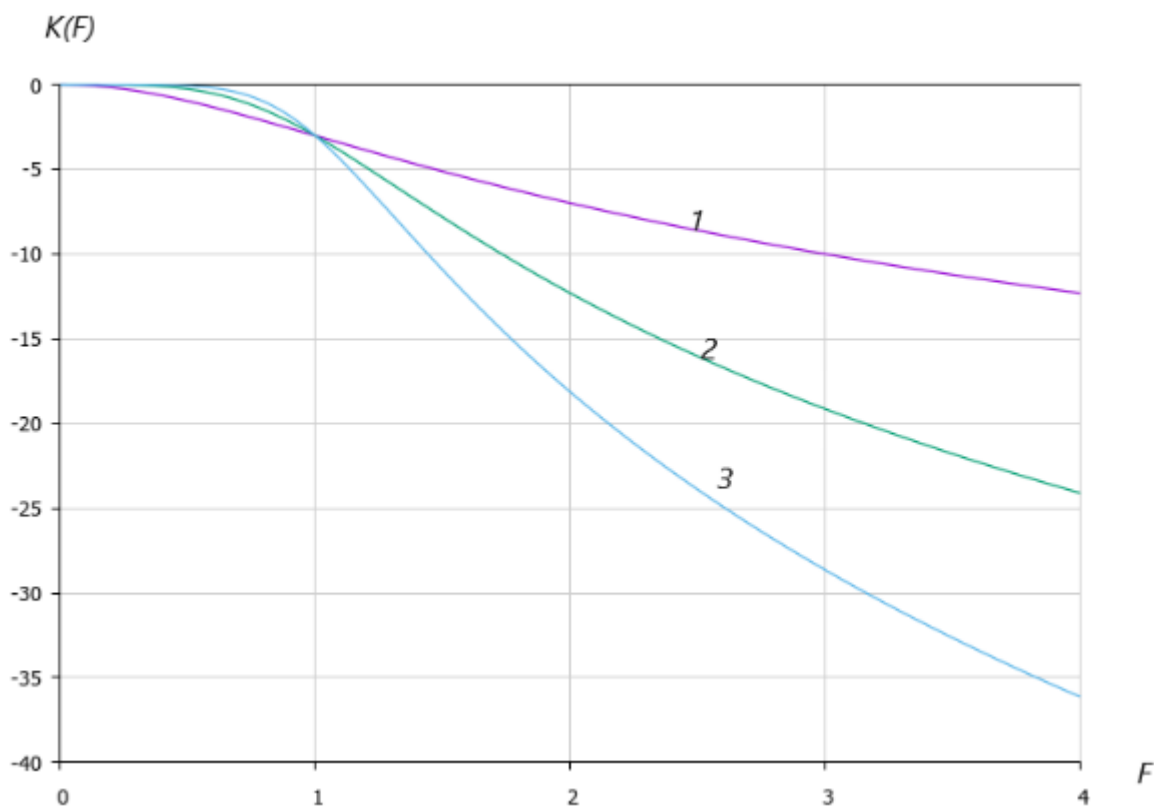
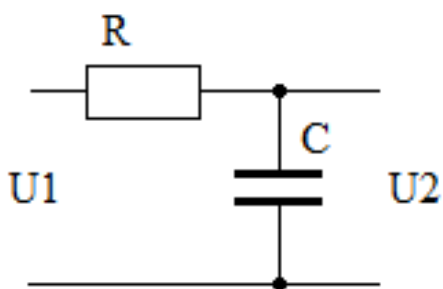
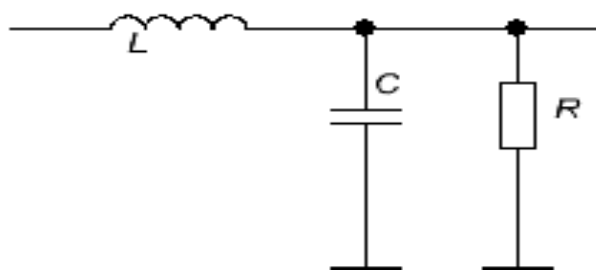


Рисунок 2.10 — Графіки залежності $Kdb(F)$

Фільтри, що реалізують цей вид АЧХ, прийнято називати «фільтрами Баттерворта» (рисунок 2.11).



а) Фільтр на елементах RC



б) Фільтр на елементах LC

Рисунок 2.11 — Фільтр Баттерворта

Зауважимо, що з $n = 1$ фільтр Баттерворта може бути реалізований звичайним RC ланцюгом (рис. 2.11, а). Наступним за складністю, але з більш крутим спадом, що дорівнює 12 дБ/октава, буде наступний фільтр (рис. 2.11, б).

Перейдемо до загальної випадку, коли потрібно зробити розрахунок при заданих значеннях крутості спаду АЧХ. Для цього потрібно визначити порядок фільтра. Порядок фільтра n визначає величину спаду АЧХ у смузі придушення (непропускання). Відповідно він визначає складність фільтра. Для розрахунку фільтра нижніх частот прийнято задавати такі параметри:

f_c — частота зрізу фільтра і відповідно коефіцієнт передачі фільтра цієї частоти K_{bc} . Зазвичай приймається $K_{dbc} = -3$ дБ, що відповідає $K(1) = 0.707$.

f_r — частота, з якої починається задане ослаблення фільтра K_{dbr}

Виходячи з цих параметрів, обчислюється порядок фільтра Баттерворта:

$$n = \frac{1}{2} \frac{\ln(10^{-K_{dbr}/10} - 1)}{\ln q}$$

Ця формула виводиться із вищезгадано формули, де $q = f_r/f_c$. Очевидно, що завжди $q > 1$. Число n округляється до найближчого цілого.

Перейдемо до реалізації ФНЧ Баттерворта. При цьому реалізацій може бути безліч, наприклад, на LC-елементах, на операційних підсилювачах, на перемикаються конденсаторах, на цифрових обчислювальних пристроях та ін. Тут ми обмежимося реалізацією на LC-елементах, тобто на котушках індуктивності та конденсаторах. Наведемо два види реалізації фільтра Баттерворт на LC-елементах.

На рисунку 2.12 показано два варіанти реалізації ФНЧ Баттерворта 5-го порядку на основі П-і Т-подібних LC-ланок.

Зауважте, що нумерація елементів загальна для L і C. Так зручно для розрахунків. Закономірність побудови схем досить проста. Для ФНЧ Баттерворта порядок фільтру дорівнює кількості LC-елементів. Опір джерела R_i та навантаження R_n беруться не довільним чином. Їх значення пов'язані з величинами елементів L і C. При довільних значеннях R_i і R_n фільтр не погоджується і його

АЧХ буде відрізнятися від бажаної. Зауважимо також, що на нульовій частоті схеми вироджуються в простий дільник резистивного напруги на резисторах R_i і R_n . Зазвичай приймають $R_i = R_n$. При рівності R_i і R_n коефіцієнт передачі дорівнює $\frac{1}{2}$ або -6 дБ. Це означає, що на частоті зрізу він повинен дорівнювати: $-6 - 3 = -9$ дБ.

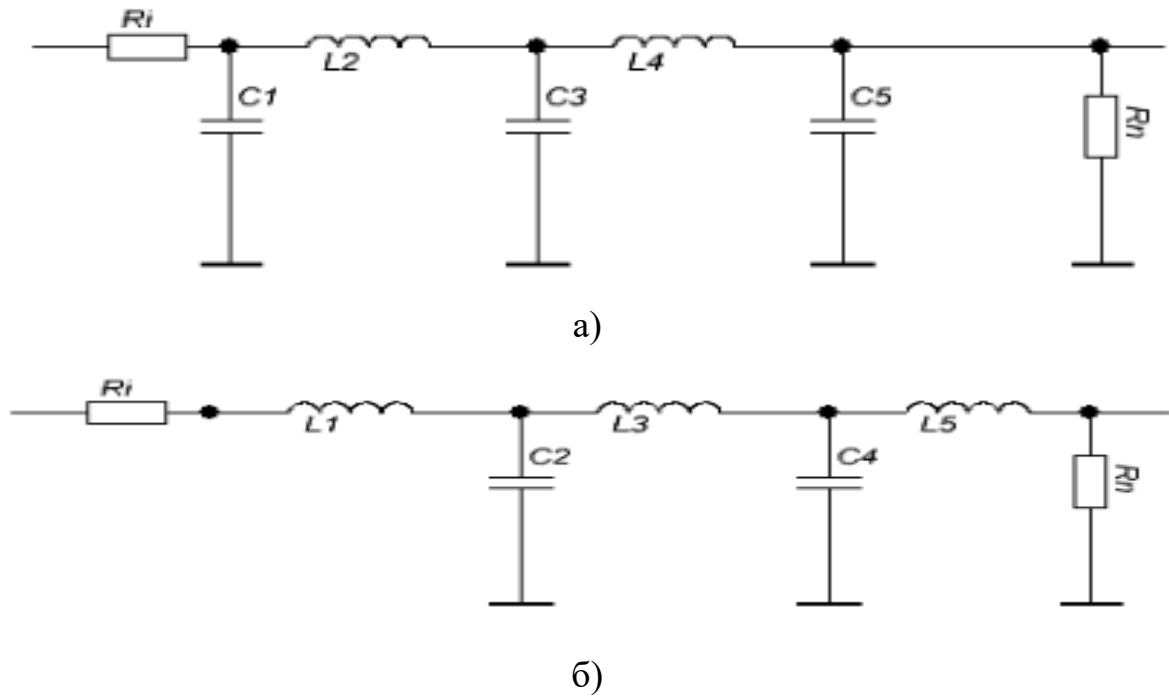


Рисунок 2.12 — Фільтр Баттерворта 5-го порядку на основі LC-ланок: а) П-подібних; б) Т-подібних

У теоретичній літературі прийнято розраховувати фільтр попередньо для значень $R = 1$ Ом і Гц. Такий фільтр називається нормованим. Для першої схеми наведені такі формули нормованого ФНЧ Баттерворта:

$$C_m = 2 \sin \left(\frac{(2m-1)\pi}{2n} \right), \quad \text{де } m - \text{ непарно,}$$

$$L_m = 2 \sin \left(\frac{(2m-1)\pi}{2n} \right), \quad \text{де } m - \text{ парно.}$$

Для іншої схеми C_m та L_m змінюються місцями:

$$L_m = 2 \sin \left(\frac{(2m-1)\pi}{2n} \right), \quad \text{де } m - \text{ непарно,}$$

$$C_m = 2 \sin\left(\frac{(2m-1)\pi}{2n}\right), \text{ де } m - \text{ парно.}$$

Значення виходять у Гн (Генрі) для L_m і Ф (Фарадах) для C_m .

Далі їх слід привести до значень реального фільтра, для якого має бути задано:

f_c — потрібна частота зрізу Гц.

R — необхідний опір навантаження (джерела).

Формули для перерахунку наступні:

$$L_{m \text{ real}} = L_m \left(\frac{R}{2\pi f_c} \right),$$

$$C_{m \text{ real}} = C_m \left(\frac{1}{2\pi f_c R} \right).$$

Для перевірки результатів розрахунку бажано змодельювати отриманий фільтр та переконатися в їх правильності. Використовуємо для цього безкоштовний пакет моделювання MicroCap-12 (рисунок 2.13).

На рисунку 2.14 наведено схеми розрахованих вище фільтрів та графіки їх АЧХ.

Результати розрахунків показано на рисунку 2.15. Значення C і L відповідають першому варіанту схеми, LL та CC — іншому. m — номери елементів у схемах. Там, де значення дорівнюють нулю, означає, що елементів із такими номерами m не існує. Зміна n , R і f_c у цій програмі дає змогу розглянути випадки, які цікавлять.

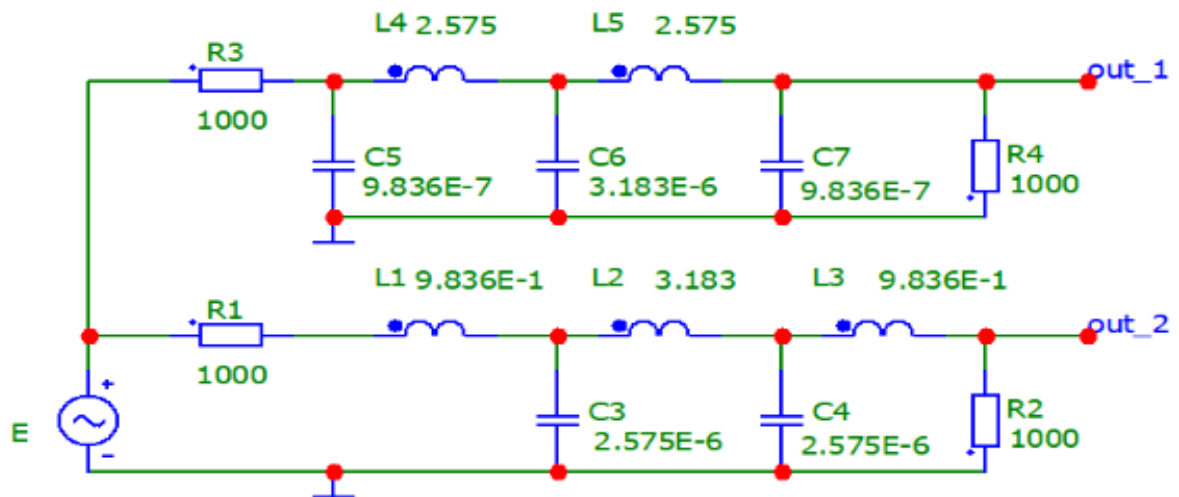


Рисунок 2.13 — Фільтр Баттерворта 5-го порядку на основі LC-ланок в MicroCap 12

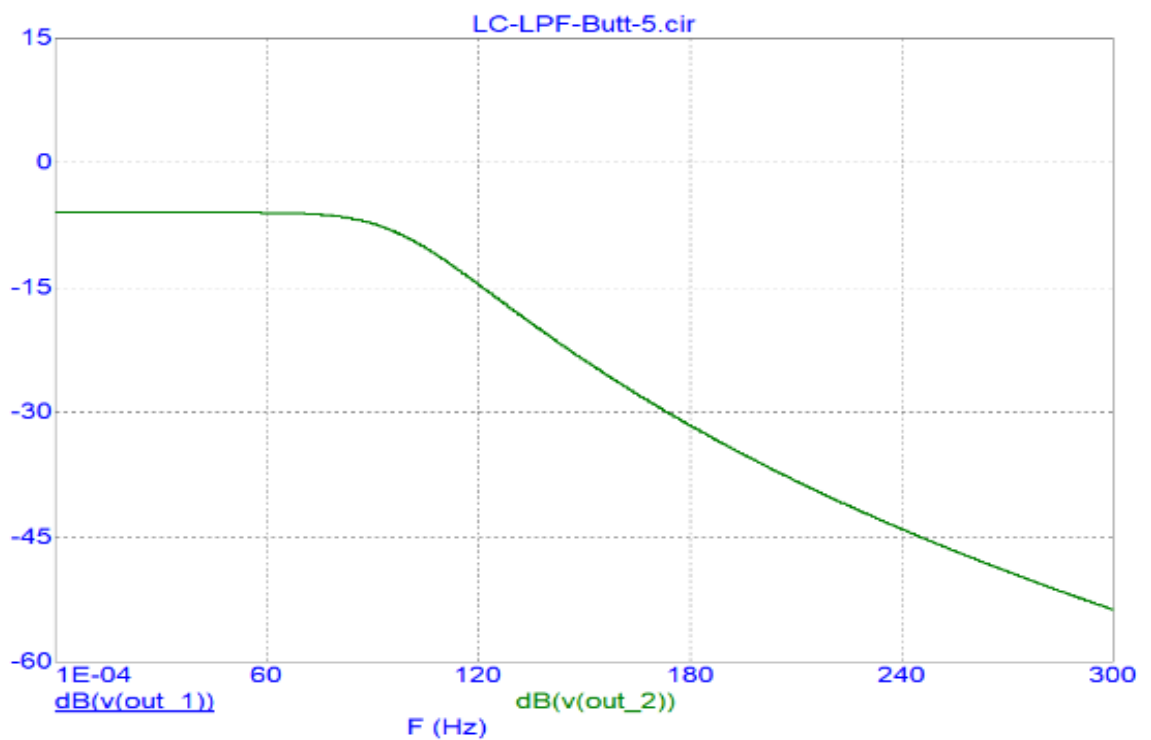


Рисунок 2.14 — АЧХ фільтрів Баттерворта 5-го порядку на основі LC-ланок

Обидві характеристики збігаються. Як і потрібно у завданні, зрізи (-3 дБ щодо нульової частоти або -9 дБ в абсолютному значенні) знаходяться на частоті 100 Гц.

стор	m	CL	LL	CC	
1	1.000E+0	9.836E-7	0.000E+0	9.836E-1	0.000E+0
2	2.000E+0	0.000E+0	2.575E+0	0.000E+0	2.575E-6
3	3.000E+0	3.183E-6	0.000E+0	3.183E+0	0.000E+0
4	4.000E+0	0.000E+0	2.575E+0	0.000E+0	2.575E-6
5	5.000E+0	9.836E-7	0.000E+0	9.836E-1	0.000E+0

Рисунок 2.15 — Результати розрахунків фільтрів Батерворта в MicroCap 12

2.6 Підсилювач

Через те, що нам необхідно отримати на виході двополярний сигнал, а це означає графік синусоїди повинен бути періодичним, спочатку в позитивний бік зростає і зменшується, потім спадає в негативну область і т.д. Але сигнал, що генерується, генератора частоти природно знаходиться тільки в одній чверті, тому що живлення даного синтезатора однополярне, в результаті чого синусоїда генерується тільки в позитивній області графіка. Нам потрібно середину тобто 0,25В зрушити на рівень 0В, щоб 0В синуса припадало на -0,25В, а 0,5 на рівень — 0,25В, тим самим синусоїда зрушить на необхідну нам половинку. Цю операцію виконує схема, яку показано на рисунку 2.16.

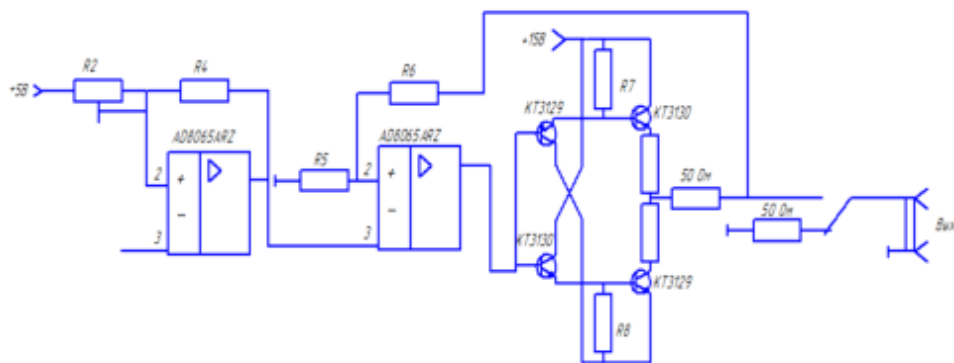


Рисунок 2.16 — Вихідний каскад

Далі вже зрушений в потрібний напрямок сигнал, тобто зміщений по осі сигнал подається на вихідний каскад, який зібраний на операційному підсилювачі AD80654RZ і чотирьох транзисторів типу KT3130 і KT3129. Даний вихідний каскад з глибоким негативним зв'язком здійснює збільшення потужності сигналу і його

збільшення амплітуди, в результаті чого розмах сигналу на виході може досягати до 10-13В, залежно від того який опір підберемо на резисторі в ланцюзі зворотного зв'язку.

В основі даної схеми знаходиться двотактний підсилювач (якщо реалізувати два підсилювачі, яких можна змусити їх посилювати позитивну та негативну напівхвилі синусоїди окремо, а потім з'єднати ці напівхвилі разом, то вийде підсилювач, що працює майже без спотворень). Подібний підсилювач отримав назву двотактного підсилювача.

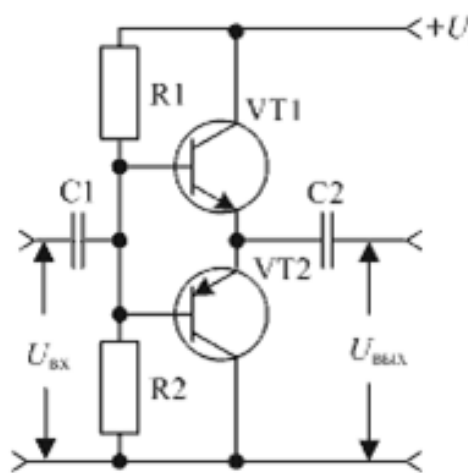


Рисунок 2.17 — Схема двотактного каскаду

Застосування двох транзисторів дозволяє їм допомагати одне одному. У наведеному рисунку позитивна напівхвиля напруги синусоїдальної відкриває транзистор VT1 і закриває VT2. Негативна напівхвиля — замикає транзистор VT1 і відкриває VT2. Таким чином кожен з транзисторів посилює тільки половину вхідної напруги, однак на виході, на опорі навантаження (у звукових підсилювачах на динаміці) ці половинки підсумовуються і форма вхідної напруги відновлюється.

3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Логіка роботи системи

Вдосконалення в системі, що проектується, можна проілюструвати на роботі генератора сигналів довільної форми (Arbitrary Waveform Generator) з комплексу віртуальних приладів NI ELVIS II. Для початку роботи з приладом, користувачем вказується файл, в якому міститься набір значень сигналу, що генерується (Waveform Name), і частота дискретизації (Update Rate), тобто частота, з якою відбувається зчитування цих значень з файлу (рисунок 3.1).

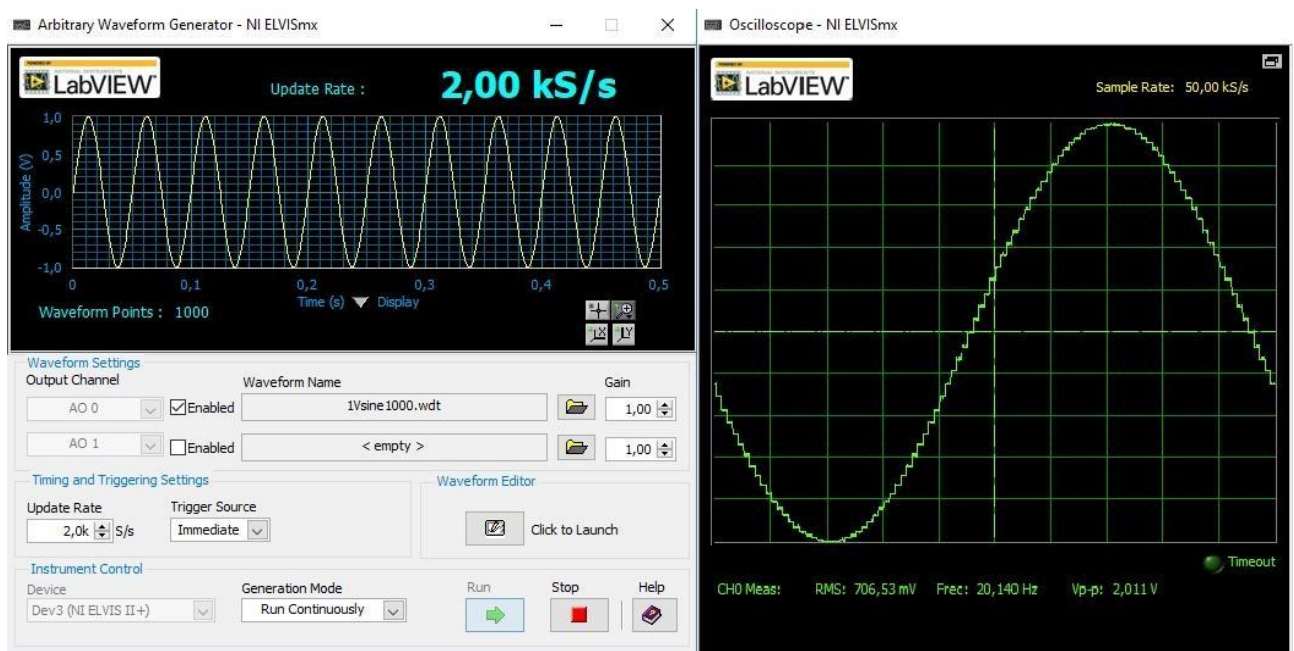


Рисунок 3.1 — Генерація гармонійного сигналу частотою 20 Гц

Збільшуючи або зменшуючи частоту дискретизації, можна відповідно збільшувати або зменшувати частоту вихідного сигналу (рисунок 3.2 та 3.3).

Отриманий сигнал має «ступінчастий» вигляд і, видно, що вимагає згладжування через ФНЧ, який взято з комплексу віртуальних приладів NI ELVIS II, а його розрахунок, проведено за формулою:

$$\omega_{\text{гп}} = \frac{1}{C \times R},$$

де $\omega_{\text{гр}}$ — гранична частота фільтра, C — ємність конденсатора, R — опір резистора, можна підібрати відповідні елементи для ФНЧ (рисунок 3.4).

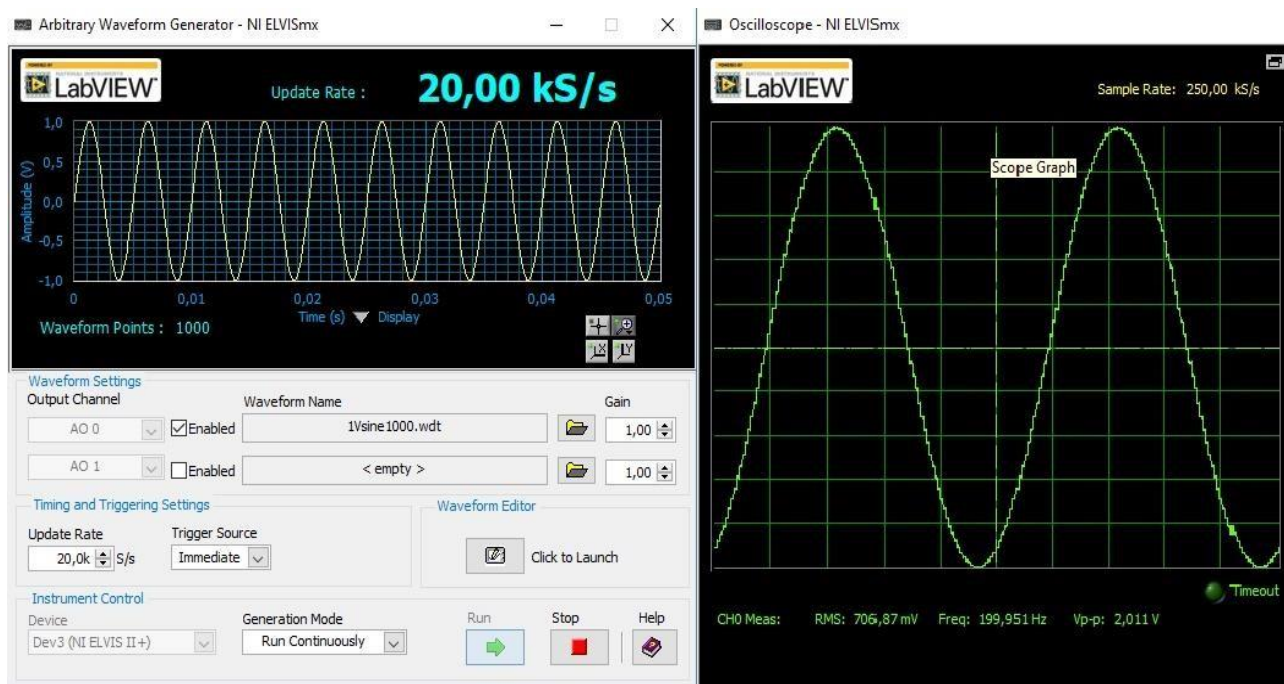


Рисунок 3.2 — Генерація гармонійного сигналу частотою 200 Гц

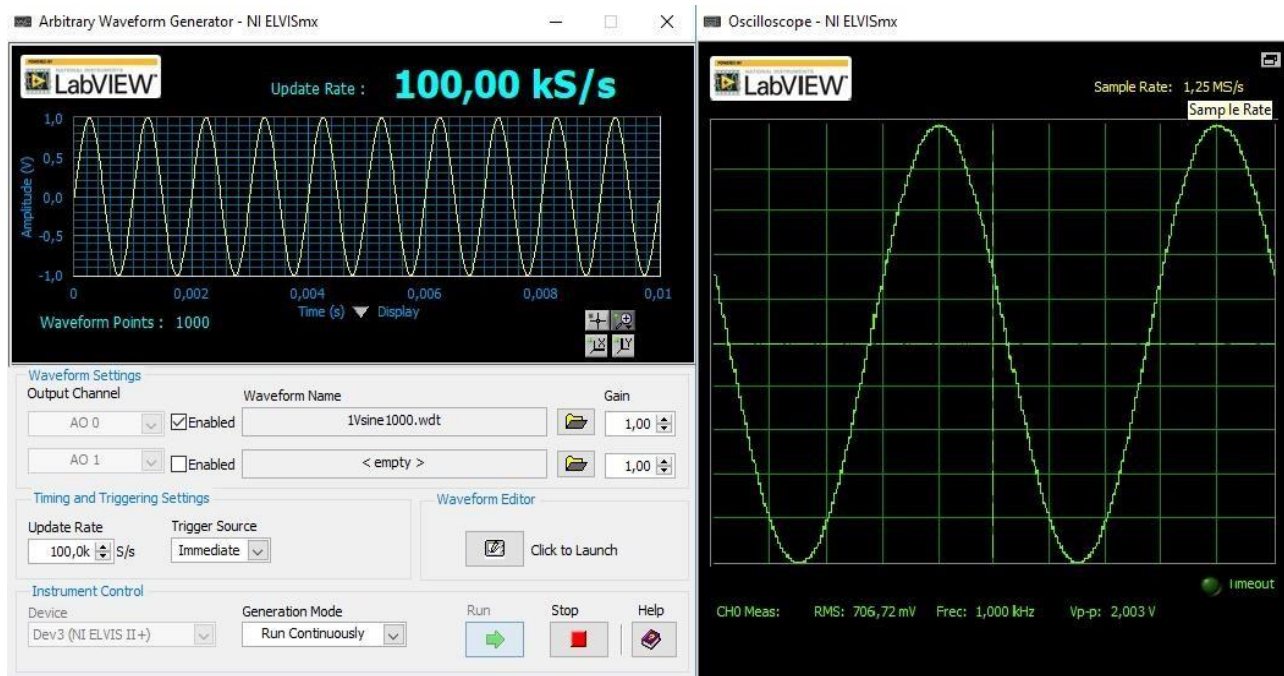


Рисунок 3.3 — Генерація гармонічного сигналу частотою 1 кГц

Амплітудно-частотна характеристика (АЧХ) фільтра наведена на рисунку 3.5.

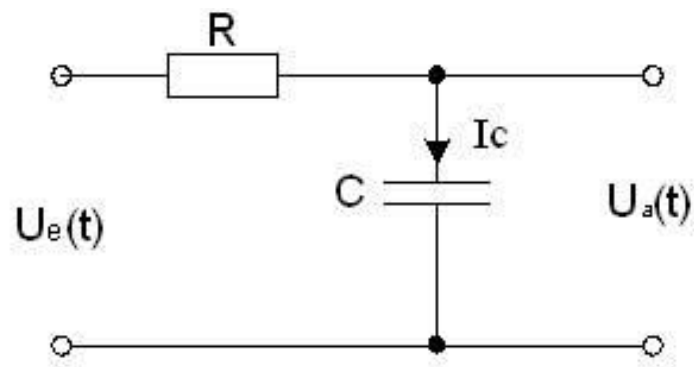


Рисунок 3.4 — Найпростіша схема ФНЧ до виходу генератора сигналів довільної форми

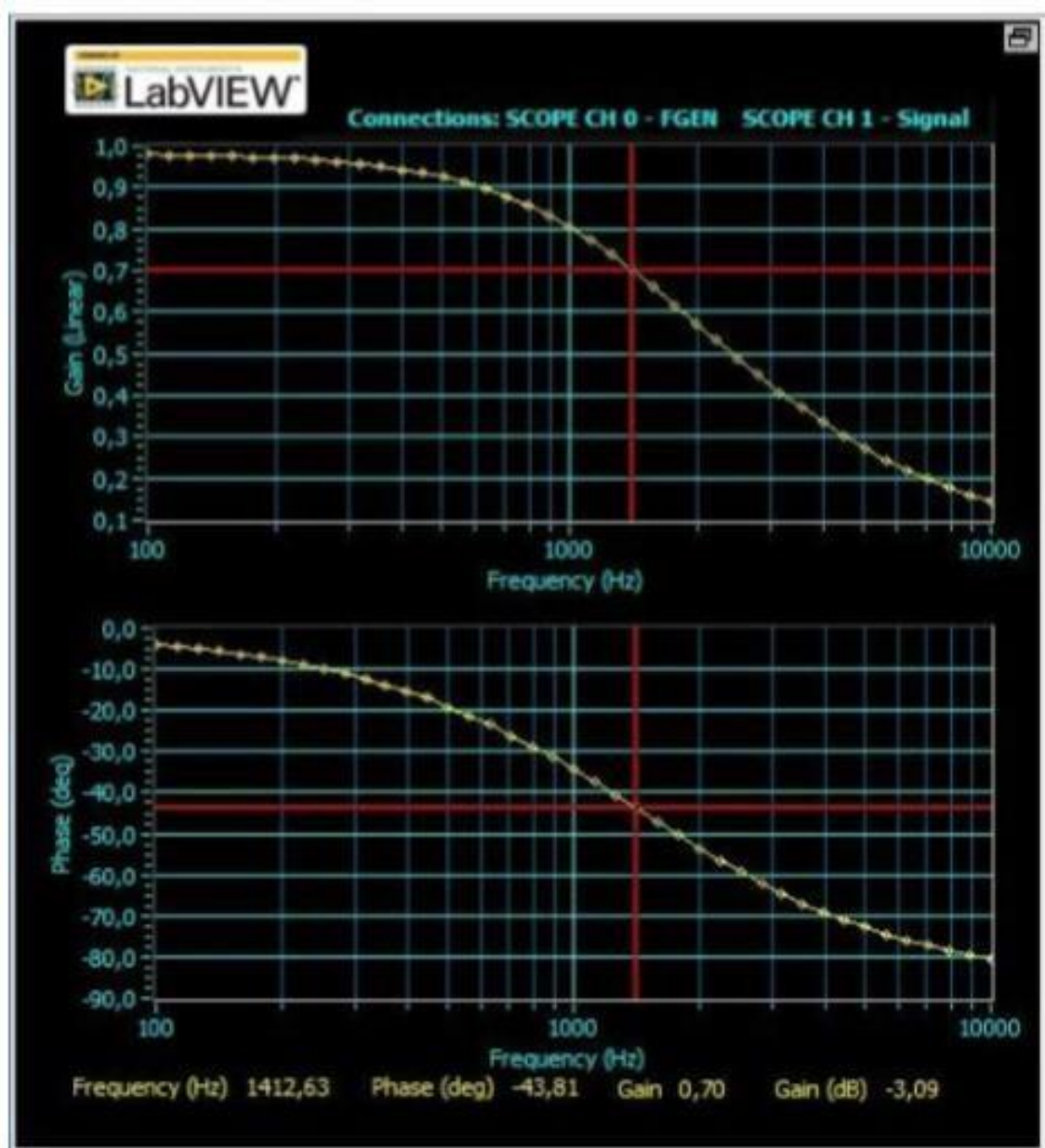


Рисунок 3.5 — Амплітудно-частотна характеристика ФНЧ

Вибираємо частоту сигналу генератора так, щоб вона потрапляла у смугу пропускання фільтра, наприклад, 700 Гц.

На рисунку 3.6 представлений вихідний сигнал генератора сигналів довільної форми до (сигнал 1) і після (сигнал 2) фільтрації. Результат згладжування сигналу фільтром помітний навіть візуально.

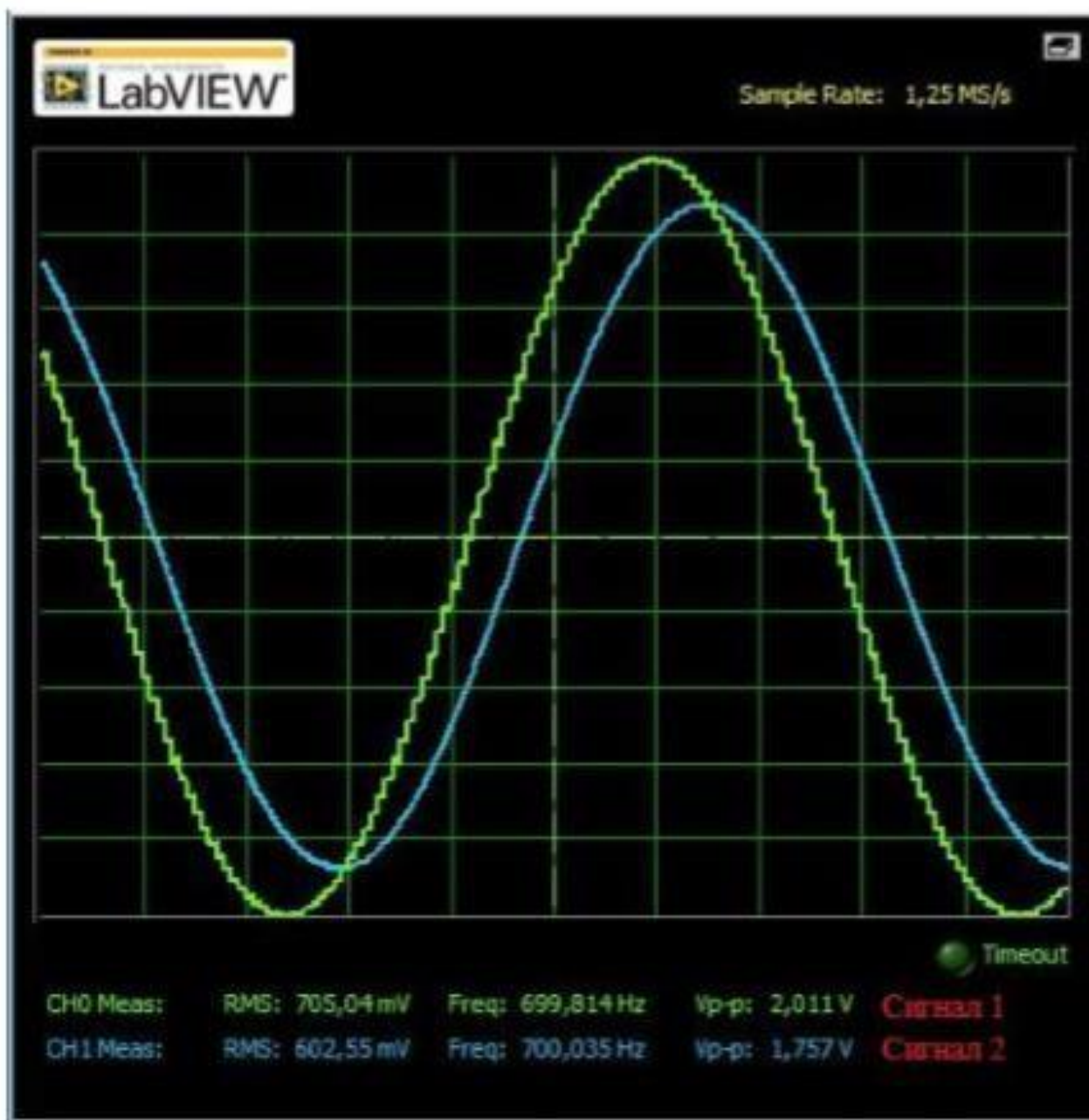


Рисунок 3.6 — Сигнал до та після фільтрації

Аналізатором спектра можна поспостерігати суттєву зміну коефіцієнта гармонійних спотворень (КГС, англ. — Total Harmonic Distorsions — THD) сигналу

після проходження через фільтр (рисунок 3.7). Як очевидно з рисунків без фільтра цей коефіцієнт дорівнює 1,07% (спектр 1), і з фільтром - 0,07% (спектр 2).

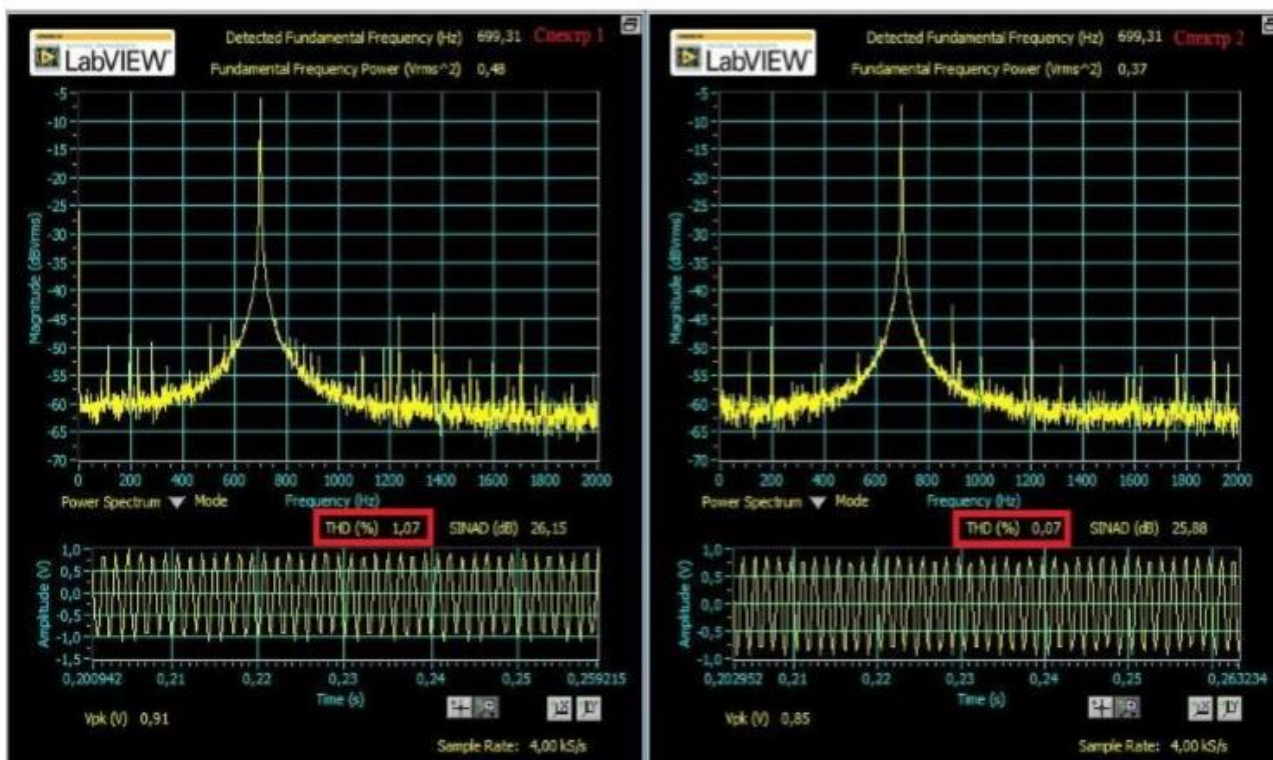


Рисунок 3.7 — Спектр сигналу до (ліворуч) та після (праворуч) фільтрації

3.2 Використання інструментів розробки

Для розробки програмного забезпечення під 32-розрядні процесори ARM (у тому числі і для плат) рекомендуються програмні продукти такі як: Atollic TrueSTUDIO, IAR Embedded Workbench, Altium TASKING VX-Toolset, Keil μ Vision з інструментами MDK-ARM.

Усі ці програмні середовища є комерційними. Безкоштовне їх використання можливе лише з обмеженнями за розміром бінарного коду програми або термінами використання.

3.1.2 Середовище розробки CoCoX CoIDE

Середовище розробки CoCoX CoIDE 1.7 (рисунок 3.8) — безкоштовний інструмент, який орієнтований на розробку для 32-розрядних ARM процесорів різних виробників: Atmel, Energy Micro, Freescale, Holtek, NXP, Nuvoton, TI. Це середовище вимагає мінімальних налаштувань для початку програмування та налагодження плати STM32F4 Discovery. Для початку розробки, крім самої CoIDE, знадобиться встановлення засобів для побудови проекту GNU Toolchain for ARM Embedded Processors [5].

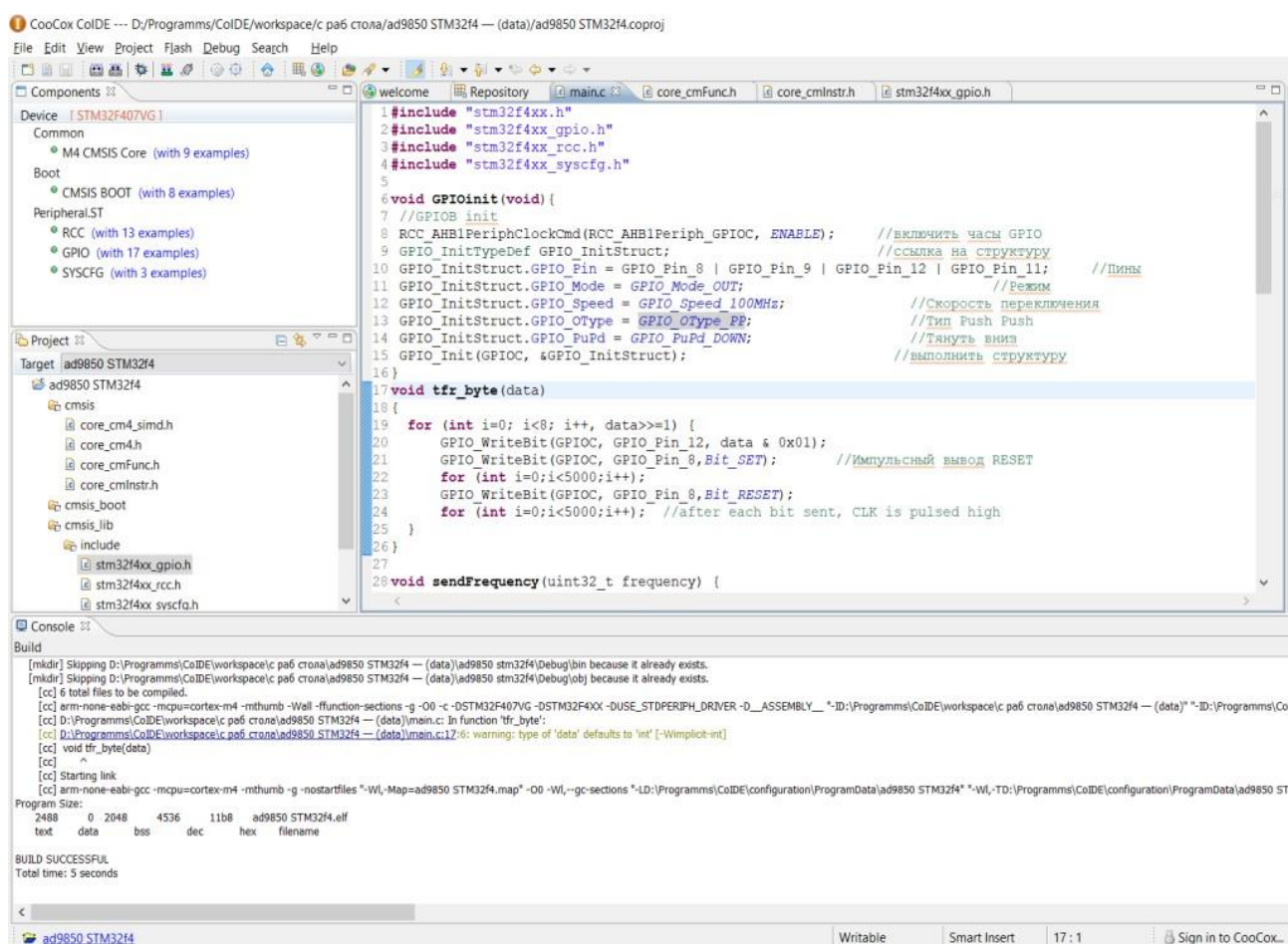


Рисунок 3.8 — Середовище розробки CoCoX CoIDE 1.7

Розробка з використанням CoIDE пов'язана з концепцією: всі доступні компоненти та бібліотеки встановлюються за допомогою майстра, виходячи з того, для якого процесора створено проект. Для завантаження бібліотек потрібне підключення до мережі Internet.

Також CooCox CoIDE вміє програмувати і є можливість налагодження використовуючи ST-Link або ST-Link/V2 . А ST-Link/V2 вже є на платі STM32F4 Discovery .

Для того, щоб скачати інсталяційний файл, потрібно буде зареєструватися. Це не складно.

Для закачування пропонується два варіанти

- Download через CoCenter (Recommend);
- Download the latest CoIDE directly.

Це через CoCenter (рекомендується) чи просто CoIDE.

Ще CooCox IDE не має компілятора. Про це попереджає напис.

Note: CoIDE не має integrated GCC compiler. Перед використанням CoIDE, ви повинні встановити GCC Toolchain вперше.

Тому перед тим, як використовувати CoIDE (рисунок 3.9), необхідно встановити компілятор. Встановлено рекомендований CooCox компілятор GCC ARM Embedded.



Рисунок 3.9 — Вікно встановлення CoIDE версії 1.7.5

Можна встановити інший компілятор не ARM GCC, а CodeSourcery GCC. Можна використовувати обидва компілятори, але слід враховувати, що ці компілятори по-різному обробляють код і проект створений в одному компіляторі не працюватиме з іншим. Потрібні будуть додаткові дії для перенесення проекту з одного компілятора до іншого. Шлях установки залишаємо за замовчуванням (рисунок 3.10). Після закачування файлу інсталяції і запуску весь процес стандартний.



Рисунок 3.10 — Вибір папки встановлення

Програми, створені за ліцензією GNU, досить часто примхливі до імен файлів або папок. Вони просто можуть не розуміти українську мову або не сприймати пробіл в імені файлу або шляху до файлу. Тому краще встановлювати такі програми туди, куди вони попросять.

Також вимагається згода на встановлення менеджера CoCenter (рисунок 3.11), який надалі оповіщатиме про всі оновлення та дає можливість встановлення додаткових утиліт розробника. Серед них варто відзначити власну операційну систему для роботи з мікроконтролерами CooCox CoOS, софт для програмування

Flash-пам'яті CooCox CoFlash, а також інструмент, що спрощує конфігурування портів контролерів CooCox CoSmart. Встановлювати CoIDE рекомендується в каталог без українських букв, без пробілів, а найкраще в варіант, що пропонується за замовчуванням.

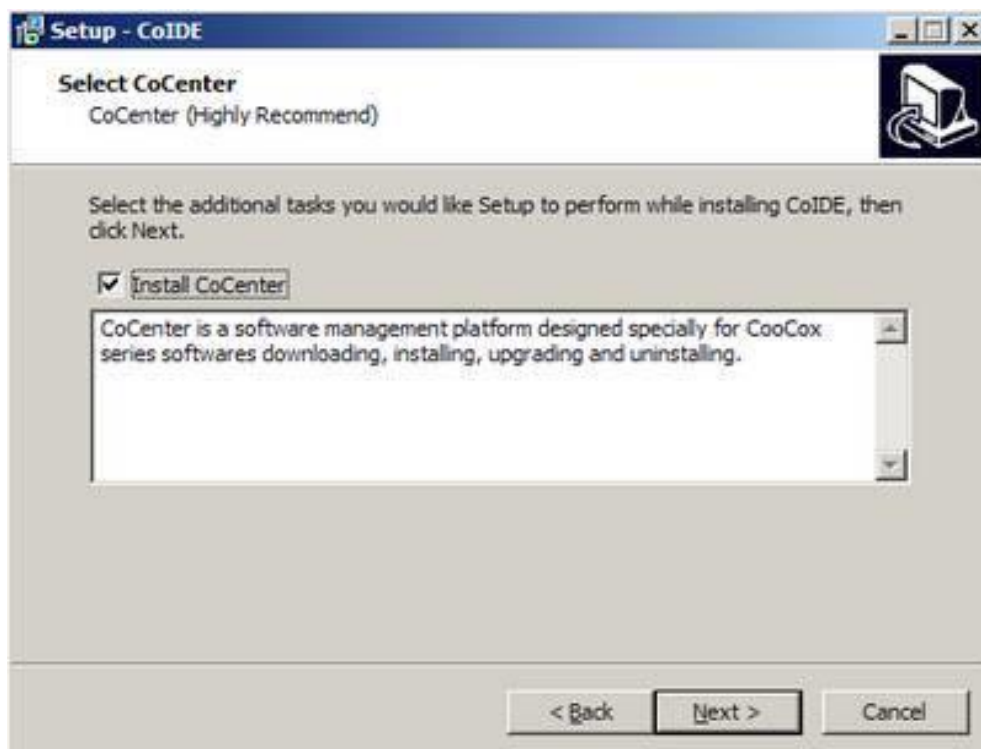


Рисунок 3.11 — Вікно встановлення CoCenter

Встановлювати CoIDE рекомендується в каталог без українських букв, без пробілів, а найкраще в варіант, що пропонується за замовчуванням (3.12).

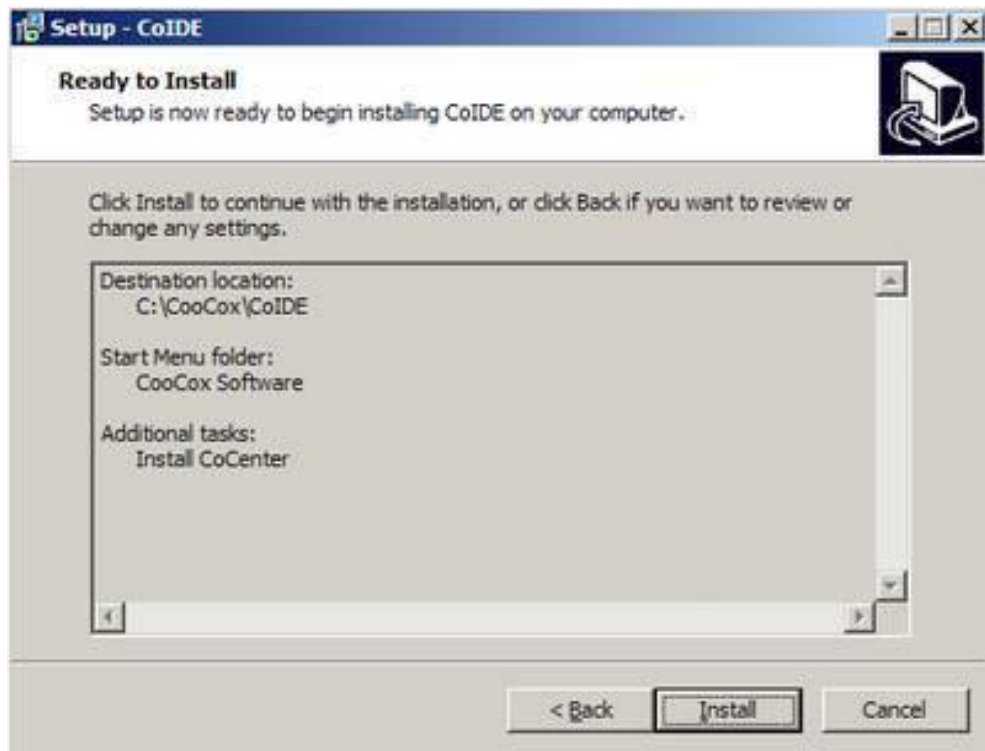


Рисунок 3.12 — Процес встановлення CoIDE версії 1.7.5

Підсумкова інформація як і куди встановлюється. Тиснемо "Install" і чекаємо закінчення установки.



Рисунок 3.13 — Завершення встановлення CoIDE

3.2.2 STM32 ST-LINK Utility

Незважаючи на те, що середовища розробки інтегрують у собі функціональність з прошивки мікроконтролерів, корисно буде знати, що існують і окремі програми, які спеціально призначені для роботи з пам'яттю (програмування, очищення, перевірка) пристроїв. Крім того, при роботі з режимами зниженого енергоспоживання, саме ці програми дозволяють виконати програмування незалежно від поточного режиму роботи, з чим можуть виникнути проблеми при виконанні даних дій із середовища розробки [6].

Програма STM32 ST-LINK Utility призначена для роботи з 32-розрядними контролерами через інтерфейс ST-LINK (у тому числі з його другою версією). Її інтерфейс (рисунок 3.14) організований максимально легко і зрозуміло.

Утиліта ST-LINK utility призначена для програмування мікроконтролерів STM8 та STM32 через програматор ST-LINK або ST-LINK/V2.

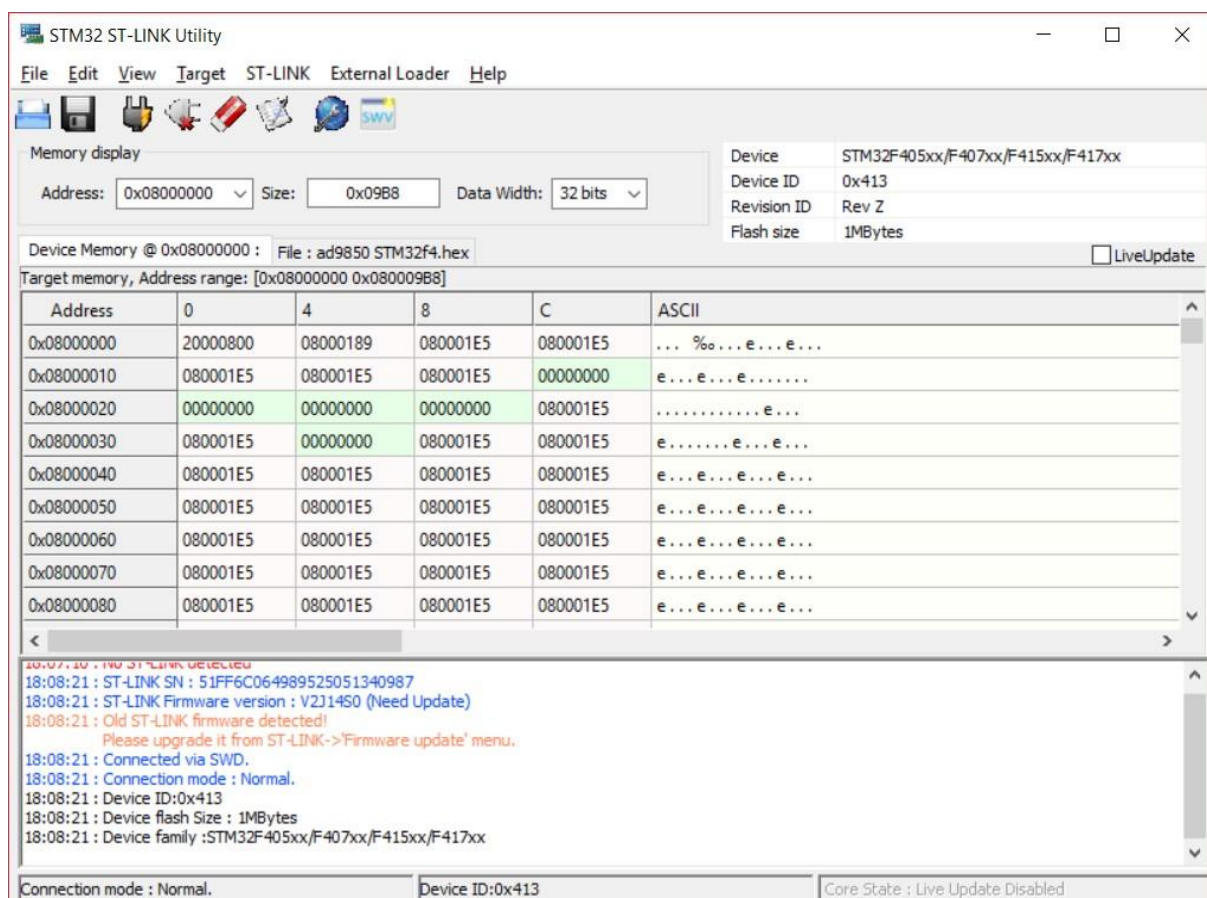


Рисунок 3.14 — Вікно програми STM32 ST-LINK Utility

Завантажити ST Link Utility можна із сайту компанії STMicroelectronics. Сайт великий і до того ж англійською, тому легше потрапити на сторінку для скачування, набравши у пошуку "STSW-LINK004" або "ST-LINK utility". В розділі "Get Software" та завантажуюмо файл. Це буде архів stsw-link004.zip, в якому знаходиться інсталяційний файл STM32 ST-LINK Utility_v3.2.1.exe. Версія може бути іншою.

Після запуску STM32 ST-LINK Utility_v3.2.1.exe розпочнеться встановлення утиліти на комп'ютер (рисунок 3.15). Буде встановлена сама утиліта ST-LINK utility та драйвер STLinkDriver.

Особливостей начебто ніяких. Тиснемо скрізь "Next".

Після встановлення підключаємо STM32F4DISCOVERY до комп'ютера через міні-USB та починається процес встановлення нового пристрою.

Під час інсталяції може виникнути повідомлення про несумісність обладнання з версією Windows.

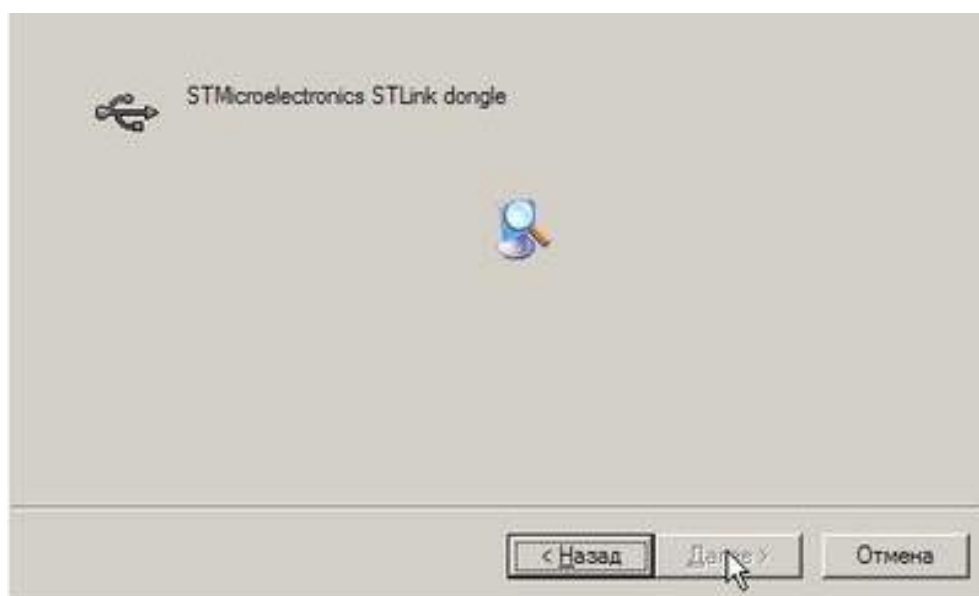


Рисунок 3.15 — Вікно встановлення утиліта ST-LINK utility

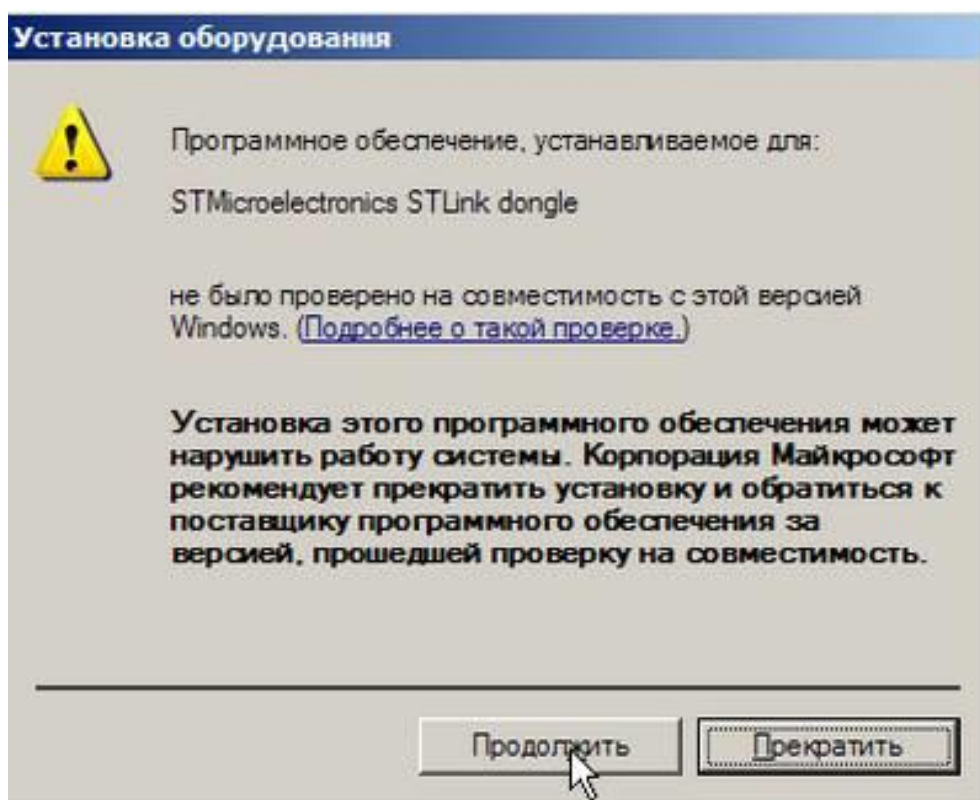


Рисунок 3.16 — Повідомлення про несумісність

Після встановлення вийде віконце про завершення. Щоб переконатися, можна переглянути в Диспетчері пристроїв, з'явився там ST-Link чи ні (рисунок 3.17). Після запуску ST-Link utility та натискання на кнопку з "вилкою" можна подивитися, що знаходиться в мікроконтролері налагоджувальної плати STM32F4Discovery.

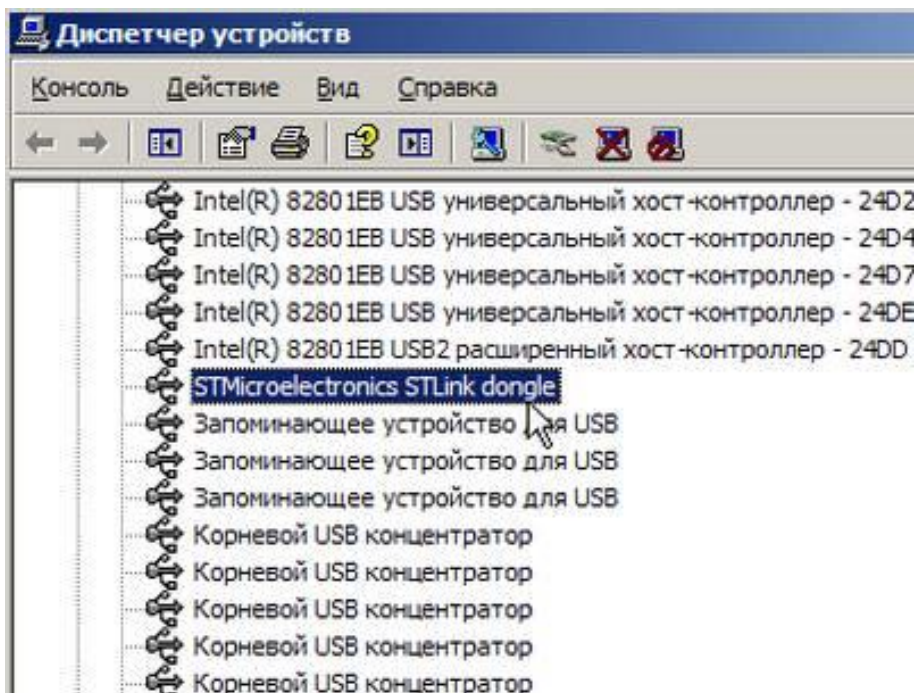


Рисунок 3.17 — Поява ST-LINK в диспетчері пристроїв

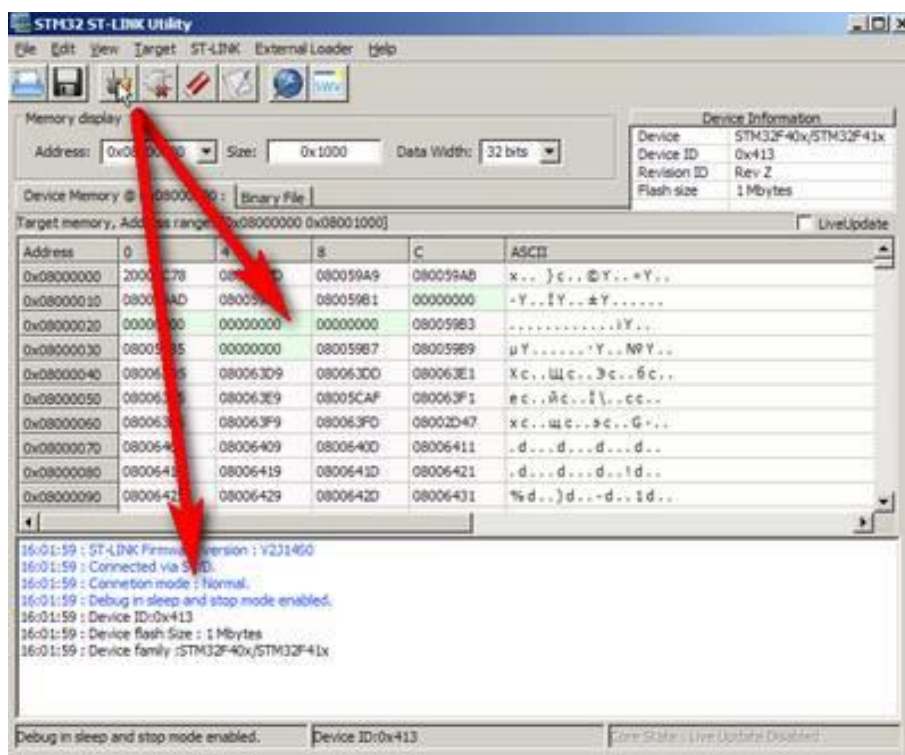


Рисунок 3.18 — Вікно запуску ST-Link utility

Утиліта STM32 ST-LINK utility для програматора / відладчика ST-Link/V2 встановлена і можна нею користуватися. Далі натискаємо на сусідню кнопку із сірою вилкою та червоним хрестом для відключення від STM32F4 Discovery та закриваємо програму.

4 ДОСЛІДЖЕННЯ СПРОЕКТОВАНОЇ СИСТЕМИ

4.1 Генерація сигналів із використанням STM32F4 Discovery

Щоб дослідити можливості генерації сигналів із використанням налагоджувальної плати STM32F4 Discovery [14, 15], необхідно завантажити у плату найпростіший код (рисунк 4.1, див. додаток Б), який дозволяє на виході отримати сигнал синусоїдальної форми заданому діапазоні частот.

```

1 #include "stdio.h"
2 #include "stdlib.h"
3
4 #include "stm32f4xx.h"
5 #include "stm32f4xx_gpio.h"
6 #include "stm32f4xx_rcc.h"
7 #include "stm32f4xx_tim.h"
8
9 const uint16_t sin[32] = {
10 2047, 2447, 2831, 3185, 3498, 3750, 3939, 4056, 4095, 4056,
11 3939, 3750, 3495, 3185, 2831, 2447, 2047, 1647, 1263, 909,
12 599, 344, 155, 38, 0, 38, 155, 344, 599, 909, 1263, 1647};
13 unsigned char i=0;
14
15 int main(void) {
16  /* Включаємо порт А */
17  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
18  /* Включаємо ЦАП */
19  RCC_APB1PeriphClockCmd(RCC_APB1Periph_DAC, ENABLE);
20  /* Включаємо таймер 6 */
21  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6, ENABLE);
22
23  /* Настроюємо ногу ЦАПа */
24  GPIO_InitTypeDef GPIO_InitStructure;
25  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
26  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
27  GPIO_Init(GPIOA, &GPIO_InitStructure);
28
29  /* Настроюємо таймер так щоб он тикал почаще */
30  TIM6->PSC = 0;
31  TIM6->ARR = 500;
32  TIM6->DIER |= TIM_DIER_UIE; //разрешаем прерывание от таймера
33  TIM6->CR1 |= TIM_CR1_CEN; // Начать отсчёт!
34  NVIC_EnableIRQ(TIM6_DAC_IRQn); //Разрешение TIM6_DAC_IRQn прерывания

```

Рисунок 4.1 — Фрагмент програмного коду серед Coocox ColIDE

Робота коду дуже проста, підключаємо необхідні бібліотеки, створюємо масив, елементи якого відправлятимемо в ЦАП, щоб на виході отримати синус, включаємо сам ЦАП, після чого ініціалізуємо порт, на який подаватимемо вихідний сигнал. Частоту задаємо за допомогою налаштування таймера.

Щоб отримати сигнал трикутної форми, потрібно буде лише змінити елементи масиву. Для генерації сигналів прямокутної форми якнайкраще підходить

включення таймера в режимі ШІМ для одного з вихідних каналів. Необхідно лише розрахувати значення для налаштування самого таймера при зазначеній тактовій частоті пристрою та включити тактування та необхідний режим роботи для інших пристроїв у складі контролера.

На рисунку 4.2 показано роботу плати з вихідною частотою 1кГц.

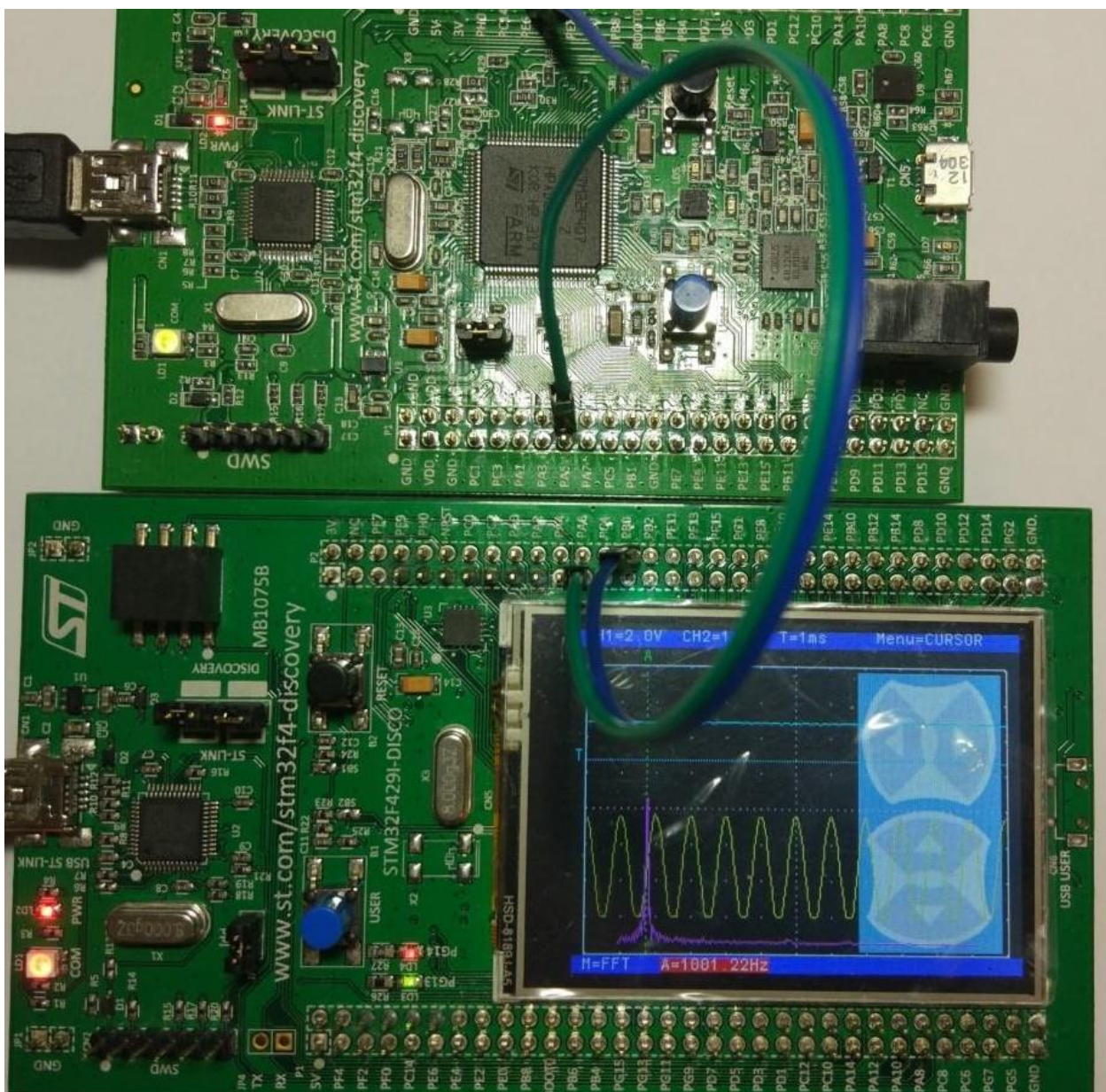


Рисунок 4.2 — Синусоїдальний сигнал частотою 1 кГц

Зазначимо, що для перевірки роботи плати в ролі осцилографа був використаний МК STM32F429I Disco [16]. Основні характеристики даного осцилографа наведено у таблиці 4.1.

Таблиця 4.1 — Основні технічні характеристики осцилографа

Тип	Цифровий
Кількість аналогових вхідних каналів	2 (CH1 - PA5, CH2 - PA7)
Кількість цифрових вихідних каналів	2 (UART-TX - PA9, тестовий сигнал 500Гц -PB2)
Смуга пропускання, кГц	500
Максимальна частота дискретизації на канал	1МГц

Дослідження показує можливість генерації сигналів з використанням налагоджувальної плати. Максимально допустима частота вихідного сигналу 10кГц.

4.2 Тестування цифрового синтезатора AD9850

Перед початком розробки та виготовлення макета генератора на платформі AD9850-STM32 [17] був реалізований макет генератора на платформі ARDUINO-AD9850.

Для здійснення можливості зміни частоти сигналу, що генерується, а також візуального контролю за значенням частоти і кроком перебудови до схеми був доданий LCD Keypad Shield — плата з дворядковим екраном та кнопками (рисунок 4.3).

Програмний код, що завантажується в плату Arduino, дозволяє використовувати чотири кнопки для керування генератором: вгору і вниз — для грубого налаштування, праворуч і ліворуч — для точного. Крок зміни змінюється в залежності від поточної частоти. Підключення синтезатора AD9850 до плати Arduino UNO здійснюється за послідовним інтерфейсом.

Лістинг коду можна знайти у додатку В.

Коротке пояснення роботи коду. Спочатку програма здійснює підключення контактів синхронізації (CLK), оновлення частоти (FQ), завантаження даних (DATA) та скидання (RST) синтезатора до цифрових каналів плати Arduino. Функція `tfr_byte` забезпечує бітову передачу байта інформації до синтезатора по

лінії DATA. Функція `sendFrequency` здійснює розрахунок необхідної частоти з опорної. Далі проводиться конфігурація контактів плати Arduino і виставляється значення частоти, що генерується.

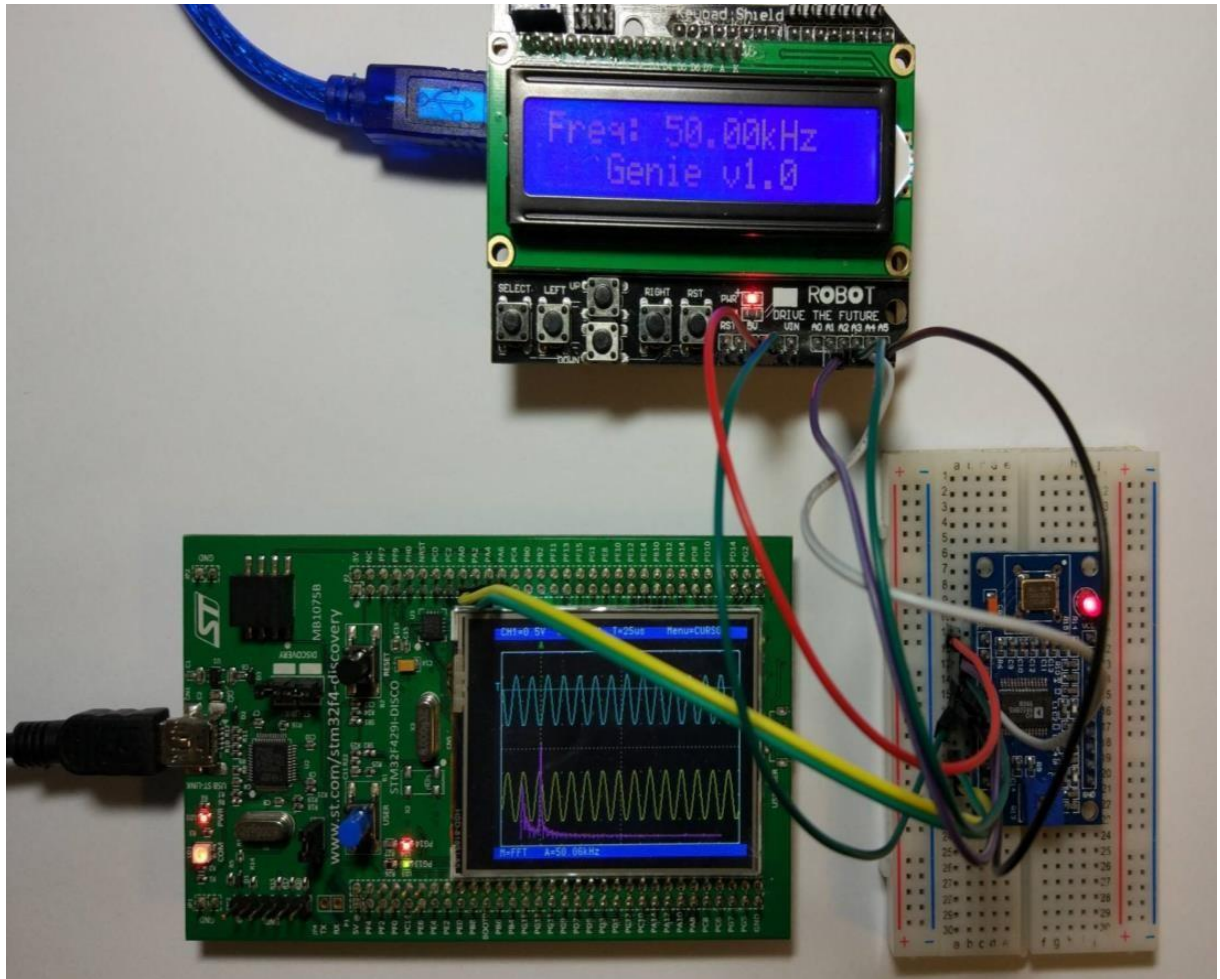


Рисунок 4.3 — Підключення AD9850 до LCD Keypad Shield

Функціонально він складається з наступних логічних блоків: підключення бібліотек і контактів, що використовуються, функція зчитування натискання кнопок, функція побитної відправки байта, перетворення і відправлення значення частоти, відправка сигналів для роботи генератора, виведення поточного значення частоти на екран.

Підключення синтезатора AD9850 до плати LCD Keypad Shield здійснюється за послідовним інтерфейсом.


```

#include < LiquidCrystal .h> // Підключення бібліотек
LiquidCrystal lcd (8, 9, 4, 5, 6, 7); // Ініціалізація LCD із зазначенням пінів
float freq = 10000; // Оголошення змінних - Частота
float bigStep = 1000; // крок зміни частоти при натисканні вгору/вниз
float littleStep = 10; // Крок зміни частоти при натисканні праворуч/ліворуч
int lcd_key = 0;
int adc_key_in = 0;
#define btnRIGHT 0 // Створення директив для кнопок
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5
#define W_CLK 15 // Пін A1 Arduino - підключений до CLK
#define FQ_UD 16 // Пін A2 Arduino - підключений до FQ (FU)
#define DATA 17 // Пін A3 Arduino - підключений до DATA
#define RESET 18 // Пін A4 Arduino - підключений до RST
#define pulsehigh(pin) { digitalWrite (pin, HIGH ); digitalWrite (pin, LOW ); }
//-----
int read_LCD_buttons(){ // Функція зчитування натискання кнопок
  adc_key_in = analogRead (0);
  if (adc_key_in > 1000) return btnNONE;
  if (adc_key_in < 50) return btnRIGHT;
  if (adc_key_in < 150) return btnUP;
  if (adc_key_in < 315) return btnDOWN;
  if (adc_key_in < 600) return btnLEFT;
  if (adc_key_in < 850) return btnSELECT;
  return btnNONE;}
//-----
void tfr_byte( byte data){ // Функція побитної відправки байта
  for ( int i=0; i<8; i++, data>>=1) { // даних у модуль генератора
    digitalWrite (DATA, data & 0x01);
    pulsehigh(W_CLK);} // подача імпульсу на CLK після кожного біта
//-----
void sendFrequency( double frequency) { // Перетворення та відправка
  int32_t freq = frequency * 4294967295/125000000; // значення частоти
  for ( int b=0; b<4; b++, freq>>=8) {tfr_byte(freq & 0xFF);}
  tfr_byte(0x000); // Відправлення завершального контрольного байта
  pulsehigh(FQ_UD);} // Оновлення частоти генератора

```

Рисунок 4.4 — Фрагмент програмного коду серед IDE

Даний спосіб надає можливості розуміння перебудови частоти у будь-який зручний момент часу, а також спостереження за її значенням на екрані. Однак макет генератора реалізовано на платі STM32. Завдяки великій кількості каналів введення-виведення та безлічі периферійних пристроїв існує можливість реалізації проектів вищого рівня складності.

4.3 Підключення цифрового синтезатора AD9850 до STM32F4 Discovery

У МК STM32F4 Discovery було завантажено код (рисунок 4.5), для генерації сигналу, який наведено в додатку Г.

Програма здійснює підключення бібліотек, необхідні роботи. Далі йде ініціалізація портів введення/виведення, які забезпечуватимуть побітову передачу байтів інформації до синтезатора за допомогою восьмибітного порту для паралельного введення даних, а саме лініями D0-D7 модуль передаються п'ять керуючих слів по вісім біт. Кожне слово супроводжується сигналом синхронізації W_CLK. Оновлення даних відбувається з настанням імпульсу на вхід FQ_UD. Функція AD9850_ParallelSend здійснює розрахунок необхідної частоти з опорної, після чого виставляється значення частоти, що генерується.

Схема підключення показана рисунку 4.6.

```

1 #include "stm32f4xx.h"
2 #include "stm32f4xx_gpio.h"
3 #include "stm32f4xx_rcc.h"
4 #include "stm32f4xx_syscfg.h"
5
6 void GPIOinit(void) {
7     //GPIOB init
8     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
9     GPIO_InitTypeDef GPIO_InitStructure;
10    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10 | GPIO_Pin_11 | GPIO_Pin_12;
11    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
12    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
13    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
14    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
15    GPIO_Init(GPIOC, &GPIO_InitStructure);
16
17    //GPIOD init
18    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
19    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_All;
20    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
21    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
22    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
23    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
24    GPIO_Init(GPIOD, &GPIO_InitStructure);
25    GPIO_Write(GPIOD, 0x0000);
26 }
27
28 void AD9850_reset(void) {
29    GPIO_WriteBit(GPIOC, GPIO_Pin_10, Bit_RESET);
30    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
31    GPIO_WriteBit(GPIOC, GPIO_Pin_12, Bit_RESET);
32    for (int i=0; i<5000; i++);

```

Рисунок 4.5 — Фрагмент програмного коду серед CoCoSox CoIDE

Як було зазначено вище, підключення синтезатора AD9850 до плати здійснюється за паралельним інтерфейсом, що забезпечує швидшу передачу даних.

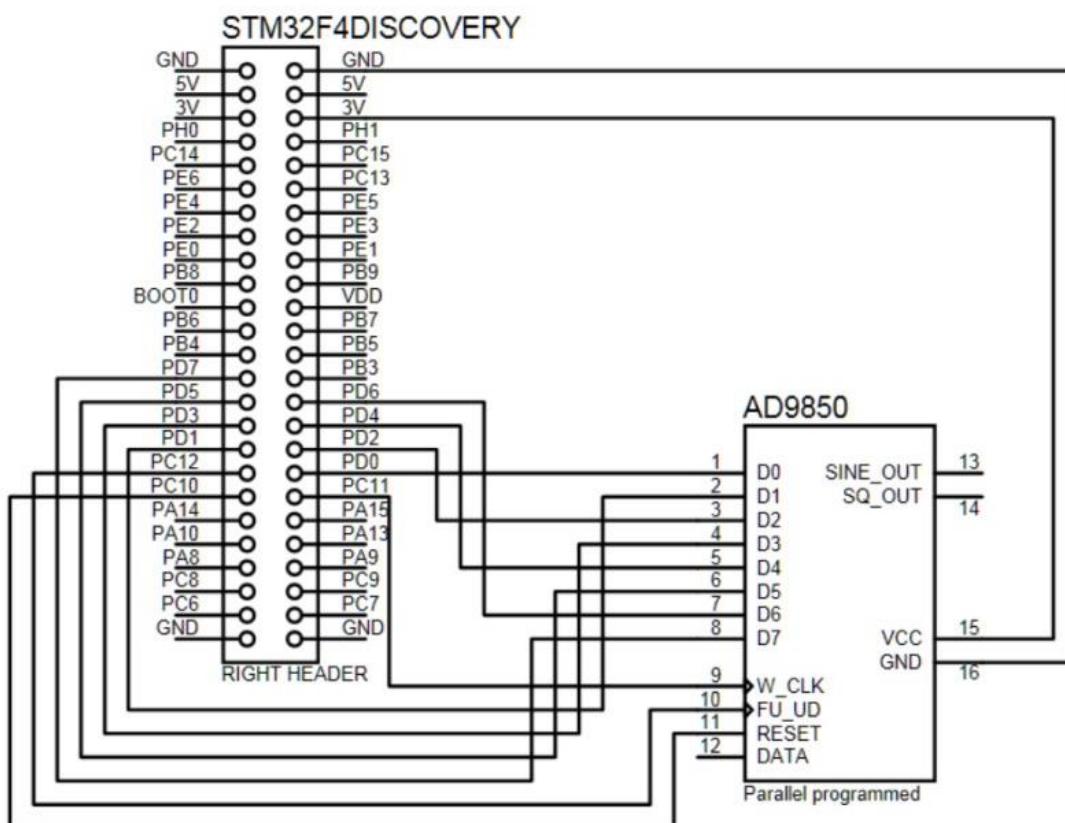


Рисунок 4.6 — Схема підключення AD9850 до STM32F4 Discovery

Макет реального пристрою представлений на рисунку 4.7.

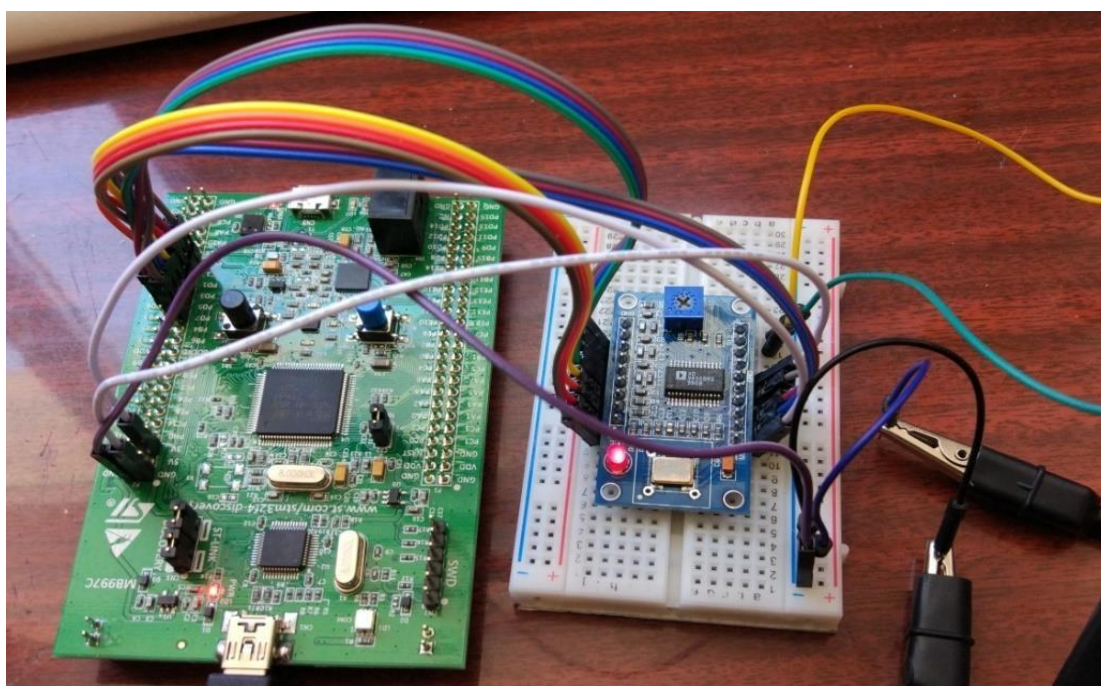


Рисунок 4.7 — Підключення AD9850 до STM32F4 Discovery

Вихідний сигнал прямокутної (рисунок 4.8) та синусоїдальної (рисунок 4.9) форми з контактів синтезатора спостерігався за допомогою осцилографа NI ELVIS II+.

Даний варіант складання простий, але має очевидні недоліки, такі як необхідність зміни та перезавантаження програмного коду для регулювання частоти та відсутність візуального контролю над значенням частоти вихідного сигналу.

Для здійснення можливості зміни частоти сигналу, що генерується, а також візуального контролю над значенням частоти до схеми планувалося додати матричну клавіатуру і рідкокристалічний дисплей (LCD).

Фрагмент програми підключення матричної клавіатури та дисплея WH1602 до STM32F4 Discovery показано на рисунку 4.10.

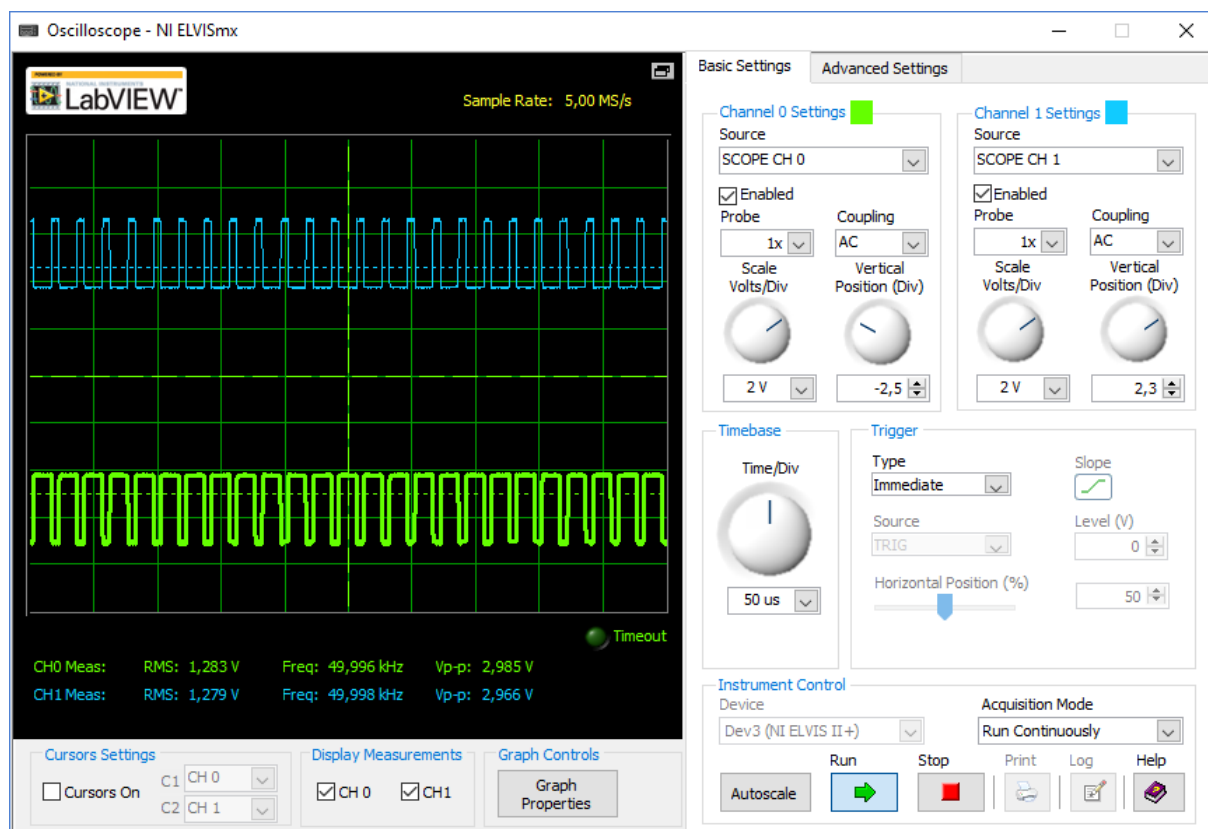


Рисунок 4.8 — Прямокутний сигнал частотою 50 кГц

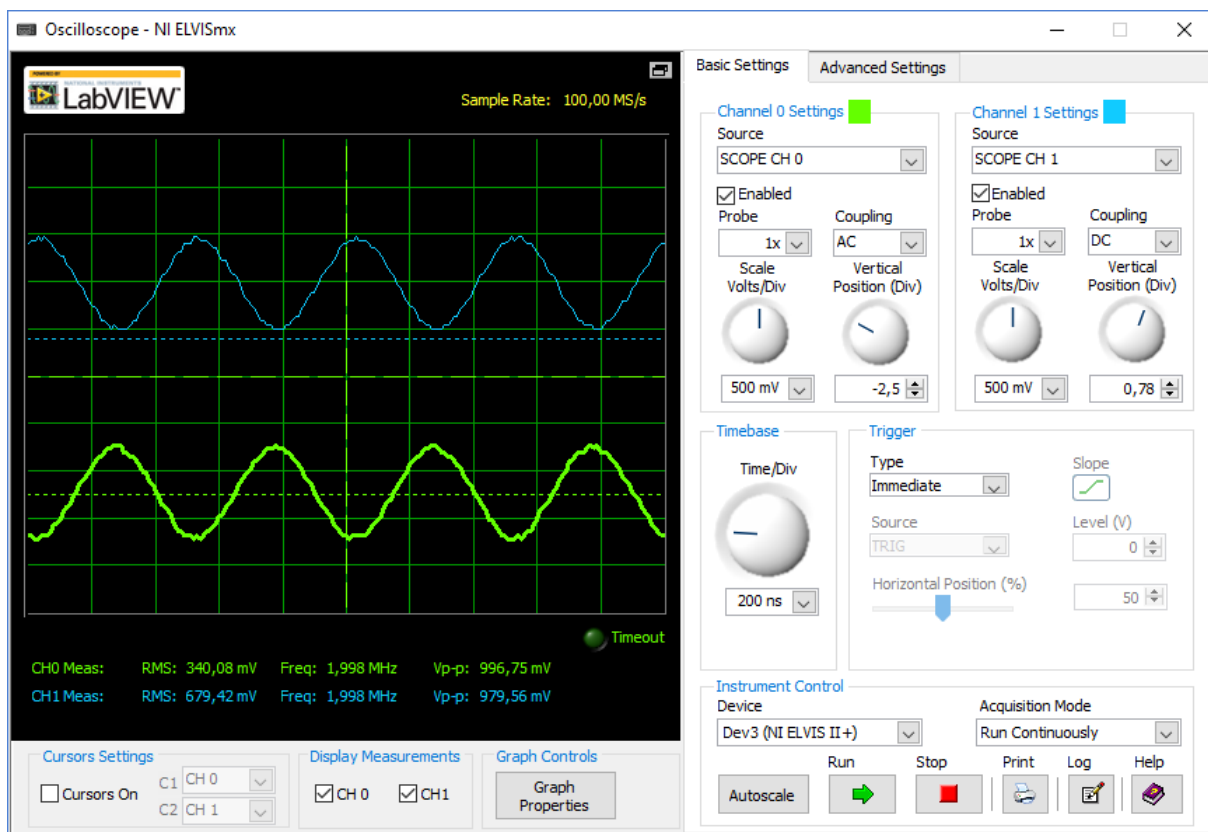


Рисунок 4.9 — Синусоїдальний сигнал частотою 2 МГц

CooCox CoIDE --- D:/Programms/CoIDE/workspace/diplom/mc385_Keyboard+WH1602d +/Library_1/Keyboard_v2.0finaly.coproj

File Edit View Project Flash Debug Search Help

Project: Keyboard_v2.0finaly

```

1 #include "stm32f4xx.h"
2 #include "stm32f4xx_gpio.h"
3 #include "stm32f4xx_rcc.h"
4 #include <stm32f4xx_tim.h>
5 #include <math.h>
6 #include <Keyboard.h>
7
8
9 typedef enum {FALSE = 0, TRUE = !FALSE} bool;
10 // Set RS port
11 #define MT_WH1602_RS_PORT (GPIOC)
12 // Set RS pin
13 #define MT_WH1602_RS_PIN (GPIO_Pin_2)
14
15 // Set RW port
16 #define MT_WH1602_RW_PORT (GPIOB)
17 // Set RW pin
18 #define MT_WH1602_RW_PIN (GPIO_Pin_10)
19
20 // Set E port
21 #define MT_WH1602_E_PORT (GPIOB)
22 // Set E pin
23 #define MT_WH1602_E_PIN (GPIO_Pin_14)
24
25 // Set DB7 port
26 #define MT_WH1602_DB7_PORT (GPIOE)
27 // Set DB7 pin
28 #define MT_WH1602_DB7_PIN (GPIO_Pin_3)
29
30 // Set DB6 port
31 #define MT_WH1602_DB6_PORT (GPIOE)
32 // Set DB6 pin
33 #define MT_WH1602_DB6_PIN (GPIO_Pin_3)

```

Рисунок 4.10 — Фрагмент програмного коду у середовищі CooCox CoIDE

Безпосередньо, саме підключення показано малюнку 4.11.

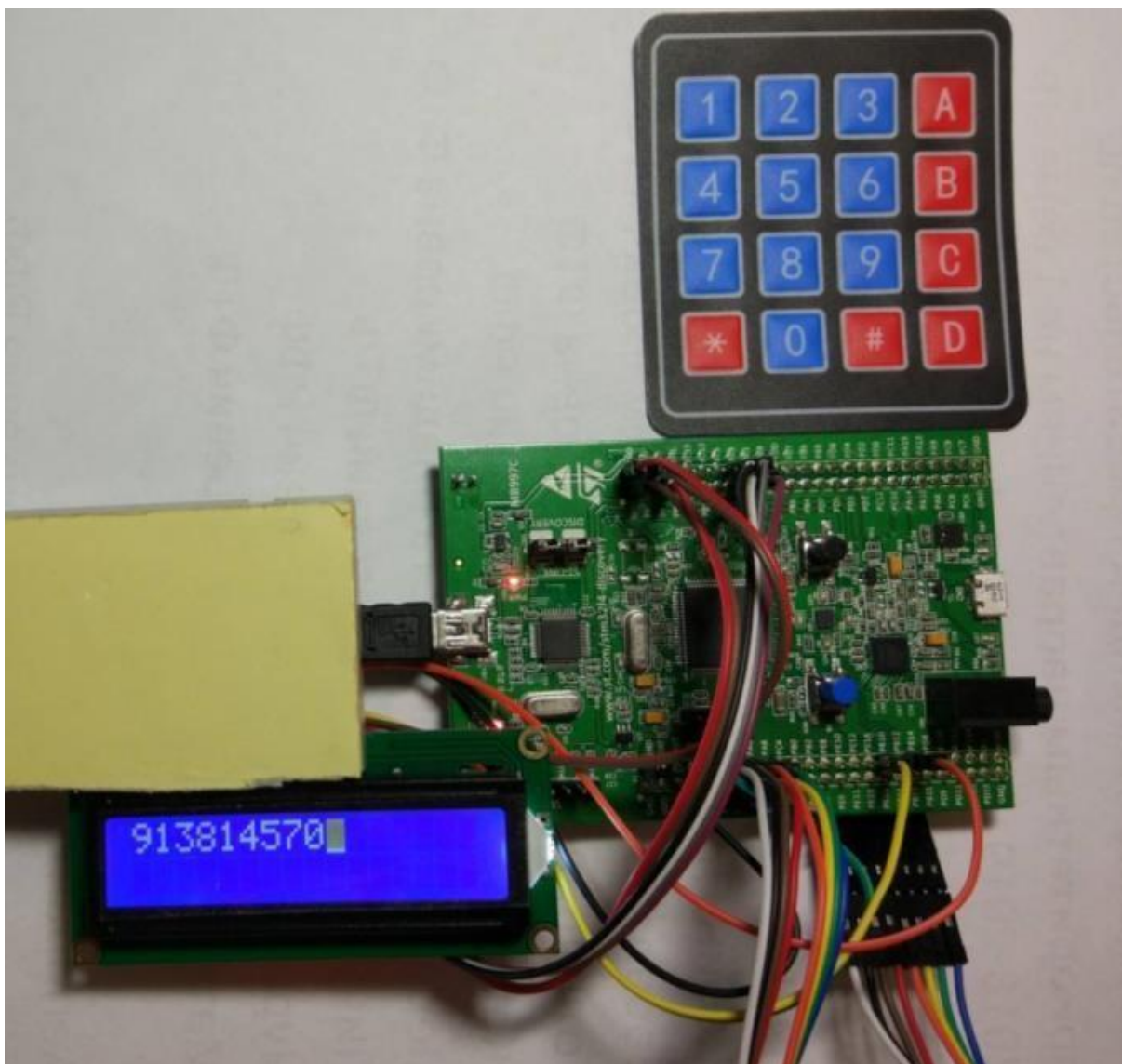


Рисунок 4.11 — Підключення матричної клавіатури та дисплея WH1602 до STM32F4 Discovery

Даний спосіб надав можливість перебудови частоти в будь-який зручний момент часу, а також виведення її значенням на дисплей. Однак цей макет був би дуже громіздким.

4.4 Функціональний DDS генератор на STM32F429I Disco

У налагоджувальну плату STM32F429I Disco [18, 19] був завантажений програмний код (Додаток Д), який дозволив реалізувати функціональний генератор

DDS (рисунк 4.12).

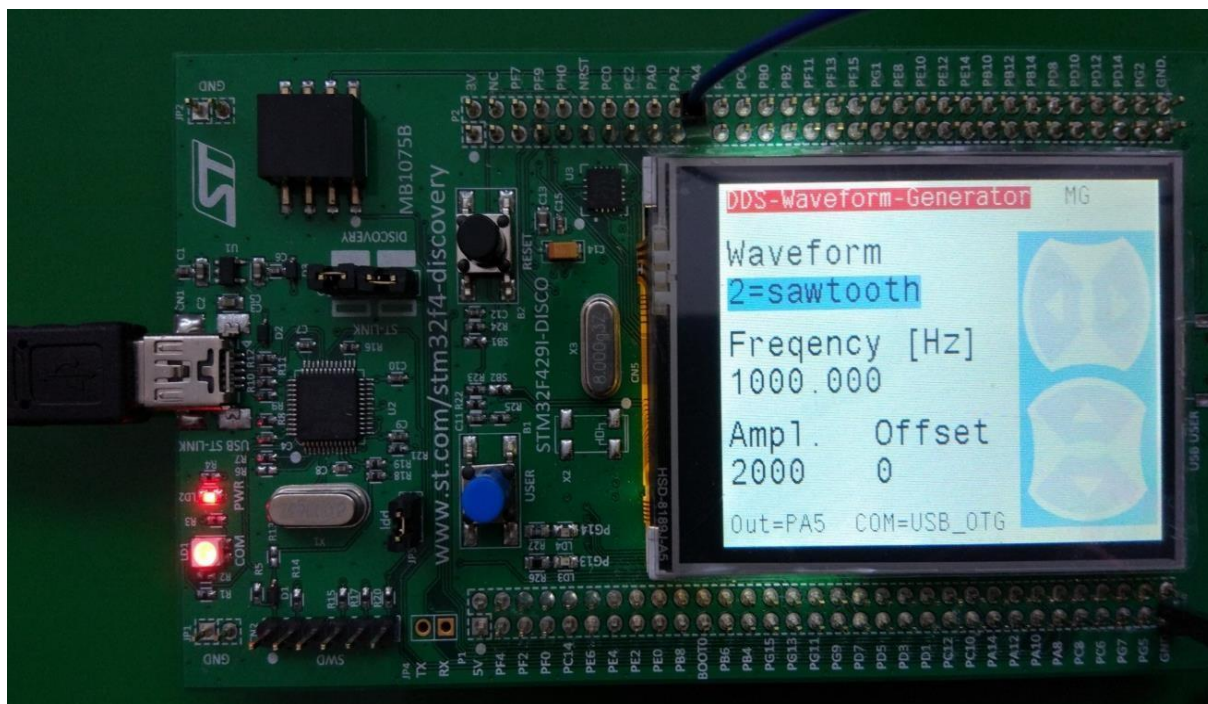


Рисунок 4.12 — Функціональний DDS генератор на STM32F429I Disco

Даний генератор дозволяє:

- генерувати сигнали синусоїдальної, квадратної, трикутної та пилкоподібної форми (рисунк 4.13);
- регулювати рівень амплітуди (0-3,3 В);
- регулювати зміщення (0-3,3 В);
- змінювати частоту діапазоні 100 – 50 кГц.

За допомогою сенсорного РК-дисплея можемо не лише візуально контролювати роботу генератора, але й можемо здійснювати зміну частоти сигналу, що генерується.

Даний DDS генератор дуже компактний і зручний у використанні, однак, у зв'язку з невеликою смугою частот і великим значенням коефіцієнта гармонічних спотворень, через відсутність вбудованого фільтра нижніх частот (ФНЧ), не можемо користуватися ним у повній мірі.

В результаті було вирішено розробити та виготовити макет генератора на платформі AD9850-STM32F429I Disco, який мав би ширший діапазон частот та

прийнятний КГС, а також можливість перебудови та відображення частоти.

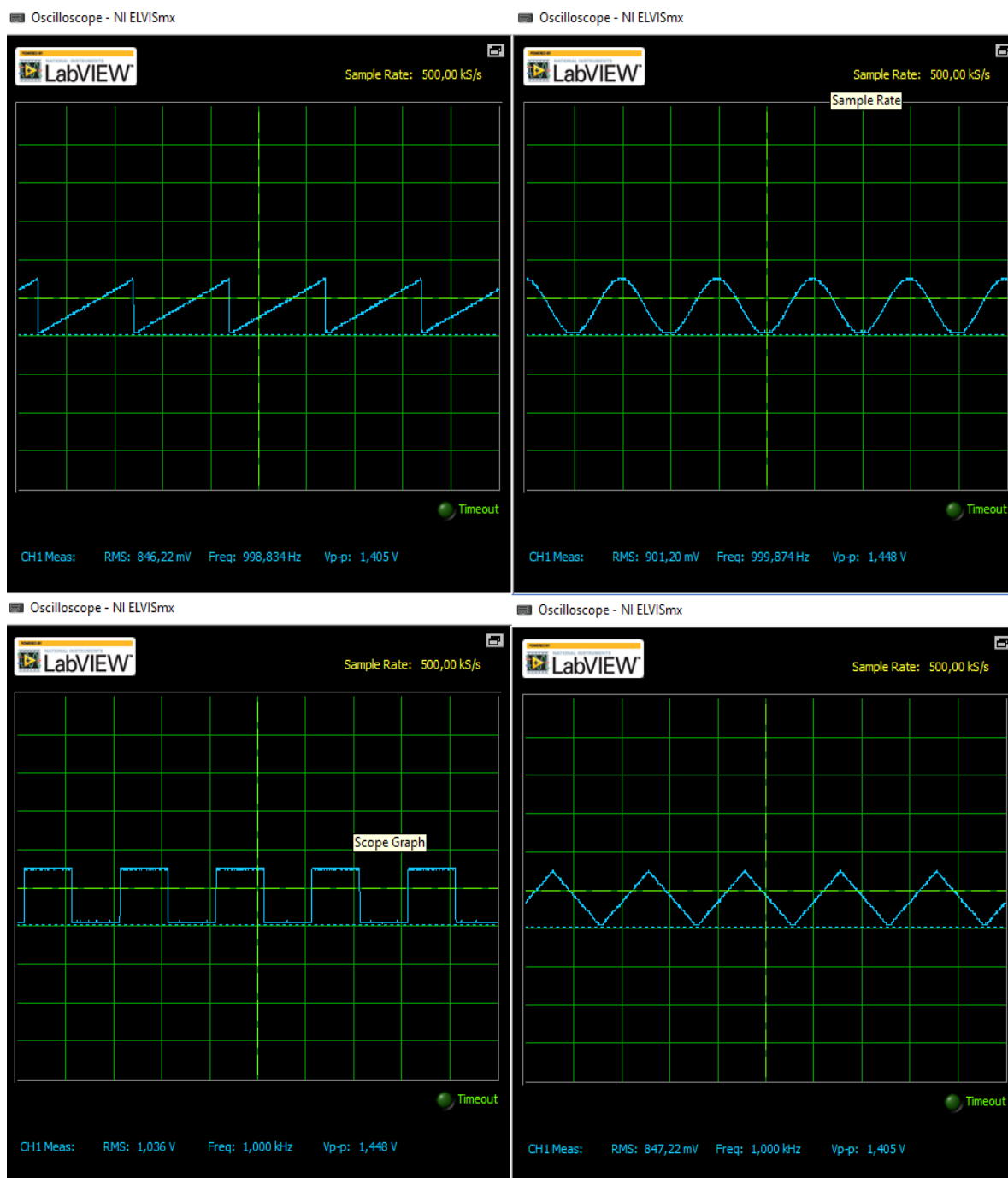


Рисунок 4.13 — Сигнали різної форми генератора на частоті 1кГц

4.5 Функціональний DDS генератор на STM32F429I Disco та AD9850

До складу макета генератора (рисунок 4.14) входять, безпосередньо, сам цифровий синтезатор частоти AD9850 і налагоджувальна плата STM32F429I Disco.

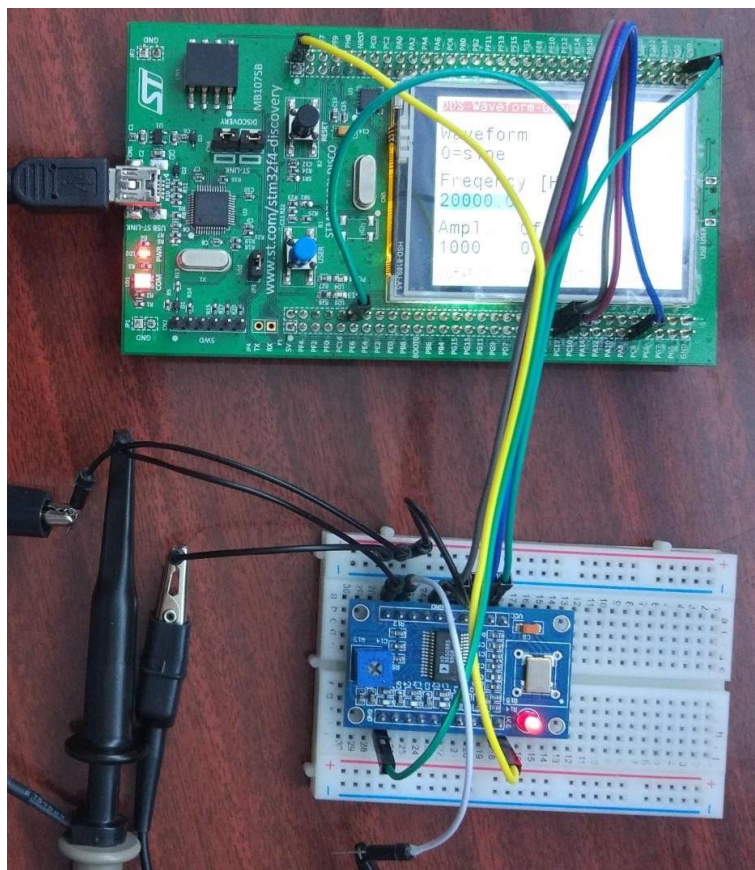


Рисунок 4.14 — Макет генератора на STM32F429I Disco та AD9850

Даний варіант складання найбільш зручний для використання і при невеликих доробках, може бути реалізований як єдиний пристрій.

Пристрій генерує сигнал по п'яти каналах, один канал безпосередньо генерує сам МК з частотою до 50 кГц і можливістю зміни форми, амплітуди, зміщення сигналу, решта чотирьох генеруватиметься вже синтезатором з частотою до 40 МГц. Управління частотою вихідного сигналу здійснюється за допомогою РК-дисплея плати. Поточне значення частоти також відображається на дисплеї.

Зазначимо, що в даному варіанті макета завантаження даних у модуль проводилося за допомогою послідовного інтерфейсу, у зв'язку з великою зайнятістю портів для роботи дисплея та реалізацією DDS генератора на самій платі.

Сигнал із виходів синтезатора був знятий за допомогою осцилографа Agilent Technologies InfiniiVision моделі DSO5052A (рисунки 4.15—4.17).

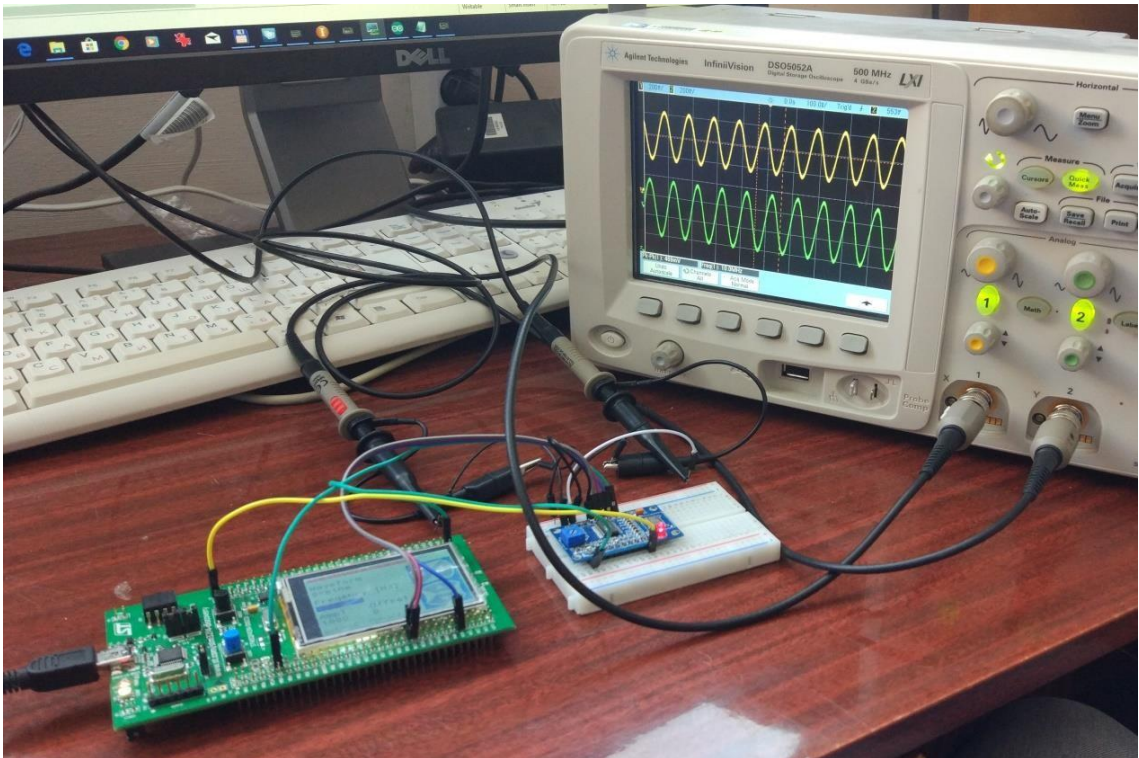


Рисунок 4.15 — Генерування сигналу частотою 10 МГц

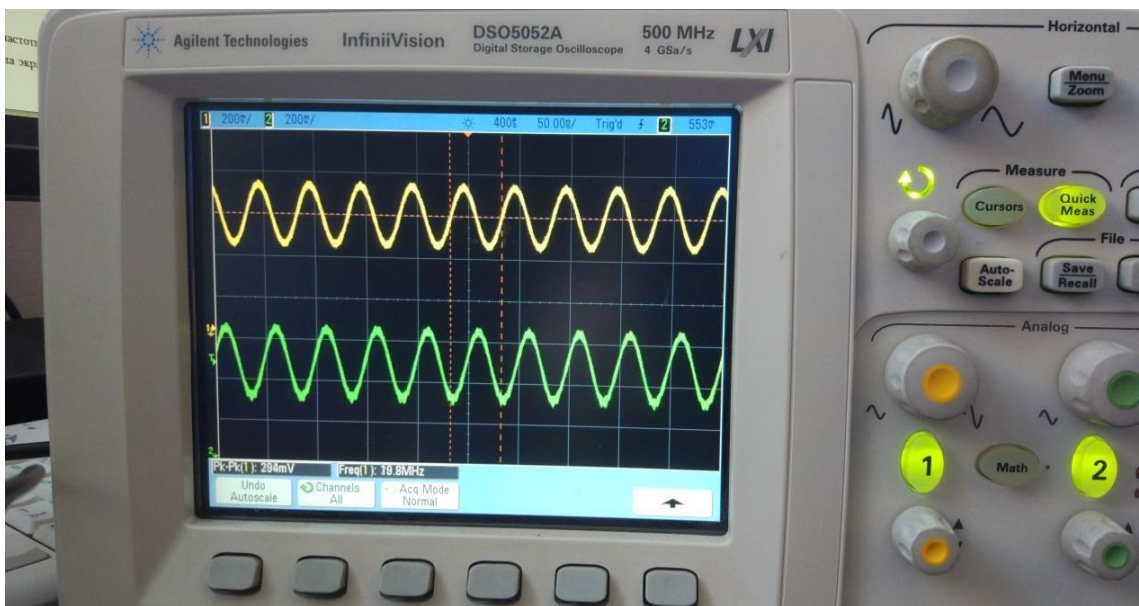


Рисунок 4.16 — Генерування сигналу частотою 19,8 МГц

За допомогою аналізатора спектра поспостерігали коефіцієнт гармонічних спотворень сигналу (рисунок 4.17).

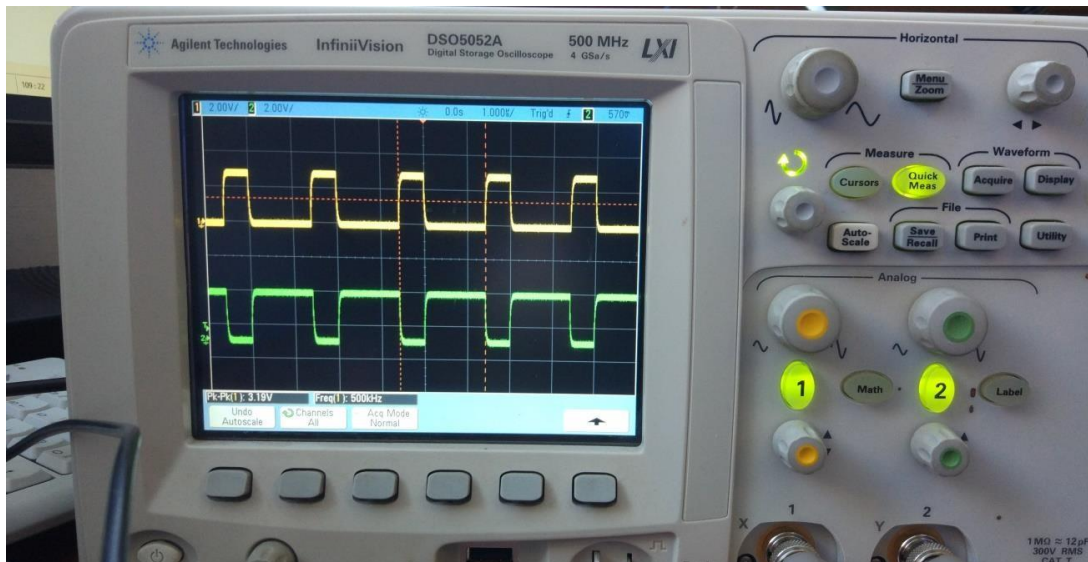


Рисунок 4.17 — Генерування сигналу частотою 500 кГц

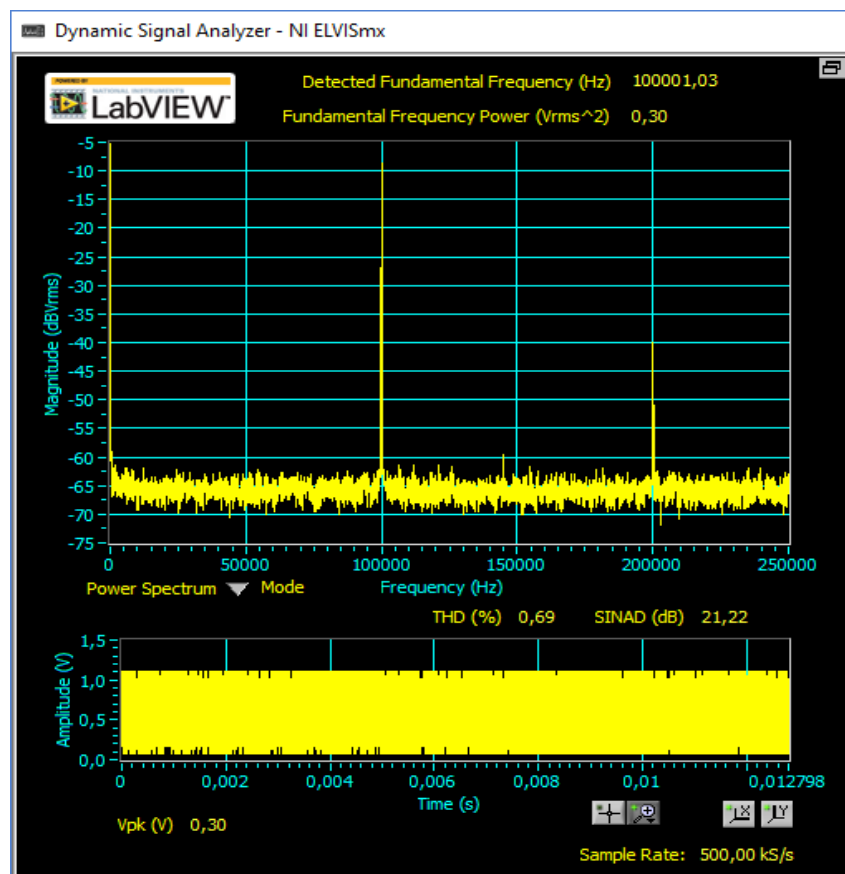


Рисунок 4.18 – Спектр сигналу

Як видно з рисунку КДМ дорівнює 0,69%. Пов'язано це з дискретним характером сигналу, що генерується. Спектральна чистота вихідного сигналу DDS залежить від якості ЦАП.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку системи цифрового генерування аналогових сигналів. Метою розробки є вдосконалення системи цифрового генерування аналогових сигналів. Особливість полягає у розробці бюджетної системи цифрового генерування аналогових сигналів, що зможе замінити дорогі та застарілі моделі генераторів у різноманітних технічних системах.

Аналогом може бути генератор сигналів DDS 2КАН 60МГц, JUNTEK JDS6600-60M, 6000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження табл. 5.1

Ринкові переваги					
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х

Продовження табл. 5.1

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	41	40	42
Середньоарифметична сума балів	$(41+40+42) / 3 = 41$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що нова система цифрового генерування аналогових сигналів є бюджетним значно дешевим варіантом, що зможе замінити дорогі та застарілі моделі генераторів у різноманітних технічних системах.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів за місяць, 20 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	32000	1600,00	38	60800,000
Програміст	28000	1400,00	38	53200,000
Всього				114000,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 11 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 11 \% / 100 \% \quad (5.2)$$

$$Z_d = (114000,00 \cdot 11 \% / 100 \%) = 12540,00 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (114000,00 + 12540,00) \cdot 22 \% / 100 \% = 27838,80 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_{\text{в}} \cdot 12} \cdot t_{\text{вик}} \quad [\text{Грн.}] \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ — термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 22000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,90 міс.

$$A_{\text{обл}} = \frac{22000}{2} \times \frac{1,9}{12} = 1741,667 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{нем.ак.}} = 11000$ грн.

Таблиця 5.5 — Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	22000	2	1,90	1741,667
Офісне обладнання (меблі)	21000	4	1,90	831,250
Приміщення	950000	20	1,90	7520,833
Всього				10093,75

Тарифи на електроенергію для не побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де V — вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

P — встановлена потужність обладнання, кВт. $P = 0,4$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,4 \cdot 8 \cdot 38 \cdot 6,2 = 678,528 \text{ (грн.)}$$

5.2.6 Інші витрати та загальновиробничі витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{iv}}{100\%}, \quad (5.6)$$

де H_{iv} — норма нарахування за статтею «Інші витрати».

$$I_e = 114000,00 * 75\% / 100\% = 85500 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 114000,00 * 150\% / 100\% = 171000 \text{ (грн.)}$$

5.2.7 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 114000,00 + 12540,00 + 27838,80 + 10093,75 + 11000 + 678,53 + 85500 + 171000 = 432651,08 \text{ грн.}$$

5.2.8 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 432651,08 / 0,5 = 865302 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

— вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

— зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

— кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

— визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

— абсолютного економічного ефекту (чистого дисконтованого доходу);

— внутрішньої економічної дохідності (внутрішньої норми дохідності);

— терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої

комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_0$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2023 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 1500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її

вдосконалення, можна буде підняти її ціну на 250 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 12000 шт., протягом другого року – на 10000 шт., протягом третього року на 8000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*250 + (1500 + 250)*12000)*0,8333*0,35*(1 - 0,18) = 4304999,828 \text{ грн.}$$

$$\Delta\Pi_2 = (0*250 + (1500 + 250)*(12000+10000))*0,8333*0,35*(1 - 0,18) = 9207916,298 \text{ грн.}$$

$$\Delta\Pi_3 = (0*250 + (1500 + 250)*(12000+10000+8000))*0,8333*0,35*(1 - 0,18) = 12556249,498 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 26069165,62 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.10)$$

де $\Delta\Pi$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T — період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t — період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\begin{aligned} \text{ПП} &= (4304999,828/(1+0,1)^1)+(9207916,298/(1+0,1)^2)+(12556249,498/ \\ &/(1+0,1)^3) = 3913636,21 + 7609848,18 + 9433696,091 = 20957180,48 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (5.11)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв}=2...5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 865302 = 1730604,31 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = \text{ПП} - PV, \quad (5.12)$$

$$E_{абс} = 20957180,48 - 1730604,31 = 19226576,17 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_ϵ . Для цього використаємо формулу:

$$E_\epsilon = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.13)$$

$T_{ж}$ — життєвий цикл наукової розробки, роки.

$$E_\epsilon = \sqrt[3]{(1 + 19226576,17/1730604,31) - 1} = 1,296$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = (0,09...0,14)$;

f — показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_b > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (5.15)$$

$$T_{ок} = 1 / 1,296 = 0,77 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,77 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 865302 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,77 роки.

ВИСНОВКИ

У магістерській кваліфікаційній роботі спроектовано систему цифрового генерування аналогових сигналів.

Виконано огляд та аналіз існуючих систем цифрового генерування аналогових сигналів, що дало змогу вивчити існуючі методи генерування аналогових сигналів та засобів для їх реалізації, принципи їх функціонування і наявні переваги та недоліки.

Відповідно до розглянутих існуючих варіантів запропоновано вдосконалення. Для цього розроблено структурну схему системи, описано нововведені блоки та їх призначення і функціонування, підібрано реальні компоненти сучасної електроніки. Вибір обґрунтовано та визначено, що основними компонентами будуть мікроконтролери сімейства STM32 із ядром Cortex-M4F фірми STMicroelectronics та цифровий синтезатор AD9850 фірми Analog Devices, Inc. За основу взято відлагоджувальні комплекси – STM32F4 Discovery та STM32F429I Disco.

Для управління мікроконтролерами розроблено програмне забезпечення із різними функціональними можливостями. Запропоновано декілька варіанти вдосконалення та модифікації генераторів аналогових сигналів, що дає можливість побудувати генератори:

- із використанням технічних можливостей плати STM32F4 Discovery;
- із використанням цифрового синтезатора AD9850 та STM32F4 Discovery, а також Arduino LCD Keypad Shield — для відображення інформації та зручного керування;
- STM32F429I Disco та цифрового синтезатора AD9850.

Пояснено переваги та недоліки запропонованих вдосконалень, проведено дослідження та моделювання варіантів цифрового генерування аналогових сигналів.

Проведено економічний розрахунок системи, що розробляється.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Carmine Noviello Mastering STM32: eBook. Leanpub, 2018. 852 pages.
2. Кравчук О.С. Розробка функціонального генератора сигналів на основі прямого цифрового синтезу / О.С. Кравчук, В.І. Манаєнков // Радіоелектронні та комп'ютерні системи. 2007. № 4 (23). С. 22 - 26.
3. STM32F103C8 - Mainstream Performance line, Arm Cortex-M3 MCU with 64 Kbytes of Flash memory, 72 MHz CPU, motor control, USB and CAN. st.com: веб-сайт. URL: <https://www.st.com/en/microcontrollersmicroprocessors/stm32f103c8.html> (дата звернення: 10.11.2023).
4. STM32CubeMX - STM32Cube initialization code generator st.com: веб-сайт. URL: <https://www.st.com/en/development-tools/stm32cubemx.html> (дата звернення: 08.06.2021).
5. STM32 32-bit Arm Cortex MCUs. URL: <https://www.st.com/en/microcontrollers/stm32-32-bit-arm-cortex-mcus.html> (дата звернення 03.10.2018).
6. Getting started with STM32 st.com: веб-сайт. URL: https://wiki.st.com/stm32mcu/wiki/STM32StepByStep:STM32_step_by_step_overview (дата звернення: 10.11.2023).
7. STM32-base project: веб-сайт. URL: <https://stm32-base.org/guides/gettingstarted.html> (дата звернення: 10.11.2023).
8. STM32 microcontroller GPIO configuration for hardware. st.com: веб-сайт. URL: [an4899-stm32-microcontroller-gpio-configuration-for-hardware-settings-and-lowpower-consumption.pdf](https://www.st.com/en/development-tools/an4899-stm32-microcontroller-gpio-configuration-for-hardware-settings-and-lowpower-consumption.pdf) (дата звернення: 10.11.2023).
9. Чинков В.Н. Цифрові засоби вимірюваної техніки військового призначення: підручник / В.Н.Чинков. Х.: ХУПС, 2007. Ч. 1. 244 с.
10. Горлач А.А. Цифровая обробка сигналів в вимірювальній техніці / А.А. Горлач, М.Я. Мінц, В.Н. Чинков. К., 2005. 156 с.
11. Майстри графіки: нове покоління STM32F4 за допомогою контролера TFT. URL:<http://www.compel.com/lib/ne/2013/8/6-mastera-grafiki-novopokolenie-stm32f4-s-podderzhkojkontrollera-tft/> (дата звернення 10.11.2023).

12. AD9850 CMOS, 125 MHz Complete DDS Synthesizer Data Sheet (REV. H). URL: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9850.pdf> (Дата звернення 10.11.2023).
13. EIM377 AD9850 Signal Generator Module. URL: https://reipoom.files.wordpress.com/2013/08/eim377_ad9850_signal_generator_module_v01.pdf (Дата звернення 10.11.2023).
14. Опис плати STM32F429IDiscovery. URL: <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090> (дата звернення 10.11.2023).
15. Приклади з вихідними кодами плати STM32F429IDiscovery. URL: <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF25909> (дата звернення 10.11.2023).
16. Відлагоджувальна плата STM32F429 I в якості осцилографа. URL: <https://docplayer.ru/66638669-Otladochnaya-plata-stm32f429i-v-kachestve-oscillografa.html> (Дата звернення 10.11.2023).
17. Han Y. Design of Data Transfer Module Базований на STM32 в Monitoring System of Cold Chain Transportation / Y. Han, L. Meng // Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control. 2015. P. 78 - 80.
18. Fang X. Design and Implementation of Constant Pressure Water Supply Monitoring System Based on STM32 / X. Fang, K. Zhang // 17th International Conference on Communication Technology. 2017. P. 1487 - 1491.
19. D. Q. Power Management of Analgesic Equipment Based on STM32 / Q. Deng, J. Zhong, Y. Liu, Y. Yong // 8th International Conference on CIS & RAM, Ningbo, China. 2017. P. 151 - 154.

ДОДАТОК А

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____ проф., д.т.н. О. Д. Азаров

«__» _____ 20__ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Система цифрового генерування аналогових сигналів»

08-54.МКР.006.00.000 ПЗ

Науковий керівник

д.т.н., проф. каф. ОТ

_____ Азаров О.Д.

виконав:

магістрант 2 курсу,

_____ Дідур І.В.

1. Підстава виконання магістерської кваліфікаційної роботи

1.1 Генератор є важливою ланкою в багатьох електронних пристроях, чи це тестове, лабораторне обладнання, медичні або радіоелектронні системи. Крім технічних характеристик, важливими для користувача залишаються такі параметри, як зручність застосування, компактність і низька вартість генераторів.

1.2 Наказ про затвердження теми МКР

2 Мета і призначенням МКР

2.1 Метою роботи є вдосконалення системи цифрового генерування аналогових сигналів.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3 Вихідні дані для виконання МКР

Вихідні дані для виконання МКР: мікроконтролери сімейства STM32 із ядром Cortex-M4F фірми STMicroelectronics та цифровий синтезатор AD9850 фірми Analog Devices, Inc, відлагоджувальні комплекси – STM32F4 Discovery та STM32F429I Disco.

4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

- розробити програмне забезпечення для управління мікропроцесором;
- вдосконалити та запропонувати модифікації генераторів аналогових сигналів.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в табл. А.1.

6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

Таблиця А.1 — Етапи МКР

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Прим.
1	Постановка мети та задач роботи	21.10.23	
2	Огляд та аналіз систем цифрового генерування аналогових сигналів	25.10-30.10.23	
3	Проектування апаратного забезпечення системи	31.10-08.11.23	
4	Складання структурної схеми системи	09.11-15.11.23	
5	Вибір інструментів розробки	16.11-.20.11.23	
6	Проектування програмного забезпечення системи	21.11-25.11.23	
7	Використання інструментів розробки	26.11-31.11.23	
8	Дослідження спроектованої системи	01.12-04.12.23	
9	Розрахунок економічної частини роботи	01.12-04.12.23	
10	Оформлення пояснювальної записки та ілюстративного матеріалу	05.12.23	
11	Аналіз виконання роботи, висновки, додатки		
12	Перевірка якості виконання магістерської роботи та усунення недоліків		

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21.

ДОДАТОК Б

Код програми для генератора на STM32F4 Discovery

```
#include "tm_stm32f4_dac_signal.h"

TM_DAC_SIGNAL_Result_t
TM_DAC_SIGNAL_SetCustomSignal(TM_DAC_SIGNAL_Channel_t      DACx,
uint16_t* Signal_Data, uint16_t Signal_Length, double frequency);

uint16_t DAC_SIGNAL_Sinus[DAC_SIGNAL_SINUS_LENGTH] = {
    2047, 2447, 2831, 3185, 3498, 3750, 3939, 4056,
    4095, 4056, 3939, 3750, 3495, 3185, 2831, 2447,
    2047, 1647, 1263, 909, 599, 344, 155, 38,
    0, 38, 155, 344, 599, 909, 1263, 1647
};

uint16_t DAC_SIGNAL_Triangle[DAC_SIGNAL_TRIANGLE_LENGTH] = {
    0, 256, 512, 768, 1024, 1279, 1535, 1791,
    2047, 2303, 2559, 2815, 3071, 3326, 3582, 3838,
    4095, 3838, 3582, 3326, 3071, 2815, 2559, 2303,
    2047, 1791, 1535, 1279, 1024, 768, 512, 256
};

uint16_t DAC_SIGNAL_Sawtooth[DAC_SIGNAL_SAWTOOTH_LENGTH] =
{
    0, 132, 264, 396, 528, 660, 792, 924,
    1057, 1189, 1321, 1453, 1585, 1717, 1849, 1981,
    2113, 2245, 2377, 2509, 2641, 2773, 2905, 3037,
    3170, 3302, 3434, 3566, 3698, 3830, 3962, 4095
};
```

```

uint16_t DAC_SIGNAL_Square[DAC_SIGNAL_SQUARE_LENGTH] = {
    0, 4095
};

TIM_TypeDef *DAC_TIM[2];
uint8_t dac_timer_set[2] = {0, 0};

TM_DAC_SIGNAL_Result_t
TM_DAC_SIGNAL_Init(TM_DAC_SIGNAL_Channel_t DACx, TIM_TypeDef*
TIMx) {
    uint16_t GPIO_Pin;

    /* Check used timer */
    /* Set proper trigger */
    if (
        TIMx == TIM2 ||
        TIMx == TIM4 ||
        TIMx == TIM5 ||
        TIMx == TIM6 ||
        TIMx == TIM7 ||
        TIMx == TIM8
    ) {
        /* Set timer */
        DAC_TIM[DACx] = TIMx;
        /* Set flag */
        dac_timer_set[DACx] = 1;
    } else {
        /* Timer is not valid */
        return TM_DAC_SIGNAL_Result_TimerNotValid;
    }
}

```

```

/* Select proper GPIO pin */
if (DACx == TM_DAC1) {
    GPIO_Pin = GPIO_PIN_4;
} else {
    GPIO_Pin = GPIO_PIN_5;
}

/* Initialize proper GPIO pin */
TM_GPIO_Init(GPIOA,          GPIO_Pin,          TM_GPIO_Mode_AN,
TM_GPIO_OType_PP, TM_GPIO_PuPd_NOPULL, TM_GPIO_Speed_Fast);

/* Return OK */
return TM_DAC_SIGNAL_Result_Ok;
}

TM_DAC_SIGNAL_Result_t
TM_DAC_SIGNAL_SetSignal(TM_DAC_SIGNAL_Channel_t          DACx,
TM_DAC_SIGNAL_Signal_t signal_type, double frequency) {
    TM_DAC_SIGNAL_Result_t result;
    switch (signal_type) {
        case TM_DAC_SIGNAL_Signal_Sinus:
            result = TM_DAC_SIGNAL_SetCustomSignal(DACx,
DAC_SIGNAL_Sinus, DAC_SIGNAL_SINUS_LENGTH, frequency);
            break;
        case TM_DAC_SIGNAL_Signal_Sawtooth:
            result = TM_DAC_SIGNAL_SetCustomSignal(DACx,
DAC_SIGNAL_Sawtooth, DAC_SIGNAL_SAWTOOTH_LENGTH, frequency);
            break;
    }
}

```

```

        case TM_DAC_SIGNAL_Signal_Triangle:
            result = TM_DAC_SIGNAL_SetCustomSignal(DACx,
DAC_SIGNAL_Triangle, DAC_SIGNAL_TRIANGLE_LENGTH, frequency);
            break;
        case TM_DAC_SIGNAL_Signal_Square:
            result = TM_DAC_SIGNAL_SetCustomSignal(DACx,
DAC_SIGNAL_Square, DAC_SIGNAL_SQUARE_LENGTH, frequency);
            break;
        default:
            result = TM_DAC_SIGNAL_Result_Error;
    }

    /* Return result */
    return result;
}

```

```

TM_DAC_SIGNAL_Result_t
TM_DAC_SIGNAL_SetCustomSignal(TM_DAC_SIGNAL_Channel_t DACx,
uint16_t* Signal_Data, uint16_t Signal_Length, double frequency) {
    DAC_InitTypeDef DAC_InitStruct;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStruct;
    DMA_InitTypeDef DMA_InitStruct;
    TM_TIMER_PROPERTIES_t Timer_Data;

    /* Check if timer is set */
    if (!dac_timer_set[DACx]) {
        return TM_DAC_SIGNAL_Result_Error;
    }

    /* Check used timer */

```

```

/* Set proper trigger */
if (DAC_TIM[DACx] == TIM2) {
    DAC_InitStruct.DAC_Trigger = DAC_Trigger_T2_TRGO;
} else if (DAC_TIM[DACx] == TIM4) {
    DAC_InitStruct.DAC_Trigger = DAC_Trigger_T4_TRGO;
} else if (DAC_TIM[DACx] == TIM5) {
    DAC_InitStruct.DAC_Trigger = DAC_Trigger_T5_TRGO;
} else if (DAC_TIM[DACx] == TIM6) {
    DAC_InitStruct.DAC_Trigger = DAC_Trigger_T6_TRGO;
} else if (DAC_TIM[DACx] == TIM7) {
    DAC_InitStruct.DAC_Trigger = DAC_Trigger_T7_TRGO;
} else if (DAC_TIM[DACx] == TIM8) {
    DAC_InitStruct.DAC_Trigger = DAC_Trigger_T8_TRGO;
} else {
    /* Timer is not valid */
    return TM_DAC_SIGNAL_Result_TimerNotValid;
}

/* Get timer data */
TM_TIMER_PROPERTIES_GetTimerProperties(DAC_TIM[DACx],
&Timer_Data);

/* Get period and prescaler values */
TM_TIMER_PROPERTIES_GenerateDataForWorkingFrequency(&Timer
_Data, frequency * Signal_Length);

/* Check valid frequency */
if (Timer_Data.Frequency == 0) {
    return TM_DAC_SIGNAL_Result_Error;
}

```

```

/* Enable DAC clock */
RCC->APB1ENR |= RCC_APB1ENR_DACEN;

/* Enable DMA1 clock */
RCC->AHB1ENR |= RCC_AHB1ENR_DMA1EN;

/* Initialize DAC */
DAC_InitStruct.DAC_WaveGeneration = DAC_WaveGeneration_None;
DAC_InitStruct.DAC_OutputBuffer = DAC_OutputBuffer_Enable;

/* Disable DMA */
if (DACx == TM_DAC1) {
    /* Init DAC channel 1 */
    DAC_Init(DAC_Channel_1, &DAC_InitStruct);
} else if (DACx == TM_DAC2) {
    /* Init DAC channel 2 */
    DAC_Init(DAC_Channel_2, &DAC_InitStruct);
}

/* Enable timer clock */
TM_TIMER_PROPERTIES_EnableClock(DAC_TIM[DACx]);

/* Time base configuration */
TIM_TimeBaseStructInit(&TIM_TimeBaseStruct);
TIM_TimeBaseStruct.TIM_Period = Timer_Data.Period - 1;
TIM_TimeBaseStruct.TIM_Prescaler = Timer_Data.Prescaler - 1;
TIM_TimeBaseStruct.TIM_ClockDivision = 0;
TIM_TimeBaseStruct.TIM_CounterMode = TIM_CounterMode_Up;

/* Initialize timer */

```



```

TIM_TimeBaseInit(DAC_TIM[DACx], &TIM_TimeBaseStruct);

/* Enable TIM selection */
TIM_SelectOutputTrigger(DAC_TIM[DACx],
TIM_TRGOSource_Update);

/* Set DMA options */
DMA_InitStruct.DMA_Memory0BaseAddr = (uint32_t)Signal_Data;
DMA_InitStruct.DMA_DIR = DMA_DIR_MemoryToPeripheral;
DMA_InitStruct.DMA_BufferSize = Signal_Length;
DMA_InitStruct.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStruct.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStruct.DMA_PeripheralDataSize =
DMA_PeripheralDataSize_HalfWord;
DMA_InitStruct.DMA_MemoryDataSize =
DMA_MemoryDataSize_HalfWord;
DMA_InitStruct.DMA_Mode = DMA_Mode_Circular;
DMA_InitStruct.DMA_Priority = DMA_Priority_High;
DMA_InitStruct.DMA_FIFOMode = DMA_FIFOMode_Disable;
DMA_InitStruct.DMA_FIFOThreshold = DMA_FIFOThreshold_HalfFull;
DMA_InitStruct.DMA_MemoryBurst = DMA_MemoryBurst_Single;
DMA_InitStruct.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;

switch (DACx) {
    case TM_DAC1:
        /* Set peripheral location = 12bit right aligned for channel 1 */
        DMA_InitStruct.DMA_PeripheralBaseAddr =
(uint32_t)&DAC->DHR12R1;

        /* Disable DMA */

```

```

DMA_DeInit(DAC_SIGNAL_DMA_DAC1_STREAM);

/* Set channel used */
DMA_InitStruct.DMA_Channel =
DAC_SIGNAL_DMA_DAC1_CHANNEL;

/* Initialize DMA */
DMA_Init(DAC_SIGNAL_DMA_DAC1_STREAM,
&DMA_InitStruct);

/* Enable DMA Stream for DAC Channel 1 */
DMA_Cmd(DAC_SIGNAL_DMA_DAC1_STREAM,
ENABLE);

/* Enable DAC Channel 1 */
DAC_Cmd(DAC_Channel_1, ENABLE);

/* Enable DMA for DAC Channel 1 */
DAC_DMAMCmd(DAC_Channel_1, ENABLE);
break;
case TM_DAC2:
/* Disable DMA */
DMA_DeInit(DAC_SIGNAL_DMA_DAC2_STREAM);

/* Set channel used */
DMA_InitStruct.DMA_Channel =
DAC_SIGNAL_DMA_DAC2_CHANNEL;

/* Set peripheral location = 12bit right aligned for channel 2 */

```

```
        DMA_InitStruct.DMA_PeripheralBaseAddr    =
(uint32_t)&DAC->DHR12R2;

        /* Initialize DMA */
        DMA_Init(DAC_SIGNAL_DMA_DAC2_STREAM,
&DMA_InitStruct);

        /* Enable DMA Stream for DAC Channel 2 */
        DMA_Cmd(DAC_SIGNAL_DMA_DAC2_STREAM,
ENABLE);

        /* Enable DAC Channel 2 */
        DAC_Cmd(DAC_Channel_2, ENABLE);

        /* Enable DMA for DAC Channel 2 */
        DAC_DMAMCmd(DAC_Channel_2, ENABLE);
        break;
    default:
        break;
}

/* Enable timer */
DAC_TIM[DACx]->CR1 |= TIM_CR1_CEN;

/* Return OK */
return TM_DAC_SIGNAL_Result_Ok;
}
```

ДОДАТОК В

Код програми для генератора на AD9850 та STM32F4 Discovery, а також

Arduino LCD Keypad Shield

```
#include < LiquidCrystal .h> // Підключення бібліотек

LiquidCrystal lcd (8, 9, 4, 5, 6, 7); // Ініціалізація LCD із зазначенням пінів
float freq = 10000; // Оголошення змінних - Частота
float bigStep = 1000; // крок зміни частоти при натисканні вгору/вниз
float littleStep = 10; // Крок зміни частоти при натисканні праворуч/ліворуч
int lcd_key = 0;
int adc_key_in = 0;
#define btnRIGHT 0 // Створення директив для кнопок
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5
#define W_CLK 15 // Пін A1 Arduino - підключений до CLK
#define FQ_UD 16 // Пін A2 Arduino - підключений до FQ (FU)
#define DATA 17 // Пін A3 Arduino - підключений до DATA
#define RESET 18 // Пін A4 Arduino - підключений до RST
#define pulseHigh(pin) { digitalWrite (pin, HIGH ); digitalWrite (pin, LOW ); }
//-----
int read_LCD_buttons(){ // Функція зчитування натискання кнопок
  adc_key_in = analogRead (0);
  if (adc_key_in > 1000) return btnNONE;
  if (adc_key_in < 50) return btnRIGHT;
  if (adc_key_in < 150) return btnUP;
  if (adc_key_in < 315) return btnDOWN;
  if (adc_key_in < 600) return btnLEFT;
```

```

if (adc_key_in < 850) return btnSELECT;
return btnNONE;}

//-----

void tfr_byte( byte data){ // Функція побитної відправки байта
for ( int i=0; i<8; i++, data>>=1) { // даних у модуль генератора
digitalWrite (DATA, data & 0x01);
pulseHigh(W_CLK);}} // Подача імпульсу на CLK після кожного біта
//-----

void sendFrequency( double frequency) { // Перетворення та відправка
int32_t freq = frequency * 4294967295/125000000; // значення частоти
for ( int b=0; b<4; b++, freq>>=8) {tfr_byte(freq & 0xFF);}
tfr_byte(0x000); // Відправлення завершального контрольного байта
pulseHigh(FQ_UD);} // Оновлення частоти генератора
//-----

void setup ()
{
lcd.begin (16, 2); // Старт бібліотеки. Вказівкою кількості символів та
рядків
pinMode (FQ_UD, OUTPUT );
pinMode (W_CLK, OUTPUT );
pinMode (DATA, OUTPUT );
pinMode (RESET, OUTPUT );
pulseHigh(RESET); // Відправлення імпульсів для запуску модуля
генератора
pulseHigh(W_CLK);
pulseHigh(FQ_UD);
}
//-----

void loop ()
{

```

```

lcd.setCursor(0,0); // Далі виведення поточного значення частоти
lcd.print ("Freq:");
lcd.setCursor(6,0);
lcd.print ("");
lcd.setCursor(6,0);
if (freq<1000){lcd.print (freq);
lcd.print ("Hz");}
if ((freq>=1000)&&(freq<1000000)){lcd.print (freq/1000);
lcd.print ("kHz");}
if ((freq>=1000000)&&(freq<50000000)){lcd.print (freq/1000000);
lcd.print ("MHz");}
lcd.setCursor(0,1);
lcd.print ("Genie v1.0");
if (freq <100) {bigStep = 10; // Визначення кроку грубої та точної
littleStep = 1;} // Налаштування в залежності від частоти
if ((freq>= 100) & & (freq <1000)) {bigStep = 100;
littleStep = 1;}
if ((freq>=1000)&&(freq<10000)){bigStep = 1000;
littleStep = 10;}
if ((freq>=10000)&&(freq<100000)){bigStep = 10000;
littleStep = 100;}
if ((freq>=100000)&&(freq<1000000)){bigStep = 100000;
littleStep = 1000;}
if ((freq>=1000000)&&(freq<10000000)){bigStep = 1000000;
littleStep = 10000;}
if ((freq>=10000000)&&(freq<40000000)){bigStep = 10000000;
littleStep = 100000;}
lcd_key = read_LCD_buttons(); // Зчитування клавiш
switch (lcd_key) // Далі обробка натискань клавiш
{

```

```
case btnRIGHT:
    { freq += littleStep;
      break ;}
case btnLEFT:
    { freq -= littleStep;
      break ;}
case btnUP:
    { freq += bigStep;
      break ;}
case btnDOWN:
    { if (freq == bigStep){ freq -= (bigStep/10);}
      else { freq -= bigStep;}
      break ;}
case btnSELECT:
    { break ;}
case btnNONE:
    { break ;}
}
if (freq<1) freq=1; // Обмеження значень частоти
if (freq>40000000) freq=40000000;
sendFrequency(freq); // Виклик функції відправлення частоти
delay (200); // Пауза 200 мс
}
```

ДОДАТОК Г

Код програми для генератора AD9850 до STM32F4 Discovery

```
#include "stm32f4xx.h"

void GPIOinit(void){
    //GPIOB init
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10 | GPIO_Pin_11 | GPIO_Pin_12;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
    GPIO_Init(GPIOC, &GPIO_InitStructure);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_All;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
    GPIO_Init(GPIOD, &GPIO_InitStructure);
    GPIO_Write(GPIOD,0x0000);
}

void AD9850_reset(void){
    GPIO_WriteBit(GPIOC, GPIO_Pin_10,Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_11,Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_12,Bit_RESET);
    for (int i=0;i<5000;i++);
    GPIO_WriteBit(GPIOC, GPIO_Pin_12,Bit_SET
    for (int i=0;i<5000;i++);
```



```

GPIO_WriteBit(GPIOC, GPIO_Pin_10, Bit_SET);
for (int i=0;i<5000;i++);
GPIO_WriteBit(GPIOC, GPIO_Pin_10, Bit_RESET);
for (int i=0;i<5000;i++);
}
uint8_t W[5] = {0,0,0,0,0};
uint32_t tword;
void AD9850_ParallelSend(uint32_t freq){
tword = ( (freq / 100) * 3436;
W[4] = tword;
W[3] = tword >> 8;
W[2] = tword >> 16;
W[1] = tword >> 24;
W[0] = 0x00;
GPIO_WriteBit(GPIOC, GPIO_Pin_12, Bit_RESET);
for (int k=0;k<5;k++){
GPIO_Write(GPIOD, W[k]);
GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
}
GPIO_WriteBit(GPIOC, GPIO_Pin_12, Bit_SET);
}
int main(void) {
uint32_t frequency = (150000);
GPIOinit();
AD9850_reset();
while(1) {
AD9850_ParallelSend(frequency);
// for (int d = 0; d<50000;d++);
}
}

```

ДОДАТОК Д

Код програми для генератора AD9850 до STM32F429I Disco

```
#include "main.h"
#include "dds.h"
int main(void)
{
    uint32_t gui_redraw=0;
    SystemInit();
    dds_init();
    #if DDD_USE_LCD==1
        GUI_CHANGE_t gui_status;
        gui_init();
    #endif
    #if DDD_USE_COM==1
        COM_CHANGE_t com_status;
        com_init();
    #endif
    dds_set_frq(100.0);
    dds_set_amplitude(2000);
    dds_start();
    while(1)
    {
        #if DDD_USE_LCD==1
            gui_status=gui_do(gui_redraw);
            if(gui_status==GUI_CHANGE_FRQ) {
                UB_Led_Toggle(LED_GREEN);
                dds_set_frq(DDS.fsoll);
            }
        #endif
        else if(gui_status==GUI_CHANGE_AMP) {
```

```
    UB_Led_Toggle(LED_GREEN);
    dds_set_amplitude(DDS.amplitude);
}
else if(gui_status==GUI_CHANGE_OFF) {
    UB_Led_Toggle(LED_GREEN);
    dds_set_offset(DDS.offset);
}
else if(gui_status==GUI_CHANGE_WAVE) {
    UB_Led_Toggle(LED_GREEN);
    dds_set_wave(DDS.wave);
}
#endif
#if DDD_USE_COM==1
    com_status=com_do();
    if(com_status==COM_CHANGE_FRQ) {
        UB_Led_Toggle(LED_GREEN);
        dds_set_freq(DDS.fsoll);
        gui_redraw=1;
    }
    else if(com_status==COM_CHANGE_AMP) {
        UB_Led_Toggle(LED_GREEN);
        dds_set_amplitude(DDS.amplitude);
        gui_redraw=1;
    }
    else if(com_status==COM_CHANGE_OFF) {
        UB_Led_Toggle(LED_GREEN);
        dds_set_offset(DDS.offset);
        gui_redraw=1;
    }
    else if(com_status==COM_CHANGE_WAVE) {
```

```
UB_Led_Toggle(LED_GREEN);  
dds_set_wave(DDS.wave);  
gui_redraw=1;  
}  
else {  
    gui_redraw=0;  
}  
#endif  
}  
}
```

ДОДАТОК Е

Структура схема системи цифрового генерування аналогових сигналів

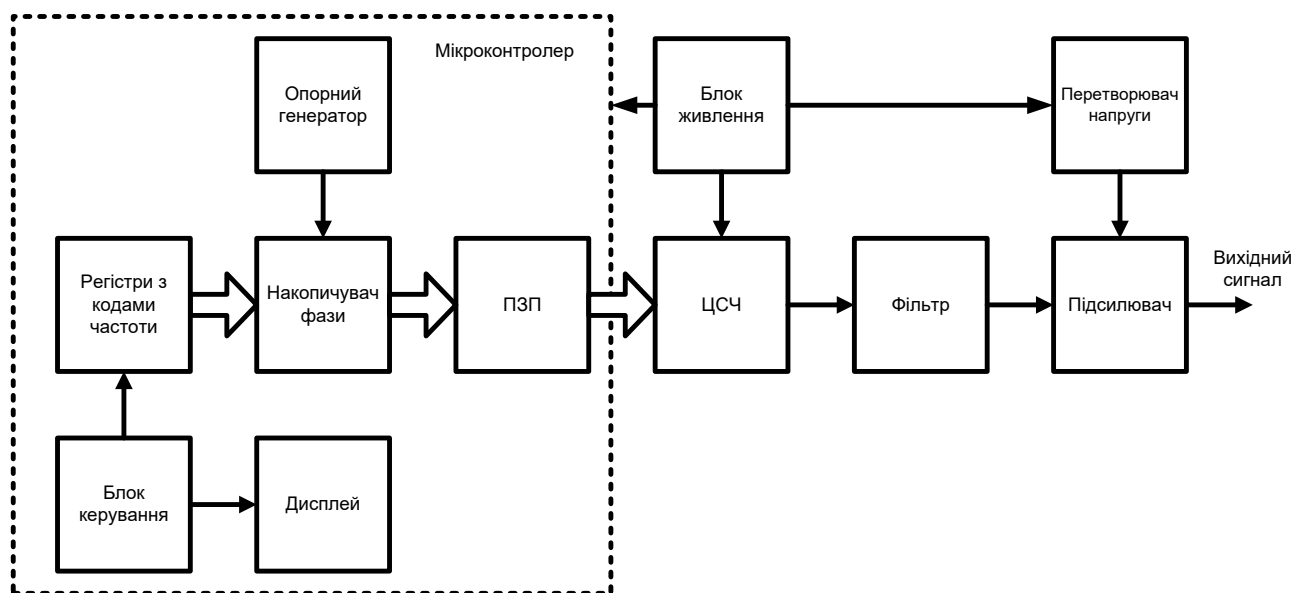


Рисунок Є1 — Структура схема системи цифрового генерування аналогових сигналів

ДОДАТОК Ж

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Система цифрового генерування аналогових сигналів

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 85,5% Схожість 14,5%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____
(підпис)

Захарченко С.М.
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____
(підпис)

Дідур І.В.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

Азаров О.Д.
(прізвище, ініціали)