

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система візуалізації
аналогових сигналів на основі STM32

Виконав: студент 4 курсу, групи СІ-42

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Курцба Т. В.

(прізвище та ініціали)

Керівник

(підпис)

Лецишин Ю.З

(прізвище та ініціали)

Нормоконтроль

(підпис)

Tum € . B.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Пастух О.А.

(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«27» січня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Курцебі Тарасу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система візуалізації аналогових
Сигналів на основі STM32

Керівник роботи Лецишин Юрій Зіновійович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » січня 2025 року № 4/7-53

2. Термін подання студентом завершеної роботи 19.06.2025р.

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання системи візуалізації аналогових сигналів

2. Проєктування системи візуалізації аналогових сигналів

3. Розробка та тестування системи візуалізації аналогових сигналів

4. Безпека життєдіяльності, основи охорони праці.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Структурна схема системи

2. Схема електрична принципова

3. Блок-схема алгоритму роботи програмного забезпечення

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>д.т.н., професор Пилипець М. І.</i>		

7. Дата видачі завдання 27.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	<i>27.01 – 02.02</i>	<i>Викон.</i>
2.	<i>Аналіз технічного завдання системи візуалізації аналогових сигналів</i>	<i>12.02 – 27.03</i>	<i>Викон.</i>
3.	<i>Проектування системи візуалізації аналогових сигналів</i>	<i>06.04 – 03.05</i>	<i>Викон.</i>
4.	<i>Розробка та тестування комп'ютеризованої системи</i>	<i>05.05 – 27.05</i>	<i>Викон.</i>
5.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>28.05 – 02.06</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>03.06 – 08.06</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>09.06 – 15.06</i>	<i>Викон.</i>
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06 – 22.06</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>26.06</i>	<i>Викон.</i>

Студент

(підпис)

Курцеба Тарас Володимирович

(прізвище та ініціали)

Керівник роботи

(підпис)

Лецишин Юрій Зіновійович

(прізвище та ініціали)

АНОТАЦІЯ

Курцеба Т.В. Комп'ютеризована система візуалізації аналогових сигналів на базі STM32: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 – комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: STM32F103C8T6, візуалізація сигналів, аналоговий сигнал, TFT-дисплей, ST7735, енкодер, ADC, LL бібліотеки, осцилограф, SPI, DMA.

Кваліфікаційна робота присвячена розробці комп'ютеризованої системи візуалізації аналогових електричних сигналів у реальному часі на основі мікроконтролера STM32F103C8T6.

У першому розділі розглянуто постановку задачі, проведено аналіз методів візуалізації аналогових сигналів у вбудованих системах, а також окреслено переваги використання мікроконтролерів STM32 для побудови портативних цифрових осцилографів.

У другому розділі представлено структурну схему системи, виконано обґрунтування вибору апаратної бази, зокрема ADC, енкодерів, дисплея, а також інтерфейсів взаємодії.

У третьому розділі детально описано реалізацію програмної частини з використанням LL бібліотек: налаштування багатоканального ADC у режимі безперервного сканування з DMA, логіку формування осцилограми, обробку сигналу та реалізацію тригерного режиму.

У четвертому розділі викладено питання безпеки життєдіяльності при роботі з лабораторними електронними пристроями та наведено заходи для мінімізації ризиків у процесі експлуатації системи.

ANNOTATION

Kurtseba T.V. Computerized system for visualizing analog signals based on STM32: Bachelor's Qualification Thesis: Specialty 123 – Computer Engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: STM32F103C8T6, signal visualization, analog signal, TFT display, ST7735, encoder, ADC, LL libraries, oscilloscope, SPI, DMA.

This bachelor's qualification thesis is dedicated to the development of a computerized system for real-time visualization of analog electrical signals based on the STM32F103C8T6 microcontroller.

The first chapter presents the problem statement, analyzes existing methods of analog signal visualization in embedded systems, and outlines the advantages of using STM32 microcontrollers for the construction of portable digital oscilloscopes.

The second chapter introduces the system architecture, justifies the selection of hardware components including ADC, encoders, TFT display, and communication interfaces.

The third chapter describes in detail the software implementation using LL (Low-Layer) libraries: configuration of a multi-channel ADC in continuous scan mode with DMA support, the logic of waveform generation, signal processing, and implementation of trigger mode.

The fourth chapter covers safety and health issues when working with laboratory electronic devices and provides recommendations for minimizing operational risks during system usage.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ АНАЛОГОВИХ СИГНАЛІВ	11
1.1 Аналіз вимог до системи візуалізації.....	11
1.2 Аналіз можливих рішень поставленого завдання	13
1.3 Технічні характеристики розроблюваної системи	19
РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ АНАЛОГОВИХ СИГНАЛІВ	23
2.1 Розробка узагальненої структури комп'ютерної системи візуалізації аналогових сигналів	23
2.2 Обґрунтування вибору апаратного забезпечення проектованої комп'ютерної системи	26
2.3 Обґрунтування вибору програмного забезпечення проектованого комп'ютерного засобу.....	29
2.4 Опис схеми електрично-принципової.....	30
2.5 Опис блок-схеми алгоритму	33
2.6 Опис макета системи.....	35
РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ АНАЛОГОВИХ СИГНАЛІВ	38
3.1 Ініціалізація мікроконтролера та периферії.....	38
3.2 Апаратна конфігурація АЦП та передача в DMA	40

					КС КРБ 123.222.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Курцеба Т.В.			Комп'ютеризована система візуалізації аналогових сигналів на основі STM32			Літ.	Арк.	Аркушів	
Перевір.		Лецишин Ю.З.								6	
Реценз.		Пастух О.А.						ТНТУ, каф. КС, гр. СІ-42			
Н. Контр.		Тиш Є.В.									
Затверд.		Осухівська Г.М.									

3.3	Керування через енкодери	42
3.4	Корекція значень	45
3.5	Особливості роботи з дисплеєм	46
3.6	Тестування пристрою.....	51
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ ..		53
4.1	Долікарська допомога при ушкодженні м'яких тканин, суглобів і кісток	53
4.2	Вплив електромагнітних полів (ЕМП) на людину та заходи щодо зменшення їх впливу.....	55
ВИСНОВКИ		57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		59
Додаток А Технічне завдання		
Додаток Б Лістинг коду програми для STM32		
Додаток В Перелік елементів схеми електричної принципової		

					КС КРБ 123.222.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

СПИСОК СКОРОЧЕНЬ

ADC – Analog-to-Digital Converter

DMA – Direct Memory Access

FPU – Floating Point Unit

GPIO – General-Purpose Input/Output

IDE – Integrated Development Environment

LL – Low Layer (STM32 low-level peripheral access library)

MCU – Microcontroller Unit

PCB – Printed Circuit Board

PWM – Pulse-Width Modulation

SPI – Serial Peripheral Interface

STM – STMicroelectronics

TFT – Thin-Film Transistor

USB – Universal Serial Bus

VCP – Virtual COM Port

CDC – Communications Device Class

					КС КРБ 123.222.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

З розвитком технологій популярності набули різні комп'ютерні рішення, зокрема мікроконтролерні системи, які стали основою численних електронних пристроїв, включаючи засоби вимірювання, керування та моніторингу. Одним із важливих інструментів для аналізу електричних сигналів у таких системах є осцилограф – пристрій, що дозволяє візуалізувати параметри аналогових сигналів у реальному часі. Традиційні осцилографи зазвичай мають велику вартість, габаритні розміри та складну архітектуру, що ускладнює їх використання в освітніх, лабораторних або аматорських умовах. Це спонукає до створення доступних, компактних та енергоефективних альтернатив на основі сучасних мікроконтролерів.

Метою даного проєкту є розробка компактної комп'ютеризованої системи для візуалізації аналогових сигналів, що виконує функції осцилографа з використанням мікроконтролера STM32F103. Обраний контролер належить до сімейства STM32 на базі ядра ARM Cortex-M3, вирізняється високою продуктивністю, широким набором периферійних модулів та енергоефективністю, що робить його придатним для задач реального часу, пов'язаних із обробкою аналогових даних.

У рамках проєкту буде реалізовано як апаратну, так і програмну частину системи. До складу пристрою входять блок аналого-цифрового перетворення (ADC), інтерфейс виводу на TFT-дисплей, система керування через енкодери та програмна обробка сигналу з використанням методів буферизації, інтерполяції та фільтрації. Також буде здійснено розрахунок енергоспоживання системи, з урахуванням живлення від літієвих акумуляторів через понижуючий стабілізатор напруги. Програмна частина створюється у середовищі STM32CubeIDE із використанням бібліотек LL (Low-Layer), що забезпечують прямий контроль над апаратними ресурсами для досягнення максимальної ефективності.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Важливим аспектом роботи є забезпечення точного вимірювання параметрів вхідного сигналу (амплітуда, частота, форма) та інтуїтивно зрозумілого відображення цієї інформації на дисплеї у вигляді графіка. Це дозволяє користувачу проводити оперативний аналіз електричних процесів у схемах, що є особливо корисним у навчальних цілях, дослідницьких роботах та для практики з цифровою обробкою сигналів.

Розроблений пристрій можна використовувати як навчальний осцилограф для вивчення електроніки, як частину лабораторного стенду або як інструмент для діагностики електронних модулів. Його модульна структура та використання відкритих рішень дозволяють у майбутньому масштабувати або доповнювати функціональність, наприклад, реалізувати передачу даних по UART, USB або через бездротові інтерфейси.

Таким чином, розробка комп'ютеризованої системи візуалізації аналогових сигналів на базі STM32F103 сприяє розширенню можливостей мікроелектроніки в галузях навчання, наукових досліджень та прикладної електроніки, а також демонструє перспективність застосування сучасних мікроконтролерів для створення багатофункціональних, доступних та ефективних пристроїв.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ АНАЛОГОВИХ СИГНАЛІВ

1.1 Аналіз вимог до системи візуалізації

З плином часу розвиток цифрової електроніки та автоматизації зростає тому і з'являється потреба у засобах, що дозволяють точно, швидко та зручно досліджувати аналогові сигнали. Такими засобами є осцилографи – електронні вимірювальні пристрої, здатні виводити графік напруги у функції від часу. З розвитком мікроконтролерної техніки з'явилась можливість реалізувати осцилографи не у вигляді дорогих лабораторних приладів, а як вбудовані пристрої з мінімальними витратами ресурсів, що використовують недорогі компоненти.

Розробка комп'ютеризованої системи візуалізації аналогових сигналів вимагає дотримання певних технічних, програмних та функціональних вимог, які повинні забезпечити точність вимірювання, оперативну обробку сигналів, та зрозумілу візуалізацію на екрані.

До основних функціональних компонентів системи належать: аналого-цифровий перетворювач (ADC) – відповідає за дискретизацію аналогового сигналу, блок обробки сигналу – реалізований у вигляді програмних алгоритмів на мікроконтролері, блок зберігання даних – відповідає за буферизацію, можливе використання кільцевого буфера або DMA, інтерфейс виводу – забезпечує передачу даних на дисплей, інтерфейс керування – реалізується кнопками, енкодерами або сенсорним екраном.

Вимоги до такої системи можна умовно поділити на функціональні, апаратні, програмні та інтерфейсні вимоги.

					КС КРБ 123.222.00.00 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Аналіз технічного завдання системи візуалізації аналогових сигналів			Літ.	Арк.	Аркушів
Розроб.		Курцеба Т.В.								
Перевір.		Лещишин Ю.З.							11	
Реценз.		Пастух О.А.						ТНТУ, каф. КС, гр. СІ-42		
Н. Контр.		Тиш Є.В.								
Затверд.		Осухівська Г.М.								

Функціональні вимоги:

- Візуалізація вхідного аналогового сигналу на дисплеї у вигляді графіка.
- Підтримка зміни масштабу по осі часу.
- Можливість фіксації сигналу (тригер).
- Вимірювання частоти, амплітуди, середнього та пікового значення.

Апаратні вимоги:

- Достатній діапазон частот вхідного сигналу (до кількох сотень кГц).
- Мінімум 10-бітний ADC.
- TFT-дисплей з інтерфейсом SPI або паралельним з'єднанням.
- Обмежене енергоспоживання.
- Підтримка роботи від автономного живлення (акумуляторів).
- Компактність та надійність корпусу.

Програмні вимоги:

- Стабільна робота під навантаженням (високошвидкісний сигнал).
- Ефективне використання периферії (ADC, DMA, SPI, таймери).
- Зрозумілий графічний інтерфейс (GUI).
- Низький рівень затримок при візуалізації.
- Можливість масштабування або оновлення прошивки.

Інтерфейсні вимоги:

- Вихід на TFT-дисплей.
- Можливість керування наприклад через енкодери або кнопки.
- Можливість додаткової передачі даних (UART, USB).

Для забезпечення усіх цих вимог необхідно розробити систему, яка не тільки підтримуватиме базову функціональність осцилографа, але й дозволить гнучке налаштування параметрів роботи користувачем [1].

					КС КРБ 123.222.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз можливих рішень поставленого завдання

В електроніці візуалізація аналогових є важливим аспектом електроніки при розробці та діагностиці помилок.

Існує кілька основних підходів до реалізації пристроїв цього класу, кожен з яких має свої переваги й недоліки залежно від контексту застосування, бюджету, доступності компонентів і вимог до характеристик системи. У цьому підрозділі розглянуто три найбільш поширені архітектурні рішення.

Програмно-апаратні рішення на базі мікроконтролерів (рис. 1.1). Цей підхід передбачає створення пристрою на основі універсального мікроконтролера, який самостійно виконує збір, обробку та відображення аналогових сигналів (рис 1.1).

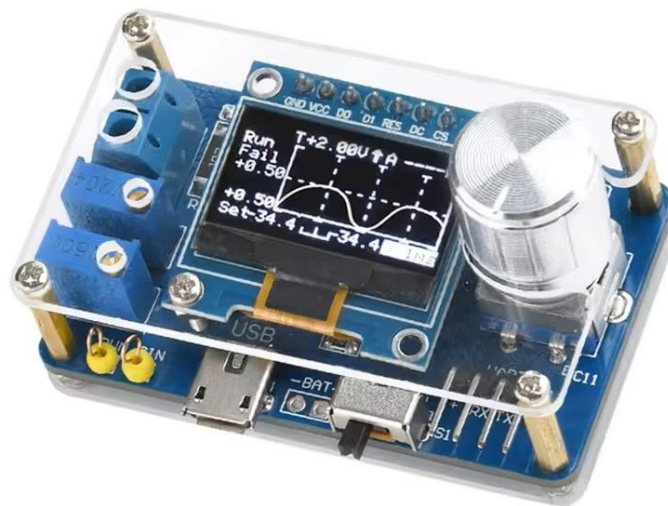


Рисунок 1.1 – Вигляд система на базі контролера

Серед основних переваг використання даного підходу при розробці системи є:

- Низька вартість реалізації завдяки доступності різноманітних мікроконтролерів зокрема STM32 і типових дисплеїв на кшталт TFT чи OLED дисплеїв.

– Гнучкість, оскільки розробка проводиться на програмованому мікроконтролері, можна реалізувати унікальні функції залежно від поставлених потреб, я043А-от масштабування, фільтрацію, автотригери тощо.

– Компактність, всі компоненти можуть розміститися на одній платі, тому що система складається лише з кількох компонентів.

– Автономність, можливість живлення від акумулятора, як вбудованого а саме типу 18650 або ж інших джерел вихідна напруга яких складає 12.

Проте, при використанні вище вказаного методу розробки виявлено певні недоліки, такі як:

– Обмежена швидкодія та низька пропускну здатність ADC, особливо в мікроконтролерах бюджетного класу.

– Невисока точність вимірювань у порівнянні з професійними цифровими осцилографами.

– Обмеженість дисплейної системи (низька частота оновлення, роздільна здатність).

Такий підхід є найкращим у випадку реалізації портативного осцилографа, оскільки дає змогу самостійно контролювати функціональність на кожному етапі – від вибору джерела сигналу до алгоритму візуалізації.

Також одним із можливих варіантів реалізації рішення є апаратні на базі цифрових осцилографічних інтегральних мікросхем. У цьому варіанті використовується спеціалізована інтегральна мікросхема або модуль, призначений для перетворення безперервного аналогового сигналу у цифрову форму, що забезпечує високошвидкісний збір і обробку сигналу, передаючи готові дані до мікроконтролера або комп'ютера. Прикладами таких рішень є аналогово-цифрові інтерфейсні мікросхеми наприклад, Analog Devices AD9283 або TI ADS7042 (рис 1.2). AD9283 – це високошвидкісний 8-розрядний паралельний АЦП з частотою дискретизації до 100 MSPS, що дозволяє точно відображати сигнали до десятків МГц. Вихідна шина

					КС КРБ 123.222.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

паралельна, тому для взаємодії з мікроконтролером чи FPGA необхідний відповідний інтерфейс. TI ADS7042 – це 12-розрядний серійний АЦП з частотою дискретизації до 1 MSPS, з інтерфейсом SPI.



Рисунок 1.2 – Вигляд модуля осцилографа на базі AD9283

Завдяки низькому енергоспоживанню та компактності добре підходить для портативних осцилографів. SPI-передача даних дає змогу зручно зчитувати оцифровану інформацію по черзі. АЦП підключається до аналогового входу (через буфер чи підсилювач), а далі генерує цифрові значення у вигляді масиву вибірок. Ці значення передаються у пам'ять, де можуть бути оброблені мікроконтролером або виведені на дисплей.

Осцилографічні модулі з USB-інтерфейсом наприклад Cypress FX2LP (рис. 1.3). Такі модулі виконують роль "зовнішнього вимірювального пристрою", який виконує лише захоплення сигналу, передаючи його у комп'ютер для візуалізації. Найчастіше побудовані на мікросхемах Cypress FX2LP або аналогах. Cypress FX2LP – це мікроконтролер із USB 2.0, який підтримує високу швидкість передачі (до 480 Мбіт/с), має вбудований інтерфейс FIFO та дозволяє зчитувати дані з АЦП у реальному часі. Він може

передавати ці дані через USB у хост-систему, де вони відображаються на екрані за допомогою спеціального програмного забезпечення.

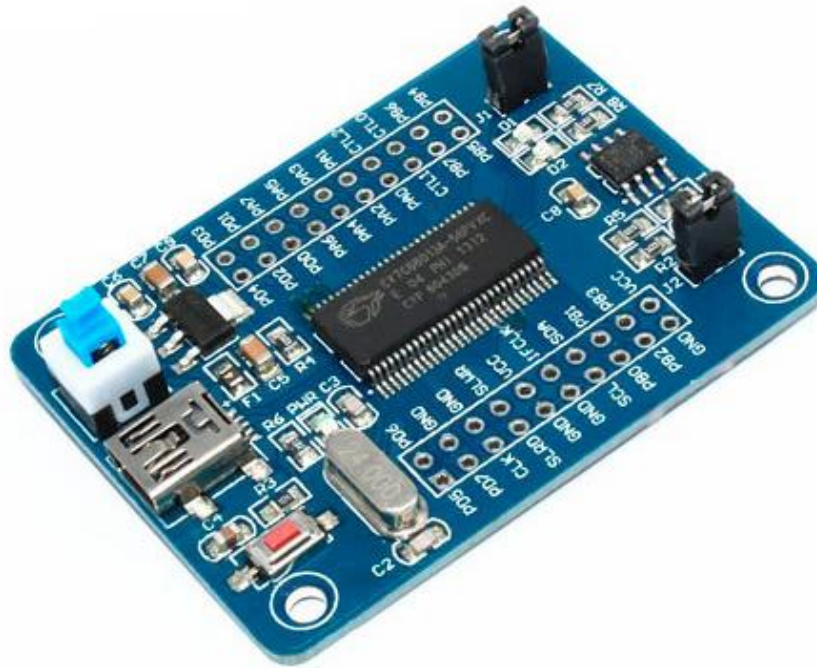


Рисунок 1.3 – Вигляд логічного аналізатора Cypress FX2LP

FPGA – це програмована схема, що дозволяє створювати високошвидкісну обробку даних у паралельному режимі. FPGA мають вбудовані або зовнішньо підключені АЦП. FPGA приймає аналогові сигнали через зовнішній АЦП або вбудований модуль, далі сигнал обробляється у реальному часі, фільтрація, вимірювання піків, автотригери і зберігається у буферну пам'ять. Після цього дані можуть передаватися на дисплей, зовнішній інтерфейс або ж відображатися у реальному часі через ПЗ.

Дані способи реалізації мають порівняно високу точність та швидкодію, що дозволяє досліджувати високочастотні сигнали, містять в собі інтегровані фільтри та буфери, що зменшують похибки дозволяють легко нарощувати кількість каналів. Однак, незважаючи на переваги такого підходу в ньому є і наступні недоліки: висока вартість компонентів, складність проектування, що вимагає глибоких знань цифрової схемотехніки та RF-дизайну, менша

гнучкість програмного налаштування – поведінка часто визначається логікою самого чипа.

Цей варіант доцільний у випадках, коли необхідна професійна точність та швидкість вимірювань, але в умовах обмеженого бюджету або навчального проєкту він надмірно складний.

Гібридні рішення з використанням зовнішнього комп'ютера або смартфона. Такий підхід поєднує просту апаратну частину (рис. 1.4), яка виконує лише оцифровування сигналу, із потужною обчислювальною системою ПК, яка забезпечує обробку та візуалізацію.

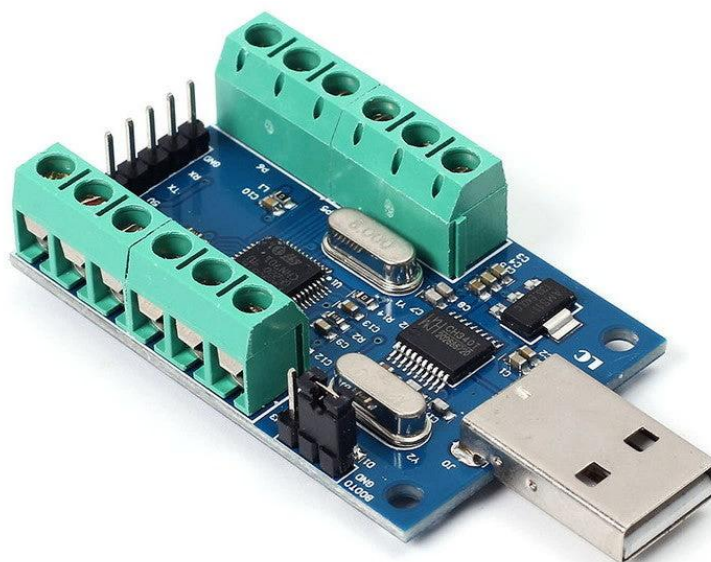


Рисунок 1.4 – Вигляд USB АЦП модуля компанії Lazmin

Мікроконтролер з ADC надсилає дані через USB або Bluetooth, програмне забезпечення на ПК або мобільному пристрої відображає графік у реальному часі. Таким чином навантаження на обробку перенесено з мікроконтролера на ПК, що зменшує вимоги до мікросхеми, а також гнучкість користувацького інтерфейсу за рахунок застосування повноцінних GUI-фреймворків, краща якість візуалізації анімовані графіки, масштабування,

спектральний аналіз тощо. Однак даний підхід має суттєві недоліки такі як: необхідність мати зовнішній пристрій, затримки при передачі даних, ускладнення налаштування ПЗ та драйверів, потреба у сумісності з різними ОС, підвищене енергоспоживання через необхідність підтримки зв'язку та передачі даних.

Цей підхід актуальний для масового споживача. Проте в рамках навчального проєкту він вносить додаткову складність.

В результаті дослідження виявлено що у контексті цього проєкту доцільно зосередитись саме на вбудованому програмно-апаратному рішенні на базі STM32F103, яке дозволяє досягти прийняттого балансу між вартістю, функціональністю та складністю реалізації, тому надалі розглядаються можливі рішення на основі вище вказаного контролера.

Застосування STM32 з внутрішнім АЦП. Мікроконтролери STM32, зокрема STM32F103, мають вбудовані 12-бітні аналого-цифрові перетворювачі, що дозволяють з частотою до 1 МГц зчитувати аналогові сигнали. У поєднанні з DMA можна досягти майже безперервного зчитування сигналу з мінімальним навантаженням на CPU.

Даний мікроконтролер має низку переваг зокрема: простота реалізації, низька вартість компонентів, пряма інтеграція з іншими модулями такими як дисплей чи енкодери, висока гнучкість у налаштуваннях [2].

Серед вище зазначених переваг даний мікроконтролер також має і певні недоліки такі як: обмежена частота дискретизації, потреба у фільтрації високочастотного шуму, залежність від точності вбудованого опорного джерела.

Застосування зовнішнього швидкодіючого ADC. Ще одним варіантом реалізації системи є використання зовнішніх АЦП таки як, AD9280 чи ADS7042, які забезпечують вищу частоту дискретизації. Зчитування даних у такому випадку потребує використання швидкісного SPI або паралельного інтерфейсу. Такий підхід забезпечить кращу точність і швидкість а також

					КС КРБ 123.222.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

можливість обробки швидкоплинних сигналів. Проте це і ускладнить схему з'єднань, збільшить енергоспоживання, а також підвищить саму вартість системи.

1.3 Технічні характеристики розроблюваної системи

У цьому підрозділі наведено технічні параметри апаратних і програмних компонентів, обраних для реалізації комп'ютеризованої системи візуалізації аналогових сигналів. Конфігурація системи сформована з урахуванням компромісу між функціональністю, енергоспоживанням, складністю реалізації та загальною вартістю проекту.

У якості основного керуючого елемента системи обрано мікроконтролер STM32F103C8T6 (рис 1.5), що належить до лінійки STM32F1 від компанії STMicroelectronics. Його характеристики оптимально підходять для задач аналогової обробки та виведення інформації на дисплей.

Основні технічні параметри STM32F103C8T6:

- Архітектура ARM Cortex-M3.
- Тактова частота до 72 МГц.
- Об'єм Flash-пам'яті 64 КБ.
- Об'єм ОЗП (SRAM) 20 КБ.
- Вбудований 12-бітний ADC з до 1 MSPS (за ідеальних умов).
- Підтримка DMA для безперервного збору даних.
- Широкий набір таймерів і інтерфейсів USART, SPI, I2C, USB, CAN тощо.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

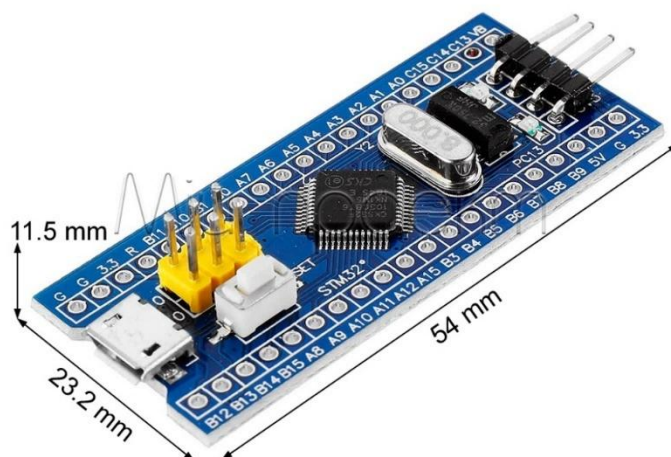


Рисунок 1.5 – Вигляд контролера STM32F103C8T6

Використовується вбудований 12-бітний ADC, що дозволяє досягати прийнятної точності (4096 рівнів квантування) при дослідженні аналогових сигналів. Швидкість дискретизації залежить від конфігурації тактованого джерела, часу зразка, кількості каналів, режиму роботи DMA та інших параметрів.

Вбудований АЦП STM32F103C8T6 має наступні характеристики:

- Роздільна здатність 12 біт.
- Максимальна частота дискретизації (на один канал) до ~ 1 MSPS.
- Кілька режимів роботи одиночний, безперервний, сканування.
- Підтримка DMA.
- Кількість каналів: до 16, однак в такому випадку падає частота дискретизації, точність вибірки та її розмір.

Для відображення сигналу використовується кольоровий TFT-дисплей з діагоналлю 1.8 дюйма та роздільною здатністю 128×160 пікселів (рис. 1.6). Передача даних до дисплея здійснюється через інтерфейс SPI, що забезпечує достатню швидкість оновлення екрану.

Характеристики дисплея:

- Тип TFT LCD.
- Контролер ST7735 або сумісний.
- Інтерфейс SPI.
- Роздільна здатність 128×160 пікселів.
- Кольоровість 16 біт (65 тис. кольорів).
- Напруга живлення 3.3 В.

Для реалізації виведення графіку сигналу застосовується спеціально розроблений буфер кадру, який періодично оновлюється в залежності від частоти кадру.

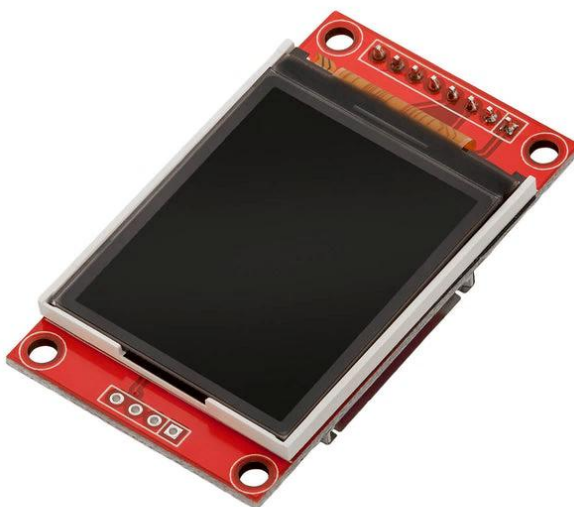


Рисунок 1.6 – Вигляд TFT дисплею

Інтерфейс керування реалізується за допомогою ротаційного енкодера з кнопкою, що забезпечує зручну навігацію по меню та налаштування параметрів тригер, канал, режим. У системі реалізовано обробку обертання та натискання, а також програмне подавлення шумів. За потреби можливе розширення інтерфейсу керування за рахунок додаткових кнопок або сенсорного екрану [3].

					КС КРБ 123.222.00.00 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Система підтримує автономне живлення від двох Li-Ion акумуляторів 18650, що перетворюється до необхідного рівня 3.3 В через стабілізатор HT7333 в максимально активному режимі роботи споживання системи складає до 90–120 мА а мінімальний час автономної роботи близько 10 годин. Також передбачена можливість живлення через microUSB для зручності зарядки та відладки.

Програмне забезпечення написане мовою C з використанням STM32CubeIDE та бібліотек Low-Layer (LL), що забезпечують низькорівневий доступ до апаратних ресурсів мікроконтролера. Основні функціональні блоки ПЗ:

- Ініціалізація периферії (ADC, SPI, DMA, таймери).
- Читання аналогових сигналів з буферизацією.
- Обробка даних яка включає усереднення та масштабування.
- Графічна візуалізація.
- Реакція на події керування (енкодер, меню).
- Відлагоджувальні інтерфейси (UART для налагодження).

					КС КРБ 123.222.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ АНАЛОГОВИХ СИГНАЛІВ

2.1 Розробка узагальненої структури комп'ютерної системи візуалізації аналогових сигналів

Проектування комп'ютеризованої системи візуалізації аналогових сигналів потребує формування чіткої структурної моделі, яка визначатиме загальні функціональні блоки, їхню взаємодію, а також основні інформаційні потоки між модулями. Така структуризація дозволяє ефективно організувати як апаратну, так і програмну реалізацію системи, забезпечуючи масштабованість, надійність та зручність обслуговування.

Узагальнена структура системи (рис 2.1) побудована за модульним принципом і включає такі основні функціональні блоки:

- Аналоговий вхідний блок (вхідний каскад).
- Аналого-цифрове перетворення (АЦП).
- Блок попередньої обробки сигналу.
- Блок керування (ядро системи – STM32).
- Інтерфейс відображення даних (дисплей).
- Блок живлення.
- Користувацький інтерфейс (енкодери, кнопки).
- Модуль збереження та/або передачі даних (UART, USB, ESP32, SD-карта тощо).

					КС КРБ 123.222.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Проектування системи візуалізації аналогових сигналів		
Розроб.		Курцеба Т.В.					
Перевір.		Лецишин Ю.З.					
Реценз.		Пастух О.А.					
Н. Контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						23	
					ТНТУ, каф. КС, гр. СІ-42		

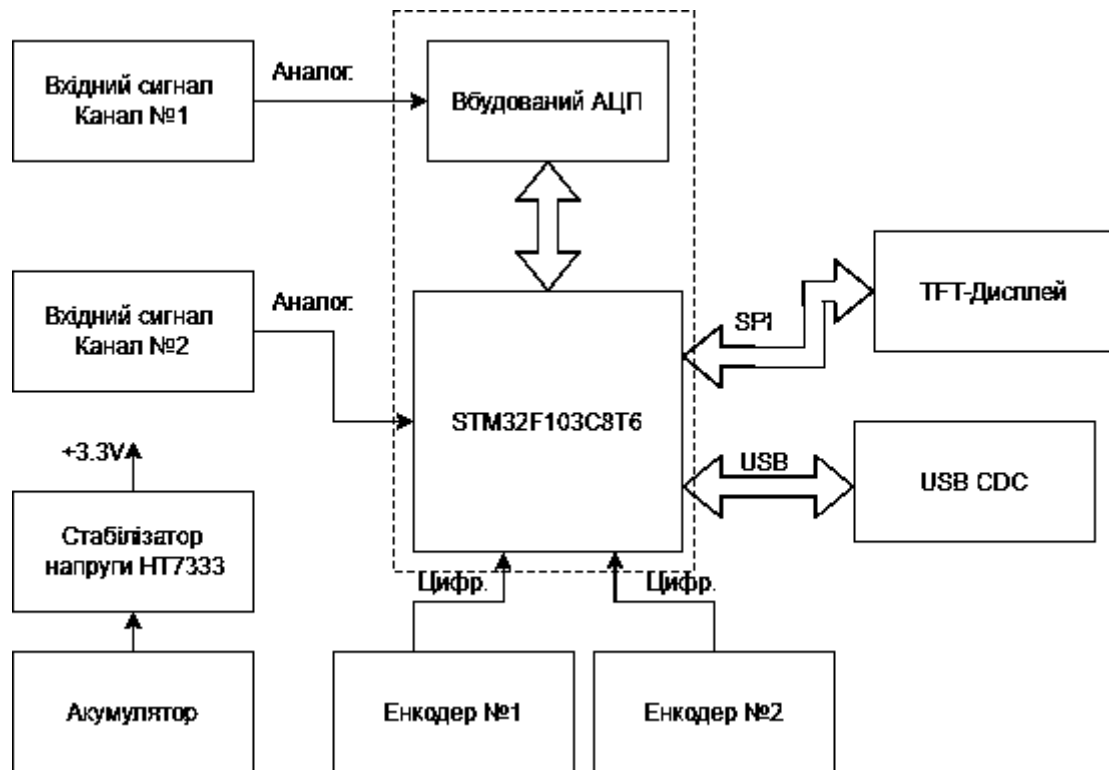


Рисунок 2.1 – Структурна схема системи комп’ютеризованої візуалізації аналогових сигналів на основі STM32

Аналоговий вхідний блок виконує попередню підготовку сигналу до оцифрування. Сюди входять елементи захисту від перенапруги, фільтрації шумів високих частот та буферизації. У більшості випадків застосовуються резистивні дільники, RC-фільтри та операційні підсилювачі (наприклад, MCP6L92 або TL072), що забезпечують масштабування сигналу до рівнів, сумісних з АЦП STM32 (0–3.3 В). Також до цього блоку може бути включено атенюатор з перемикачем коефіцієнта наприклад, 1:1 або 1:10, який дозволяє змінювати масштаб вимірювання для широкого динамічного діапазону.

Блок аналого-цифрового перетворення використовується вбудований 12-розрядний АЦП, який забезпечує частоту дискретизації до 1 Мзс при використанні DMA і мінімального циклу перетворення. Для забезпечення двоканального захоплення сигналів передбачено мультиплексування входів та використання DMA-контролера для безперервного збору даних. У більш складних варіантах можливе застосування зовнішніх АЦП, наприклад

ADS7042 або AD9283, які передають дані через SPI або паралельний інтерфейс відповідно, дозволяючи збільшити частоту дискретизації або точність перетворення.

Перед відображенням даних на дисплеї, сирі вибірки обробляються контролером STM32. Основні задачі цього блоку:

- компенсація зміщення нуля (offset).
- масштабування (gain).
- обмеження перепадів (кліпінг).
- фільтрація шумів (наприклад, цифрове згладжування).
- пошук тригера або аналіз пікових значень.

У разі необхідності можливе збереження вибірок у кільцевому буфері, що забезпечує реалізацію функції «затримки» зображення.

Далі відбувається керування за допомогою головного контролера системи. Він відповідає за ініціалізацію периферії, обробку сигналів, обробку введення з інтерфейсу користувача, управління дисплеєм, а також взаємодію з зовнішніми модулями. Робота мікроконтролера організовується за допомогою FreeRTOS або state machine, з чітким розділенням задач: обробка вибірок, обробка інтерфейсу, оновлення екрану тощо.

Для відображення оброблених даних використовується TFT-дисплей з інтерфейсом SPI (наприклад, 1.8" ST7735 або аналог). Графічна бібліотека реалізує функції виведення даних, сітки, значень каналів, коефіцієнтів масштабування, значень тригера тощо. Частота оновлення дисплея може бути обмежена пропускнуою здатністю SPI, тому слід оптимізувати буферизацію та режим відображення (наприклад, відображення лише зміни, частковий перерисунок).

Система живиться як від USB, так і від акумулятора 18650. Напруга 5 В стабілізується до 3.3 В за допомогою LDO-регулятора типу HT7333 або AMS1117. Особливу увагу слід приділити фільтрації живлення для АЦП та

					КС КРБ 123.222.00.00 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

аналогових каскадів, де зазвичай використовуються RC-фільтри та феритові намистини[4].

Інтерфейс реалізується через енкомери з кнопками, перемикачі режимів або сенсорні кнопки. Енкодер забезпечує зручну навігацію меню та зміну параметрів (наприклад, часова база, масштаб напруги, вибір каналу). Програмно реалізується антидребезг, підтримка довгого натискання, подвійного кліку тощо.

Для розширення функціоналу передбачена можливість передавати захоплені дані на зовнішній пристрій. Це може бути:

- передача по UART (відлагодження або моніторинг).
- передача через ESP32 на веб-сервер або смартфон.
- зв'язок із ПК через USB-COM емуляцію.

Узагальнена структурна схема є основою для подальшої деталізації функціональних блоків, розробки електричних схем, програмного коду та моделювання роботи системи. Такий підхід забезпечує системність розробки та дозволяє адаптувати пристрій до різних вимог користувача без повного перегляду архітектури.

2.2 Обґрунтування вибору апаратного забезпечення проектованої комп'ютерної системи

Розробка комп'ютеризованої системи візуалізації аналогових сигналів передбачає комплексний підхід до вибору апаратних компонентів. Центральним елементом такої системи є мікроконтролер, що відповідає за зчитування аналогових сигналів, їх оцифрування, обробку, буферизацію та передачу на дисплей. З огляду на функціональні вимоги проєкту, ключовим критерієм при виборі апаратної платформи є співвідношення між обчислювальними можливостями, доступними периферійними інтерфейсами, енергоспоживанням та гнучкістю програмування.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

У якості базової обчислювальної платформи обрано мікроконтролер STM32F103C8T6 серії STM32 на основі ядра ARM Cortex-M3. Цей мікроконтролер поєднує в собі високу продуктивність (72 МГц), низьке енергоспоживання, наявність швидкодіючих 12-бітових АЦП, підтримку DMA, SPI, I2C, UART та інші переваги. Саме ці характеристики дозволяють ефективно реалізувати зчитування аналогових сигналів з високою частотою дискретизації та передачею результатів на дисплей у режимі реального часу.

Для обґрунтування вибору було проведено аналіз найбільш популярних мікроконтролерів, що часто використовуються в аматорських та напівпрофесійних рішеннях. У таблиці 2.1 наведено порівняння ключових технічних характеристик деяких з них:

Таблиця 2.1 – Порівняльна характеристика обчислювальних платформ

Назва платформи	Ядро/Мікро контролер	Частота, МГц	RAM, КБ	Flash, КБ	АЦП	DMA	Вартість	GPIO
Arduino Uno	ATmega328P	16	2	32	10-біт, 6 каналів	Ні	Низька	14
ESP32	Xtensa Dual-Core	240	520	448	12-біт, 18 каналів	Так	Середня	34
STM32F103C8T6	ARM Cortex-M3	72	20	64	12-біт, 10 каналів	Так	Низька	37
Raspberry Pi	Broadcom BCM2711	до 1500	1024–8192	SD-карта	Зовнішній	Так	Висока	26

Як видно з таблиці, STM32F103C8T6 забезпечує достатню обчислювальну продуктивність при низькому енергоспоживанні, компактних розмірах, низькій вартості та розвиненій периферії. Arduino Uno не підтримує DMA і має надто низьку тактову частоту та обсяг RAM для обробки сигналів у реальному часі. ESP32 хоч і має більшу продуктивність, проте є складнішою у програмуванні, має вищу енергозатратність і в багатьох випадках

надлишкову функціональність. Raspberry Pi – це повноцінний одноплатний комп’ютер, який перевищує потреби даного проєкту та потребує окремої операційної системи, що ускладнює забезпечення реального часу. Окрім основного контролера, у системі використано:

- TFT-дисплей 1.8" SPI з роздільною здатністю 128x160 пікселів для виводу графіків.
- Енкодери з кнопками для керування меню та налаштувань.
- Джерело живлення: літієві акумулятори 18650 з понижуючим стабілізатором (HT7333).
- Операційний підсилювач (наприклад, MCP6L92) для узгодження аналогового сигналу з рівнем АЦП.
- Захисні діоди, пасивні елементи, RC-фільтри.

Енкодери з кнопками відіграють роль основного елементу керування інтерфейсом користувача. Це пристрої, що дозволяють зчитувати поворот (у праву чи ліву сторону) та натискання на вісь. У проєкті застосовано інкрементальні енкодери, які генерують два квадратурні сигнали (А і В), з яких мікроконтролер визначає напрямок обертання. Обробка сигналів здійснюється через GPIO з перериванням або таймером у режимі енкодера. Натискання використовується для вибору пунктів меню або підтвердження дії [5]. Це забезпечує зручну та інтуїтивно зрозумілу взаємодію користувача із системою. Дисплей використовується для виведення графічної інформації, зокрема графіків сигналу, параметрів вимірювань та меню. Працює за протоколом SPI, з роздільною здатністю 128x160 пікселів, це дозволяє передавати дані з високою швидкістю при мінімальному використанні виводів мікроконтролера. Для роботи з дисплеєм використовуються бібліотеки з підтримкою контролера ST7735. Візуалізація сигналу здійснюється через буфер, у якому формуються координати кривої, після чого зображення виводиться на екран. Колірна гама (16-біт) дозволяє зручно розрізняти

					КС КРБ 123.222.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

графічні елементи та текстові позначення. Висока контрастність і компактні розміри дисплея роблять його ідеальним для портативного осцилографа.

Таким чином, вибір STM32F103C8T6 як базового мікроконтролера є найкращим через його відповідність технічним вимогам проєкту, простотою програмування, наявністю великої кількості ресурсів, прикладів та активної спільноти розробників. Такий вибір дозволяє створити компактну, енергоефективну та доступну комп'ютеризовану систему візуалізації аналогових сигналів, адаптовану для використання у портативних приладах, навчальних осцилографах або системах технічної діагностики.

2.3 Обґрунтування вибору програмного забезпечення проєктованого комп'ютерного засобу

Розробка комп'ютеризованої системи візуалізації аналогових сигналів передбачає використання сучасного програмного забезпечення для реалізації як електричної, так і програмної частин проєкту. Основними критеріями при виборі програмного забезпечення були: підтримка платформи STM32, зручність роботи з апаратною частиною, можливості відлагодження, наявність бібліотек для периферійних модулів, а також інструменти для візуалізації та обміну даними.

У якості основного середовища для розробки програмного коду обрано STM32CubeIDE – офіційне середовище від STMicroelectronics, що забезпечує повну підтримку мікроконтролерів STM32. Воно об'єднує в собі функціонал STM32CubeMX (для конфігурації периферії та генерації ініціалізаційного коду) та Eclipse-подібного середовища для розробки на C/C++ з вбудованим компілятором GCC та можливістю налагодження через ST-Link.

Для створення електричної принципової схеми використано Altium Designer – професійну CAD-систему для проєктування електроніки. Altium

					КС КРБ 123.222.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

надає широкі можливості для моделювання, перевірки цілісності сигналів, трасування багатошарових плат та створення бібліотек компонентів.

Передача та візуалізація даних через USB. Для налагодження з'єднання через USB-UART інтерфейс, а також для перевірки правильності передачі даних у текстовому форматі застосовано RealTerm – спеціалізований термінальний емулятор для роботи з COM-портами. Цей інструмент дозволив зручно приймати та виводити значення з АЦП у режимі реального часу, а також тестувати протокол обміну між мікроконтролером та ПК [6].

Вибір STM32CubeIDE, Altium Designer та RealTerm як основних інструментів для розробки та налагодження системи забезпечив гнучкість, швидкість та зручність реалізації програмної та електричної частин проєкту. Сукупність цих інструментів дозволила ефективно реалізувати повний цикл розробки – від проєктування принципової схеми до налагодження програмного коду та передачі результатів вимірювань на зовнішній пристрій.

2.4 Опис схеми електрично-принципової

На зображеній схемі (рис. 2.2) представлено функціональне підключення основних апаратних модулів до мікроконтролера STM32F103C8T6, що є центральним керуючим елементом вбудованої системи цифрового осцилографа. Конфігурація охоплює інтеграцію з периферійними пристроями, такими як енкодери для керування меню, кольоровий TFT-дисплей на базі контролера ST7735, стабілізатор напруги HT7333, а також зарядний модуль TP4056, призначений для живлення і захисту акумуляторного джерела енергії. Дисплей з діагоналлю 1.8 дюйма та роздільною здатністю 160×128 пікселів використовує послідовний SPI-інтерфейс для обміну даними з мікроконтролером.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

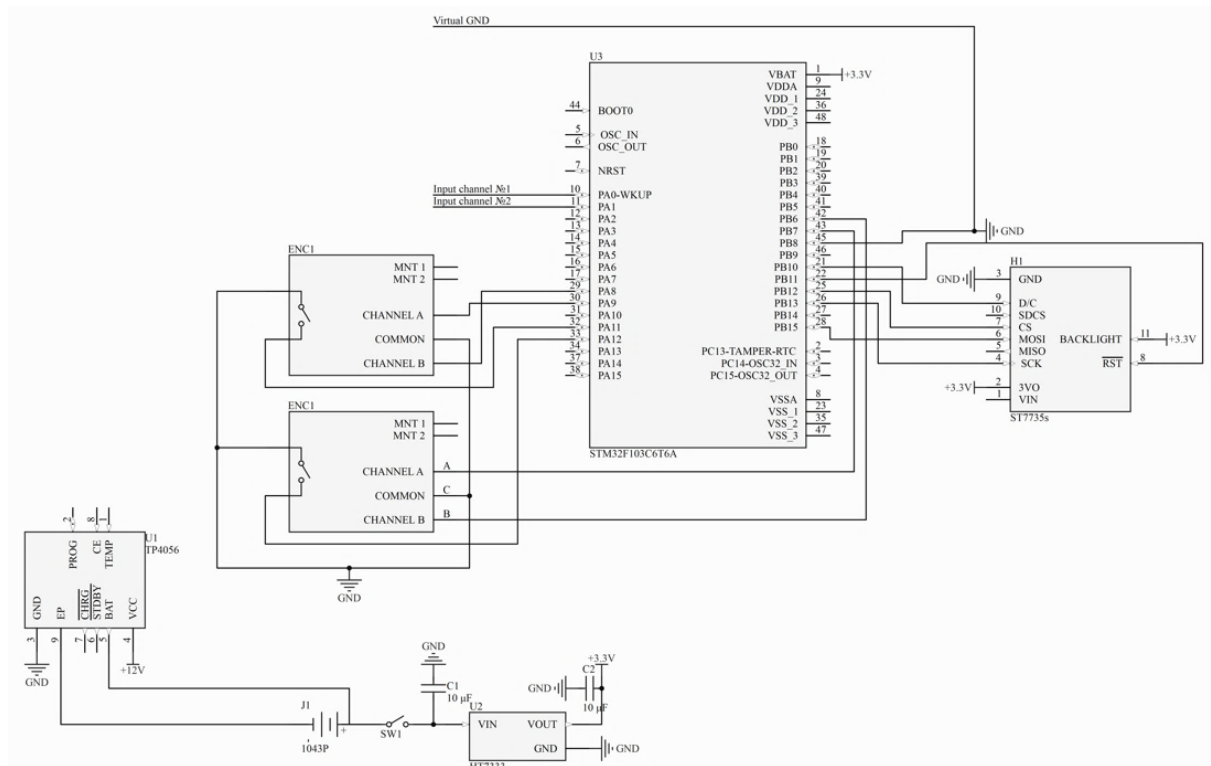


Рисунок 2.2 – Схема електрична принципова системи візуалізації

Передача відбувається апаратно через модуль SPI2 STM32F103, що забезпечує високу швидкість оновлення екрана, необхідну для безперервного відображення осцилограм. Для коректної роботи інтерфейсу залучено наступні виводи:

- SCL (PB13) – тактова лінія (Serial Clock);
- MOSI (PB15) – лінія передачі даних (Master Out Slave In);
- CS (PB12) – сигнал вибору пристрою (Chip Select);
- DC (PB14) – перемикання між передачею команд і даних (Data/Command);

Живлення всієї системи забезпечується за допомогою двох послідовно з'єднаних Li-Ion акумуляторів типу 18650, що мають загальну номінальну напругу до 8.4 В. Ця напруга знижується до рівня 3.3 В за допомогою стабілізатора HT7333, який є малопотужним лінійним регулятором низького струму споживання. Стабілізатор живить мікроконтролер STM32F103C8T6, TFT-дисплей та енкодери, забезпечуючи при цьому стабільну напругу логіки без надмірних пульсацій, що критично для надійної роботи АЦП та чутливих

до шумів периферійних пристроїв. Окрему увагу варто приділити модулю TP4056, який слугує для заряджання акумуляторів через інтерфейс microUSB. Модуль обладнаний інтегрованим захистом від перевантаження по струму, надлишкової напруги та глибокого розряду акумуляторів, що дозволяє використовувати його у переносних пристроях з автономним живленням. Виходи модуля підключено до стабілізатора, забезпечуючи плавну подачу живлення на всю систему.

Порт програмування реалізовано через стандартний інтерфейс SWD (Serial Wire Debug), що включає сигнали SWDIO, SWCLK, NRST та GND. Це дозволяє ефективно налагоджувати мікропрограмне забезпечення, проводити діагностику та оновлення прошивки без необхідності вилучення мікроконтролера з плати [7].

Для взаємодії користувача з пристроєм передбачено два енкодери типу АМТ, які з'єднані з цифровими входами STM32F103C8T6. Їхні сигнали зчитуються через GPIO-виводи, які налаштовані як входи з підтягуванням. Це дозволяє точно визначати положення енкодера, обертання вліво або вправо, та реагувати на натискання кнопки енкодера. Енкодери використовуються для навігації по меню осцилографа, зміни режимів відображення, вибору каналів, масштабування сигналу тощо.

Таким чином, розглянута схема є прикладом ефективної реалізації низькопотужного автономного пристрою для обробки аналогових сигналів у реальному часі. Усі компоненти об'єднані в єдину логічну структуру з оптимізованим енергоспоживанням, високою швидкістю та гнучкою інтеграцією периферії. Завдяки цьому вона може слугувати базою для створення портативних приладів на кшталт осцилографів, аналізаторів сигналів або систем відображення параметрів убудованих обчислювальних платформ.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5 Опис блок-схеми алгоритму

Представлена блок-схема (рис 2.3) описує логіку функціонування комп'ютеризованої системи візуалізації аналогових сигналів.

На першому етапі, після запуску, відбувається ініціалізація елементів системи, що включає налаштування портів вводу/виводу, внутрішніх модулів (ADC, таймерів, DMA, дисплея тощо), а також перевірку наявності живлення і готовності периферії до роботи. Далі виконується конфігурація системного годинника, що визначає частоту роботи мікроконтролера та усіх його внутрішніх компонентів.

Після цього здійснюється калібрування аналогово-цифрових перетворювачів (ADC1 та ADC2) для зменшення похибок при вимірюванні сигналів. Калібрування дозволяє компенсувати внутрішні зміщення та забезпечити достовірне оцифрування аналогових вхідних сигналів.

Наступний крок перевірка, чи встановлено зв'язок з зовнішніми компонентами (наприклад, з дисплеєм або модулем передавання даних). У разі, якщо з'єднання не встановлено, система переходить у режим очікування і періодично повторює перевірку. Якщо зв'язок успішно налагоджено, здійснюється запуск таймера TIM3, який використовується для синхронізації зчитування даних з АЦП з заданою частотою.

У рамках основного циклу система зчитує дані з АЦП, після чого копіює їх у буфер для подальшої обробки. Буферизація дає змогу забезпечити безперервний прийом сигналу без втрати даних навіть при наявності часових затримок у виведенні або аналізі.

Далі виконується корекція значень, яка може включати нормалізацію, масштабування або фільтрацію сигналів для усунення шумів і приведення даних до потрібного діапазону.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

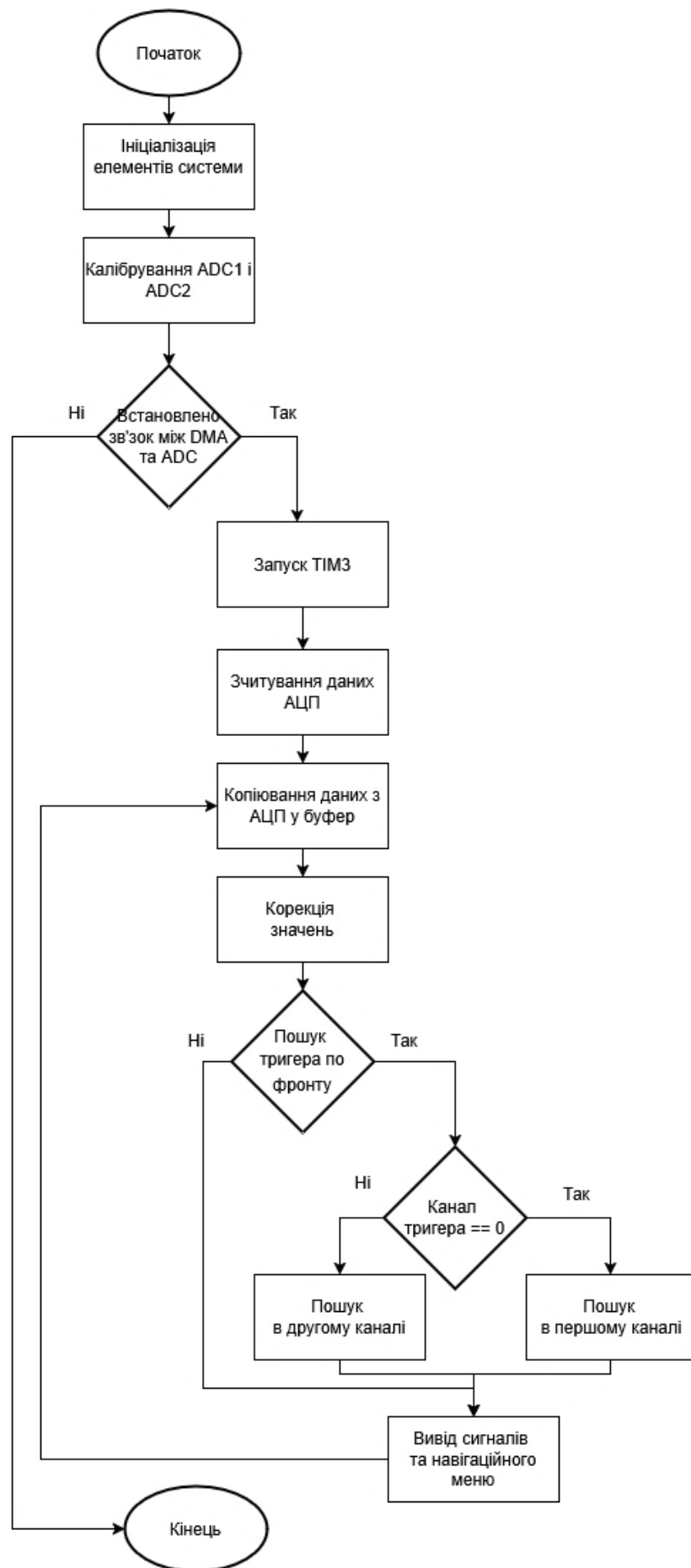


Рисунок 2.3 – Блок-схема алгоритму роботи програми

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.222.00.00 ПЗ

Арк.

34

Після цього здійснюється пошук тригера, тобто визначення моменту часу або умов, за яких відбувається фіксація події або збурення в сигналі. Це може бути, наприклад, фронт імпульсу, перетин певного порогу чи інша умова.

Якщо тригер знайдено, система перевіряє, у якому з каналів (першому чи другому) він відбувся. В залежності від цього виконується аналіз відповідного каналу – або першого, або другого.

Завершальним етапом є відображення меню та результатів обробки на дисплеї або передача їх до зовнішнього інтерфейсу. Це дозволяє користувачу переглядати зібрану інформацію у зручному форматі та взаємодіяти з системою через меню.

Описаний алгоритм є типовим для систем цифрового збору даних у реальному часі. Його особливістю є структурована організація етапів підготовки, збору та аналізу даних, що дозволяє забезпечити надійність та точність вимірювань. Такий підхід є ефективним у задачах осцилографії, аналізу сигналів та побудови систем реального часу.

2.6 Опис макета системи

На схемі (рис. 2.4) зображено елементи системи, дана схема демонструє взаємодію між різними компонентами.

Джерелом живлення виступають два під'єднані паралельно літій-іонні акумулятори формату 18650. Він служить основним джерелом енергії для всієї системи. Забезпечують постійну напругу 8.4 В.

Модуль зарядки та захисту акумулятора TP4056. Цей модуль підключений до акумулятора виводи OUT+ та OUT- забезпечують його зарядку через порт Micro USB IN+ та IN-. Також він інтегрує схеми захисту від перезаряду, перерозряду та перевантаження струму, що є критично

					КС КРБ 123.222.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

важливим для безпечної та довготривалої експлуатації літій-іонних акумуляторів.

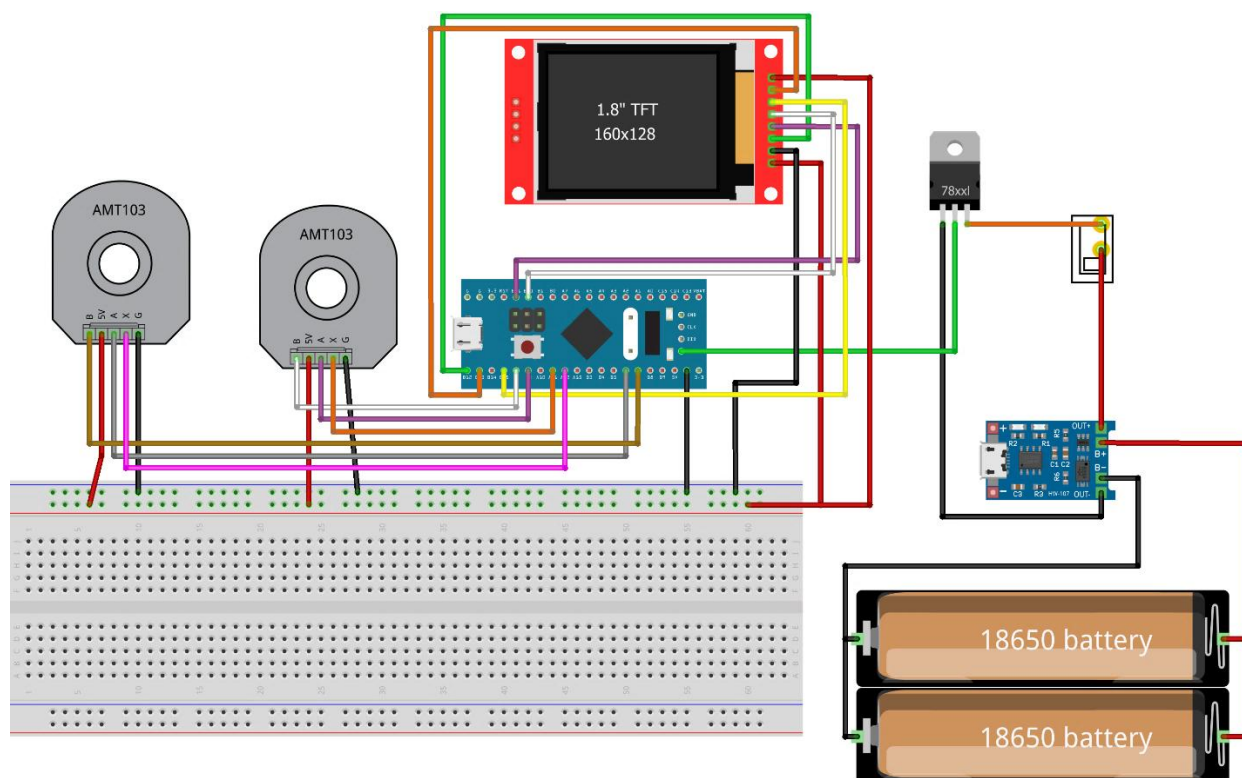


Рисунок 2.4 – Схема тестового макеиу системи візуалізації

Лінійний стабілізатор напруги HT7333: Це мікросхема, яка перетворює вхідну напругу від модуля зарядки або безпосередньо від акумулятора після підвищення, на схемі прямого зв'язку з OUT+ модуля немає, адже він здійснюється через кнопку. HT7333 стабілізує вихідну напругу на рівні 3.3 В. Цей компонент є важливим для живлення мікроконтролера та інших компонентів, що вимагають стабілізованої напруги. На схемі вихід HT7333 підключений до живлення контролера STM32F103 який в свою чергу живить макетну плату та решту елементів.

На схемі присутній мікроконтролер STM32F103C8T6, який програмується для виконання поставлених завдань. Плата живиться від стабілізованої напруги 3.3 В, що надходить від HT7333. Даний мікроконтролер виступає в ролі головного керуючого елемента системи, в ньому здійснюються

всі операції. Мікроконтролер має численні цифрові піни, які використовуються для взаємодії з іншими компонентами.

Керування здійснюється через два інкрементальні енкодери, які перетворюють обертовий рух у послідовність цифрових імпульсів. Кожен енкодер має п'ять виводів: GND (земля), VCC (живлення), A(сигнал повороту в одну сторону), B(сигнал повороту в іншу сторону) та Key(сигнал кнопки). Енкодери живляться від шини 3.3 В макетної плати. Вихідні сигнали обох енкодерів підключені до цифрових входів мікроконтролера. Мікроконтролер інтерпретує імпульси від енкодерів для визначення напрямку оберт.

TFT-дисплей з роздільною здатністю 160x128 пікселів. Використовується для відображення інформації наприклад: даних з енкодерів, стану системи тощо.

TFT-дисплеї такого типу спілкуються з мікроконтролером за допомогою послідовних інтерфейсів, таких як SPI (Serial Peripheral Interface). На схемі видно кілька ліній даних та керування, що з'єднують дисплей з мікроконтролером (D8, D9, D10, D11, D13). Піни дисплея позначені як LED, SCK, SDA, A0, RST, CS, GND, VCC.

Живлення дисплея відбувається через 3.3 В. На схемі lbcgktq підключений до “Землі” макетної плати та до шини живлення(+3.3В) макетної плати.

SCK (Serial Clock), SDA (Serial Data/MOSI), A0 (Data/Command select), RST (Reset), CS (Chip Select): Це типові піни для SPI-інтерфейсу, що дозволяють мікроконтролеру передавати дані та команди на дисплей.

LED вивід для керування підсвічуванням дисплея [8].

					КС КРБ 123.222.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ АНАЛОГОВИХ СИГНАЛІВ

3.1 Ініціалізація мікроконтролера та периферії

У процесі розробки системи на базі STM32F103C8T6 ключовим етапом є налаштування внутрішніх модулів мікроконтролера для забезпечення коректної роботи периферійних пристроїв, зокрема: дисплея ST7735, енкодерів, аналого-цифрового перетворювача, UART та таймерів. Конфігурація виконувалася за допомогою бібліотеки STM32 LL (Low-Layer), що забезпечує більшу продуктивність і контроль над апаратними ресурсами, порівняно з HAL [9].

Для коректної роботи всіх модулів системи необхідне стабільне системне тактування. В даній розробці використано зовнішній кварцовий резонатор на 8 МГц, який множиться за допомогою PLL до 72 МГц – максимально можливої частоти ядра STM32F103 (рис 3.1).

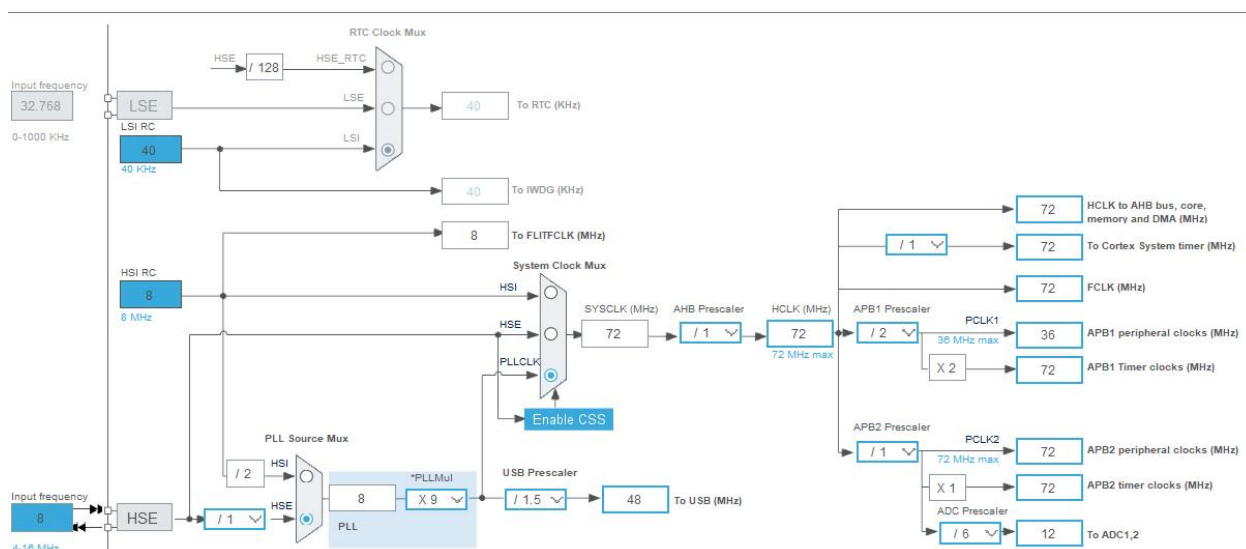


Рисунок 3.1 – Налаштування тактової частоти мікроконтролера

					КС КРБ 123.222.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Курцеба Т.В.			Розробка та тестування системи візуалізації аналогових синалів			Лім.	Арк.	Аркушів	
Перевір.		Лещишин Ю.З.								38	
Реценз.		Пастух О.А						ТНТУ, каф. КС, гр. СІ-42			
Н. Контр.		Тиш Є.В.									
Затверд.		Осухівська Г.М.									

Всі піни, які використовуються для керування дисплеєм, енкодерами, кнопками та зчитуванням аналогових сигналів, налаштовуються як вихідні або вхідні відповідно до призначення (рис 3.2). Для зв'язку з TFT-дисплеєм використовується інтерфейс SPI1, що працює в режимі Master. Виводи SCK, MOSI, CS, DC, RESET задаються вручну відповідно до підключення згідно схеми.

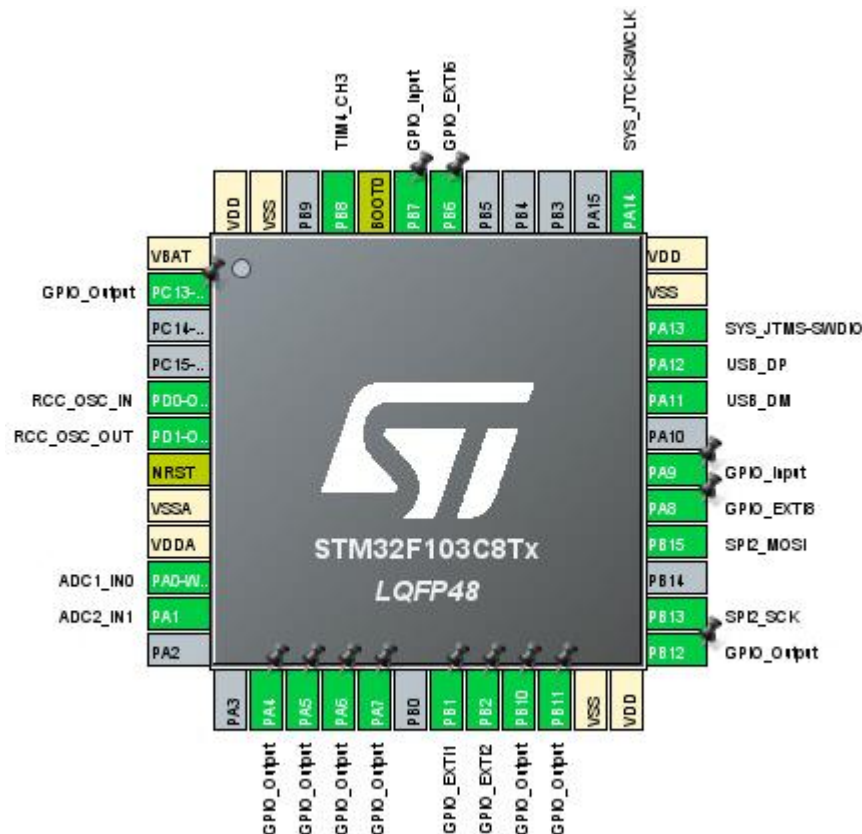


Рисунок 3.2 – Налаштування виходів мікроконтролера

Для зчитування аналогового сигналу з вхідного каналу використовується ADC1, налаштований на неперервне перетворення з DMA або ручним зчитуванням. ADC2 працює по тому ж принципу. UART використовується для передачі даних по USB CDC використовуючи USB_DM та USB_DP на портах PA11 та PA12 відповідно. Таймери (TIM2, TIM3) використовуються для опитування енкодерів та побудови часової шкали сигналу [10].

Основою для захоплення аналогових сигналів у даній системі є аналогово-цифровий перетворювач (ADC1) мікроконтролера STM32F103C8T6. У проекті використовується режим безперервного сканування декількох каналів, що дозволяє проводити оцифрування сигналів з двох аналогових входів одночасно – для кожного з каналів осцилографа (CH1 та CH2).

3.2 Апаратна конфігурація АЦП та передача в DMA

АЦП налаштовано у режим сканування (Scan Mode) з перетворенням у безперервному режимі (Continuous Conversion Mode), де відразу після завершення оцифрування одного каналу автоматично починається перетворення наступного. Це дозволяє здійснювати чергове зчитування сигналів з двох каналів без участі програмного коду на рівні основного циклу. Для підвищення продуктивності системи та зниження навантаження на ядро мікроконтролера, оцифровані значення передаються безпосередньо у пам'ять за допомогою прямого доступу до пам'яті (DMA).

```
LL_DMA_SetPeriphAddress(DMA1, LL_DMA_CHANNEL_1, (uint32_t) &ADC1->DR);
LL_DMA_SetMemoryAddress(DMA1, LL_DMA_CHANNEL_1, (uint32_t) &adc_buffer);
LL_DMA_SetDataLength(DMA1, LL_DMA_CHANNEL_1, BUFFERSIZE);
LL_DMA_EnableIT_TC(DMA1, LL_DMA_CHANNEL_1); //Enable IRQ at transfer complete
LL_DMA_EnableChannel(DMA1, LL_DMA_CHANNEL_1);
```

Рисунок 3.3 Лістинг коду налаштування DMA

SetPeriphAddress(...) Встановлює адресу периферійного пристрою а саме регістр даних ADC1->DR. Саме звідти будуть зчитуватися результати ADC.

SetMemoryAddress(...) Встановлює адресу буфера в оперативній пам'яті adc_buffer, куди DMA буде зберігати значення з ADC.

SetDataLength(...) Задає довжину переданих даних тобто кількість елементів, які будуть перенесені з ADC в буфер.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

EnableIT_TC(...) Увімкнення переривання по завершенню передачі – коли DMA завершить заповнення буфера, буде згенеровано IRQ що дозволяє виконати обробку даних в відповідній функції обробнику.

EnableChannel(...) Активує сам DMA-канал і запускає передачу.

DMA автоматично переміщує результати перетворення з регістра результату АЦП у спеціально виділений буфер у пам'яті, без потреби втручання центрального процесора (рис 3.3). Це дозволяє досягти високої частоти дискретизації, що критично важливо для правильної реконструкції швидкоплинних аналогових сигналів. Відбувається це завдяки встановленню пріоритетності для DMA в пункті налаштування ADC1 [11].

Для зберігання результатів оцифрування створено масив типу uint32_t далі ці данні передаються з DMA буфера в 2 масиви uint16_t які мають додаткове явне приведення типів на випадок виходу значень за встановлені межі.

У цьому ж рядку відбувається поділ більшого 32 числового значення масиву даних з значеннями АЦП на 2 менші 16-бітні масиви, що в результаті утворює 2 канали даних (рис 3.4). Тобто з 32-бітного слова добуваються два 16-бітних значення, які далі проходять корекцію. Сам осцилограф не містить додаткових дільників напруги адже розрахований на роботу з низькими напругами, в результаті такого рішення використовуються лише зовнішні атенюатори, ті які містяться в середині щупів. В зв'язку з цим необхідно робити поправку на коефіцієнт ділення, це відбувається використовуючи змінну ATTENUATOR_MULTIPLY.

```
trace_1_buffer[i] = (uint16_t) ((adc_buffer[i] >> 21)*ATTENUATOR_MULTIPLY); //separate both channels
trace_2_buffer[i] = (uint16_t) (((adc_buffer[i] & 0x0000FFFF) >> 5)*ATTENUATOR_MULTIPLY);
```

Рисунок 3.4 Лістинг коду передачі даних з DMA в буфери

У налаштуваннях DMA використовується режим циклічної роботи (circular mode), завдяки якому після заповнення буфера нові дані автоматично перезаписують найстаріші, забезпечуючи постійне оновлення інформації у

					КС КРБ 123.222.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

режимі реального часу. Хоча ADC працює у безперервному режимі, обробка вибірок здійснюється періодично за допомогою таймера TIM2. Таймер генерує переривання з фіксованою частотою, що визначає частоту оновлення екрана або частоту передачі даних через UART.

У перериванні TIM2_IRQHandler виконується фільтрація та нормалізація сигналу – зменшення шуму та приведення до шкали напруги, яку можна відобразити на TFT-дисплеї.

Копіювання даних з adc_buffer у робочий буфер draw_buffer, який використовується для відображення сигналу.

Таким чином, основне ядро системи не блокується ні зчитуванням, ні обробкою сигналів – завдяки апаратному збиранню через DMA та програмному обслуговуванню у перериваннях з пріоритетами.

3.3 Керування через енкодери

У системі реалізовано підтримку двох енкодерів, які забезпечують інтуїтивне керування параметрами осцилографа. Вони використовуються для:

- вибору масштабу по осі Y (amplitude);
- вибору часової розгортки (time base);
- навігації по меню користувача;
- перемикання коефіцієнта ослаблення (наприклад, $\times 1$ або $\times 10$).
- налаштування тригерного режиму (рівень спрацювання, канал, тип фронту).

Кожен енкодер має два сигнальні виводи (А та В) та одну кнопку. Для обробки енкодерних імпульсів використовуються апаратні можливості таймерів STM32F103 (рис 3.5).

					КС КРБ 123.222.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

```
while (LL_TIM_IsActiveFlag_UPDATE(TIM2)) LL_TIM_ClearFlag_UPDATE(TIM2);

rotary_1_updown = 0;
rotary_2_updown = 0;

rotary_1_pinA <=<= 1;
rotary_1_pinB <=<= 1;
rotary_2_pinA <=<= 1;
rotary_2_pinB <=<= 1;
```

Рисунок 3.5 Лістинг коду підготовки енкодерів до роботи

У циклі while відбувається очищення прапора, якщо він встановлений, адже може бути кілька подій.

Далі відбувається ініціалізація змінних різниці.

rotary_1_pinA <=<= 1 відповідає за зсув бітів для формування історії сигналу.

```
if (LL_GPIO_IsInputPinSet(GPIOA, LL_GPIO_PIN_8) == 0) rotary_1_pinA++;
if (LL_GPIO_IsInputPinSet(GPIOA, LL_GPIO_PIN_9) == 0) rotary_1_pinB++;
if (LL_GPIO_IsInputPinSet(GPIOB, LL_GPIO_PIN_6) == 0) rotary_2_pinA++;
if (LL_GPIO_IsInputPinSet(GPIOB, LL_GPIO_PIN_7) == 0) rotary_2_pinB++;

if (rotary_1_pinA == 0x80)
{
    if (rotary_1_pinB == 0x00) rotary_1_updown = -1;
    if (rotary_1_pinB == 0xFF) rotary_1_updown = 1;
}

if (rotary_2_pinA == 0x80)
{
    if (rotary_2_pinB == 0x00) rotary_2_updown = -1;
    if (rotary_2_pinB == 0xFF) rotary_2_updown = 1;
}
```

Рисунок 3.6 Лістинг коду обробки обертання енкодера

Наступним етапом необхідно зчитати новий стан кожного енкодера що відповідатиме вхідним сигналам А і В.

Після зчитування нових значень відбувається порівняння з шаблоном що дозволяє визначити напрямок повертання енкодерів. Таким чином в залежності від активного режиму відбувається зміна перемикання коефіцієнтів атенюації, напруга на каналах, інтервал розгортки (рис 3.6).

Після кожного читання, значення лічильника зазвичай скидається. Кнопка енкодера підключена до GPIO з налаштуванням на зовнішнє переривання (EXTI). У перериванні виконується перемикання режимів або відкриття підменю.

```
#define TIMEBASE_MODE      0
#define TRIGGER_LEVEL_MODE 1
#define TRIGGER_CHANNEL_MODE 2
```

Рисунок 3.7 – Лістинг режимів одного з енкодерів

На початку програми визначено режими роботи (рис 3.7), кожен режим має своє числове значення. При натисканні кнопки викликається переривання, це переривання збільшує лічильник який переходить між числовими значеннями режимів, таким чином якщо встановлено режим TIMEBASE_MODE то натискання кнопки встановить режим TRIGGER_LEVEL_MODE, подальші зміни параметрів в режимах відбуваються повертанням енкодерів [12].

Окрім відображення сигналу, осцилограф дозволяє задавати умови запуску сигналу за допомогою тригера. Його головна функція – знайти точку у масиві відцифрованих значень, у якій починається сигнал, і від неї вивести графік на дисплей. Це дозволяє отримати стабільне зображення періодичного сигналу. Параметри тригера встановлюються за допомогою енкодера:

- Рівень тригера – визначає амплітуду, при досягненні якої відбувається зупинка захоплення сигналу.
- Канал тригера – вибирається канал CH1 або CH2.
- Тип спрацювання – по фронту (на підйомі ↑ або спаданню ↓).

Отже як працює цей код (рис 3.8), trigger_level – це рівень або ж значення, при якому має бути запущено відображення сигналу.

```

i = 1; //find trigger_point
if (trigger_channel == 0) //with channel 1
{
    while ((trace_1_buffer[i - 1] < trigger_level) || (trace_1_buffer[i] >= trigger_level)) i++;
}
else //with channel 2
{
    while ((trace_2_buffer[i - 1] < trigger_level) || (trace_2_buffer[i] >= trigger_level)) i++;
}
if (i > X_MAX) trigger_point = 1; //if the triggerpoint isn't found
else trigger_point = i; //show data from beginning of arrays

```

Рисунок 3.8 – Лістинг реалізації тригера по фронту

Змінна i – індекс, з якого почнеться виведення графіку, тобто зсув. У циклах `while` відбувається перевірка на перехід від меншого до більшого відносно `trigger_level` таким чином реалізується тригер по фронту. Графіки малюються з зсувом, який відповідає знайденому `trigger_point`, а на екрані малюється горизонтальна лінія, яка показує, де на екрані встановлений рівень тригера. Якщо не знайдено жодного місця, де сигнал перетинає `trigger_level` по зростанню, то `trigger_point` встановлюється в 1, і графік просто виводиться з початку буфера [13].

3.4 Корекція значень

Після отримання даних з ADC1 і ADC2 в масив `adc_buffer`, дані обробляються у циклі (рис 3.9).

```

//corrections because of the 3.3V reference of the ADCs
temp1 = trace_1_buffer[i];
trace_1_buffer[i] = (temp1 + (temp1 >> 2) - 16);
if (trace_1_buffer[i] > 9999) trace_1_buffer[i] = 0; // >9999 means negative
if (trace_1_buffer[i] > 127) trace_1_buffer[i] = 127;

temp2 = trace_2_buffer[i];
trace_2_buffer[i] = (temp2 + (temp2 >> 2) - 16);
if (trace_2_buffer[i] > 9999) trace_2_buffer[i] = 0;
if (trace_2_buffer[i] > 127) trace_2_buffer[i] = 127;

```

Рисунок 3.9 – Лістинг коду корекції

Тут відбувається попередня корекція амплітуди сигналу, отриманого з АЦП, для обох каналів `trace_1_buffer` і `trace_2_buffer`.

$\text{trace_1_buffer}[i] = (\text{temp1} + (\text{temp1} \gg 2) - 16)$ Загалом, це наближена

					КС КРБ 123.222.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

лінійна трансформація, що компенсує: вплив опорної напруги 3.3 В, зміщення каналу, похибки масштабування.

$\text{temp1} \gg 2$ – це арифметичне зменшення значення temp1 на чверть (поділ на 4), тобто відносна поправка це виступатиме як, компенсація температурного дрейфу або нелінійності.

Додавання ($\text{temp1} + (\text{temp1} \gg 2)$) трохи підсилює сигнал (приблизно на 25%), тобто це корекція підсилення, для компенсації втрати амплітуди в лінійці вимірювання опорних 3.3V АЦП [14].

“-16” – це зміщення вниз на 16 одиниць, Умовно кажучи – зсув нуля вниз, таким чином усувається зміщення сигнального рівня.

Далі відбувається обмеження можливих значень обчислень. Значення 9999 тут використовується як означення помилки чи переповнення. Тобто якщо після обчислень значення стало дуже великим це означає, що воно вийшло за межі або стало від’ємне у вигляді переповнення беззнакового типу. Наприклад, якщо $\text{trace_1_buffer}[i]$ – це uint8_t (8-біт), і в наслідок обчислень з’явилося значення -5, то воно стане ≈ 251 , або навіть більше у випадку інших типів – і тоді відбудеться переповнення >9999 . Тому для уникнення ситуація з переповненням некоректні значення скидаються в 0, аби не вносити завади при виведенні графіка. Наступним етапом обмеження зверху, бо 127 – максимальне значення для 8-бітного сигналу, який буде виводитись на екран. Це вже обмеження для того щоб значення не виходили за розмір екрану [15].

3.5 Особливості роботи з дисплеєм

У даному проєкті застосовано TFT-дисплей із роздільною здатністю 128×160 пікселів, який керується графічним контролером ST7735. Цей чип широко використовується в низьковольтних пристроях завдяки своїй високій енергоефективності, підтримці кольорової палітри 16 біт (65 536 кольорів) та сумісності з інтерфейсами SPI.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Драйвер ST7735 є інтегрованим контролером з вбудованою відеопам'яттю (VRAM), яка забезпечує збереження зображення без постійного оновлення з боку мікроконтролера. Він підтримує обробку пікселів у форматах RGB565 та RGB666, дозволяючи досягти високої якості графічного представлення при відносно низькому навантаженні на центральний процесор системи.

Крім стандартного режиму зображення, ST7735 дозволяє реалізувати режим горизонтального або вертикального перегортання, інверсії кольору та прокрутки екрана, що може бути корисним для побудови прокручуваних графіків або меню користувача. Також доступна функція часткового оновлення екрану, яка зменшує час виводу та енергоспоживання в системах з обмеженими ресурсами.

Ще однією перевагою контролера ST7735 є підтримка функції апаратного скидання, яка забезпечує стабільну ініціалізацію дисплея після ввімкнення живлення. Це особливо важливо у пристроях з автономним живленням, де можлива нестабільність напруги в момент запуску системи. Вивід інформації здійснюється за допомогою попередньо визначених команд (наприклад, ST7735_RAMWR, ST7735_CASET, ST7735_RASET), які визначають область малювання та передають безпосередньо масив пікселів у відеопам'ять дисплея.

Взаємодія з контролером здійснюється через послідовний інтерфейс SPI (Serial Peripheral Interface), що забезпечує просту фізичну реалізацію з'єднання з мікроконтролером типу STM32.

У системах цифрової обробки та візуалізації аналогових сигналів, інтерфейс SPI використовується для передачі даних на дисплей з метою їх подальшого відображення. SPI – це синхронний послідовний протокол обміну даними, що забезпечує ефективну взаємодію центрального процесора з периферійними компонентами, а саме з графічним драйвером ST7735, який керує TFT-дисплеєм. SPI характеризується передачею даних у повнодуплексному режимі, що дозволяє одночасно надсилати та приймати

					КС КРБ 123.222.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

інформацію, забезпечуючи високу пропускну здатність і мінімальні затримки що критично важливо для оперативного відображення динамічних сигналів у реальному часі. Архітектура SPI побудована на чотирьох основних сигнальних лініях:

- MOSI (Master Out Slave In) – канал передачі даних від мікроконтролера (ведучого пристрою) до периферії (веденого пристрою, ST7735);
- MISO (Master In Slave Out) – зворотній канал обміну, який не використовується у даному випадку;
- SCLK (Serial Clock) – лінія синхронізації, яка генерується ведучим пристроєм і задає часову структуру передавання кожного біта;
- CS або SS (Chip Select / Slave Select) – сигнальна лінія, що активує потрібний периферійний пристрій.

Передача даних через SPI здійснюється синхронно, тобто кожна одиниця інформації супроводжується відповідним імпульсом на лінії SCLK. Це дозволяє точно керувати моментами зміни логічного стану на лінії MOSI. У системах візуалізації аналогових сигналів такий принцип синхронізації забезпечує точне й стабільне оновлення вмісту дисплея, що особливо важливо при виведенні осцилограм з високою частотою оновлення.

Ініціація передавання відбувається за рахунок встановлення низького логічного рівня на сигнальній лінії CS відповідного периферійного пристрою. Після цього мікроконтролер починає передавання команд і графічних даних до контролера дисплея ST7735 через регістр зсуву. Як правило, передавання здійснюється байтами (8 біт), хоча деякі реалізації SPI можуть підтримувати інші довжини кадру. Під час кожного імпульсу синхронізації відбувається зчитування та передавання по одному біту з кожного пристрою – від ведучого до веденого по лінії MOSI та, за потреби, від веденого до ведучого по MISO.

У схемі побудови осцилографа, SPI забезпечує швидке надсилання векторних команд тобто малювання ліній та пікселів до ST7735, що дозволяє ефективно відображати відформатовані дані з АЦП на TFT-дисплеї. Завдяки

					КС КРБ 123.222.00.00 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

чіткій синхронізації й апаратній простоті реалізації, SPI зручний для програмної емуляції і дозволяє адаптувати частоту передавання відповідно до характеристик обох пристроїв без втрати узгодженості у переданих даних.

Також варто зазначити, що у структурі SPI-передавання часто використовується лічильник бітів для визначення закінчення кадру даних. У мікроконтролері та графічному контролері він оновлюється автоматично при кожному імпульсі тактового сигналу і, зазвичай, скидається після завершення сесії зв'язку або деактивації лінії SS [16].

Завдяки високій швидкодії, низькому рівню апаратних затрат та можливості точного контролю над передаванням, SPI став ключовим компонентом для забезпечення візуального зворотного зв'язку в системах, орієнтованих на моніторинг та аналіз аналогових сигналів у реальному часі.

У процесі відображення оцифрованих сигналів графічний буфер дисплея оновлюється побітно координати ліній формуються з ADC-даних, оброблених у реальному часі, і передаються в драйвер через SPI шляхом побудови вертикальних або похилих ліній, які представляють миттєві значення сигналу (рис 3.10).

```
for (i = 1; i < X_MAX; i++)
{
    j = i + 1;

    x1 = i + previous_trigger_point;
    x2 = x1 + 1;
    x3 = i + trigger_point;
    x4 = x3 + 1;

    y1 = erase_1_buffer[x1];
    y2 = erase_1_buffer[x2];
    y3 = trace_1_buffer[x3];
    y4 = trace_1_buffer[x4];

    z1 = erase_2_buffer[x1];
    z2 = erase_2_buffer[x2];
    z3 = trace_2_buffer[x3];
    z4 = trace_2_buffer[x4];

    ST7735_Line(i, y1, j, y2, ST7735_BLACK);           //remove old trace
    ST7735_Line(i, z1, j, z2, ST7735_BLACK);           //remove old trace

    ST7735_Line(i, y3, j, y4, ST7735_GREEN);           //show new trace
    ST7735_Line(i, z3, j, z4, ST7735_MAGENTA);         //show new trace
}
```

Рисунок 3.10 – Лістинг коду виведення на дисплей

					КС КРБ 123.222.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

У цьому фрагменті коду реалізовано виведення графіка сигналу на TFT-дисплей із використанням функції ST7735_Line(), яка малює лінії між двома точками. Це є основною візуалізацією сигналу в осцилографі.

Цикл проходить по осі X дисплея (від 1 до максимальної ширини графіка X_MAX). Кожна ітерація відповідає одному пікселю по горизонталі, на якому малюється відрізок сигналу (лінія з точки i до i+1). Потім відбувається розрахунок координат де trigger_point – це зсув, від якого починається виведення сигналу (визначається тригером), previous_trigger_point – попередній зсув (для стирання старого зображення), x1, x2 – координати X для старого сигналу (з минулого буфера), x3, x4 – координати X для нового сигналу тобто поточного буфера.

Далі відбувається Зчитування Y-координат які відповідають за значення сигналу.

trace_1_buffer, trace_2_buffer – основні буфери каналів, у яких зберігаються значення сигналів для осцилографа (перетворені з ADC у координати на дисплеї).

erase_1_buffer, erase_2_buffer – копії попереднього зображення, які використовуються, щоб стерти лінії попереднього кадру.

При передаванні в функцію ST7735_Line(i, y1, j, y2, ST7735_BLACK) відбувається затирання старого графіка, змінивши вхідні параметри на i, y3, j, y4, ST7735_GREEN, відбувається виведення нового сигналу.

Таким чином, використання TFT-дисплея з драйвером ST7735 дозволяє реалізувати високоякісну графічну інтерфейсну підсистему в умовах обмежених апаратних ресурсів, що робить його ідеальним вибором для застосувань, де необхідна візуалізація аналогових сигналів у режимі реального часу.

3.6 Тестування пристрою

На рисунку 3.11 представлено систему цифрової візуалізації аналогових сигналів на основі STM32 зібрану на макетній платі. Система на рисунку має спрощену систему живлення, оскільки перебуває на етапі розробки.

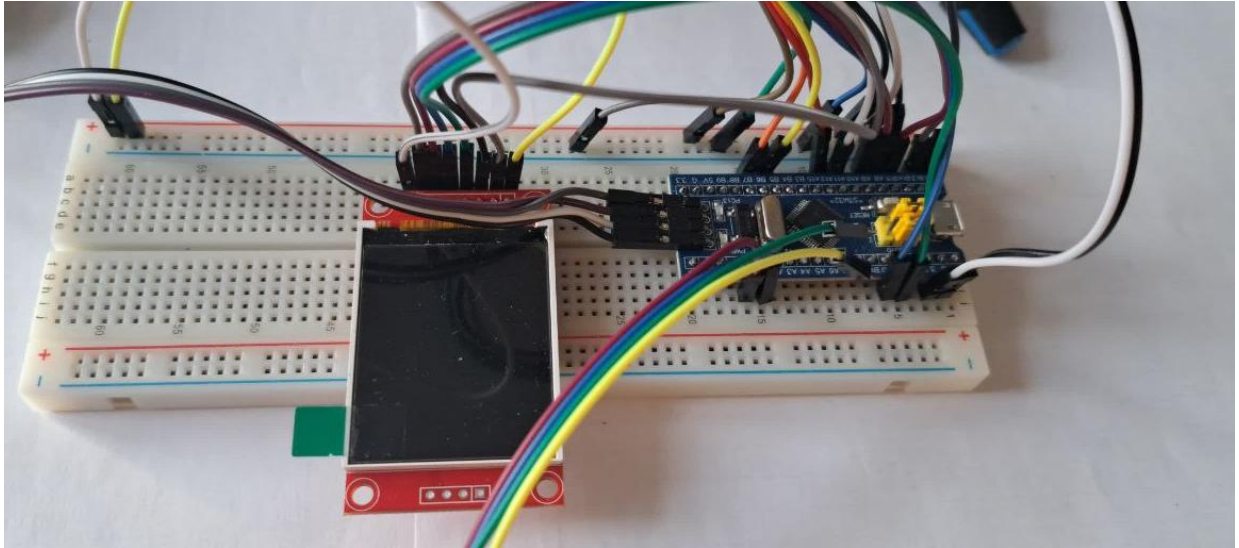


Рисунок 3.11 – Вигляд системи на макетній платі

На рисунку 3.12 система ввімкнена та відображає шумові данні при спробі вимірювання шкіри людини. Ці шуми це резудьтат впливу електромережі оскільки частота в електромережі 50 Гц тіло людини діє як антена і вловлює електромагнітні поля від побутових приладів, ламп, проводів у стінах тощо.

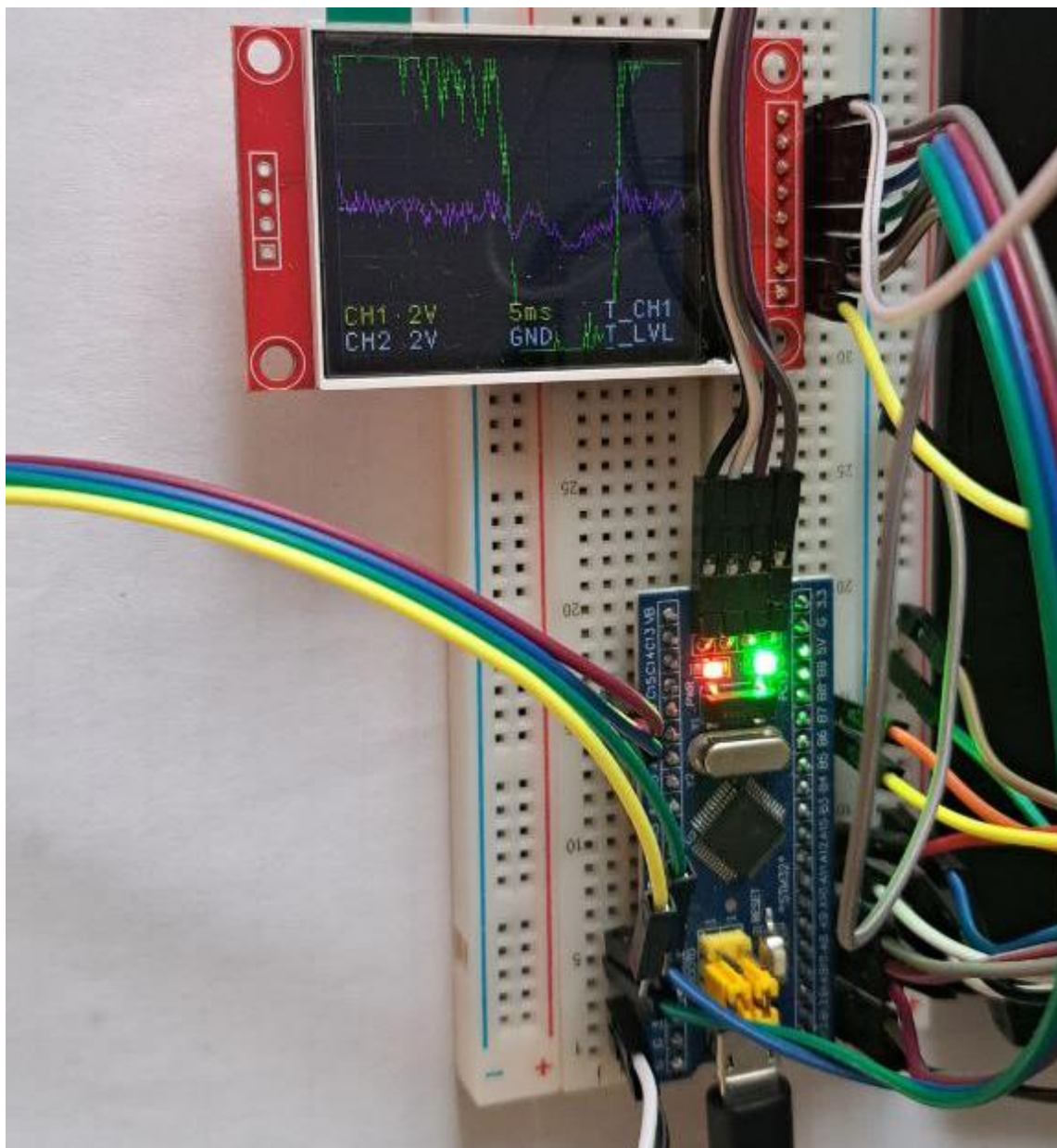


Рисунок 3.12 – Вигляд системи під час тестування

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при ушкодженні м'яких тканин, суглобів і кісток

Ушкодження м'яких тканин, суглобів та кісток є поширеними серед травматичних ситуацій, які виникають як у побутових умовах, так і під час дорожньо-транспортних пригод, занять спортом чи виконання виробничих обов'язків. До таких травм належать забої, розтягнення, вивихи, переломи та розриви зв'язок. Ефективне надання долікарської допомоги при вказаних ушкодженнях є вирішальним чинником зменшення ймовірності ускладнень, полегшення стану постраждалого та скорочення часу реабілітації.

Пошкодження м'яких тканин охоплюють травми шкіри, підшкірної клітковини, м'язів, сухожиль та зв'язок. Найбільш типовими є забої та розтягнення, які супроводжуються болем, набряком, гематомами, обмеженням рухів у ділянці ушкодження. Основною метою долікарської допомоги при таких травмах є зниження больового синдрому, попередження подальших пошкоджень та забезпечення іммобілізації. Стандартна послідовність дій включає охолодження (аплікація холоду на 15–20 хвилин), створення компресії (накладання еластичного бинта), підняття ушкодженої кінцівки вище рівня серця, а також знерухомлення ураженої ділянки.

Вивихи — це порушення анатомічної цілісності суглоба з роз'єднанням його поверхонь. Найчастіше вивихи виникають у плечовому, ліктьовому, пальцевому та кульшовому суглобах. Клінічними проявами є різкий біль, вимушене положення кінцівки, деформація суглоба, набряк. Категорично заборонено самотійно вправляти суглоб, оскільки це може призвести до пошкодження судин і нервів. Долікарська допомога полягає у фіксації суглоба

					КС КРБ 123.222.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Курцеба Т.В.			Безпека життєдіяльності, основи охорони праці			Лім.	Арк.	Аркушів	
Перевірив		Лецишин Ю.З.								53	
Консульт.		Пилипець М.І.						ТНТУ, каф. КС, гр. СІ-42			
Н. Контр.		Тиш Є.В.									
Затверд.		Осухівська Г.М.									

в тому положенні, в якому він перебуває після травми, з наступним охолодженням ураженої ділянки. Постраждалого необхідно транспортувати до лікувального закладу якомога швидше [17].

Переломи кісток можуть бути відкритими (з порушенням цілісності шкіри) або закритими. Симптомами перелому є біль, патологічна рухливість, деформація, набряк, порушення функції кінцівки, а при відкритому — наявність рани з кістковими уламками. Долікарська допомога при переломах передбачає іммобілізацію ушкодженої ділянки за допомогою шин (стандартних або імпровізованих). Забезпечити фіксацію не лише місця перелому, а й сусідніх суглобів. При відкритому переломі спочатку зупинити кровотечу (накладанням пов'язки або джгута), обробити рану антисептиком, після чого здійснити іммобілізацію.

У разі ушкодження суглобів або кісток забезпечити повний спокій травмованій ділянці. Постійно моніторити стан постраждалого, зокрема свідомість, дихання, пульс. При різкому погіршенні загального стану (втрата свідомості, ознаки шоку) — надати базову реанімаційну допомогу до прибуття медичного персоналу. Особливу увагу слід приділяти профілактиці гіпотермії у холодну пору року та правильному транспортуванню постраждалих, уникаючи додаткових ушкоджень [18].

Знання базових принципів долікарської допомоги є складовою загальної безпеки життєдіяльності громадян. Навички правильного застосування засобів першої допомоги, таких як еластичні бинти, іммобілізаційні шини, холодові компреси та антисептики, мають практичне значення у надзвичайних ситуаціях. Водночас необхідно враховувати індивідуальні особливості постраждалого (вік, хронічні хвороби, алергічні реакції), щоб уникнути погіршення стану через необережні дії.

При розробці комп'ютеризованої системи візуалізації аналогових сигналів на основі STM32 враховані вимоги щодо надання долікарської допомоги при ушкодженні м'яких тканин, суглобів і кісток.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2 Вплив електромагнітних полів (ЕМП) на людину та заходи щодо зменшення їх впливу

Електромагнітні поля належать до факторів, які впливають на фізіологічні та біохімічні процеси в організмі людини. Інтенсивність та частота ЕМП визначають характер їхнього впливу, що може призводити до короточасних суб'єктивних симптомів або до кумулятивних ефектів при тривалому впливі.

ЕМП поділяються за частотним діапазоном: від наднизькочастотних (50–60 Гц) до радіочастот (від десятків кГц до гігагерц) і міліметрових хвиль. Найбільше занепокоєння викликає вплив радіочастотного випромінювання, яке використовується в телекомунікаційних системах, зокрема у 4G та 5G технологіях. Встановлено, що тривалий вплив таких полів на організм може викликати функціональні порушення в нервовій системі, зниження концентрації уваги, порушення сну, підвищену втомлюваність та інші вегетативні розлади

Довготривале опромінення електромагнітними полями високочастотного діапазону (радіочастоти) спричиняє підвищення температури тканин, що підвищує ризик термічних ушкоджень та порушень обміну речовин. Згідно з ДСанПіН 3.3.2-976-00, максимально допустимий рівень потужності на робочому місці не повинен перевищувати 10 мВт/см² [19].

При низьких частотах переважає індукція струмів у тканинах, що призводить до стимуляції нервових і м'язових структур. Особливої уваги потребує питання впливу ЕМП на дітей, вагітних жінок і людей похилого віку. Для цих груп ризику рекомендується мінімізувати час перебування поблизу джерел випромінювання, зокрема обмежити використання бездротових пристроїв та не встановлювати Wi-Fi роутери у спальнях чи дитячих кімнатах. Уникати тривалого користування мобільними телефонами без гарнітури або гучного зв'язку.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Заходи технічного характеру передбачають екранування джерел ЕМП за допомогою провідних матеріалів (металеві екрани, сітки) та створення захисних огорожень відповідно до норм ДСТУ EN 50289-1:2013. Екранування дозволяє знизити рівень поля у зонах перебування персоналу до безпечних значень [20].

Адміністративні заходи включають організацію графіка роботи з джерелами ЕМП: обмеження тривалості безпосереднього контакту, чергування операторів, планування перерв та моніторинг робочого часу. Такі заходи допомагають зменшити кумулятивний вплив на організм працівників.

Індивідуальні засоби захисту персоналу передбачають використання діелектричних рукавичок, екранів на руках та головних уборах, що містять металічні волокна. Застосування таких засобів значно знижує рівень електромагнітного опромінення тіла.

Регулярний медичний контроль працівників, зайнятих у зонах підвищеного ЕМП, передбачає проведення щорічних профілактичних оглядів із вимірюванням артеріального тиску, електрокардіографії та загального аналізу крові.

Моніторинг рівня ЕМП на робочих місцях здійснюється за допомогою портативних електромагнітометрів та стаціонарних вимірювальних пунктів. Дані вимірів фіксуються у журналах з урахуванням часових інтервалів та рівня поля, що дає змогу оперативно реагувати на можливі перевищення норм.

Навчання та інструктаж персоналу охоплює ознайомлення з джерелами ЕМП, принципами вимірювання, алгоритмами реагування при виявленні перевищень та правилами експлуатації засобів захисту.

Комплексний підхід до оцінки ризиків, поєднання технічних, адміністративних та індивідуальних заходів створює ефективну систему захисту персоналу від шкідливого впливу електромагнітних полів. Дотримання нормативних документів та регулярний контроль забезпечують безпечні умови праці. При розробці комп'ютеризованої системи всі згадані норми враховані.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи спроектовано та реалізовано комп'ютеризовану систему візуалізації аналогових сигналів на базі мікроконтролера STM32F103C8T6. Розроблена система дозволяє здійснювати обробку та графічне відображення електричних сигналів у реальному часі на TFT-дисплеї з використанням драйвера ST7735. Такий підхід забезпечує зручний інтерфейс для аналізу аналогових процесів у вбудованих та лабораторних застосуваннях.

Апаратна частина системи включає АЦП мікроконтролера, двоканальне оцифрування, енкодери для налаштування параметрів відображення, буферну пам'ять для зберігання відліків та інтерфейс SPI для керування дисплеєм. Особливу увагу приділено забезпеченню точного синхронного відображення сигналів, що досягається шляхом реалізації тригерного механізму з автоматичним визначенням точки спрацьовування. Це дозволяє стабілізувати зображення періодичних сигналів на екрані та полегшує їх аналіз.

Програмна реалізація системи була побудована з використанням бібліотек низького рівня (LL) та передбачає оптимальне використання апаратних ресурсів мікроконтролера. Було реалізовано ефективне виведення графічних примітивів, стирання та оновлення осцилограм, а також адаптивну корекцію значень сигналу для коректного відображення. Буферизація даних та циклічне оновлення дисплея забезпечили плавне зображення сигналів навіть за високої частоти дискретизації.

Виконане тестування підтвердило стабільність роботи системи, відповідність графічного відображення дійсним значенням вхідних аналогових сигналів та коректність програмної логіки. Розроблене рішення відзначається високою гнучкістю, модульністю та потенціалом для подальшого вдосконалення – зокрема, через підключення інших типів дисплеїв, розширення кількості каналів або впровадження функцій спектрального аналізу та цифрової фільтрації.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, створена система є надійним та ефективним інструментом для візуалізації аналогових сигналів і має перспективи впровадження у навчальні, наукові та технічні галузі, де необхідне компактне та точне засобо вимірювання та відображення електричних параметрів.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.

2. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.

3. Оконський М.В., Лупенко С.А., Паламар А.М. Інформаційно-вимірювальна система для контролю метеорологічних параметрів на основі Інтернету речей. Матеріали ІХ науково-технічної конференції "Інформаційні моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 8–9 грудня 2021 року), Тернопіль: ТНТУ, 2021. С. 118.

4. Palamar A., Karpinski M., Palamar M., Osukhivska H., Mytnyk M. Remote Air Pollution Monitoring System Based on Internet of Things. CEUR Workshop Proceedings, 2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2022), Ternopil, Ukraine, November 22–24, 2022. Vol. 3309. P. 194-204.

5. Оконський М.В., Лупенко С.А., Паламар А.М. Комп'ютерна система для моніторингу метеорологічних параметрів на основі IoT. Актуальні задачі сучасних технологій : збірник тез доповідей X міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 24–25 листопада 2021 року), Тернопіль: ТНТУ, 2021. С. 112.

6. Ясінський Р.В., Осухівська Г.М., Паламар А.М. Апаратно-програмна система для регулювання мікроклімату теплиць. Матеріали X науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 7-8 грудня 2022 року), Тернопіль: ТНТУ, 2022. С. 102.

					КС КРБ 123.222.00.00 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

7. Palamar A., Palamar M., Osukhivska H. Real-time Health Monitoring Computer System Based on Internet of Medical Things. CEUR Workshop Proceedings, 3rd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2023), Ternopil, Ukraine, Opole, Poland, November 22–24, 2023. Vol. 3628. P. 106-115.

8. Інтерфейси обміну даними UART, SPI. Їх застосування і програмування для мікроконтролерів URL: <https://dl.tntu.edu.ua/content.php?cid=494855> (дата звернення: 15.04.2025).

9. STM32CubeIDE, Integrated Development Environment for STM32. URL: <https://www.st.com/en/development-tools/stm32cubeide.html> (дата звернення: 07.05.2025).

10. Зовсім просто про Virtual COM Port поверх USB плати STM32F3Discovery URL: <https://indrekis2.blogspot.com/2017/12/virtual-com-port-usb-stm32f3discovery.html> (дата звернення: 10.05.2025).

11. STM32H7R/S: How to configure an ADC DMA transfer in circular mode using STM32CubeMX URL: <https://community.st.com/t5/stm32-mcus/stm32h7r-s-how-to-configure-an-adc-dma-transfer-in-circular-mode/tap/737149> (дата звернення: 12.05.2025).

12. STM32 Timer Encoder Mode – STM32 Rotary Encoder Interfacing URL: <https://deepbluembedded.com/stm32-timer-encoder-mode-stm32-rotary-encoder-interfacing/> (дата звернення: 14.05.2025).

13. Edge Triggering and Level Triggering URL: <https://www.geeksforgeeks.org/digital-logic/edge-triggering-and-level-triggering/> (дата звернення: 17.05.2025).

14. RM0477 Reference manual STM32H7Rx/7Sx Arm® -based 32-bit MCUs URL: https://www.st.com/resource/en/reference_manual/rm0477-stm32h7rx7sx-armbased-32bit-mcus-stmicroelectronics.pdf (дата звернення: 21.05.2025).

					КС КРБ 123.222.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

15. John G. Proakis & Dimitris G. Manolakis "Digital Signal Processing"
URL: <https://mrce.in/ebooks/Digital%20Signal%20Processing%204th%20Ed.pdf>
(дата звернення: 24.05.2025).

16. Tymkiv P., Leshchyshyn Y. Algorithm Reliability of Kalman Filter Coefficients Determination for Low-Intensity Electroretinosignal. IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM). 2019. P.1-5.

17. Безпечна поведінка в екстремальних ситуаціях: Навчальний посібник / за ред. О.Ю. Іщенко. — Київ: Центр учбової літератури, 2021. — 144 с.

18. Основи медичних знань: Навчальний посібник / В.І. Чередниченко, Л.О. Шишкіна. — Харків: ХНУ імені В.Н. Каразіна, 2020. — 138 с.

19. ДСанПіН 3.3.2-976-00 «Гігієнічні вимоги до електромагнітних полів» (2000). URL: <https://moz.gov.ua/health/hygienic-standards> 1234 (дата звернення: 01.06.2025).

20. ДСТУ EN 50289-1:2013 «Кабелі та провідні матеріали». URL: <https://zakon.rada.gov.ua/laws/show/z0782-13> 1234 (дата звернення: 01.06.2025).

					КС КРБ 123.222.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2025 р

Комп'ютеризована система візуалізації аналогових сигналів на базі
мікроконтролера STM32

ТЕХНІЧНЕ ЗАВДАННЯ

на _8_ листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-42

_____ к.т.н., доц. Лещишин Ю.З.

_____ Курцеба Т.В.

« ____ » _____ 2025 р.

« ____ » _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система візуалізації аналогових сигналів на базі мікроконтролера STM32».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.222.00.00

1.2 Виконавець

Студент групи СІ-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Курцеба Т.В.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-53 від 27.01.2025 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Система розробляється для автоматизованого збору, оцифрування, обробки та візуалізації аналогових електричних сигналів у реальному часі на вбудованому TFT-дисплеї. Система побудована на основі мікроконтролера STM32F103C8T6 і використовується для дослідження та аналізу форми сигналу, контролю параметрів, а також налаштування режимів відображення через енкодери.

2.2 Мета створення системи

Мета роботи полягає у створенні компактного, енергоефективного, апаратно-програмного комплексу, здатного замінити базові функції цифрового осцилографа для аналізу аналогових сигналів. Реалізація передбачає використання АЦП із DMA, таймерів, інтерфейсу SPI для дисплея ST7735, а також інтерактивного управління через енкодери для зміни параметрів відображення сигналу.

2.3 Характеристика об'єкту

Об'єктом дослідження є аналогові сигнали, що потребують візуалізації у вигляді осцилограм. Система повинна забезпечувати точну побудову графіків у реальному часі навіть при високочастотних змінах сигналу. Пристрій орієнтовано на застосування у вимірювальних установках, працюючи в умовах обмежених апаратних ресурсів та зовнішніх завад.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Система повинна мати модульну структуру, що включає блок оцифрування сигналів на базі STM32F103C8T6, інтерфейс виведення даних на TFT-дисплей з драйвером ST7735, блок керування параметрами через енкодери, а також систему тригерного захоплення. Апаратна реалізація передбачає використання DMA для збору сигналів із кількох аналогових каналів, таймерів для синхронізації та периферії SPI для графічного виводу.

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Передача даних від АЦП до пам'яті виконується за допомогою DMA без навантаження на основне ядро. Комунікація між STM32 і TFT-дисплеєм здійснюється через SPI-інтерфейс. Керування параметрами (масштаб, тригер, коефіцієнти) відбувається через енкодери, підключені до таймерів у режимі енкодерного інтерфейсу, а обробка натискань реалізована через зовнішні переривання (EXTI).

3.1.3 Вимоги до режимів функціонування системи

У стандартному режимі система працює в циклі збору аналогових сигналів, обробки даних і виведення результатів у вигляді осцилограми на дисплей із частотою оновлення, що задається користувачем. При відсутності

стабільного сигналу система активує автотрігер. У разі зміни користувачем налаштувань тригера або масштабу, система негайно оновлює режими збору та обробки даних.

3.1.4 Вимоги по діагностуванню системи

Після увімкнення система повинна проводити перевірку працездатності АЦП, SPI-інтерфейсу дисплея, енкодерів та основних параметрів таймерів. У разі виявлення несправностей, повідомлення про помилки мають бути виведені на дисплей у вигляді текстових повідомлень або символів. Також передбачене виведення тестової осцилограми для перевірки графічного модуля.

3.1.5 Перспективи розвитку, проектування системи

Подальший розвиток системи передбачає розширення кількості каналів, впровадження алгоритмів цифрової фільтрації та спектрального аналізу, можливість передачі даних через UART до ПК, а також інтеграцію SD-карти для запису осцилограм. Додатково розглядається розширення дисплейного інтерфейсу для роботи з більшими екранами та виведенням декількох сигналів одночасно.

3.2 Показники призначення

Система візуалізації аналогових сигналів призначена для одночасного аналізу щонайменше двох каналів аналогових вхідних сигналів. Система забезпечує відображення даних у режимі реального часу на TFT-дисплеї, а також дозволяє змінювати параметри масштабування, затримки, рівня тригера та коефіцієнтів ослаблення сигналу.

3.2.1 Вимоги до надійності

Система повинна забезпечувати безперервну роботу впродовж щонайменше 72 годин без потреби у перезавантаженні або втручанні

користувача. Після відключення живлення система має автоматично відновлювати працездатність, зберігаючи останні налаштування режимів відображення.

3.3 Вимоги до безпеки

Усі зовнішні інтерфейси повинні бути захищені від випадкового дотику згідно з вимогами електробезпеки, особливо у разі підключення до джерел сигналу з потенційно небезпечними рівнями напруги. Живлення системи має мати вбудовані захисти від коротких замикань, перенапруги й перегріву стабілізатора напруги.

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Система має бути придатною до експлуатації при температурах від 0 до +40 °C та відносній вологості 20–80 %. Рекомендовано проводити профілактичну перевірку з'єднань, енкодерів і тест дисплея не рідше одного разу на рік. Компактний формфактор і модульність дозволяють зручно зберігати пристрій у лабораторних умовах.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

У разі реалізації функції передачі сигналів на зовнішній пристрій через UART, USB або Bluetooth, система повинна забезпечувати базовий рівень захисту переданих даних: контроль доступу, аутентифікацію, а при необхідності шифрування.

3.4.1 Вимоги по збереженню інформації при аваріях

Основні конфігураційні параметри (масштаб, рівень тригера, активний канал, коефіцієнт ослаблення тощо) повинні зберігатися у внутрішній flash-пам'яті мікроконтролера і відновлюватися після перезапуску. Для

майбутнього розширення передбачається можливість запису осцилограм на карту пам'яті microSD.

3.4.2 Вимоги по стандартизації і уніфікації

Усі апаратні компоненти повинні відповідати чинним технічним стандартам (наприклад, RoHS, CE) та забезпечувати сумісність з типовими джерелами живлення й лабораторними зондами. Програмне забезпечення має відповідати принципам модульності, читабельності й бути адаптованим для подальшого вдосконалення.

3.4.3 Вимоги до функцій (завдань), що виконуються системою

Система повинна забезпечувати: безперервне відображення аналогового сигналу на дисплеї з частотою оновлення не нижче 10 Гц; зміну режимів масштабування по горизонталі та вертикалі; встановлення рівня тригера та його відображення; графічне стирання попередньої осцилограми перед відображенням нової; інтуїтивний інтерфейс користувача на TFT-дисплеї; використання енкодерів для налаштування параметрів без клавіатури чи меню.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 - а) Структурна схема.
 - б) Блок схема роботи комп'ютеризованої системи.
 - в) Схема електрична принципова.
 - г) Схема електрична монтажна (з'єднань).

*Примітка: У комплект документації можуть вноситися зміни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання.	27.01.2025р. – 02.02.2025р.
2	Аналіз технічного завдання для розроблення системи візуалізації аналогових сигналів на основі STM32.	12.02.2025р. – 27.03.2025р.
3	Проектування системи візуалізації аналогових сигналів	06.04.2025р. – 03.05.2025р.
4	Розробка та тестування системи візуалізації аналогових сигналів	05.05.2025р. – 27.05.2025р.
5	Безпека життєдіяльності, основи охорони праці	27.05.2025р. – 02.06.2025р.
6	Оформлення пояснювальної записки і графічного матеріалу	03.06.2025р. – 09.06.2025р.
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	09.06.2025р. – 15.06.2025р.
8	Попередній захист кваліфікаційної роботи бакалавра	16.06.2025р. – 22.06.2025р.
9	Захист кваліфікаційної роботи бакалавра	26.06.2025

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Лістинг коду програми для STM32

```
/*
 * dual trace oscilloscope with a blue pill (STM32F103C8)
 * MUCH overclocked on 120 MHz with the ADC on 30 MHz
 * 1.8 inch TFT screen 160x128 pixel resolution
 *
 * rotary Encoder for setting of CH1/CH2 ATTENUATION and
GND_level
 * rotary Encoder for setting of TIMEBASE and TRIGGERLEVEL
 *
 * timebase from 20us to 200ms / division
 * sensitivity from 500mV to 2V / division
 *
 * XTAL now 8MHz
 */

#define BUFFERSIZE 320
#define LASTESTTRIGPOINT 160
#define X_SIZE 160
#define Y_SIZE 128
#define X_MAX 159
#define Y_MAX 127

#define TIMEBASE_MODE 0
#define TRIGGER_LEVEL_MODE 1
#define TRIGGER_CHANNEL_MODE 2

#define SPECTRUM_MODE 3

#define ATTENUATOR1_MODE 0
#define ATTENUATOR2_MODE 1
#define OFFSET_MODE 2

#define ATTENUATOR_MULTIPLY_MODE 3

#define FFT_SIZE 256

#include "main.h"

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_ADC1_Init(void);
static void MX_ADC2_Init(void);
static void MX_SPI2_Init(void);

static void MX_TIM2_Init(void);
static void MX_TIM3_Init(void);
```

```

static void MX_TIM4_Init(void);

void DrawRaster(void);

uint32_t adc_buffer[BUFFERSIZE];
uint16_t trace_1_buffer[BUFFERSIZE];
uint16_t trace_2_buffer[BUFFERSIZE];
uint16_t erase_1_buffer[BUFFERSIZE];
uint16_t erase_2_buffer[BUFFERSIZE];

const int8_t attenuator_multiply_values[] = {1, 10};

volatile int8_t
conversion_ready;
//0, 1
volatile int8_t
function_button_encoder1;
//0, 1          timebase, triggerlevel
volatile int8_t
function_button_encoder2;
//0, 1, 2, 3    atten1, atten2, virtual_ground
volatile int8_t
timebase;
//0 .. 13       20us .. 200ms / division
volatile int16_t
trigger_level;
/0 .. 127       top to bottom of screen
volatile int8_t
trigger_channel;
//0, 1          channel1, channel2

volatile int8_t
attenuator_ch1;
//0, 1, 2, 3    500mV, 1V, 2V, 5V / division
volatile int8_t
attenuator_ch2;
//0, 1, 2, 3    500mV, 1V, 2V, 5V / division
volatile int16_t
virtual_ground;
/0 .. 127       top to bottom of screen

volatile uint16_t attenuator_ch1_color;
volatile uint16_t attenuator_ch2_color;
volatile uint16_t timebase_color;
volatile uint16_t trigger_level_color;
volatile uint16_t trigger_channel_color;
volatile uint16_t virtual_ground_color;

volatile uint16_t attenuator_multiply_color;
volatile int8_t attenuator_multiply_idx = 0;

#define ATTENUATOR_MULTIPLY
(attenuator_multiply_values[attenuator_multiply_idx])

```

```

int main(void)
{
    uint16_t temp1, temp2;

    uint16_t i, j;

    uint16_t x1, y1, z1;
    uint16_t x2, y2, z2;
    uint16_t x3, y3, z3;
    uint16_t x4, y4, z4;

    uint16_t trigger_point;
    uint16_t previous_trigger_point;
    uint16_t previous_trigger_level;

    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_AFIO);
    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_PWR);

    NVIC_SetPriorityGrouping(NVIC_PRIORITYGROUP_4);
    LL_GPIO_AF_Remap_SWJ_NOJTAG();

    SystemClock_Config();
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_ADC1_Init();
    MX_ADC2_Init();
    MX_SPI2_Init();
    MX_TIM2_Init();
    MX_TIM3_Init();
    MX_TIM4_Init();

    LL_ADC_Enable(ADC1);
    LL_ADC_StartCalibration(ADC1);
    while (LL_ADC_IsCalibrationOnGoing(ADC1));
    LL_mDelay(10);

    LL_ADC_Enable(ADC2);
    LL_ADC_StartCalibration(ADC2);
    while (LL_ADC_IsCalibrationOnGoing(ADC2));
    LL_mDelay(10);

    LL_TIM_EnableCounter(TIM2);
    //10 kHz irqs for debouncing

    LL_TIM_CC_EnableChannel(TIM4, LL_TIM_CHANNEL_CH3);
    LL_TIM_CC_EnableChannel(TIM4, LL_TIM_CHANNEL_CH4);
    LL_TIM_EnableCounter(TIM4);
    //virtual_ground OPAMPS

    LL_SPI_Enable(SPI2);
    //control DISPLAY

```

```

    ST7735_Init();
    ST7735_FillScreen(ST7735_BLACK);

    //default settings
    timebase =
9;
//timebase on 200us/div
    LL_TIM_SetPrescaler(TIM3, 0);
    LL_TIM_SetAutoReload(TIM3, 749);

    conversion_ready =
0; //wait
for conversion to complete
    function_button_encoder1 = TIMEBASE_MODE;
    function_button_encoder2 = ATTENUATOR1_MODE;

    attenuator_ch1 =
0; //2V/
div
    attenuator_ch2 = 0;

    virtual_ground =
64; //0V
    trigger_level =
64; //mi
ddle of screen
    previous_trigger_level = 64;

    trigger_point =
1; //st
art of array
    previous_trigger_point = 1;

    attenuator_ch1_color = ST7735_YELLOW;
    attenuator_ch2_color = ST7735_WHITE;
    timebase_color = ST7735_YELLOW;
    trigger_level_color = ST7735_WHITE;
    trigger_channel_color = ST7735_WHITE;
    virtual_ground_color = ST7735_WHITE;

    LL_DMA_SetPeriphAddress(DMA1, LL_DMA_CHANNEL_1, (uint32_t)
&ADC1->DR);
    LL_DMA_SetMemoryAddress(DMA1, LL_DMA_CHANNEL_1, (uint32_t)
&adc_buffer);
    LL_DMA_SetDataLength(DMA1, LL_DMA_CHANNEL_1, BUFFERSIZE);
    LL_DMA_EnableIT_TC(DMA1,
LL_DMA_CHANNEL_1); //Enable IRQ
at transfer complete
    LL_DMA_EnableChannel(DMA1, LL_DMA_CHANNEL_1);

```

```

    LL_ADC_REG_StartConversionExtTrig(ADC1,
LL_ADC_REG_TRIG_EXT_RISING);
    LL_ADC_REG_StartConversionExtTrig(ADC2,
LL_ADC_REG_TRIG_EXT_RISING);
    LL_TIM_EnableCounter(TIM3);
        //triggering ADCs for selected timebase setting

    while (1)
    {
        //data processing
        if (conversion_ready ==
1) //ADC's are
ready
        {
            for (i = 0; i < BUFFERSIZE;
i++) //copy data from
adc_buffer to display_buffer
            {
                //the adc is continuously converting
                trace_1_buffer[i] = (uint16_t) ((adc_buffer[i]
>> 21)*ATTENUATOR_MULTIPLY); //separate both channels
                trace_2_buffer[i] = (uint16_t) ((adc_buffer[i]
& 0x0000FFFF) >> 5)*ATTENUATOR_MULTIPLY);

                //corrections because of the 3.3V reference of
the ADCs
                temp1 = trace_1_buffer[i];
                trace_1_buffer[i] = (temp1 + (temp1 >> 2) - 16);
                if (trace_1_buffer[i] > 9999) trace_1_buffer[i]
= 0; // >9999 means negative
                if (trace_1_buffer[i] > 127) trace_1_buffer[i] =
127;

                temp2 = trace_2_buffer[i];
                trace_2_buffer[i] = (temp2 + (temp2 >> 2) - 16);
                if (trace_2_buffer[i] > 9999) trace_2_buffer[i]
= 0;
                if (trace_2_buffer[i] > 127) trace_2_buffer[i] =
127;

            }

            i =
1;
            //find trigger_point
            if (trigger_channel ==
0) //with channel 1
            {
                while ((trace_1_buffer[i - 1] < trigger_level)
|| (trace_1_buffer[i] >= trigger_level)) i++;
            }
            else
                //with channel 2

```

```

        {
            while ((trace_2_buffer[i - 1] < trigger_level)
|| (trace_2_buffer[i] >= trigger_level)) i++;
        }
        if (i > X_MAX) trigger_point =
1;                                     //if the triggerpoint isn't
found
            else trigger_point =
i;                                     //show data from
beginning of arrays

        DrawRaster();
        //draw the raster before drawing the traces

        ST7735_HLine(1, 10, previous_trigger_level,
ST7735_BLACK);           //draw small trigger-level line
        ST7735_HLine(1, 10, (trigger_level),
trigger_level_color);
        previous_trigger_level = trigger_level;

        for (i = 1; i < X_MAX; i++)
        {
            j = i + 1;

            x1 = i + previous_trigger_point;
            x2 = x1 + 1;
            x3 = i + trigger_point;
            x4 = x3 + 1;

            y1 = erase_1_buffer[x1];
            y2 = erase_1_buffer[x2];
            y3 = trace_1_buffer[x3];
            y4 = trace_1_buffer[x4];

            z1 = erase_2_buffer[x1];
            z2 = erase_2_buffer[x2];
            z3 = trace_2_buffer[x3];
            z4 = trace_2_buffer[x4];

            ST7735_Line(i, y1, j, y2,
ST7735_BLACK);           //remove old trace
            ST7735_Line(i, z1, j, z2,
ST7735_BLACK);           //remove old trace

            ST7735_Line(i, y3, j, y4,
ST7735_GREEN);           //show new trace
            ST7735_Line(i, z3, j, z4,
ST7735_MAGENTA);         //show new trace
        }

        for (i = 0; i < BUFFERSIZE;
i++)                         //store values, used
next time to erase the old trace

```

```

        {
            erase_1_buffer[i] = trace_1_buffer[i];
            erase_2_buffer[i] = trace_2_buffer[i];
        }
        previous_trigger_point = trigger_point;

        conversion_ready =
0; //start next
sequence of conversion
        LL_DMA_ClearFlag_TE1(DMA1);
            //just in case...
        LL_DMA_DisableChannel(DMA1,
LL_DMA_CHANNEL_1); //disable DMA so that it
can be
        LL_DMA_SetDataLength(DMA1, LL_DMA_CHANNEL_1,
BUFFERSIZE); //set to the number of transfers
        LL_DMA_EnableChannel(DMA1,
LL_DMA_CHANNEL_1); //re-enable DMA

        LL_TIM_EnableCounter(TIM3);
            //and THEN start TIM3 --> triggers the ADCs
    }

    //draw text after raster and traces
    //upper text line
    if (attenuator_ch1 == 0) ST7735_WriteString(1, 106,
"CH1 2V ", Font_7x10, attenuator_ch1_color, ST7735_BLACK);
    else if (attenuator_ch1 == 1) ST7735_WriteString(1, 106,
"CH1 1V ", Font_7x10, attenuator_ch1_color, ST7735_BLACK);
    else
        ST7735_WriteString(1, 106,
"CH1 500mV", Font_7x10, attenuator_ch1_color, ST7735_BLACK);

        if (timebase == 0) ST7735_WriteString(55, 106,
"200ms", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 1) ST7735_WriteString(55, 106,
"100ms", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 2) ST7735_WriteString(55, 106,
"50ms ", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 3) ST7735_WriteString(55, 106,
"20ms ", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 4) ST7735_WriteString(55, 106,
"10ms ", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 5) ST7735_WriteString(55, 106,
"5ms ", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 6) ST7735_WriteString(55, 106,
"2ms ", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 7) ST7735_WriteString(55, 106,
"1ms ", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 8) ST7735_WriteString(55, 106,
"500us", Font_7x10, timebase_color, ST7735_BLACK);
    else if (timebase == 9) ST7735_WriteString(55, 106,
"200us", Font_7x10, timebase_color, ST7735_BLACK);

```

```

        else if (timebase == 10) ST7735_WriteString(55, 106,
"100us", Font_7x10, timebase_color, ST7735_BLACK);
        else if (timebase == 11) ST7735_WriteString(55, 106,
"50us ", Font_7x10, timebase_color, ST7735_BLACK);
        else
            ST7735_WriteString(55, 106,
"20us ", Font_7x10, timebase_color, ST7735_BLACK);

        if (trigger_channel == 0) ST7735_WriteString(120, 106,
"T_CH1", Font_7x10, trigger_channel_color, ST7735_BLACK);
        else
            ST7735_WriteString(120, 106,
"T_CH2", Font_7x10, trigger_channel_color, ST7735_BLACK);

        //lower text line
        if (attenuator_ch2 == 0) ST7735_WriteString(1, 117,
"CH2 2V ", Font_7x10, attenuator_ch2_color, ST7735_BLACK);
        else if (attenuator_ch2 == 1) ST7735_WriteString(1, 117,
"CH2 1V ", Font_7x10, attenuator_ch2_color, ST7735_BLACK);
        else
            ST7735_WriteString(1, 117,
"CH2 500mV", Font_7x10, attenuator_ch2_color, ST7735_BLACK);

        ST7735_FillRectangle(55, 117, 65, 10, ST7735_BLACK);
        if (function_button_encoder2 ==
ATTENUATOR_MULTIPLY_MODE)
        {

            char mult_str[16];
            sprintf(mult_str, "MULT x%d", ATTENUATOR_MULTIPLY);
            ST7735_WriteString(55, 117, mult_str, Font_7x10,
attenuator_multiply_color, ST7735_BLACK);
        }
        else
        {
            ST7735_WriteString(55, 117, "GND", Font_7x10,
virtual_ground_color, ST7735_BLACK);
        }

        ST7735_WriteString(120, 117, "T_LVL", Font_7x10,
trigger_level_color, ST7735_BLACK);

    }
}

void DrawRaster(void)
{
    //show raster
    ST7735_HLine(0, 159, 0, ST7735_GRAY);
    ST7735_HLine(0, 159, 21, ST7735_GRAY);
    ST7735_HLine(0, 159, 42, ST7735_GRAY);
    ST7735_HLine(0, 159, 63, ST7735_GRAY);
    ST7735_HLine(0, 159, 84, ST7735_GRAY);
    ST7735_HLine(0, 159, 105, ST7735_GRAY);
    ST7735_HLine(0, 159, 127, ST7735_GRAY);

```

```

    ST7735_VLine(0, 0, 127, ST7735_GRAY);
    ST7735_VLine(31, 0, 105, ST7735_GRAY);
    ST7735_VLine(63, 0, 105, ST7735_GRAY);
    ST7735_VLine(95, 0, 105, ST7735_GRAY);
    ST7735_VLine(127, 0, 105, ST7735_GRAY);
    ST7735_VLine(159, 0, 127, ST7735_GRAY);
}

void
SystemClock_Config(void)
    // XTAL 8 or 12 MHz
{
    LL_FLASH_SetLatency(LL_FLASH_LATENCY_2);
    if(LL_FLASH_GetLatency() != LL_FLASH_LATENCY_2)
Error_Handler();
    LL_FLASH_EnablePrefetch();
    LL_RCC_HSE_Enable();
    while(LL_RCC_HSE_IsReady() != 1);

    LL_RCC_PLL_ConfigDomain_SYS(LL_RCC_PLLSOURCE_HSE_DIV_1,
LL_RCC_PLL_MUL_15); // 120 MHz with 8 MHz xtal
    //LL_RCC_PLL_ConfigDomain_SYS(LL_RCC_PLLSOURCE_HSE_DIV_1,
LL_RCC_PLL_MUL_10); // 120 MHz with 12 MHz xtal

    LL_RCC_PLL_Enable();
    while(LL_RCC_PLL_IsReady() != 1);
    LL_RCC_SetAHBPrescaler(LL_RCC_SYSCLK_DIV_1);
    LL_RCC_SetAPB1Prescaler(LL_RCC_APB1_DIV_2);
    LL_RCC_SetAPB2Prescaler(LL_RCC_APB2_DIV_1);
    LL_RCC_SetSysClkSource(LL_RCC_SYS_CLKSOURCE_PLL);
    while(LL_RCC_GetSysClkSource() !=
LL_RCC_SYS_CLKSOURCE_STATUS_PLL);
    LL_InitlmsTick(120000000);
    LL_SetSystemCoreClock(120000000);
        //MCU is very much overclocked (120 MHz)
    LL_RCC_SetADCClockSource(LL_RCC_ADC_CLKSRC_PCLK2_DIV_4);
        //the ADC's are overclocked too (30 MHz)
}

static void MX_ADC1_Init(void)
{
    LL_ADC_InitTypeDef ADC_InitStruct = {0};
    LL_ADC_CommonInitTypeDef ADC_CommonInitStruct = {0};
    LL_ADC_REG_InitTypeDef ADC_REG_InitStruct = {0};
    LL_GPIO_InitTypeDef GPIO_InitStruct = {0};

    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_ADC1);
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOA);

    GPIO_InitStruct.Pin = LL_GPIO_PIN_0;
    GPIO_InitStruct.Mode = LL_GPIO_MODE_ANALOG;
    LL_GPIO_Init(GPIOA, &GPIO_InitStruct);

```

```

    LL_DMA_SetDataTransferDirection(DMA1, LL_DMA_CHANNEL_1,
LL_DMA_DIRECTION_PERIPH_TO_MEMORY);
    LL_DMA_SetChannelPriorityLevel(DMA1, LL_DMA_CHANNEL_1,
LL_DMA_PRIORITY_HIGH);
    LL_DMA_SetMode(DMA1, LL_DMA_CHANNEL_1, LL_DMA_MODE_NORMAL);
    LL_DMA_SetPeriphIncMode(DMA1, LL_DMA_CHANNEL_1,
LL_DMA_PERIPH_NOINCREMENT);
    LL_DMA_SetMemoryIncMode(DMA1, LL_DMA_CHANNEL_1,
LL_DMA_MEMORY_INCREMENT);
    LL_DMA_SetPeriphSize(DMA1, LL_DMA_CHANNEL_1,
LL_DMA_PDATAALIGN_WORD);
    LL_DMA_SetMemorySize(DMA1, LL_DMA_CHANNEL_1,
LL_DMA_MDATAALIGN_WORD);

    ADC_InitStruct.DataAlignment = LL_ADC_DATA_ALIGN_RIGHT;
    ADC_InitStruct.SequencersScanMode = LL_ADC_SEQ_SCAN_DISABLE;
    LL_ADC_Init(ADC1, &ADC_InitStruct);

    ADC_CommonInitStruct.Multimode =
LL_ADC_MULTI_DUAL_REG_SIMULT;
    LL_ADC_CommonInit(__LL_ADC_COMMON_INSTANCE(ADC1),
&ADC_CommonInitStruct);

    ADC_REG_InitStruct.TriggerSource =
LL_ADC_REG_TRIG_EXT_TIM3_TRGO;
    ADC_REG_InitStruct.SequencerLength =
LL_ADC_REG_SEQ_SCAN_DISABLE;
    ADC_REG_InitStruct.SequencerDiscont =
LL_ADC_REG_SEQ_DISCONT_DISABLE;
    ADC_REG_InitStruct.ContinuousMode = LL_ADC_REG_CONV_SINGLE;
    ADC_REG_InitStruct.DMATransfer =
LL_ADC_REG_DMA_TRANSFER_UNLIMITED;
    LL_ADC_REG_Init(ADC1, &ADC_REG_InitStruct);

    LL_ADC_REG_SetSequencerRanks(ADC1, LL_ADC_REG_RANK_1,
LL_ADC_CHANNEL_0);
    LL_ADC_SetChannelSamplingTime(ADC1, LL_ADC_CHANNEL_0,
LL_ADC_SAMPLINGTIME_1CYCLE_5);
}

static void MX_ADC2_Init(void)
{
    LL_ADC_InitTypeDef ADC_InitStruct = {0};
    LL_ADC_REG_InitTypeDef ADC_REG_InitStruct = {0};
    LL_GPIO_InitTypeDef GPIO_InitStruct = {0};

    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_ADC2);
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOA);

    GPIO_InitStruct.Pin = LL_GPIO_PIN_1;
    GPIO_InitStruct.Mode = LL_GPIO_MODE_ANALOG;
    LL_GPIO_Init(GPIOA, &GPIO_InitStruct);

```

```

    ADC_InitStruct.DataAlignment = LL_ADC_DATA_ALIGN_RIGHT;
    ADC_InitStruct.SequencersScanMode = LL_ADC_SEQ_SCAN_DISABLE;
    LL_ADC_Init(ADC2, &ADC_InitStruct);

    ADC_REG_InitStruct.TriggerSource = LL_ADC_REG_TRIG_SOFTWARE;
    ADC_REG_InitStruct.SequencerLength =
LL_ADC_REG_SEQ_SCAN_DISABLE;
    ADC_REG_InitStruct.SequencerDiscont =
LL_ADC_REG_SEQ_DISCONT_DISABLE;
    ADC_REG_InitStruct.ContinuousMode = LL_ADC_REG_CONV_SINGLE;
    ADC_REG_InitStruct.DMATransfer =
LL_ADC_REG_DMA_TRANSFER_NONE;
    LL_ADC_REG_Init(ADC2, &ADC_REG_InitStruct);

    LL_ADC_REG_SetSequencerRanks(ADC2, LL_ADC_REG_RANK_1,
LL_ADC_CHANNEL_1);
    LL_ADC_SetChannelSamplingTime(ADC2, LL_ADC_CHANNEL_1,
LL_ADC_SAMPLINGTIME_1CYCLE_5);
}

static void MX_SPI2_Init(void)
{
    LL_SPI_InitTypeDef SPI_InitStruct = {0};
    LL_GPIO_InitTypeDef GPIO_InitStruct = {0};

    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_SPI2);
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOB);

    GPIO_InitStruct.Pin = LL_GPIO_PIN_13|LL_GPIO_PIN_15;
    GPIO_InitStruct.Mode = LL_GPIO_MODE_ALTERNATE;
    GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_MEDIUM;
    GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    LL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    SPI_InitStruct.TransferDirection = LL_SPI_FULL_DUPLEX;
    SPI_InitStruct.Mode = LL_SPI_MODE_MASTER;
    SPI_InitStruct.DataWidth = LL_SPI_DATAWIDTH_8BIT;
    SPI_InitStruct.ClockPolarity = LL_SPI_POLARITY_LOW;
    SPI_InitStruct.ClockPhase = LL_SPI_PHASE_1EDGE;
    SPI_InitStruct.NSS = LL_SPI_NSS_SOFT;
    SPI_InitStruct.BaudRate = LL_SPI_BAUDRATEPRESCALER_DIV8;
    SPI_InitStruct.BitOrder = LL_SPI_MSB_FIRST;
    SPI_InitStruct.CRCCalculation =
LL_SPI_CRCCALCULATION_DISABLE;
    SPI_InitStruct.CRCPoly = 10;
    LL_SPI_Init(SPI2, &SPI_InitStruct);
}

static void
MX_TIM2_Init(void)
    //debouncing

```

```

{
    LL_TIM_InitTypeDef TIM_InitStruct = {0};

    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM2);

    NVIC_SetPriority(TIM2_IRQn,
NVIC_EncodePriority(NVIC_GetPriorityGrouping(),0, 0));
    NVIC_EnableIRQ(TIM2_IRQn);

    TIM_InitStruct.Prescaler =
119;                                     //1 MHz
    TIM_InitStruct.CounterMode = LL_TIM_COUNTERMODE_UP;
    TIM_InitStruct.Autoreload =
99;                                     //10 kHz
    TIM_InitStruct.ClockDivision = LL_TIM_CLOCKDIVISION_DIV1;
    LL_TIM_Init(TIM2, &TIM_InitStruct);

    LL_TIM_DisableARRPreload(TIM2);
    LL_TIM_SetClockSource(TIM2, LL_TIM_CLOCKSOURCE_INTERNAL);
    LL_TIM_SetTriggerOutput(TIM2, LL_TIM_TRGO_RESET);
    LL_TIM_DisableMasterSlaveMode(TIM2);
}

static void
MX_TIM3_Init(void)
    //triggering ADC
{
    LL_TIM_InitTypeDef TIM_InitStruct = {0};

    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM3);

    NVIC_SetPriority(TIM3_IRQn,
NVIC_EncodePriority(NVIC_GetPriorityGrouping(),0, 0));
    NVIC_EnableIRQ(TIM3_IRQn);

    TIM_InitStruct.Prescaler =
0;                                     // 120 MHz
    TIM_InitStruct.CounterMode = LL_TIM_COUNTERMODE_UP;
    TIM_InitStruct.Autoreload =
749;                                     // default
setting 200us/div
    TIM_InitStruct.ClockDivision = LL_TIM_CLOCKDIVISION_DIV1;
    LL_TIM_Init(TIM3, &TIM_InitStruct);

    LL_TIM_DisableARRPreload(TIM3);
    LL_TIM_SetClockSource(TIM3, LL_TIM_CLOCKSOURCE_INTERNAL);
    LL_TIM_SetTriggerOutput(TIM3, LL_TIM_TRGO_UPDATE);
    LL_TIM_DisableMasterSlaveMode(TIM3);
}

static void
MX_TIM4_Init(void)
    //virtual_ground OPAMPS (PWM)

```

```

{
    LL_TIM_InitTypeDef TIM_InitStruct = {0};
    LL_TIM_OC_InitTypeDef TIM_OC_InitStruct = {0};
    LL_GPIO_InitTypeDef GPIO_InitStruct = {0};

    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM4);
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOB);

    GPIO_InitStruct.Pin = LL_GPIO_PIN_8;
    GPIO_InitStruct.Mode = LL_GPIO_MODE_ALTERNATE;
    GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_LOW;
    GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSH_PULL;
    LL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    TIM_InitStruct.Prescaler = 0;
    TIM_InitStruct.CounterMode = LL_TIM_COUNTERMODE_UP;
    TIM_InitStruct.Autoreload = 1280;
    TIM_InitStruct.ClockDivision = LL_TIM_CLOCKDIVISION_DIV1;
    LL_TIM_Init(TIM4, &TIM_InitStruct);

    LL_TIM_DisableARRPreload(TIM4);
    LL_TIM_SetClockSource(TIM4, LL_TIM_CLOCKSOURCE_INTERNAL);

    LL_TIM_OC_EnablePreload(TIM4, LL_TIM_CHANNEL_CH3);
    TIM_OC_InitStruct.OCMode = LL_TIM_OC_MODE_PWM2;
    TIM_OC_InitStruct.OCState = LL_TIM_OC_STATE_DISABLE;
    TIM_OC_InitStruct.OCNState = LL_TIM_OC_STATE_DISABLE;
    TIM_OC_InitStruct.CompareValue = 640;
    TIM_OC_InitStruct.OCpolarity = LL_TIM_OC_POLARITY_HIGH;
    LL_TIM_OC_Init(TIM4, LL_TIM_CHANNEL_CH3,
&TIM_OC_InitStruct);
    LL_TIM_OC_DisableFast(TIM4, LL_TIM_CHANNEL_CH3);

    LL_TIM_SetTriggerOutput(TIM4, LL_TIM_TRGO_RESET);
    LL_TIM_DisableMasterSlaveMode(TIM4);
}

static void MX_DMA_Init(void)
{
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_DMA1);

    NVIC_SetPriority(DMA1_Channel1_IRQn,
NVIC_EncodePriority(NVIC_GetPriorityGrouping(), 0, 0));
    NVIC_EnableIRQ(DMA1_Channel1_IRQn);
}

static void MX_GPIO_Init(void)
{
    LL_EXTI_InitTypeDef EXTI_InitStruct = {0};
    LL_GPIO_InitTypeDef GPIO_InitStruct = {0};

    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOC);
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOD);

```

```

    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOA);
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_GPIOB);

    GPIO_InitStruct.Pin = LL_GPIO_PIN_4 | LL_GPIO_PIN_5 |
LL_GPIO_PIN_6 | LL_GPIO_PIN_7;
    GPIO_InitStruct.Mode =
LL_GPIO_MODE_OUTPUT; //to
attenuators (CD4066)
    GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_LOW;
    GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    LL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    GPIO_InitStruct.Pin =
LL_GPIO_PIN_13; //LED for
debugging
    GPIO_InitStruct.Mode = LL_GPIO_MODE_OUTPUT;
    GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_LOW;
    GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    LL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    GPIO_InitStruct.Pin =
LL_GPIO_PIN_10|LL_GPIO_PIN_11|LL_GPIO_PIN_12; //SPI to
display
    GPIO_InitStruct.Mode = LL_GPIO_MODE_OUTPUT;
    GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_MEDIUM;
    GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    LL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    GPIO_InitStruct.Pin =
LL_GPIO_PIN_7; //rotary
encoder1 pin B
    GPIO_InitStruct.Mode = LL_GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = LL_GPIO_PULL_UP;
    LL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    GPIO_InitStruct.Pin =
LL_GPIO_PIN_9; //rotary
encoder2 pin B
    GPIO_InitStruct.Mode = LL_GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = LL_GPIO_PULL_UP;
    LL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    LL_GPIO_AF_SetEXTISource(LL_GPIO_AF_EXTI_PORTB,
LL_GPIO_AF_EXTI_LINE6); //EXTI lines to rotary encoders
    LL_GPIO_AF_SetEXTISource(LL_GPIO_AF_EXTI_PORTA,
LL_GPIO_AF_EXTI_LINE8);
    LL_GPIO_AF_SetEXTISource(LL_GPIO_AF_EXTI_PORTA,
LL_GPIO_AF_EXTI_LINE11);
    LL_GPIO_AF_SetEXTISource(LL_GPIO_AF_EXTI_PORTA,
LL_GPIO_AF_EXTI_LINE12);

```

```

    EXTI_InitStruct.Line_0_31 =
LL_EXTI_LINE_6;                                     //rotary
encoder1 pin A
    EXTI_InitStruct.LineCommand = ENABLE;
    EXTI_InitStruct.Mode = LL_EXTI_MODE_IT;
    EXTI_InitStruct.Trigger = LL_EXTI_TRIGGER_FALLING;
    LL_EXTI_Init(&EXTI_InitStruct);

    EXTI_InitStruct.Line_0_31 =
LL_EXTI_LINE_8;                                     //rotary
encoder2 pin A
    EXTI_InitStruct.LineCommand = ENABLE;
    EXTI_InitStruct.Mode = LL_EXTI_MODE_IT;
    EXTI_InitStruct.Trigger = LL_EXTI_TRIGGER_FALLING;
    LL_EXTI_Init(&EXTI_InitStruct);

    EXTI_InitStruct.Line_0_31 =
LL_EXTI_LINE_11;                                    //rotary
encoder1 button
    EXTI_InitStruct.LineCommand = ENABLE;
    EXTI_InitStruct.Mode = LL_EXTI_MODE_IT;
    EXTI_InitStruct.Trigger = LL_EXTI_TRIGGER_FALLING;
    LL_EXTI_Init(&EXTI_InitStruct);

    EXTI_InitStruct.Line_0_31 =
LL_EXTI_LINE_12;                                    //rotary
encoder2 button
    EXTI_InitStruct.LineCommand = ENABLE;
    EXTI_InitStruct.Mode = LL_EXTI_MODE_IT;
    EXTI_InitStruct.Trigger = LL_EXTI_TRIGGER_FALLING;
    LL_EXTI_Init(&EXTI_InitStruct);

    LL_GPIO_SetPinPull(GPIOB, LL_GPIO_PIN_6, LL_GPIO_PULL_UP);
    LL_GPIO_SetPinPull(GPIOA, LL_GPIO_PIN_8, LL_GPIO_PULL_UP);
    LL_GPIO_SetPinPull(GPIOA, LL_GPIO_PIN_11, LL_GPIO_PULL_UP);
    LL_GPIO_SetPinPull(GPIOA, LL_GPIO_PIN_12, LL_GPIO_PULL_UP);

    LL_GPIO_SetPinMode(GPIOB, LL_GPIO_PIN_6,
LL_GPIO_MODE_INPUT);
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_8,
LL_GPIO_MODE_INPUT);
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_11,
LL_GPIO_MODE_INPUT);
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_12,
LL_GPIO_MODE_INPUT);

    NVIC_SetPriority(EXTI9_5_IRQn,
NVIC_EncodePriority(NVIC_GetPriorityGrouping(),0, 0));
    NVIC_EnableIRQ(EXTI9_5_IRQn);
    NVIC_SetPriority(EXTI15_10_IRQn,
NVIC_EncodePriority(NVIC_GetPriorityGrouping(),0, 0));
    NVIC_EnableIRQ(EXTI15_10_IRQn);

```

```

        LL_GPIO_SetOutputPin(GPIOA, LL_GPIO_PIN_4 | LL_GPIO_PIN_5 |
LL_GPIO_PIN_6 | LL_GPIO_PIN_7);
        LL_GPIO_SetOutputPin(GPIOB, LL_GPIO_PIN_1);
        LL_GPIO_SetOutputPin(GPIOB, LL_GPIO_PIN_10 | LL_GPIO_PIN_11
| LL_GPIO_PIN_12);
    }

void Error_Handler(void)
{

}

#ifdef  USE_FULL_ASSERT

void assert_failed(uint8_t *file, uint32_t line)
{

}
#endif

```

Додаток В

Перелік елементів схеми електричної принципової

[illegible]