

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютеризована система контролю введення ін'єкцій
у пацієнтів з цукровим діабетом*

Виконав: студент 4 курсу, групи СІ-41
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)



(підпис)

Скочинець Н.О.

(прізвище та ініціали)

Керівник



(підпис)

Шингера Н.Я.

(прізвище та ініціали)

Нормоконтроль



(підпис)

Тили Є.В.

(прізвище та ініціали)

Завідувач кафедри



(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент



(підпис)

Марценко С.В.

(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)

студента Скочипця Назарія Олеговича
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система контролю введення ін'єкцій
пацієнтів з цукровим діабетом

Керівник роботи Шингера Н.Я.
(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 19.06.2025 р.

3. Вихідні дані до роботи Технічне завдання

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

Вступ. 1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

II. Структурна схема системи

2. Система електрична принципова

3. Модель тестового макету

6. Консультанти розділів роботи

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

7. Дата видачі завдання 27.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітки
1.	Розробка технічного завдання	27.01 – 02.02	Виконано
2.	Аналіз технічного завдання	14.02 – 27.03	Виконано
3.	Проектування системи контролю введення ін'єкцій	03.04 – 02.05	Виконано
4.	Реалізація системи контролю введення ін'єкцій	08.05 – 26.05	Виконано
5.	Безпека життєдіяльності, основи охорони праці	27.05 – 02.06	Виконано
6.	Оформлення пояснювальної записки і графічного матеріалу	03.06 – 09.06	Виконано
7.	Перевірка на академічний плагіат, перевірка керівником та консультантами	9.06 – 15.06	Виконано
8.	Попередній захист кваліфікаційної роботи бакалавра	16.06 – 22.06	Виконано
9.	Захист кваліфікаційної роботи бакалавра	27.06.2025	Виконано

Студент

(підпис)

 Скочинець Н.О.
(прізвище та ініціали)

Керівник роботи

(підпис)

 Шингера Н.Я.
(прізвище та ініціали)

АНОТАЦІЯ

Скочипець Н.О. Комп'ютеризована система контролю введення ін'єкцій у пацієнтів з цукровим діабетом: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: система контролю, цукровий діабет, ін'єкція, esp32, rtc, eeprom, lcd, меню, звуковий сигнал, енкодер

Пояснювальна записка складається з чотирьох розділів.

У першому розділі проведено аналіз технічного завдання. Визначено функціональні вимоги до комп'ютерної системи контролю введення ін'єкцій, розглянуто можливі варіанти реалізації наявної задачі.

У другому розділі здійснено проектування системи контролю введення ін'єкцій. Побудовано узагальнену структурну схему. Проведено обґрунтування вибору апаратного та програмного забезпечення, необхідного для реалізації системи.

У третьому розділі описано реалізацію апаратної та програмної частин системи. Наведено алгоритм взаємодії користувача з меню пристрою. Проведено тестування працездатності пристрою за заданими сценаріями.

У четвертому розділі висвітлено аспекти безпеки життєдіяльності та охорони праці.

ANNOTATION

Skochypets N.O. Methods and tools for monitoring the surrounding noise:
Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil:
Ternopil Ivan Puluj National Technical University, 2025. — 86 p.

Keywords: CONTROL SYSTEM, DIABETES MELLITUS, INJECTION,
ESP32, RTC, EEPROM, LCD, MENU, AUDIO SIGNAL, ENCODER

The explanatory note consists of three sections.

The first section presents an analysis of the technical task. The functional requirements for the computerized injection control system are defined. Possible implementation options are considered.

The second section covers the design of the injection control system. A generalized structural diagram is developed. The choice of hardware and software required for system implementation is substantiated.

The third section describes the implementation of the hardware and software components of the system. The algorithm of user interaction with the device menu is provided. The system was tested under predefined scenarios.

The fourth chapter addresses occupational safety and health aspects.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	11
1.1 Аналіз вимог до комп'ютерної системи контролю введення ін'єкцій	11
1.2 Аналіз можливих рішень поставленого завдання.....	14
РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ КОНТРОЛЮ ВВЕДЕННЯ ІН'ЄКЦІЙ	17
2.1 Розробка узагальненої структури системи контролю введення ін'єкцій	17
2.2 Обґрунтування вибору апаратного забезпечення.....	19
2.2.1 Вибір мікроконтролерної плати.....	19
2.2.2 Вибір модуля годинника реального часу.....	22
2.2.3 Вибір дисплейного модуля.....	25
2.2.4 Вибір пристрою керування.....	26
2.2.5 Вибір модуля звукового сповіщення	28
2.3 Обґрунтування вибору програмного забезпечення.....	29
2.4 Проектування комп'ютерного засобу для контролю введення ін'єкцій ..	31
РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ВВЕДЕННЯ ІН'ЄКЦІЙ...	35
3.1 Реалізація апаратної частини	35
3.2 Реалізація програмної частини	37
3.3 Тестування системи контролю введення ін'єкцій.....	51
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .	57
4.1 Фізіологічний вплив факторів існування на життєдіяльність людини ...	57

				КС КРБ 123.235.00.00 ПЗ			
№	Арк.	№ докум.	Підпис	Дата	Лім.		
1	1	1	1	1			
2	2	2	2	2	6		
3	3	3	3	3	ТНТУ, каф. КС,		
4	4	4	4	4	гр. СІ-41		
5	5	5	5	5			

4.2 Психоемоційний вплив факторів існування на життєдіяльність людини60

ВИСНОВКИ 64

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 66

Додаток А Технічне завдання 69

Додаток Б Лістинг програми 74

Додаток В Заголовок додатку

					КС КРБ 123.XXX.00.00 ПЗ	Арк.
						7
№	Арж.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

ARM – Advanced RISC Machine

CPU – Central Processing Unit

HDMI – High Definition Multimedia Interface

NAS – Network Attached Storage

SoC – System-on-a-chip

USB – Universal Serial Bus

				КС КРБ 123.XXX.00.00 ПЗ	Арк.
					8
	докум.	Підпис	Дата		

ВСТУП

Цукровий діабет характеризується необхідністю регулярного введення інсуліну, що обумовлює потребу в систематичному контролі за виконанням графіка ін'єкцій. У разі порушення цього графіка виникає ризик погіршення стану пацієнта, що унеможлиблює ведення лікувального процесу без використання спеціалізованих технічних засобів контролю.

У разі амбулаторного лікування пацієнт самостійно виконує ін'єкції згідно з призначеним режимом. Це потребує засобу, який забезпечує нагадування про необхідність введення інсуліну відповідно до заданого розкладу. Технічна реалізація такого пристрою передбачає використання вбудованих систем з автономним живленням, модулем годинника реального часу, інтерфейсом для керування параметрами, енергонезалежною пам'яттю та звуковим або візуальним сигналом сповіщення.

Метою кваліфікаційної роботи є розробка комп'ютеризованої системи контролю введення ін'єкцій для осіб, які здійснюють введення інсуліну самостійно.

Проект передбачає апаратну та програмну реалізацію пристрою з урахуванням вимог до автономності, точності відліку часу, збереження даних та взаємодії з користувачем через інтерфейс. Передбачається використання мікроконтролера ESP32, інтеграція компонентів у єдину систему, застосування енкодера для керування параметрами, використання годинника реального часу (RTC) для підтримки точного розкладу, енергонезалежної пам'яті EEPROM для збереження налаштувань та рідкокристалічного дисплея для відображення інформації.

Визначені завдання:

- виконати аналіз технічного завдання;
- виконати аналіз можливих рішень реалізації;
- розробити структурну схему пристрою;
- розробити принципову схему пристрою;

				КС КРБ 123.235 .00.00 ПЗ	Арк.
					9
	№ докум.	Підпис	Дата		

- реалізувати апаратну та програмну частини системи;
- провести тестування на макетній платі.

Об'єкт дослідження — процес автоматизованого контролю дотримання розкладу введення ін'єкцій у користувацькому середовищі.

Предмет дослідження — технічна реалізація мікроконтролерної системи нагадування про ін'єкції з використанням ESP32, зовнішньої пам'яті EEPROM, годинника реального часу, інтерфейсу введення/виведення та засобів індикації.

					КС КРБ 123.235 .00.00 ПЗ	Арк.
						10
Арк.	№ докум.	Підпис	Дата			

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютерної системи контролю введення ін'єкцій

Комп'ютеризована система контролю введення ін'єкцій призначена для забезпечення моніторингу дотримання розкладу інсулінотерапії в умовах самостійного введення лікарських засобів. Система має функціонувати автономно та забезпечувати точне відображення часу, а також повідомлення користувача про необхідність виконання ін'єкції відповідно до заданого графіка.

Основною функціональною вимогою є забезпечення коректного обліку поточного часу, що вимагає застосування автономного джерела відліку з мінімальним відхиленням протягом тривалого періоду. Це передбачає наявність компонента, здатного зберігати точні часові дані навіть за відсутності зовнішнього живлення.

З боку програмного забезпечення необхідне зберігання множини нагадувань, кожне з яких характеризується принаймні параметрами години та хвилин. Для цього система повинна підтримувати роботу з енергонезалежною пам'яттю, у якій інформація зберігається без втрат після перезавантаження чи відключення живлення [1].

Для коректної взаємодії з користувачем потрібно реалізувати інтерфейс, що дозволяє налаштовувати час, вводити або змінювати нагадування, переглядати поточні параметри. Це передбачає наявність індикаторного пристрою та пристрою введення команд, які забезпечують базову навігацію між параметрами та можливість їх зміни.

					КС КРБ 123.235.00.00 ПЗ			
Зм.	Арх.	№ докум.	Підпис	Дата	Аналіз технічного завдання	Літ.	Арк.	Аркушів
		Окочинська	[Підпис]	06.06				
		Миницький Н.І.	[Підпис]	06.06			11	
		Морозова	[Підпис]	25.06		ТНТУ, каф. КС, гр. СІ-41		
		Місес Г.В.	[Підпис]	09.06				
		Осуківська Г.М.	[Підпис]	25.06				

Сповіщення про необхідність виконання ін'єкції має бути забезпечене на апаратному рівні за допомогою звукового або візуального сигналу. Сигнал має бути достатньо відчутним для користувача, щоб забезпечити своєчасну реакцію.

Живлення системи має бути автономним. Пристрій повинен зберігати працездатність у режимі очікування з низьким енергоспоживанням, а також підтримувати тривалу роботу без підключення до мережі живлення.

З точки зору обробки подій система повинна контролювати настання заданого часу нагадування, перевіряти поточний час з урахуванням циклу роботи та у разі збігу з нагадуванням активувати сигнал сповіщення. Після підтвердження сповіщення користувачем або по завершенню заданого інтервалу система повертається до режиму очікування.

Умови експлуатації вимагають забезпечення стабільної роботи системи за кімнатної температури, в побутовому середовищі. Рівень електромагнітного випромінювання повинен бути мінімальним, що обмежує використання надмірно інтенсивних джерел сигналів.

Технічна реалізація має враховувати обмежені апаратні ресурси, що характерно для вбудованих систем. Це обумовлює необхідність оптимізації використання пам'яті, обчислювальних ресурсів, енергоспоживання та мінімізації кількості периферійних модулів [2].

Збереження налаштувань нагадувань повинно бути гарантованим у разі повного знеструмлення пристрою. Це означає, що запис параметрів має виконуватись у момент їхнього внесення або редагування, з мінімізацією операцій запису для зменшення зносу енергонезалежної пам'яті.

Взаємодія користувача з системою має реалізовуватись через інтерфейс введення, що забезпечує зміну параметрів з мінімальною кількістю керувальних елементів. Для цього доцільно застосувати пристрій, що дозволяє як навігацію, так і підтвердження вибору.

Користувач має мати змогу в реальному часі переглядати поточний час, кількість заданих нагадувань, їхній список, час наступної ін'єкції та отримувати відповідні сигнали. Для цього необхідний дисплей, здатний відображати текстову або символічну інформацію.

Формат зберігання даних про нагадування має бути уніфікованим, що дозволяє зберігати й зчитувати час введення ін'єкцій в обмеженому за обсягом середовищі. Це передбачає двобайтову або подібну структуру даних з параметрами години та хвилин.

Система повинна передбачати можливість масштабування кількості нагадувань, щонайменше до десяти подій на добу. Усі події мають бути незалежними одна від одної та мати окремі часові мітки.

Реалізація має враховувати ймовірність помилок при зчитуванні або записі до пам'яті. Система повинна мати логіку перевірки коректності введених значень та уникати виходу за межі допустимих значень (0–23 години, 0–59 хвилин).

Алгоритм керування режимами повинен забезпечувати зміну станів пристрою відповідно до дій користувача. Це може бути реалізовано через послідовність етапів: очікування, редагування часу, редагування нагадувань, сигналізація, підтвердження.

Для покращення інформативності інтерфейсу передбачається можливість візуального виділення активного параметра, який редагується. Це досягається через зміну символів, що виводяться на дисплей, або позицію курсора.

У разі редагування часу або нагадувань необхідна підтримка інкрементного або декрементного керування значенням, що дозволяє змінювати параметри в межах допустимого діапазону без введення числового значення вручну.

Таким чином, система повинна забезпечувати безперебійну роботу в автономному режимі, підтримувати задану кількість нагадувань, дозволяти

					КС КРБ 123.235.00.00 ПЗ	Арк.
						13
Арк.	№ докум.	Підпис	Дата			

редагування параметрів, зберігати дані у постійній пам'яті та сигналізувати про необхідність введення ін'єкцій з точністю до хвилини.

1.2 Аналіз можливих рішень поставленого завдання

Рішення задачі контролю введення ін'єкцій реалізується за допомогою спеціалізованих технічних засобів, що забезпечують нагадування про необхідність інсулінотерапії. На ринку присутні пристрої, що виконують подібні функції, з використанням вбудованих мікроконтролерних систем та автономного живлення. Аналіз наявних рішень дозволяє визначити типові підходи до побудови таких пристроїв, основні функціональні блоки та варіанти реалізації інтерфейсу.

У загальному випадку технічна структура таких пристроїв включає годинник реального часу, джерело живлення, індикаторний модуль, пристрій введення, звуковий або візуальний модуль сигналізації, енергонезалежну пам'ять та мікроконтролер. Алгоритм роботи орієнтований на постійне відстеження поточного часу з порівнянням його з параметрами записаних нагадувань. У разі збігу система активує сповіщення. Для налаштування часу та редагування параметрів застосовуються кнопкові або енкодерні інтерфейси.

На ринку медичних товарів представлені окремі пристрої нагадування, які виконують подібні функції, проте мають обмежені можливості налаштування або інтегровані лише в певні моделі шприц-ручок. До таких пристроїв належать: MedAngel ONE, Timesulin та InsulClock. Усі вони реалізують базову функціональність контролю часу введення інсуліну, однак мають відмінності в архітектурі, можливості налаштувань та способах взаємодії з користувачем.

MedAngel ONE — це датчик температури та трекер інсуліну, який передає дані на смартфон. Пристрій орієнтований на контроль умов зберігання

інсуліну, але також має функціонал нагадувань через мобільний застосунок. Взаємодія з користувачем реалізується через Bluetooth-з'єднання та мобільну платформу. Вбудований дисплей відсутній. Нагадування налаштовуються в застосунку [3].

Після аналізу існуючих рішень, була створена порівняльна таблиця 1.1 для подальшого дослідження.

Таблиця 1.1 – Порівняння існуючих рішень

Характеристика	MedAngel ONE	Timesulin	InsulClock
Тип взаємодії	Через смартфон (BLE)	Автономний, без зв'язку	Через смартфон (BLE)
Відображення часу	Через застосунок	Дисплей з таймером	Через застосунок
Активне нагадування	Є (через застосунок)	Немає	Є (через застосунок)
Зміна параметрів	Через застосунок	Не підтримується	Через застосунок
Енергонезалежність	Часткова	Повна	Часткова
Автономність налаштувань	Немає	Немає	Немає
Вбудований дисплей	Немає	Є	Немає
Звукова сигналізація	Немає	Немає	Немає
Цільове застосування	Зберігання, нагадування	Облік часу останньої дози	Реєстрація ін'єкцій

Timesulin — кришка для шприц-ручки, яка відображає час, що минув з моменту останнього введення інсуліну. Пристрій не має активних нагадувань, проте дозволяє пасивно контролювати розклад, орієнтуючись на показник часу. Користувач має самостійно оцінювати, чи настав момент для чергової ін'єкції. Введення та зміна налаштувань відсутні [4].

InsulClock — модуль, що кріпиться до шприц-ручки. Пристрій фіксує час кожної ін'єкції, передає інформацію через Bluetooth до мобільного застосунку та формує звіт про історію введення. Вбудованого дисплея або елементів керування не має. Всі дії виконуються через мобільний пристрій [5].

Аналіз зазначених рішень показує, що наявні пристрої переважно орієнтовані на взаємодію з мобільним застосунком, що унеможливорює їх використання в умовах відсутності смартфона або стабільного з'єднання. У більшості з них не реалізовано автономного інтерфейсу введення/виведення, а також гнучкої системи налаштувань часу нагадувань без залучення сторонніх пристроїв.

У контексті поставленого завдання доцільною є розробка автономного пристрою з вбудованим інтерфейсом налаштування, індикацією поточного часу, збереженням нагадувань в незалежній пам'яті та роботою без підключення до мобільного пристрою. Такий підхід забезпечує незалежність системи від зовнішньої інфраструктури та підвищує надійність функціонування в умовах побутової експлуатації.

				КС КРБ 123.235 .00.00 ПЗ		Арк.
						16
Арк.	№ докум.	Підпис	Дата			

РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ КОНТРОЛЮ ВВЕДЕННЯ ІН'ЄКЦІЙ

2.1 Розробка узагальненої структури системи контролю введення ін'єкцій

Функціонування системи контролю введення ін'єкцій передбачає взаємодію між мікроконтролером та допоміжними електронними модулями, які забезпечують обробку даних, інтерфейс користувача, зберігання налаштувань та генерацію сповіщень. У загальному вигляді система може бути представлена як сукупність логічно пов'язаних компонентів, що виконують визначені функції згідно з технічним завданням.

На (рис. 2.1) представлено загальну структурну схему системи контролю введення ін'єкцій. Схема ілюструє функціональні блоки та їх зв'язки без зазначення конкретних типів компонентів. Зв'язки між блоками позначають напрями передачі керуючих та інформаційних сигналів. Це дозволяє сформувати загальне уявлення про архітектуру системи та визначити необхідність використання відповідних інтерфейсів.

				КС КРБ 123.235 .00.00 ПЗ						
	Арк.	№ докум.	Підпис	Дата	Проектвання системи контролю введення ін'єкцій			Літ.	Арк.	Аркушів
		Скочинський	ВВ	16.06						
		Шендеров Н.З.	НЗ	16.06					17	
		Матвишин С.В.	СВ	15.06						
		Місєєв С.В.	СВ	9.06						
		Осухівська Г.М.	ГМ	15.06						
					ТНТУ, каф. КС, гр. СІ-41					

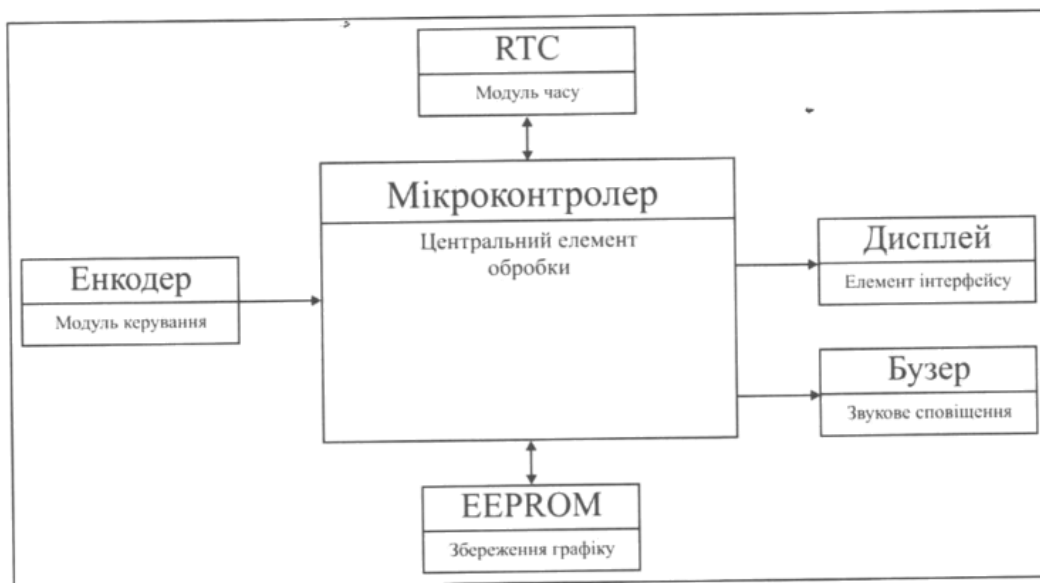


Рисунок 2.1 – Узагальнена структурна схема системи контролю введення ін'єкцій

Центральним елементом структури є мікроконтролер, який виконує обробку поточного часу, контроль розкладу нагадувань, генерацію керуючих сигналів до пристроїв індикації та взаємодію з користувачем через інтерфейс введення та виведення даних. В якості апаратної платформи для мікроконтролера передбачається використання системи-на-кристалі з вбудованими периферійними інтерфейсами [6], що дозволяють підключення зовнішніх пристроїв за допомогою послідовних шин (I²C, SPI, GPIO тощо).

Модуль годинника реального часу відповідає за автономне відстеження поточного часу незалежно від стану живлення основного контролера. Його інтерфейс взаємодіє з мікроконтролером для отримання точних даних про дату та час, необхідних для перевірки відповідності поточного моменту запланованому часу ін'єкції.

Для забезпечення збереження розкладу нагадувань між перезапусками системи застосовується модуль енергонезалежної пам'яті типу EEPROM [7]. Цей модуль дозволяє зчитування та запис конфігураційних параметрів без втрати даних після вимкнення живлення. Мікроконтролер звертається до

нього у процесі роботи для зчитування графіка нагадувань та оновлення даних відповідно до змін, які були внесені користувачем.

Інтерфейс користувача представлений рідкокристалічним дисплеєм та пристроєм керування. Дисплей виконує функції відображення інформації про поточний час, активні нагадування та параметри меню налаштувань. Пристрій керування, що базується на енкодері або комбінації кнопок, забезпечує навігацію в меню, зміну часу ін'єкцій та інші параметри.

Функція сповіщення реалізується за допомогою звукового сигналу, що генерується бужером, підключеним до одного з цифрових виходів мікроконтролера. Активація звукового сигналу виконується програмно в момент настання часу, що відповідає записаному нагадуванню.

Розробка узагальненої структури дозволяє виділити основні функціональні модулі системи та визначити напрямки подальшого проєктування, зокрема вибору елементної бази, розробки принципової схеми та програмного забезпечення.

2.2 Обґрунтування вибору апаратного забезпечення

2.2.1 Вибір мікроконтролерної плати

Для реалізації комп'ютеризованої системи контролю введення ін'єкцій було обрано мікроконтролер ESP32-Wroom-32, який розроблений компанією Espressif Systems. Це популярний чіп для вбудованих систем, що має широкий спектр застосувань завдяки багатьом інтерфейсам і можливостям для розширення функціональності. Зовнішній вигляд цієї плати наведено на рисунку 2.2.

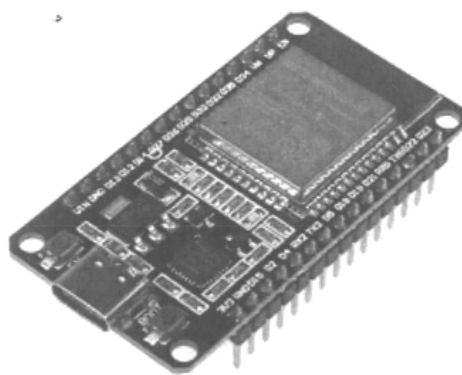


Рисунок 2.2 – Зовнішній вигляд ESP32-Wroom-32

ESP32-Wroom-32 містить двоядерний процесор з тактовою частотою до 240 МГц, що дозволяє виконувати обчислювальні операції в реальному часі. Завдяки такій обчислювальній потужності чіп здатен одночасно обробляти дані від датчиків, керувати відображенням на дисплеї, зберігати та отримувати дані з пам'яті, а також взаємодіяти з іншими компонентами системи.

У складі чіпа ESP32-Wroom-32 є вбудовані модулі пам'яті Flash об'ємом 4 МБ, що дозволяє зберігати програмний код і налаштування користувача. Крім того, плата підтримує підключення зовнішніх модулів пам'яті, зокрема EEPROM, для збереження даних, таких як графік введення ін'єкцій. Це дозволяє зберігати інформацію навіть при відключенні живлення, що підвищує надійність системи.

ESP32-Wroom-32 має вбудовані 12-розрядні аналого-цифрові перетворювачі (АЦП), які дозволяють здійснювати вимірювання аналогових сигналів. Хоча в рамках поточної реалізації проєкту АЦП не використовуються, їх наявність може бути корисною при подальшому розширенні функціональності системи, наприклад, для зчитування даних від додаткових сенсорів або датчиків.

Плата містить понад 30 пінів, що забезпечують підключення до різноманітних зовнішніх компонентів. Піновий роз'єм плати розрахований на рівень напруги 3.3 В, що потрібно враховувати при підключенні периферійних пристроїв, оскільки більшість компонентів працюють при цьому рівні

напруги. Окремі піни підтримують виходи цифрових та аналогових сигналів, а також інтерфейси I2C, SPI, UART, що забезпечує гнучкість у виборі компонентів для підключення до системи [8]. Розташування пнів наведено на рисунку 2.3.

ESP32 PINOUT DIAGRAM

ESP32-WROOM-32

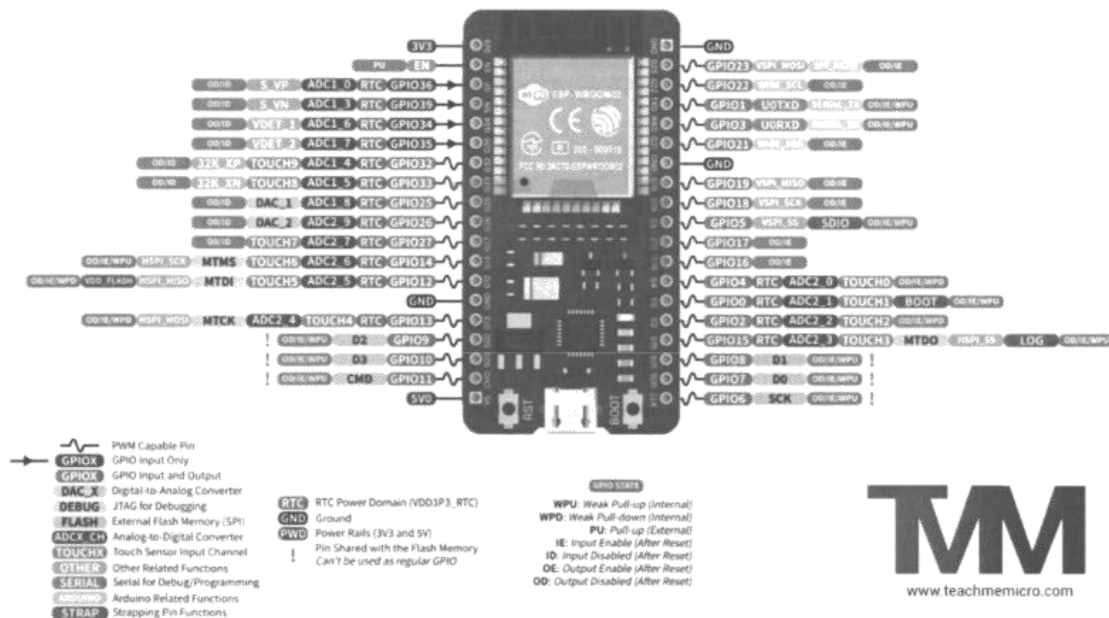


Рисунок 2.3 – Розташування виводів плати ESP32-Wroom-32

Для програмування ESP32-Wroom-32 використовуються стандартні інтерфейси, такі як USB-UART або через програматор. Плата має вбудований USB-порт для підключення до комп'ютера та програмування, що спрощує процес розробки та налаштування. Програмування здійснюється через середовище розробки, таке як Arduino IDE або ESP-IDF [9], що дає змогу розробляти, тестувати та деплоїти програмний код.

ESP32-Wroom-32 також підтримує бездротові технології Wi-Fi і Bluetooth, що дозволяє інтегрувати систему в більші мережі або забезпечити

віддалений доступ до пристрою, проте ці функції наразі не використовуються в проєкті.

2.2.2 Вибір модуля годинника реального часу

Для реалізації функціональності годинника реального часу в системі контролю введення ін'єкцій було обрано модуль TinyRTC з RTC DS1307 й EEPROM 24C32, які підключаються через інтерфейс I2C. Цей модуль забезпечує відстеження часу і зберігання даних, навіть при відключенні живлення. Зовнішній вигляд модуля TinyRTC зображений на рисунку 2.4.

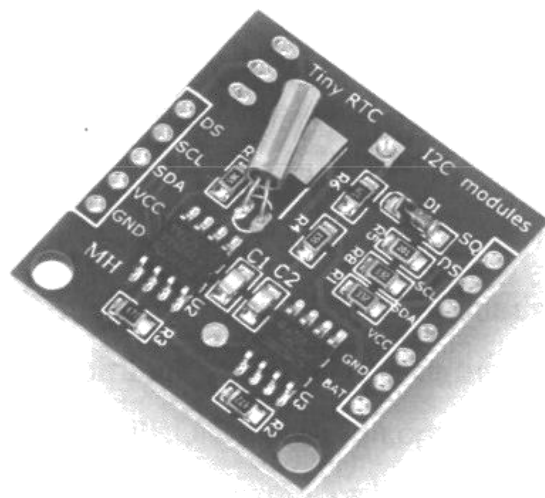


Рисунок 2.4 – Зовнішній вигляд модуля RTC DS1307 з EEPROM 24C32

Модуль DS1307 у складі TinyRTC є інтегрованим чіпом для відліку часу в реальному часі. Він використовує кварцовий осцилятор для забезпечення стабільності та точності часу. Модуль здатний зберігати час і дату, причому підтримує формат року, місяця, дня, години, хвилин і секунд. Для підтримки роботи без живлення на платі передбачена можливість підключення зовнішнього джерела живлення у вигляді батареї типу CR2032, що дозволяє зберігати час, навіть якщо основне живлення відсутнє. Це забезпечує

безперервну роботу годинника і збереження даних про час. Схема підключення чіпа DS1307 у складі даного модуля наведена на рисунку 2.5 [10].

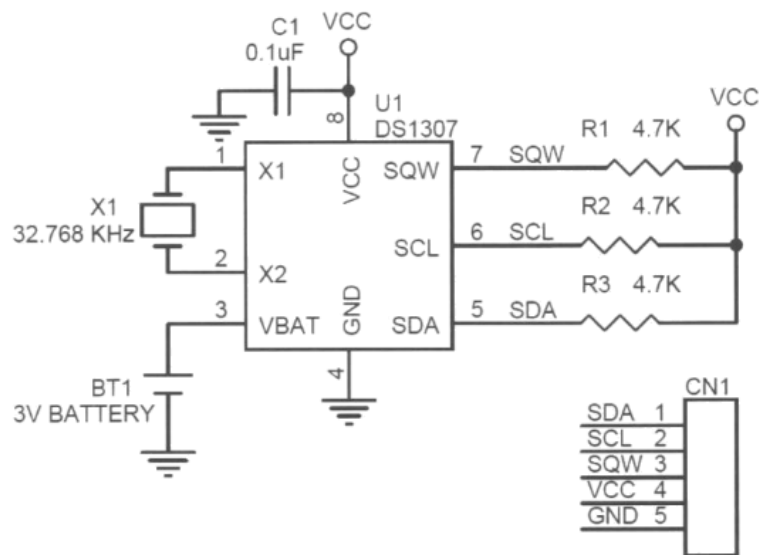


Рисунок 2.5 – Схема підключення чіпа DS1307 у складі модуля годинника реального часу

Модуль TinyRTC також включає EEPROM пам'ять об'ємом 32 кБ, яка використовується для зберігання даних, таких як налаштування системи або розклад введення ін'єкцій. EEPROM 24C32 використовує 2-байтову адресацію, що означає, що для доступу до кожної окремої комірки пам'яті необхідно вказати 16-бітну адресу [11]. Така адресація дозволяє організувати зберігання даних у пам'яті та забезпечує можливість доступу до великої кількості інформації.

Модуль DS1307 має кілька основних виводів, через які здійснюється комунікація з іншими компонентами системи.

Принципова схема TinyRTC наведена на рисунку 2.6.

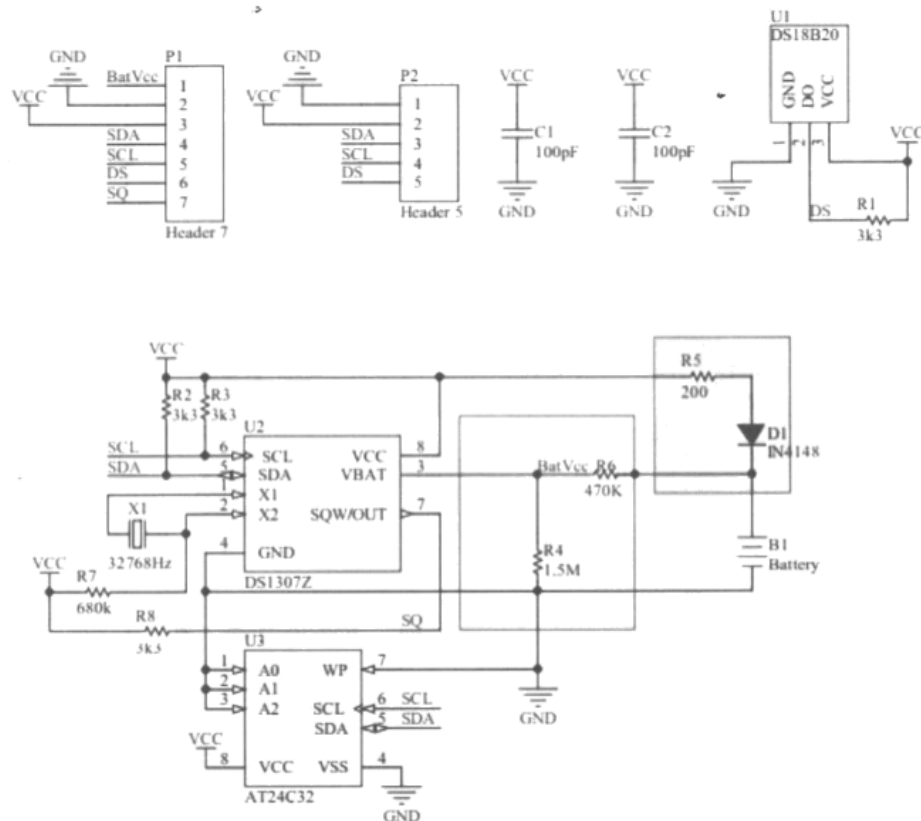


Рисунок 2.6 – Принципова схема модуля TinyRTC

Виводи SDA (дані) та SCL (годинниковий сигнал) використовуються для передачі даних через інтерфейс I2C. Вивід SQ (секундний імпульс) генерує сигнал, що може використовуватися для синхронізації або в інших специфічних застосуваннях. Вивід DS є виходом для підключення датчика температури, що дозволяє вимірювати температуру навколишнього середовища. Зазначений датчик може використовуватися для коригування похибки годинника, оскільки температура може впливати на точність кварцового осцилятора. Однак у рамках поточної системи цей вивід не буде використовуватися, хоча його можливості можуть бути реалізовані в подальших версіях системи.

Вивід BAT призначений для підключення батареї типу CR2032, яка живить модуль в умовах відключення основного джерела живлення [12]. Це забезпечує збереження налаштувань та правильне відстеження часу навіть під час перебоїв з живленням.

2.2.3 Вибір дисплейного модуля

Для відображення інформації у комп'ютеризованій системі контролю введення ін'єкцій було обрано модуль LCD 1602 з підключенням через інтерфейс I2C за допомогою адаптера HW-061.

На рисунку 2.7 наведено зовнішній вигляд модуля LCD 1602 з адаптером HW-061.

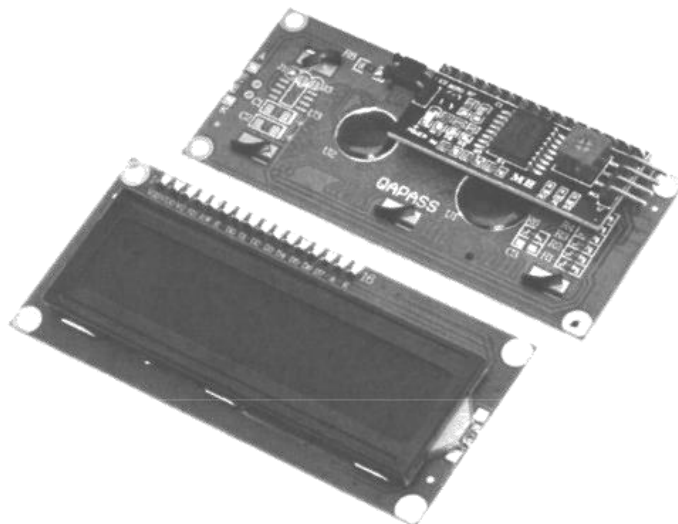
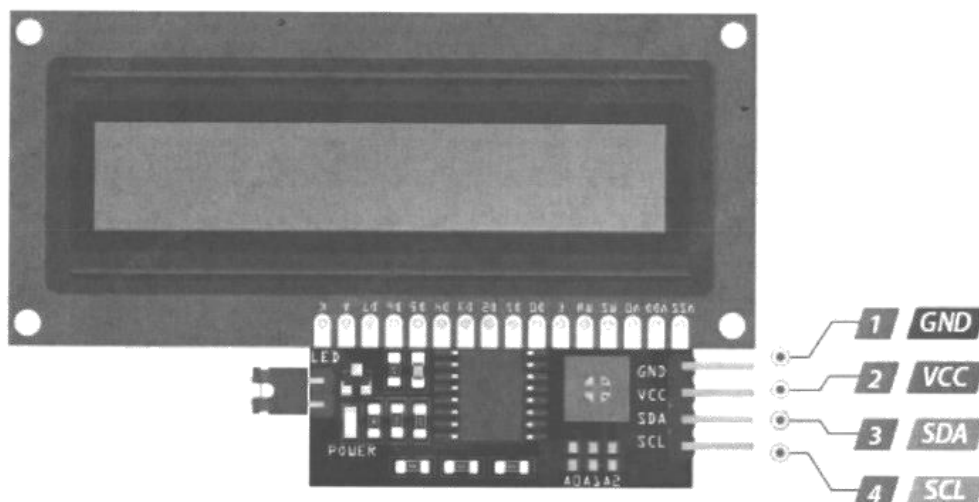


Рисунок 2.7 – Зовнішній вигляд модуля LCD 1602 з адаптером HW-061

Дисплей 1602 є символьним рідкокристалічним індикатором, що здатен відображати 2 рядки по 16 символів. Кожен символ формується у матриці розміром 5×8 точок.

Базовий модуль 1602 має 16 виводів, однак за наявності адаптера HW-061 реалізується підключення через шину I2C, що дозволяє скоротити кількість необхідних пінів мікроконтролера до двох: SDA (Serial Data Line) і SCL (Serial Clock Line). В адаптері HW-061 використовується мікросхема PCF8574, яка виконує функцію розширення введення/виведення та інтерфейсу між мікроконтролером і контролером дисплея HD44780 [13].

На рисунку 2.8 показано розташування виводів інтерфейсного адаптера HW-061 та його зв'язок з контролером дисплея.



I2C LCD Pinout



Рисунок 2.8 – Розташування виводів адаптера HW-061 та його зв'язок з контролером дисплея

Передача даних до дисплея здійснюється через I2C-шину за допомогою адресації. Стандартна адреса I2C для модуля становить 0x27 або 0x3F, залежно від ревізії модуля. Передача символів та команд відбувається з використанням спеціального формату команд для контролера HD44780. Дисплей підтримує режим 4-бітної передачі даних, що також знижує навантаження на мікроконтролер у порівнянні з прямим підключенням.

Живлення модуля дисплея здійснюється від 5 В. Попри це, рівні сигналів I2C адаптеру сумісні з 3.3 В, що дозволяє підключати модуль безпосередньо до виводів ESP32. На платі адаптера передбачено потенціометр, який дозволяє регулювати контрастність зображення. Існує також вивід для керування підсвічуванням, який можна відключити для зменшення енергоспоживання.

2.2.4 Вибір пристрою керування

Для керування комп'ютеризованою системою контролю введення ін'єкцій було обрано обертовий енкодер типу KY-040. З урахуванням вимог до компактності та обмеженої кількості доступних апаратних ресурсів, було

прийнято рішення не використовувати окремі кнопки або клавіатури, а реалізувати функції введення та навігації за допомогою одного пристрою. На рисунку 2.9 представлено зовнішній вигляд енкодера KY-040.

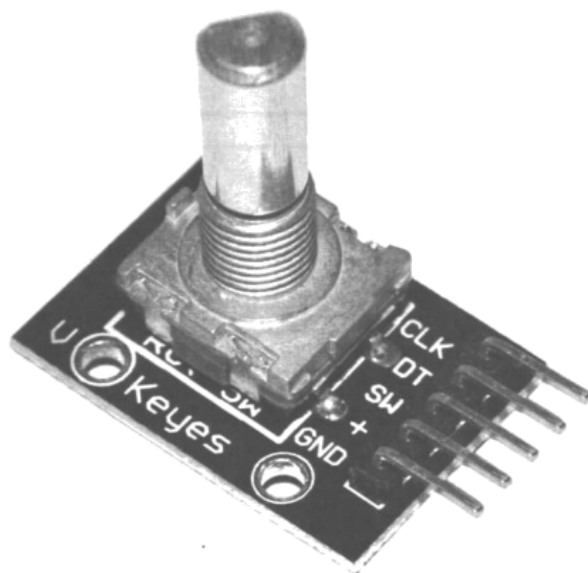


Рисунок 2.9 – Зовнішній вигляд енкодера KY-040

Енкодер KY-040 складається з обертового диска з контактними доріжками, який дозволяє визначати напрямок і кількість обертів, та вбудованої кнопки, що активується натисканням на вісь. Пристрій має п'ять виводів: CLK (сигнал імпульсів при обертанні), DT (сигнал напрямку обертання), SW (вивід натискання кнопки), а також живлення GND і VCC. Робоча напруга складає 5 В, однак сигнальні виводи сумісні з рівнями 3.3 В, що дозволяє підключення безпосередньо до мікроконтролера ESP32.

На рисунку 2.10 показано схему внутрішнього підключення енкодера KY-040, яка демонструє електричні зв'язки між контактами живлення, виходами CLK і DT, кнопкою SW та внутрішніми резисторами, що формують стандартну конфігурацію для цифрового зчитування імпульсів та натискань [14].

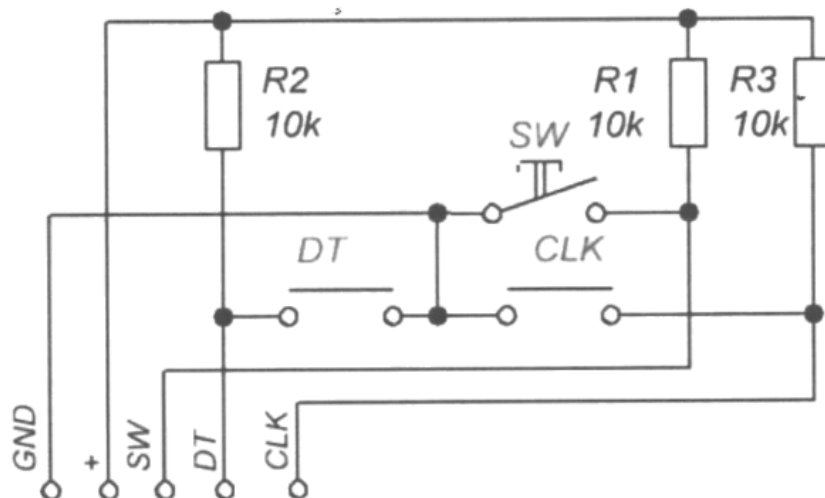


Рисунок 2.10 – Схема внутрішнього підключення енкодера KY-040

Зчитування сигналів здійснюється за допомогою опитування або переривань. При обертанні енкодера формується послідовність імпульсів на лініях CLK і DT. Аналізуючи порядок змін цих сигналів, мікроконтролер визначає напрямок обертання. Натискання кнопки SW використовується для активації вибраного пункту меню або підтвердження параметра.

2.2.5 Вибір модуля звукового сповіщення

Для реалізації звукових сповіщень у системі було обрано модуль МН-FMD з п'єзоелектричним випромінювачем. Його зовнішній вигляд наведено на рисунку 2.11, а внутрішнє підключення на рисунку 2.12.

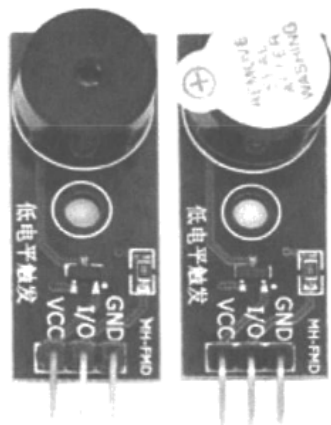


Рисунок 2.11 – Зовнішній вигляд модулю МН-FMD

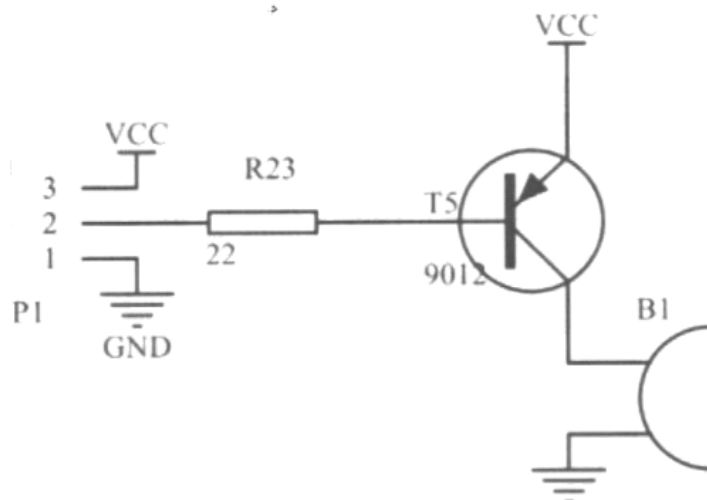


Рисунок 2.12 – Внутрішнє підключення модулю MH-FMD

Конструктивно пристрій містить три виводи: живлення (+), землю (–) та керуючий вхід (I/O). Модуль забезпечує генерування звукового сигналу при подачі відповідного електричного керуючого імпульсу на вхід I/O.

З аналізу конструкції видно наявність дискретних компонентів, зокрема транзистора, що виконує функцію підсилювального елемента та комутатора. Характерною особливістю модуля є інверсна логіка керування: при подачі логічного нуля на вхід I/O активується генерація звуку. Це може бути пов'язано з внутрішнім підключенням бази транзистора через резистор до позитивної шини живлення, що формує інверсну реакцію на вхідний сигнал [15].

Даний тип модулів використовується для створення коротких або довготривалих звукових сповіщень, необхідних для подачі індикації користувачу про події, такі як час введення ін'єкції.

2.3 Обґрунтування вибору програмного забезпечення

Для розробки програмного забезпечення комп'ютеризованої системи контролю введення ін'єкцій було обрано середовище розробки Arduino IDE.

Arduino IDE є популярною платформою для розробки програмного забезпечення для мікроконтролерів. Це середовище надає зручний інтерфейс для програмування, компіляції та завантаження коду на мікроконтролери різних типів, включаючи ESP32 [16]. Arduino IDE підтримує мови програмування, зокрема C та C++, що дозволяє працювати на низькому рівні з периферійними пристроями.

Однією з переваг Arduino IDE є наявність великої кількості бібліотек, розроблених для роботи з різними периферійними пристроями, що значно знижує час, необхідний для розробки програм. Бібліотеки для роботи з дисплеями, датчиками, модулями зв'язку та іншими пристроями інтегровані в середовище та дозволяють знижувати складність програмування. Arduino IDE підтримує стандартні функції, що дозволяють користувачам зосередитися на вирішенні завдань, пов'язаних із взаємодією з апаратними частинами, а не на низькорівневому програмуванні мікроконтролерів.

Мова програмування C++ є основною для Arduino IDE. Ця мова дозволяє працювати з низьким рівнем апаратного забезпечення, надаючи можливість прямого доступу до пам'яті, портів вводу/виводу та інших апаратних ресурсів. Завдяки C++ можна реалізувати гнучку та компактну логіку роботи з мікроконтролером. Arduino IDE включає інтегровану бібліотеку стандартних функцій, що забезпечують простоту написання коду для взаємодії з периферією, одночасно даючи можливість налаштовувати та адаптувати програму до конкретних умов.

Для реалізації даної системи були використані бібліотеки, що оптимізують роботу з периферією, зокрема бібліотека Wire.h для роботи з шиною I2C, LiquidCrystal_I2C.h для відображення інформації на дисплеї, ESP32Encoder.h для обробки сигналів енкодера, а також RTCLib.h для роботи з годинником реального часу. Wire.h дозволяє працювати з пристроями, підключеними по I2C, такими як годинники реального часу або EEPROM. Ця бібліотека забезпечує просту реалізацію обміну даними між

мікроконтролером та іншими компонентами. LiquidCrystal_I2C.h забезпечує зручну роботу з рідкокристалічними дисплеями, що використовують інтерфейс I2C, і дає можливість виводити на екран дані, що надаються користувачеві. ESP32Encoder.h [17] розроблена спеціально для мікроконтролерів ESP32 і дозволяє коректно працювати з енкодерами, враховуючи особливості роботи ESP32. Це дає можливість точно фіксувати зміни положення енкодера, що треба для налаштування параметрів системи. RTCLib.h [18] використовує модуль DS1307 для забезпечення точного відліку часу і надає функції для роботи з годинниками реального часу, що зберігають дату та час навіть при відключенні живлення.

Вибір середовища Arduino IDE для розробки програмного забезпечення для даного проєкту обумовлений зручністю використання, наявністю необхідних бібліотек для інтеграції з периферійними пристроями, а також підтримкою програмування на мові C++, що дозволяє реалізувати доступ до апаратних ресурсів мікроконтролера.

2.4 Проектування комп'ютерного засобу для контролю введення ін'єкцій

Проектування комп'ютерного засобу для контролю введення ін'єкцій передбачає реалізацію взаємодії між компонентами системи через конкретно визначені інтерфейси зв'язку та пінів мікроконтролера ESP32. Для забезпечення роботи системи розроблені структурна та принципова схеми підключення, які визначають з'єднання між основними пристроями та мікроконтролером. Розглянемо кожну з них детальніше.

Структурна схема (рис. 2.13) відображає загальну архітектуру системи та взаємодію компонентів.

					КС КРБ 123.235.00.00 ПЗ	Арк.
						31
Арк.	№ докум.	Підпис	Дата			

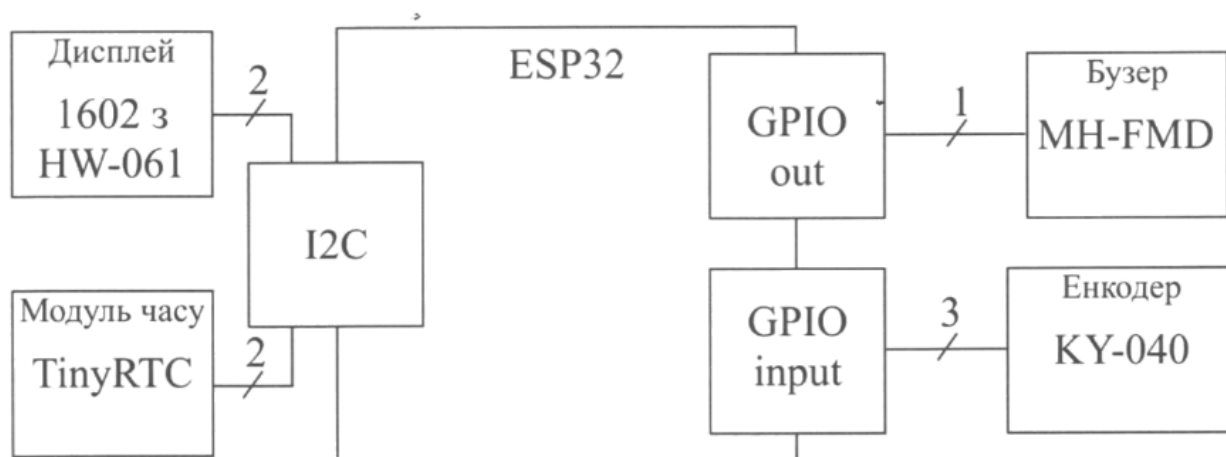


Рисунок 2.13 – Структурна схема приладу

Структурна схема показує підключення дисплея, годинника реального часу, звукового сповіщувача та енкодера до ESP32. Комунікація між ESP32 та дисплеєм LCD, а також з модулем TinyRTC, здійснюється через шину I2C [19]. Це дозволяє зменшити кількість проводів, необхідних для підключення цих пристроїв до мікроконтролера. Звуковий сповіщувач (MH-FMD) підключається до одного з пінів ESP32 для виведення сигналів, які ініціюють його роботу. Вхідні сигнали від енкодера KY-040 підключаються до пінів ESP32 для обробки руху енкодера, що використовується для налаштування параметрів роботи системи.

Принципова схема (рис. 2.14) розроблена засобами KiCad [20] й надає детальну інформацію про фізичні з'єднання пінів ESP32 з підключеними пристроями.

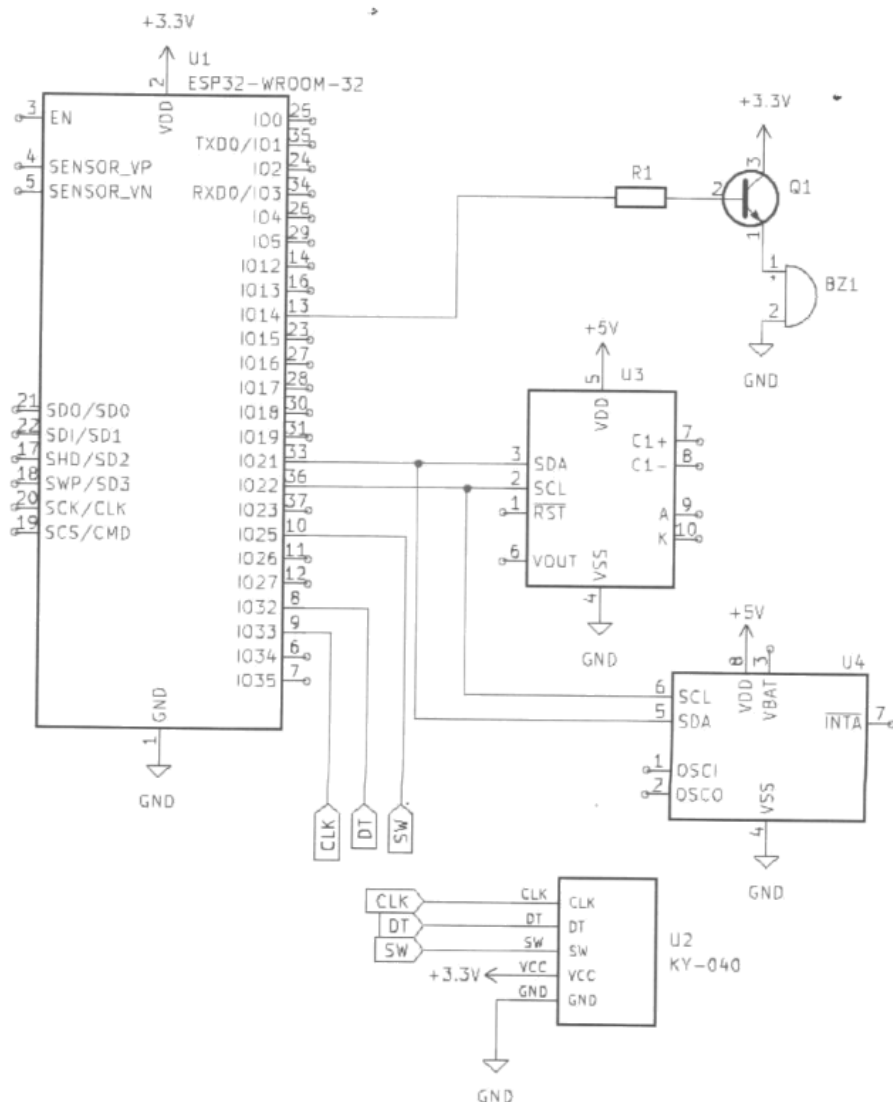


Рисунок 2.14 – Принципова схема приладу

Підключення SDA та SCL пінів ESP32 до відповідних пінів на дисплеї LCD та TinyRTC забезпечує передачу даних між ними за допомогою шини I2C. Це дозволяє контролювати виведення інформації на дисплей, а також працювати з годинником реального часу для синхронізації часу та зберігання даних при відключенні живлення. Для роботи з звуковим сповіщувачем MH-FMD використовуються GPIO пін ESP32, що дає змогу генерувати звукові сигнали при виконанні певних умов. Піни ESP32 25, 32 та 35 з'єднуються з енкадером KY-040 для зчитування руху енкадера, що використовується для налаштування часу введення ін'єкцій або інтервалів нагадувань.

Живлення компонентів системи забезпечується різними джерелами в залежності від вимог кожного пристрою. Дисплей LCD потребує живлення від 5 В, тоді як інші компоненти, такі як енкодер KY-040, звуковий сповіщувач MH-FMD та модуль TinyRTC, працюють при живленні 3.3 В.

Таким чином, проектування комп'ютерного засобу для контролю введення ін'єкцій передбачає правильне вибір і підключення компонентів, що дозволяє забезпечити надійну взаємодію між пристроями та роботу системи в цілому.

				КС КРБ 123.235.00.00 ПЗ	Арк.
					34
Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ВВЕДЕННЯ ІН'ЄКЦІЇ

3.1 Реалізація апаратної частини

Апаратну частину комп'ютерного засобу реалізовано на макетній платі шляхом підключення всіх компонентів відповідно до розробленої принципової схеми. Для з'єднання використовувались провідники типу Dupont, що забезпечують рознімне підключення без потреби пайки. Живлення подавалось від зовнішнього стабілізованого джерела напругою 5 В через роз'єм живлення макетної плати. Також передбачено можливість живлення від USB інтерфейсу мікроконтролера ESP32.

Мікроконтролер ESP32 встановлено на макетній платі через відповідний адаптер. Для реалізації інтерфейсу I2C використовуються виводи ESP32: GPIO 21 — як лінія даних SDA, GPIO 22 — як лінія синхронізації SCL. До цих виводів приєднано два пристрої: модуль годинника реального часу TinyRTC та дисплейний модуль 1602 з I2C-адаптером HW-061. З'єднання виконано послідовно, з урахуванням того, що обидва пристрої працюють через одну шину. Логічні рівні для I2C-інтерфейсу становлять 3.3 В, однак модулі мають інтегровані перетворювачі рівнів, що дозволяє підключати їх без додаткових узгоджуючих схем.

Живлення для модуля дисплея подається від лінії +5 В, оскільки I2C-адаптер HW-061 розрахований на такий рівень живлення. Живлення модуля TinyRTC також подається від лінії +5 В, що відповідає вимогам до роботи як DS1307, так і інтегрованої пам'яті EEPROM. Водночас для збереження часу при відсутності живлення передбачено встановлення елемента живлення типу CR2032 у відповідний тримач на модулі.

				КС КРБ 123.235.00.00 ПЗ			
Арк.	№ докум.	Підпис	Дата				
1	00000001	Осипенко	26.06	Реалізація системи комп'ютерного введення ін'єкції			
2	00000002	Шиндиг Н.І.	26.06				
3	00000003	Шиндиг С.В.	25.06				
4	00000004	Шиндиг С.В.	25.06				
5	00000005	Шиндиг С.В.	25.06	ТНТУ, каф. КС, гр. СІ-41			
6	00000006	Осипівська Г.М.	25.06				

Модуль звукового сповіщення MH-FMD з'єднано з ESP32 через GPIO 14. На модулі реалізована схема на транзисторі, яка вмикає п'єзогенератор [21] при подачі логічного нуля на керуючий вхід. Це зумовлено інверсною логікою роботи пристрою.

Живлення модуля подається від лінії 3.3 В, відповідно до його специфікації. Для введення інформації та взаємодії з користувачем використовується енкодер KY-040. Його механічна конструкція дозволяє виявляти обертання та натискання кнопки. Вивід SW енкодера підключено до GPIO 25 мікроконтролера ESP32. Лінії CLK та DT підключено до пінів GPIO 35 та GPIO 32 відповідно. Ці входи використовуються для зчитування напрямку обертання енкодера. Живлення енкодера підключено до лінії 3.3 В, що відповідає логічним рівням ESP32.

Загальна збірка системи виконана на одній макетній платі без застосування пайки. Усі з'єднання прокладені вручну, з дотриманням вимог до ізоляції та мінімізації довжини сигнальних ліній для зменшення електричних перешкод. На рисунку 3.1 наведено зовнішній вигляд зібраної апаратної частини системи контролю введення ін'єкцій.

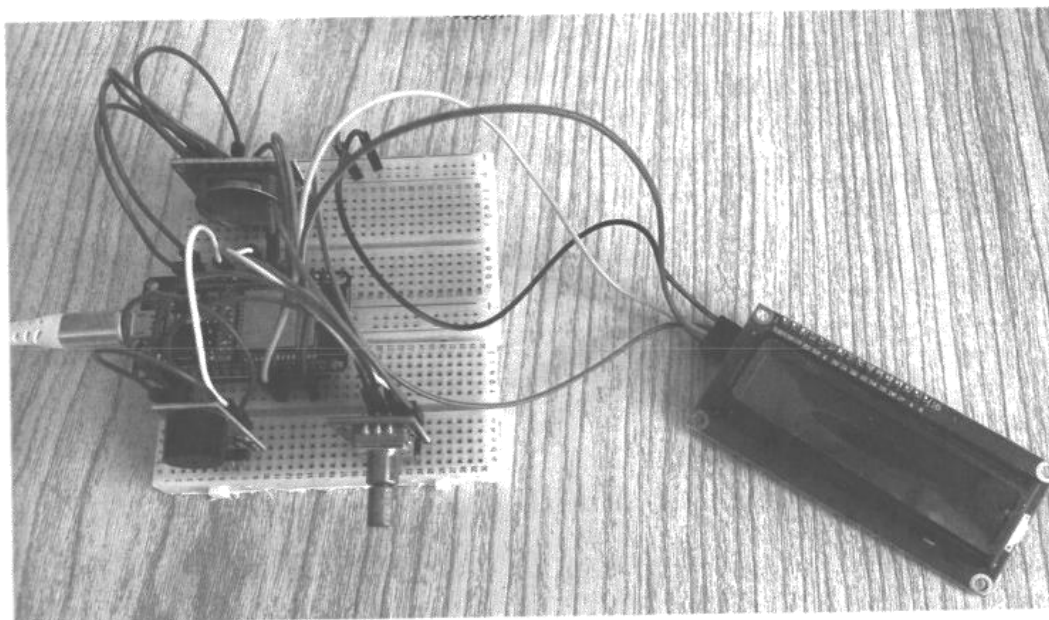


Рисунок 3.1 – Зовнішній вигляд спроектованої системи

3.2 Реалізація програмної частини

Програмна частина реалізує функціонування нагадувань та взаємодію користувача із системою через багаторівневе меню, що керується за допомогою енкодера з кнопкою. Алгоритм роботи системи реалізований у вигляді нескінченного циклу, в якому здійснюється опитування годинника реального часу, обробка введення з енкодера та перехід між режимами відповідно до поточного стану системи.

Інтерфейс взаємодії представлений у вигляді набору режимів роботи, кожен з яких відповідає окремій функціональності. Структура меню відображена на рисунку 3.2.

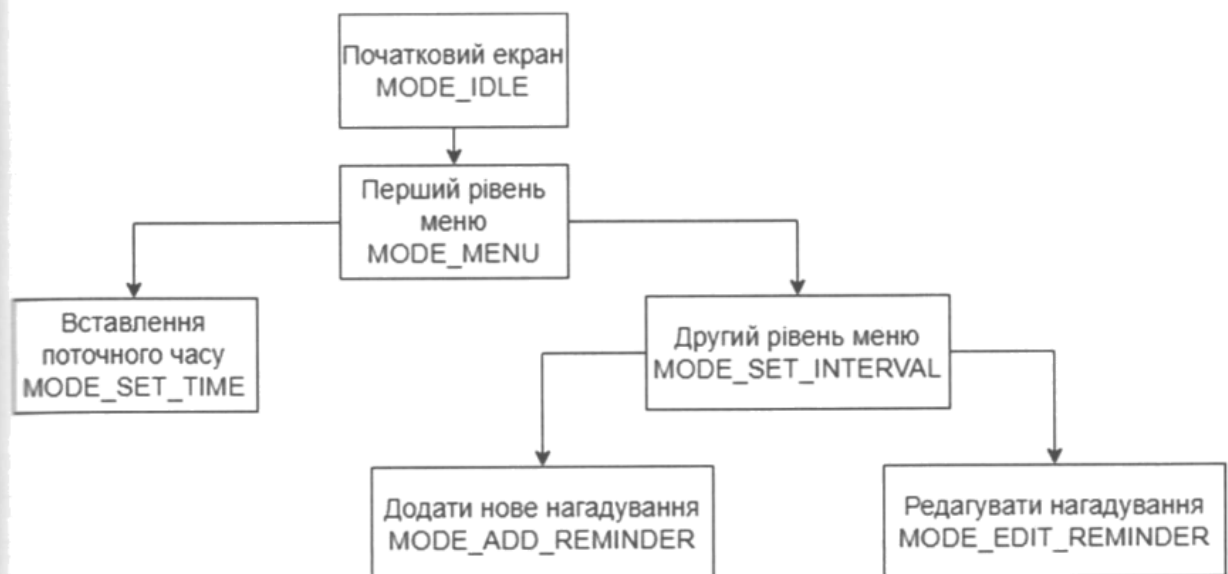


Рисунок 3.2 – Структура меню

У режимі очікування `MODE_IDLE` на дисплей виводиться поточний час, зчитаний з модуля RTC. Перехід до меню виконується натисканням кнопки енкодера. Меню першого рівня відповідає режиму `MODE_MENU` і містить основні пункти налаштувань.

У режимі `MODE_SET_TIME` користувач встановлює поточний час системи. Встановлені значення передаються модулю `RTC` та зберігаються у його внутрішній пам'яті.

Меню другого рівня `MODE_SET_INTERVAL` відповідає за керування системою нагадувань. У цьому режимі реалізовано два підрежими:

— `MODE_ADD_REMINDER` — додавання нового нагадування до списку. Користувач встановлює час спрацювання, після чого дані зберігаються у `EEPROM`.

— `MODE_EDIT_REMINDER` — перегляд та редагування наявних нагадувань. Доступна зміна часу спрацювання або повне видалення нагадування з пам'яті.

Всі режими реалізовано як незалежні функціональні блоки, між якими здійснюється перехід на основі введення з енкодера. У підрозділах нижче наведено структури даних, логіку опитування часу, алгоритм сигналізації та описано обробку введення користувача.

Реалізація перемикання між режимами взаємодії з користувачем та обробки поточної логіки здійснюється в основному циклі програми. Основний цикл побудований з використанням конструкції `switch/case`, яка виконує виклик відповідної функції обробки в залежності від значення змінної `currentMode`. Структуру головного циклу зображено на рисунку 3.3.

Функції `handleEncoder()` та `handleButton()` реалізують обробку фізичного введення з енкодера: зчитування обертання та натискання кнопки. Ці дії оновлюють внутрішній стан системи та можуть спричинити зміну режиму, що, у свою чергу, впливає на подальше виконання програми в межах конструкції `switch/case`.

Кожен режим має окрему функцію обробки виду `handleModeX()`, яка реалізує специфічну логіку взаємодії з користувачем та зміну станів у межах відповідного режиму. У наступних розділах описується вміст кожної функції обробки режиму, а також логіка опрацювання введення з енкодера.

```

void loop() {
    handleEncoder();
    handleButton();
    switch (currentMode) {
        case MODE_IDLE:
            handleModeIdle();
            break;
        case MODE_MENU:
            handleMenuMode();
            break;
        case MODE_SET_TIME:
            handleSetTimeMode();
            break;
        case MODE_SET_INTERVAL:
            handleSetIntervalMode();
            break;
        case MODE_ADD_REMINDER:
            handleAddReminderMode();
            break;
        case MODE_EDIT_REMINDER:
            handleEditReminderMode();
            break;
    }
}

```

Рисунок 3.3 – Лістинг головного циклу програми

Обробка обертання енкодера реалізована у функції `handleEncoder()`. Ця функція зчитує поточне положення енкодера, порівнює його з попереднім та, у разі зміни, виконує оновлення стану інтерфейсу. Код функції наведено на рисунку 3.4.

```

void handleEncoder() {
    int newPosition = encoder.getCount();
    if (newPosition != prevPosition) {
        buzz(2000, 100);
        int delta = newPosition - prevPosition;
        prevPosition = newPosition;
        if (currentMode == MODE_MENU) {
            menuIndex += delta;
            if (menuIndex < 0) menuIndex = 0;
        }
        lcd.clear();
    }
}

```

Рисунок 3.4 – Лістинг методу обробки обертання енкодера

У межах функції `handleEncoder()` відбувається зчитування нового значення позиції енкодера за допомогою методу `getCount()`. Якщо зафіксовано зміну значення, що свідчить про обертання ручки енкодера, викликається функція звукового сигналу `buzz()` з параметрами частоти та тривалості. Це дозволяє користувачеві отримати зворотний зв'язок про взаємодію з інтерфейсом.

Різниця між поточним і попереднім положенням енкодера обчислюється у вигляді змінної `delta`, яка відображає напрямок і величину обертання. Значення попередньої позиції (`prevPosition`) оновлюється поточним значенням для забезпечення коректної обробки наступних змін.

У випадку, якщо поточний режим роботи пристрою відповідає значенню `MODE_MENU`, відбувається зміна індексу активного пункту меню (`menuIndex`) відповідно до значення `delta`. Передбачена перевірка на обмеження знизу для уникнення переходу в область від'ємних індексів.

Після кожного виявленого обертання енкодера відбувається очищення дисплея викликом функції `lcd.clear()`. Це необхідно для оновлення графічного інтерфейсу та виведення актуальної інформації відповідно до зміненого стану меню.

Обробка натискання кнопки енкодера реалізована у функції `handleButton()`. Її функціональне призначення полягає у визначенні типу натискання — короткого або тривалого — шляхом аналізу часової різниці між моментами натискання та відпускання. У цьому методі реалізовано логіку виявлення змін стану кнопки, фіксацію часу початку натискання, генерацію звукового сигналу (через виклик `buzz()`) та встановлення відповідних програмних прапорців (`shortPressDetected`, `longPressDetected`). Код функції наведено на рисунку 3.5.

					КС КРБ 123.235.00.00 ПЗ	Арк.
						40
Арк.	№ докум.	Підпис	Дата			

```

void handleButton() {
    static bool lastButtonState = HIGH;
    bool currentButtonState = digitalRead(ENCODER_SW);
    unsigned long currentMillis = millis();
    if (lastButtonState == HIGH && currentButtonState == LOW) {
        lastButtonPress = currentMillis;
        buttonPressed = true;
        longPressHandled = false;
    }
    if (lastButtonState == LOW && currentButtonState == HIGH) {
        if (!longPressHandled && buttonPressed) {
            buzz(2000, 100);
            shortPressDetected = true;
        }
        buttonPressed = false;
        longPressHandled = false;
    }
    if (buttonPressed && !longPressHandled && (currentMillis -
        lastButtonPress > 1000)) {
        buzz(2000, 250);
        longPressDetected = true;
        longPressHandled = true;
    }
    lastButtonState = currentButtonState;
}

```

Рисунок 3.5 – Лістинг методу обробки натискання кнопки енкодера

У межах програмної логіки метод `buzz()` використовується для генерації звукового сигналу на п'єзоелектричному бузері. Цей метод згадується у функціях `handleEncoder()` та `handleButton()`, де виконує роль індикатора дій користувача, таких як обертання енкодера або натискання кнопки.

Метод `buzz()` приймає два параметри: частоту сигналу в герцах (`freq`) та тривалість звучання в мілісекундах (`duration_ms`). Його реалізацію наведено на рисунку 3.6.

```

void buzz(int freq, int duration_ms) {
    int delay_us = 1000000 / freq / 2;
    long cycles = (long)freq * duration_ms / 1000;
    Serial.println("Buzz");
    for (long i = 0; i < cycles; i++) {
        digitalWrite(BUZZER_PIN, LOW);
        delayMicroseconds(delay_us);
        digitalWrite(BUZZER_PIN, HIGH);
        delayMicroseconds(delay_us);
    }
}

```

Рисунок 3.6 – Лістинг методу генерації звукового сигналу

				КС КРБ 123.235.00.00 ПЗ		Арк.
						41
Арк.	№ докум.	Підпис	Дата			

Стан `MODE_MENU` обробляється окремим методом `handleMenuMode()`, який реалізує логіку меню першого рівня для взаємодії з користувачем. У цьому режимі на дисплеї користувач бачить два пункти меню: "Set Time" для встановлення поточного часу та "Config Reminder" для налаштування нагадувань. Меню дозволяє переходити між різними режимами за допомогою енкодера, при цьому один із пунктів завжди підсвічується символом `>`, що вказує на активний вибір. Реалізацію методу наведено на рисунку 3.7.

```
void handleMenuMode() {
    int selectedItem = menuIndex % 2;
    lcd.setCursor(0, 0);
    lcd.print(selectedItem == 0 ? ">Set Time" : " Set Time");
    lcd.setCursor(0, 1);
    lcd.print(selectedItem == 1 ? ">Config Reminder" : " Config
    Reminder");
    if (longPressDetected) {
        lcd.clear();
        longPressDetected = false;
    }
    if (shortPressDetected) {
        lcd.clear();
        shortPressDetected = false;
        if (selectedItem == 0) {
            DateTime now = rtc.now();
            setHour = now.hour();
            setMinute = now.minute();
            encoder.setCount(0);
            prevPosition = 0;
            currentMode = MODE_SET_TIME;
            lcd.clear();
        } else {
            currentMode = MODE_SET_INTERVAL;
        }
    }
}
```

Рисунок 3.7 – Лістинг методу реалізації стану `MENU_IDLE`

На початку функції обчислюється індекс активного пункту меню на основі значення `menuIndex`. Відповідно до обраного пункту, один з рядків дисплея позначається символом `>` як активний. Перемикання між пунктами меню здійснюється через енкодер.

У випадку виявлення тривалого натискання (`longPressDetected`) дисплей очищується для оновлення інтерфейсу. Якщо зафіксовано коротке натискання кнопки (`shortPressDetected`), залежно від обраного пункту меню, встановлюється режим `MODE_SET_TIME` або `MODE_SET_INTERVAL`.

У разі переходу до режиму встановлення часу зчитується поточний час з модуля RTC, значення годин та хвилин передаються у змінні `setHour` та `setMinute`. Значення енкодера скидається для забезпечення коректної взаємодії у новому режимі.

У цьому стані на дисплеї пристрою виводиться поточний час, а також час наступного запланованого нагадування. Така інформація дозволяє користувачеві візуально контролювати актуальний стан графіка введення ін'єкцій. Крім відображення даних, у цьому режимі постійно виконується перевірка часу з метою своєчасного спрацювання сигналу нагадування. Зазначені функції реалізовано в методах `displayCurrentAndReminderTime` та `checkReminder`.

Метод `displayCurrentAndReminderTime`, реалізацію якого подано на рисунку 3.8, отримує поточний час з модуля реального часу (RTC) та порівнює його зі збереженим значенням в енергонезалежній пам'яті EEPROM. Якщо виявляється розбіжність між отриманими та збереженими значеннями годин і хвилин, відповідні дані в EEPROM оновлюються. Після цього здійснюється виведення поточного часу на дисплей у форматі "година:хвилина", з урахуванням форматування для двоцифрового відображення кожної складової.

Метод `checkReminder`, представлений у додатку Б.1, виконує аналіз запланованих нагадувань з метою перевірки на збіг поточного часу з часом найближчого нагадування. У разі збігу генерується звуковий сигнал заданої тривалості, після чого програма переходить до обчислення наступного нагадування. Для цього виконується обхід усіх наявних записів у масиві нагадувань з розрахунком різниці в хвилинах між поточним часом і кожним елементом. З отриманих варіантів обирається той, для якого розрахована

різниця мінімальна. Індекс відповідного елемента присвоюється змінній `currentReminderIndex`, що визначає активне нагадування для наступної перевірки. Таким чином забезпечується циклічність роботи алгоритму та безперервне нагадування відповідно до встановленого розкладу.

```
void displayCurrentAndReminderTime() {
    DateTime now = rtc.now();

    byte hour = readFromEEPROM(0);
    byte minute = readFromEEPROM(1);

    if(hour != now.hour()){
        writeToEEPROM(0, now.hour());
    }
    if(minute != now.minute()){
        writeToEEPROM(1, now.minute());
    }

    lcd.setCursor(0, 0);
    lcd.print("Time now: ");
    lcd.setCursor(10, 0);
    if (now.hour() < 10) lcd.print("0");
    lcd.print(now.hour());
    lcd.print(":");
    if (now.minute() < 10) lcd.print("0");
    lcd.print(now.minute());
    lcd.print(" ");
}
```

Рисунок 3.8 – Лістинг методу `displayCurrentAndReminderTime`

Стан `MODE_SET_TIME` реалізує логіку встановлення поточного часу користувачем за допомогою енкадера. Обробка цього стану здійснюється методом `handleSetTimeMode()`, який забезпечує покрокове налаштування годин і хвилин. Код цього методу наведено у додатку Б.2.

Метод починається з отримання поточного часу з модуля RTC. Далі на першому рядку дисплея виводиться текстовий заголовок "Config Time", що інформує користувача про активний режим налаштування часу. На другому рядку дисплея послідовно виводяться значення годин та хвилин, а також символи ">", які вказують на поточний етап налаштування. Значення змінної

timeSettingStep визначає, який саме компонент — години або хвилини — доступний для редагування.

Зміна значень здійснюється за допомогою енкодера. Поточна позиція енкодера зчитується методом `encoder.getCount()`, а зміщення визначається як різниця між новим значенням та попереднім (`prevPosition`). У разі зафіксованої зміни виконується відповідне коригування значення години або хвилин залежно від активного кроку. Значення годин обмежено в діапазоні від 0 до 23, хвилин — від 0 до 59. При досягненні граничних значень здійснюється циклічне повернення в межах допустимого діапазону. Для оновлення відображення дисплей очищується методом `lcd.clear()`.

Натискання кнопки енкодера обробляється через відповідні прапори `shortPressDetected` та `longPressDetected`. У разі короткого натискання система переходить до наступного кроку налаштування (з години до хвилин або навпаки). Після завершення другого етапу нові значення встановлюються на модулі реального часу за допомогою методу `rtc.adjust()`. Також ці значення записуються в енергонезалежну пам'ять EEPROM через функцію `writeToEEPROM()` для збереження між перезавантаженнями пристрою. Стан перемикається на `MODE_IDLE`, після чого ініціалізується індекс поточного нагадування за допомогою функції `initCurrentReminderIndex()`.

У разі довгого натискання кнопки налаштування також зберігається та завершується без переходу до другого етапу налаштування. Це дозволяє користувачеві за необхідності завершити конфігурацію без зміни обох компонентів. Таким чином, метод `handleSetTimeMode()` забезпечує повний цикл налаштування системного часу пристрою в інтерактивному режимі через інтерфейс енкодера.

Стан `MODE_SET_INTERVAL` реалізує логіку роботи з масивом нагадувань, зокрема вибір між додаванням нового нагадування або редагуванням уже існуючого. Обробка цього стану здійснюється методом `handleSetIntervalMode()`, код якого наведено в додатку Б.3.

				КС КРБ 123.235.00.00 ПЗ	Арк.
					45
Арк.	№ докум.	Підпис	Дата		

Метод починається з обчислення загальної кількості пунктів, що відображаються в інтерфейсі: це кількість наявних нагадувань плюс один додатковий пункт для створення нового. Значення індексу `currentIndex`, яке задається за допомогою енкодера, обмежується в допустимому діапазоні від 0 до `reminderCount`. Якщо поточне значення виходить за межі, воно коригується та синхронізується з енкодером викликом `encoder.setCount()`.

На першому рядку дисплея виводиться пункт меню "Add New", що відповідає опції створення нового нагадування. Цей пункт підсвічується символом ">", якщо він є активним відповідно до значення `currentIndex`.

Другий рядок дисплея відображає наявне нагадування, яке відповідає індексу `currentIndex - 1`. Якщо вибраний пункт не є першим, відображається відповідне значення з масиву структур `reminders`. Час нагадування формується у форматі ГГ:ХХ із додаванням провідних нулів. Якщо в системі немає збережених нагадувань (`reminderCount == 0`), виводиться повідомлення "No reminders". Для вирівнювання виводу застосовуються пробіли.

Обробка натискань кнопки відбувається за допомогою прапорів `shortPressDetected` та `longPressDetected`. У разі короткого натискання, якщо активним є перший пункт списку (`currentIndex == 0`), виконується перехід до стану `MODE_ADD_REMINDER`. В іншому випадку зберігається індекс вибраного нагадування в змінну `menuIndex`, і здійснюється перехід до стану `MODE_EDIT_REMINDER`. Це дозволяє реалізувати функціонал редагування часу окремого нагадування, обраного через меню.

Довге натискання слугує для виходу з режиму налаштування інтервалів. У цьому випадку пристрій повертається до стану `MODE_MENU`, дисплей очищується для уникнення залишкової інформації з попереднього режиму.

Таким чином, метод `handleSetIntervalMode()` реалізує функціональність вибору між додаванням нового нагадування або редагуванням наявного через інтерактивний інтерфейс, що ґрунтується на позиції енкодера.

Стан `MODE_ADD_REMINDER` призначено для створення нового нагадування користувачем. Його обробка реалізована в методі `handleAddReminderMode()`, повний код якого наведено в додатку Б.4.

На першому етапі роботи методу відбувається вивід службового повідомлення "Add Reminder" у першому рядку дисплея. Другий рядок містить значення встановленого часу у форматі годин:хвилин з можливістю виділення активного поля символом ">". Поле вказується залежно від поточної фази налаштування, що визначається змінною `timeSettingStep`, яка приймає значення 0 або 1 (відповідно редагування годин або хвилин).

Подальша обробка базується на зчитуванні нового положення енкодера. Якщо значення лічильника енкодера (`encoder.getCount()`) змінилося порівняно з попереднім (`prevPosition`), обчислюється дельта та відповідним чином модифікується змінна `setHour` або `setMinute`. У разі виходу за допустимі межі значення коригуються (циклічно в межах 0–23 для годин і 0–59 для хвилин). При зміні значення дисплей очищується для оновлення відображення.

При короткому натисканні кнопки (`shortPressDetected`), якщо редагується поле годин (`timeSettingStep == 0`), перемикається на редагування хвилин. Якщо обидва значення встановлено, виконується збереження нагадування.

Перед збереженням перевіряється, чи не перевищено ліміт збережених нагадувань (`MAX_REMINDERS`). Якщо ліміт не перевищено, нове нагадування додається до масиву `reminders`, після чого кількість нагадувань (`reminderCount`) збільшується на одиницю. Далі викликається функція `sortReminders()`, яка сортує масив нагадувань у порядку зростання часу.

Після сортування всі актуальні нагадування записуються в енергонезалежну пам'ять EEPROM. Для цього кожне нагадування зберігається починаючи з адреси 3, де година і хвилина розташовуються послідовно (година — за непарною адресою, хвилина — за парною). Крім цього, кількість нагадувань зберігається в EEPROM за адресою 2.

Операція підтверджується також через монітор порту, де виводиться інформація про кожне нове нагадування і нове значення лічильника reminderCount. Після завершення введення змінна timeSettingStep скидається до нуля, режим пристрою переводиться до MODE_IDLE, а дисплей очищується.

У разі довгого натискання (longPressDetected) виконуються ті ж самі дії, що і при короткому натисканні в останній фазі: нагадування зберігається, якщо дозволяє ліміт, інформація записується в EEPROM, режим скидається, дисплей очищується.

Стан MODE_EDIT_REMINDER призначено для редагування або видалення вже наявного нагадування. Реалізацію цього стану забезпечує метод handleEditReminderMode(), повний код якого наведено в додатку Б.5.

На початку виконання методу перевіряється прапорець editInitDone, який визначає, чи було ініціалізовано дані для редагування. Якщо це перший виклик методу після переходу в стан, зчитуються поточні значення години та хвилини з масиву reminders за індексом menuIndex, що відповідає обраному користувачем нагадуванню. Також скидається крок редагування timeSettingStep, прапорець ініціалізації встановлюється у значення true, а дисплей очищується.

На дисплеї у першому рядку виводиться службовий напис "Edit Reminder". У другому рядку відображаються значення години та хвилини з виділенням активного поля символом ">". Активне поле визначається поточним кроком (timeSettingStep), аналогічно до методу додавання нагадування.

Далі обробляється нове положення енкодера. Якщо користувач змінив позицію, обчислюється різниця між новим і попереднім значенням. У залежності від активного поля, змінюється або значення години (setHour), або хвилини (setMinute) з циклічним обмеженням у діапазонах 0–23 і 0–59 відповідно. Після внесення змін дисплей очищується для оновлення.

При короткому натисканні кнопки (shortPressDetected), якщо ще редагуються години, відбувається перехід до редагування хвилин. Якщо обидва значення вже встановлено, нові значення записуються в масив reminders за відповідним індексом. Далі виконується сортування нагадувань за часом за допомогою функції sortReminders().

Після сортування нагадування перезаписуються в EEPROM. Для кожного нагадування години та хвилини зберігаються послідовно, починаючи з адреси 3. Після завершення запису прапорець editInitDone скидається, фаза редагування (timeSettingStep) скидається, режим пристрою змінюється на MODE_IDLE, а дисплей очищується.

У разі довгого натискання кнопки (longPressDetected) поточне нагадування видаляється. Це реалізується шляхом зсуву елементів масиву reminders вліво, починаючи з індексу menuIndex. Кількість нагадувань (reminderCount) зменшується на одиницю. Після цього масив нагадувань повторно записується в EEPROM, включно з оновленим значенням reminderCount. Після завершення операції змінні скидаються, режим переходить до MODE_IDLE, дисплей очищується.

У всіх описаних вище режимах роботи пристрою реалізовано взаємодію з енергонезалежною пам'яттю (EEPROM), зокрема — зчитування з неї та запис даних. Це дозволяє зберігати кількість і параметри нагадувань у разі вимкнення живлення. Операції запису й зчитування реалізовано у методах writeToEEPROM та readFromEEPROM.

Метод writeToEEPROM, наведений на рис. 3.9, реалізує запис одного байта даних за вказаною адресою EEPROM, яка вбудована у мікросхему годинника реального часу DS1307.

```
void writeToEEPROM(uint16_t addr, byte data) {
    Wire.beginTransmission(DS1307_EEPROM_ADDRESS);
    Wire.write((addr >> 8) & 0xFF);
    Wire.write(addr & 0xFF);
    Wire.write(data);
    Wire.endTransmission();
    delay(5);
}
```


}

Рисунок 3.9 – Лістинг методу writeToEEPROM

Цей метод приймає два аргументи: 16-бітну адресу `addr` і байт `data`, який необхідно записати. У тілі функції відбувається ініціалізація передавання через інтерфейс I²C за допомогою `Wire.beginTransmission()` на адресу пристрою `DS1307_EEPROM_ADDRESS`. Далі адреса розбивається на два байти — старший і молодший — і передається в шину I²C, після чого передається сам байт даних. Після завершення передавання викликається `Wire.endTransmission()`. Затримка `delay(5)` забезпечує завершення циклу запису до EEPROM, оскільки запис у цю пам'ять займає певний час.

Метод `readFromEEPROM`, наведений на рис. 3.10, реалізує зчитування одного байта даних із зазначеної адреси EEPROM.

```
byte readFromEEPROM(uint16_t addr) {
    Wire.beginTransmission(DS1307_EEPROM_ADDRESS);
    Wire.write((addr >> 8) & 0xFF);
    Wire.write(addr & 0xFF);
    Wire.endTransmission();

    Wire.requestFrom(DS1307_EEPROM_ADDRESS, 1);
    if (Wire.available()) {
        return Wire.read();
    }
    return 0xFF;
}
```

Рисунок 3.10 – Лістинг методу readFromEEPROM

Аргументом цієї функції є 16-бітна адреса `addr`. Спочатку за допомогою `Wire.beginTransmission()` і `Wire.write()` передається адреса пам'яті в два байти (старший і молодший). Після завершення передавання викликається `Wire.endTransmission()`. Далі запитується один байт даних від пристрою через `Wire.requestFrom()`. Якщо байт доступний для зчитування, він зчитується та повертається. У разі відсутності даних метод повертає значення `0xFF`.

Таким чином, зазначені функції забезпечують базову низькорівневу взаємодію з EEPROM, що дозволяє зберігати та відновлювати нагадування та інші конфігураційні параметри системи.

3.3 Тестування системи контролю введення ін'єкцій

Для тестування розробленої системи контролю введення ін'єкцій здійснюється її підключення до джерела живлення. З метою забезпечення зручності спостереження за процесом роботи під час тестування застосовується живлення від порту USB, що дозволяє одночасно використовувати Arduino IDE для відлагодження і моніторингу виводу через серійний порт.

Після подачі живлення на екран виводиться початкове вікно — так званий головний екран системи. На ньому відображається поточний час, однак без попереднього налаштування RTC цей час може бути некоректним. Для переходу до налаштувань користувач здійснює довге натискання на кнопку енодера. У результаті відкривається перший рівень меню, який показано на рис. 3.11. У цьому меню відображаються пункти вибору: встановлення часу, налаштування інтервалу нагадувань, додавання нового нагадування та редагування наявного.

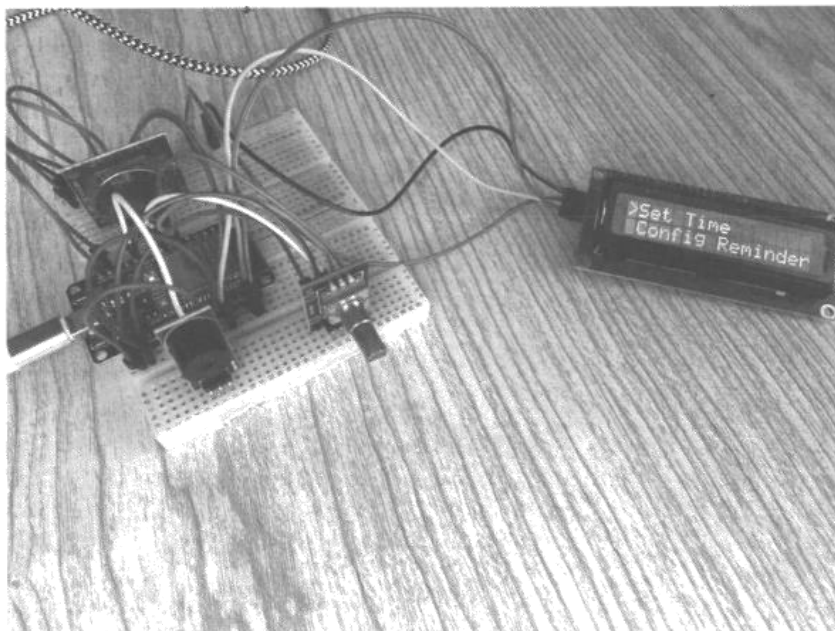


Рисунок 3.11 – Результат відображення першого рівня меню

Далі для перевірки встановлення поточного часу виконується коротке натискання на кнопку, внаслідок чого активується режим налаштування часу. Відповідний інтерфейс показано на рис. 3.12. На екрані відображається напис "Set Time", а нижнім рядком виводиться значення годин і хвилин, що редагуються. Поточний параметр редагування виділяється спеціальним символом. Обертання енкодера дозволяє змінити значення годин та хвилин, перемикавання між ними здійснюється коротким натисканням.

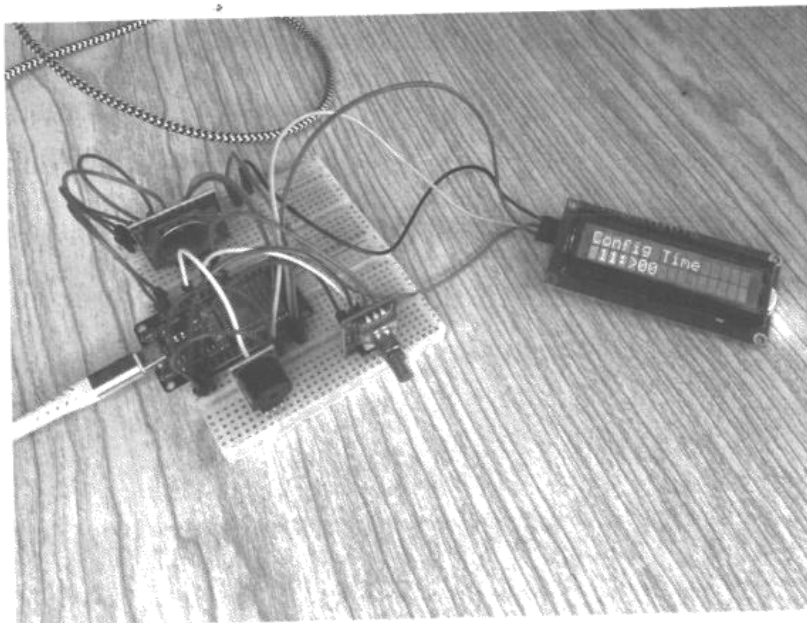


Рисунок 3.12 – Вигляд меню налаштування поточного часу

Під час тестування було встановлено час 11:00. Після завершення введення часу та виходу з режиму налаштування, користувач повертається на головний екран, де тепер відображається коректний встановлений час. Вигляд головного екрана після встановлення показано на рис. 3.13.

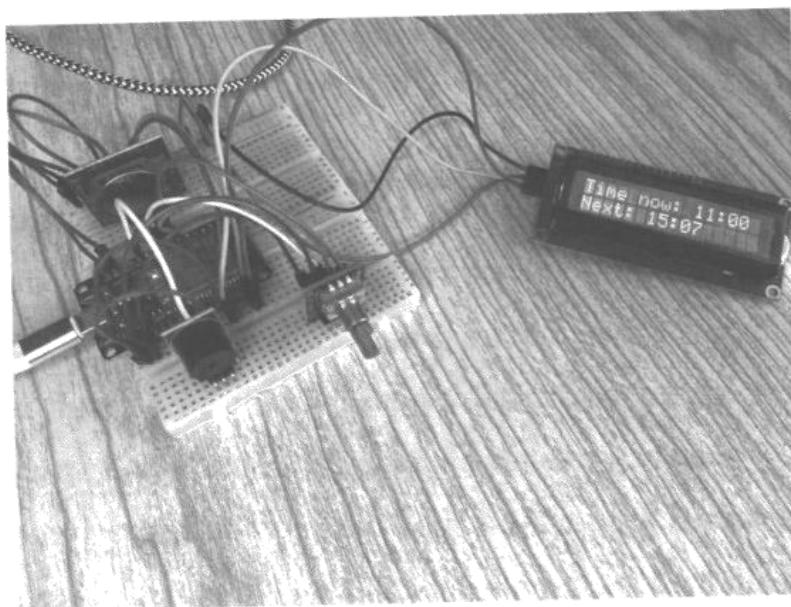


Рисунок 3.13 – Вигляд головного екрану після налаштування поточного часу

Далі перевіряється функціональність додавання нового нагадування. Для цього здійснюється перехід у перший рівень меню за допомогою довгого натискання на кнопку енкодера. Після цього шляхом обертання енкодера обирається пункт меню "Config Reminder", що відповідає за роботу з нагадуваннями. Для підтвердження вибору виконується коротке натискання на кнопку енкодера. У результаті відкривається підменю конфігурації нагадувань, яке зображено на рис. 3.14.

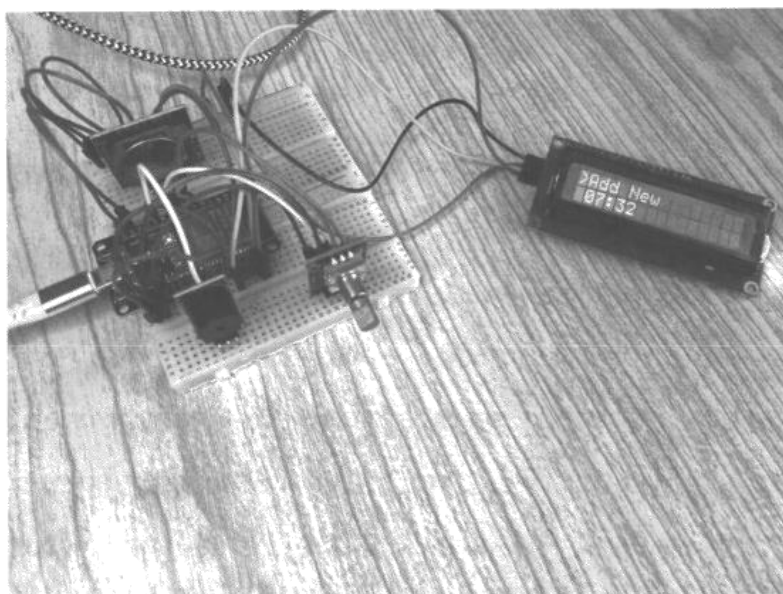


Рисунок 3.14 – Вигляд меню роботи з нагадуваннями

У підменю відображається доступний пункт "Add New", який дозволяє додати нове нагадування. Обравши його за допомогою енкодера і натиснувши кнопку, користувач переходить у режим встановлення параметрів нового нагадування. Інтерфейс цього режиму показано на рис. 3.15. На дисплеї виводиться заголовок "Add Reminder", а також значення годин і хвилин, які редагуються по черзі. Перемикання між годинами та хвилинами здійснюється коротким натисканням, зміна значення — шляхом обертання енкодера.

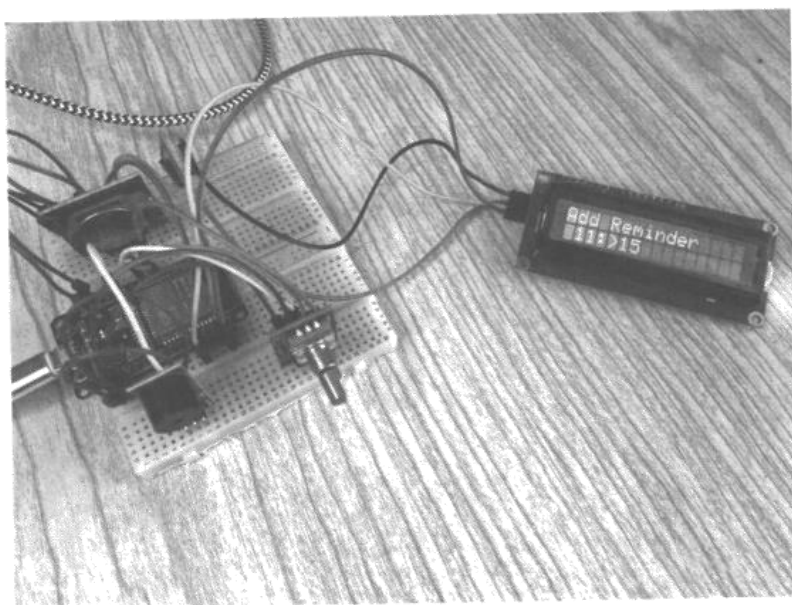


Рисунок 3.15 – Вигляд пункту додавання нового нагадування

У межах тестування встановлюється час нагадування на 11:15. Після підтвердження вибору нагадування зберігається до пам'яті пристрою, і користувач автоматично повертається на головний екран. На оновленому головному екрані, що наведено на рис. 3.16, додатково до поточного часу відображається встановлене нагадування. Це свідчить про коректну роботу функції додавання нового нагадування.

Аналогічним чином було додано ще одне нагадування на 16:00. Після встановлення параметрів і підтвердження вибору нове нагадування збереглося до енергонезалежної пам'яті пристрою.

Після настання часу першого нагадування — 11:15 — було зафіксовано спрацювання звукового сигналу за допомогою бузера. Одночасно на дисплеї відобразився наступний актуальний час нагадування, тобто 16:00. Це підтверджує, що логіка переходу між нагадуваннями реалізована коректно, а також, що система відтворення сигналу працює відповідно до заданого алгоритму. Оновлений вигляд дисплея у цьому стані наведено на рис. 3.17.

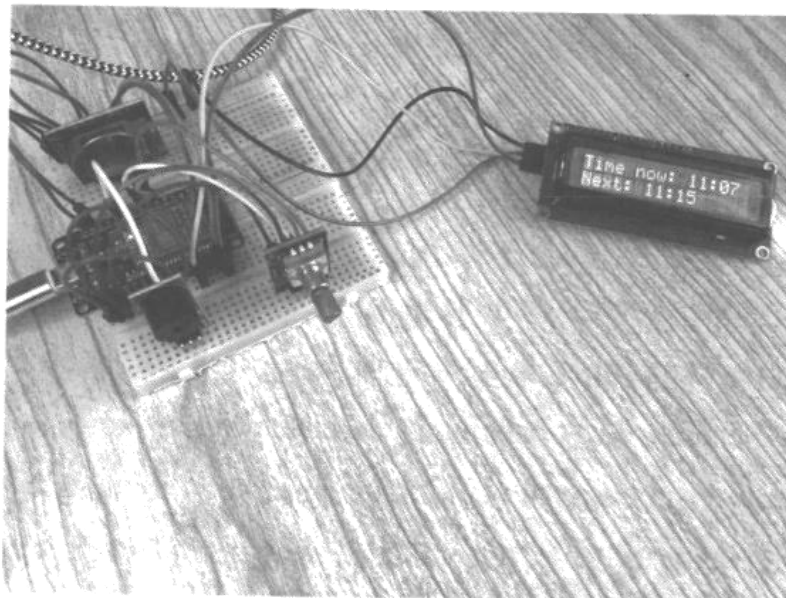


Рисунок 3.16 – Вигляд головного екрану після налаштування нагадування

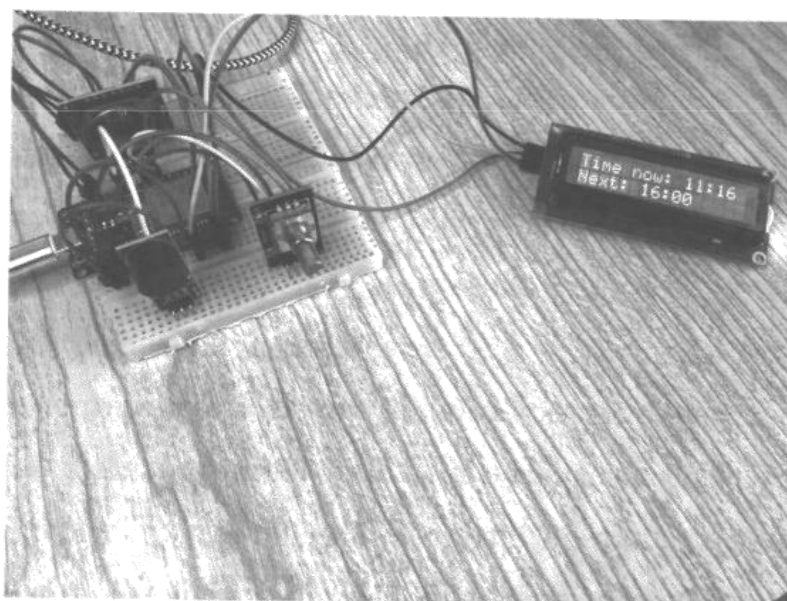


Рисунок 3.17 – Вигляд головного екрану після спрацювання першого нагадування

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Фізіологічний вплив факторів існування на життєдіяльність людини

Фізіологічний вплив факторів існування на життєдіяльність людини є комплексним і тісно пов'язаним з умовами використання комп'ютеризованої системи контролю введення ін'єкцій у пацієнтів з цукровим діабетом. Така система виконує не лише технічну функцію контролю дозування інсуліну, а й безпосередньо впливає на функціональні системи організму через фізичне, сенсорне та м'язове навантаження. Її ефективність визначається не тільки точністю роботи електронних компонентів, а й комфортом, з яким пацієнт щодня взаємодіє з пристроєм. Важливу роль відіграє адаптація пристрою до індивідуальних особливостей користувача, таких як вік, стан здоров'я та когнітивні можливості. Система є інтуїтивно зрозумілою для користувача незалежно від його рівня технічної підготовки. У випадку ускладненого інтерфейсу підвищується ризик помилкових дій, які можуть негативно позначитися на фізіологічному стані.

Система моніторингу й автоматичного контролю часу та дози інсуліну критично важлива для підтримки стабільного рівня глюкози в крові. Її тривале використання передбачає постійний фізичний контакт, зорове та тактильне навантаження, що може призводити до фізіологічного виснаження, особливо у пацієнтів літнього віку або з порушеннями моторної функції. У деяких випадках хронічне напруження м'язів кисті через часту взаємодію з енкодером може призводити до розвитку м'язово-скелетних розладів. Обмеження рухливості пальців може також ускладнити точне налаштування параметрів введення інсуліну. Додаткові ризики з'являються при використанні системи

					КС КРБ 123.XXX.00.00 ПЗ		
Арк.	№ докум.	Підпис	Дата				
Розроб.	Соснін	[Підпис]	06.08	Безпека життєдіяльності, основи охорони праці	Літ.	Арк.	Архувів
Перевірив	Монора	[Підпис]	06.08			57	
Консульт.	Димитрюк	[Підпис]	25.08				
Контр.	Мисюк	[Підпис]	06.08				
Затверд.	Осухівська Г.М.	[Підпис]	25.08		ТНТУ, каф. КС, гр. СІ-41		

однією рукою або при слабкій координації рухів. Тремор, характерний для літніх людей, може ускладнити точне позиціонування енкодера. У таких випадках рекомендованим є введення голосових команд або сенсорного управління як альтернативного інтерфейсу.

Незручне або неправильне розташування дисплея (1602 з модулем HW-061) відносно лінії зору користувача спричиняє додаткове м'язове напруження в шиї та плечах. Світлове навантаження від екрана також має значення: постійна яскравість або мерехтіння дисплея, особливо в умовах низької освітленості, здатне викликати зорове перенапруження, сухість очей або головний біль. Використання антирефлексійного покриття може зменшити навантаження на зір під час тривалого використання. Особливо це стосується людей із наявними офтальмологічними проблемами, які є поширеними серед пацієнтів із діабетом. Пристрій налаштування контрасту, щоб покращити видимість символів у різних умовах освітлення. Адаптивне підсвічування або автоматичний режим зміни яскравості сприяють зниженню втоми очей. Надмірне навантаження на зоровий апарат може також вплинути на якість сну пацієнта, особливо якщо пристрій використовується у вечірній час.

Регулярне використання кнопочового інтерфейсу може також викликати механічне подразнення шкіри пальців або сприяти розвитку мікротравм при надмірному натисканні. Точність дозування інсуліну — ще один надзвичайно важливий фізіологічний аспект. Введення неправильної дози через помилкове налаштування або некоректне зчитування інформації може спричинити гіпоглікемію (критичне зниження рівня глюкози) або гіперглікемію (надмірне її підвищення), що безпосередньо впливає на функціонування мозку, серцево-судинної системи та загальний стан організму. Уникнення цих порушень на пряму залежить від точності технічної реалізації пристрою та від ступеня розуміння пацієнтом процесу його експлуатації. Необхідним є періодичне оновлення програмного забезпечення для покращення алгоритмів діагностики помилок. Наявність функції підтвердження дій перед введенням інсуліну також допомагає уникати небажаних наслідків.

					КС КРБ 123.235.00.00 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

Компонент Tiny RTC DS1307 забезпечує високоточне вимірювання часу, що дає змогу дотримуватися оптимального графіку введення інсуліну. Регулярність ін'єкцій є критичною умовою стабільного метаболізму, тому навіть незначні збої у хронології можуть негативно впливати на фізіологічні процеси. Сенсор MH-FMD використовується для наближеного визначення моменту потреби у введенні інсуліну — його сигнали допомагають пацієнту не пропустити важливий момент, що, у свою чергу, запобігає метаболічному дисбалансу. Модуль ESP-32 не тільки виконує функцію контролера, а й слугує передавачем даних у хмарну систему, де зібрана інформація аналізується лікарем. Завдяки цьому забезпечується зворотній зв'язок і можливість дистанційного коригування терапії, що позитивно впливає на загальну фізіологічну стабільність пацієнта. Крім того, передача даних в реальному часі дозволяє лікарю своєчасно реагувати на критичні зміни у стані здоров'я пацієнта. Пацієнт, своєю чергою, отримує психологічне підкріплення, знаючи, що лікар бачить його показники.

Тривале носіння пристрою або його близьке розташування до тіла може створювати додатковий електромагнітний вплив. Відповідно до вимог ДСТУ EN 60601-1:2015, цей вплив залишиться в межах безпечних норм, однак систематичне випромінювання навіть низької інтенсивності може мати кумулятивний ефект у чутливих осіб. З цієї причини необхідно здійснювати регулярні перевірки відповідності пристрою нормам електромагнітної безпеки, а також надавати рекомендації щодо оптимального розміщення системи при її використанні. Виробник враховує використання якісного екранування та сертифікованих радіоелементів. Забезпечення електромагнітної сумісності з іншими медичними пристроями, які можуть одночасно функціонувати поруч, також має велике значення.

Важливо враховувати й сенсорне навантаження: часте натискання, обертання або перегляд даних на дисплеї можуть зумовити мікронавантаження на пальці та очі, що призводить до локального перевтомлення. У зв'язку з цим доцільним є впровадження режимів

					КС КРБ 123.235.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

автоматичного зниження активності пристрою після завершення введення даних (блокування дисплея, деактивація підсвітки або автоматичне вимкнення неактивних елементів керування). Такий підхід знижує ризик перевантаження та забезпечує фізіологічний комфорт при тривалому використанні. Додаткові функції, як-от нагадування про перерву або вправи для розслаблення кисті, можуть мати позитивний ефект. У перспективі передбачено персоналізацію пристрою під конкретного пацієнта на основі біометричних параметрів.

Фізіологічний вплив комп'ютеризованої системи на життєдіяльність людини охоплює широкий спектр чинників — від ергономіки взаємодії та якості візуальних елементів до електромагнітної безпеки й точності дозування. Комплексний підхід до оцінки цих факторів дозволяє мінімізувати можливі негативні наслідки та підвищити рівень фізіологічної безпеки при використанні пристрою в повсякденному житті. Високий рівень індивідуалізації й адаптивності таких систем дозволяє ефективніше інтегрувати їх у повсякденний ритм життя пацієнта. Усе це сприяє не тільки фізичному, але й психоемоційному добробуту користувача, що вкрай важливо при довготривалому хронічному захворюванні.

4.2 Психоемоційний вплив факторів існування на життєдіяльність людини

Психоемоційний вплив факторів існування на життєдіяльність людини, особливо в умовах використання комп'ютеризованих систем медичного призначення, є надзвичайно важливим і часто недооціненим аспектом. Людина, яка постійно перебуває в стані моніторингу та самоконтролю, стикається з психологічними викликами, що можуть викликати стрес, тривожність або емоційне виснаження. Особливо це стосується пацієнтів із хронічними захворюваннями, зокрема цукровим діабетом, де систематичне введення інсуліну є життєво необхідною процедурою. Щоденна необхідність взаємодії з технічним пристроєм може створити додаткове когнітивне

					КС КРБ 123.235.00.00 ПЗ	Арк.
Арк.	№ докум.	Підпис	Дата			60

навантаження та підвищити ризик психологічного виснаження. Деякі пацієнти можуть відчувати страх перед технологією або недовіру до автоматизованої системи, особливо в разі попереднього негативного досвіду.

Постійна прив'язаність до пристрою, який контролює життєво важливі функції, може викликати відчуття вразливості або залежності, що погіршує емоційний стан людини. Іноді пацієнти відчують провину або самозвинувачення в разі помилок у використанні системи, навіть якщо вони не пов'язані з їхніми діями. Це може знижувати впевненість у собі та спричинити уникнення взаємодії з пристроєм. Виникає своєрідний психологічний бар'єр, подолати який без зовнішньої підтримки складно. У той же час для іншої категорії пацієнтів така система може виступати фактором підтримки впевненості, знижуючи рівень тривоги, пов'язаної з ризиком помилки під час введення інсуліну вручну.

Наявність чіткої структури та передбачуваних дій, закладених у програмне забезпечення, створює відчуття контролю над ситуацією. Важливо також, щоб користувач мав можливість отримати швидку технічну або медичну підтримку, що знижує рівень емоційної напруги. Часто користувачі відчують ізольованість у своєму досвіді хвороби, тому функції обміну даними або зв'язку з лікарем чи спільнотою можуть позитивно впливати на емоційний стан. У процесі взаємодії з системою важливою є також якість інтерфейсу: занадто складне або неінтуїтивне управління здатне викликати роздратування та знижувати мотивацію до її використання.

Особливо це актуально для пацієнтів старшого віку, які можуть мати обмежений досвід користування цифровими пристроями. Простий, доступний інтерфейс та наявність візуальних підказок або інструкцій суттєво зменшують психоемоційне напруження. Для людей з когнітивними порушеннями або проблемами зору варто передбачити спеціальні режими відображення, наприклад, великі шрифти або голосовий супровід. Корисною є також наявність функцій персоналізації: відображення імені користувача,

нагадування у формі доброзичливих повідомлень або візуальні елементи, що викликають позитивні асоціації.

Звукові сигнали або вібрація, які повідомляють про завершення введення даних чи помилки, також впливають на психоемоційний стан. Їх тональність, гучність і частота мають бути оптимально підібрані, щоб не викликати роздратування або стрес. Надмірно гучні або різкі сигнали можуть спричинити подразнення або навіть панічні реакції. Пацієнт повинен мати змогу самостійно регулювати інтенсивність звуків відповідно до власних уподобань. Позитивне підкріплення у вигляді спокійного звуку або короткого повідомлення про успішну дію може підвищити рівень задоволення від використання системи.

Важливим аспектом є й естетичний вигляд пристрою. Якщо зовнішній вигляд системи викликає асоціації з лікарняним обладнанням, це може негативно впливати на самооцінку користувача та підсилювати відчуття хворобливості. Натомість сучасний, компактний і нейтральний дизайн сприяє тому, що пацієнт сприймає пристрій як частину повсякденного життя. Дизайн має бути мінімалістичним, але привабливим, з використанням м'яких кольорів та заокруглених форм. Такий підхід покращує візуальне сприйняття пристрою та зменшує психологічну дистанцію до нього.

Психоемоційний вплив також пов'язаний із соціальними факторами. Використання медичного пристрою в публічних місцях може викликати почуття незручності або навіть сорому. Пацієнт може уникати використання пристрою у присутності інших, що може призводити до порушень режиму введення інсуліну. Для запобігання цьому важливо забезпечити користувачу почуття конфіденційності та автономії. Надання змоги непомітного використання пристрою або керування ним через мобільний застосунок зменшують соціальний тиск і тривожність.

Додаткову психологічну підтримку забезпечувати інтеграцію мотиваційних функцій у програмне забезпечення, таких як відстеження досягнень, нагороди за стабільні показники або позитивні повідомлення. Це

КС КРБ 123.235.00.00 ПЗ					Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	62

не тільки сприяє покращенню настрою, а й стимулює до регулярного використання системи. Забезпечення доступу до аналітики власного стану здоров'я дозволяє пацієнту краще розуміти свій організм і знижує рівень тривожності, пов'язаної з невизначеністю. Якщо система демонструє покращення показників у вигляді графіків або звітів, це може стати джерелом натхнення та психологічного полегшення.

Позитивним є також надання можливості встановлювати власні цілі та отримувати візуальні підтвердження їх досягнення. Це створює ефект прогресу та зменшує страх перед хворобою. Застосування ігрових елементів, таких як бали або рівні, також може підтримувати емоційну мотивацію. У деяких випадках доречним є включення функції нагадування з позитивними фразами або цитатами, які підтримують користувача морально. Не менш важливою є адаптація пристрою до психологічного стану користувача.

У разі виявлення ознак емоційного виснаження або зниження активності система може подавати м'які нагадування або пропонувати скористатися режимом зменшеного навантаження. Застосування кольоротерапії в інтерфейсі, наприклад, використання заспокійливих відтінків синього або зеленого, також сприяє нормалізації психоемоційного стану. Пристрій працювати не як суворий контрольний інструмент, а як особистий асистент, що розуміє та підтримує користувача. Відчуття турботи та людяності з боку технології може стати фактором емоційного комфорту.

Психоемоційний вплив факторів існування в умовах використання комп'ютеризованої системи контролю ін'єкцій охоплює широкий спектр — від особистих переживань та емоційних реакцій на дизайн і функціонал до соціальної взаємодії та відчуття автономії. Усвідомлення цих аспектів і врахування їх на етапі проектування системи дає змогу створити дійсно ефективний, безпечний та емоційно прийнятний інструмент для підтримки здоров'я.

					КС КРБ 123.235.00.00 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах кваліфікаційної роботи була реалізована повноцінна комп'ютеризована система контролю введення ін'єкцій, орієнтована на пацієнтів з цукровим діабетом для використання в побутових умовах. Було виконано повний цикл розробки — від аналізу технічного завдання до проведення тестування макетного зразка.

Результатом виконання завдань стало створення пристрою на основі мікроконтролера ESP32, який забезпечує функціональність введення, збереження, редагування та видалення нагадувань про ін'єкції. У складі пристрою реалізовано інтерфейс для взаємодії з користувачем за допомогою енкодера та LCD-дисплея, а також підтримка точного відліку часу завдяки використанню годинника реального часу DS1307. Для зберігання даних використовуються енергонезалежна пам'ять EEPROM, що дозволяє зберігати налаштування навіть при вимкненні живлення. Для сповіщення про події передбачено використання звукового сигналу.

У процесі тестування було підтверджено коректну роботу всіх програмних режимів, зокрема:

- встановлення поточного часу;
- додавання нового нагадування;
- редагування та видалення існуючого нагадування;
- збереження даних у пам'ять та їх зчитування;
- автоматичне оновлення дисплея при настанні часу ін'єкції;
- спрацювання звукового сигналу відповідно до заданого розкладу.

Під час тестування на макетній платі виявлено стабільну роботу всіх апаратних і програмних компонентів. Усі встановлені нагадування коректно зберігалися в EEPROM та відображалися на екрані при перезавантаженні пристрою. Система забезпечує високу точність функціонування завдяки використанню зовнішнього RTC, що гарантує точність відліку часу, а також

дозволяє змінювати налаштування без необхідності у зовнішньому програмуванні.

Розроблене рішення відповідає поставленій меті і може бути використане як основа для створення кінцевого пристрою з автономним живленням і в відповідному корпусі. Подальша розробка включатиме адаптацію пристрою для використання в реальних умовах, а також можливість інтеграції в медичні системи для покращення контролю над процесом введення інсуліну у пацієнтів з цукровим діабетом.

					КС КРБ 123.235.00.00 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чабанна О. С. Лікування пацієнтів із цукровим діабетом препаратами інсуліну гларгіну. 2023. URL: <http://dspace.zsmu.edu.ua/bitstream/123456789/19423/1/%d1%8175-76.pdf> (дата звернення: 09.04.2025).
2. Пріменко Д. Ю. Інформаційна система аналізу та корегування рівня цукру при цукровому діабету. 2023. URL: <https://ela.kpi.ua/server/api/core/bitstreams/269dce88-b0af-4f50-9b83-5b6c47d5728d/content> (дата звернення: 04.04.2025).
3. MedAngel ONE. Zorginnovatie. URL: <https://zorginnovatie.nl/innovaties/medangel-one> (дата звернення: 14.04.2025).
4. Ковпачок timesulin. Mhealth. URL: <https://mhealth.com.ua/timesulin-cap.html> (дата звернення: 10.04.2025).
5. Diabetes under control. Insulcloud. URL: <https://www.insulcloud.com/en/> (дата звернення: 11.04.2025).
6. Бондаренко І. М., Бородин О. В., Карнаушенко В. П. Сучасна компонентна база електронних систем : навч. посібник для студентів ЗВО. Харків: ХНУРЕ, 2020. 268 с. URL: <https://openarchive.nure.ua/bitstreams/9e3e144c-6c38-4484-b587-5dedca6a42ba/download> (дата звернення: 12.04.2025).
7. Коржик М. В. Мікропроцесорна техніка. Комп'ютерний практикум. 2021. URL: <https://ela.kpi.ua/server/api/core/bitstreams/fc0bf386-fd49-495d-a7fd-923ec1b3b134/content> (дата звернення: 08.04.2025).
8. ESP32-Wroom-32 Datasheet. Espressif Systems. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (дата звернення: 07.04.2025).
9. ESP-IDF programming guide. Espressif Systems. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html> (дата звернення: 06.04.2025).

					КС КРБ 123.235.00.00 ПЗ	Арк.
						66
Арк.	№ докум.	Підпис	Дата			

10. Rtc-ds1307-module-smd-schematic. Electronics-Lab URL: <https://www.electronics-lab.com/project/ds1307-rtc-module/rtc-ds1307-module-smd-schematic/> (дата звернення: 15.04.2025).
11. 24C32. Microchip. URL: <https://ww1.microchip.com/downloads/en/devicedoc/21061h.pdf> (дата звернення: 13.04.2025).
12. TinyRTC I2C Module. Gizmo. URL: <https://pdf.direnc.net/upload/tinyrtc-i2c-modul-datasheet.pdf> (дата звернення: 05.04.2025).
13. I2C LCD with ESP32 on Arduino IDE. Random Nerd Tutorials. URL: <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/> (дата звернення: 10.04.2025).
14. Rotary encoder for arduino/raspberry. Handson Technology. URL: <https://www.handsontec.com/dataspecs/module/Rotary%20Encoder.pdf> (дата звернення: 14.04.2025).
15. Hight quality passive buzzer module. e-Gizmo. URL: <https://e-gizmo.net/oc/kits%20documents/High%20Quality%20Passive%20Buzzer/High%20quality%20passive%20buzzer.pdf> (дата звернення: 13.04.2025).
16. Arduino ESP32 latest documentation. Espressif Systems. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html> (дата звернення: 06.04.2025).
17. Harrington K. ESP32Encoder. GitHub. URL: <https://github.com/madhhephaestus/ESP32Encoder/tree/master> (дата звернення: 15.04.2025).
18. RTCLib. Docs Arduino. URL: <https://docs.arduino.cc/libraries/rtclib/> (дата звернення: 09.04.2025).
19. E-Gizmo Mechatronix Central. RTC DS1307 AT24C32 real time clock module. GitHub. URL: <https://github.com/e-Gizmo/RTC-DS1307-AT24C32-Real>

				КС КРБ 123.235.00.00 ПЗ		Арк.
Арх.	№ докум.	Підпис	Дата			67

Time-Clock-Module/blob/master/TinyRTC%20i2c%20module%20%20Techincal%20Manual%20rev1.pdf (дата звернення: 11.04.2025).

20. Мешко Р. О. Базові принципи проектування та розробки технічної документації сучасними засобами САПР. 2023. URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/69140/1/09111403.pdf> (дата звернення: 07.04.2025).

21. Piezo generators. Americanpiezo. URL: <https://www.americanpiezo.com/knowledge-center/piezo-theory/generators/> (дата звернення: 04.04.2025)

					КС КРБ 123.235.00.00 ПЗ	Арк.
						68
№	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.

“02” 02 2025 р

Комп'ютеризована система контролю введення ін'єкцій у пацієнтів з цукровим
діабетом

ТЕХНІЧНЕ ЗАВДАННЯ

на 9 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

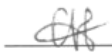
Керівник кваліфікаційної роботи

 к.т.н., доц. Шингера Н.Я.

«24» січня 2025 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-42

 Скочипець Н.О.

«24» січня 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система контролю введення ін'єкцій у пацієнтів з цукровим діабетом».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.235.00.00

1.2 Виконавець

Студент групи СІ-42, факультету комп'ютерно-інформаційних систем та програмної інженерії, кафедри комп'ютерних систем та мереж Тернопільського національного технічного університету імені Івана Пулюя Скочипець Н.О.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ 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, 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

Медіасервер що розробляється призначений для:

- Підтримки роботи мобільних пристроїв.
- Зберігання мультимедійних даних.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка та налаштування мультимедійного сервера на основі одноплатного мікрокомп'ютера Raspberry Pi.

2.3 Характеристика об'єкту

Розробляюваний медіасервер призначено для використання в домашніх мультимедійних системах.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Мультимедійний сервер повинен складатись з:

- Програмного додатку.
- Системи адміністрування.

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

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це їх узгодженість.

3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- нормальний режим функціонування;
- аварійний режим функціонування.

Основним режимом функціонування є нормальний режим.

Для забезпечення нормального режиму функціонування системи необхідно виконувати вимоги і дотримуватись умов експлуатації програмного забезпечення і комплексу технічних засобів системи, вказані у відповідних технічних документах (технічна документація, інструкції з експлуатації і т.д.).

Аварійний режим функціонування системи характеризується відмовою одного або декількох компонент програмного і (або) технічного забезпечення. При цьому функції роботи системи продовжують підтримувати роботу системи моніторингу в межах базових налаштувань.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються інструменти діагностування основних процесів системи, які вмонтовані в операційну систему і програмне забезпечення, а також засоби для діагностики апаратного забезпечення.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки використанню додаткових програмних компонентів, а також передбачено розширення пам'яті для збільшення ресурсів медіатеки.

3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні наступних ситуацій:

- при збоях в системі електропостачання апаратної частини;
- при помилках в роботі апаратних засобів;
- при помилках, пов'язаних з програмним забезпеченням (ОС і драйвери пристроїв).

Для захисту апаратури від стрибків напруги і комутаційних завад повинні застосовуватися мережні фільтри.

3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби мати занулення або захисне заземлення.

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також аварійне ручне вимикання.

Загальні вимоги пожежної безпеки повинні відповідати нормам побутове електрообладнання. У разі пожежі не мають виділятися отруйні гази і дим. Після зняття електроживлення має бути доступне застосування будь-яких засобів пожежогасіння.

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

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від $+10^{\circ}\text{C}$ до $+35^{\circ}\text{C}$;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск 760 ± 25 мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, пристрої безперебійного живлення.

На підставі результатів тестування технічних засобів повинен проводитися аналіз причин виникнення виявлених дефектів і приймаються заходи по їх ліквідації.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче встановленого вимогами, що пред'являються до категорії

по класифікації документа, що діє, “Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем”.

Компоненти підсистеми захисту від НСД повинні забезпечувати:

- ідентифікацію користувача;
- перевірку повноважень користувача при роботі з системою;
- розмежування доступу користувачів.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку конфіденційної інформації, повинен відповідати вимогам класу захищеності згідно вимогам документу “Засоби обчислювальної техніки. Захист від несанкціонованого доступу до інформації. Показники захищеності від несанкціонованого доступу до інформації”.

3.4.1 Вимоги по збереженню інформації при аваріях

Інформація, при виникненні аварійних ситуацій повинна бути збережена на резервних носіях.

3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування високоякісним обладнанням (ЕОМ, монітор і інше обладнання), що має необхідні сертифікати відповідності і безпеки.

3.4.3 Вимоги до функцій (завдань), що виконуються системою:

- забезпечення зручного інтерфейсу;
- забезпечення зберігання медіафайлів;
- пошук необхідної інформації;
- забезпечення високої швидкодії.

4 Вимоги до документації

Документація повинна відповідати вимогам ЕСКД 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 доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

[illegible]

6 Додаткові умови виконання кваліфікаційної роботи

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

ся зміни

ї роботи

Виконання
роботи

02.02

27.03

02.05

26.05

02.06

09.06

15.06

22.06

2025

Додаток Б

Лістинг програми

```
void checkReminder() {
    DateTime now = rtc.now();

    if (currentReminderIndex == -1) return;

    Reminder reminderCurrent
reminders[currentReminderIndex];

    lcd.setCursor(0, 1);
    lcd.print("Next: ");
    if (reminderCurrent.hour < 10) lcd.print("0");
    lcd.print(reminderCurrent.hour);
    lcd.print(":");
    if (reminderCurrent.minute < 10) lcd.print("0");
    lcd.print(reminderCurrent.minute);
    lcd.print(" ");

    if (now.hour() == reminderCurrent.hour && now.minute() ==
reminderCurrent.minute) {
        buzz(500, 1000);
        delay(1000);

        int minDiff = 1440;
        int index = -1;
        for (int i = 0; i < reminderCount; i++) {
            int reminderMinutes = reminders[i].hour * 60
reminders[i].minute;
            int nowMinutes = now.hour() * 60 + now.minute();
            int diff = reminderMinutes - nowMinutes;

            if (diff <= 0) {
                diff += 1440;
            }

            if (diff < minDiff) {
                minDiff = diff;
                index = i;
            }
        }

        currentReminderIndex = index;
    }
}
```

Лістинг Б.2 - Текст методу handleSetTimeMode

```
void handleSetTimeMode() {
    DateTime now = rtc.now();
```

no

```

lcd.setCursor(0, 0);
lcd.print("Config Time      ");

lcd.setCursor(0, 1);
if (timeSettingStep == 0) {
    lcd.print(">");
} else {
    lcd.print(" ");
}

if (setHour < 10) lcd.print("0");
lcd.print(setHour);
lcd.print(":");

if (timeSettingStep == 1) {
    lcd.print(">");
} else {
    lcd.print(" ");
}

if (setMinute < 10) lcd.print("0");
lcd.print(setMinute);
lcd.print("      ");

int newPosition = encoder.getCount();
if (newPosition != prevPosition) {
    int delta = newPosition - prevPosition;
    prevPosition = newPosition;

    if (timeSettingStep == 0) {
        setHour += delta;
        if (setHour < 0) setHour = 23;
        if (setHour > 23) setHour = 0;
    } else {
        setMinute += delta;
        if (setMinute < 0) setMinute = 59;
        if (setMinute > 59) setMinute = 0;
    }

    lcd.clear();
}

if (shortPressDetected) {
    shortPressDetected = false;

    if (timeSettingStep == 0) {
        timeSettingStep = 1;
    } else {
        rtc.adjust(DateTime(now.year(),
                                now.month(),
                                now.day(), setHour, setMinute, 0));
        writeToEEPROM(0, setHour);
        writeToEEPROM(1, setMinute);
    }
}

```

```

        timeSettingStep = 0;
        currentMode = MODE_IDLE;
        initCurrentReminderIndex();
        lcd.clear();
    }
}

if (longPressDetected) {
    longPressDetected = false;

    rtc.adjust(DateTime(now.year(), now.month(), now.day(),
setHour, setMinute, 0));
    writeToEEPROM(0, setHour);
    writeToEEPROM(1, setMinute);

    timeSettingStep = 0;
    currentMode = MODE_IDLE;
    initCurrentReminderIndex();
    lcd.clear();
}
}

```

ЛІСТИНГ Б.3 - Текст методу handleSetIntervalMode

```

void handleSetIntervalMode() {
    int totalItems = reminderCount + 1;
    int currentIndex = encoder.getCount();

    if (currentIndex < 0) {
        currentIndex = 0;
        encoder.setCount(0);
    }
    if (currentIndex >= totalItems) {
        currentIndex = totalItems - 1;
        encoder.setCount(currentIndex);
    }

    lcd.setCursor(0, 0);
    if (currentIndex == 0) {
        lcd.print(">Add New      ");
    } else {
        lcd.print(" Add New      ");
    }

    lcd.setCursor(0, 1);
    if (currentIndex > 0) {
        Reminder r = reminders[currentIndex - 1];
        if (currentIndex != 0) lcd.print(">");
        else lcd.print(" ");

        if (r.hour < 10) lcd.print("0");
        lcd.print(r.hour);
        lcd.print(":");
    }
}

```

```

        if (r.minute < 10) lcd.print("0");
        lcd.print(r.minute);
        lcd.print(" ");
    } else {
        if (reminderCount > 0) {
            Reminder r = reminders[0];
            lcd.print(" ");
            if (r.hour < 10) lcd.print("0");
            lcd.print(r.hour);
            lcd.print(":");
            if (r.minute < 10) lcd.print("0");
            lcd.print(r.minute);
            lcd.print(" ");
        } else {
            lcd.print(" No reminders ");
        }
    }

    if (shortPressDetected) {
        shortPressDetected = false;
        lcd.clear();
        if (currentIndex == 0) {
            currentMode = MODE_ADD_REMINDER;
        } else {
            menuIndex = currentIndex - 1;
            currentMode = MODE_EDIT_REMINDER;
        }
    }

    if (longPressDetected) {
        longPressDetected = false;
        lcd.clear();
        currentMode = MODE_MENU;
    }
}

```

Лістинг Б.4 - Текст методу handleAddReminderMode

```

void handleAddReminderMode() {
    lcd.setCursor(0, 0);
    lcd.print("Add Reminder ");

    lcd.setCursor(0, 1);
    if (timeSettingStep == 0) {
        lcd.print(">");
    } else {
        lcd.print(" ");
    }

    if (setHour < 10) lcd.print("0");
    lcd.print(setHour);
    lcd.print(":");

    if (timeSettingStep == 1) {

```



```

        Serial.println("New    reminderCount:    "    +
String(reminderCount));
    }

    timeSettingStep = 0;
    currentMode = MODE_IDLE;
    lcd.clear();
}

}

if (longPressDetected) {
    longPressDetected = false;

    if (reminderCount < MAX_REMINDERS) {
        reminders[reminderCount].hour = setHour;
        reminders[reminderCount].minute = setMinute;
        reminderCount++;

        sortReminders();

        for (int i = 0; i < reminderCount && i < MAX_REMINDERS;
i++) {
            writeToEEPROM(3 + i * 2, reminders[i].hour);
            writeToEEPROM(4 + i * 2, reminders[i].minute);
            Serial.println("Reminder    added:    "    +
String(reminders[i].hour) + ":" + String(reminders[i].minute));
            delay(5);
        }

        writeToEEPROM(2, reminderCount);
        delay(5);

        Serial.println("New    reminderCount:    "    +
String(reminderCount));
    }

    timeSettingStep = 0;
    currentMode = MODE_IDLE;
    lcd.clear();
}

}

MAX_REMINDERS;

hour);
minute);
added: "
.minute));

void handleEditReminderMode() {
    if (!editInitDone) {
        setHour = reminders[menuIndex].hour;
        setMinute = reminders[menuIndex].minute;
        timeSettingStep = 0;
        editInitDone = true;
        lcd.clear();
    }

    lcd.setCursor(0, 0);

```

ЛІСТИНГ Б.5 - Текст методу handleEditReminderMode

```

lcd.print("Edit Reminder    ");

lcd.setCursor(0, 1);
if (timeSettingStep == 0) {
    lcd.print(">");
} else {
    lcd.print(" ");
}

if (setHour < 10) lcd.print("0");
lcd.print(setHour);
lcd.print(":");

if (timeSettingStep == 1) {
    lcd.print(">");
} else {
    lcd.print(" ");
}

if (setMinute < 10) lcd.print("0");
lcd.print(setMinute);
lcd.print("    ");

int newPosition = encoder.getCount();
if (newPosition != prevPosition) {
    int delta = newPosition - prevPosition;
    prevPosition = newPosition;

    if (timeSettingStep == 0) {
        setHour += delta;
        if (setHour < 0) setHour = 23;
        if (setHour > 23) setHour = 0;
    } else {
        setMinute += delta;
        if (setMinute < 0) setMinute = 59;
        if (setMinute > 59) setMinute = 0;
    }

    lcd.clear();
}

if (shortPressDetected) {
    shortPressDetected = false;

    if (timeSettingStep == 0) {
        timeSettingStep = 1;
    } else {
        reminders[menuIndex].hour = setHour;
        reminders[menuIndex].minute = setMinute;

        sortReminders();
    }
}

```

i++)

String

i++)

String

```

        for (int i = 0; i < reminderCount && i < MAX_REMINDERS;
i++) {
            writeToEEPROM(3 + i * 2, reminders[i].hour);
            writeToEEPROM(4 + i * 2, reminders[i].minute);
            Serial.println("Reminder    updated:    "    +
String(reminders[i].hour) + ":" + String(reminders[i].minute));
            delay(5);
        }

        timeSettingStep = 0;
        currentMode = MODE_IDLE;
        editInitDone = false;
        lcd.clear();
    }
}

if (longPressDetected) {
    longPressDetected = false;

    for (int i = menuIndex; i < reminderCount - 1; i++) {
        reminders[i] = reminders[i + 1];
    }
    reminderCount--;

    for (int i = 0; i < reminderCount && i < MAX_REMINDERS;
i++) {
        writeToEEPROM(3 + i * 2, reminders[i].hour);
        writeToEEPROM(4 + i * 2, reminders[i].minute);
        Serial.println("Reminder    re-written:    "    +
String(reminders[i].hour) + ":" + String(reminders[i].minute));
        delay(5);
    }

    writeToEEPROM(2, reminderCount);
    delay(5);

    editInitDone = false;
    currentMode = MODE_IDLE;
    lcd.clear();
}
}

```