

БАКАЛАВРСЬКА РОБОТА

КРБ.СІ-12.00.000 ПЗ

Група СІ-23-1К

Максим Ваврикович

2025

Івано-Франківський національний технічний університет нафти і газу  
Інститут інформаційних технологій  
Кафедра інформаційно-телекомунікаційних технологій та систем

---

Ваврикович Максим Петрович  
(прізвище, ім'я, по батькові)

УДК 004.738  
(індекс)

## БАКАЛАВРСЬКА РОБОТА

Розроблення безпроводної системи передавання даних із використанням  
технології LoRa  
(назва роботи)

---

Системна інженерія – Інтернет речей  
(назва освітньої програми)

---


174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

---

(шифр і назва спеціальності)

**Робота містить результати власних досліджень, використання ідей, результатів і  
текстів інших авторів мають посилання на відповідне джерело:**

Здобувач освітнього ступеня \_\_\_\_\_ М.П. Ваврикович  
(підпис, ініціали та прізвище здобувача)

Науковий керівник  \_\_\_\_\_ Паньків Христина Василівна, доц., к.т.н.  
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту  
Завідувач кафедри

\_\_\_\_\_ *Заміховський Л.М.*  
(посада) (підпис) (дата) (ініціали та прізвище)

# Івано-Франківський національний технічний університет нафти і газу

(повне найменування закладу вищої освіти)

Інститут Інформаційних технологій

Кафедра Інформаційно-телекомунікаційних технологій та систем

Освітній рівень бакалавр

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(шифр і назва)

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри ІТТС д.т.н., проф.**

**Л.М.Заміховський**

«\_\_\_» \_\_\_\_\_ 2025 року

## **ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ**

Вавриковичу Максиму Петровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення безпроводної системи передавання даних із використанням технології LoRa.

керівник роботи Паньків Христина Василівна, доц., к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від "15" квітня 2025 року  
№ 299/7

2. Строк подання студентом роботи 23.06.2025

3. Вихідні дані до роботи: Матеріали та результати отримані під час проходження переддипломної практики, алгоритм програмного забезпечення, методичні вказівки.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Порівняльний аналіз існуючих систем безпроводної передачі даних із використанням технології LoRa; Вибір програмних технологій та середовища розробки; Розробка програмної частини.

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

Структурна схема; Принципова схема, Перелік елементів; Блок схема.

6. Дата видачі завдання 15.03.2025 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Терміни виконання етапів проєкту	Примітка
1.	Видача завдань на бакалаврську роботу, розробка плану виконання проєкту, підбір літературних джерел	10.03.2025	<b>Виконав</b>
2.	Аналіз предметної області	17.03.2025	<b>Виконав</b>
3.	Розробка структурної схеми засобу автоматички для вирішення поставленого завдання	05.04.2025	<b>Виконав</b>
4.	Підбір елементної бази, розробка схеми електричної принципової	15.04.2025	<b>Виконав</b>
5.	Розробка програмного забезпечення для застосовуваного в проєкті типу ЕОМ	23.04.2025	<b>Виконав</b>
6.	Оформлення пояснювальної записки та графічних матеріалів	15.05.2025	<b>Виконав</b>
7.	Захист бакалаврської роботи	26.06.2025	

Студент

\_\_\_\_\_

(підпис)

Ваврикович М.П.

(прізвище та ініціали)

Керівник роботи

  
\_\_\_\_\_

(підпис)

Паньків Х.В.

(прізвище та ініціали)

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

SPI – (Serial Peripheral Interface) — послідовний синхронний повнодуплексний стандарт передачі даних.

I2C – (Inter-Integrated Circuit) — послідовна шина даних для зв'язку інтегральних схем.

IDE – (Integrated Development Environment) — Інтегроване середовище розробки.

OLED – (Organic Light-Emitting Diode) — Органічний світлодіод.

LoRa – (Long Range) – Велика дальність.

IoT – (Internet of Things) – Інтернет речей.

BLE – (Bluetooth Low Energy) – Низькопотужний Блютуз.

UART – (Universal Asynchronous Receiver/Transmitter) – протокол обміну даними між пристроями, який дозволяє передавати дані послідовно, а не паралельно.

SDK – (Software Development Kit) – Середовище розробки.

GPS – (Global Positioning System) – Глобальна система позиціонування.

MCU – (Microcontroller Unit) – Мікроконтролер.

OTA – (Over-the-Air) – Оновлення по повітряю.

IoT – (Internet of Things) – Інтернет речей.

ОС – Операційна система.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

## РЕФЕРАТ

Бакалаврська робота на тему: «Розроблення безпроводної системи передавання даних із використанням технології LoRa». Містить 62 сторінки, 5 таблиць, 24 рисунки. Перелік посилань нараховує 14 найменувань.

Метою роботи є аналіз, обґрунтування та розроблення безпроводної системи для передачі текстових даних, та даних місцезнаходження із використанням технології LoRa.

Об'єктом дослідження є саме безпроводна система для передачі даних із використанням технології LoRa, можливість її оновлення та застосування на практиці.

Методи дослідження. У процесі виконання бакалаврської роботи була використана сукупність загальних і специфічних методів дослідження, та розробки. В результаті були застосовані: абстрактно-логічний метод, метод логічних узагальнень, причинно-наслідковий аналіз, методи групування і т.п.

Результати роботи. За результатами проведеного дослідження було розроблено пристрій для безпроводної передачі даних із використанням технології LoRa, та додаток для ОС (операційна система) Android для зв'язку з даним пристроєм із використанням технології Bluetooth Low Energy.

Рекомендації щодо використання результатів роботи. Результати дослідження можуть бути використані для автономної, зручної, та швидкої передачі текстових даних, та даних місцезнаходження у разі відсутності покриття стандартного зв'язку, або під час аварійних відключень світла, чи стандартного зв'язку.

Результати впровадження досліджень. Результати дослідження були представлені керівнику бакалаврської роботи, який підтвердив працездатність пристрою, та який можна застосувати для приватного, чи промислового використання.

Ключові слова: безпроводна передача, пристрій, текст, GPS, BLE, Android.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

## ABSTRACT

Bachelor's thesis on the topic: "Development of a wireless data transmission system using LoRa technology". Contains 62 pages, 5 tables, 24 figures. The bibliography list consists of 14 items.

The purpose of the work is to analyze, substantiate and develop a wireless system for transmitting text data and location data using LoRa technology.

The object of the study is a wireless data transmission system using LoRa technology, the possibility of its updating and application in practice.

Research methods. In the process of performing the bachelor's thesis, a set of general and specific research and development methods was used. As a result, the following were applied: abstract-logical method, method of logical generalizations, cause-and-effect analysis, grouping methods, etc.

Results of the work. Based on the results of the research, a device for wireless data transmission using LoRa technology was developed, and an application for the Android OS (operating system) for communication with this device using Bluetooth Low Energy technology was developed.

Recommendations for using the results of the work. The results of the research can be used for autonomous, convenient, and fast transmission of text data and location data in the absence of standard communication coverage, or during emergency power outages or standard communication.


Results of the research implementation. The results of the research were presented to the supervisor of the bachelor's thesis, who confirmed the operability of the device, and which can be used for private or industrial use.

Keywords: wireless transmission, device, text, GPS, BLE, Android.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

## Зміст

	с.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	4
РЕФЕРАТ.....	5
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Загальні відомості про безпроводну систему передавання даних із використанням технології LoRa.....	9
1.2 Історія створення ESP.....	10
1.3 Безпроводна передача даних.....	12
1.4 Аналоги розроблювальної системи на ринку .....	14
1.5 Постановка завдання.....	23
2 КОНСТРУКТОРСЬКА ЧАСТИНА.....	24
2.1 Розроблення структурної схеми.....	24
2.2 Розроблення електричної принципової схеми.....	25
2.3 Обґрунтування вибору платформи ESP.....	27
2.4 Модулі використані у проєкті.....	29
3 РОЗРОБЛЕННЯ АЛГОРИТМУ РОБОТИ ПРИСТРОЮ.....	35
4 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	39
Висновки.....	45
Список використаних джерел.....	46
Додаток А – Код програми для мікроконтролера ESP32.....	48
Додаток Б – Схема електрична структурна.....	59
Додаток В – Схема електрична принципова.....	60
Додаток Г – Перелік елементів.....	61
Додаток Д – Блок схема.....	62

<i>КРБ.СІ-12.00.000 ПЗ</i>							
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вабрикович М.П.			Літ.	Арк.	Аркушів
Перевір.		Паньків Х.В.			H	7	62
Н.Контр.		Возний А.В.			<i>ІФНТУНГ СІ-23-1К</i>		
Затвердив		Заміховський Л. М.					

Розроблення безпроводної системи передавання даних із використанням технології LoRa  
Пояснювальна записка

## ВСТУП

Метою даної бакалаврської роботи є розроблення безпроводної системи передавання даних із використанням технології LoRa (Long Range) у поєднанні з мобільним додатком для ОС Android, та технологію BLE (Bluetooth Low Energy) для зв'язку із даним модулем. Така система призначена для забезпечення стабільної, енергоефективної та зручної передачі текстових повідомлень на відстані кількох кілометрів в умовах обмеженого доступу до традиційної мережевої інфраструктури або у середовищах із підвищеними вимогами до надійності зв'язку.

Призначення розроблюваної системи полягає у збиранні, обробці та передаванні текстової інформації між пристроями за допомогою LoRa-модуля, з можливістю надсилання і отримання повідомлень, а також із можливістю налаштовувати роботу модуля через Android-додаток. Отримані дані можуть бути переглянуті користувачем безпосередньо на смартфоні, що значно підвищує мобільність та зручність використання системи. Такий підхід дозволяє застосовувати систему в широкому спектрі сфер — від промислового моніторингу до персонального зв'язку в умовах надзвичайних ситуацій.

У процесі виконання бакалаврської роботи необхідно вирішити низку завдань, а саме: розробити структурну електричну схему пристрою, підібрати відповідні електронні компоненти, створити принципову схему цифрового пристрою, розробити блок-схему алгоритму роботи, написати програмне забезпечення для мікроконтролера та Android-дodatка, виготовити робочий зразок пристрою та провести тестування його функціональності. Також необхідно оформити повний комплект конструкторської документації.

Ця бакалаврська робота спрямована на досягнення результатів освітньо-професійної програми, зокрема формування здатності вирішувати складні спеціалізовані завдання та практичні проблеми у сфері автоматизації, що передбачає застосування сучасних методів, інструментів та засобів цифрової техніки й бездротових комунікацій.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Загальні відомості про безпроводну систему передавання даних із використанням технології LoRa

Автоматизовані системи передачі даних знаходять широке застосування в сучасному світі — як у повсякденному житті, так і в промисловості, науці, сільському господарстві та інших галузях. Можливість дистанційного отримання інформації в режимі реального часу значно підвищує ефективність роботи систем моніторингу та керування [1].

Модуль безпроводної системи для передачі даних із використанням технології LoRa, який розроблюється в даній бакалаврській роботі, являє собою пристрій на основі апаратної платформи ESP32 для отримання, та передачі текстових повідомлень, а також даних місцезнаходження другого користувача, принцип дії якого ґрунтується на LoRa радіомодулі SX1278, синьому OLED (organic light-emitting diode) дисплеї 128x64 пікселі для виведення даних, а також цифрового датчика температури DS18B20 для відстеження технічного стану відповідного модуля. Даний модуль використовує технологію BLE для зв'язку із Android-додатком, та дозволяє здійснювати обмін текстовою інформацією із використанням смартфона, а також налаштовувати роботу модуля під бажання користувача із даного смартфона.

При ввімкненні пристрою відбувається ініціалізація його компонентів, увімкнення радіомодуля, увімкнення BLE чіпа, та виведення даних для під'єднання спеціально створеної Bluetooth мережі на OLED дисплей. Після підключення користувача, Android-додаток очікує даних про користувача від модуля, а також дозволяє відправляти/відобразити текстові повідомлення, чи місцезнаходження від другого пристрою. Також додаток має окрему сторінку налаштувань, яка дозволяє змінювати Bluetooth назву пристрою, ім'я користувача, ключ шифрування повідомлень, а також увімкнення/вимкнення синього світлодіоду, який призначений для попередження користувача про прихід нового повідомлення, яке ще не було прочитано.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

## 1.2 Історія створення ESP

У світі стрімкого розвитку технологій IoT (Internet of Things) зростає потреба в недорогих, енергоефективних і водночас функціональних рішеннях для бездротового передавання даних. Відповіддю на ці потреби стали мікроконтролери серії ESP, розроблені компанією Espressif Systems — китайським виробником із головним офісом у Шанхаї, заснованим у 2008 році. Компанія спеціалізується на проектуванні енергоефективних чипів для бездротового зв'язку, зокрема Wi-Fi, Bluetooth, Zigbee та інших протоколів [2].

Першим продуктом компанії, який отримав світове визнання, став ESP8266 (рисунок 1.1), випущений у 2014 році. Це був недорогий модуль з вбудованим Wi-Fi, розрахований на використання в парі з іншими мікроконтролерами (через UART-інтерфейс (Universal Asynchronous Receiver/Transmitter)). Проте незабаром компанія відкрила SDK (Software Development Kit), що дозволило програмувати мікроконтролер безпосередньо, без потреби в зовнішньому MCU (Microcontroller Unit). Цей крок зробив ESP8266 унікальним на ринку — розробники отримали повноцінний однокристальний Wi-Fi-комп'ютер за вкрай низьку ціну [2].

Основні характеристики ESP8266:

- 32-бітний процесор Tensilica Xtensa L106, до 80/160 МГц;
- до 160 КБ оперативної пам'яті;
- підтримка Wi-Fi 802.11 b/g/n;
- SPI, I2C, UART, PWM, GPIO;
- енергоефективні режими сну;
- низька вартість (~2–3 USD);
- активна спільнота розробників (esp8266.com, GitHub, Arduino core) [2].

ESP8266 став основою для створення тисяч DIY-проектів (Do It Yourself), а також для комерційних IoT-пристроїв: розумних розеток, термостатів, сигналізацій, телеметрії тощо.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

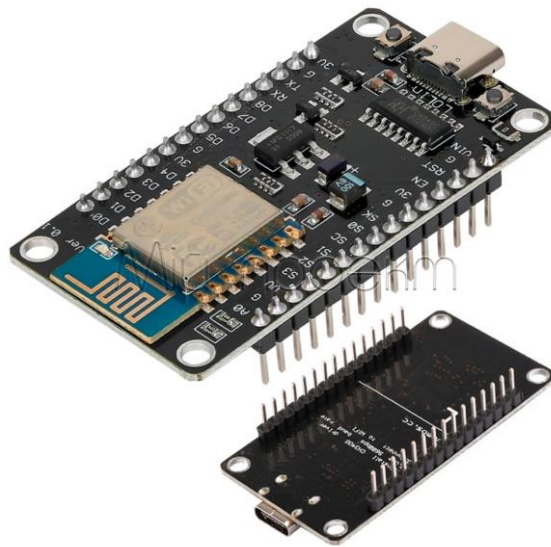


Рисунок 1.1 – Зображення мікроконтролера ESP8266

У 2016 році компанія представила ESP32 (рисунок 1.2) — потужніший, гнучкіший і сучасніший мікроконтролер, який значно розширив можливості IoT-розробників.

Ключові особливості ESP32:

- двоядерний 32-бітний процесор Xtensa LX6 (240 МГц);
- до 520 КБ SRAM і до 4 МБ flash-пам'яті;
- підтримка Wi-Fi 802.11 b/g/n та Bluetooth 4.2 (Classic і BLE);
- апаратне прискорення криптографії (AES, SHA, RSA);
- 34 GPIO з підтримкою цифрових/аналогових функцій;
- периферія: SPI, I2C, UART, CAN, DAC, ADC, сенсорні входи;
- розширені режими сну для зменшення енергоспоживання;
- сумісність з Arduino IDE, PlatformIO, Espressif IDF, MicroPython, ESPHome тощо [2].

На відміну від ESP8266, нова платформа забезпечила стабільнішу роботу з Bluetooth, додала можливість обробки кількох задач одночасно (FreeRTOS), підтримку датчиків, аудіо, криптографічних протоколів та хмарних сервісів [2].

У наступні роки Espressif продовжила розширювати серію, враховуючи потреби розробників у різних сферах:

- ESP32-S2/S3 — фокус на покращену безпеку, підтримку USB OTG, прискорення для нейромереж;
- ESP32-C3 — використання архітектури RISC-V, що є відкритою альтернативою ARM, зменшене енергоспоживання;
- ESP32-H2 — мікроконтролер із підтримкою IEEE 802.15.4, для протоколів Zigbee/Thread/Matter, орієнтований на Smart Home;
- ESP8685 / ESP8285 — мініатюрні варіанти ESP8266 із вбудованою flash-пам'яттю [2].



Рисунок 1.2 – Зображення мікроконтролера ESP32

На сьогодні мікроконтролери ESP є одними з найпопулярніших платформ для створення пристроїв з бездротовим зв'язком. Вони широко використовуються в системах моніторингу, автоматизації, телеметрії, розумного дому, аграрній індустрії та наукових розробках.

### 1.3 Безпроводна передача даних

Бездротова передача даних — це спосіб обміну інформацією між електронними пристроями без використання фізичного (дротового) з'єднання. Така передача здійснюється за допомогою радіохвиль, інфрачервоного випромінювання, мікрохвиль або лазерного сигналу в повітряному середовищі

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

(таблиця 1.1). Завдяки своїй гнучкості, мобільності та простоті в розгортанні, бездротові технології є основою для розвитку систем інтернету речей, телеметрії, мобільного зв'язку, розумних пристроїв, автоматизованих систем тощо [4].

Основні характеристики:

- відсутність фізичного кабельного з'єднання, що спрощує розгортання мереж у важкодоступних місцях;
- передача на різні відстані — від кількох сантиметрів (наприклад, NFC) до десятків кілометрів (наприклад, LoRa);
- частотний діапазон залежить від технології: 2.4 ГГц (Wi-Fi, Bluetooth), 433/868 МГц (LoRa), 900 МГц (Zigbee), 5 ГГц (Wi-Fi 5), тощо;
- протоколи передачі забезпечують надійність зв'язку, кодування даних, контроль помилок [4].

Таблиця 1.1 – Поширені технології бездротового зв'язку [4]

Технологія	Частота	Дальність	Особливості
<b>Wi-Fi</b>	2.4 / 5 ГГц	до 100 м	Висока швидкість, потребує енергоживлення
<b>Bluetooth/BLE</b>	2.4 ГГц	до 50 м	Низьке енергоспоживання, зв'язок із мобільними пристроями
<b>LoRa</b>	433 / 868 / 915 МГц	до 10 км	Низька швидкість, але надвелика дальність
<b>Zigbee</b>	2.4 ГГц	до 100 м	Сіткова архітектура, підходить для сенсорних мереж
<b>NFC</b>	13.56 МГц	до 10 см	Дуже близький зв'язок, часто в платіжних системах

Переваги бездротової передачі даних:

- мобільність: дозволяє пристроям залишатися підключеними без фіксованого розташування;
- гнучкість у встановленні: зручно для віддалених або важкодоступних об'єктів;
- масштабованість: легко додавати нові пристрої до мережі;
- низьке енергоспоживання (залежно від технології, наприклад BLE або LoRa).

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

- недоліки безпроводних систем:
- чутливість до перешкод (електромагнітні завади, будівельні конструкції);
- менша безпека, якщо не використовуються відповідні методи шифрування;
- обмеження в пропускній здатності порівняно з дротовими рішеннями (наприклад, Ethernet) [4].

#### 1.4 Аналоги розроблювальної системи на ринку

Як аналог безпроводної системи передавання даних із використанням технології LoRa можна привести самостійні LoRa комунікатори Armachat – сумісні із програмною платформою Meshtastic самостійні пристрої, які вже мають на борту необхідні для роботи, та використання даного пристрою елементи, як роз’єми для потужних антен, багаторазова батарея із можливістю заряджання (наприклад літій-іонні акумулятори), плата заряджання батареї (зазвичай на вхід приходять 5В через велике поширення блоків живлення даної напруги), великий кольоровий екран, повна клавіатура типу QWERTY, динаміки, та інші компоненти [10].

Meshtastic — це безкоштовна, відкрита програмно-апаратна платформа, призначена для створення децентралізованої мережі зв’язку, яка не потребує мобільного зв’язку або Інтернету. Вона дає змогу користувачам передавати текстові повідомлення, координати GPS та інші дані навіть у найвіддаленіших куточках, де звичний зв’язок відсутній. Завдяки використанню радіотехнології LoRa та саморегульованої mesh-сітки, Meshtastic став популярним інструментом серед туристів, волонтерів, тактичних груп і людей, які готуються до надзвичайних ситуацій [9].

Проект був започаткований приблизно у 2020 році розробником Кевіном Хестером (Kevin Hester) у співпраці зі спільнотою ентузіастів open source. Весь код Meshtastic є відкритим і розміщується на GitHub за ліцензією Apache 2.0, що дозволяє будь-кому вільно вивчати, змінювати та використовувати проєкт навіть у комерційних цілях. Основний репозиторій знаходиться за покликанням: <https://github.com/meshtastic> [9].

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Meshtastic створено для забезпечення зв'язку там, де інші технології недоступні або дорогі. Його використовують у походах, на природі, в експедиціях, при рятувальних роботах, у районах із поганим покриттям мобільного зв'язку, а також для резервного зв'язку у кризових ситуаціях. Платформа може слугувати як базовий засіб комунікації між кількома людьми або перетворитись на розгалужену мережу, що покриває великі відстані.

У центрі технології — LoRa — бездротовий стандарт, що забезпечує передачу даних на великі відстані з мінімальним енергоспоживанням. Пристрої Meshtastic працюють як вузли mesh-мережі: кожен із них може приймати, передавати та ретранслювати повідомлення до інших пристроїв, створюючи гнучку та децентралізовану систему зв'язку. Якщо один вузол не може дістатись до адресата, інші пристрої автоматично передадуть повідомлення далі, поки воно не досягне цілі (рисунок 1.3) [9].

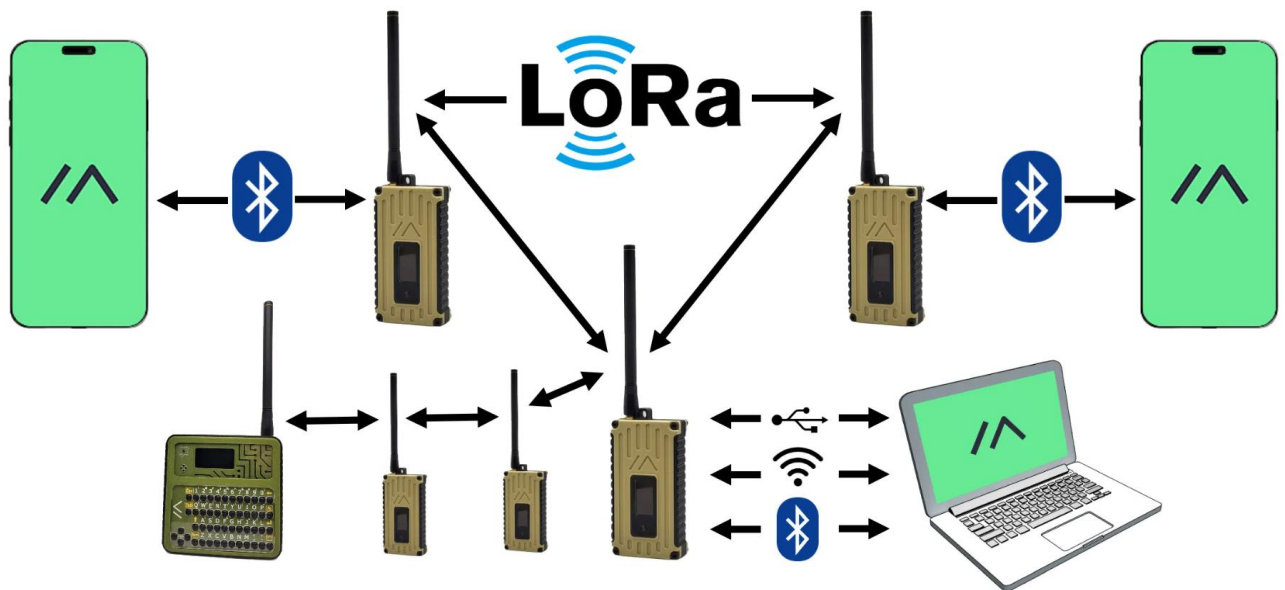


Рисунок 1.3 – Схема підключення пристроїв та передачі даних між ними [9]

Платформа підтримує різні модулі на базі мікроконтролера ESP32 у поєднанні з LoRa-модулями (популярні моделі: LilyGO T-Beam (рисунок 1.4), TTGO LoRa32, Heltec LoRa32 тощо). Усі пристрої можна конфігурувати за допомогою мобільного додатку (рисунок 1.5) (Android або частково iOS), веб-інтерфейсу (рисунок 1.6) або командного рядка через meshtastic-cli [9].



Рисунок 1.4 – Зображення мікроконтролера LilyGO T-Beam

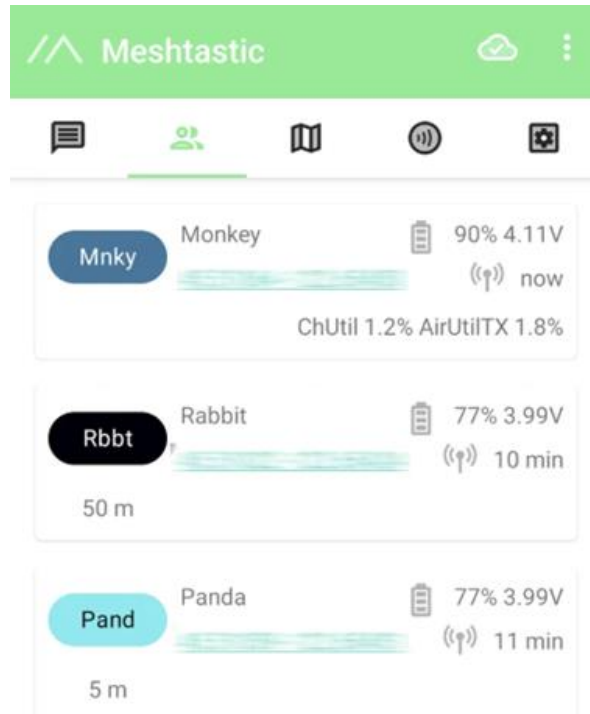


Рисунок 1.5 – Зображення вкладки доступних пристроїв у Android-додатку Meshtastic [9]

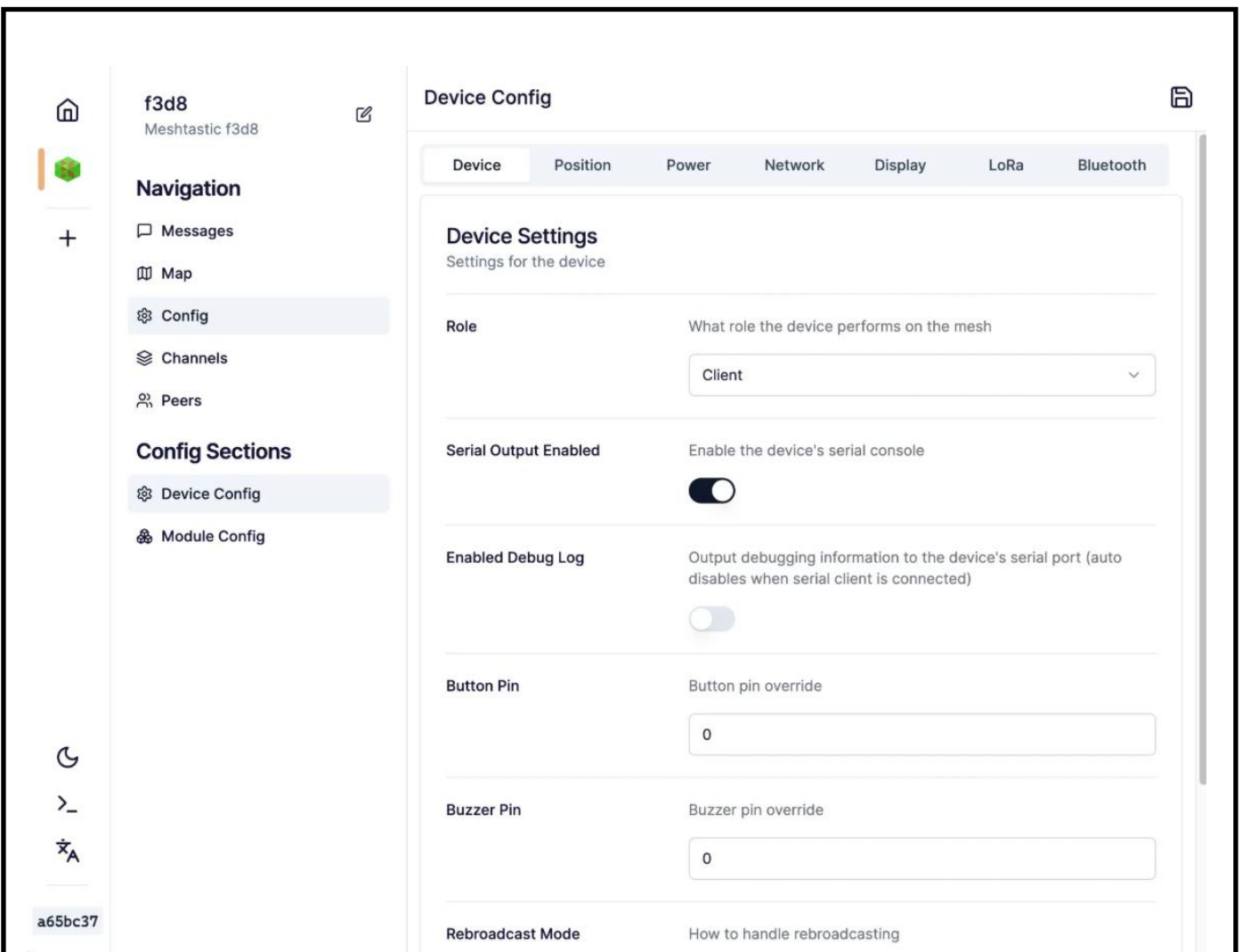


Рисунок 1.6 – Зображення вкладки налаштувань веб-інтерфейсу Meshtastic [9]

Передача даних у мережі шифрується за допомогою алгоритму AES128, а кожен канал має свій ключ безпеки. Це забезпечує приватність комунікації навіть у відкритих мережах. Крім текстових повідомлень, пристрої можуть автоматично надсилати координати GPS, рівень заряду батареї, температуру або інші показники від підключених датчиків [9].

Система складається з кількох важливих частин. По-перше, це сама прошивка, яку встановлюють на сумісний пристрій. По-друге, мобільні додатки, які дають змогу зручно надсилати повідомлення, налаштовувати параметри пристрою та візуалізувати маршрути. Додатково можна використовувати десктопні інтерфейси, у тому числі через CLI або веб-інтерфейс. Для інтеграції з іншими системами існує підтримка MQTT-брокерів, Home Assistant, Node-RED, Grafana тощо [9].

Щоб почати користуватись даною програмною платформою, слід обрати сумісний пристрій. Найпопулярнішими є LilyGO T-Beam та TTGO LoRa32. Після придбання потрібно прошити пристрій прошивкою Meshtastic, скориставшись офіційним інструментом meshtastic-flasher. Далі пристрій підключається до телефону або комп'ютера, де можна налаштувати канал зв'язку, задати ім'я, ключ безпеки та інші параметри. Створення мережі відбувається автоматично — достатньо, щоб кілька пристроїв працювали на одному каналі [9].

Типові сценарії використання даної програмною платформи:

- meshtastic часто застосовується в умовах, де інші форми зв'язку недоступні.

Наприклад:

- туризм і кемпінг: зв'язок між членами групи на великій території без необхідності у стільниковому покритті;
- рятувальні служби та волонтери: під час пошуків людей у горах, лісах або на руїнах;
- airsoft та військові ігри: координація команд без сторонніх засобів зв'язку;
- надзвичайні ситуації (EMCOMM): резервний зв'язок у разі природних катастроф або збоїв інфраструктури;
- сільське господарство та IoT: передача даних із сенсорів або керування пристроями на відстані [9].

Перевагами Meshtastic є повна автономність, можливість працювати без Інтернету, значна дальність зв'язку (до 10–15 км і більше), низьке енергоспоживання, відкритий вихідний код і активна спільнота. Платформа також добре масштабується — від двох пристроїв до десятків [9].

Із обмежень можна вказати невисоку швидкість передачі даних (що характерно для LoRa), необхідність початкового налаштування пристроїв, а також чутливість до рельєфу та умов навколишнього середовища. Підтримка iOS на сьогодні частково обмежена, а користувачам без технічного досвіду може знадобитись допомога на етапі конфігурації [9].

Meshtastic має активну міжнародну спільноту. Форум <https://meshtastic.discourse.group> — головне місце обговорень, де користувачі

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

діляться досвідом, запитують поради та обговорюють нові функції. Також існують активні групи у Telegram, на Reddit (r/meshtastic) та численні YouTube-канали з оглядами, інструкціями та прикладами використання. GitHub-проект має понад 4800 зірок, а кількість користувачів щомісяця зростає [9].

Meshtastic — це приклад того, як відкриті технології можуть стати ефективним інструментом автономного зв'язку в реальному світі. Від туристів до рятувальників, від хобістів до серйозних експедицій — платформа дає змогу залишатись на зв'язку там, де інші засоби не працюють. Дана платформа дозволяє створити простий, надійний і безкоштовний засіб зв'язку без Інтернету, коли неможливо скористатись звичними стандартами зв'язку.

Як приклад системи для безпроводної передачі даних на програмній платформі Meshtastic можна навести комунікатор «судного дня» Armachat (рисунок 1.7) (таблиця 2.2, таблиця 2.3, таблиця 2.4).

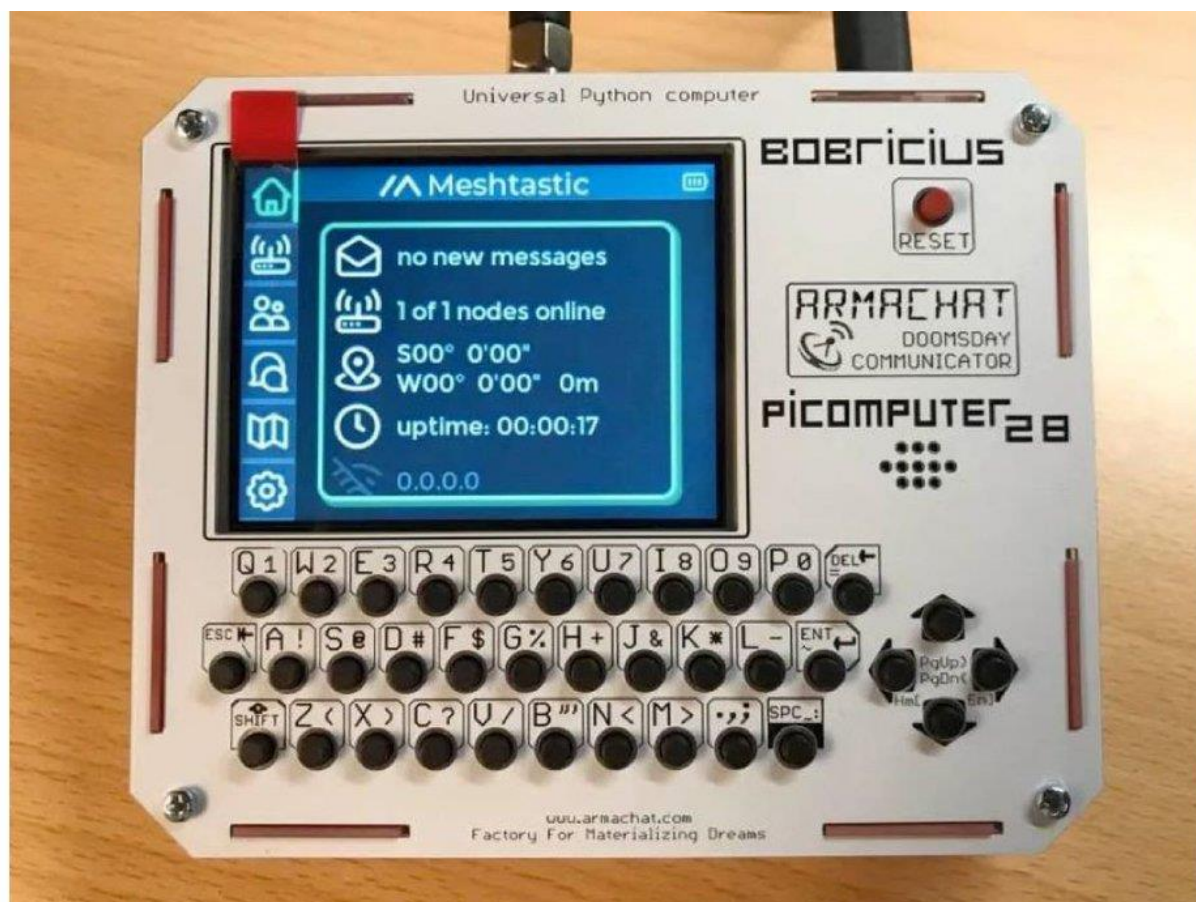


Рисунок 1.7 – Зображення пристрою комунікатора безпроводної передачі даних судного дня Armachat

Технічні характеристики:

- процесор: мікроконтролер ESP32S3 (необхідний для сумісності з MESHTASTIC);
- дисплей: 2,8" IPS кольоровий LCD з роздільною здатністю 320x240;
- вхід: 36-кнопкова розкладка клавіатури QWERTY;
- клавішний перемикач: високоякісні тактильні перемикачі Diptronic (тиск 100 гФ/1 Н), що забезпечують м'яке, але чуйне відчуття, схоже на механічне
- радіо: модуль RFM95 LoRa для дальнього зв'язку;
- програмне забезпечення: прошивка Armachat, реалізована на CircuitPython - повністю з відкритим вихідним кодом і прозора;
- живлення: зарядка акумулятора USB-C для стандартного літієвого елемента 18650 (акумулятор не входить в комплект);
- розміри: 115 мм × 95 мм × 18 мм, включаючи внутрішній батарейний відсік [10].

Meshtastic пропонує комплексну мережеву платформу зі встановленою підтримкою спільноти. Ця реалізація функціонує як автономний комунікатор і шлюз.

- інтеграція з екосистемою: працює як шлюз, підключений через Bluetooth до програм телефону;
- гнучкість введення: надсилайте повідомлення через фізичну клавіатуру або підключену програму для телефону;
- обмеження відображення: на екрані відображається лише останнє отримане повідомлення (історія повідомлень на пристрої немає);
- підтвердження доставки: просте підтвердження останнього повідомлення (повне відстеження доставки доступне в додатку);
- підключення: інтерфейси WiFi і Bluetooth;
- конфігурація: всі параметри необхідно налаштувати через мобільний додаток;
- сумісність: працює з будь-яким пристроєм, сумісним з Meshtastic — не

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

потрібно купувати кілька одиниць цього конкретного обладнання [10].

Таблиця 2.2 – Основні компоненти необхідні для збірки [10]

Компонент	Кількість	Опис
BPI-PicoW-S3	1	Головний мікроконтролера
RFM95W	1	LoRa радіомодуль
2.8" IPS screen	1	320x240 ST7789V (IPS HC28010-D)
DTS63K	37	Тактильні кнопки (7мм висота, 1N)

Таблиця 2.3 – Електронні компоненти [10]

Компонент	Кількість	Опис
BSS123	1	Транзистор
10R 0603	4	Резистор
100k 0603	1	Резистор
200k 0603	1	Резистор
100n 0603	1	Конденсатор
100n 0603	1	Конденсатор
100uF 1206	1	Конденсатор
LPT1109DS	1	П'єзoeлемент

Таблиця 2.4 – Опціональні компоненти (Нижній батарейний відсік) [10]

Компонент	Кількість	Опис
JS102011SAQN C&K	1	Боковий перемикач
USB-C зарядний модуль	1	Роз'єм Type-C USB 5V 1A TP4056

Компонент	Кількість	Опис
ВНС-18650-1P	1	Тримач батареї

Перевірений діапазон зв'язку (максимальна потужність, коефіцієнт розповсюдження 12):

- 700 м: пристрій розміщено на садовому столі на рівні землі;
- 1500 м: блок розташований у вікні приблизно на висоті 4 м над землею;
- 4300 м: пристрій встановлено у вікні горища на висоті 10 м над землею (переривчасте з'єднання, не завжди надійне) [10].

Також у ролі аналога розроблювальної системи безпроводної передачі даних із використанням технології LoRa можна навести передавач від tuzu works – H1 (рисунок 1.8).



Рисунок 1.8 – Зображення передавача H1

Це готовий до використання пристрій, як вже зібраний вручну, перевірено та прошито на останню прошивку. Заряджання батареї здійснюється через порт USB-C [11].

Даний пристрій при покупці включає:

- плата Heltec V3;
- SMA Pigtail;
- універсальна антена Stubby 915/868 МГц;
- літій-іонний полімерний акумулятор ємністю 1200 мАг [Кабель USB-C не входить в комплект].
- розміри: 45,5 мм x 64,6 мм x 19,5 мм;
- вага: 59 г [11].

Даний пристрій має мінімальну компонентну базу, маленький дисплей, та маленький розмір, але за допомогою такого рішення підвищилась надійність пристрою, а для зв'язку із даним пристроєм достатньо використати смартфон із відповідним android-додатком Meshtastic.

### 1.5 Постановка завдання

Розроблюваний у даній бакалаврській роботі модуль безпроводної системи передавання даних із використанням технології LoRa повинен бути таким, щоб його можна було встановлювати в виробничих та побутових приміщеннях та повинен відповідати умовам кліматичного виконання згідно ДСТУ 15150-69, УХЛ4 [3].

Прилад, що проектується, орієнтований на використання його у мікрокліматичних районах з помірним та холодним кліматом та відноситься до категорії виробів призначених для експлуатації в приміщеннях з штучно регульованими кліматичними умовами. Експлуатація приладу повинна задовольняти умови ДСТУ 16019, С1. До групи апаратури, вказаної в ДСТУ 16019, С1 відноситься стаціонарна апаратура, яка встановлюється в опалюваних наземних і підземних спорудах [3].

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

## 2 КОНСТРУКТОРСЬКА ЧАСТИНА

### 2.1 Розробка структурної схеми

Безпроводна система передавання даних із використанням технології LoRa складається із декількох основних частин. Розглянемо структурну схему даної системи, яка наведена на рисунку 2.1. До неї входить LoRa радіомодуль, OLED дисплей, датчик температури, та сам мікроконтролер.

ESP сам створює BLE мережу, до якої необхідно підключитись із спеціального Android-додатку.

Дисплей слугує виводом інформації про назву BLE мережі, а також може відображати текст, що було отримано нове повідомлення, яке ще не було прочитане. Після підключення до Android-додатку користувач може надсилати/приймати текстові повідомлення, та повідомлення місцезнаходження, а також налаштовувати сам пристрій.

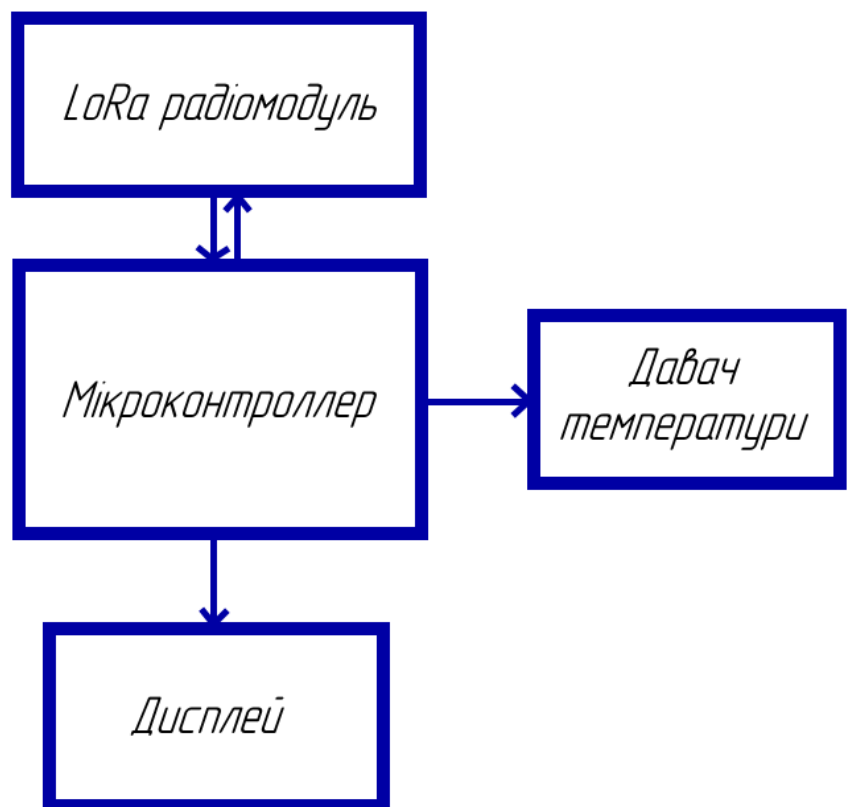


Рисунок 2.1 – Структурна схема безпроводної системи передавання даних із використанням технології LoRa

					КРБ.СІ-12.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

- мікроконтролер – ESP32;
- дисплей – OLED дисплей 128x64 SSD1306;
- LoRa радіомодуль – SX1278;
- датчик температури – DS18B20.

## 2.2 Розробка електричної принципової схеми

Принципова схема безпроводної системи передавання даних із використанням технології LoRa наведена на рисунку 2.2.

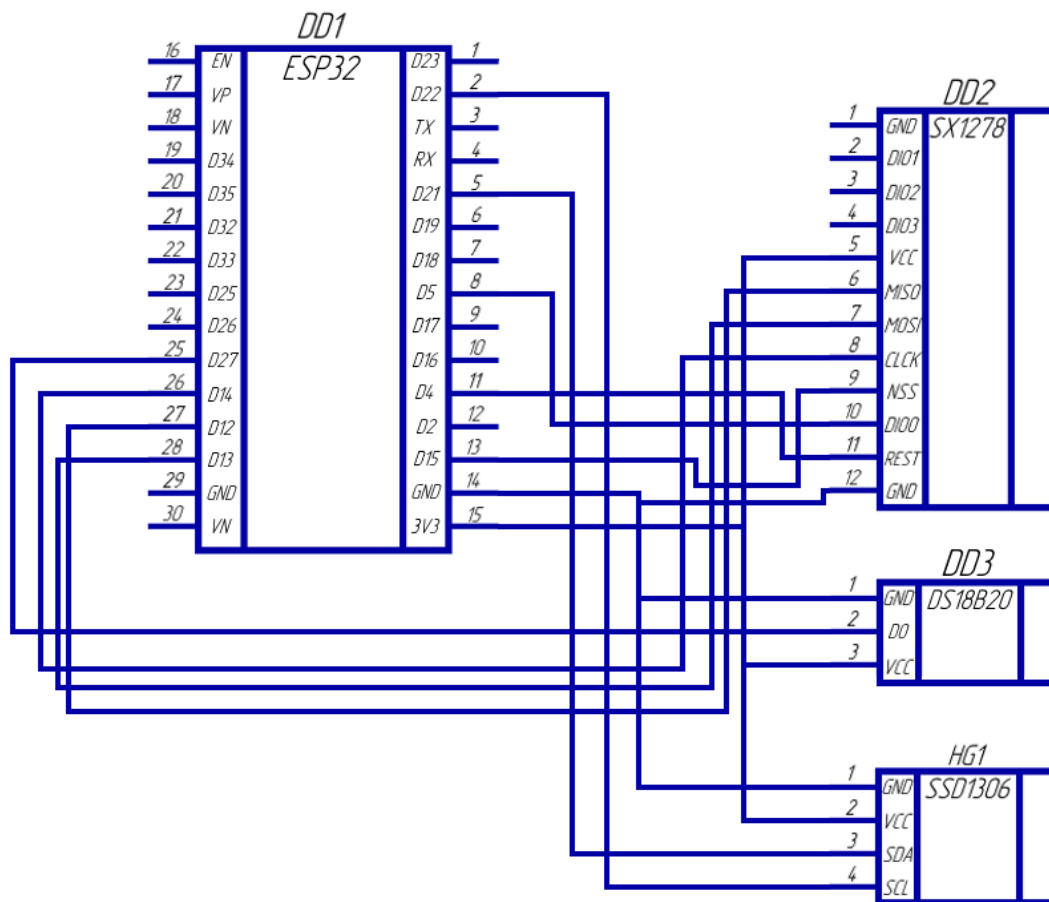


Рисунок 2.2 – Принципова схема безпроводної системи передавання даних із використанням технології LoRa

Фотографія зібраного модуля проєкту безпроводної системи передавання даних із використанням технології LoRa зображено на рисунку 2.3 та на рисунку 2.4.

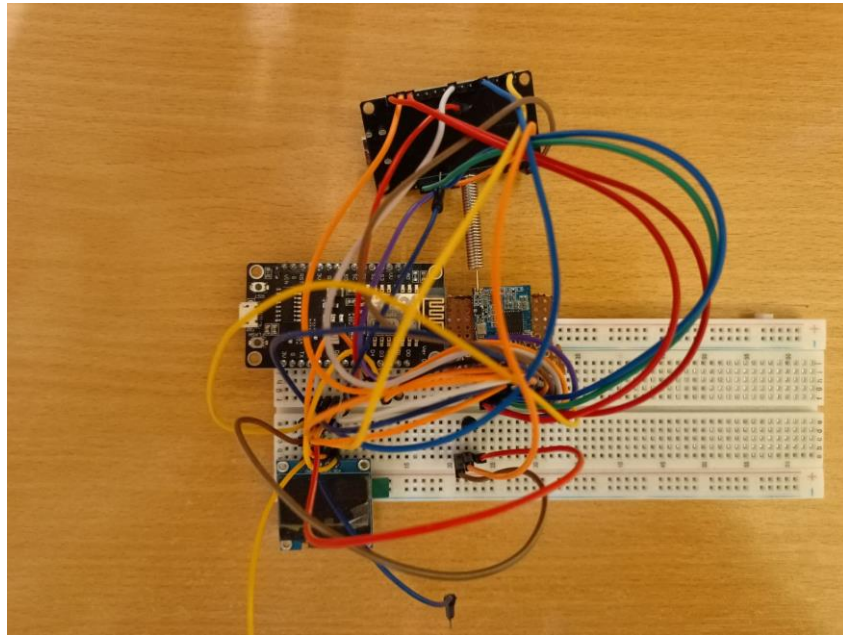


Рисунок 2.3 – Зображення першого зібраного пристрою безпроводної передачі даних із використанням технології LoRa

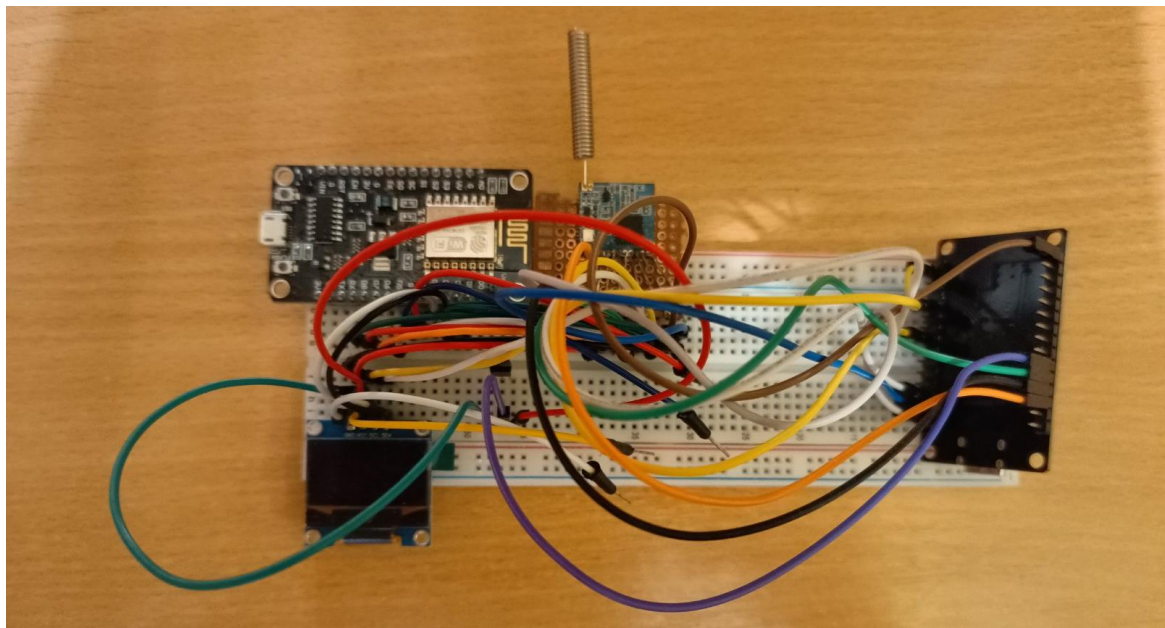


Рисунок 2.4 – Зображення другого зібраного пристрою безпроводної передачі даних із використанням технології LoRa

Живлення до пристрою подається від USB порта на платі мікроконтролера від джерела живлення +5В.

До ESP32 підключені роз'єми модулів SX1278, SSD1306, та DS18B20.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

До роз'єму SSD1306 приходять по шині живлення +3.3В від мікроконтролера, та використовує технологію I2C для підключення до мікроконтролера.

Роз'єм SX1278 підключений по інтерфейсу SPI до мікроконтролера для швидкого обміну інформацією, та керування модулем. Живлення модуль отримує по лінії +3.3В від мікроконтролера.

Вивід DS18B20 підключений до 27-го виводу GPIO мікроконтролера по протоколу One-Wire. Живлення давач отримує по лінії +3.3В від мікроконтролера.

### 2.3 Обґрунтування вибору платформи ESP

У межах бакалаврської роботи для побудови бездротової системи передавання даних було обрано мікроконтролер ESP32 виробництва компанії Espressif Systems. Основним критерієм вибору цього мікроконтролера стала необхідність реалізації системи, що поєднує енергоефективність, високу обчислювальну здатність, наявність вбудованих інтерфейсів бездротового зв'язку, а також гнучкість при розробці мобільного інтерфейсу [2].

ESP32 (таблиця 2.1) — це сучасний 32-бітний мікроконтролер, який поєднує в собі двоядерний процесор, великий набір периферії та підтримку бездротових інтерфейсів Wi-Fi та Bluetooth/BLE. Завдяки вбудованому Bluetooth Low Energy (BLE) цей мікроконтролер ідеально підходить для інтеграції з Android-додатками, що дозволяє зручно передавати дані між користувачем і пристроєм у режимі реального часу без необхідності постійного підключення до Wi-Fi або мобільної мережі [2].

Крім того, ESP32 забезпечує низьке енергоспоживання, що є критично важливим у системах з живленням від батарей або в умовах віддаленого розташування [2].

Таблиця 2.1 – Технічні характеристики мікроконтролера ESP32 [2]

Параметр	Значення / Особливості
----------	------------------------

<b>Процесор</b>	Xtensa® Dual-Core 32-bit LX6 (до 240 МГц)
<b>Оперативна пам'ять (RAM)</b>	520 КБ SRAM
<b>Пам'ять для зберігання</b>	Підтримка зовнішньої SPI Flash до 16 МБ
<b>Wi-Fi</b>	IEEE 802.11 b/g/n (2.4 ГГц)
<b>Bluetooth</b>	Bluetooth v4.2 BR/EDR та BLE
<b>GPIO</b>	До 34 універсальних цифрових входів/виходів
<b>Аналогові входи (ADC)</b>	До 18 каналів (роздільна здатність 12 біт)
<b>Цифрові виходи PWM</b>	Підтримка PWM на більшості пінів
<b>UART / SPI / I2C / I2S</b>	Підтримка стандартних інтерфейсів для комунікації з периферією
<b>DAC</b>	2 канали цифрово-аналогового перетворення
<b>Вбудований температурний сенсор</b>	Так
<b>Підтримка енергозбереження</b>	Режими Deep Sleep, Light Sleep, Modem Sleep
<b>Живлення</b>	2.2–3.6 В (типово 3.3 В)
<b>Розміри</b>	18×25 мм

Переваги використання ESP32 у проєкті:

- вбудований Bluetooth Low Energy — дозволяє створити прямий зв'язок з Android-додатком без додаткових модулів;
- наявність апаратної підтримки LoRa через SPI для роботи з модулем SX1278;
- достатня обчислювальна потужність для обробки сенсорних даних (наприклад, з датчика температури DS18B20);
- режими енергозбереження, що знижують енергоспоживання у періоди неактивності;
- підтримка OTA (Over-the-Air) оновлень, що спрощує розгортання оновлень у польових умовах.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

## 2.4 Модулі використані у проєкті

### 2.4.1 OLED дисплей 128x64 SSD1306

OLED дисплей 128x64 SSD1306 – це яскравий, економічний, дуже контрастний OLED дисплей гідно прикрасить будь-яку конструкцію, для якої важливі розміри і зовнішній вигляд. Контрастність дисплея дозволить впевнено зчитувати з нього інформацію навіть при дуже яскравому світлі. Просте підключення і поширений I2C інтерфейс дозволить підключити декілька дисплеїв до будь-якого мікроконтролера або міні-комп'ютера. Адресу можна вибрати перепайкою перемички. Типова адреса I2C 0x3c.

#### Характеристики:

- розмір екрану 0.96;
- тип екрану OLED;
- роз'єм 4-pin;
- напруга живлення +3.3~ +6 В;
- роздільна здатність дисплея 128 \* 64 пк;
- товщина 4 мм;
- ширина 27 мм;
- висота 28 мм;
- вага 5 г.

#### Специфікація:

- драйвер OLED модуля: SSD1306;
- кут огляду:> 160 градусів;
- напруга живлення: 3.3-6V;
- рівні вхідних сигналів: 3.3V / 5V.

#### Цоколювання роз'єму:

- GND: загальний;
- VCC: напруга живлення;
- SDA: шина даних;
- SCL: шина тактування даних [6].

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Підключення OLED дисплею SSD1306: (рис 2.5).

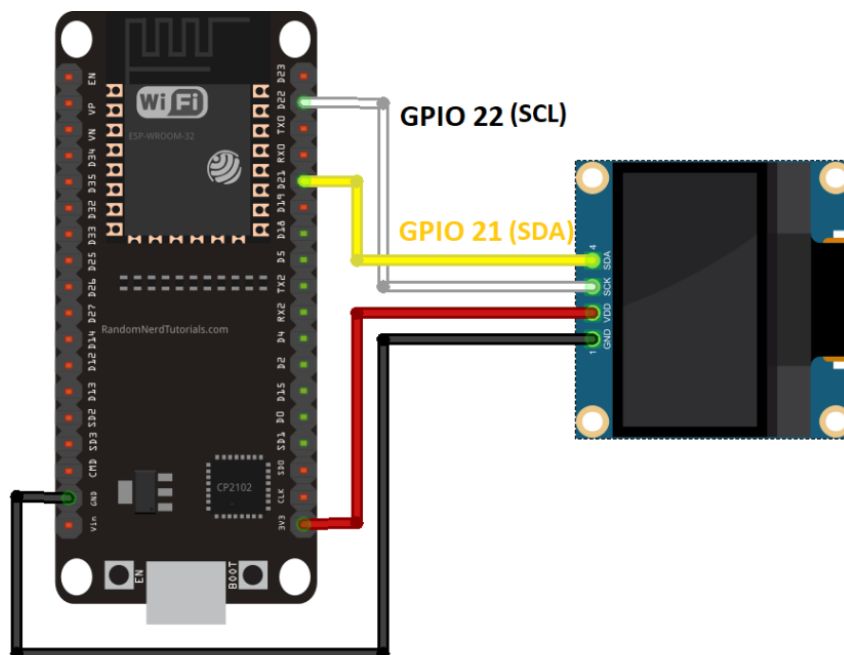


Рисунок 2.5 – Підключення OLED дисплею SSD1306

#### 2.4.2 LoRa радіомодуль SX1278

LoRa радіомодуль SX1278 – це приймач оснащений модемом дальнього радіуса дії LoRa®, який забезпечує над широкосмуговий зв'язок з розширеним спектром і високу стійкість при мінімальному споживанні струму. Використовуючи запатентовану Semtech техніку модуляції LoRa, SX1278 може досягти чутливості понад -148 дБм, використовуючи недорогі мікросхеми та супутні комплектуючі. Висока чутливість в поєднанні з вбудованим підсилювачем потужності +20 дБм забезпечує кращий в галузі бюджет каналу зв'язку, що робить його оптимальним для будь-якого застосування, що вимагає частотного діапазону або надійності. LoRa-модуль також забезпечує переваги як в блокуванні, так і в вибірковості в порівнянні з традиційними методами модуляції, вирішуючи традиційний компромісний задум між діапазоном, стійкістю до перешкод та споживанням енергії.

Характеристики:

- метод модуляції: FSK/GFSK, LoRa;
- тип зв'язку: напівдуплексних зв'язок;

- робоча частота: 433 МГц;
- відхилення каналу (ADJ): 56 dBm;
- чутливість RX: -148 dBm;
- робочий діапазон: ISM multiband;
- інтелектуальне скидання, монітор низької напруги, синхронний wakeup, режим низької потужності, режим сну;
- енергоспоживання в режимі прийому: 12 ~ 13 мА;
- внутрішній буфер: 256 байт FIFO TX/RX;
- виявлення сигналу каналу передачі даних: ISSI;
- режим передачі: FIFO/прямий режим (рекомендований режим пакета FIFO).

Варіанти конфігурації:

- AFC;
- пробудження при наявності радіосигналу;
- зниження споживаної потужності;
- виявлення несучої;
- FEC корекція помилок;
- ЕС шифрування.
- особливості використання:
  - напруга має бути в діапазоні 1,8 ~ 3,6 В. Якщо напруга живлення перевищить 3,6 В мікросхема модуля вийде з ладу. Рекомендоване напруга 3,3 В;
  - антена повинна розташовуватися далеко від системи заземлення;
  - максимальна швидкість SPI інтерфейсу не більше 10 МГц;
  - виходи DIO0 і DIO2 є звичайними виводами введення/виведення і можуть управлятися користувачем;
  - у багатоканальній системі рознос частот каналів має становити не менше 1 МГц. Якщо рознос менше 1 МГц це призведе до взаємних перешкод пристроїв.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		



### 2.4.3 Цифровий давач температури DS18B20

DS18B20 – це давач температури із діапазоном вимірюваних температур від -55 до +125 °С. Зчитуваний з приладу цифровий код є прямим безпосереднім кодом вимірюваного значення температури і не потребує додаткових перетворень. Програмована користувачем роздільна здатність вбудованого АЦП може бути змінена в діапазоні від 9 до 12 розрядів вихідного коду. Абсолютна похибка перетворення менше 0.5 °С в діапазоні контрольованих температур -10 до +85 °С. Максимальний час повного 12-ти розрядного перетворення ~750 мс (при роздільній здатності 12 розрядів). Для підключення потрібно резистор 4.7 кОм.

Внутрішня енергонезалежна пам'ять температурних установок забезпечує запис довільних значень верхньої та нижньої межі установок. Крім того, мікросхема містить вбудований логічний механізм пріоритетної сигналізації в лінію про факт виходу температури за один з обраних порогів. Вузол 1-Wire-інтерфейсу приладу організований таким чином, що існує теоретична можливість адресації необмеженої кількості подібних пристроїв на однопровідній лінії.

Основні характеристики:

- корпус: ТО-92;
- розрядність: 9-12 біт;
- час перетворення: 750nS (максимум);
- точність виміру  $\pm 0.5\%$  в області температур: -10 ...+85°C;
- напруга живлення для точності виміру  $\pm 0.5\%$ : 3,0-5,5В [8].

Підключення давача температури: (рис 2.7).

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

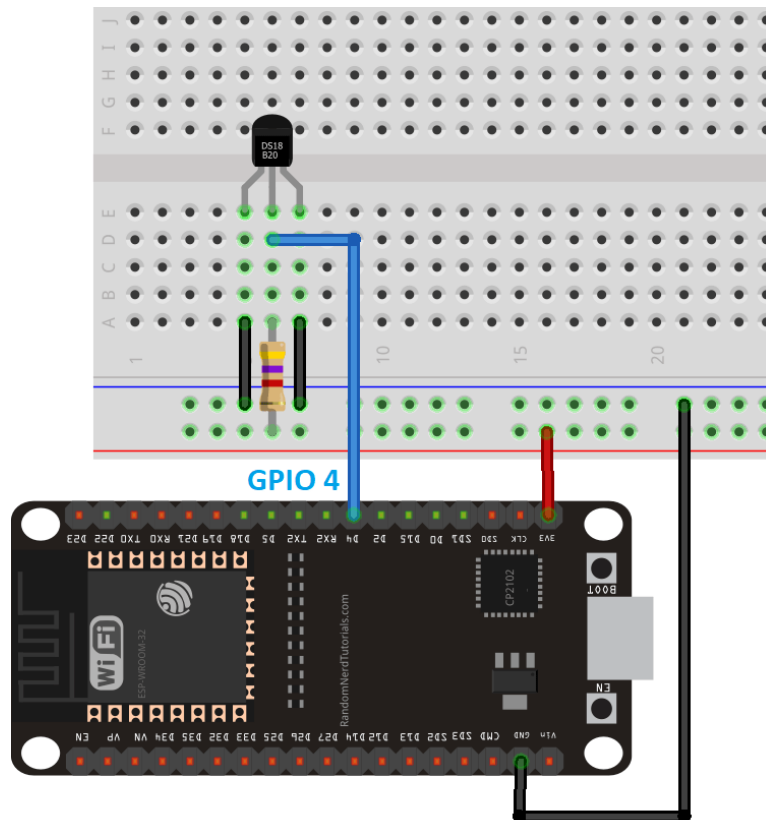


Рисунок 2.7 – Підключення датча температури DS18B20

У даному розділі було підібрано оптимальну, та потужну компонентну базу для розроблення безпроводної системи для передачі даних із використанням технології LoRa, відображення системних даних на дисплеї, та наявністю технології BLE для зв'язку із Android-додатком для користування пристроєм.

## 3 РОЗРОБКА АЛГОРИТМУ РОБОТИ ПРИСТРОЮ

### 3.1 Вибір мови програмування для апаратної платформи ESP

Сучасні мікроконтролери підтримують програмування різними мовами, які за допомогою компіляторів або інтерпретаторів перетворюються на низькорівневий код, зрозумілий апаратній частині пристрою. Хоча використання мови Assembler, або C/C++ забезпечує найвищу продуктивність і мінімальне використання ресурсів мікроконтролера, вона є складною у розробці та потребує значних зусиль і часу від розробника [1].

Мікроконтролер ESP32, завдяки своїм апаратним можливостям, таким як підтримка Bluetooth Low Energy, Wi-Fi, наявність великої кількості GPIO, АЦП, ЦАП і достатнього обсягу пам'яті, дозволяє ефективно використовувати високорівневі мови програмування без суттєвих втрат у продуктивності. Це відкриває широкі можливості для реалізації сучасних IoT-рішень та систем бездротової передачі даних [2].

У даній бакалаврській роботі для програмування мікроконтролера ESP32 було використано мову MicroPython — легку та оптимізовану версію Python, спеціально створену для роботи на мікроконтролерах. Основні причини вибору цієї мови програмування [12]:

- простий та зрозумілий синтаксис, що значно пришвидшує розробку;
- можливість виконання команд у реальному часі через інтерактивну консоль REPL;
- великий вибір бібліотек для роботи з периферією, сенсорами, дисплеями, модулями зв'язку та мережевими протоколами;
- відсутність потреби у компіляції — код виконується безпосередньо на пристрої, що полегшує тестування та налагодження [12].

Як середовище розробки було використано Thonny IDE, яке підтримує роботу з MicroPython. Воно автоматично розпізнає підключений мікроконтролер, має зручний графічний інтерфейс і дозволяє швидко завантажувати, редагувати й виконувати програми без потреби у складних налаштуваннях. Це особливо

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

важливо для швидкої розробки та зручності використання [13].

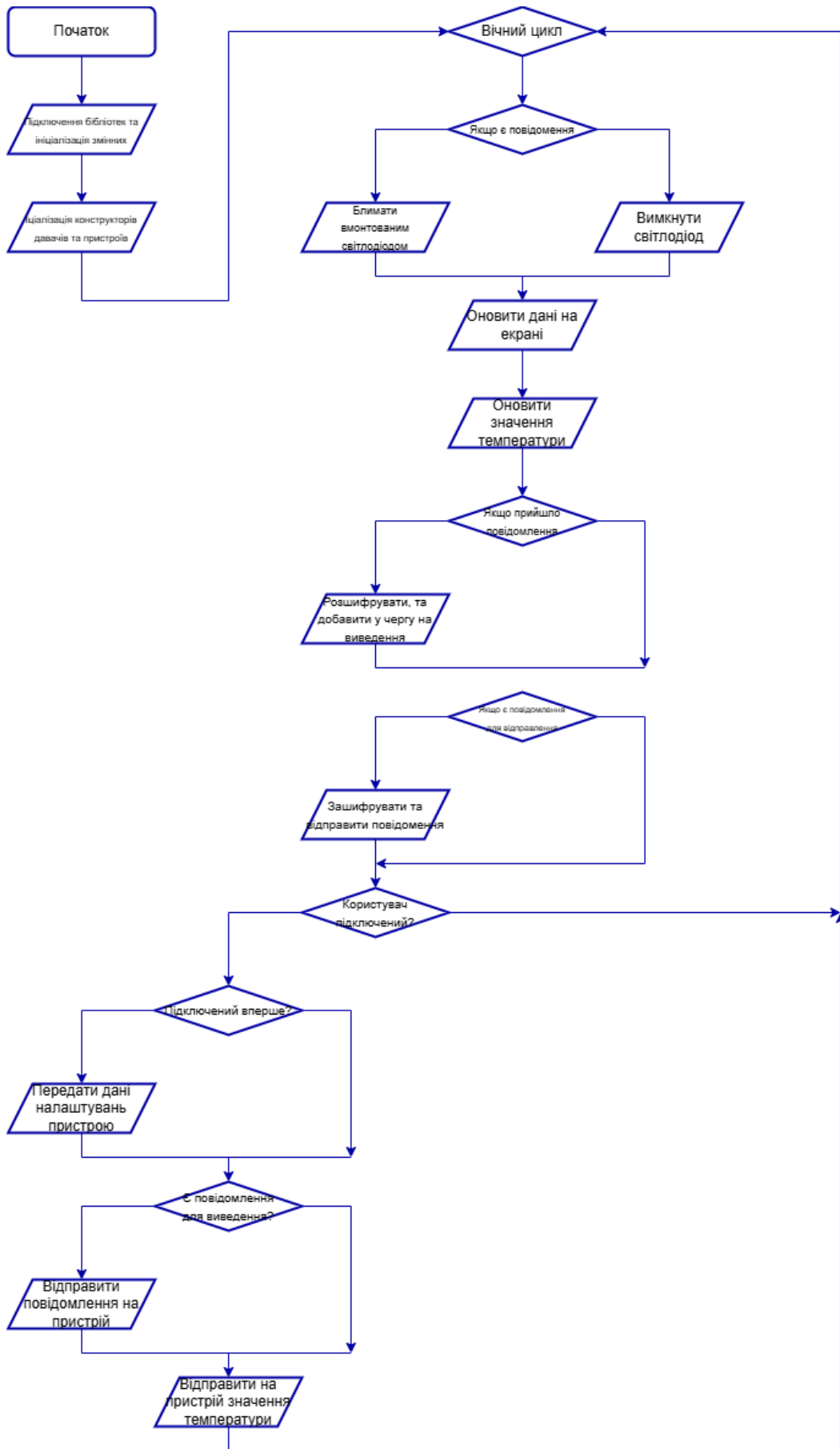


Рисунок 3.1 – Блок схема програми для апаратної платформи

Змн.	Арк.	№ докум.	Підпис	Дата

КРБ.СІ-12.00.000 ПЗ

Арк.

36

Для зручного керування пристроєм з боку користувача та організації двостороннього обміну даними по Bluetooth було розроблено мобільний Android-додаток у середовищі MIT App Inventor. MIT App Inventor — це візуальне середовище програмування, що дозволяє створювати Android-застосунки без глибоких знань мов програмування за допомогою блокової структури. Завдяки блочній системі побудови логіки, розробка додатку для приймання та надсилання повідомлень через Bluetooth Low Energy була реалізована швидко та ефективно. Додаток забезпечує зручний інтерфейс користувача, можливість перегляду отриманих даних від ESP32, а також надсилання параметрів налаштувань або повідомлень до модуля [14].

Програма, яка розроблена відносно даної блок схеми імпортує необхідні для роботи бібліотеки для роботи із радіомодулем, дисплеєм, давачем температури, BLE, затримками, та компонентами мікроконтролера, шифруванням та закодуванням повідомлень; оголошує конструктори для роботи із пристроями; ініціалізує дисплей та радіомодуль; активує BLE мережу; виводить відповідну інформацію на дисплей; у циклі відбувається постійне очікування на під'єднання користувача до даної BLE мережі із Android-додатка (рисунок 3.2), та очікується на його під'єднання; при під'єднанні модуль пересилає основні дані про себе та користувача у Android-додаток; при наявності вхідних повідомлень надсилає їх у Android-додаток; при наявності вихідних повідомлень, модуль приймає їх, декодує, шифрує, та надсилає до іншого модуля; при зміні налаштувань (рисунок 3.3), модуль приймає дані через BLE, та здійснює відповідні зміни у системі, та записує зміни у пам'ять.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

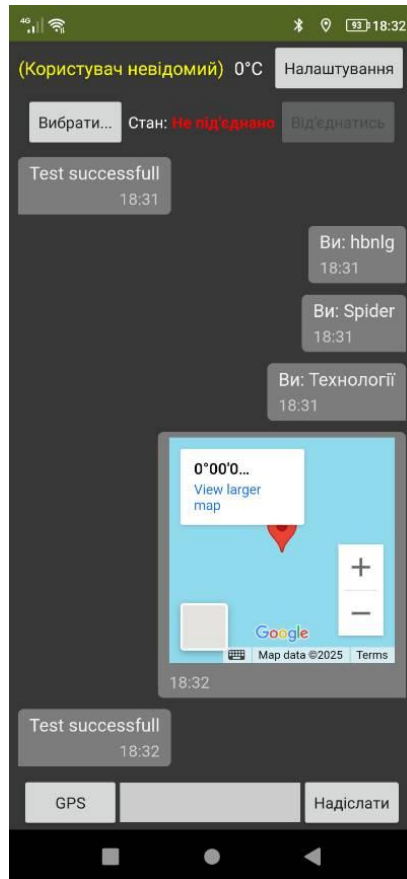


Рисунок 3.2 – Зовнішній вигляд Android-додатка

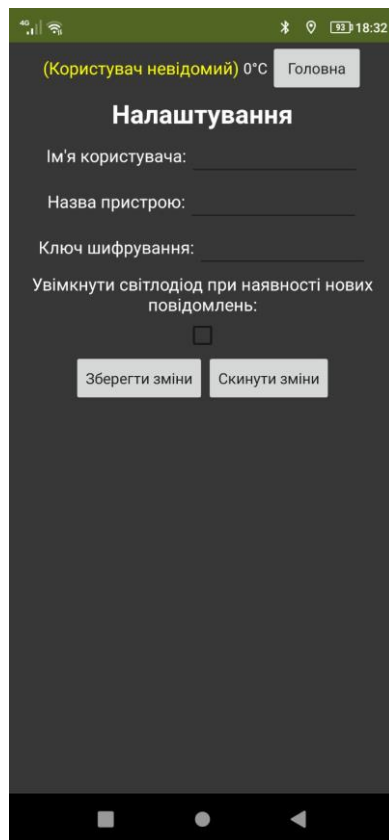


Рисунок 3.3 – Сторінка налаштувань Android-додатку

					<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

## 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

На основі описаних вище елементів мови для програмування MicroPython написано програмний код для управління мікроконтролером ESP32, лістинг якого має вигляд (повний код для ESP32 зображено у додатку Д).

Дана програма забезпечує роботу безпроводної системи передавання даних із використанням технології LoRa. Android-додаток, розроблений у онлайн середовищі розробки MIT App Inventor (рисунок 4.1) здійснює підключення за допомогою технології BLE до модуля, та здійснює відображення/передачу даних листування, а також дає змогу редагувати налаштування модуля, як користувач вважає за потрібне.

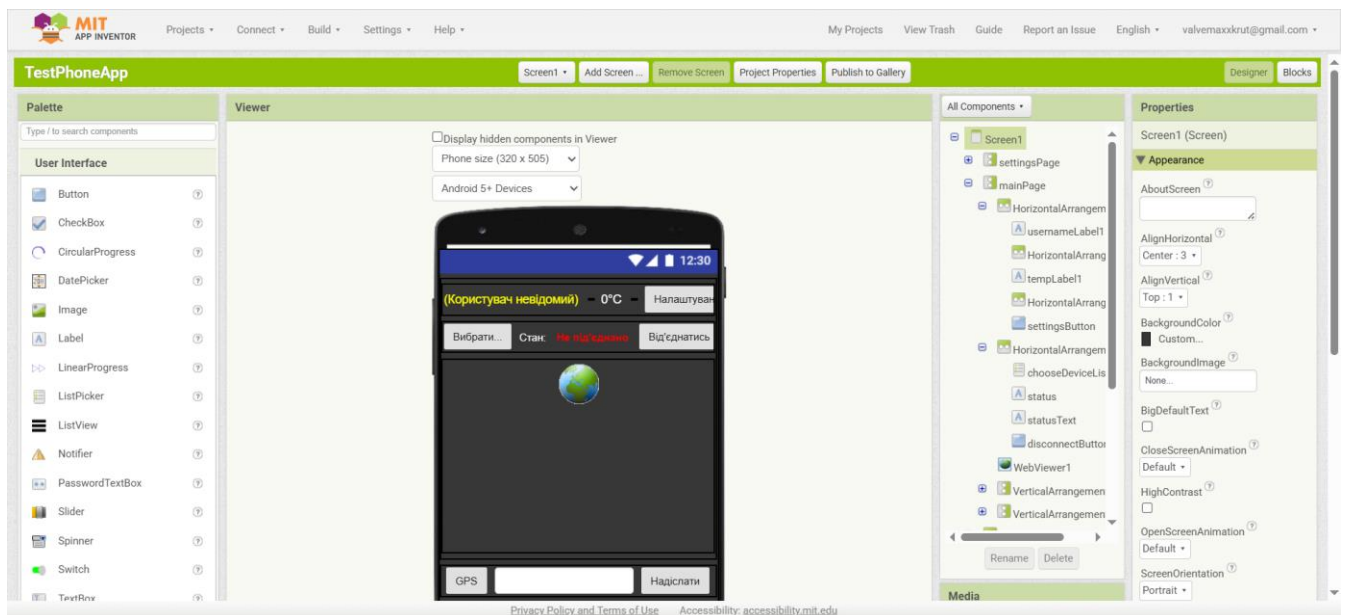


Рисунок 4.1 – Зображення головної сторінки розроблення Android-додатку у MIT App inventor [14]

Перший блок (рисунок 4.2) відповідає за саму поведінку модуля Bluetooth Low Energy. А саме ініціалізує унікальні ID призначені для активації каналу прийому/передачі даних BLE модуля; активує сканування доступних пристроїв у межах доступності, та добавляє ці пристрої у список готових пристроїв до підключення, при заході у програму; а для підключення необхідно обрати кнопку «Вибрати...», та вибрати власну системну назву BLE модуля пристрою; після

									Арк.
									39
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.СІ-12.00.000 ПЗ

цього Android-додаток здійснить підключення до пристрою, та активує канал для приймання даних, а також активує решту кнопок, та поле вводу тексту; якщо буде натиснуто кнопку «Від'єднатись», то буде здійснено розрив з'єднання, блокування всіх кнопок крім вибору пристрою для підключення, та розпочнеться нове сканування доступних пристроїв.

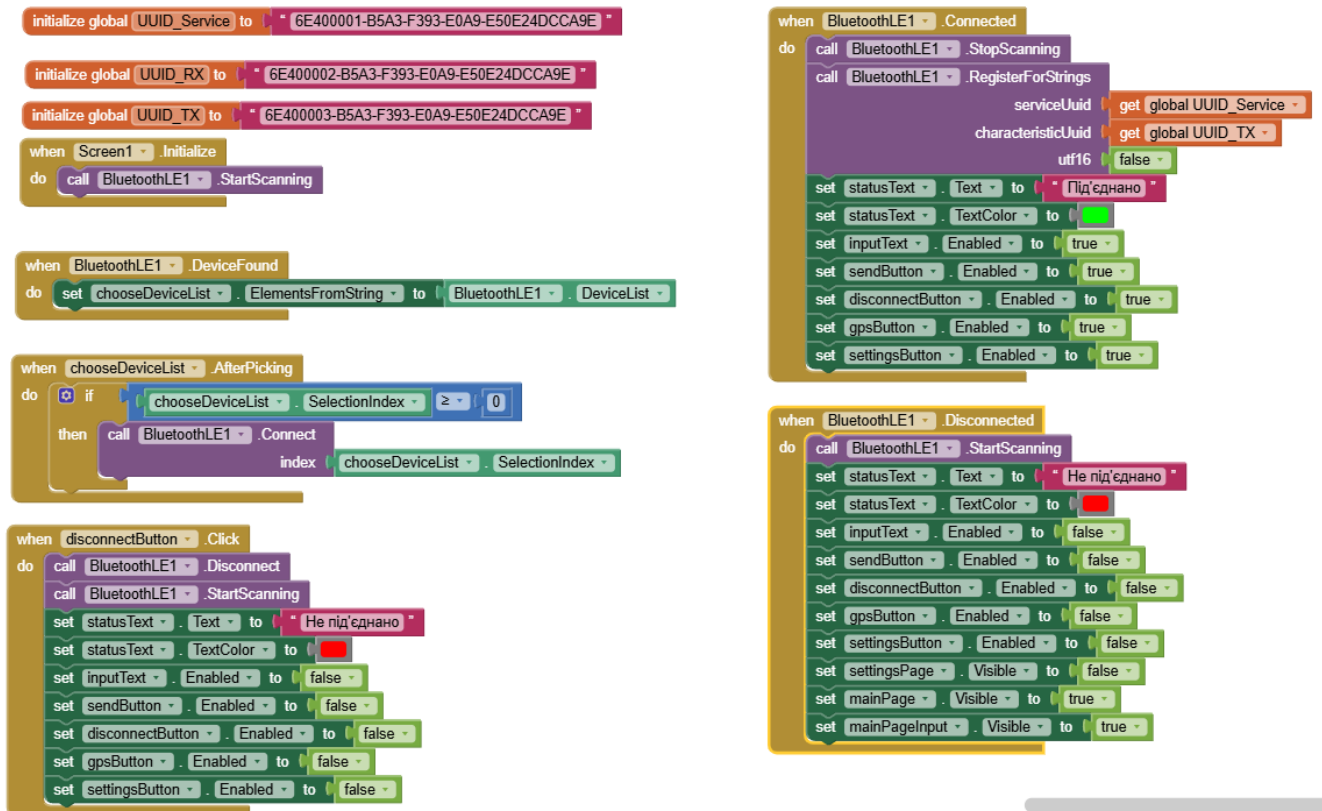


Рисунок 4.2 – Блок налаштування поведінки BLE модуля смартфона [14]

У другому блоці (рисунок 4.3) зображено алгоритм передачі текстового повідомлення, та повідомлення геолокації на пристрій безпроводної передачі даних. При натисканні кнопки «Надіслати», текст із поля введення передається у Java Script скрипт у внутрішньому веб-сервері для відображення введеного користувачем текстового повідомлення на внутрішній HTML сторінці, із використанням спеціальних CSS ефектів для зручної навігації додатком. Далі дане повідомлення пройде через блок кодування у формат Base64, для зручного надсилання через BLE через обмеження 20 байт на одне надсилання за раз, а потім текст у полі очиститься для введення нового повідомлення користувачем.

Для надсилання даних поточної геолокації користувачем, необхідно

						<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			40

натиснути кнопку GPS (Global Positioning System). Для цього були оголошенні глобальні змінні значення довготи, широти, та прапор оновлення даних. Саме тому із першого разу здійснити дану операцію не є можливим через відсутність самих даних місцезнаходження, та відсутності активації даного прапора. Для цього необхідно оновити дані місцезнаходження шляхом пересування самого смартфона, чи шляхом пошуку кращого сигналу GPS. Якщо дані місцезнаходження були успішно оновлені, буде активовано спеціальний прапор, який дозволить здійснити надсилання даних геолокації користувачем. Після цього координати користувача будуть надіслані на внутрішній сервер, для відображення у HTML сторінці, але із використанням зручної технології Google Maps для зрозумілого відображення місцезнаходження користувача. Далі координати будуть закодовані у формат Base64, та надіслані на пристрій безпроводної передачі даних.

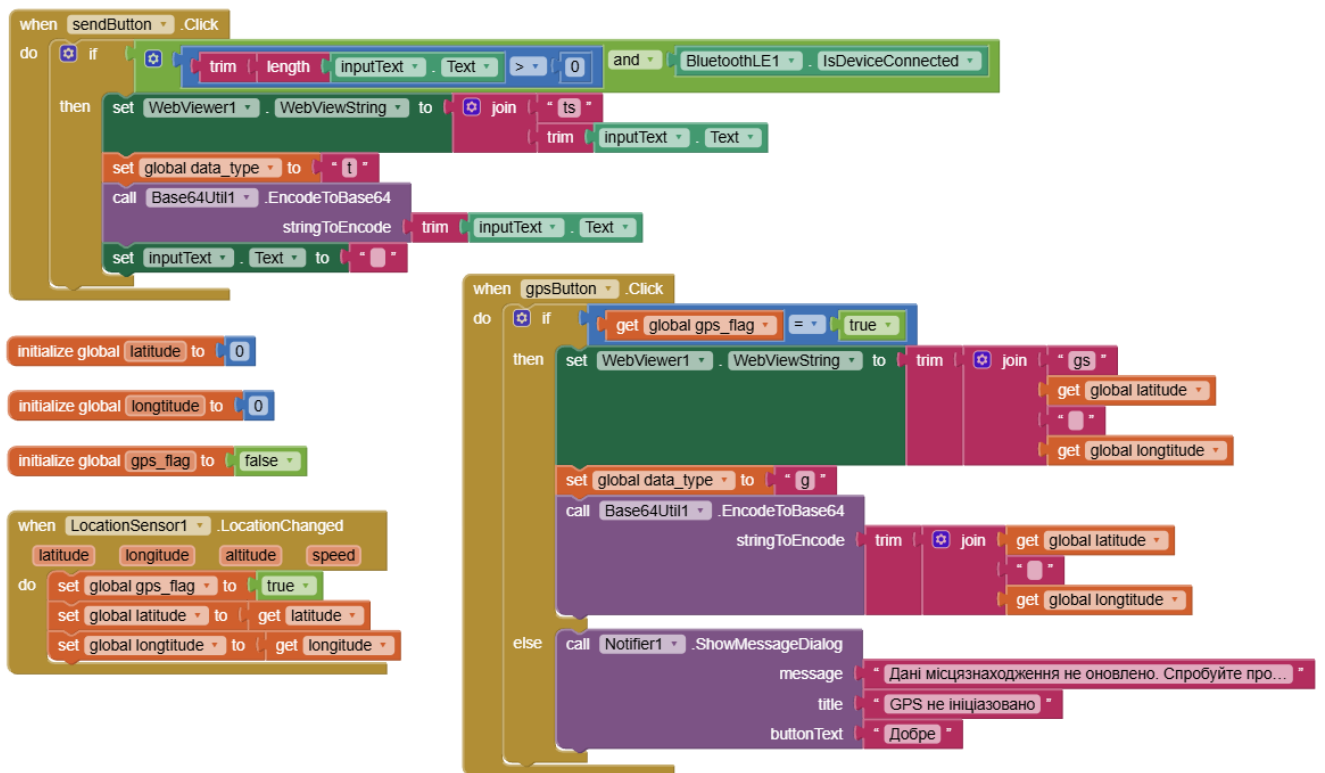


Рисунок 4.3 – Блок надсилання текстових даних, та даних геолокації [14]

У третьому блоці (рисунок 4.4) здійснюється кодування даних у формат Base64. Він дозволяє перетворити любий текстовий ряд у один довгий рядок латиницею, із цифрами, або спецсимволами, які зручно обробляти. Якщо довжина

закодованого рядка із вказаним типом даних, та зупинним байтом є менше 20-ти байт, тоді даний рядок надсилається моментально, у іншому випадку весь рядок Base64 перетворюється у текстовий масив (із вказаним типом даних, та зупинним байтом), та надсилається частинками по 20 байт кожна, поки рядок Base64 не буде повністю розділений на частинки по 20 байт.

Також у даному додатку активовано із самого початку його роботи активовано таймер на 50 мс, при активації якого здійснюється надсилання Base64 рядка через BLE, який розділений у масиві по частинках 20 байт, поки ніяких даних у масиві не залишиться. Для цього також активується канал для передачі даних із використанням відповідного унікального ID.

```

initialize global string to ""
initialize global data_type to ""
initialize global data_to_send to create empty list
initialize global splitted_list to create empty list

when Base64Util1.OnEncodeFinished
  result
  do if length join get global data_type <= 20
    then call BluetoothLE1.WriteString
         serviceUuid get global UUID_Service
         characteristicUuid get global UUID_RX
         utf16 false
         values trim join get global data_type
         get result
    else append to list list1 get global splitted_list
         list2 split text get result
         at 0
         insert list item list get global splitted_list
         index 1
         item get global data_type
         add items to list list get global splitted_list
         item
         for each item in list get global splitted_list

do if get global i == 18
  then set global string to join get global string
       get item
       add items to list list get global data_to_send
       item get global string
       set global i to 1
       set global string to ""
  else set global string to join get global string
       get item
       set global i to get global i + 1

if length get global string > 0
  then add items to list list get global data_to_send
       item get global string
       set global i to 1
       set global string to ""
       set global data_type to ""
       set global splitted_list to create empty list

when Clock1.Timer
  do if not is list empty? list get global data_to_send
    then call BluetoothLE1.WriteString
         serviceUuid get global UUID_Service
         characteristicUuid get global UUID_RX
         utf16 false
         values select list item list get global data_to_send
         index 1
         remove list item list get global data_to_send
         index 1
  
```

Рисунок 4.4 – Блок кодування та надсилання даних [14]

У четвертому блоці (рисунок 4.5) здійснюється прийом, та декодування даних із формату Base64. Для прийому даних було оголошено окремий глобальний масив, і якщо при отриманні даних було виявлено зупинний байт, значить передача даних була закінчена, і необхідно визначити тип отриманих



У п'ятому блоці (рисунок 4.6) здійснюється зміна налаштувань пристрою безпроводної передачі даних. Після натиснення користувачем кнопки «Налаштування», стандартне вікно відображення повідомлень пропаде, та буде відображатись вікно налаштувань. У даному вікні присутні 4 поля – ім'я користувача, назва BLE модуля пристрою безпроводної передачі даних, 16-ти байтовий ключ шифрування, та стан сигналізуючого синього світлодіода. Користувач може змінити їх улюбий момент використання Android-додатком, але тільки якщо він під'єднаний до пристрою безпроводної передачі даних.

Якщо дані були змінені, і користувач натиснув на кнопку «Зберегти зміни», то дані будуть закодовані у формат Base64, надіслані на пристрій безпроводної передачі даних через BLE, декодовані, записані у пам'ять пристрою, та вступають у силу в той же момент, про що повідомить вікно повідомлень Android-додатка. Також дані будуть моментально змінені у самому Android-додатку, як наприклад ім'я користувача.

Якщо користувач натиснув на кнопку «Скинути зміни», то дані у текстовому полі будуть замінені на стандартні, але ще не будуть надіслані на пристрій безпроводної передачі даних – тільки після підтвердження користувачем. Якщо користувач не виявить бажання здійснювати дану заміну, достатньо вийти із вікна налаштувань, та зайти знову, щоб побачити минулі оригінальні параметри.

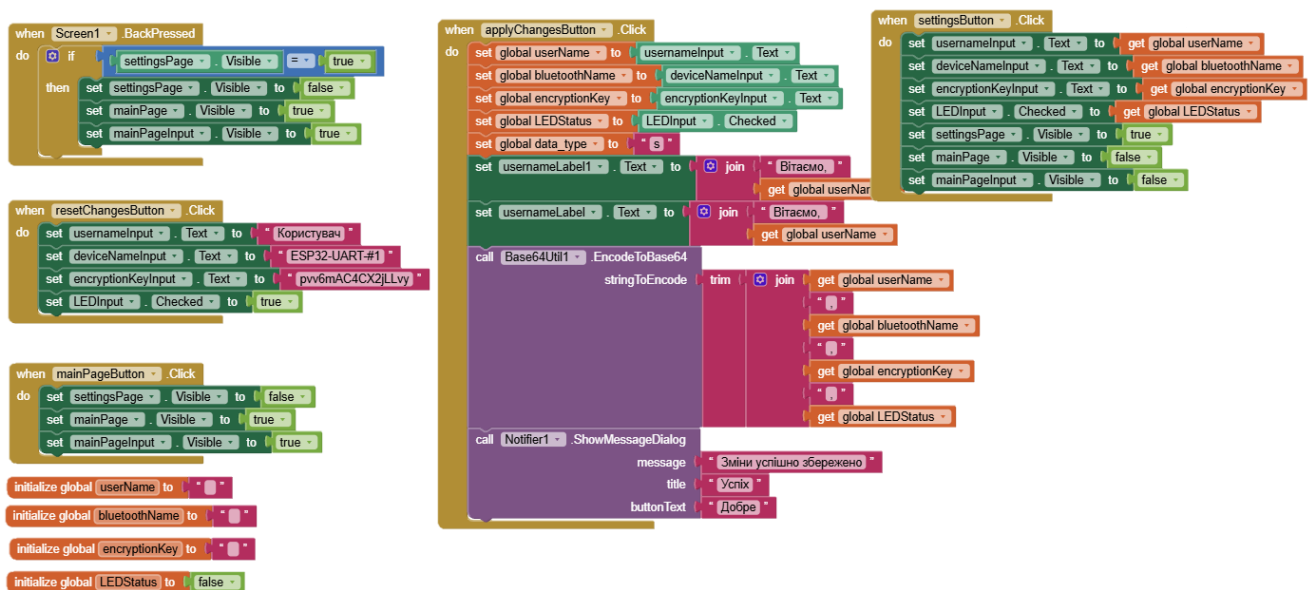


Рисунок 4.6 – Блок налаштувань [14]

						<i>КРБ.СІ-12.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			44

## ВИСНОВКИ

Бакалаврська робота на тему створення автоматизованої системи бездротової передачі даних із використанням технології LoRa, на базі мікроконтролера ESP32 у поєднанні з радіомодулем LoRa SX1278, підтвердила ефективність застосування сучасних бездротових технологій для побудови надійних, енергоефективних і масштабованих комунікаційних рішень.

На відміну від попередніх реалізацій, в оновленій версії системи використано мікроконтролер ESP32, що забезпечує більшу обчислювальну потужність, розширені можливості з'єднання (зокрема підтримку Bluetooth Low Energy) та кращу інтеграцію з мобільними пристроями. Наявність розробленого Android-додатку для надсилання та приймання повідомлень, а також налаштування роботи модуля через BLE зробила систему зручною та інтуїтивно зрозумілою для кінцевого користувача, значно розширивши сферу її застосування.

Додатковим елементом проекту став цифровий температурний датчик DS18B20, який дозволив реалізувати функції моніторингу технічного стану пристрою, яка дозволить спостерігати за станом роботи модуля, і дозволить запобігти його виведення із ладу завчасно.

Ключовими перевагами розробленої системи є: повна бездротова архітектура, низьке енергоспоживання, зручність взаємодії з користувачем через Android-додаток, а також гнучкість і масштабованість системи. У результаті проєкт продемонстрував високий потенціал для подальшого розвитку, зокрема — можливості розширення мережі, інтеграції з хмарними платформами, розширення типів передаваних даних (наприклад, зображення або файли), або розробки мережевих топологій для побудови розподілених систем.

Таким чином, дана бакалаврська робота засвідчила доцільність і перспективність використання мікроконтролера ESP32 у поєднанні з LoRa, BLE і мобільними застосунками для створення сучасних бездротових систем передавання інформації для автоматизації, моніторингу та зв'язку в різних сферах діяльності.

					<i>КРБ.СІ-12.00.000 ПЗ</i>	45
Змн.	Арк.	Недокум.	Підпис	Дата		

## Список використаних джерел

1. Трегуб В.Г. Основи комп'ютерно-інтегрованого управління: Навчальний посібник / В.Г. Трегуб. – К. : НУХТ, 2015. – 191 с.
2. Офіційний сайт Espressif [веб-ресурс] – URL: <https://www.espressif.com/>
3. Науково-технічна бібліотека [веб-ресурс] ІФНТУНГ – <http://library.nung.edu.ua>
4. Бездротові технології. Вікіпедія – вільна енциклопедія [веб-ресурс] – URL: [https://uk.wikipedia.org/wiki/Бездротові\\_технології](https://uk.wikipedia.org/wiki/Бездротові_технології)
5. Інтернет-портал «Arduino в Україні». Мікроконтролер ESP32 [веб-ресурс] – URL: <https://arduino.ua/prod7342-wi-fi-modul-esp32-wroom-32-4mb-type-c-30-pin>
6. Інтернет-портал «Arduino в Україні». Синій OLED дисплей SSD1306 [веб-ресурс] – URL: <https://arduino.ua/prod1264-oled-displei-modul-sinii>
7. Інтернет-портал «Arduino в Україні». LoRa радіомодуль SX1278 [веб-ресурс] – URL: <https://arduino.ua/prod3299-modem-lora-na-chipe-sx1278>
8. Інтернет-портал «Arduino в Україні». Цифровий давач температури DS18B20 [веб-ресурс] – URL: <https://arduino.ua/prod190-datchik-temperatury-ds18b20-cifrovoi>
9. Meshtastic – програмна платформа для створення малопотужних LoRa передавачів. Сторінка документації. Офіційний сайт [веб-ресурс] – URL: <https://meshtastic.org/docs>
10. Armachat – сумісний із Meshtastic LoRa передавач. Інтернет-портал для продажу DIY-розробок Tindie [веб-ресурс] – URL: <https://www.tindie.com/products/bobricius/armachat-meshtastic-compatible-lora-messenger/>
11. Пристрій H1. Інтернет-портал виробника Muzi-Works. Сторінка моделі H1 [веб-ресурс] – URL: <https://muzi.works/products/h1-complete-device-heltec-v3-running-meshtastic>

					<i>КРБ.СІ-12.00.000 ПЗ</i>	
Змн.	Арк.	Ледокум.	Підпис	Дата		46

12. Прошивка Micropython. Інтернет-портал розробників python ядра для мікроконтролерів. Сторінка документації ядра [веб-ресурс] – URL: <https://docs.micropython.org/en/latest/>
13. Середовище розробки Thonny. Офіційний сайт програми [веб-ресурс] – URL: <https://thonny.org/>
14. Онлайн середовище розробки Android-додатків MIT App Inventor. Офіційна сторінка середовища [веб-ресурс] – URL: <https://appinventor.mit.edu/>

					<i>КРБ.СІ-12.00.000 ПЗ</i>	
<i>Змн.</i>	<i>Арк.</i>	<i>Недокум.</i>	<i>Підпис</i>	<i>Дата</i>		47

## Додаток А

### Код програми для мікроконтролера ESP32

```
# Імпорт бібліотек для активації Bluetooth та для імпорту елементів ядра
Micropython

import bluetooth

from ble_advertising import advertising_payload

from micropython import const

import gc # Очишувач сміття
import time # Затримки
import json # Взаємодія із файлом JSON для збереження налаштувань пристрою
import sx127x # Бібліотека радіомодуля LoRa SX1278
import onewire # Бібліотека для зв'язку із давачем температури DS18B20 по одному
проводу
import _thread # Бібліотека для паралельних обчислень
import ssd1306 # Бібліотека для зв'язку із дисплеєм OLED 128x64
import ds18x20 # Бібліотека для зв'язку із давачем температури DS18B20
import network # Бібліотека для роботи із WIFI модулем (необхідно для його повного
вимкнення)
import cryptolib # Бібліотека для роботи із шифруванням повідомлень у стандарті AES
import ubinascii # Бібліотека для кодування повідомлення у форматі Base64
import config_lora # Додаткова бібліотека для взаємодія із радіомодулем LoRa SX1278
from math import ceil # Математичні функції
from machine import Pin, SoftI2C # Бібліотека для взаємодії із апаратною частиною
мікроконтролера

# Глобальні змінні
temp = 0 # Температура
config = None # Налаштування пристрою
response = "" # Зберігання відповіді із Android додатку через Bluetooth Low Energy
refresh_flag = True # Блимаючий баннер у дисплеї
connection_flag = False # Перевірка підключення користувача до пристрою
system_data_flag = True # Перевірка надсилання даних користувача до Android додатку

input_messages = [] # Повідомлення до надсилання
output_messages = [] # Повідомлення для відображення у Android додатку
```

## Продовження додатку Д

```
led = Pin(2, Pin.OUT, value = 0) # Вмонтований LED для відображення наявності
нового повідомлення

ds_pin = Pin(27) # Давач температури DS18B20
ds_sensor = ds18x20.DS18X20(onewire.OneWire(ds_pin))

i2c = SoftI2C(sda=Pin(21), scl=Pin(22)) # OLED дисплей 128x64
display = ssd1306.SSD1306_I2C(128, 64, i2c)

display.fill(0) # Обов'язкове очищення дисплею при запуску пристрою
display.show()

with open("config.json", "r") as f: # Зчитування записаних налаштувань
    config = json.load(f)

# Ініціалізація Bluetooth модуля
_IRQ_CENTRAL_CONNECT = const(1)
_IRQ_CENTRAL_DISCONNECT = const(2)
_IRQ_GATTS_WRITE = const(3)

_FLAG_WRITE = const(0x0008)
_FLAG_NOTIFY = const(0x0010)

_UART_UUID = bluetooth.UUID("6E400001-B5A3-F393-E0A9-E50E24DCCA9E")
_UART_TX = (
    bluetooth.UUID("6E400003-B5A3-F393-E0A9-E50E24DCCA9E"),
    _FLAG_NOTIFY,
)
_UART_RX = (
    bluetooth.UUID("6E400002-B5A3-F393-E0A9-E50E24DCCA9E"),
    _FLAG_WRITE,
)
_UART_SERVICE = (
    _UART_UUID,
    (_UART_TX, _UART_RX),
)
```

## Продовження додатку Д

```
_ADV_APPEARANCE_GENERIC_COMPUTER = const(128)

class BLEUART:
    def __init__(self, ble, name=config["bluetooth_name"], rxbuf=100):
        self._ble = ble
        self._ble.active(True)
        self._ble.irq(self._irq)

        ((self._tx_handle, self._rx_handle),) =
self._ble.gatts_register_services((_UART_SERVICE,))
        self._ble.gatts_set_buffer(self._rx_handle, rxbuf, True)
        self._connections = set()
        self._rx_buffer = bytearray()
        self._handler = None

        self._payload = advertising_payload(name=name,
appearance=_ADV_APPEARANCE_GENERIC_COMPUTER)
        self._advertise()

    def irq(self, handler):
        self._handler = handler

    def _irq(self, event, data):
        if event == _IRQ_CENTRAL_CONNECT: # При приєднанні пристрою до Anrroid
додатку - почати передавати повідомлення
            global connection_flag
            connection_flag = True

            conn_handle, _, _ = data
            self._connections.add(conn_handle)
            print("Connected")

        elif event == _IRQ_CENTRAL_DISCONNECT: # При від'єднанні пристрою до
Anrroid додатку - заборонити передавати повідомлення
            global connection_flag
            global system_data_flag
            connection_flag = False
            system_data_flag = True
```

## Продовження додатку Д

```
conn_handle, _, _ = data
if conn_handle in self._connections:
    self._connections.remove(conn_handle)
self._advertise()
print("Disconnected")

elif event == _IRQ_GATTS_WRITE:
    conn_handle, value_handle = data
    if conn_handle in self._connections and value_handle ==
self._rx_handle:
        self._rx_buffer += self._ble.gatts_read(self._rx_handle)
        if self._handler:
            self._handler()

def any(self):
    return len(self._rx_buffer)

def read(self, sz=None):
    if not sz:
        sz = len(self._rx_buffer)
    result = self._rx_buffer[0:sz]
    self._rx_buffer = self._rx_buffer[sz:]
    return result

def write(self, data):
    for conn_handle in self._connections:
        self._ble.gatts_notify(conn_handle, self._tx_handle, data)

def close(self):
    for conn_handle in self._connections:
        self._ble.gap_disconnect(conn_handle)
    self._connections.clear()

def _advertise(self, interval_us=500000):
    self._ble.gap_advertise(interval_us, adv_data=self._payload)
```

## Продовження додатку Д

```
# Вимкнення WIFI модуля
w = network.WLAN(network.STA_IF)
w.active(False)

# Вимкнення WIFI модуля
ap = network.WLAN(network.AP_IF)
ap.active(False)

# Ініціалізація конструкторів для взаємодії із Bluetooth модулем
ble = bluetooth.BLE()
uart = BLEUART(ble)
controller = config_lora.Controller()
lora = controller.add_transceiver(sx127x.SX127x(name = config["lora_name"]),
                                pin_id_ss =
config_lora.Controller.PIN_ID_FOR_LORA_SS,
                                pin_id_RxDone =
config_lora.Controller.PIN_ID_FOR_LORA_DIO0)
lora.setTxPower(17)

# Виведення тексту на екран
def screen_output():
    global refresh_flag

    while True:
        display.fill(0)
        refresh_flag = not refresh_flag

        if len(output_messages) > 0: # Якщо виявлено нове непрочитане повідомлення,
то активувати блимаючий баннер
            display.fill_rect(0, 0, 128, 9, int(refresh_flag))
            display.text("New message!", 21, 1, int(not refresh_flag))
        else: # У іншому випадку вивести вітальне повідомлення
            display.fill_rect(0, 0, 128, 9, 1)
            display.text("WELCOME!", 36, 1, 0)

    display.rect(0, 12, 128, 12, 1)
```

## Продовження додатку Д

```
display.text("BLE device name:", 1, 14, 1)

if len(config["bluetooth_name"]) < 16: # Якщо довжина назви BLE модуля
менше 16 символів, то вивести його повністю

    display.text(config["bluetooth_name"], 1, 27, 1)

    display.rect(0, 25, 128, 12, 1)

else: # Якщо довжина назви BLE модуля більше 16 символів, то вивести його
частинами, та обвести лініями

    y = 27

    i = 0

    slice_count = ceil(len(config["bluetooth_name"]) / 15)

    display.rect(0, 25, 128, (12*slice_count-1), 1)

    while slice_count > 0:

        display.text(config["bluetooth_name"][i:i+15], 1, y, 1)

        y += 12

        i += 15

        slice_count -= 1

display.show()

time.sleep(1)

def handle_available_messages():

    while True:

        if lora.receivedPacket(): # Якщо було прийнято повідомлення радіомодулем

            print("Отримано повідомлення!")

            try:

                payload = lora.read_payload()

                output_messages.append(decrypt_message(payload)) # Розшифрувати
його вміст, та додати до виведення

            except Exception as e:

                print(e)

            time.sleep_ms(50)

        if len(input_messages) > 0: # Якщо присутнє повідомлення до надсилання

            print("Надіслано повідомлення!")

            lora.println(encrypt_message(input_messages[0])) # Зашифрувати його
вміст, та надіслати
```

## Продовження додатку Д

```
input_messages.pop(0)

time.sleep_ms(50)

# Шифрування повідомлення із використанням стандарту AES, та спеціального 16-ти
байтового ключа
def encrypt_message(message):
    block_size = 16
    pad_length = block_size - (len(message) % block_size)
    padded_message = message + chr(pad_length) * pad_length

    aes_encrypt = cryptolib.aes(config["aes_128_key"], 1)
    ciphertext = aes_encrypt.encrypt(padded_message.encode())

    return ciphertext

# Розшифрування повідомлення із використанням стандарту AES, та спеціального 16-ти
байтового ключа
def decrypt_message(ciphertext):
    aes_decrypt = cryptolib.aes(config["aes_128_key"], 1)
    decrypted = aes_decrypt.decrypt(ciphertext)

    pad_length = decrypted[-1]
    decrypted_message = decrypted[:-pad_length].decode()

    return decrypted_message

# Якщо присутнє нове повідомлення для виводу, то починає мигати вмонтований LED
def new_message_led_blink():
    while True:
        while len(output_messages) > 0 and int(config["led_status"]):
            led.on()
            time.sleep(1)
            led.off()
            time.sleep(1)

        #print("LED is off")
        led.off()
```

## Продовження додатку Д

```
# Зчитування даних із датчика температури DS18B20
def temp_read():
    global temp
    roms = ds_sensor.scan()

    while True:
        ds_sensor.convert_temp()
        time.sleep_ms(750)

        for rom in roms:
            temp = str(int(ds_sensor.read_temp(rom)))

        time.sleep(2)

# Прийом даних із Android додатку через Bluetooth Low Energy
def on_rx():
    global response
    response += uart.read().decode().strip()

    if "|" in response: # Якщо виявлено кінцевий байт, то почати зчитування
повідомлення
        response_type = str(response[0], "utf-8")

        if response_type == "s": # Якщо це рядок із налаштуваннями, то перезаписати
ix
            new_settings = str(ubinascii.a2b_base64(response[1:len(response)-1]),
"utf-8").split(",")

            config["user_name"] = new_settings[0] # Ім'я користувача
            config["bluetooth_name"] = new_settings[1] # Назва BLE модуля
            config["aes_128_key"] = new_settings[2] # Ключ шифрування

            if new_settings[3] == "true": # Стан вмонтованого LED при наявності
повідомлення для виведення
                config["led_status"] = "1"
            else:
```

## Продовження додатку Д

```
config["led_status"] = "0"
```

```
with open("config.json", "w") as f: # Запис даних у файл config.json
    json.dump(config, f)
```

```
else: # У іншому випадку - на надсилання повідомлення через радіомодуль
    input_messages.append(response)
```

```
response_type = ""
```

```
response = ""
```

```
# Надсилання даних до Android додатку через Bluetooth Low Energy
```

```
def ble_send(message, message_type):
```

```
    e_data = ubinascii.b2a_base64(message) # Перетворення повідомлення у Base64
    формат
```

```
    e_data = message_type + e_data.decode("utf-8") + "|" # Додавання типу
    повідомлення та кінцевого байту
```

```
"""
```

```
Типи повідомлень:
```

```
t - текст
```

```
g - місцезоташування
```

```
d - температура
```

```
s - налаштування
```

```
"""
```

```
if len(e_data) < 20: # Якщо довжина повідомлення менше 20 байт - надсилати
цілком за раз
```

```
    uart.write(e_data)
```

```
    time.sleep_ms(50)
```

```
else: # Якщо довжина повідомлення більше 20 байт - надсилати повідомлення
чанками до 20 байт
```

```
for x in range(0, len(e_data), 18):
```

```
    uart.write(e_data[x:x+18])
```

```
    time.sleep_ms(50)
```

```
# Присвоєння функції для приймання повідомлень із Android додатку
```

## Продовження додатку Д

```
uart.irq(handler=on_rx)

# Активація функцій, які будуть виконуватись паралельно основній функції
_thread.start_new_thread(handle_available_messages, ())
_thread.start_new_thread(new_message_led_blink, ())
_thread.start_new_thread(screen_output, ())
_thread.start_new_thread(temp_read, ())

try:
    while True:
        if not connection_flag: # Якщо користувач не приєднався до пристрою - не
робити нічого
            continue

        if system_data_flag: # Якщо користувач приєднався після запуску Android
додатку - вивести його ім'я, та дані у вікно "Налаштування"
            time.sleep(1)

            ble_send(config["user_name"] + "," + config["bluetooth_name"] + "," +
config["aes_128_key"] + "," + config["led_status"], "s")

            ble_send(config["user_name"], "u")

            system_data_flag = False

        if len(output_messages) > 0: # Якщо присутні повідомлення для виводу
            raw_data = str(output_messages[0], "utf-8") # Перетворення тексту у
формат Base64, та надсилання даного повідомлення у Anrdoid додаток для вивлення
            message = str(ubinascii.a2b_base64(raw_data[1:len(raw_data)-1]), "utf-
8")

            message_type = raw_data[0]
            ble_send(message, message_type)
            output_messages.pop(0)

            message_type = ""
            raw_data = ""
            message = ""

        # Виведення температури у Android додаток
        ble_send(temp, "d")
```

## Продовження додатку Д

```
# Очистка від мусору
gc.collect()

except KeyboardInterrupt: # Якщо відбувся вихід із програми - закрити з'єднання
    pass

uart.close()
```

## **БІБЛОГРАФІЧНА ДОВІДКА**

Тема бакалаврської роботи – «Розроблення безпроводної системи передавання даних із використанням технології LoRa».

Обсяг пояснювальної записки в аркушах – 63.

### **Перелік графічного матеріалу:**

КРБ.СІ-12.00.000 Е1 – Безпроводна система передавання даних із використанням технології LoRa. Схема електрична структурна (аркушів – 1).

КРБ.СІ-12.00.000 Е3 – Безпроводна система передавання даних із використанням технології LoRa. Схема електрична принципова (аркушів – 1).

КРБ.СІ-12.00.000 ПЕ3 – Безпроводна система передавання даних із використанням технології LoRa. Перелік елементів (аркушів – 1).

КРБ.СІ-12.00.000 БС – Безпроводна система передавання даних із використанням технології LoRa. Блок схема (аркушів – 1).

Дата закінчення бакалаврської роботи «20» червня 2025 р.

Студент \_\_\_\_\_ Ваврикович М.П.