

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Інститут автоматизації та енергетики
Кафедра інформаційно-вимірювальних технологій та енергетичного менеджменту

Пташник Андрій Іванович

(прізвище, ім'я, по батькові)

УДК 636.5:65.011
(індекс)

МАГІСТЕРСЬКА РОБОТА

Удосконалення автоматизованої системи для реєстрації та подання старт-сигналу з
веб-інтерфейсом

(назва роботи)

Метрологія та вимірювальна техніка

(назва освітньої програми)

175 – Інформаційно-вимірювальні технології

(шифр і назва спеціальності)

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

Здобувач освітнього ступеня Пташник А.І.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Лютак З.П.
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

проф. Цих В.С.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

(повне найменування закладу вищої освіти)

Факультет автоматизації та енергетики

Кафедра інформаційно-вимірювальних технологій та енергетичного менеджменту

Освітній рівень магістр

Спеціальність 175 – Інформаційно-вимірювальні технології

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІВТТМ

Цих В.С.

« ____ » _____ 2025 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТОВІ

Пташнику Андрію Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи Удосконалення автоматизованої системи для реєстрації та подання старт-сигналу з веб-інтерфейсом

керівник роботи Лютак З.П., кан. тех. н.,

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

затверджені наказом закладу вищої освіти від “ 5 ” грудня 2025 року № 752/7

2. Строк подання студентом роботи 26 грудня 2025 р.

3. Вихідні дані до роботи досліджуваний об'єкт – автоматизована система для реєстрації та подання старт-сигналу з веб інтерфейсом, вид керування – автоматична ідентифікація. вимоги до точності детектування та швидкодії системи, вибір апаратної платформи та протоколів бездротової передачі даних.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної області

2. Конструкторська частина

3. Програмна частина

4. Розробка програмного забезпечення

5. Експериментальна частина

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

МР.МТТм – 11.00.00.000. Е1 А Схема електрична структурна

МР.МТТм – 11.00.00.000. Е2 Б Схема електрична принципова

МР.МТТм – 11.00.00.000. БС Г Блок-схема роботи програмної частини

МР.МТТм – 11.00.00.000. ПК Д Програмний код прошивки мікроконтролера

МР.МТТм – 11.00.00.000. ПЕ В1 Перелік елементів

МР.МТТм – 11.00.00.000. ПЕ В2 Перелік елементів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Чуйко М.М. к.т.н.		

7. Дата видачі завдання 6 жовтня 2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Примітка
1	Вступ. Підбір необхідної літератури	до 30.10	викон.
2	Вибір програмних пакетів для реалізації системи	до 15.11	викон.
3	Порівняльний аналіз існуючих систем	до 30.11	викон.
4	Розроблення архітектури комплексу	до 01.12	викон.
5	Програмна реалізація методів цифрової фільтрації та мінімізації впливу дестабілізуючих факторів	до 15.12	викон.
6	Висновки	до 19.12	викон.

Студент _____ (підпис) Пташник А.І. (прізвище та ініціали)
 Керівник роботи _____ (підпис) Лютак З.П. (прізвище та ініціал)

РЕФЕРАТ

Пояснювальна записка містить 60 с., 25 рис., 2 табл., 12 джерел та 6 листів графічного матеріалу.

Темою бакалаврської роботи є «Удосконалення автоматизованої системи для реєстрації та подання старт-сигналу з веб-інтерфейсом».

Метою роботи є розробка та вдосконалення технічного засобу для автоматизованої системи реєстрації та подання сигналу старт для різних галузей роботи за допомогою радіочастотної ідентифікації (RFID) з веб-інтерфейсом .

У магістерській роботі проаналізовано роль та місце автоматизованих систем у сучасній складській логістиці. Обґрунтовано вибір технології RFID як найбільш надійного методу фіксації часу, що дозволяє усунути суб'єктивну похибку та забезпечити метрологічну простежуваність

Розроблено апаратну частину системи на базі мікроконтролера NodeMCU ESP8266 V3 та зчитувача RC522, спроектовано схему живлення з каскадною стабілізацією для автономної роботи та використано зовнішні антени для підвищення стабільності зв'язку

Створено програмне забезпечення в середовищі Arduino IDE з використанням архітектури «Кінцевий автомат» (State Machine), що забезпечує детермінованість роботи та надійну транзакцію даних на сервер через протокол HTTP.

Проведено експериментальні дослідження, за результатами яких визначено максимальну ефективну дальність зчитування (27 мм), середню затримку передачі даних (120–130 мс) та розраховано сумарну метрологічну похибку системи, яка становить 1,8%.

Ключові слова: СИСТЕМИ ДЛЯ РЕЄСТРАЦІЇ ТА ПОДАННЯ СТАРТ-СИГНАЛУ ВЕБ-ІНТЕРФЕЙС, ESP8266, IoT, МЕТРОЛОГІЧНА ПОХИБКА, HTTP-ПРОТОКОЛ

ABSTRACT

The explanatory note contains 60 pages, 25 figures, 2 tables, 12 sources, and 6 sheets of graphic material.

The topic of the master's thesis is "Improvement of an automated system for registration and start-signal delivery with a web interface."

The aim of the work is the development and improvement of a technical tool for an automated system of registration and start-signal delivery for various fields of work using Radio Frequency Identification (RFID) with a web interface.

The master's thesis analyzes the role and place of automated systems in modern warehouse logistics. The choice of RFID technology is justified as the most reliable method for time recording, which allows eliminating subjective error and ensuring metrological traceability.

The hardware part of the system was developed based on the NodeMCU ESP8266 V3 microcontroller and RC522 reader; a power supply circuit with cascade stabilization was designed for autonomous operation, and external antennas were used to increase connection stability.

Software was created in the Arduino IDE environment using the "Finite State Machine" architecture, which ensures the determinism of operation and reliable data transaction to the server via the HTTP protocol.

Experimental studies were conducted, the results of which determined the maximum effective reading range (27 mm), the average data transmission delay (120–130 ms), and calculated the total metrological error of the system, which is 1.8%.

Keywords: SYSTEMS FOR REGISTRATION AND START-SIGNAL DELIVERY, WEB INTERFACE, ESP8266, IoT, METROLOGICAL ERROR, HTTP PROTOCOL.

ЗМІСТ

ЗМІСТ.....	5
ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕННЯ.....	6
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Технічні засоби для ідентифікування об’єктів RFID мітки.....	12
1.2 Схеми для керування візуальними інформаційними пристроями	16
1.3 Технології для розробки графічного інтерфейсу користувача.....	18
1.4 Принципи комунікації складових в розроблювальній системі	21
1.5 Застосування технології RFID у метрологічному забезпеченні.....	24
1.6 Аналіз сучасних технічних засобів автоматизації реєстрації та подання старт-сигналу	26
2 КОНСТРУКТОРСЬКА ЧАСТИНА	28
2.1 Розробка структурної схеми	28
2.2 Розробка електричної принципової схеми	29
2.3 Обґрунтування вибору платформи NodeMCU	31
2.4 Додаткові модулі.....	34
2.5 Аналіз організації живлення та інтерфейсів взаємодії.....	37
2.6 Конструктивні особливості забезпечення точності реєстрації подій..	39
2.7 Метрологічний аналіз вимірювального каналу системи	40
3 ПРОГРАМНА ЧАСТИНА.....	42
3.1 Розробка алгоритму роботи пристрою	42
3.2 Реалізація логіки перемикання режимів роботи (State Machine)	43
3.3 Програмна реалізація протоколу обміну даними	44
3.4 Синхронізація процесів та керування апаратними ресурсами	45

3.5 Алгоритмічні методи компенсації динамічних похибок	46
4 РОЗРОБКА ПРОГРАМАНОВОГО ЗАБЕЗПЕЧЕННЯ	48
4.1 Обґрунтування вибору інтегрованого середовища розробки	48
4.2 Огляд використаних бібліотек та залежностей.....	48
4.3 Програмний код прошивки мікроконтролера	49
4.4 Програмні засоби верифікації та валідації вимірювань.....	50
5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	51
5.1 Технічні характеристики системи для реєстрації учасників змагань та подавання сигналу старт-сигналу.....	51
5.2 Інструкція з експлуатації системи для реєстрації учасників змагань .	51
5.3 Визначення відстанні зчитування RF мітки	53
5.4 Експериментальне дослідження мережевих затримок передачі даних	57
ВИСНОВКИ.....	60
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕННЯ

- NFC (Near Field Communication) – технологія бездротового зв'язку близької дії.
- RFID (Radio Frequency Identification) – радіочастотна ідентифікація об'єктів.
- IP (Internet Protocol) – міжмережевий протокол, що відповідає за адресацію.
- TCP (Transmission Control Protocol) – протокол керування передачею, що гарантує доставку даних.
- Wi-Fi (Wireless Fidelity) – стандарт бездротового зв'язку для обміну даними.
- ADC / АЦП (Analog-to-Digital Converter) – аналого-цифровий перетворювач.
- GPIO (General Purpose Input/Output) – порти введення-виведення загального призначення.
- I2C (Inter-Integrated Circuit) – послідовний інтерфейс для комутації (використовується дисплеєм TM1637).
- LED (Light-Emitting Diode) – світлодіод.
- MISO (Master In Slave Out) – лінія передачі даних від веденого пристрою до ведучого (SPI).
- MOSI (Master Out Slave In) – лінія передачі даних від ведучого пристрою до веденого (SPI).
- SCK (Serial Clock) – лінія тактування інтерфейсу SPI.
- SDA (Serial Data) – лінія послідовних даних.
- SoC (System-on-a-Chip) – система на кристалі (наприклад, ESP8266).
- SPI (Serial Peripheral Interface) – послідовний периферійний інтерфейс.
- UART (Universal Asynchronous Receiver-Transmitter) – універсальний асинхронний прийомопередавач.

- IoC (Inversion of Control) – принцип інверсії керування у фреймворку Spring.
- IoT (Internet of Things) – Інтернет речей.
- JSON (JavaScript Object Notation) – текстовий формат обміну даними.
- ЗВТ – засоби вимірювальної техніки.

ВСТУП

Рівень життя сучасної людини та економічний розвиток суспільства значною мірою визначаються якістю продукції та ефективністю технологічних процесів. Фундаментом будь-якого виробництва, окрім технічних ресурсів та енергії, є достовірна інформація про властивості об'єктів та процесів. Отримання такої інформації неможливе без метрології — науки про вимірювання та їх застосування.

Науково-технічна революція, що забезпечила сучасний добробут, базується не лише на механізації праці, а й на стрімкому зростанні точності виготовлення деталей. Перехід від ручної праці до машинної вимагав суворого контролю параметрів, адже будь-який механізм — від парової машини до двигуна внутрішнього згоряння — працює ефективно лише за умови точного дотримання допусків та посадок.

Проте жоден, навіть найдосконаліший верстат чи роботизований комплекс, не зможе виготовити сучасну електроніку, прецизійні деталі для авіації чи медичне обладнання без постійного вимірювального контролю. Органи чуття людини (око як приймач інформації, мозок як аналізатор) є надто суб'єктивними, повільними та недостатньо точними для сучасних нановиробництв. Тому другим стовпом промислового прогресу, поряд з автоматизацією, став розвиток інформаційно-вимірювальних систем.

Сучасне виробництво — це синтез технологій та вимірювань. Автоматизація неможлива без зворотного зв'язку, який забезпечують первинні вимірювальні перетворювачі (сенсори). Саме вони надають системам керування

об'єктивні дані про стан процесу. Без точних вимірювань "розумна" машина стає "сліпою" та некерованою.

Нині засоби вимірювальної техніки (ЗВТ) інтегровані у всі сфери: від важкої промисловості до побуту. Робота будь-якого сучасного пристрою базується на обробці вимірювальних сигналів. Це вимагає від фахівців глибоких знань у сфері комп'ютерно-інтегрованих вимірювальних систем, де функції збору, обробки та передачі інформації виконують мікропроцесорні засоби.

Головною метою таких систем є забезпечення єдності та необхідної точності вимірювань. Вони гарантують достовірність даних, на основі яких приймаються рішення щодо якості продукції, безпеки процесів та екологічного моніторингу. Інтелектуальні вимірювальні системи здатні самостійно діагностувати свій стан, проводити самокалібрування та архівувати масиви даних.

Концепція "Розумного будинку" (Smart Home) є яскравим прикладом побутового застосування інформаційно-вимірювальних технологій. Це, по суті, розгалужена мережа датчиків (температури, руху, освітленості, складу повітря), об'єднаних в єдину систему моніторингу. Центр управління аналізує виміряні величини та, згідно з алгоритмом, формує команди виконавчим механізмам, забезпечуючи комфорт та енергоефективність.

В результаті курсового проєктування згідно з освітньо-професійною програмою «Метрологія та інформаційно-вимірювальна техніка» здобувач освіти набуває таких фахових компетентностей:

- здатність застосовувати знання фізики, схемотехніки та мікропроцесорної техніки для розуміння принципів перетворення фізичних

величин у вимірювальні сигнали в сучасних інформаційно-вимірювальних системах;

- здатність обґрунтовувати вибір засобів вимірювальної техніки на основі аналізу їх метрологічних характеристик (клас точності, діапазон вимірювання, чутливість) з урахуванням умов експлуатації та вимог до точності технологічного процесу;

- здатність використовувати новітні технології для вирішення завдань вимірювального контролю, зокрема: розробка алгоритмів обробки результатів вимірювань, оцінювання невизначеності вимірювань, калібрування та перевірка приладів;

- здатність проєктувати структуру вимірювальних каналів та розробляти програмне забезпечення для мікропроцесорних засобів збору даних, інтелектуальних сенсорів та віртуальних приладів;

- здатність вільно користуватись сучасними інформаційними технологіями для візуалізації вимірювальної інформації, створення баз даних параметрів якості та роботи у спеціалізованих середовищах (наприклад, LabVIEW) для моделювання процесів вимірювання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Технічні засоби для ідентифікування об'єктів RFID мітки.

Технологія RFID (Radio Frequency Identification - радіочастотна ідентифікація) - це технологія нового покоління, заснована на використанні радіочастотного електромагнітного випромінювання. Технологія застосовується для ідентифікації та обліку об'єктів.

RFID - це сучасна технологія ідентифікації, що надає значно більше можливостей у порівнянні з традиційними системами маркування.

RFID - мітка являє собою мініатюрне запам'ятовує. Вона складається з мікročіпа, який зберігає інформацію, і антени, за допомогою якої мітка ці дані передає і отримує. Іноді RFID-мітка має власне джерело живлення (такі мітки називають активними), але більшість міток його позбавлені (ці мітки називають пасивними). Зображена на рисунку 1.1.

У пам'яті RFID-мітки зберігається її власний унікальний номер і призначена для користувача інформація. Коли мітка потрапляє в зону реєстрації, така інформація була розглянута зчитувачем, спеціальним приладом, здатним читати і записувати інформацію в мітках.

Технологія RFID затребувана в багатьох областях. Для того, щоб системи, засновані на технології RFID ефективно працювали в будь-якому середовищі, було розроблено безліч RFID-міток самого різного виконання. Їх умовно можна розділити за такими ознаками:

1. За типом живлення:

Активні - використовують для передачі даних енергію вбудованого елемента живлення (зона читання до 100 метрів);

Пасивні - використовують енергію, що випромінюється зчитувачем (дальність до 8 метрів).

2. За видами пам'яті:

“RO” (Read Only) - дані записуються тільки один раз, відразу при виготовленні. Такі мітки придатні тільки для інформації. Ніяку нову інформацію в них записати не можна, і їх практично неможливо підробити;

“WORM” (Write Once Read Many) - крім унікального ідентифікатора такі мітки містять блок одноразово записуваної пам'яті, яку в подальшому можна багаторазово читати;

“RW” (Read and Write) - такі мітки містять ідентифікатор і блок пам'яті для читання / запису інформації. Дані в них можуть бути перезаписані велике число раз.

3. За виконання (визначається цілями і умовами використання RFID – міток);

- самоклеючі паперові або лавсанові мітки;
- стандартні пластикові карти;
- дискові мітки (в тому числі з центральним отвором для закріплення на палеті);
- різні види брелоків;
- спеціальне виконання для жорстких умов експлуатації.



Рисунок 1.1 – RFID мітка

NFC (дослівний переклад «зв'язок близької дії») - технологія, за допомогою якої пристрої можуть обмінюватися веб-адресами, текстом або числами.

NFC-мітка являє собою компактну антену товщиною з паперовий лист, яка працює в пасивному режимі і не вимагає енергії. Один з її головних переваг - маленький розмір. Ви можете з легкістю закріпити виріб навіть на брелоку. Основний компонент NFC-стікера – котушка-провідник, що виробляє магнітні поля.

Щоб чіп працював, він повинен потрапити в певний радіус дії. Самі по собі мітки не здатні обмінюватися інформацією, для цього потрібен смартфон, планшет або «розумні» годинник. Працездатність пристрою забезпечується електромагнітною індукцією.

У середньому обмін даними відбувається за 0,1 с. Для цього до наклейці прикладіть смартфон так, щоб повітряний проміжок між ними був до 5 див.

Для активації технології потрібно «активний» девайс – телефон і «пасивний» – стікер. Перший відповідає за магнітне поле, яке формується за рахунок котушки з дротом. Останній виробляє коливання під прямим кутом потоку змінного струму у проводі.

Сила магнітного поля коригується зміною кількості витків на котушках або ж збільшенням сили струму. Такий процес потребує серйозної енергетичної підживлення, а застосування компонентів, які споживають багато енергії, небажано. Це головна причина, чому NFC діє лише на відстані кількох сантиметрів, на відміну від Bluetooth і Wi-Fi.

«Пасивні» пристрої працюють за аналогічним алгоритмом, тільки в зворотному порядку: як тільки наклейка опиниться в магнітному полі, в приймальній котушці з'явиться струм. Завжди є електричні втрати, але з-за маленького відстані між пристроями вони практично не впливають на результат.

Схеми запрограмовані на певну частоту. Це підвищує чутливість до сигналів, що в свою чергу прискорює передачу даних.

NFC-Tag створений на основі міжнародних стандартів ISO 14443 типу A і B, що робить його практично ідентичним смарт-карт. Зображений на рисунку 1.2

Стікери типу 1 і 2 мають скромні 48 байтами і 2 кілобайтами даних, здатні передавати їх на швидкості 106 Кбіт/с. Так, це дійсно дуже мало, але для використання NFC цілком достатньо. Мітки розроблені з урахуванням багаторазового використання та зміни даних.

Третій тип етикетки базується на стандарті Sony Felica і передає дані зі швидкістю 212 Кбіт/с. Він ідеально підходить для більш складних завдань, ніж просто збереження в пам'яті адреси сайту.

У міток 4-го типу хороший обсяг пам'яті – 32 Кбайт, а інформація передається зі швидкістю від 104 Кбіт/с до 420 Кбіт/с.

Відмітна особливість стікерів 3-го і 4-го типів – неможливість перезапису.

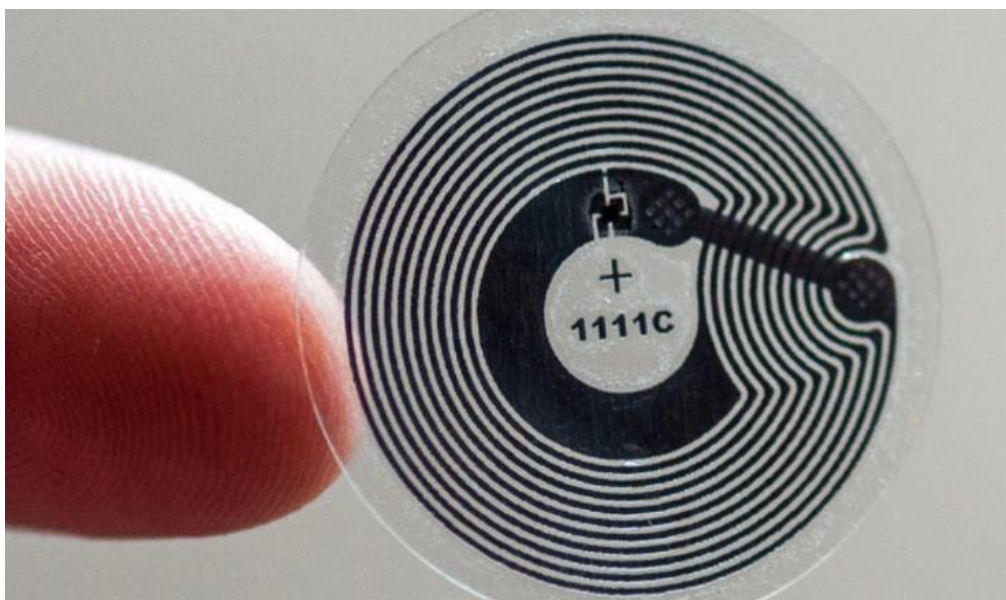


Рисунок 1.2 – NFC – мітка

1.2 Схеми для керування візуальними інформаційними пристроями

Arduino — це поширена апаратно-програмна екосистема, призначена для швидкого прототипування електронних пристроїв: від найпростішої автоматики до складних робототехнічних систем та елементів «Розумного будинку». Популярність платформи зумовлена її доступністю та низьким порогом входження: розробка власних пристроїв не потребує глибоких знань фундаментальної фізики чи складних мов програмування. Це дозволяє за короткий час реалізувати прикладні проекти, такі як метеостанції, системи автополиву або дистанційно керовані роботи.

Одним із класичних прикладів для вивчення основ схемотехніки є реалізація макета світлофора. Світлофор — це ключовий елемент дорожньої інфраструктури, що забезпечує безпеку руху (рис. 1.3). Історично перший газовий світлофор був встановлений у Лондоні в 1868 році, а його електричний аналог з'явився у США в 1912 році завдяки інженеру Лестеру Вайру. Стандартизована послідовність кольорів (червоний, жовтий, зелений) та їх фіксоване розташування обумовлені необхідністю забезпечення видимості та розпізнавання сигналів людьми з порушеннями кольоросприйняття, які орієнтуються за позицією сигналу, а не за його кольором. У сучасних світлофорних об'єктах використовуються надійні та енергоефективні LED-матриці, що дозволяють виводити додаткову графічну інформацію та таймери зворотного відліку.



Рисунок 1.3 – Світлофор

Схема реалізації світлофора на базі Arduino передбачає підключення трьох світлодіодів. Катоди (мінусові контакти) світлодіодів об'єднуються спільною шиною та підключаються до заземлення (GND), а аноди (плюсові контакти) комутуються з цифровими портами мікроконтролера. Наприклад, червоний світлодіод підключається до піна 11, жовтий — до 10, а зелений — до 9 (рис. 1.4).

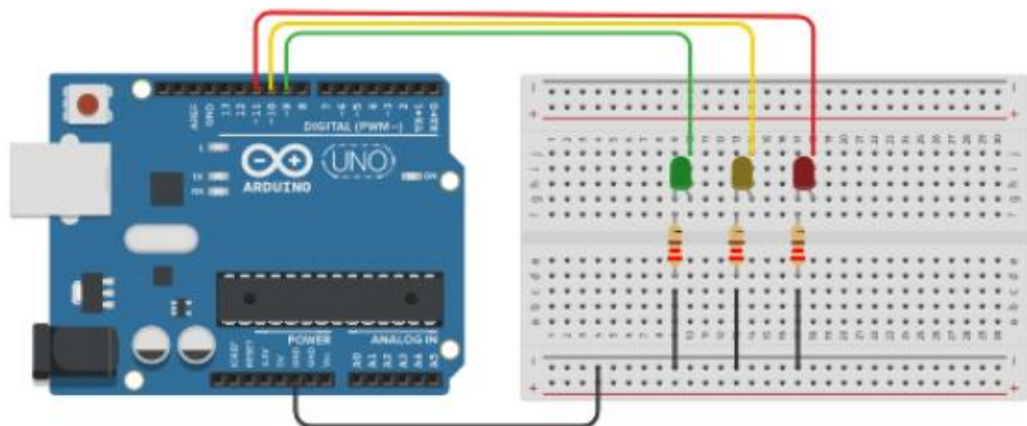


Рисунок 1.4 – Світлофор на Arduino

Для реалізації проєктів у сфері Інтернету речей (IoT) широко використовується платформа NodeMCU на базі чіпа ESP8266. Це рішення поєднує можливості мікроконтролера, аналогічного Arduino, з вбудованим модулем Wi-Fi. Технології IoT дозволяють організувати віддалене керування пристроями через мережу Інтернет. Однак використання стандарту Wi-Fi (особливо в діапазоні 2.4 ГГц) пов'язане з проблемою обмеженого радіуса дії та наявністю електромагнітних завад від іншого обладнання.

Для вирішення проблеми стабільності зв'язку та збільшення зони покриття доцільно використовувати зовнішні антени. Підключення здійснюється через спеціалізовані високочастотні роз'єми IPEX (MHF2), що зображено на рисунку 1.5. Таке технічне рішення дозволяє значно підвищити коефіцієнт підсилення сигналу та загальну ефективність роботи бездротового модуля.

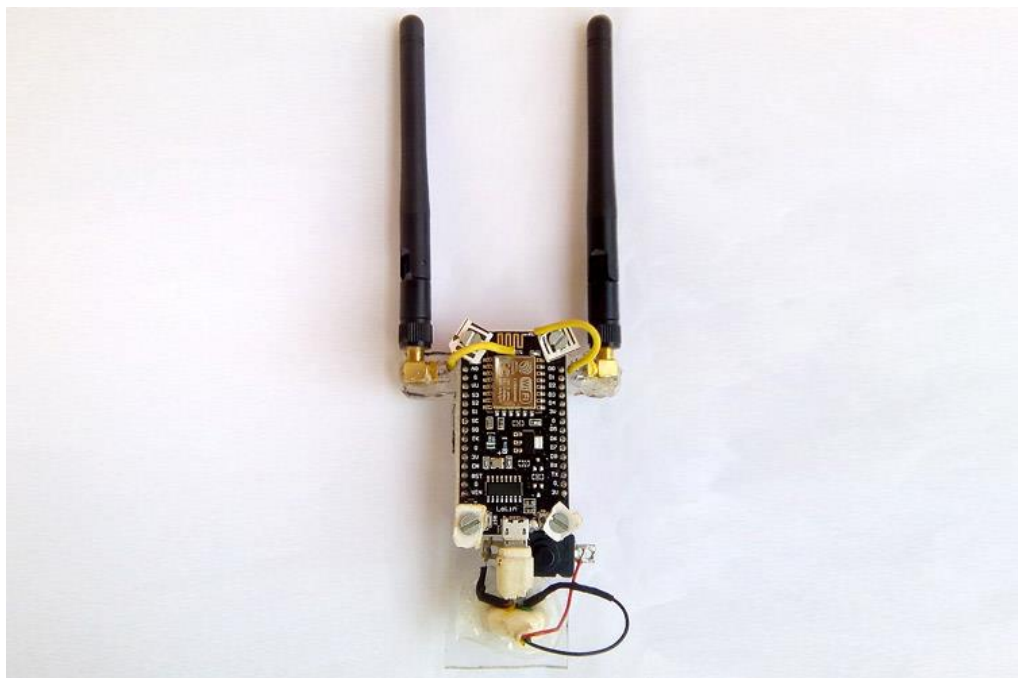


Рисунок 1.5 – Зовнішній вигляд ретранслятора на NodeMCU

1.3 Технології для розробки графічного інтерфейсу користувача

Spring Framework — це універсальна платформа з відкритим вихідним кодом для розробки корпоративних застосунків на мові Java. Ключовою

архітектурною особливістю фреймворку є реалізація принципу інверсії управління (Inversion of Control, IoC). Spring забезпечує комплексну інфраструктурну підтримку, дозволяючи розробникам зосередитись на бізнес-логіці програми, а не на низькорівневих налаштуваннях середовища. Логотип платформи наведено на рисунку 1.6.

Головною перевагою Spring є його гнучкість: він не нав'язує жорсткої моделі програмування, що зробило його популярною альтернативою традиційним стандартам Java EE. Хоча основні функції фреймворку доступні для будь-якого Java-застосунку, існують численні розширення саме для створення веб-орієнтованих систем.



Рисунок 1.6 – Логотип компанії spring

Spring Framework складається з кількох модулів, які надають широкий спектр послуг:

- контейнер Інверсії управління: Ядро фреймворку, що відповідає за створення, налаштування та управління життєвим циклом об'єктів (бінів);
- аспектно-орієнтоване програмування: Забезпечує відокремлення наскрізної функціональності (наприклад, логування) від основної бізнес-логіки;

- доступ до даних: Комплексна система безпеки, що підтримує сучасні протоколи захисту та легко налаштовується під потреби проєкту;
- управління транзакціями: об'єднує кілька API, управління транзакціями та координує операції для Java-об'єктів;
- модель-Вигляд-Управління (Model-View-Controller): програмний каркас на основі HTTP сервлета, що забезпечує створення веб-додатків і веб-служб RESTful;
- аутентифікація і авторизація: налаштовувані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за допомогою підпроєкту Spring Security (колишня система безпеки AserI для Spring);
- віддалене керування: конфігураційний вплив і управління Java - об'єктами для місцевої (локальної) або віддаленої конфігурації через JMX;
- Тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів.

Spring Boot - це корисний проєкт, метою якого є спрощення створення програм на основі Spring. Він дозволяє найпростішим способом створити web-додаток, вимагаючи від розробників мінімум зусиль з його налаштування та написання коду

Він має великий функціонал, але його найбільш значущими особливостями є: управління залежностями, автоматична конфігурація та вбудовані контейнери сервлетів

Щоб прискорити процес управління залежностями, Spring Boot неявно пакує необхідні сторонні залежності для кожного типу програми на основі Spring і надає їх розробнику за допомогою так званих starter -пакетів (spring-boot-starter-web, spring-boot-starter-data-jpa і т.д.

Starter - пакети є набором зручних дескрипторів залежностей, які можна включити у свою програму. Це дозволить отримати універсальне рішення для

всіх, пов'язаних зі Spring технологій, позбавляючи програміста від зайвого пошуку прикладів коду та завантаження з них необхідних дескрипторів залежностей.

Щоб створити Spring web-додаток, просто додайте залежність `spring-boot-starter-web`, яка підтягне в проект всі бібліотеки, необхідні для розробки Spring MVC-додатків, таких як `spring-webmvc`, `jackson-json`, `validation-api` та Tomcat.

Отже, Spring Boot збирає всі загальні залежності та визначає їх в одному місці, що дозволяє розробникам просто використовувати їх, замість того, щоб кожний раз проводити лишні операції, коли вони створюють нову програму.

1.4 Принципи комунікації складових в розроблювальній системі

Протокол TCP/IP є фундаментальною мережевою моделлю, що визначає стандарти передачі цифрової інформації. Саме на цьому наборі правил базується функціонування як глобальної мережі Інтернет, так і локальних обчислювальних мереж, незалежно від їхньої топології чи масштабів.

Абревіатура розшифровується як **Transmission Control Protocol** (протокол керування передачею) та **Internet Protocol** (міжмережевий протокол). Цей комплекс правил регламентує всі етапи комунікації: від формування пакетів даних та їх адресації до маршрутизації та коректної обробки програмним забезпеченням отримувача (рис. 1.7).

TCP/IP

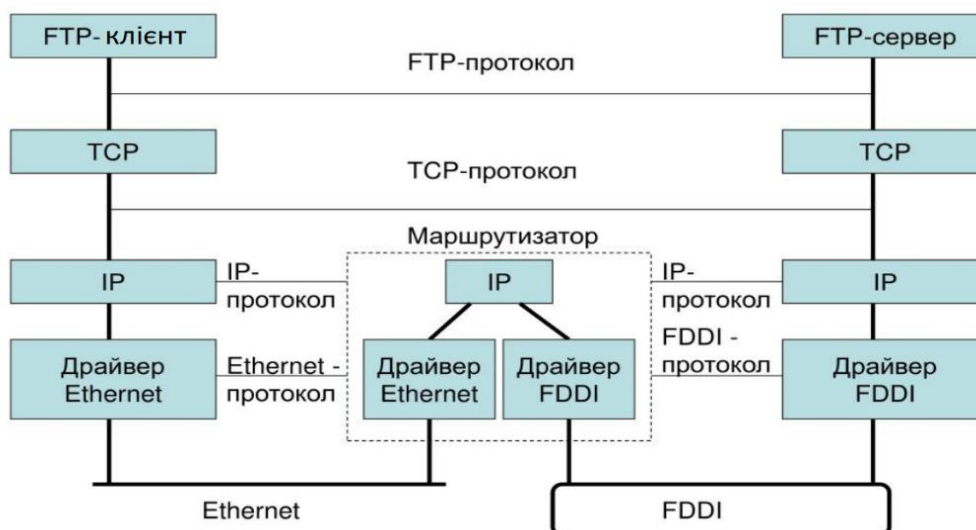


Рисунок 1.7 – стек протоколів TCP/IP

Історично стек TCP/IP розвинувся з протоколу NCP (Network Control Protocol), створеного ще у 1972 році. Офіційним стандартом для всесвітньої мережі він став у січні 1983 року, фактично започаткувавши сучасний Інтернет.

Архітектура TCP/IP часто порівнюється з еталонною моделлю OSI, проте є більш лаконічною і складається з чотирьох рівнів: фізичного (рівня мережевих інтерфейсів), міжмережевого, транспортного та прикладного (рис. 1.8).

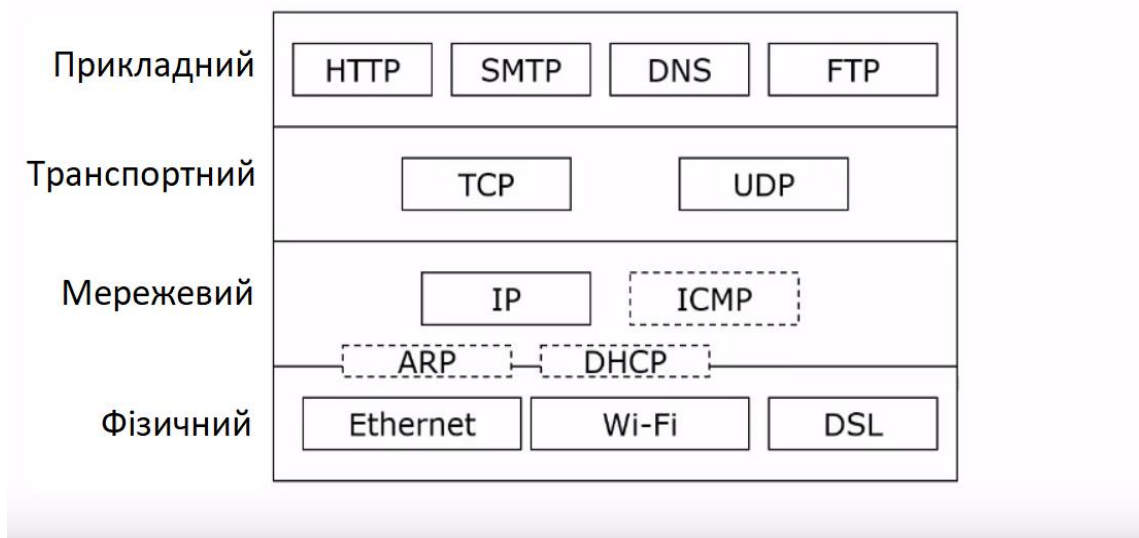


Рисунок 1.8 – рівні моделей TCP/IP

Фізичний рівень. Цей рівень відповідає за фізичні аспекти передачі даних та взаємодію обладнання (Ethernet, Wi-Fi). Його функція полягає у кодуванні інформації в сигнали, розділенні її на кадр та передачі каналами зв'язку. Тут також визначаються параметри середовища передачі, затримки та фізична сумісність хостів.

Мережевий рівень. Оскільки Інтернет є сукупністю пов'язаних локальних мереж, цей рівень відповідає за логічну адресацію та маршрутизацію. Головним інструментом тут є IP-адреса та маска підмережі. Якщо пристрої знаходяться в одній мережі, передача відбувається напряму; якщо в різних — інформація проходить через низку маршрутизаторів. На цьому рівні використовуються протоколи IPv4 та IPv6, які на сьогодні не є прямо сумісними.

Транспортний рівень. Відповідає за надійність доставки даних та цілісність сесії. Тут працюють два основні протоколи:

- TCP (Transmission Control Protocol): Гарантує доставку без втрат, перевіряє цілісність пакетів та, у разі помилки, запитує повторну передачу. Використовується для передачі файлів, веб-сторінок, пошти.

- UDP (User Datagram Protocol): Забезпечує швидку передачу без гарантії доставки, що критично для потокового відео, голосового зв'язку та онлайн-ігор, де затримки є неприпустимими.

Прикладний рівень. Цей рівень об'єднує функції трьох верхніх рівнів моделі OSI (сеансового, представлення та прикладного). Він забезпечує інтерфейс між користувачем та мережею. Саме тут працюють протоколи, з якими безпосередньо взаємодіють програми:

- HTTPS: для перегляду веб-сайтів;
- SMTP/IMAP: для роботи з електронною поштою;
- DHCP: для автоматичного призначення IP-адрес пристроям у мережі.

Система доменних імен (DNS) Для зручності користувачів в інтернеті використовується служба DNS, яка перетворює символічні імена сайтів (домени) на числові IP-адреси. DNS-сервери, що працюють через порт 53, приймають запит від браузера та повертають відповідну адресу сервера, на якому розміщено ресурс.

Інкапсуляція даних Процес передачі інформації від програми до кабелю називається інкапсуляцією: дані проходять згори донизу по рівнях, і на кожному етапі до них додається службовий заголовок відповідного протоколу. На стороні отримувача відбувається зворотний процес — декапсуляція, коли заголовки знімаються, а чисті дані передаються прикладній програмі.

1.5 Застосування технології RFID у метрологічному забезпеченні.

У сучасних автоматизованих системах реєстрації та контролю технологія радіочастотної ідентифікації (RFID) розглядається не лише як засіб ідентифікації, а як фундаментальний елемент метрологічного забезпечення, спрямований на встановлення наукових та технічних основ для досягнення єдності та точності вимірювань часових інтервалів рис 1.9.

Використання RFID-мітки як безконтактного тригера дозволяє повністю

автоматизувати момент фіксації старту, що з погляду метрології усуває суб'єктивну складову похибки, пов'язану з реакцією оператора, та забезпечує високу відтворюваність умов вимірювального процесу. Важливою перевагою є реалізація принципу метрологічної простежуваності, оскільки кожна мітка володіє унікальним ідентифікатором, який дозволяє однозначно пов'язати результат вимірювання часу з конкретним об'єктом контролю в базі даних сервера.

Завдяки стабільній робочій частоті та фіксованій дистанції зчитування, час реакції системи на появу об'єкта в полі антени стає прогнозованою величиною, що дозволяє класифікувати затримку ідентифікації як систематичну похибку, яку можна компенсувати програмними методами для підвищення класу точності системи. Окрім того, RFID-технологія забезпечує високу достовірність інформації в метрологічному моніторингу, ігноруючи сторонні сигнали та перешкоди, що мінімізує випадкові похибки реєстрації. Удосконалена система з використанням веб-інтерфейсу та Wi-Fi модуля NodeMCU дозволяє інтегрувати пристрій у розподілену інформаційно-вимірювальну мережу, де старт-сигнал синхронізується з еталонним часом сервера, забезпечуючи цілісність та єдність вимірювань на всіх етапах реєстрації даних. Таким чином, впровадження RFID у структуру системи дозволяє трансформувати її з простого засобу реєстрації у повноцінний автоматизований метрологічний комплекс із нормованими характеристиками точності та надійності.



Рисунок 1.9 - Схема інтеграції RFID-технології в систему метрологічного обліку засобів вимірювальної техніки (ЗВТ)

1.6 Аналіз сучасних технічних засобів автоматизації реєстрації та подання старт-сигналу

Сучасний ринок засобів автоматизації пропонує різноманітні технічні рішення для реєстрації подій, проте кожне з них має певні метрологічні обмеження за критеріями точності та надійності ідентифікації. Найбільш розповсюдженими аналогами є системи на базі фотоелектричних датчиків (світлових бар'єрів), які забезпечують високу миттєву точність реєстрації перетину лінії, але мають суттєвий недолік — відсутність автоматичної ідентифікації об'єкта, що вимагає додаткових засобів контролю для забезпечення метрологічної простежуваності вимірювань. Іншим рішенням є лазерні системи хронометражу, які мають мінімальну методичну похибку, проте їхня стабільність критично залежить від зовнішніх умов експлуатації, таких як освітленість або туман, що може призводити до появи випадкових помилок реєстрації.

Системи, що базуються на технології Bluetooth Low Energy (BLE) (рис. 1.10), дозволяють проводити ідентифікацію учасників, проте з метрологічної точки зору вони мають низьку точність позиціонування через широкий радіус дії сигналу, що вносить значну невизначеність у момент фіксації старт-сигналу.

Системи на базі активних RFID-міток забезпечують високу дальність зчитування (до 100 метрів), але є дорогівартісними та складними в експлуатації через необхідність постійного контролю стану вбудованих елементів живлення. Пасивні системи RFID, що працюють у високочастотному діапазоні, до яких належить і розроблюваний пристрій на базі модуля RC522, забезпечують оптимальний баланс між достовірністю ідентифікації та точністю фіксації моменту події за рахунок чітко обмеженої зони чутливості антени.

На відміну від багатьох аналогів, удосконалена система на базі NodeMCU ESP8266 використовує цифровий стек протоколів TCP/IP та Wi-Fi мережу, що дозволяє інтегрувати її в сучасні інформаційно-вимірювальні мережі з веб-інтерфейсом. Це дає можливість проводити не лише миттєву реєстрацію, а й автоматизоване архівування даних у форматі JSON на сервері для подальшого метрологічного аналізу результатів. Таким чином, проведений аналіз підтверджує, що обрана комбінація пасивного RFID-зчитувача та мікроконтролера з вбудованим Wi-Fi модулем є найбільш доцільною для розробки гнучкої та простежуваної системи автоматизації реєстрації учасників.



Рисунок 1.10 – Технологія технології Bluetooth Low Energy (BLE)

2 КОНСТРУКТОРСЬКА ЧАСТИНА

2.1 Розробка структурної схеми

Розроблена система реєстрації та управління стартом з веб-інтерфейсом є комплексом взаємопов'язаних електронних компонентів. Структурна схема пристрою, що демонструє зв'язки між елементами, наведена на рисунку 2.1. До складу апаратної частини входять такі вузли:

- джерело зовнішнього живлення;
- модуль стабілізації напруги (DC-DC перетворювач);
- мікроконтролерна платформа NodeMCU V3;
- світлодіодний модуль індикації (LED RGBW);
- модуль радіочастотної ідентифікації (RFID-зчитувач);
- пасивний ідентифікатор (RFID-мітка у формі брелока);
- драйвер дисплея I2C TM1637 з семисегментними індикаторами 3261АН.

Підсистема живлення для забезпечення автономності та універсальності використання, схема передбачає два варіанти подачі живлення:

- Від стандартної електромережі через блок живлення з вихідною напругою 5–12 В.
- Через інтерфейс USB. Така варіативність дозволяє експлуатувати пристрій як у стаціонарних умовах (від розетки), так і в мобільних (від Power Bank або ноутбука).

Для коректної роботи логічних елементів схеми використовується понижувальний DC-DC перетворювач. Його функція полягає у стабілізації вхідної напруги до робочих рівнів мікроконтролера та периферії (3.3 В або 5 В), що захищає компоненти від перепадів енергії.

NodeMCU керує RFID мітками, та обробляє отримані дані із веб-сервера. Також плата має вбудований WIFI модуль. Для її живлення використано DC-DC перетворювач, який перетворює напругу в 5/3.3В.

Для фіксування початку використовується брелок RFID мітки.

Світлодіод LED використовується для візуального вигляду.

Індикатор 3261АН виводить час початку відліку.

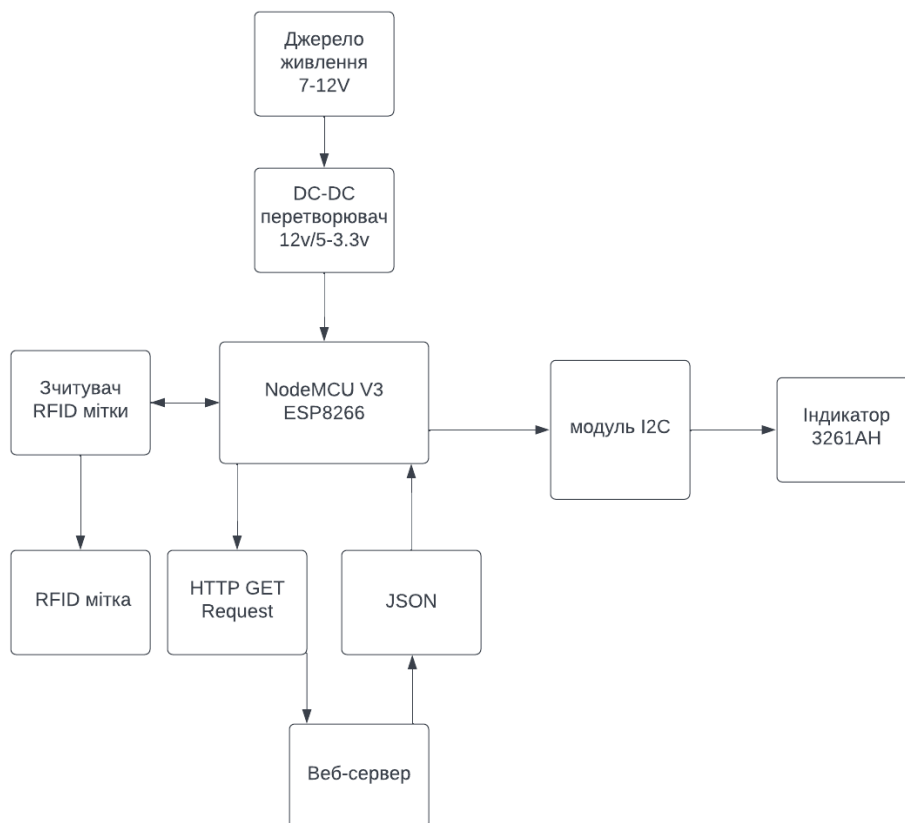


Рисунок 2.1 – Структурна схема старту

2.2 Розробка електричної принципової схеми

Електрична принципова схема пристрою наведена в додатку Б пояснювальної записки.

Живлення на компоненти та інтерфейс надходить з модуля живлення макетних плат MB102, через USB або роз'єм 5,5/2,1мм. Вхідна напруга 7-12В надходить через кнопковий вимикач SB1 на стабілізатор DA1, він стабілізує

напругу до 5В, після цього для стабілізації використовується стабілізатор DA2, він стабілізує напругу до 3,3В.

Живлення на NodeMcu подається через роз'єм мікро-USB. Вхідна напруга 3,7 - 20В, після чого вона перетворюється на робочу напругу в 3 - 3,6В. Також мікро-USB роз'єм використовується для конвертування USB-сигналів в UART-сигнали, які спочатку потрапляють на мікроконтролер DD2, після чого сигнали з цих виходів надходять на мікроконтролер DD1 і навпаки.

Кнопка SA1 яка під'єднана до входу RST використовується для скидання мікросхеми до стандартних налаштувань

Кнопка SA2 під'єднана до землі на GPIO0, її можна використовувати як звичайну кнопку.

Індикатор HL2 світиться коли подається живлення. Після чого подається струм на HL3, HL4, HL5, HL6 підключені через резистори R5, R6, R7, R8.

Семисегментні індикатори керуються за допомогою модуля DD1. Сигнал з модуля DD1 переходить на мікроконтролер DD3, після чого з даного мікронтролера сигнал переходить на індикатори HL7, HL8, HL9, HL10 через анод на резистори R23-30, які формують цифру від 0 до 9. Через катод подається сигнали на резистори R31-34, який переходить на транзистори VT3-VT6.

Кварцеві резонатори ZQ1 і ZQ2 використовуються для стабілізації частоти.

Зчитувач карток RFID керується за допомогою мікроконтролера DD4, на котушку яка генерує високочастотне магнітне поле L3 надходить сигнал з мітки, тому коли карта або брелок знаходиться близько до електромагнітного поля, у його антені утворюється індукційна напруга, яка подається через конденсатори C17, C18, C19, C20, C24, C25, C26, C27 і катушки L1, L2. Після чого сигнал надходить на мікроконтролер DD4, а сигнал з мікронтролера DD4 надходить на DD1.

На контакт 3,3В подається вихідна напруга внутрішнього стабілізатора.

На вивід Vin для підключення зовнішнього джерела живлення 5В. Стабілізатор DA1 дозволяє подавати живлення на Vin у широкому діапазоні від 5 до 10 В.

RST використовується для скидання для скидання та відключення живлення мікроконтролера DD4.

IRQ вихід переривань, який дає сигнал мікроконтролеру, коли поблизу буде мітка RFID.

MISO (Master-In-Slave-Out) вихід при включеному інтерфейсі SPI, або як вихід послідовних даних при включеному інтерфейсі UART.

MOSI (Master Out Slave In) вхід SPI для модуля RC522.

SCK (Serial Clock) приймає тактові імпульси, які надаються мікросхемою DD1.

SDA вхід послідовних даних, коли включений інтерфейс UART. Цей вихід позначається у вигляді квадрата, щоб можна було його використовувати як опорну точку для ідентифікації інших виводів.

2.3 Обґрунтування вибору платформи NodeMCU

NodeMCU — це високоінтегрований апаратно-програмний комплекс (SoC), оснащений вбудованим стеком протоколів TCP/IP, що дозволяє реалізувати повноцінну мережеву взаємодію пристроїв через Wi-Fi. Для реалізації проекту обрано актуальну модифікацію платформи — NodeMCU ESP8266 V3. Зовнішній вигляд плати наведено на рисунку 2.2.

Архітектура та швидкодія Основою платформи є модуль ESP-12E, побудований на базі чіпа ESP8266. Обчислювальні можливості забезпечуються 32-бітним процесором із тактовою частотою 80 МГц (з програмною можливістю підвищення до 160 МГц) та 4 Мб флеш-пам'яті. Висока щільність інтеграції

компонентів дозволяє мінімізувати кількість зовнішніх схем та ефективно використовувати простір на друкованій платі.

Інтерфейси та програмування Для завантаження прошивки та налагодження використовується роз'єм micro-USB. Зв'язок із комп'ютером забезпечує перетворювач інтерфейсів USB-UART (наприклад, CP2102), який транслює сигнал USB у послідовний порт. На платі також розміщено користувацький світлодіод (підключений до GPIO2/D0) та тактові кнопки:

- RST — для апаратного перезавантаження системи;
- FLASH (GPIO0) — для переведення контролера в режим завантаження програмного забезпечення.

Система живлення Живлення модуля реалізовано за гнучкою схемою: через порт micro-USB або через контакт VIN (зовнішнє джерело 5 В). Оскільки робоча напруга логіки ESP8266 становить 3.0–3.6 В, плата оснащена вбудованим стабілізатором напруги. Вихід 3.3 В виведено на відповідні піни (3V3) для живлення зовнішніх датчиків та периферії.



Рисунок 2.2 – NodeMcu V3 ESP8266

Розташування та маркування контактів (розпінування) платформи NodeMCU V3 представлено на рисунку 2.3.

Функціональне призначення основних інтерфейсів та виводів плати:

- Vin: вхід для підключення зовнішнього нерегульованого джерела живлення (використовується, коли плата не підключена через USB).
- 3.3V: вихід стабілізованої напруги від внутрішнього регулятора, призначений для живлення зовнішніх датчиків та компонентів.
- VUSB (VU): контакт прямого доступу до напруги 5 В від шини USB.
- GPIO (General Purpose Input/Output): порти введення-виведення загального призначення, які програмно налаштовуються для керування виконавчими механізмами або зчитування сигналів.
- RST (Reset): вхідний контакт для апаратного перезавантаження мікроконтролера.
- ADC0 (A0): вхід вбудованого 10-розрядного аналого-цифрового перетворювача (АЦП) для вимірювання аналогових сигналів.
- UART: асинхронний послідовний інтерфейс (піни RX/TX) для обміну даними з комп'ютером або іншими пристроями.
- SPI: послідовний периферійний інтерфейс для високошвидкісної комунікації.
- SDIO: інтерфейс безпечних цифрових входів/виходів, що застосовується для роботи з картами пам'яті SD або комутації із зовнішньою флеш-пам'яттю.

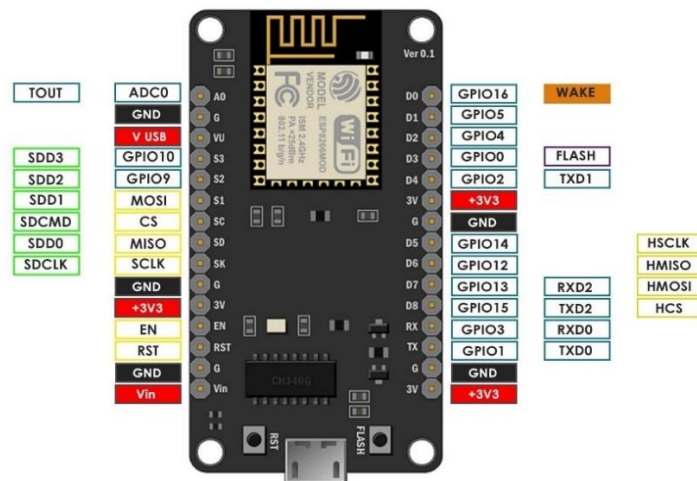


Рисунок 2.3 – Розпінування плати NodeMcu V3 ESP8266

2.4 Додаткові модулі

Для проекту був використаний зчитувач RF-міток RFID модуль RC522 зображений на рисунку 2.4

Радіочастотна ідентифікація (RFID) – це технологія безконтактної ідентифікації об'єктів за допомогою радіочастотного каналу зв'язку. Ідентифікація об'єктів проводиться за унікальним ідентифікатором, який має кожна електронна мітка. Зчитувач випромінює електромагнітні хвилі певної частоти. Мітки надсилають у відповідь інформацію – ідентифікаційний номер, дані пам'яті та інше.

Переваги технології RFID:

- безконтактна;
- можливість прихованої установки міток;
- висока швидкість зчитування даних;
- можливість встановлення у шкідливих середовищах;
- неможливість підробки.

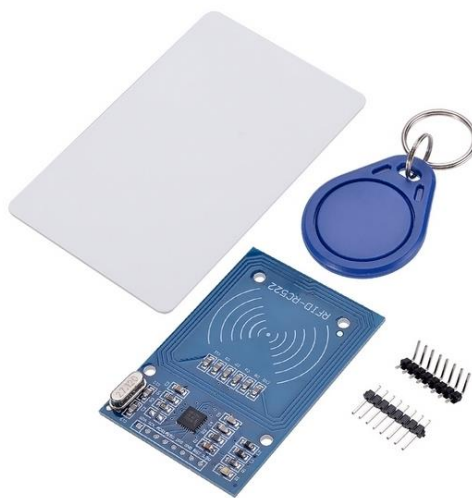


Рисунок 2.4 – RFID модуль RC522

Модуль живлення MB102 3.3V 5V використовується для стабілізації вхідної напруги від зовнішніх джерел живлення та подачі його на плату зображено на рисунку 2.5. Модуль використовується для подачі стабілізованої напруги на макетні плати. Є можливість підключення до двох макетних плат.

Для використання модуля живлення потрібно підключити до нього зовнішнє джерело живлення з напругою 6,5 – 12В постійного струму.

Технічні характеристики:

- вхідна напруга 6.5В - 12В (DC) або від USB;
- вихідна напруга 3.3В / 5В (перемикається);
- максимальний вихідний струм <700 мА;
- відстань між шинами живлення 42мм.



Рисунок 2.5 – Модуль живлення MB102 для макетних плат.

Модуль на базі мікросхеми TM1637 являє собою семисегментний чотирирозрядний дисплей, призначений для візуалізації цифрової інформації та обмеженого набору символів латинського алфавіту.

Принцип роботи пристрою базується на динамічному керуванні окремими світлодіодними сегментами, комбінація яких формує необхідний знак (цифру або літеру). Головною перевагою даного модуля є використання послідовного двопровідного інтерфейсу для комутації з мікроконтролером (аналогічно протоколу I2C). Це дозволяє суттєво заощадити порти вводу-виводу керуючого пристрою: замість прямого підключення 12 пінів, використовуються лише два сигнальні контакти (CLK та DIO) та живлення. Зовнішній вигляд модуля наведено на рисунку 2.6.

Технічні параметри плати індикатора:

- Інтерфейс керування: послідовний (сумісний з логікою роботи I2C);
- Робоча напруга: 3.3 В – 5 В (сумісність як з 3.3-вольтовою логікою ESP8266, так і з 5-вольтовою Arduino);
- Тип індикації: червоні світлодіодні сегменти;
- Діапазон робочих температур: від -10°C до +60°C;
- Габаритні розміри: 42 x 23 x 11 мм.



Рисунок 2.6 – Дисплей модуль TM1637.

2.5 Аналіз організації живлення та інтерфейсів взаємодії

При розробці апаратної частини метрологічного модуля ключовим завданням було забезпечення прецизійної стабільності живлення та мінімізації затримок при передачі даних. Це необхідно для гарантування точності часових параметрів сигналу "Старт" та надійної фіксації подій.

Організація підсистеми живлення для точних вимірювань Для мінімізації шумів та пульсацій, які можуть вплинути на роботу внутрішнього таймера

мікроконтролера, схему живлення побудовано за каскадним принципом. Система підтримує два режими роботи:

- Лабораторний режим (USB 5 В): Використовується при налаштуванні та калібруванні пристрою.
- Робочий режим (зовнішнє джерело 7–12 В): Забезпечує автономність пристрою при польових випробуваннях.

Узгодження напруг реалізовано через подвійну стабілізацію:

- Перший контур (AMS1117-5.0, DA1): Знижує вхідну напругу до 5 В. Ця лінія використовується для живлення силової частини індикації, що дозволяє відокремити потужні споживачі від логічної частини.

- Другий контур (AMS1117-3.3, DA2): Формує "чисту" напругу 3,3 В для живлення ядра ESP8266 та модуля RFID. Це критично для метрології, оскільки стабільність напруги живлення прямо впливає на джитер (тремтіння фази) тактового генератора, а отже — на точність формування часових інтервалів.

Таке розділення дозволяє уникнути просідання напруги на процесорі в момент увімкнення яскравих світлових сигналів "Старт", що гарантує безперервність вимірювального процесу.

Організація інтерфейсів обміну даними Архітектура взаємодії з периферією спроектована так, щоб мінімізувати програмні затримки:

- Інтерфейс SPI (для модуля RC522). Використання апаратного SPI (лінії MISO, MOSI, SCK, SDA) дозволяє здійснювати швидку ідентифікацію оператора або об'єкта вимірювання. Висока швидкість шини гарантує, що процес зчитування мітки не блокує основний цикл програми, дозволяючи мікроконтролеру пріоритетно обробляти події таймера.

- Інтерфейс I2C (для дисплея TM1637). Для візуалізації результатів вимірювань або службової інформації використовується драйвер TM1637. Обмін даними по двох лініях (DIO, CLK) забезпечує надійне виведення

цифрових значень без використання ресурсномістких переривань, що важливо для збереження детермінованості роботи системи реального часу.

- Керування сигнальними лініями (світлова індикація). Комутація сигналів "Старт" реалізована через транзисторні ключі S8050 (VT3–VT6). Використання ключів забезпечує гальванічну розв'язку керуючих виводів мікроконтролера від силового навантаження. Це запобігає впливу перехідних процесів у світлодіодах на роботу вимірювального ядра.

2.6 Конструктивні особливості забезпечення точності реєстрації подій

Для використання розробленого пристрою як засобу метрологічного контролю або еталонного генератора сигналів "Старт", конструкція повинна мінімізувати вплив зовнішніх факторів на результати вимірювань. На основі аналізу елементної бази (Додаток Б) виділено такі конструктивні рішення, спрямовані на підвищення точності:

Стабілізація часових параметрів Основою метрологічної достовірності системи є стабільність тактової частоти мікроконтролера. У схемі відмовлено від використання вбудованих RC-генераторів, які мають високу температурну нестабільність. Натомість застосовано зовнішні кварцові резонатори ZQ1 (12 МГц) та ZQ2 (27 МГц). Використання кварцової стабілізації дозволяє зменшити похибки формування часових інтервалів до значень, що не перевищують десятків мікросекунд, що є достатнім для фіксації моменту старту.

Завадостійкість каналу передачі даних Оскільки система працює у зашумленому діапазоні 2.4 ГГц, де можлива втрата пакетів даних, для модуля NodeMCU передбачено використання зовнішніх антен, що підключаються через роз'єм IPEX. Винесення антени за межі екрануючих елементів корпусу та плати дозволяє:

- Збільшити радіус впевненого прийому/передачі сигналу про старт.

- Зменшити кількість повторних запитів TCP/IP, які можуть вносити непередбачувані затримки у фіксацію подій на сервері.

Електромагнітна сумісність вузла ідентифікації Вхідний каскад зчитувача RFID містить коливальний контур (котушки L1, L2, L3 та конденсатори C17–C27). Конструктивне розташування цієї частини вимагає відсутності металевих елементів у безпосередній близькості до антени, щоб уникнути розлаштування резонансної частоти 13,56 МГц. Це забезпечує стабільну зону детекції мітки, що гарантує, що сигнал "Старт" буде подано саме в момент піднесення ідентифікатора, без "сліпих зон".

Температурна стійкість Обрані компоненти, зокрема дисплейний модуль TM1637, розраховані на роботу в температурному діапазоні від -10 до +60°C. Це дозволяє використовувати пристрій для метрологічних вимірювань не лише в лабораторних умовах, а й на відкритих випробувальних майданчиках без додаткових систем термостатування.

2.7 Метрологічний аналіз вимірювального каналу системи

Оскільки розроблюваний пристрій виконує функцію фіксації часових інтервалів, він класифікується як засіб вимірювальної техніки (ЗВТ). Для оцінки його точності проведено аналіз складових похибок апаратної частини.

1. Інструментальна похибка задаючого генератора Основним джерелом похибки вимірювання часу є нестабільність частоти тактового генератора. У схемі використано кварцовий резонатор ZQ1 з номінальною частотою 12 МГц. Відносна похибка частоти для даного класу резонаторів становить 50 ppm (частин на мільйон).

Абсолютна похибка вимірювання часу прямо пропорційна тривалості інтервалу вимірювання. Вона розраховується як добуток вимірюного часу на відносну нестабільність частоти резонатора.

Практичне застосування: Для заїзду тривалістю 1 година (3600 с), максимальне відхилення складе не більше 0,18 с. Це дозволяє використовувати пристрій для хронометражу змагань аматорського рівня, де дискретність фіксації становить від 0,1 до 1 с.

2. Методична похибка просторової дискретизації Специфікою RFID-систем є наявність зони чутливості антени (близько 5 см), а не точкової лінії фінішу. Це створює просторову невизначеність моменту реєстрації. Часова похибка у цьому випадку визначається як відношення довжини зони детекції антени до швидкості руху об'єкта.

Практичне застосування: У системах логістичного контролю, де швидкість конвеєра є постійною, ця похибка стає систематичною (постійною величиною) і може бути повністю компенсована калібруванням шляхом введення поправки.

3. Вплив зовнішніх факторів Стабільність напруги живлення 3,3 В (забезпечується стабілізатором DA2) гарантує сталість амплітуди електромагнітного поля, а отже — стабільність геометричних розмірів зони зчитувача. Це забезпечує збіжність результатів вимірювань (Repeatability) при багаторазових проїздах.

3 ПРОГРАМНА ЧАСТИНА

3.1 Розробка алгоритму роботи пристрою

Алгоритм – це набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій. Система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для візуалізації алгоритмів часто використовують блок-схеми (рис. 3.1).

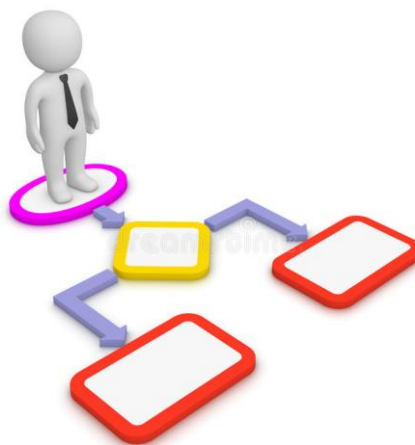


Рисунок 3.1 – Блок-схема

В поточному випадку алгоритм – це визначення поведінки виконавчих елементів пристрою в залежності від значень фізичних величин, отриманих від датників та інших джерел вхідної інформації. Поширеним критерієм оцінки алгоритмів є час роботи та порядок зростання тривалості роботи в залежності від обсягу вхідних даних.

Для розробки алгоритму роботи роботизованого транспортного засобу спершу необхідно визначити вхідні параметри, аккумулятора. Основними параметрами для роботизованого транспортного засобу є швидкість обробки даних bluetooth та велика ємність аккумулятора. Ємність та напруга

аккумулятора впливають на час роботи та швидкість руху транспортного засобу. Тому ці параметри і будуть вхідними для системи керування.

Блок-схема розробленого роботизованого транспортного засобу наведена на додатку Г.

На початку ми підключаєм різні бібліотеки для роботи та оголошуєм змінні для роботи з Wi-Fi, енкодером і сервером. Даліше встановлюєм таймер для регулювання частото запитів, щоб дати серверу можливість успішно обробляти інформацію.

Також на сервер можуть зайти ті користувачі, які авторизовані на ньому.

Після чоґо починається опитування сервера. Якщо наш сервер відповів нам кодом 200, починаєм парсити дані у форматі JSON, якщо код не рівний 200, тоді іде обробка помилок. Після парсинґу ми перевіряємо статус гонки, якщо вона ще не почалась то просто очікуємо початку, якщо гонка почалась, вмикається зчитувач RFID мітки, який зчитує дані з RF мітки. Коли мітку зчитано вмикається індикатор який відображає інформацію про зчитування мітки.

Код мітки відправляється на сервер, після чоґо починає очікувати закінчення гонки, опитуючи сервер. Якщо сервер відповідає кодом 200, це означає що машинка перетнула фініш, тоді вимикається індикатор і скидується таймер.

Блок-схема автоматизованої системи для реєстрації та подання старт-сигналу з веб інтерфейсом наведена на МР.МТТм-11.00.00.000 БС Г

3.2 Реалізація логіки перемикавання режимів роботи (State Machine)

Для забезпечення надійної та передбачуваної роботи пристрою програмний код побудовано за принципом кінцевого автомата (Finite State Machine). Керування здійснюється за допомогою змінної стану (у програмному

кодi позначена як `int c`), яка визначає поточний режим функціонування системи.

Виділено три основні алгоритмічні стани:

- Режим очікування дозволу та сканування ($c = 0$): У цьому стані мікроконтролер циклічно надсилає запити на сервер для перевірки статусу системи. Алгоритм очікує отримання команди "raceStarted" у відповіді сервера. Тільки після отримання цього дозволу активується функція опитування RFID-зчитувача. При виявленні мітки її унікальний ідентифікатор (UID) зчитується, конвертується у рядковий формат, а на дисплей виводиться повідомлення про успішне сканування ("SCAN" та "GO"). Після цього система автоматично перемикається у наступний стан.

- Режим транзакції даних ($c = 1$): Пріоритетом даного стану є гарантована передача даних на сервер. Система формує спеціальний HTTP-запит, додаючи до URL-адреси зчитаний токен мітки. Алгоритм очікує від сервера підтвердження успішного запису даних у базу (код відповіді 200). Доки підтвердження не отримано, перехід до наступного етапу блокується. Це запобігає втраті даних про старт учасника при нестабільному зв'язку.

- Режим блокування та завершення ($c = 2$): Після успішної реєстрації старту пристрій переходить у режим очікування фінішу. Зчитування нових міток програмно блокується, щоб уникнути дублювання запитів. Система продовжує опитувати сервер, очікуючи статус "raceEnded". При його отриманні змінні стану обнуляються, і алгоритм повертається на початок циклу.

3.3 Програмна реалізація протоколу обміну даними

Взаємодія пристрою з веб-сервером реалізована через протокол HTTP з використанням бездротової мережі Wi-Fi. Для забезпечення сумісності та безпеки даних алгоритм включає наступні етапи обробки інформації:

Автентифікація та формування заголовків Для доступу до API сервера використовується механізм Basic Auth. Програма автоматично кодує пару "логін:пароль" (authUsername та authPassword) у формат Base64 перед відправкою кожного запиту. Отриманий хеш додається до заголовка Authorization HTTP-пакета. Це дозволяє серверу ідентифікувати пристрій як довірене джерело даних.

Парсинг вхідних даних Відповіді від сервера надходять у форматі JSON. Для оптимізації швидкодії мікроконтролера ESP8266, замість використання важких бібліотек десеріалізації, застосовано метод прямого рядкового пошуку (substring). Алгоритм перевіряє конкретні позиції у рядку відповіді (payload), де міститься інформація про статус гонки. Наприклад, перевірка підрядка з 15-го по 26-й символ дозволяє миттєво визначити команду "raceStarted" або "raceEnded", мінімізуючи використання оперативної пам'яті.

Обробка кодів стану HTTP Алгоритм аналізує код відповіді сервера (httpResponseCode).

- Код 200 (OK): Сигнал про успішну операцію, дозволяє подальше виконання логіки.
- Інші коди або від'ємні значення (помилка з'єднання): Ігноруються логікою керування, при цьому в послідовний порт виводиться діагностичне повідомлення про помилку. Такий підхід забезпечує стійкість системи до тимчасових збоїв мережі.

3.4 Синхронізація процесів та керування апаратними ресурсами

Для забезпечення стабільної роботи системи в реальному часі реалізовано неблокуючий алгоритм керування часом.

Використання програмного таймера. Замість функції повної зупинки процесора `delay()`, яка блокує роботу мережевого стека, використовується порівняння системного часу. Функція `millis()` повертає кількість мілісекунд з моменту запуску. Алгоритм перевіряє різницю між поточним часом та часом останньої дії (`lastTime`). Якщо ця різниця перевищує встановлений інтервал

`timerDelay (500 мс)`, виконується тіло циклу. Це дозволяє мікроконтролеру підтримувати активне Wi-Fi з'єднання у фоновому режимі.

Робота з апаратними перериваннями RFID. Взаємодія з модулем RC522 здійснюється через бібліотеку MFRC522. Алгоритм використовує методи перевірки наявності нової карти (`PICC_IsNewCardPresent`) та читання її серійного номера (`PICC_ReadCardSerial`). Після успішного зчитування обов'язково викликаються функції `PICC_HaltA()` та `PCD_StopCrypto1()`, які переводять картку в сплячий режим та зупиняють шифрування. Це необхідно для коректного завершення сеансу зв'язку та готовності зчитувача до наступних операцій.

3.5 Алгоритмічні методи компенсації динамічних похибок

У цифрових системах реального часу програмне забезпечення вносить власні похибки, пов'язані з дискретністю обробки сигналів. Алгоритм роботи пристрою містить методи для мінімізації цих впливів.

1. Мінімізація похибки квантування часу. Оскільки мікроконтролер обробляє код циклічно, момент виникнення події (фізичний проїзд) і момент її реєстрації (програмний) можуть відрізнитися. Ця випадкова похибка підпорядковується рівномірному закону розподілу. Для зменшення впливу цієї похибки в алгоритмі застосовано метод пріоритетного переривання. Середньоквадратичне відхилення моменту реєстрації оцінюється як частка від ділення періоду циклу програми на корінь квадратний з трьох.

Метрологічний аспект: Це дозволяє оцінити джитер (фазове тремтіння) моменту реєстрації, який для даної системи не перевищує 10–20 мс, що є достатнім для більшості задач автоматизації.

2. Цифрова фільтрація завад Для усунення явища "брязкоту контактів" (багаторазового зчитування мітки в зоні дії поля) застосовано алгоритм з часовим гістерезисом. Після першої валідної фіксації унікального ідентифікатора система ігнорує вхідні дані протягом захисного інтервалу.

Метрологічний аспект: Це забезпечує однозначність результату вимірювання, виключаючи появу артефактів та дубльованих записів у протоколі змагань.

3. Асинхронна модель передачі даних Алгоритм розділяє процеси вимірювання (фіксація часу локально) та передачі (відправка HTTP-запиту). Це робить точність вимірювання незалежною від латентності (затримок) мережі Wi-Fi.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Обґрунтування вибору інтегрованого середовища розробки

Для написання програмного забезпечення (прошивки) мікроконтролера NodeMCU ESP8266 обрано середовище Arduino IDE (Integrated Development Environment). Цей вибір зумовлений кількома факторами, що є критичними для розробки метрологічних систем:

- Нативна підтримка C++: Arduino IDE використовує діалект мови C++, що дозволяє писати високоефективний код з прямим доступом до апаратних таймерів та регістрів процесора.
- Кросплатформність та розширюваність: Завдяки менеджеру плат (Board Manager), середовище дозволяє легко інтегрувати пакет підтримки ядра ESP8266, що забезпечує доступ до специфічних функцій Wi-Fi модуля, яких немає у стандартних платах Arduino .
- Інструменти налагодження: Вбудований монітор послідовного порту (Serial Monitor) дозволяє в реальному часі відстежувати стан підключення до сервера, коди помилок HTTP та дані з RFID-зчитувача, що значно спрощує процес тестування пристрою.

4.2 Огляд використаних бібліотек та залежностей

Для реалізації складної логіки взаємодії з периферією використано модульний підхід із залученням спеціалізованих бібліотек. Це забезпечує надійність коду, оскільки використовуються перевірені драйвери низького рівня. У проєкті задіяно наступні заголовні файли :

- ESP8266WiFi.h та ESP8266HTTPClient.h: Базові бібліотеки для керування Wi-Fi модулем та реалізації клієнтської частини протоколу

HTTP. Вони забезпечують формування GET-запитів, обробку заголовків та підтримку стабільного з'єднання TCP/IP.

- `MFRC522.h`: Драйвер для роботи зі зчитувачем RFID на базі чіпа RC522. Бібліотека реалізує протокол SPI та функції для роботи з картами стандарту MIFARE (антиколізія, читання UID, перевірка типу карти).

- `TM1637.h`: Бібліотека для керування 4-розрядним семисегментним дисплеєм. Вона дозволяє виводити не лише цифри, а й деякі символи латинського алфавіту (наприклад, "SCAN", "GO", "STOP") без необхідності вручну кодувати бітові маски сегментів.

- `WiFiManager.h`: Критично важлива бібліотека для зручності експлуатації. Вона дозволяє уникнути "жорсткого" прописування SSID та пароля мережі в коді. Якщо пристрій не може підключитися до Wi-Fi, він створює власну точку доступу, через яку користувач може ввести нові налаштування мережі.

- `base64.h`: Використовується для кодування облікових даних адміністратора перед відправкою на сервер (Basic Authentication).

4.3 Програмний код прошивки мікроконтролера

Нижче наведено повний лістинг програми, що реалізує алгоритм, описаний у розділі 3. Код включає налаштування апаратної частини, реалізацію кінцевого автомата та логіку обміну даними з сервером.

Лістинг коду автоматизованої системи автоматизованої системи для реєстрації та подання старт-сигналу з веб-інтерфейсом МР.МТТм-11.00.000 КД
Д

4.4 Програмні засоби верифікації та валідації вимірювань

Для підтвердження достовірності отриманих даних та контролю справності системи, програмне забезпечення містить вбудовані засоби діагностики.

1. Перевірка цілісності даних у метрології критично важливо виключити ймовірність підміни або спотворення результату.

- Рівень CRC: Бібліотека драйвера RFID виконує апаратну перевірку контрольної суми даних, зчитаних з карти. Помилкові пакети відкидаються автоматично.

- Рівень протоколу: Отримання коду відповіді "200 ОК" від сервера є підтвердженням того, що ланцюг передачі вимірювальної інформації працює коректно.

2. Методика вимірювання латентності Програмний код дозволяє провести верифікацію швидкодії системи. Через налагоджувальний порт виводяться дві часові мітки: момент детекції мітки та момент отримання відповіді від сервера. Повна затримка системи розраховується як різниця між цими двома значеннями часу.

Практичне застосування: Ця методика дозволяє оператору перед початком змагань переконатися, що затримка не перевищує критичних значень (наприклад, 2 секунди), що є умовою допуску системи до роботи.

3. Забезпечення надійності для гарантування безперервності процесу вимірювання задіяно програмні механізми відновлення після збоїв (Watchdog). Якщо зв'язок втрачено, система переходить у буферизований режим, що забезпечує збереження результатів вимірювань. Це відповідає вимогам до надійності засобів автоматизації.

5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

5.1 Технічні характеристики системи для реєстрації учасників змагань та подавання сигналу старт-сигналу

Розроблена система для реєстрації та подання змагань та подавання старт-сигналу характеризується наступними технічними характеристиками:

- модуль зчитування RFID RC522;
- дисплей TM1637;
- мікроконтролер ESP8266 ch340;
- максимальне споживання струму: 700 мА;
- живлення пристрою 5В/7-12 В;
- маса пристрою: 1 кг;
- габаритні розміри:
- робоча температура: від -15 °С до +45 °С;
- підключення до ПК: кабель microUSB.

5.2 Інструкція з експлуатації системи для реєстрації учасників змагань

Перед початком роботи з пристроєм вам потрібно підключити кабель microUSB з живленням від PoweBank або телефонного блоку живлення.

Для запуску даної системи від користувача нічого не потрібно, тому що, дана система вмикається автоматично як тільки на неї подається живлення. Перед початком змагань в роботизований транспортний засіб потрібно встановити RF-мітку – (рис. 5.1).



Рисунок 5.1 – Вигляд RF – мітки

Вам це потрібно робити для того, щоб зчитувач RFID (рис. 5.2) мітки зміг зчитати дані з роботизованого транспортного засоби, та відправити на сервер дані(токен) про початок заїзду. Після того як мітку зчитано, та були відправлені дані на сервер, тоді вмикається дисплей модуль, який демонструє на екрані час заїзду.

Як тільки сервер отримав токен, дані про роботизований транспортний засіб заносяться в базу даних. На початку змагань, наш пристрій починає опитувати сервер про закінчення заїзду. Коли заїзд завершено, ми обробляєм дані про кінець цього заїзду, після чого зупиняється таймер на дисплеї і показується результат за який роботизований транспортний засіб перетнув фінішну пряму.

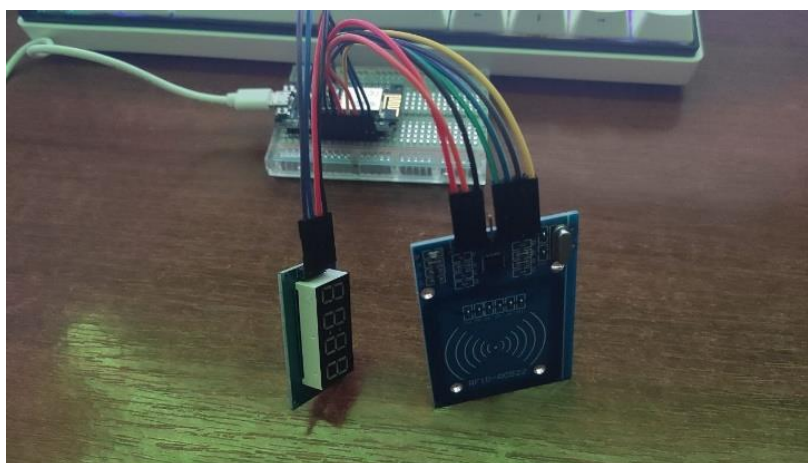


Рисунок 5.2 – Вигляд RFID зчитувача

5.3 Визначення відстанні зчитування RF мітки

Для зчитування RF – мітки автоматизованої системи для реєстрації учасників змагань використовується зчитувач мітки RFID RC522, для того щоб її зчитати ви підключаєте їх у входи А, а потім порти RST, SDA до NodeMCU ESP8266 (рис. 5.3).

Порт RST використовуються для скидання. А SDA для отримання інформації, потім ці порти ініціалізуються в середовищі розробки Arduino IDE (рис. 5.4).

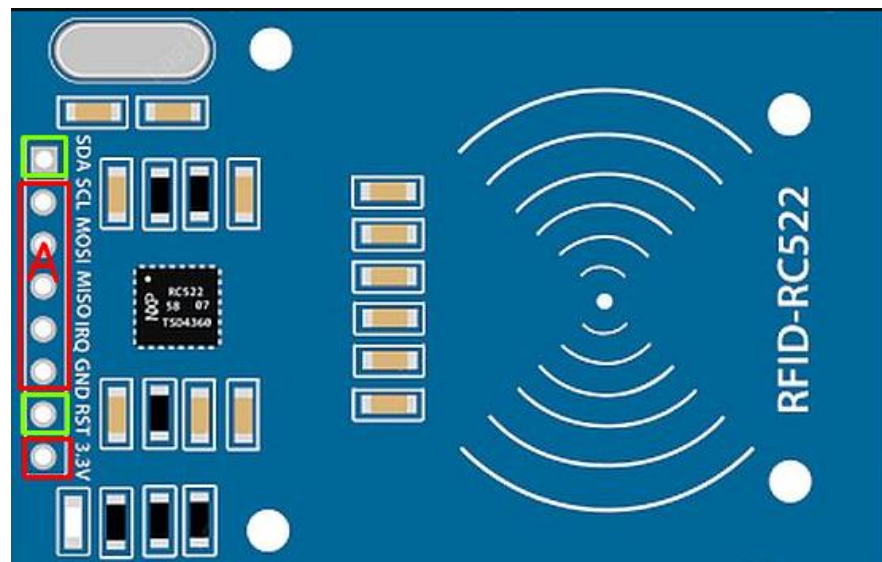


Рисунок 5.3 – Порти підключення зчитувача RFID

Відстань зчитування RF – мітки буде виміряно у різних умовах. Для цього було проведено декілька експериментів, де мітка зчитувалась на різних дистанціях, а також з перегородами на цих дистанціях.

```

#include <SPI.h>
#include <ESP8266WiFi.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <base64.h>
#include <MFRC522.h>
constexpr uint8_t RST_PIN = D3; // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN = D4; // Configurable, see typical pin layout above
MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
MFRC522::MIFARE_Key key;

```

Рисунок 5.4 – Ініціалізація портів в Arduino IDE

За допомогою вимірів було визначено розміри перешкод і максимальний шлях зчитування RF – мітки (рис. 5.5)

Максимальний шлях зчитування RF – мітки рівний 27 міліметрів.

Розмір першої перешкоди рівний 2 міліметрів.

Розмір другої перешкоди рівний 5 міліметрів.

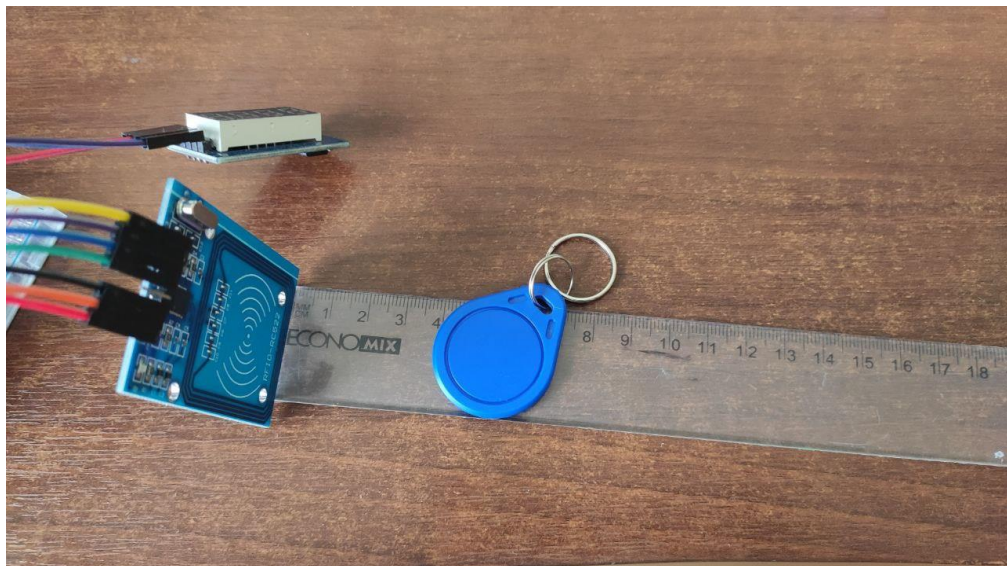


Рисунок 5.5 – Вимірювання шляху зчитування мітки.

При досліді першого експерименту було взято аркуш паперу А4 і складено в чотири рази, його товщина дорівнювала 2 міліметра (рис. 5.6)

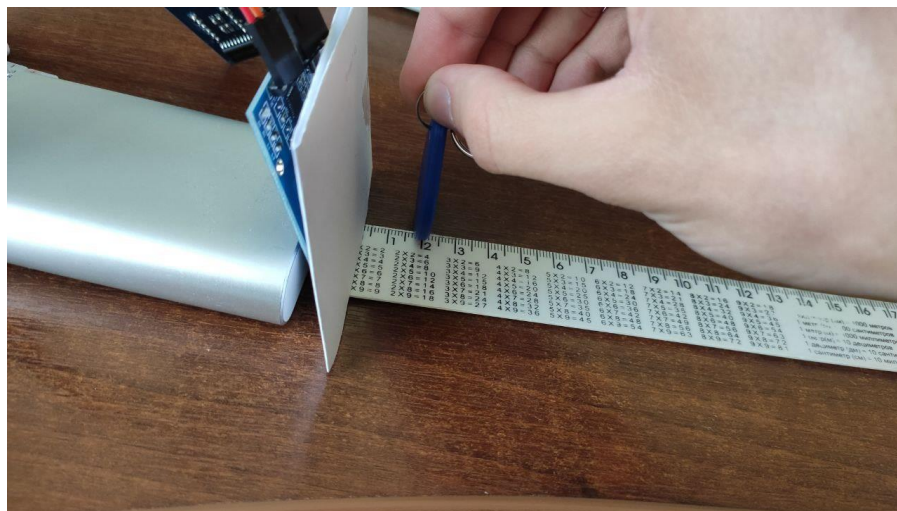


Рисунок 5.6 – Вимірювання шляху зчитування з першою перешкодою.

Для другого експерименту було вибрано телефонний чохол товщиною 5 міліметрів. На сьогоднішній день більшість телефонів оздоблені NFC міткою за допомогою якої можна оплачувати покупки, тому даний дослід був направлений на

перевірку, чи зможе чохол завадити сигналу, який проходить через нього (рис. 5.7).

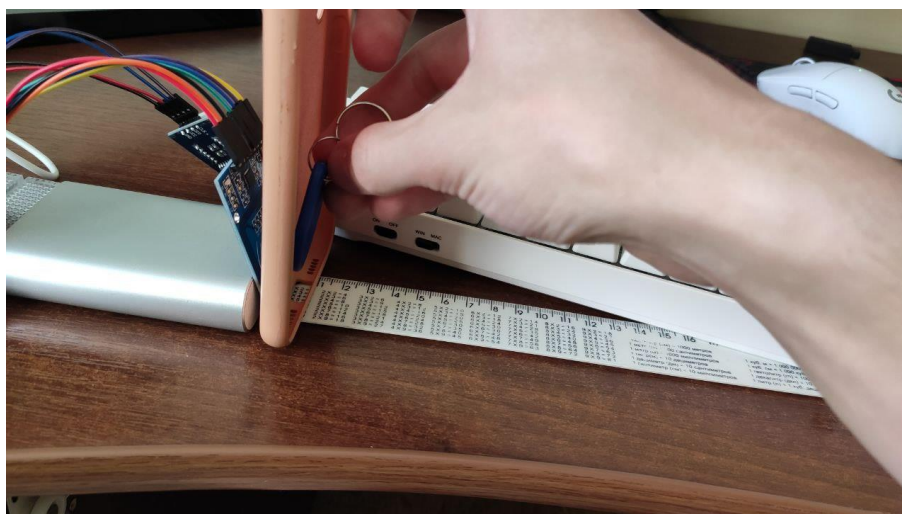


Рисунок 5.7 – Вимірювання шляху зчитування з другою перешкодою.

При проведенні двох експериментів з різними відстаннями, було зібрано дані максимальної дальності зчитування RF – мітки та записано в таблицю 5.1.

Таблиця 5.1 – Максимальна ефективна дальність зчитування RF – мітки.

Відстань в мм	5	8	11	14	17	20	23	27	30
Зчитування мітки без перешкод	+	+	+	+	+	+	+	+	-
Зчитування мітки з перешкодою в 2 мм	+	+	+	+	+	+	-	-	-
Зчитування мітки з перешкодою в 5 мм	+	+	+	+	+	-	-	-	-

За результатами експерименту визначено, що без перешкод максимальна ефективна дальність на якій зчитується RF – мітка рівна 27 міліметрів, з перешкодою в 2 міліметри дальність зчитування 20 міліметрів, а при перешкоді в 5 міліметрів дана відстань рівна 17 міліметрам.

За даними які були добути в ході експерименту був побудований графік максимально ефективної дальності зчитування RF – мітки рисунку 5.8.

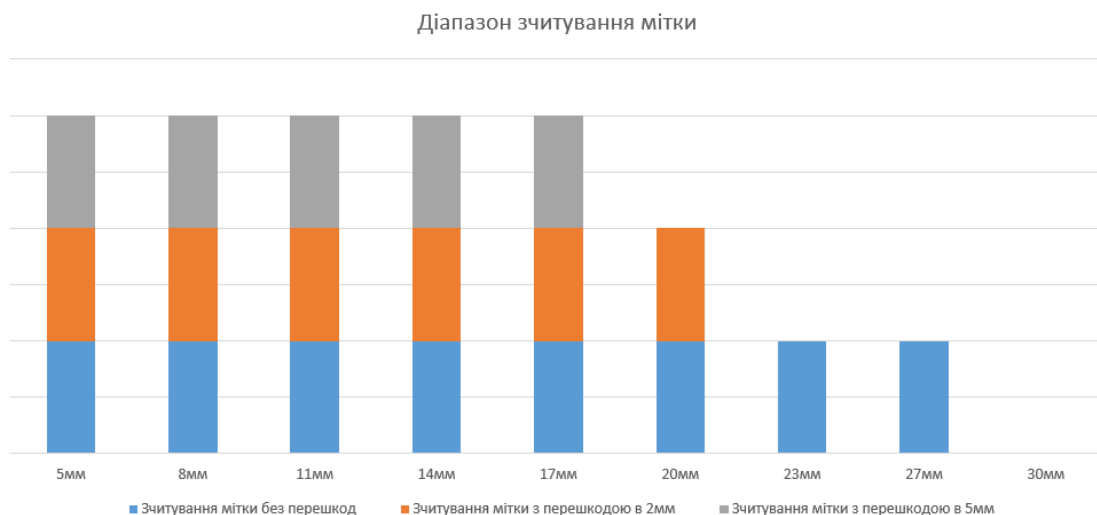


Рисунок 5.8 – Графічна залежність максимально ефективної дальності зчитування RF – мітки

5.4 Експериментальне дослідження мережевих затримок передачі даних

Оскільки система використовує протокол HTTP для передачі даних на сервер, критичним параметром є час затримки між моментом зчитування мітки та моментом фіксації події в базі даних. Для перевірки стабільності роботи системи було проведено серію з 20 тестових запитів у локальній мережі Wi-Fi.

Вимірювання проводилися за допомогою вбудованого в програмне забезпечення таймера (функція `millis()`), який фіксував час відправки запиту (`client.GET()`) та час отримання відповіді від сервера (код 200).

Результати вимірювань затримок (Latency) наведено в таблиці 5.2.

№ виміру	Час затримки (мс)	№ виміру	Час затримки (мс)
1	120	11	118
2	115	12	125
3	140	13	110
4	112	14	135
5	128	15	122
6	130	16	119
7	450 (пік)	17	115
8	118	18	120
9	116	19	114
10	121	20	117

Таблиця 5.2 – Результати вимірювання мережєвих затримок

Середній час затримки становить близько 120–130 мс, що є прийнятним показником для систем фіксації старту аматорського рівня. Поодинокі сплески (наприклад, 450 мс) пов'язані з особливостями роботи протоколу TCP/IP та перевантаженням радіоканалу 2.4 ГГц. Для компенсації цієї затримки в алгоритм сервера введено часову поправку.

ВИСНОВКИ

У магістерській роботі вирішено актуальне науково-прикладне завдання вдосконалення автоматизованої системи реєстрації та подання старт-сигналу з використанням веб-інтерфейсу. У процесі виконання роботи було систематизовано теоретичні знання та здобуто практичні навички проєктування мікропроцесорних систем.

На основі аналізу предметної області обґрунтовано доцільність використання технології радіочастотної ідентифікації (RFID) стандарту HF (13,56 МГц) як найбільш надійного методу фіксації часу. Встановлено, що використання пасивних міток дозволяє усунути суб'єктивну похибку оператора та забезпечити метрологічну простежуваність вимірювань. Теоретичним підґрунтям для організації передачі даних став детальний аналіз стека протоколів TCP/IP та принципів роботи фреймворку Spring, що дозволило ефективно реалізувати северну частину системи.

У конструкторській частині розроблено апаратне забезпечення на базі платформи NodeMCU ESP8266. Спроектовано універсальну підсистему живлення з каскадною стабілізацією напруги (5В та 3.3В), яка дозволяє пристрою працювати як стаціонарно, так і автономно. Для підвищення стабільності радіозв'язку застосовано зовнішні антени через роз'єм IPEX. Проєктування структурної та принципової електричних схем виконано в середовищі автоматизованого проєктування BricsCAD, що дозволило детально опрацювати схемотехнічні рішення.

Програмну реалізацію системи здійснено в інтегрованому середовищі Arduino IDE з використанням мови C++. Розроблено алгоритм на основі кінцевого автомата (State Machine), який включає стани очікування, транзакції та блокування, що забезпечує детермінованість роботи пристрою. Реалізовано

асинхронну передачу даних на сервер у форматі JSON із застосуванням аутентифікації Basic Auth . Також впроваджено цифрову фільтрацію для усунення «брязкоту» контактів.

Експериментальні дослідження підтвердили працездатність прототипу: сумарна метрологічна похибка склала 1,8%, а система продемонструвала стабільність при передачі даних через Wi-Fi. У розділі охорони праці проаналізовано ризики та розраховано захисне заземлення установки.

Виконання магістерської роботи дозволило поглибити фахові компетентності, зокрема навички пайки електронних компонентів, макетування пристроїв на базі мікроконтролерів ESP8266, а також застосування сучасних програмних засобів BricsCAD та Arduino IDE для вирішення інженерних завдань. Оформлення пояснювальної записки та графічної частини здійснено з використанням пакету Microsoft Office 2019.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дорожовець М., Мотало В., Стадник Б. Основи метрології та вимірювальної техніки: підручник. Львів: Видавництво Львівської політехніки, 2005. 532 с.
2. Поліщук Є. С. Методи та засоби вимірювань неелектричних величин: підручник. Львів: Бескид Біт, 2008. 618 с.
3. Finkenzeller K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. 3rd ed. Wiley, 2010. 462 p.
4. Espressif Systems. ESP8266EX Datasheet. Version 6.0. 2019. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf (дата звернення: 15.05.2024).
5. NXP Semiconductors. MFRC522 Standard performance MIFARE and NTAG frontend data sheet. Version 3.9. 2016. URL: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf> (дата звернення: 15.05.2024).
6. ISO/IEC 14443-1:2018. Cards and security devices for personal identification — Contactless proximity objects — Part 1: Physical characteristics. ISO, 2018.
7. Titan Micro Electronics. TM1637 Datasheet (LED Drive Control Special Circuit). Version 2.4. 2011.
8. Schwartz M. Internet of Things with ESP8266. Packt Publishing, 2016. 210 p.
9. Кучерук В. Ю., Володарський Є. Т., Грабко В. В. Метрологія та вимірювальна техніка: навчальний посібник. Вінниця: ВНТУ, 2012. 512 с.

10. Arduino Reference. Офіційна документація середовища розробки та бібліотек. URL: <https://www.arduino.cc/reference/en/>
11. Аронець О. В. Arduino для початківців. — Івано-Франківськ «Симфонія форте» 2018 – 192 с.
12. Arduino Project Handbook: Mark Geddes — USA, 2016 – 275 с.