

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

освітньо-кваліфікаційний рівень

на тему: «Система керування освітлення гаражу
через мобільний застосунок на основі платформи Arduino»

Виконав: студент 2 курсу, групи ІІСТ-21м

Напрямок підготовки:

126 – Інформаційні системи та технології

(шифр і назва напряму підготовки, спеціальності)

Сакун Є.В.

(прізвище та ініціали)

Керівник: доцент каф.АІТ

Кулий Я.А.

(прізвище та ініціали)

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації Кафедра
автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність - 126 Інформаційні системи та технології
Освітньо-професійна програма - Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри АІТ
Бісікало О.В.
_____ 2022 року

ЗАВДАННЯ **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Сакун Євгеній Вікторович

(прізвище, ім'я, по батькові)

1. Тема роботи Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino

керівник роботи Кулик Я.А., доцент каф.АІТ

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

затверджена наказом вищого навчального закладу від “___” _____ 2022 року №__

2. Строк подання студентом роботи: 21 грудня 2022 року

3. Вихідні дані до роботи: робоча система, що функціонує і відповідає вимогам.

4. Зміст текстової частини роботи: Вступ, Аналіз сучасного старну методів і засобів, Розробка технічного і робочого проекту, Розробка системи, Економічний розділ.

5. Перелік графічного матеріалу: Схеми розведеного проладу, та схеми збору елементів обладнання.

6. Консультанти розділів роботи:

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Завдання прийняв
Розділи 1, 2, 3	Кулик Я.А., доц.каф.АІТ		

Економічний розділ	професор кафедри ЕПВМ, професор, к.е.н. Лесько О.Й.		
--------------------	---	--	--

7. Дата видачі завдання _____ 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітки
1	Загальні відомості	Від 01.06.2022 до 01.09.2022	
2	Розробка структурної схеми	Від 01.06.2022 до 01.08.2022	
3	Розробка програмного забезпечення	Від 01.06.2022 до 01.08.2022	
4	Оформлення пояснювальної записки і графічного матеріалу	Від 01.12.2022 до 21.12.2022	
5	Попередній захист	07.12.2022	
6	Остаточний захист	23.12.2022	

Студент _____ Сакун Є.В. _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ Кулик Я.А. _____

АНОТАЦІЯ

УДК 628.94

Сакун Є.В. Система керування освітлення гаражу через мобільний застосунок на основі платформи Arduino. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітня програма – інтелектуальні інформаційні технології. Вінниця: ВНТУ, 2022. 92 с.

На укр. мові. Бібліогр.: 23 назви; рис.: 24; табл. 12.

У магістерській кваліфікаційній роботі розроблено систему керування освітлення гаражу через мобільний застосунок на основі платформи Arduino, який базується на комбінування методів співпраці людини та комп'ютера, що дозволяє значно підвищити зручність. У першому розділі роботи розглянуто теоретичні відомості необхідні для роботи. У другому розділі виконана розробка схеми пристрою. У третьому розділі розроблено систему що відповідає вимогам.

Графічна частина складається з 8 графіків.

У економічному розділі було проведено дослідження доцільності роботи.

Ключові слова: Андроїд, ардуїно, розробка, мосфет, логи.

ABSTRACT

UDC 628.94

Sakun E.V. A garage lighting control system through a mobile application based on the Arduino platform. Master's qualification thesis on specialty 126 - information systems and technologies, educational program - intelligent information technologies. Vinnytsia: VNTU, 2022. 92 p.

In Ukrainian speech Bibliography: 23 titles; Fig.: 24; table 12.

In the master's qualification work, a garage lighting control system was developed through a mobile application based on the Arduino platform, which is based on the combination of human and computer cooperation methods, which allows to significantly increase convenience. The first chapter of the work deals with the theoretical information necessary for the work. In the second section, the development of the device scheme is carried out. In the third section, a system that meets the requirements is developed.

The graphic part consists of 8 graphs.

In the economic section, a feasibility study was conducted.

Keywords: Android, arduino, development, mosfet, logs.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ СУЧАСНОГО СТАНУ МЕТОДІВ І ЗАСОБІВ	11
1.1 Розгляд існуючих платформ для розробки	11
1.2 Опис платформи Arduino Nano	13
1.3 Отримання даних	16
1.4 Програмування Arduino Nano	18
1.5 Мікроконтролер ATMEGA328P	20
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ	21
2.1 Опис та обґрунтування вибору плати для розробки проекту	21
2.2 Опис та обґрунтування вибору структури та методи організації вхідних та вихідних даних	22
2.3 Опис середовища розробки Arduino IDE	23
2.4 Опис середовища розробки Microsoft Visual Studio	25
2.4.1 Елементи управління	27
2.4.2 Відображення і обробка даних	27
2.5 Опис мови C#	28
3 РОЗРОБКА СИСТЕМИ	30
3.1 Загальна розробка системи	30
3.2 Розробка програми для застосунку	31
3.3 Розробка обладнання	34
3.4 Розробка програми для обладнання	40
3.5 Поєднання програми та обладнання	42
3.6 Додаткова розробка	43

	5
4 ЕКОНОМІЧНА ЧАСТИНА	44
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	44
4.2 Розрахунок узагальненого коефіцієнта якості розробки	48
4.3 Розрахунок витрат на проведення науково-дослідної роботи	50
4.3.1 Витрати на оплату праці	50
4.3.2 Відрахування на соціальні заходи.....	54
4.3.3 Сировина та матеріали	54
4.3.4 Розрахунок витрат на комплектуючі	56
4.3.5 Спецустаткування для наукових (експериментальних) робіт	57
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт	58
4.3.7 Амортизація обладнання, програмних засобів та приміщень.....	59
4.3.8 Паливо та енергія для науково-виробничих цілей	60
4.3.9 Службові відрядження	62
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації	62
4.3.11 Інші витрати	63
4.3.12 Накладні (загальновиробничі) витрати	63
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	65
4.5 Висновки до розділу.....	69
ВИСНОВОК.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72
Додаток А. (обов'язковий)	74
Технічне завдання	74

	6
Додаток Б. (обов'язковий) Лістинг програми	77
Додаток В. (обов'язковий) Графічна частина	82

ПЕРЕЛІК СКОРОЧЕНЬ УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

ШИМ - широтно-імпульсна модуляція

AVR - сімейство восьмибітних мікроконтролерів фірми Atmel

USB - Universal Serial Bus - універсальна послідовна шина

C++ (читається сі-плюс-плюс) - здатний до компіляції, статичний типізована мова в програмуванні загального призначення

C# (промовляється сі шарп) - об'єктно-орієнтована мова програмування

Arduino - торговий бренд апаратно-програмних засобів для побудови простих систем автоматички

Умовні позначення:

В – Вольт

кОМ – кілоОм (10³ Ом)

мА – міліампер (10⁻³ А)

мкА – мікроампер (10⁻⁶ А)

гПа – гектоПаскаль (10² Па)

ВСТУП

Arduino - апаратна обчислювальна платформа, основними компонентами якої є плата вводу/виводу та середовище розробки на мові Processing/Wiring. Arduino може використовуватися як для створення автономних об'єктів автоматики, так і підключатися до програмного забезпечення на комп'ютері через стандартні дротові і бездротові інтерфейси. Arduino складається з мікроконтролера Atmel, а також елементів обв'язки для програмування та інтеграції з іншими пристроями. На багатьох платах наявний лінійний стабілізатор напруги +5В або +3,3В. Тактування здійснюється на частоті 84, 16 або 8 МГц кварцовим резонатором. У мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен. Мікроконтролер являє з себе мікропроцесор з пам'яттю і різними периферійними пристроями, реалізований на одній мікросхемі. Фактично це однокристальний мікрокомп'ютер, який здатний виконувати відносно прості завдання. Різні моделі з колекції Arduino оснащені різними мікроконтролерами. Вся робота мікроконтролера синхронізується генератором тактової частоти, який може бути внутрішнім або зовнішнім. Тактова частота, або, більш точно, швидкість шини, визначає, скільки обчислень може бути виконана за одиницю часу. Усі плати програмуються через USB, що можливо завдяки мікросхемі конвертера USB-to-мікроконтролера у зовнішніх схемах. Також існує багато видів зовнішніх плат розширення, які називаються «shields» («щити»), які приєднуються до плати Arduino через штирьові роз'єми. Інтегроване середовище розробки Arduino це багатоплатформенний додаток на Java, що включає в себе редактор коду, компілятор і модуль передачі прошивки в плату. Середовище розробки засноване на мові програмування Processing та спроектована для програмування новачками, не знайомими близько 6 з розробкою програмного забезпечення. Мова програмування аналогічна мові Wiring. Строго кажучи, це C ++, доповнений деякими бібліотеками. Програми обробляються за допомогою

препроцесора, а потім компілюється за допомогою AVR-GCC. Тема дипломного проекту: «Програмування датчика тиску на базі android ». Метою дипломного проекту є розробка пристрою на базі плати Arduino та на базі android, додаток, що дасть змогу аналізувати данні тиску, температури та вологості повітря в реальному часі . Цей пристрій дуже зручний у використанні . Створений для використання у різноманітних сферах. Завдяки гнучкій системі налаштування можливо змінювати проміжок часу між даними , що надходять з датчик тиску.

Актуальність роботи полягає в спрощені, та удосконалені використання освітлення сучасними технологіями.

Метою та завданням дослідження є спрощення роботи та взаємодії користувача з системи та можливістю покращення відношення до сучасних технологій.

Об'єктом дослідження є процес створення системи керування освітлення гаражу через мобільний застосунок на основі платформи arduino

Предметом дослідження є методи та засоби функціонування розподілених інформаційних систем управління

Наукова новизна одержаних результатів:

- Розроблено альтернативний варіант взаємодії користувача з системою освітлення
- Встановлено список першочергових функцій необхідних для роботи системи

Практичне значення одержаних результатів полягає в тому, що на основі отриманих теоретичних положень розроблено систему, що дозволяє вдосконалювати та автоматизувати процеси даної сфери діяльності

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням усіх методів та засобів функціонування інформаційних систем, вживаючи Agile методику при розробці системи та безперебійною роботою розроблено програмного забезпечення.

Особистий внесок здобувача. Усі результати та роботи отримано автором самостійно.

1 АНАЛІЗ СУЧАСНОГО СТАНУ МЕТОДІВ І ЗАСОБІВ

1.1 Розгляд існуючих платформ для розробки

Розробка проекту може вестись на трьох платформах: Arduino, BeagleBone, RaspberryPi. Всі три моделі обрані мною для порівняння зовсім не випадково: вони всі досить доступні, знаходяться в компактному розмірі та можуть використовуватися для створення різних пристроїв цифрового призначення. Платформу на базі Arduino можна вважати одним із основних компонентів в товаристві радіоаматорів. Наразі плати Arduino є в доступі в різних форм-факторах (рис.1.1), та з різним набором периферичних пристроїв. Велика частина яких виконана на 8-розрядному мікроконтролері компанії під назвою Atmel. Головною перевагою плат на базі Arduino є просте проектування та збірка макетів. Написання програми під мікроконтролер здійснюється через спеціальну консоль, яка є середовищем розробки і терміналом послідовного порту. Завантаживши написаний код до пам'яті мікроконтролера, кнопкою на платі що призначена для цього (або командою «логічна одиниця») на обраний цифровий порт вводу / виводу, запускається програмний код на циклічне виконання.

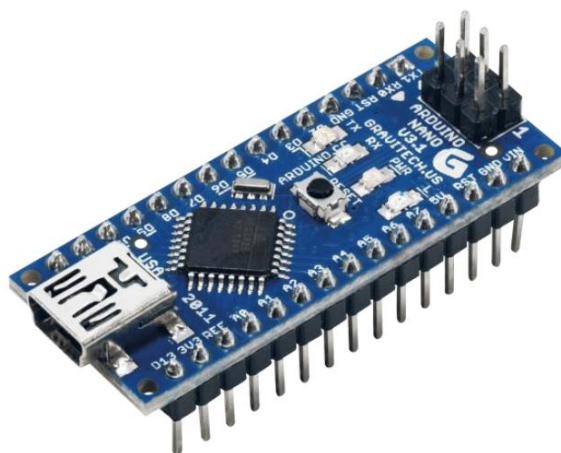


Рисунок 1.1 – Платформа Arduino Nano

Плата на базі Raspberry Pi (рис. 1.2) є новинкою на ринку. Незважаючи на дуже скромний вигляд плати і доволі низьку вартість на ринку, платформа являє собою окремий комп'ютер, що використовується для розробки великої кількості проектів. Плата може використовувати SD-карту як пристрій для зберігання прикладного та системного програмного забезпечення. З можливістю підключення досить широкого асортименту всіляких інтерфейсів, зокрема, таких як, мережа Ethernet, 3G, різного роду дисплеїв. Плата побудована на базі мікропроцесора ARM1176JZFS з тактовою частотою 700 мегаГерц.

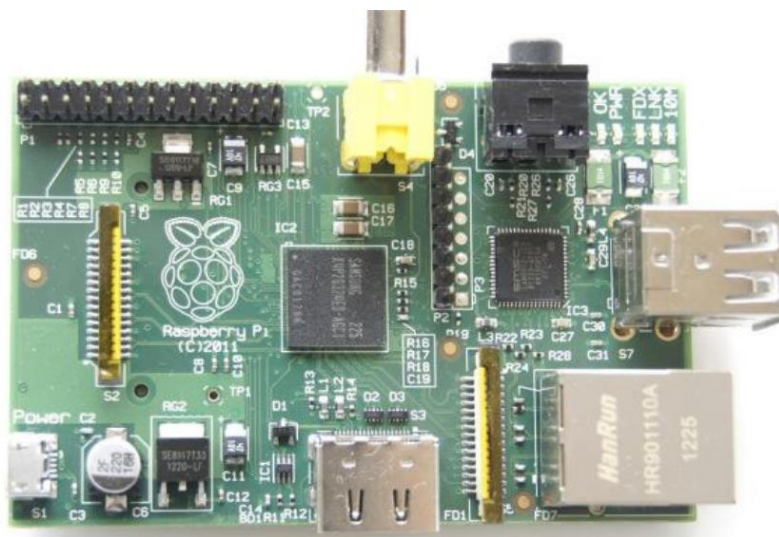


Рисунок 1.2 – Одноплатний комп'ютер Raspberry Pi

BeagleBone (рис. 1.3) являє собою досить потужний Linux-комп'ютер досить компактних розмірів, що підтримує роботу на операційних системах Android і Ubuntu. В основі плата досить подібна до плати Raspberry Pi, однак відрізняється можливістю проводити більші обчислення і має більші можливості щодо розширення. Її система побудована на базі обчислювального процесора TI Sitara AM335x з ядром A8 Cortex, який може використовуватись навіть для роботи потужних смартфонів та планшетів.

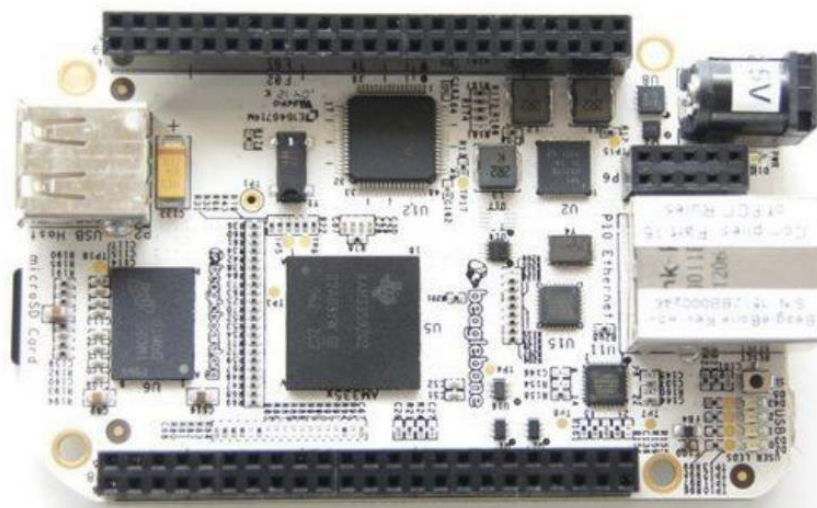


Рисунок 1.3 – Одноплатний комп'ютер BeagleBone

Arduino являє собою досить функціональну та гнучку платформу для розробки вбудованих додатків з величезними можливостями для взаємодії із зовнішнім світом. Платформа прекрасно, підходить для вивчення можливостей мікроконтролерів і може використовуватись як основа для невеликих проектів та втілювати мрії. Raspberry Pi є оптимальною для додатків, для яких наявність дисплея є критичною. Що мають графічне призначення для користувача, взаємодії з різного роду датчиками. При роботі з платформом на базі BeagleBone ідеальною є поєднання Arduino та продуктивності процесора плати на базі Raspberry Pi і ОС Linux. При наявності достатньої кількості входів / виходів, BeagleBone надає можливість легкого підключення до мережі і надає можливість реалізувати web-сервер [1].

1.2 Опис платформи Arduino Nano

Я буду використовувати плату на базі Arduino Nano (рис. 1.4) для виконання свого дипломного проекту. Плата на базі Arduino побудована з мікроконтролера Atmel AVR (ATmega328P), а також навколишніх елементів для можливості програмування програми та інтеграції її з іншими схемами. Багато

плат включають в себе лінійний вирівнювач напруги + 5В або +3,3В. Генерація частоти 16 або 8 МГц виконується за допомогою кварцового резонатора (в деяких версіях керамічний резонатор). Мікроконтролер має попередньо прошитий завантажувач BootLoader, тому нема необхідності у зовнішньому програматору. Концептуальномий рівень плати програмується через RS-232 (послідовне з'єднання), але впровадження даного способу відрізняється від версії до версії плати. Платформа на базі Arduino Uno використовує Atmega8 в якості мікроконтролера в SMD-корпусі. Даний спосіб виконання дає можливість програмувати конвертер, аби платформа визначалася як миша, джойстик або інший пристрій на розсуд розробника з усіма необхідними додатковими сигналами управління.

Плата на базі Arduino Nano може отримувати живлення від нерегульованого 6-20 В або через підключення Mini-USB, або регульованого 5 В, зовнішнього джерела живлення. При роботі плата автоматично вибирає джерело з найвищою наявною напругою. Плата на базі мікроконтролера ATmega168 має 16 кБ флеш-пам'яті для зберігання програми, а плата на базі мікроконтролера ATmega328, в свою чергу, має 32 кБ (в обох випадках 2 кБ пам'яті зарезервовано під завантажувач). ATmega168 має 1 кБ ОЗУ і 512 байт EEPROM, а ATmega328 - 2 кБ ОЗУ і 1 Кб EEPROM [1].

Із 14-ти доступних цифрових виходів Nano, за допомогою функції pinMode(), digitalWrite (), і digitalRead (), можна запрограмувати як вхід або вихід. Виходи працюють за напруги 5 В. Вивід має навантажувальний резистор (стандартно відключений) 20-50 кОм та може проводити через себе до 40 мА. У деяких виходах доступні особливі функції:

Послідовна шина: 0 (RX) і 1 (TX). Дані виходи можуть використовуватись для передачі (TX) та отримання (RX) даних TTL. Ці виходи підключені до відповідних виходів мікросхеми послідовної шини FTDI USB-to-TTL. Обидва з виходів дає можливість ШІМ з роздільною здатністю 8 біт за допомогою функції analogWrite (). За допомогою даних виходів може здійснюватись зв'язок SPI,

який, хоча і підтримується за допомогою апаратної частини, але не включений в мову Arduino.

LED: 13. Це вбудований світлодіод, що має підключення до цифрового виводу 13. Світлодіод горить якщо значення на виході має не нельове значення.

На Arduino Nano встановлено 8 аналогових входів, кожен має дозвіл в 10 біт (що означає що він може приймати 1024 різних значень). Стандартний діапазон висновків прирівнюється до 5 В щодо землі, але є можливість змінити верхню межу значення використовуючи функції `analogReference()`. Також, доступні виводи, що мають додаткові функції: I2C: A4 (SDA) і A5 (SCL). За допомогою таких виводів здійснюється зв'язок I2C (TWI).

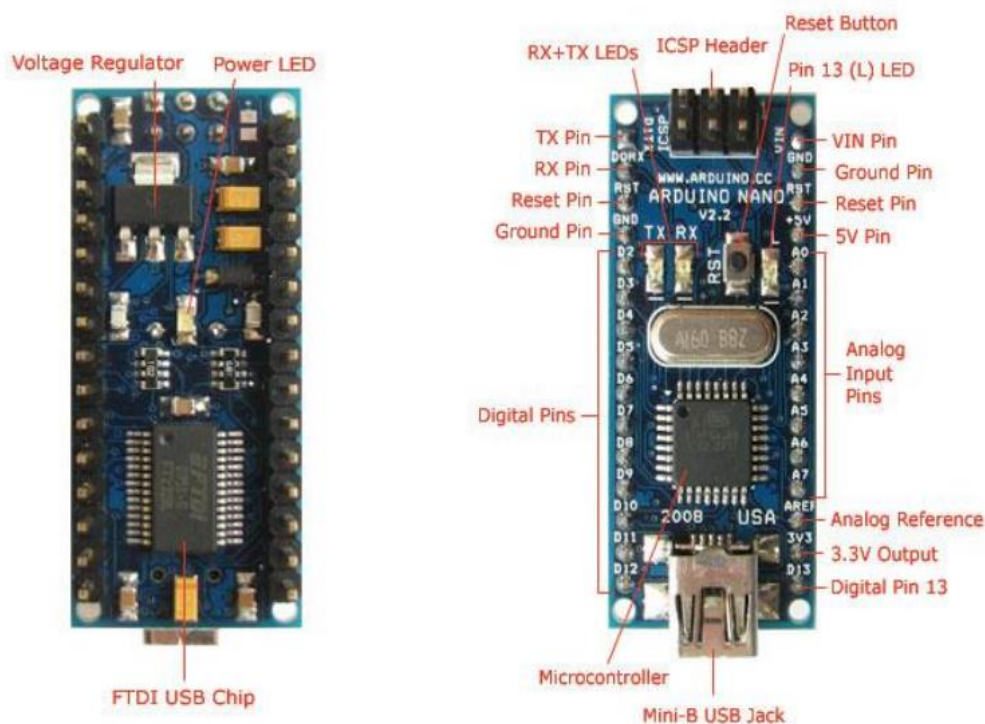


Рисунок 1.4 – Входи і виходи на платі Arduino Nano

Програміст не зобов'язаний прописувати назви використовуваних стандартних (автоматично підключених) бібліотек в текст програми (скетчу). Передпроцесор Arduino автоматично додасть необхідні заголовки відповідно до конфігурації проекту. Потрібно вказувати призначені для користувача

бібліотеки. В менеджера проекту Arduino IDE нестандартний механізм додавання бібліотек. Бібліотеки додаються в спеціальну папку в робочому каталозі IDE у вигляді вихідних текстів на стандартному C ++. Але назва бібліотеки буде додана до списку бібліотек в меню IDE. Програміст сам визначає необхідні бібліотеки і система вносить їх до списку компіляції.

1.3 Отримання даних

Барометричний датчик тиску BMP180 призначений для вимірювання барометричного, абсолютного, диференціального, надлишкового тиску, а також значення температури навколишнього середовища. Для використання датчика тиску спочатку потрібно зібрати на його основі макет, попередньо підключивши висновки управління і харчування. Потім датчик тиску потрібно відкалібрувати за допомогою спеціальних програм. Також для нормальної роботи датчика з Arduino контролерами потрібна спеціальна бібліотека.

Після виконання вищеописаних дій можна приступати до роботи. Практичне застосування інтегрується в GPS навігатори, смартфони, прилади і апарати, які переміщуються в атмосфері, вимірювання тиску всередині і зовні, спорт, для визначення погоди, для барометричних вимірювань висоти. На корпусі датчика є один отвір для закріплення його на будь-якій поверхні і в будь-якому положенні.

Управління датчиком здійснюється або від Arduino контролера, або від іншого мікропроцесорного керуючого пристрою. Чутливим елементом датчика є мембрана в корпусі, яка працює по п'єзорезистивного принципом. П'єзорезистивного ефект - залежність опору матеріалу від величини його деформації.

Барометричний датчик тиску BMP180 має одну колодку з 5 клемами:

- VCC - напруга живлення;
- SDA - ІІС інтерфейс;
- SCL - ІІС інтерфейс;

- GND - загальний висновок.



Рисунок 1.5 – Барометричний датчик тиску BMP180

Виміряні величини тиску і температури передаються по послідовній шині даних цифрового барометра-термометра. Висота над рівнем моря обчислюється за методикою, викладеною в документації мікросхеми BMP180.

Кількість вимірювань в секунду може досягати 128. Зазвичай використовують датчик в режимі один вимір в секунду. Це збільшує точність і економить струм споживання. Датчик може вимірювати атмосферний тиск з різною точністю. Необхідна точність вимірювання повідомляється датчику мікроконтролером. Важливо правильно задати затримку в програмі перед читанням регістрів даних.

Характеристики:

- Напруга живлення: 3.3 або 5 Вольт
- Струм: 5 мкА при швидкості опитування 1 Герц
- Струм: 0.1 мкА в режимі очікування

- Діапазон вимірювання тиску: від 300 до 1100 гПа (від -500 від +9000 метрів над рівнем моря)
- Шум: 0.06 гПа (0.5м) в грубому режимі (ultra low power mode) і 0.02 гПа (0.17м) в режимі максимального дозволу (advanced resolution mode)
- Діапазон вимірювання температури: від 0 до 65 ° C
- Точність вимірювання температури: ± 2 ° C
- Роздільна здатність: 16 біт
- Максимальна швидкість інтерфейсу: 3,4 Мбод
- Робоча температура: від -40 ° C ~ +85 ° C
- Діаметр монтажного отвору: 3 мм
- Розміри: 12 x 10 мм
- Вага: 1.18 гр.

1.4 Програмування Arduino Nano

Вбудоване середовище розробки Ардуїно - це багато платформенний додаток на Java, що дає можливість редагувати та писати код, а також компілювати його і передавати на плату. Середовище розробки засноване на мові програмування Processing.

Мова що використовується для написання програми аналогічна до мови Wiring. Дуже схожа до C++, що доповнений деякими зручними бібліотеками. Програми обробляються за допомогою препроцесора, а потім компілюється за допомогою AVR-GCC.

Програми Arduino пишуться на мові програмування C або C++. Середовище розробки Arduino поставляється разом із бібліотекою програм, яка називається «Wiring», яка бере початок від проекту Wiring, який дозволяє робити багато стандартних операцій вводу/виводу набагато простіше [2].

Середовище розробки Arduino складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виведення тексту (консолі), панелі інструментів з кнопками часто використовуваних команд і

декількох меню. Для завантаження програм і зв'язку середовище розробки підключається до апаратної частини Arduino.

Програма, що написана в середовищі Arduino, називається скетч. Скетч програмується в текстовому редакторі, що має інструменти для вирізки / вставки, пошуку заміни тексту. Під час збереження та експорту проекту в області повідомлень з'являються інформативні повідомлення, також можуть відобразитися виникли помилки. Вікно виведення друкованого тексту (консоль) показує повідомлення Arduino, що включають повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевірити та надіслати програму на пристрій, створити, відкрити і зберегти скетч, відкрити та переглядати моніторинг послідовної шини.



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help

Blink

/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);              // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000);             // waits for a second
}

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)

22
```

Рисунок 1.6 – Arduino IDE з прикладом простої програми

1.5 Мікроконтролер ATMEGA328P

ATMEGA328P є восьми бітовим мікроконтролером на базі ядра AVR з архітектурою RISC, розроблений компанією Atmel. Мікроконтролер працює максимальною тактовою частотою 20MHz. Пристрій містить статичну оперативну пам'ять SRAM розміром 2KB, незалежну пам'ять EEPROM розміром 1024B, Flash пам'ять на 32KB. ATMEGA328P має шести каналний 10-bit A / D конвертер, програмований сторожовий таймер з внутрішнім генератором, двох провідний послідовний інтерфейс SPI з послідовним портом, внутрішні і зовнішні переривання, 32 регістра загального призначення, 3 гнучких таймера / лічильника з функціями порівняння. У набір можливостей входять: 23 лінії введення виводу загального призначення, а також програмне забезпечення з п'ятьма програмованими режимами енергозбереження [3].

Основні характеристики:

- Ядро: AVR;
- Ширина шини даних: 8 bit;
- Максимальна тактова частота: 20MHz;
- Розмір програмної пам'яті: 32KB;
- Розмір ОЗУ даних: 2KB;
- Розмір ПЗУ даних: 1KB;
- Тип інтерфейсу: 2-Wire, SPI, USART;
- Максимальна робоча температура: + 85 °C;
- Мінімальна робоча температура : -40 °C;
- Кількість програмованих входів / виходів: 23;
- Тип пам'яті програм: Flash;
- Максимальна напруга живлення: 5.5В;
- Мінімальна напруга живлення: 1.8В.

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Опис та обґрунтування вибору плати для розробки проекту

Мій вибір плати для розробки проекту зупинився на платформі ArduinoNano. Головним конкурентом у виборі для Arduino була платформа Raspberry Pi. За тактовою частотою Raspberry Pi в 40 разів швидше, ніж Arduino. Ще більше розбіжність у оперативній пам'яті: Raspberry Pi має в 128000 разів більше оперативної пам'яті ніж Arduino. Raspberry Pi є комп'ютером, на якому може бути запущена операційна система Linux, що підтримує багатозадачність. До USB-портів можна підключати різні пристрої, наприклад для бездротового підключення до мережі інтернет. Загалом ця крихітна плата є досить потужною і може функціонувати як повноцінного комп'ютера (хоча, звичайно з сучасними персональними комп'ютерами і ноутбуками конкурувати за потужністю Raspberry Pi не може). Може скластися враження, що Raspberry Pi перевершує Arduino. Але ця перевага скоріше в програмному забезпеченні. Для чисто апаратних проектів набагато краще підходить Arduino. Таку перевагу Arduino має за рахунок своєї простоти. Arduino здатна краще ніж Raspberry Pi в реальному часі зчитувати аналогові сигнали. Ця гнучкість дозволяє Arduino працювати практично з будь-яким видом датчиків або чіпів. Raspberry Pi не така гнучка, наприклад, для читання аналогових датчиків потрібні додаткові апаратні засоби. Arduino менш вимоглива до живлення. Так, рекомендоване живлення для Arduino 7-12 Вольт, напруга стабілізується до 5 Вольт. А плата Raspberry Pi вимагає строго 5 Вольт на вході, тому для роботи з нею не обійтися без фільтра живлення зі струмом 1А. Arduino може працювати з будь-яким комп'ютером і може працювати від батареї [4]. Arduino IDE значно простіше у використанні, ніж Linux. Наприклад, якщо ви хочете написати програму для миготіння світлодіодом з Raspberry Pi, ви повинні встановити операційну систему і деякі бібліотеки коду - і це тільки початок. Оскільки Arduino не призначений для запуску операційної системи або безлічі додатків, ви можете просто підключити

плату і почати роботу. Raspberry Pi дозволяє використовувати переваги багатозадачності, кілька програм у фоновому режимі можуть працювати одночасно. Наприклад, Raspberry Pi, може працювати і як сервер друку, і як сервер VPN одночасно. З іншого боку, ви можете залишити Arduino, що виконує будь-який процес підключеною протягом тривалого часу, і просто відключити плату потім, коли ви вирішили. У цьому плані Arduino простіше, її складніше пошкодити. Arduino працює з будь-яким комп'ютером і може працювати від батареї. Arduino можна включити і благополучно вимкнути в будь-який час. Операційна система на Raspberry Pi може бути пошкоджена, якщо відключити плату без належного завершення роботи. З Raspberry Pi ви повинні знати дещо про Linux, а також про програмне середовище, наприклад, Python. З іншого боку, Raspberry Pi має вигоду від десятиліть програмного забезпечення Linux. Таким чином, для даного проекту буде доцільно використовувати саме плату Arduino, використовуючи потрібні переваги. Великою перевагою є саме те, що плата Arduino може працювати з більшістю датчиків без встановлення додаткового елемента, що зменшує її розміри, підвищує практичність. Іншою досить великою перевагою, є те, що Arduino споживає набагато менше живлення для роботи і може працювати від батареї, та може зчитувати данні з датчиків в режимі реального часу, що є безперечним плюсом для проекту. Через те, що Arduino легше програмувати ніж Raspberry Pi, майже будь-яка людина зможе змінити код програми під свої потреби, можна додати нові датчики, удосконалити старі, простіше кажучи Arduino більш гнучка в налаштуваннях і модернізації ніж її головний конкурент.

2.2 Опис та обґрунтування вибору структури та методи організації вхідних та вихідних даних

Для отримання даних про яскравість освітлення, що потрібно виставити, буде використовуватись блютуз датчик HC-06. У цього датчика є одна перевага - ціна (близько 40 гривень в роздріб). Працює по I2C інтерфейсу, завдяки чому

можна використовувати із будь-якими іншими платами / контролерами, просто підключивши до необхідних пінів. Модуль отримує значення у реальному часі. Значення, які отримав модуль по безпроводній мережі передається на ардуіно.

2.3 Опис середовища розробки Arduino IDE

Середовище розробки Arduino представляє собою текстовий редактор програмного коду, область повідомлень, вікно виведення тексту (консоль), панель інструментів і кілька меню. Для завантаження програм і зв'язку середовище розробки підключається до апаратної частини Arduino.

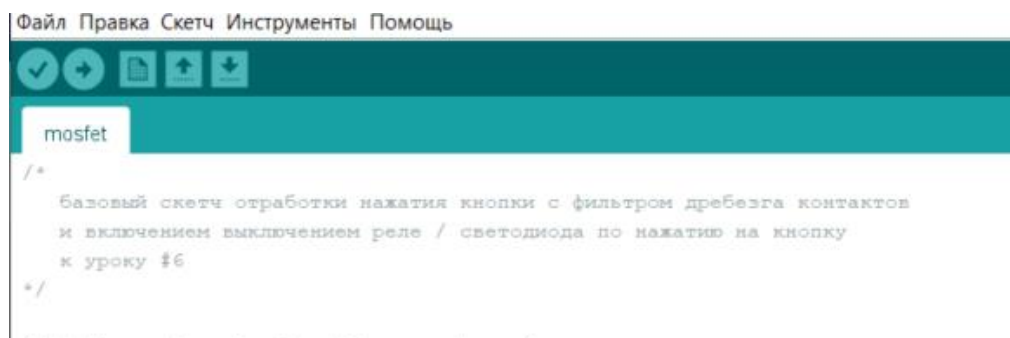


Рисунок 2.1 – Інтерфейс Arduino IDE

Меню редактора має наступні елементи: файл, правка, скетч, інструменти і довідка. Розглянемо докладніше кожен з них.

У меню «Файл» можна знайти команди, що відповідають за створення нової програми, читання старої, збереження її змін, а також команди для завантаження програми на мікроконтролер.

- Створити - створити нову програму (скетч);
- Відкрити - відкрити існуючу програму;
- Папка зі скетчами - відкрити програму із заданої папки;
- Приклади - відкрити приклад програми;
- Закрити – закрити поточне вікно.
- Зберегти - зберегти зміни в попередньо збереженій програмі;

- Зберегти як - зберегти нову програму, із зазначенням імені;
- Завантажити - завантажити програму в Arduino;
- Завантажити за допомогою програматора - завантажити програму за допомогою програматора;
- Налаштування друку - налаштування принтера;
- Друк - виведення на друк коду програми;
- Налаштування – налаштування редактора;
- Вихід - вихід з Arduino IDE.

В меню «Правка» знаходяться команди для роботи з кодом написаної програми. Команди що часто використовуються зручні наявністю комбінацій для швидкого доступу за допомогою клавіатурних клавіш. Також, зручними можливостями є функціонал копіювання для форумів і в html форматі, що дозволяють ділитися вашими скетчами, зберігаючи видимість розмітки у вигляді BB кодів або html розмітки відповідно [5]. У меню «Скетч» розміщуються команди для контролю за процесом компіляції програми.

- Перевірити / Компілювати - компілювати програму;
- Показати папку скетчів - відкрити системну папку з програмами;
- Додати файл - додати до проекту файл з даними або програмою;
- Імпортувати бібліотеку - підключити до програми бібліотеку зі списку встановлених.

Додатково хочеться зупинитися на процесі завантажені бібліотек. Arduino IDE вміщує безліч попередньо завантажених бібліотек. Бібліотеки додають користувачеві додаткову функціональність скетчам, наприклад, при роботі з апаратною частиною або при обробці даних. Одна або кілька директив `#include` будуть розміщені на початку коду скетчу з подальшою компіляцією бібліотек і разом зі скетчем. Завантаження бібліотек вимагає додаткового місця в пам'яті Arduino. Для встановлення сторонніх бібліотек можна скористатися командою «Імпортувати бібліотеку». Меню «Довідка» містить докладний опис всіх функцій самого редактора Arduino IDE, а також команди і прийоми роботи з платформою Arduino.

Пункт меню «Інструменти» включає в себе допоміжні функції для роботи з самим мікро контролером.

- Автоформатування - автоматична розстановка відступів, переносів і т.п.;
- Архівувати скетч - архівація папки з програмою, і збереження;
- виправити кодування і перезавантажити
- Монітор порту - відкрити вікно для обміну даними з мікро контролером;
- Плата - вибір поточної плати;
- Послідовний порт - вибір порту, до якого підключений пристрій;
- Програматор - вибір програм;
- Записати завантажувач - запис програми загрузчика в мікроконтролер.

1. Меню іконок.

2. Вкладки. Кожна програма для Arduino може складатися з декількох файлів. Для перемикання між цими файлами служить система вкладок у редакторі. Там же, можна створити нову вкладку, і асоціювати з нею файл в папці з проектом [5].

3. Вікно програми. Безпосередньо, текст програми створюється і редагується в головному вікні програми.

4. Вікно редактора являє собою типовий текстовий редактор, з підсвічуванням конструкцій коду у різні кольори.

5. Вікно повідомлень самому низу редактора Arduino IDE є невелике вікно, що служить для виведення повідомлень про проблеми, які виникають в процесі.

2.4 Опис середовища розробки Microsoft Visual Studio

Visual Studio – належить до групи програмного забезпечення корпорації Майкрософт, які надають зручне вбудоване середовище розробки програм та цілий ряд інших інструментів. Це програмне забезпечення дозволяє розробляти як консольні програми, так і програми з графічним інтерфейсом користувача з підтримкою Windows Forms, а також веб-сайти, веб-застосунки, веб-служби в

рідному, і в керованому кодах для всіх платформ, яким надається підтримка від програмного забезпечення корпорації Майкрософт: MicrosoftWindows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NETCompact Framework та Microsoft Silverlight [6].

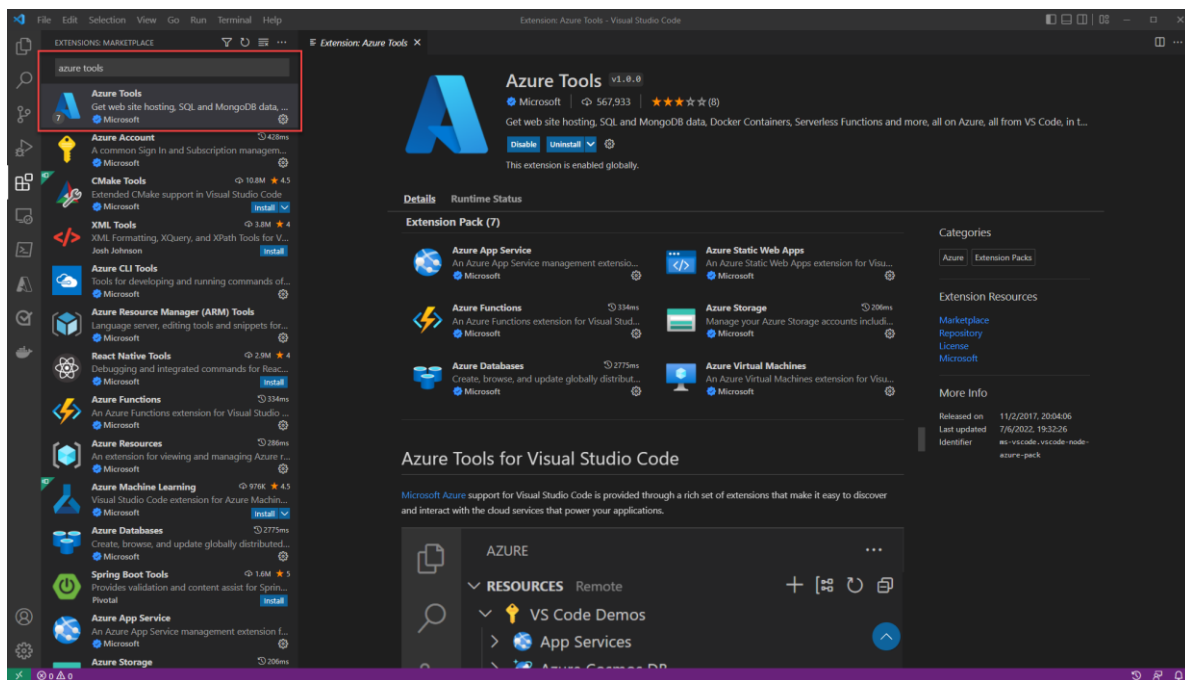


Рисунок 2.2 – Інтерфейс Visual Studio

Windows Forms - це елемент смарт-клієнтів .NET Framework, набір бібліотек для виконання типових завдань, таких як читання і запис в файли. За допомогою середовища розробки, як Visual Studio, можна створювати застосунки Windows Forms, які показують дані, очікують введення даних від користувачів і обмінюються даними з віддаленими комп'ютерами по мережі. У Windows Forms форми - це видиме вікно, на якому показується інформація для користувача. Windows Forms будується шляхом приміщення елементів керування у формах і написання коду для реагування надії користувача, такі як клацання мишею або клавішею. Елемент керування - це елемент призначеного для користувача інтерфейсу для відображення або введення відомостей користувача [6].

2.4.1 Елементи управління

Windows Forms включає широкий набір елементів керування, що розміщуються на формах: елементи керування, що відображають текстові дані (текстові поля), кнопки, списки, що розкриваються, перемикачі та навіть веб-сторінки. Список всіх елементів керування, які використовуються в формах, представлені в розділі «Елементи керування» для використання в Windows Forms. Якщо існуючий елемент керування не має все необхідне, в Windows Forms можна створювати власні елементи керування через UserControl. До Windows Forms входять багатофункціональні елементи користувацького інтерфейсу, які дозволяють відтворювати можливості таких складних застосунків, як Microsoft Office. За допомогою ToolStrip і MenuStrip елемента керування, можна створити панелі інструментів і меню, що відображають текст, зображення і інші елементи керування, такі як текстові поля і поля зі списком. За допомогою конструктора і перетягнень форм Visual Studio можна легко створювати додатки Windows Forms: достатньо виділити елемент керування курсором і перемістити їх в потрібне місце на формі. Конструктор надає лінії сітки і «лінії прив'язки» для подолання складнощів вирівнювання елементів керування. Через Visual Studio або компіляції з терміналу командного рядка, можна використовувати FlowLayoutPanel, TableLayoutPanel і SplitContainer елементи керування для розроблення складних макетів з мінімальними витратами часу і зусиль [7].

2.4.2 Відображення і обробка даних

У багатьох застосунках потрібно відображати дані з бази даних, XML- та JSON- файлів, веб-служби на XML. Windows Forms дає гнучкий елемент керування, що називається DataGridView для відображення таких табличних даних у звичному для людини вигляді рядків і стовпців, де кожен фрагмент даних має власну комірку. З використанням Data-GridView можна налаштувати зовнішній вигляд окремих елементів, зафіксувати координатне положення

рядків і стовпців та відображати складні елементи керування всередині елементів. При застосуванні інтелектуальних клієнтів Windows Forms можна легко підключитися до різних джерел даних по мережі. Компонент Windows Forms BindingSource, що вперше став наявний в Visual Studio 2005 і .NET Framework 2.0, представляє собою підключення до джерела даних і містить методи для прив'язування даних до елементів керування, переходу до попередньої або наступної записи, редагування записів і збереження змін у вихідному джерелі даних. Елемент керування BindingNavigator дає інтерфейс на основі елемента BindingSource для здійснення переходу між записами [8].

2.5 Опис мови С#

С# - об'єктно-орієнтована мова програмування з безпечною системою для типізації на основі платформи .NET. Розроблена Андерсом Гелсбергом, Скотом Вілтамутом та Пітером Гольде під егідою MicrosoftReserch [9].

С++ - мова програмування, що підтримує декілька підходів програмування: об'єктно-орієнтованого, узагальненого та процедурного. Розроблена AT&TBell Laboratories у 1979. Є розширення мови програмування С. У 1990-х роках С++ стала однією з найбільш використовуваних мов програмування. Її застосовують для системного програмування, розробки програмного забезпечення, розробки драйверів, а також великих серверних та клієнтських програм, для розробки компютерних ігор. Дана мова програмування С++ сильно вплинула на інші, більш використовувані в даний час, мови, такі як: С# та Java [10].

С# є об'єктно-орієнтованою мовою, яка використовує і компонентно-орієнтоване програмування. Створення застосунків стає все більш подібне до розроблення програмних компонентів у вигляді автономних і генерованих модулів, що виконують окремі функціональні задачі. Дані модулі - це підхід програмування на основі властивостей, методів і подій. Кожен модуль має атрибути, які визначають декларативні відомості про компоненти, а також

елементи документації, що створюється автоматично на основі опису компонентів. У С# наявні мовні конструкції, які підтримують таку концепцію роботи. Тому С# підходить для розробки і застосування програмних модулів [9].

С# є мовою, що більш безпечніша за С++. У ній використовуються лише безпечними неявні перетворення, такі як розширення цілих чисел. Даний підхід використовується під час компіляції, в тому числі і «на льоту»(JIT) і, в окремих випадках, під час запуску. Неявних перетворень між такими типами даних, як логічні та цілочисельні, а також між елементами типу перерахування і цілочисельними (крім «0», який може бути неявно перетворений в будь-який нумерований тип), не відбувається. Всі користувацькі перетворення повинні позначатись як або явні, або неявні [9, 10].

Незважаючи на технічний прогрес, більшість сучасних комп'ютерів побудовано подібно, що й обчислювальні машини минулого століття. В їх основі покладена архітектура фон Неймана (вчений, що першим описав головні принципи побудови електронних обчислювальних машин). Система команд, або інструкції, для виконання процесором, називається машинною мовою. Для людини ця мова є незручною, оскільки потребує значних знань про побудову апаратної частини комп'ютера й знання особливостей запуску програми. Дані мови застосовуються розробники апаратної частини комп'ютерів і де-які інші спеціалісти [11].

3 РОЗРОБКА СИСТЕМИ

3.1 Загальна розробка системи

Загальна розробка системи складається з прийняття рішень по вирішенню проблем та етапів роботи.

Основним питанням, яке було вирішено в процесі розробки системи є: спосіб надання електроенергії для Arduino і для користувачів.

Для надання електроенергії на плату та користувачам було вирішено використовувати блок живлення що на вихід надає 12 вольт напруги та до 8,3 ампера сили струму. В результаті було зібрано систему що дає можливість надати напругу на обладнання (рис. 3.1).

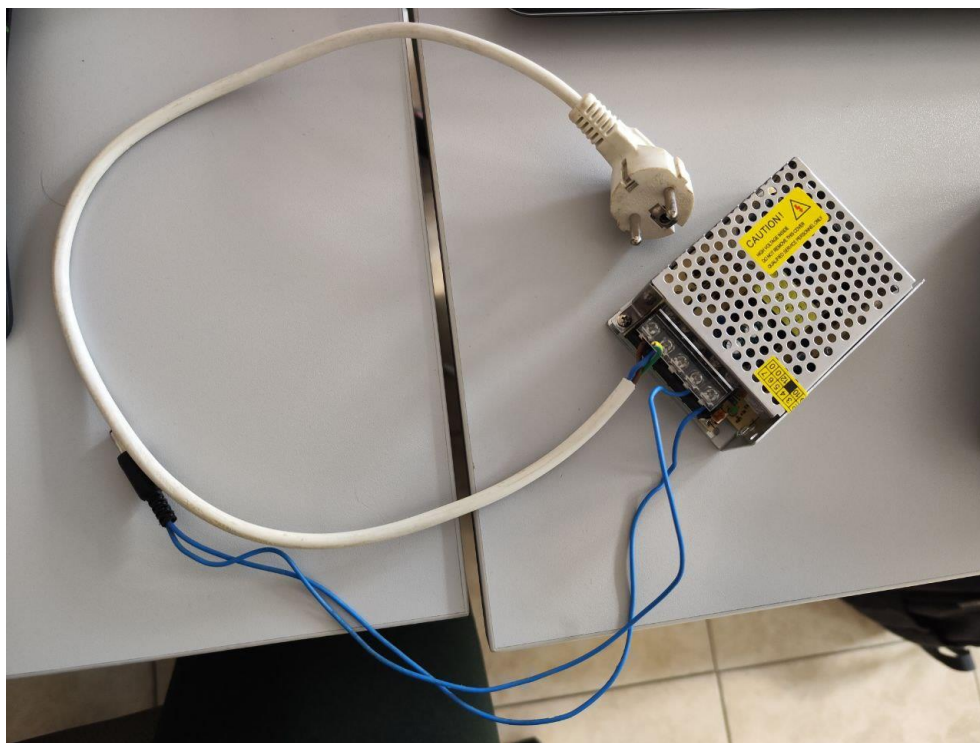


Рисунок 3.1 - Система надання напруги на обладнання

3.2 Розробка програми для застосунку

Розробку програмного забезпечення було проведено згідно поставлених задач.

Програмне забезпечення на платформі андроїд було розроблено та перевірено.

Основними етапами розробки є:

1) Розробка дизайну застосунку. Основним екраном додатку виступає `ControlActivity.kt`, що відображає основний функціонал додатку (рис. 3.2).



Рисунок 3.2 - Основний екран застосунку

До основного функціоналу додатку входить: вибір девайса до якого можна підключитись, кнопка підключення/відключення до вибраного девайсу, можливість вибирати кнопками інформацію, що буде надіслана до обладнання на платформі Arduino. До дизайну також входить екран списку доступних

девайсів до яких можна підключитись за допомогою Bluetooth. За екран вибору девайсу відповідає `BtListActivity.kt`, що відображає список девайсів в якості списку з парами даних (мак адреса девайсу та його назва) (рис. 3.3).

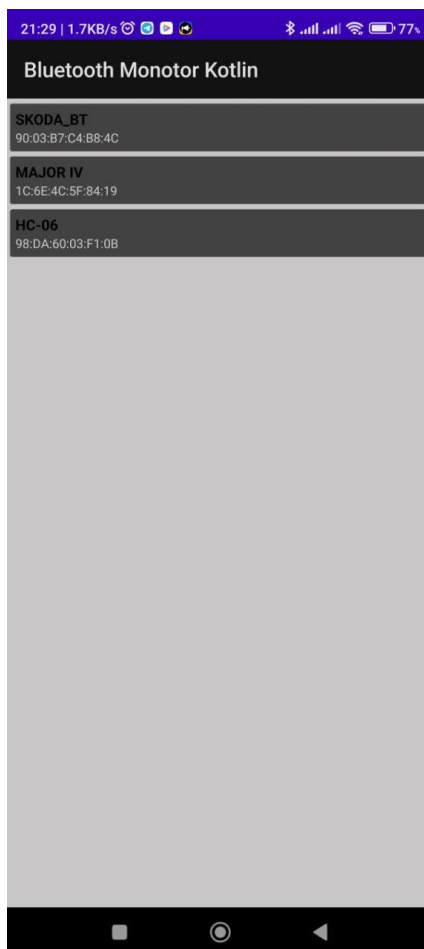


Рисунок 3.3 - Список пристроїв що доступні для підключення

2) Розробка каналу зв'язку. Канал зв'язку побудований на базі стандартного андроїд сервісу, що можна отримати від системи за допомогою методу `Activity.getSystemService(Name)` та передавши йому параметр, необхідний для роботи з Bluetooth – сервіс (`Context.BLUETOOTH_SERVICE`). Даний сервіс є об'єктом типу `BluetoothManager`, та дає можливість отримати доступ до девайса за допомогою інформації про його мак адресу після чого отримати канал зв'язку з девайсом, та можливістю його відкрити для обміну даними. Ці дії відбуваються за допомогою наступного коду:

```

val device = adapter.getRemoteDevice(mac)
    device.let {
        cThread = ConnectThread(it)
        cThread.start()
    }

```

Що в результаті надає потік як канал для зв'язку між застосунком, та девайсом до якого підключився застосунок.

Після приєднання до девайсу є можливість двосторонньої передачі даних. Отримання даних відбувається за допомогою наступного коду:

```

while (true){
    try{
        val size = inStream?.read(buf)
        val message = String(buf, 0, size!!)
        Log.d("MyLog", "Message: $message")
    } catch (i: IOException){
        Log.d("MyLog", "Message error:$Log.getStackTraceString(i)")
        break
    }
}

```

Відправка даних відбувається за допомогою наступного коду:

```

fun sendMessage(byteArray: ByteArray){
    try {
        for (byte in byteArray){
            Log.d("MyLog", "send Message: $byte")
        }
        outputStream?.write(byteArray)
    } catch (i: IOException){
        Log.d("MyLog", "send Message: error" + i.message)
    }
}

```

3.3 Розробка обладнання

Розробку обладнання було розпочато з моделювання схеми що показує зв'язки та способи взаємодії всіх елементів необхідних для роботи.

Було розроблено схему за допомогою платформи EasyEDA Додаток В easyEDA - крос-платформне веб-орієнтоване середовище автоматизації проектування електроніки, що включає редактор принципів схем, редактор топології друкованих плат, SPICE-симулятор, хмарне сховище даних, систему управління проектами, а також засоби замовлення виготовлення друкованих плат.

Середовище працює на основі клієнт-серверної моделі. Клієнтська частина програми повністю виконується у браузері, що підтримує HTML5. Інсталяція будь-яких програм або плагінів не потрібна. Графічна середовище програми використовує двигун векторного графічного формату SVG, доступний у сучасних браузерах [12].

Після чого було розроблено систему на макетній платі, та проведено перевірку системи на працездатність. (рис. 3.4 – 3.8)

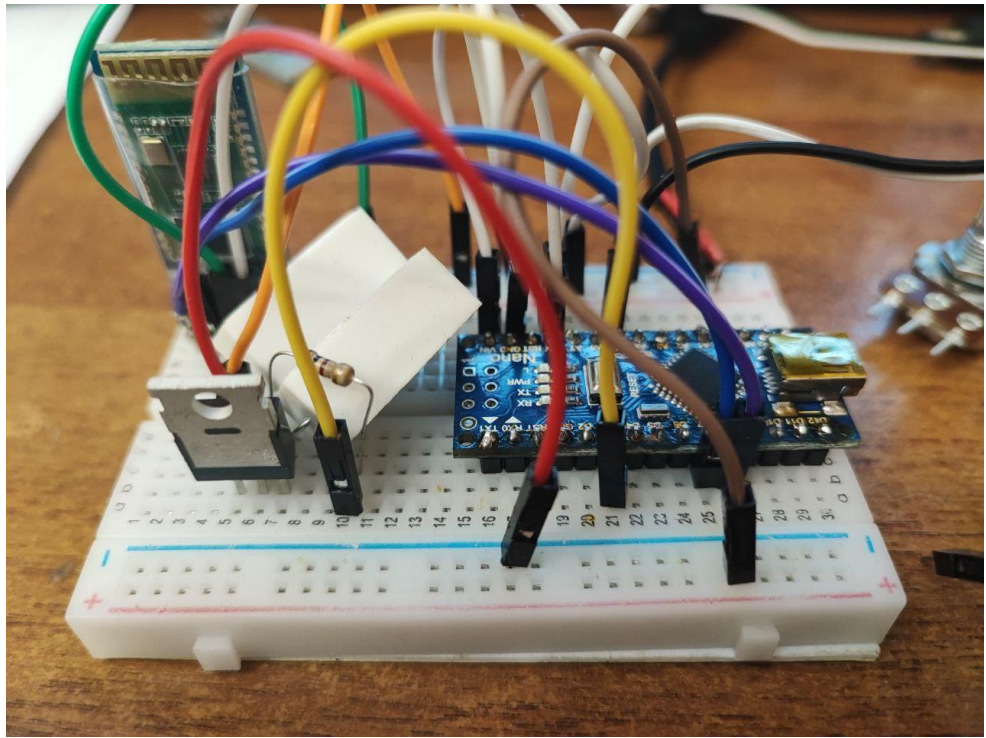


Рисунок 3.4 - Обладнання на макетній платі

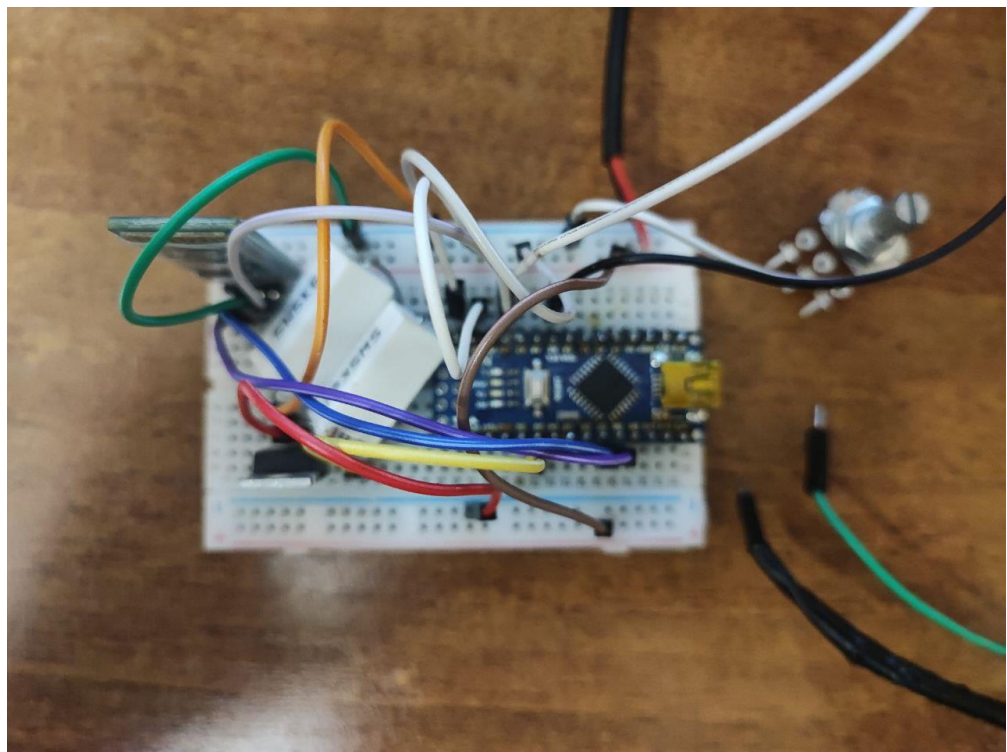


Рисунок 3.5 - Обладнання на макетній платі

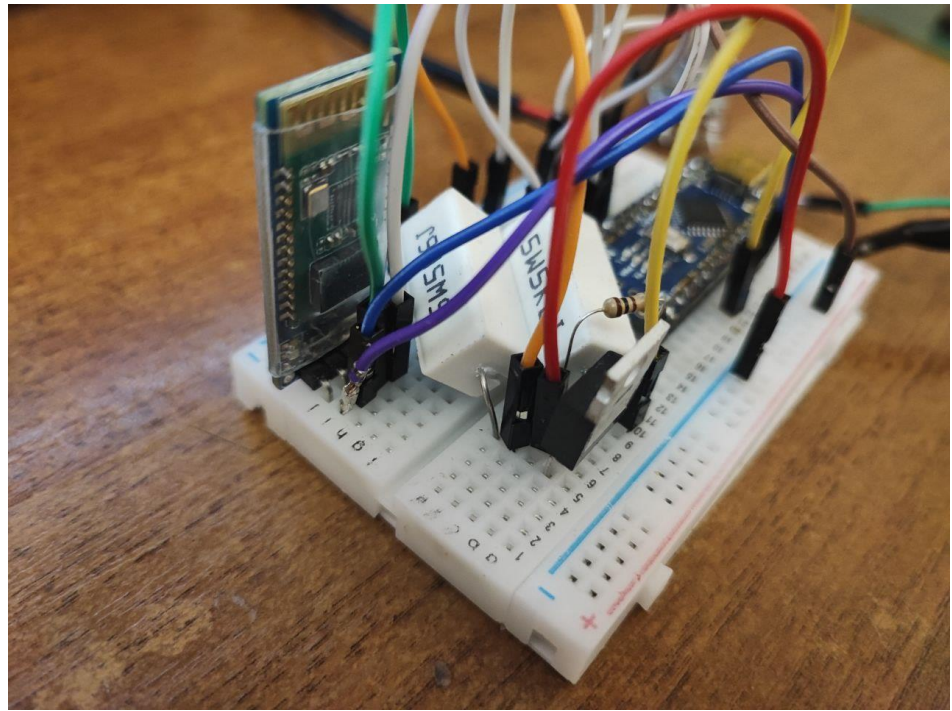


Рисунок 3.6 - Обладнання на макетній платі

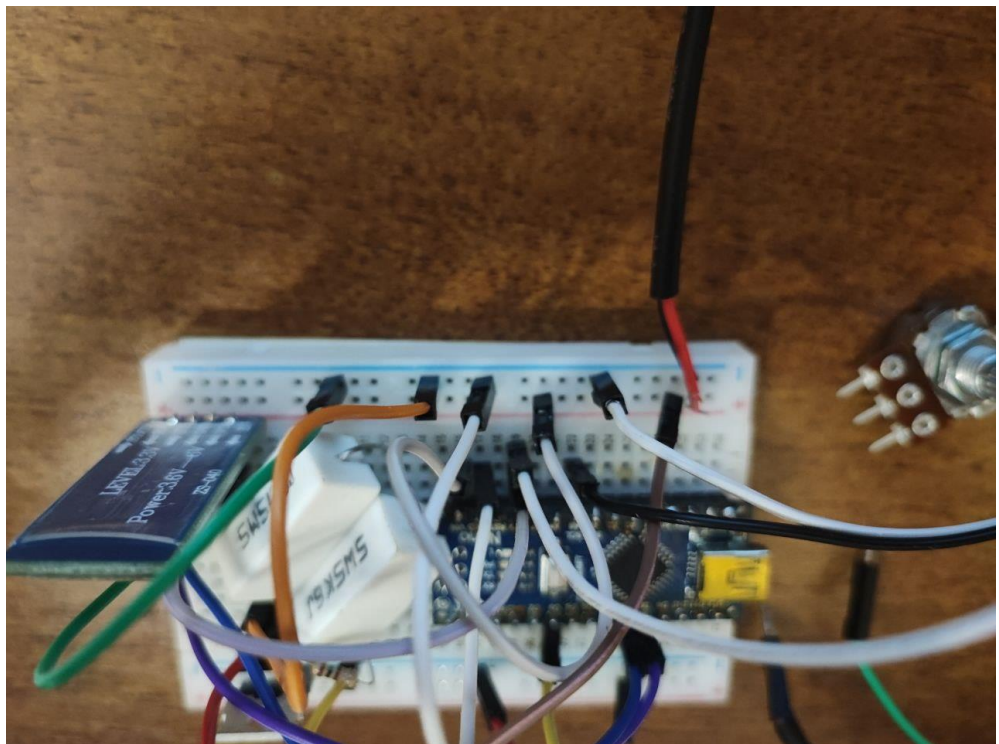


Рисунок 3.7 - Обладнання на макетній платі

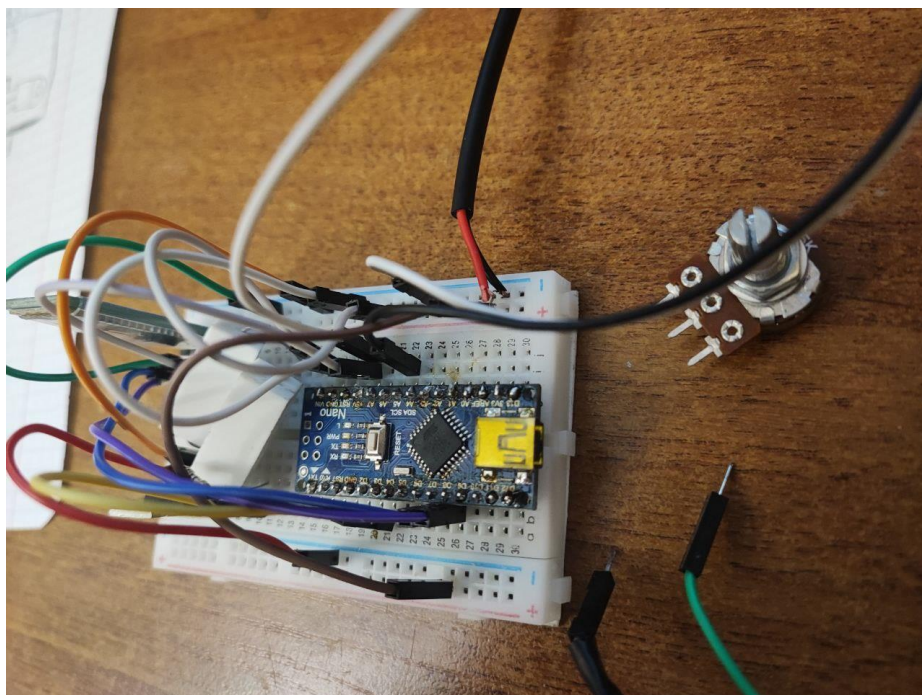


Рисунок 3.8 - Обладнання на макетній платі

Після успішної перевірки системи, запроєктованої, та розробленої на макетній платі, обладнання було перенесено на печатну плату, де було повторно перевірено на працездатність (рис. 3.9 - 3.10).

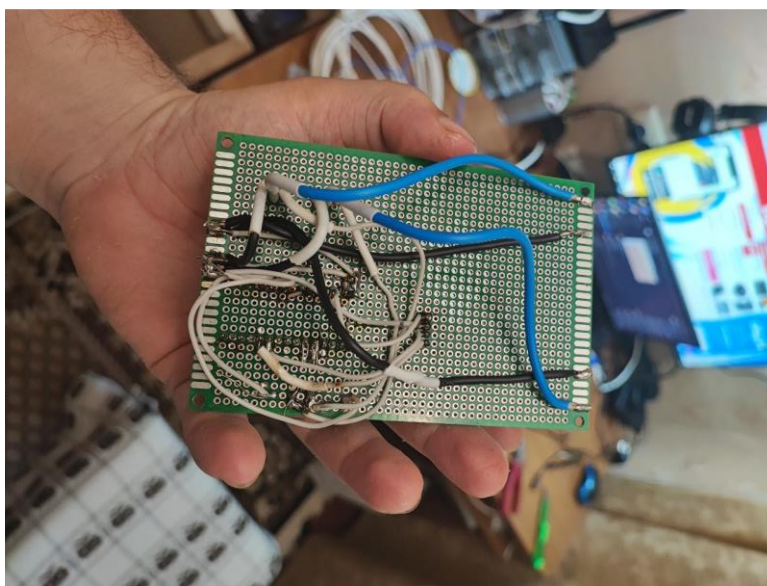


Рисунок 3.9 - Обладнання на печатній платі

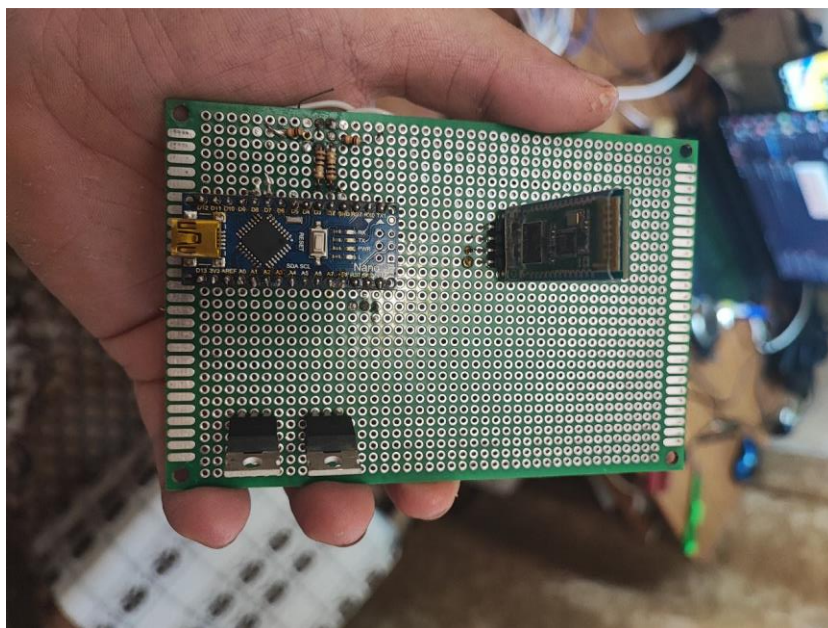


Рисунок 3.10 - Обладнання на печатній платі

Після завершення розробки обладнання на печатній платі, обладнання було розміщено у пластиковий ящик, що був змодельований та роздрукований за допомогою 3д принтера (рис. 3.11 - 3.14).



Рисунок 3.11 - Обладнання у власному корпусі



Рисунок 3.12 - Обладнання у власному корпусі

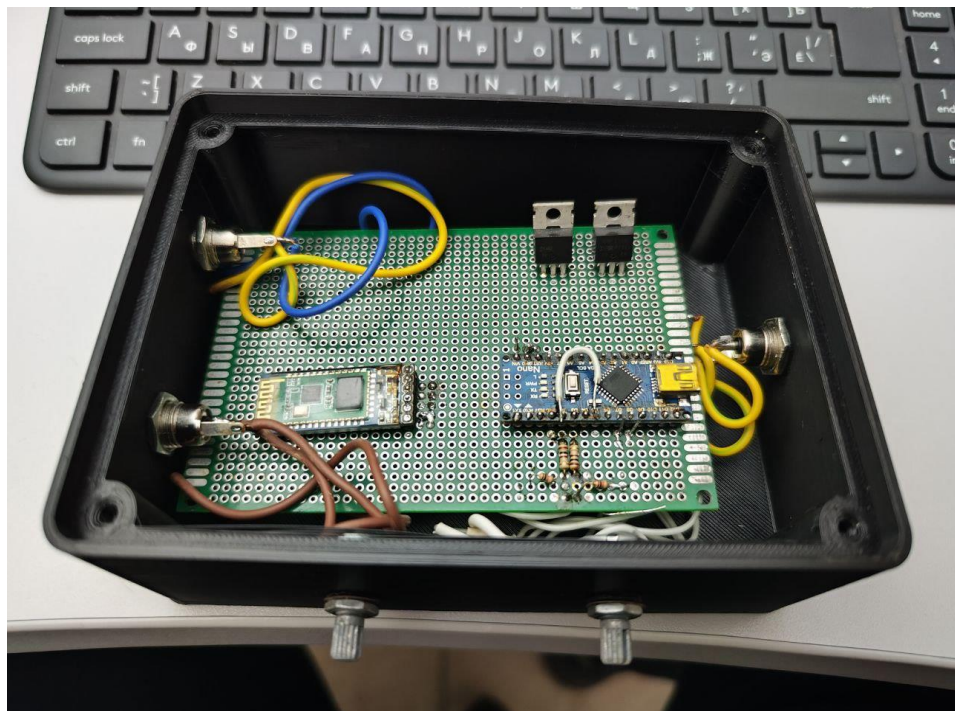


Рисунок 3.13 - Обладнання у власному корпусі

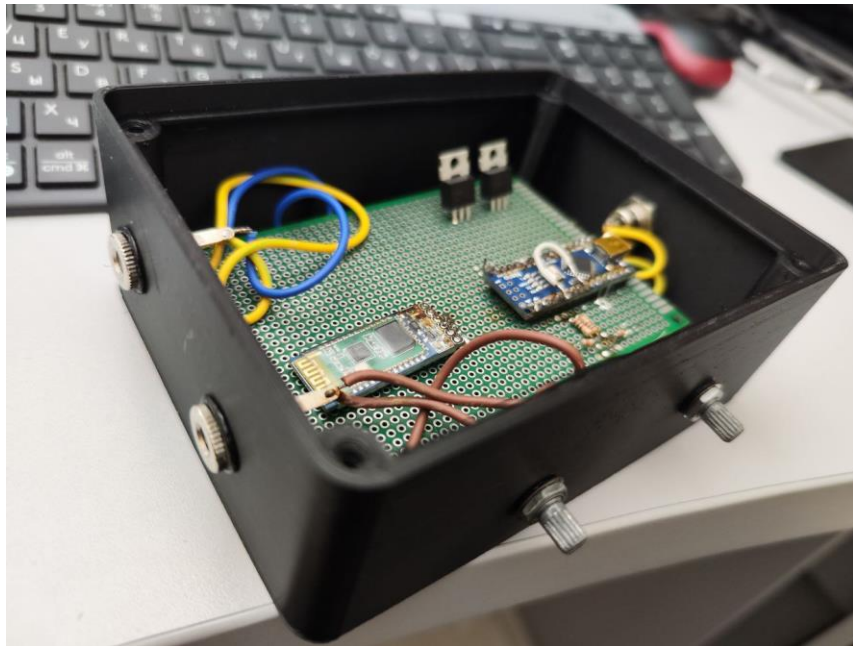


Рисунок 3.14 - Обладнання у власному корпусі

3.4 Розробка програми для обладнання

Програмне забезпечення що було розроблено для обладнання відповідає до вимог поставлених до нього. А саме:

- можливість керування двома лініями навантаження на систему.
- можливість регулювання яскравості освітлення саме процесом широтно-імпульсної модуляції.
- можливість регулювання яскравості освітлення кожної лінії окремо, в залежності від отриманих вхідних даних з резисторів.
- можливість регулювання яскравості загального освітлення на вхідних даних від Bluetooth модуля

Після розробки програмного забезпечення було отримано наступний код:

```
#include "SoftwareSerial.h"

SoftwareSerial BTSerial(7, 6);
int potentValue1 = 0;
int potentValue2 = 0;
```

```
void setup() {
  // put your setup code here, to run once:
  BTSerial.begin(9600);
  Serial.begin(19200);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (BTSerial.available()) {
    int data = BTSerial.read() * 2;
    Serial.println("bluetooth" + data);
    BTSerial.write(data);
    analogWrite(5, data);
    analogWrite(3, data);
  }

  int newPotentValue1 = analogRead(5);
  newPotentValue1 = map(newPotentValue1, 0, 1023, 0, 255);
  if (newPotentValue1 != potentValue1) {
    potentValue1 = newPotentValue1;
    analogWrite(3, potentValue1);
    Serial.println("potent1");
    Serial.println(potentValue1);
  }

  int newPotentValue2 = analogRead(3);
  newPotentValue2 = map(newPotentValue2, 0, 1023, 0, 255);
  if (newPotentValue2 != potentValue2) {
    potentValue2 = newPotentValue2;
    analogWrite(5, potentValue2);
    Serial.println("potent2");
    Serial.println(potentValue2);
  }

  delay(50);
}
```

Код працює належним чином, і відповідає поставленим задачам.

3.5 Поєднання програми та обладнання

Для поєднання двох систем (програми на обладнанні, та застосунку на андроїд) було використано систему логування інформації в логувальну систему андроїда.

Система перевірки поєднання систем має два основних правила та працює за наступними принципами:

- Програмне забезпечення на обладнанні на платформі Arduino, приймає інформацію. Після чого зберігає інформацію, і передає отриману інформацію назад на програмний застосунок, підтверджуючи отримання даних.

- Програмне забезпечення застосунка з можливістю відправляти дані на обладнання з програмним забезпеченням після передачі інформації на обладнання очікує від обладнання інформацію в зворотному напрямку. Після отримання інформації з обладнання виводить в систему логування інформацію про дані що були відправлені та дані що були отримані.

Дана система дозволяє перевіряти вірність отримання та обміну інформації між пристроями (рис 3.15). Також система дозволяє наглядно бачити інформацію що була надіслана, та як обладнання змогло розпізнати та на скільки вірно обробити інформацію що була надіслана.

```

6487-6548 MyLog com.example.bluetoothmonotorKotlin D Connecting...
6487-6548 MyLog com.example.bluetoothmonotorKotlin D Connected
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 0
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 0
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 0
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 0
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 0
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 0
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127
6487-6487 MyLog com.example.bluetoothmonotorKotlin D send Message: 127

```

Рисунок 3.15 - Вивід результатів обміну між пристроями

3.6 Додаткова розробка

Після розробки системи та проектування всіх елементів, було проведено процес розводки печатної плати, для можливості майбутнього вдосконалення за допомогою платформи EasyEDA. Результати роботи наведені на Додатку В

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» є оцінювання науково-технічного рівня та рівня

комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [22].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за п'ятибальною шкалою)					
	0	1	2	3	4
<i>I</i>	2	3	4	5	6
<i>Технічна здійсненність концепції</i>					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено робоздатність продукту в реальних умовах
<i>Ринкові переваги (недоліки)</i>					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижча за ціни аналогів	Ціна продукту значно нижча за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
<i>Ринкові перспективи</i>					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження табл. 4.1

<i>Практична здійсненність</i>					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більший 5-ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій менший 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що потребує значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту потребує незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	4	5
2. Ринкові переваги (наявність аналогів)	1	2	1
3. Ринкові переваги (ціна продукту)	3	2	3
4. Ринкові переваги (технічні властивості)	3	2	2
5. Ринкові переваги (експлуатаційні витрати)	2	3	2
6. Ринкові перспективи (розмір ринку)	2	2	2
7. Ринкові перспективи (конкуренція)	4	4	3
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	3	4	3
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	3	4
Сума балів	40	40	39
Середньоарифметична сума балів $СБ_c$	39,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [22].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ_c$, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» становить 39,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою [23]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при

цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де I_{ni} та I_{na} – чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість каналів передачі керування сигналу	шт.	1	2	2	0,35
Швидкість реагування	мс	1000	10	100	0,25
Маса	кг	1,5	0,5	3	0,15
Діапазон регулювання з кроком 1	0	1	256	256	0,1
Доступність інтерфейсу	бали	6	8,5	1,42	0,15

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 2 \cdot 0,35 + 100 \cdot 0,25 + 3 \cdot 0,15 + 256 \cdot 0,1 + 1,42 \cdot 0,15 = 51,96.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 51,96 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [22]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=22$ дні.

$$Z_o = 15240,00 \cdot 34 / 22 = 23552,73 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник	15240,00	692,73	34	23552,73
Інженер-розробник систем автоматизованого керування	15110,00	686,82	34	23351,82
Науковий співробітник за напрямом досліджень систем автоматизованого керування	15200,00	690,91	22	15200,00
Лаборант	6820,00	310,00	17	5270,00
Всього				67374,55

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M = 6700,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [22];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_i = 6700,00 \cdot 1,50 \cdot 1,65 / (22 \cdot 8) = 94,22 \text{ грн.}$$

$$Z_{pl} = 94,22 \cdot 6,00 = 565,31 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення допоміжного обладнання проведення розробки	6,00	4	1,50	94,22	565,31
Встановлення програмного забезпечення розробника систем автоматизованого керування	4,00	5	1,70	106,78	427,13
Підготовка робочого місця інженера-дослідника	6,00	2	1,10	69,09	414,56
Монтаж блоків системи керування освітлення гаражу	8,00	4	1,50	94,22	753,75
Налагодження блоків системи керування освітлення гаражу	2,50	3	1,35	84,80	211,99
Контроль роботи базових блоків системи	4,00	5	1,70	106,78	427,13
Всього					2799,87

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.7)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (67374,55 + 2799,87) \cdot 11 / 100\% = 7719,19 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{zn}}{100\%} \quad (4.8)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (67374,55 + 2799,87 + 7719,19) \cdot 22 / 100\% = 17136,59 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (4.9)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3,0 \cdot 220,00 \cdot 1,11 - 0 \cdot 0 = 732,60 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір офісний А4, Crystal Pro80, клас С, 500 л, UPM	220,00	3,0	-	-	732,60
Папір офісний А3, Maestro Standart+, клас В, 80 г/м2, 500 л, Mondi	456,00	1,0	-	-	506,16
Папір офісний Офіс Центр А5 80г/м2 500 аркушів клас С	125,00	3,0	-	-	416,25
Органайзер для канцелярського начиння	209,00	2,0	-	-	463,98
ДИСК CD-R VERBATIM 700MB 80MIN 52X BULK 50	10,20	4,0	-	-	45,29
USB флеш накопичувач 32 ГБ	172,00	1,0	-	-	190,92
Склотекстоліт СТФ 2–1.5	126,00	0,1	-	-	13,99
Хлорне залізо	190,00	0,1	-	-	10,55

Продовження табл.4.7

Дріт монтажний	90,00	0,1	-	-	5,00
Лак УР-231	345,00	0,05	-	-	19,15
Спирт етиловий	170,00	0,25	-	-	47,18
Припій ПОС-61	528,00	0,03	-	-	17,58
Флюс БС-2	165,00	0,01	-	-	1,83
Кабель силовий	85,00	0,21	-	-	19,81
Всього					2490,27

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_6 = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.10)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$$K_6 = 1 \cdot 220,00 \cdot 1,11 = 244,20 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
мікроконтролер arduino	1	220,00	244,20
датчик рівня освітлення	2	95,00	210,90
корпус блоку системи керування	1	65,00	72,15
блок живлення	1	190,00	210,90
комплект інтерфейсів	2	99,00	219,78
блок випромінювача	1	415,00	460,65
блок приймача сигналу керування	1	310,00	344,10
інше	1	120,00	133,20
Всього			1895,88

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (4.11)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$B_{спец} = 9750,00 \cdot 1 \cdot 1,11 = 10822,50 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Осцилограф ОС-114/1-4	1	9750,00	10822,50
Паяльна станція ПС-280-А/8	1	1652,00	1833,72
Вольтметр цифровий	1	4820,00	5350,20
Всього			18006,42

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прз} = \sum_{i=1}^k C_{инпрз} \cdot C_{прз.i} \cdot K_i, \quad (4.12)$$

де $C_{инпрз}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прз.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прз}} = 4690,00 \cdot 1 \cdot 1,11 = 5205,90 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Пакет Visual System Simulator	1	4690,00	5205,90
Пакет Microwave Office	1	3815,00	4234,65
Пакет MATLAB SIMULINK	1	4210,00	4673,10
Всього			14113,65

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{е}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.13)$$

де $Ц_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{е}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (41799,00 \cdot 2) / (2 \cdot 12) = 3483,25 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс проектування на базі ПК ASUS i7-JD3200-UA23	41799,00	2	2	3483,25
Обладнання виводу інформації Лазерний принтер HP LaserJet Pro M102w c Wi-Fi (G3Q35A)	7599,00	4	2	316,63
Робоче місце інженера-дослідника спеціалізоване	7750,00	5	2	258,33
Офісна оргтехніка	7999,00	5	2	266,63
Приміщення лабораторії досліджень	635000,00	20	2	5291,67
ОС Windows 11	8465,00	2	2	705,42
Пакет Microsoft Office 2019	7956,00	2	2	663,00
Метрологічний комплекс	14870,00	4	2	619,58
Мобільний телефон (смартфон) XIAOMI MI 10	12000,00	3	2	666,67
Всього				12271,18

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 6,20$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,23 \cdot 250,0 \cdot 6,20 \cdot 0,95 / 0,97 = 356,50 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс проектування на базі ПК ASUS i7-JD3200-UA23	0,23	250,0	356,50
Обладнання виводу інформації Лазерний принтер HP LaserJet Pro M102w с Wi-Fi (G3Q35A)	0,12	15,0	11,16
Робоче місце інженера-дослідника спеціалізоване	0,13	250,0	201,50
Офісна оргтехніка	0,72	2,5	11,16
Метрологічний комплекс	0,32	180,0	357,12
Осцилограф ОС-114/1-4	0,12	180,0	133,92
Паяльна станція ПС-280-А/8	0,06	15,0	5,58
Вольтметр цифровий	0,08	180,0	89,28
Всього			1166,22

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.15)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», приймемо $H_{cv} = 20\%$.

$$B_{cv} = (67374,55 + 2799,87) \cdot 20 / 100\% = 14034,88 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.16)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 35\%$.

$$B_{cn} = (67374,55 + 2799,87) \cdot 35 / 100\% = 24561,04 \text{ грн.}$$

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_g = (Z_o + Z_p) \cdot \frac{H_{ig}}{100\%}, \quad (4.17)$$

де H_{ig} – норма нарахування за статтею «Інші витрати», прийmemo $H_{ig} = 60\%$.

$$I_g = (67374,55 + 2799,87) \cdot 60 / 100\% = 42104,65 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.18)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 110\%$.

$$B_{нзв} = (67374,55 + 2799,87) \cdot 110 / 100\% = 77191,85 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_v + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_v + B_{нзв}. \quad (4.19)$$

$$B_{заг} = 67374,55 + 2799,87 + 7719,19 + 17136,59 + 2490,27 + 1895,88 + 18006,42 + 14113,65 + 12271,18 + 1166,22 + 14034,88 + 24561,04 + 42104,65 + 77191,85 = 302866,24 \text{ грн.}$$

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

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,9$.

$$ZB = 302866,24 / 0,9 = 336518,04 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

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

ΔN – збільшення кількості споживачів пристрою, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	500	750	1000	1000

N – кількість споживачів які використовували аналогічний пристрій у році до впровадження результатів нової науково-технічної розробки, прийmemo 10000 осіб;

C_0 – вартість пристрою у році до впровадження результатів розробки, прийmemo 2500,00 грн;

$\pm \Delta C_0$ – зміна вартості пристрою від впровадження результатів науково-технічної розробки, прийmemo 107,50 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [22]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\mathcal{G}}{100}\right), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 30\%$;

\mathcal{G} – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році $\mathcal{G} = 18\%$.

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (107,50 \cdot 10000,00 + 2607,50 \cdot 500) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 485693,18 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (107,50 \cdot 10000,00 + 2607,50 \cdot 1250) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 884992,69 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (107,50 \cdot 10000,00 + 2607,50 \cdot 2250) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 1417392,04 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (107,50 \cdot 10000,00 + 2607,50 \cdot 3250) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 1949791,39 \text{ грн.}$$

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

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.22)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,27$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 485693,18/(1+0,27)^1 + 884992,69/(1+0,27)^2 + 1417392,04/(1+0,27)^3 + \\ &+ 1949791,39/(1+0,27)^4 = 382435,57 + 548696,56 + 691956,55 + 749502,81 = \\ &= 2372591,49 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot ЗВ, \quad (4.23)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2,1$;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 336518,04 грн.

$$PV = k_{инв} \cdot ЗВ = 2,1 \cdot 336518,04 = 706687,89 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.24)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 2372591,49 грн;

PV – теперішня вартість початкових інвестицій, 706687,89 грн.

$$E_{абс} = ПП - PV = 2372591,49 - 706687,89 = 1665903,61 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = T_{ж} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.25)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 1665903,61 грн;

PV – теперішня вартість початкових інвестицій, 706687,89 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_g = T_{ж} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 1665903,61/706687,89)^{1/4} - 1 = 0,35.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = 0,10$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,20.

$\tau_{\min} = 0,10 + 0,20 = 0,30 < 0,35$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.27)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,35 = 2,83 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino» становить 39,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 51,96 рази.

Також термін окупності становить 2,83 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino».

ВИСНОВОК

На сьогоднішній день швидко розвиваються цифрові технології, все більше механічних пристроїв можна побачити в цифровому вигляді. Таке перевтілення є дуже зручним, адже пристрої стають більш точними в вимірах, більш компактними, набувають більших можливостей. Тому люди все більше стають зацікавленими в тому, щоб зробити своє життя зручнішим. Була поставлена задача створити компактний пристрій, який мав фіксувати показники навколишнього середовища. Пристрій повинен бути компактным, показувати усереднені дані, також пристрій повинен мати порівняно малу вартість.

В результаті проведених досліджень щодо проекту, було проаналізовано літературу, здобуті нові практичні та теоретичні навички, використанні знання з багатьох предметів. Проаналізувавши інформацію було вибрано платформу для розробки проекту, мову написання апаратну частину. Після вибору платформи для розробки, мови програмування та компонентів системи було складено схему проекту, та запрограмовано пристрій за допомогою IDE. В ході виконання завдань були вирішені складнощі, пов'язані з інтерпретацією бібліотек в IDE платформи для розробки, максимальна оптимізація коду та використання потужностей мікроконтролеру. Також були вирішені проблеми з з'єднанням двох середовищ розробки.

В ході виконання проекту були здобуті наступні навички:

- Програмування платформ Arduino;
- Налаштування зв'язку між Arduino та Android ;
- Здобуто навички при роботі з електронікою. Пристрій був виконаний у повному обсязі, проте він може бути модернізований в залежності від бажання власника, або ж в залежності від певної специфікації використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Блум Дж. Вивчаємо Arduino: [пер. з англ. Петі - СПб.: БХВ-Петербург - 2015. 336 с.
2. Голіцина, О.Л. Інформаційні технології: Підручник / О.Л. Голіцина, Н.В. Максимов, Т.Л. Партика, І.І. Попов. - М.: Форум, ИНФРА-М, 2013. - 608с.
3. Голубенко, Н.Б. Інформаційні технології в бібліотечній справі / Н.Б. Голубенко.-Рн / Д: Фенікс, 2012. - 282 с.
4. Демидович, Н.Б. Програмування та ЕОМ. Навчальний посібник з факультативного курсу для учнів 9, 10 класів / Н.Б. Демидович, В.М. Монахов.- М.: Просвещение, 2014. - 240 с.
5. Еванс Б. Arduino. Блокнот програміста. [пер. з англ. Голобов В.] - СПб.: БХВ-Петербург 2007. 40с.
6. Microsoft Visual Studio [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio#:~:text=%D0%A6%D1%96%20%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%B8%20%D0%B4%D0%B0%D1%8E%D1%82%D1%8C%20%D0%B7%D0%BC%D0%BE%D0%B3%D1%83%20%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BB%D1%8F%D1%82%D0%B8,Windows%20Mobile%2C%20Windows%20Phone%2C%20Windows.
7. РОЗРОБЛЕННЯ ЗАСТОСУВАНЬ WINDOWS FORMS – Львів: Львівська політехніка, 2017.
8. Read XML data into a dataset [Електронний ресурс] // Microsoft – Режим доступу до ресурсу: [https://learn.microsoft.com/uk-ua/visualstudio/data-tools/read-xml-data-into-a-dataset?view=vs-2022&tabs=csharp.](https://learn.microsoft.com/uk-ua/visualstudio/data-tools/read-xml-data-into-a-dataset?view=vs-2022&tabs=csharp)
9. C Sharp [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/C_Sharp.](https://uk.wikipedia.org/wiki/C_Sharp)
10. C++ [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/C%2B%2B.](https://uk.wikipedia.org/wiki/C%2B%2B)

11. Тарарака В.Д. Архітектура комп'ютерних систем: навчальний посібник. – Житомир : ЖДТУ, 2018. – 383 с.
12. EasyEDA [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/EasyEDA>.
13. Gerz E. di Justo P. Atmospheric Monitoring with Arduino - O'Reilly Media, Inc. 2013 89с.
14. Платформа Arduino: <http://Arduino-diy.com> (дата звернення 01.12.2022).
15. Характеристики та робота платформи: <http://Arduino-project.net> (дата звернення 01.12.2022).
16. Современные операционные системы, Таненбаум Э., Бос Х., Питер, 2015
17. Навчальні конструктори: <http://hobbytech.com.ua> (дата звернення 01.12.2022).
18. Амперка: <http://wiki.amperka.ru> (дата звернення 01.12.2022).
19. Програмування платформи: <http://forum.Arduino.ua> (дата звернення 01.12.2022).
20. Стартові набори Arduino: <http://Arduino-diy.com> (дата звернення 01.12.2022).
21. Архивы нормативно-технической, проектно-конструкторской, финансовой и др. документации: www.aLee.ru (дата звернення 01.12.2022).
22. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
23. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.

Додаток А. (обов'язковий)**Технічне завдання****ЗАТВЕРДЖЕНО**

Завідувач кафедри АІТ

Бісікало О.В.

(прізвище та ініціали)

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

**«Розробка багатофункціонального телеграм боту для вивчення англійської мови
з використанням сторонніх АРІ»**

08-02.МКР.000.09.000 ТЗ

Виконав: студент 2 курсу, групи ІІСТ-21м
спеціальності 126 – Інформаційні системи
та технології Сагун Є.В.

(шифр і назва спеціальності)

Керівник: доцент кафедри АІТКулик Я. А.

Вінниця ВНТУ 2022

1. Назва та галузь застосування

Магістерська кваліфікаційна робота: «Система керування освітлення гаражу через мобільний застосунок на основі платформи arduino». Галузь застосування – інформаційні технології.

2. Підстава для проведення розробки

Тема магістерської кваліфікаційна роботи затверджена наказом по ВНТУ № ____ від «__» _____ 2022 р.

3. Мета та призначення розробки

Метою роботи є спрощення роботи та взаємодії користувача з системи та можливістю покращення відношення до сучасних технологій.

4. Джерела розробки

– <https://alexgyver.ru/> – Site– AlexGyverTechnologies.

5. Показники призначення

Необхідні вхідні дані для коректної роботи системи:

- вхідні 12 вольт
- Місце в пам'яті на пристрої під встановлення додатку

Результати роботи системи:

- Можливість керування освітленням
- Можливість та зручність у використанні.

6. Економічні показники

- Прогнозовані витрати на розробку – близько 330 тис. грн;
- Термін окупності витрат для виробника – 2,8р.

7. Стадії розробки

1. Розділ 1 « Аналіз сучасного стану методів і засобів » має бути виконаний до 11.12.2022.

2. Розділ 2 « Розробка Технічного та робочого проекту» має бути виконаний до 12.12.2022.

3. Розділ 3 « Розробка системи» має бути виконаний до 13.12.2022.

4. Економічний розділ має бути виконаний до 14.12.2022.

8. Порядок контролю та приймання

1. Рубіжний контроль. Провести до 05.12.2022.

2. Попередній захист магістерської кваліфікаційної роботи. Провести до 7.12.2020.

3. Захист магістерської кваліфікаційної роботи. Провести в період з 19.12.2022 до 22.12.2022.

Додаток Б. (обов'язковий)

Лістинг програми

```

package com.example.bluetoothmonotorkotlin;

import androidx.appcompat.app.AppCompatActivity;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothManager;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    private BluetoothAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }

    private void init() {
        BluetoothManager bluetoothManager = (BluetoothManager) getSystemService(BLUETOOTH_SERVICE);
        if (bluetoothManager != null) adapter = bluetoothManager.getAdapter();
    }
}

package com.example.bluetoothmonitor

import android.bluetooth.BluetoothAdapter
import android.bluetooth.BluetoothDevice
import android.bluetooth.BluetoothManager
import android.content.Context
import android.content.Intent
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.bluetoothmonotorkotlin.databinding.ActivityMainBinding

class BtListActivity : AppCompatActivity(), RcAdapter.Listener {
    private var btAdapter: BluetoothAdapter? = null
    private lateinit var binding: ActivityMainBinding
    private lateinit var adapter: RcAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        init()
    }

    private fun init(){
        val btManager = getSystemService(Context.BLUETOOTH_SERVICE) as BluetoothManager
        btAdapter = btManager.adapter
    }
}

```

```

        adapter = RcAdapter(this)
        binding.rcView.layoutManager = LinearLayoutManager(this)
        binding.rcView.adapter = adapter
        getPairedDevises()
    }

    private fun getPairedDevises(){
        val pairedDevices: Set<BluetoothDevice>? = btAdapter?.bondedDevices
        val tempList = ArrayList<ListItem>()
        pairedDevices?.forEach {
            tempList.add(ListItem(it.name, it.address))
        }
        adapter.submitList(tempList)
    }

    companion object{
        const val DEVICE_KEY = "device_key"
    }

    override fun onClick(item: ListItem) {
        val i= Intent().apply {
            putExtra(DEVICE_KEY, item)
        }
        setResult(RESULT_OK, i)
        finish()
    }
}

package com.example.bluetoothmonitor

import android.bluetooth.BluetoothManager
import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.Menu
import android.view.MenuItem
import androidx.activity.result.ActivityResultLauncher
import androidx.activity.result.contract.ActivityResultContracts
import com.example.bluetoothmonotorkotlin.R
import com.example.bluetoothmonotorkotlin.databinding.ActivityControlBinding

class ControlActivity : AppCompatActivity() {
    private lateinit var binding: ActivityControlBinding
    private lateinit var actListLauncher: ActivityResultLauncher<Intent>
    lateinit var btConnection: BtConnection
    private var listItem: ListItem? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityControlBinding.inflate(layoutInflater)
        setContentView(binding.root)
        onBtListResult()
        init()
        binding.apply {
            bA.setOnClickListener {
                var byteArray = ByteArray(1)
                byteArray[0] = 0x00.toByte()
            }
        }
    }
}

```



```

        btConnection.sendMessage(byteArray)
    }
    bB.setOnClickListener {
        var byteArray = ByteArray(1)
        byteArray[0] = 0x7f.toByte()
        btConnection.sendMessage(byteArray)
    }
}

}

private fun init() {
    val btManager = getSystemService(Context.BLUETOOTH_SERVICE) as BluetoothManager
    val btAdapter = btManager.adapter
    btConnection = BtConnection(btAdapter)
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.control_menu, menu)
    return super.onCreateOptionsMenu(menu)
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    if (item.itemId == R.id.id_list) {
        actListLauncher.launch(Intent(this, BtListActivity::class.java))
    } else if (item.itemId == R.id.id_connect) {
        listItem.let {
            btConnection.connect(it?.mac!!)
        }
    }
    return super.onOptionsItemSelected(item)
}

private fun onBtListResult() {
    actListLauncher = registerForActivityResult(
        ActivityResultContracts.StartActivityForResult()
    ) {
        if (it.resultCode == RESULT_OK) {
            listItem = it.data?.getSerializableExtra(BtListActivity.DEVICE_KEY) as ListItem
        }
    }
}
}
}

```

```
package com.example.bluetoothmonitor
```

```
import android.bluetooth.BluetoothDevice
import android.bluetooth.BluetoothSocket
import android.util.Log
import java.io.IOException
import java.util.*
```

```
class ConnectThread(private val device: BluetoothDevice) : Thread() {
    val uuid = "00001101-0000-1000-8000-00805F9B34FB"
    var mSocket: BluetoothSocket? = null
    lateinit var rThread: ReceiveThread

    init {
        try {
            mSocket = device.createRfcommSocketToServiceRecord(UUID.fromString(uuid))

```

```

    }catch (i: IOException){
    }
}

override fun run() {
    try {
        Log.d("MyLog","Connecting...")
        mSocket?.connect()
        Log.d("MyLog","Connected")
        rThread = ReceiveThread(mSocket!!)
        rThread.start()
    }catch (i: IOException){
        Log.d("MyLog","Can not connect to device")
        closeConnection()
    }
}

fun closeConnection(){
    try {
        mSocket?.close()
    }catch (i: IOException){

    }
}
}

```

```

package com.example.bluetoothmonitor

import android.bluetooth.BluetoothAdapter

class BtConnection(private val adapter: BluetoothAdapter) {
    lateinit var cThread: ConnectThread
    fun connect(mac: String) {
        if (adapter.isEnabled && mac.isNotEmpty()) {
            val device = adapter.getRemoteDevice(mac)
            device.let {
                cThread = ConnectThread(it)
                cThread.start()
            }
        }
    }
    fun sendMessage(message: ByteArray){
        cThread.rThread.sendMessage(message)
    }
}

```

```

package com.example.bluetoothmonitor

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.DiffUtil
import androidx.recyclerview.widget.ListAdapter
import androidx.recyclerview.widget.RecyclerView
import com.example.bluetoothmonotorkotlin.R

```

```
import com.example.bluetoothmonotorkotlin.databinding.ListItemBinding
```

```
class RcAdapter(private val listener: Listener) : ListAdapter<ListItem, RcAdapter.ItemHolder>(ItemComparator()) {
```

```
    class ItemHolder(view: View) : RecyclerView.ViewHolder(view){
        val binding = ListItemBinding.bind(view)
```

```

        fun setData(item: ListItem, listener: Listener) = with(binding){
            tvName.text = item.name
            tvMac.text = item.mac
            itemView.setOnClickListener {
                listener.onClick(item)
            }
        }
    }

```

```
    companion object{
        fun create(parent: ViewGroup): ItemHolder{
            return ItemHolder(LayoutInflater.
                from(parent.context).
                inflate(R.layout.list_item, parent, false))
        }
    }

```

```
}
```

```
class ItemComparator : DiffUtil.ItemCallback<ListItem>(){
```

```
    override fun areItemsTheSame(oldItem: ListItem, newItem: ListItem): Boolean {
        return oldItem.mac == newItem.mac
    }

```

```
    override fun areContentsTheSame(oldItem: ListItem, newItem: ListItem): Boolean {
        return oldItem == newItem
    }

```

```
}
```

```
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemHolder {
        return ItemHolder.create(parent)
    }

```

```
}
```

```
    override fun onBindViewHolder(holder: ItemHolder, position: Int) {
        holder.setData(getItem(position), listener)
    }

```

```
}
```

```
interface Listener{
```

```
    fun onClick(item: ListItem)
```

```
}
```

```
}
```

Додаток В. (обов'язковий)

Графічна частина

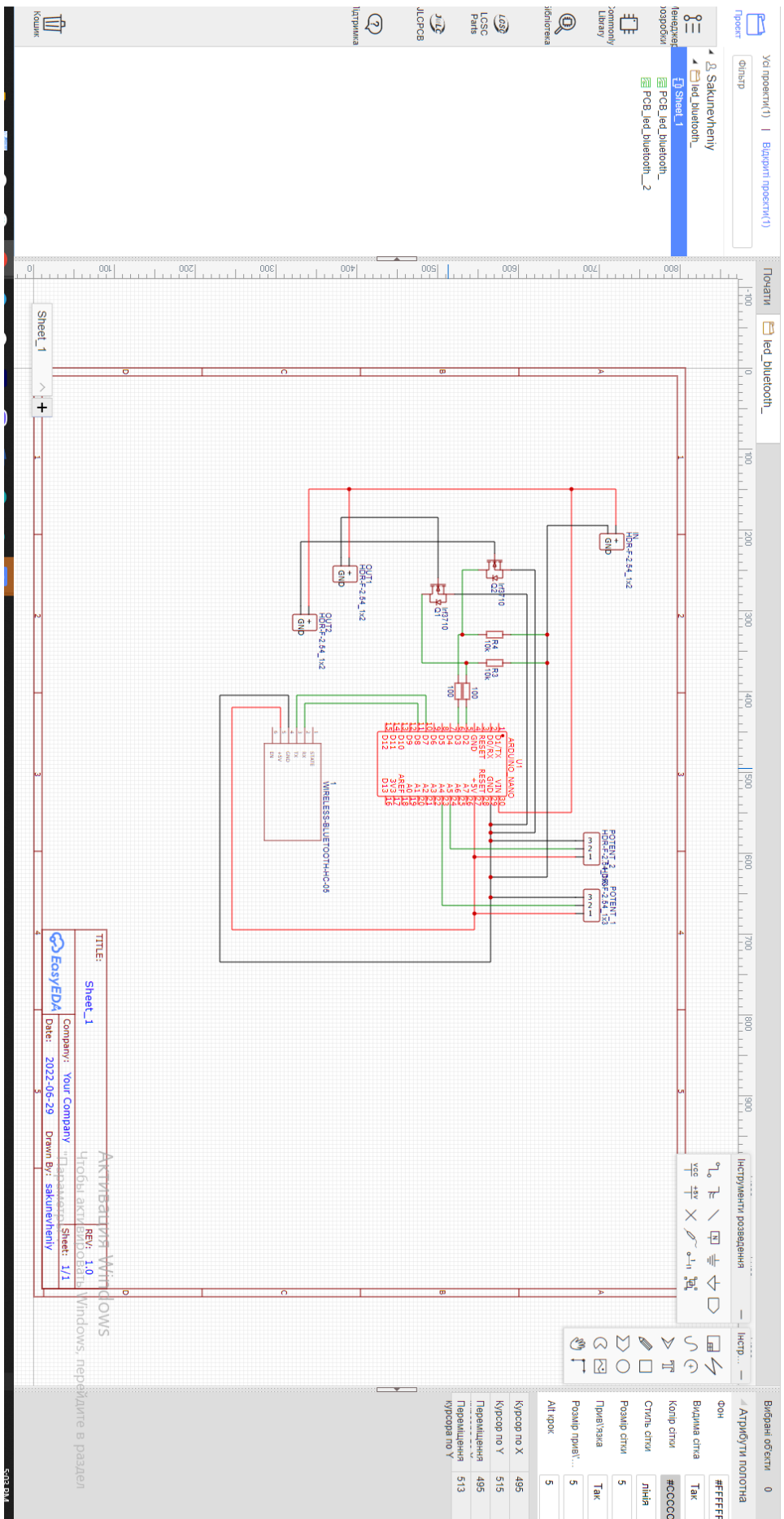


Рисунок В.1 Схема підключення елементів у системі

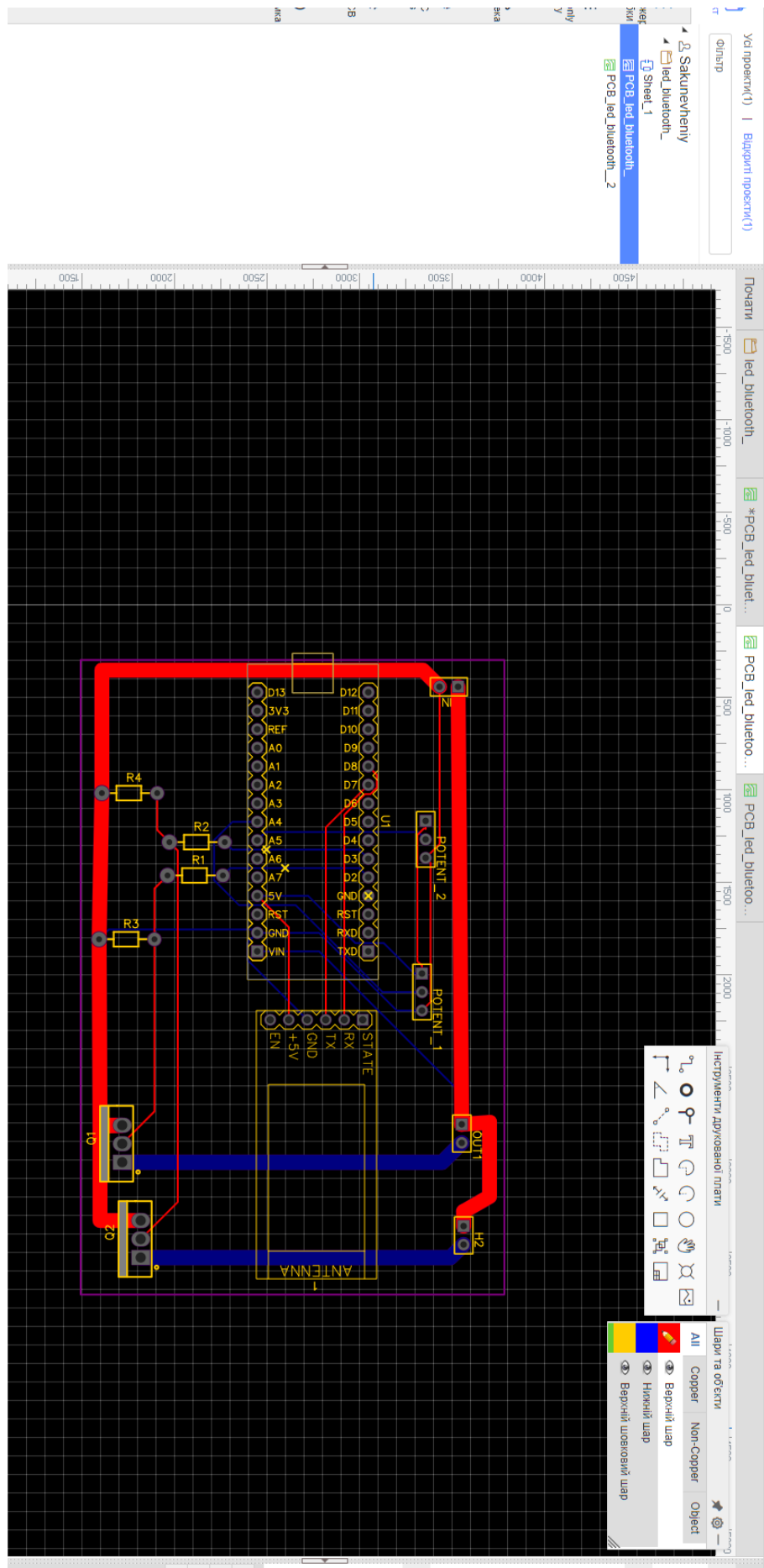


Рисунок В.2 Розведене обладнання для печатної плати

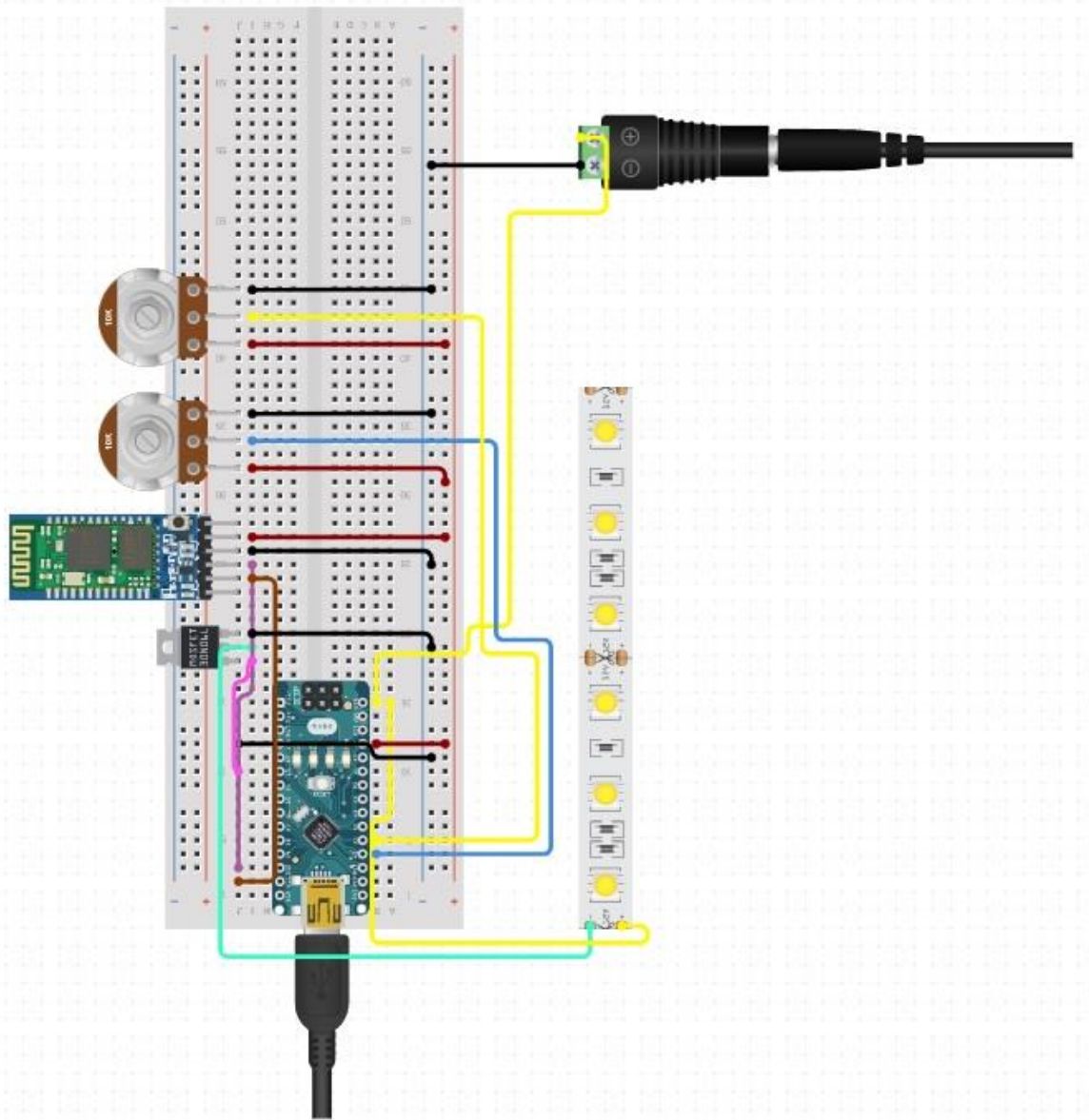


Рисунок В.3 Можлива модифікація системи

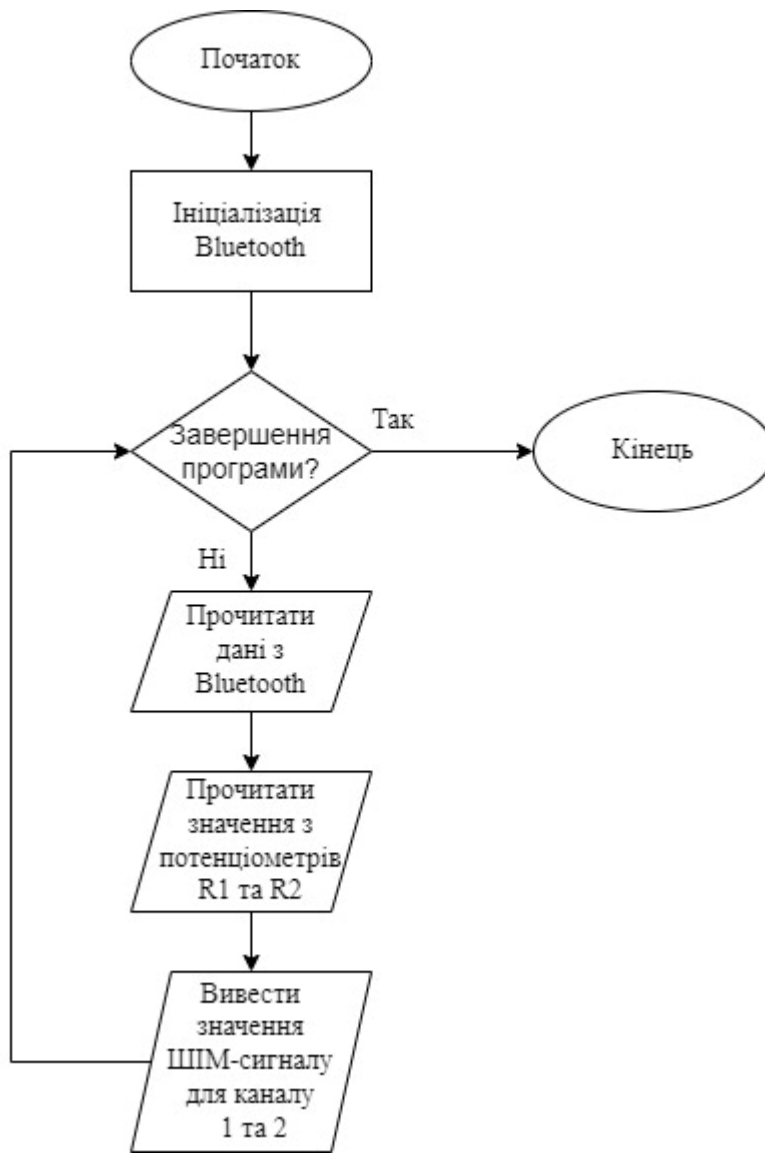


Рисунок В.4 Схема роботи програмного забезпечення обладнання без участі смартфона

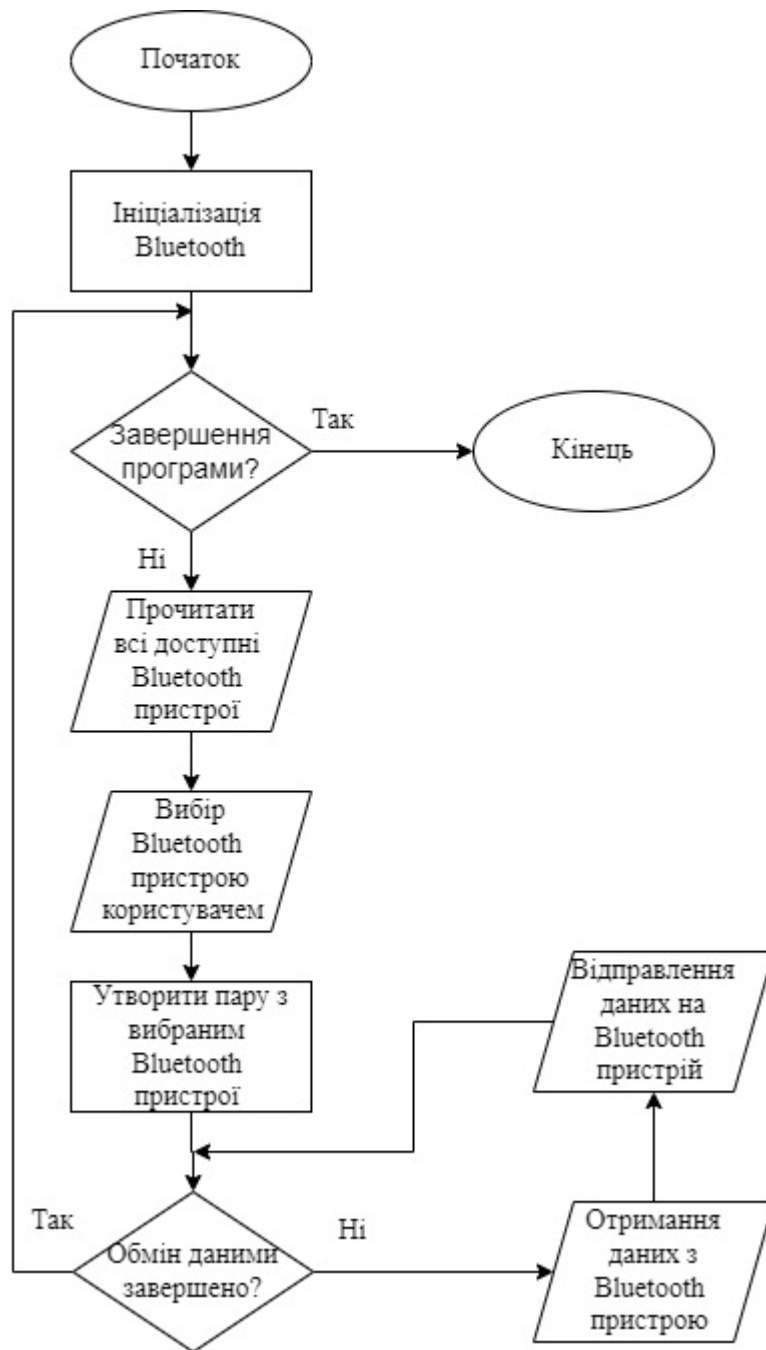


Рисунок В.5 Схема роботи програмного застосунку.

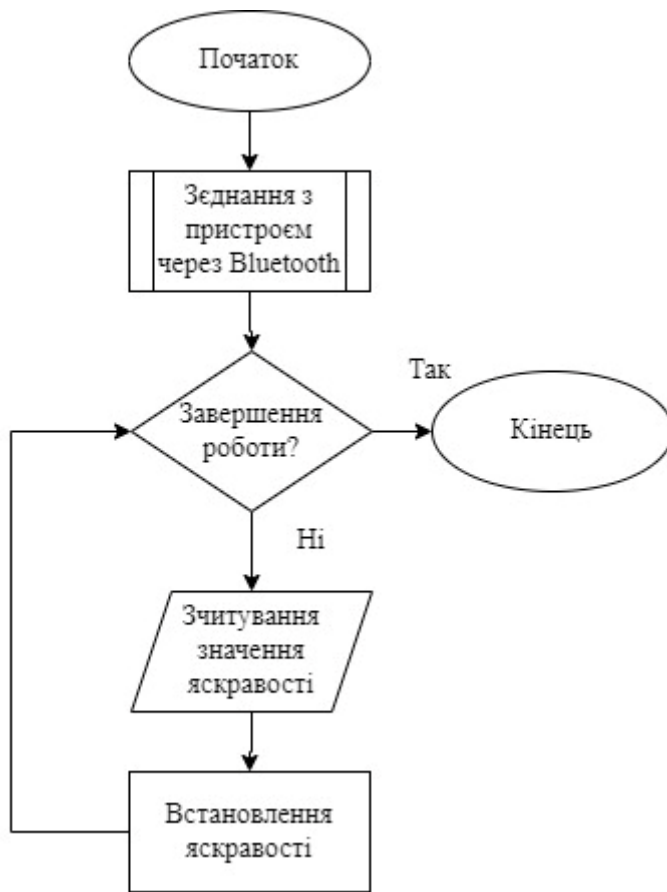


Рисунок В.6 Схема роботи програмного забезпечення обладнання з смартфоном

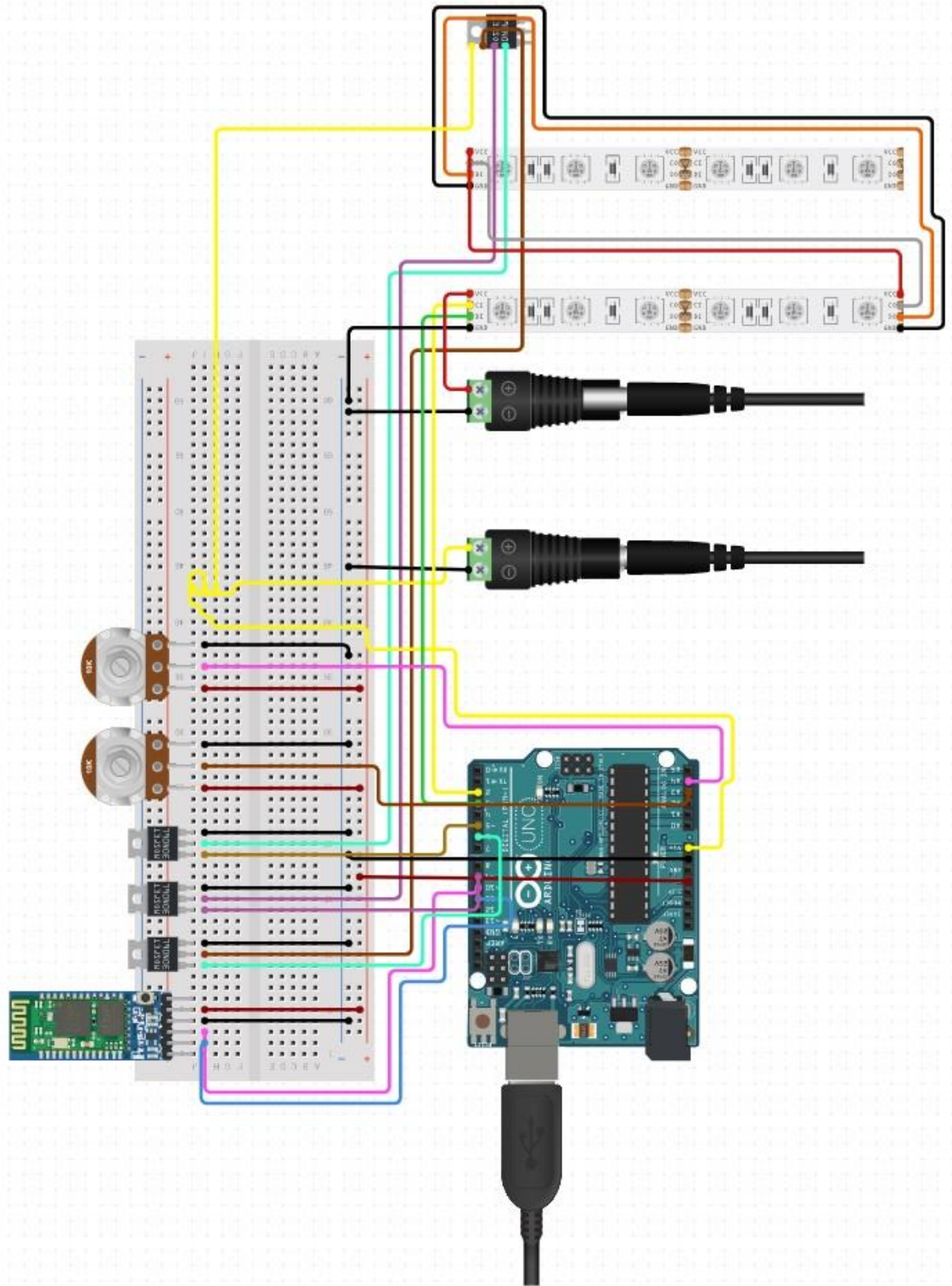


Рисунок В.7 Можлива модифікація системи

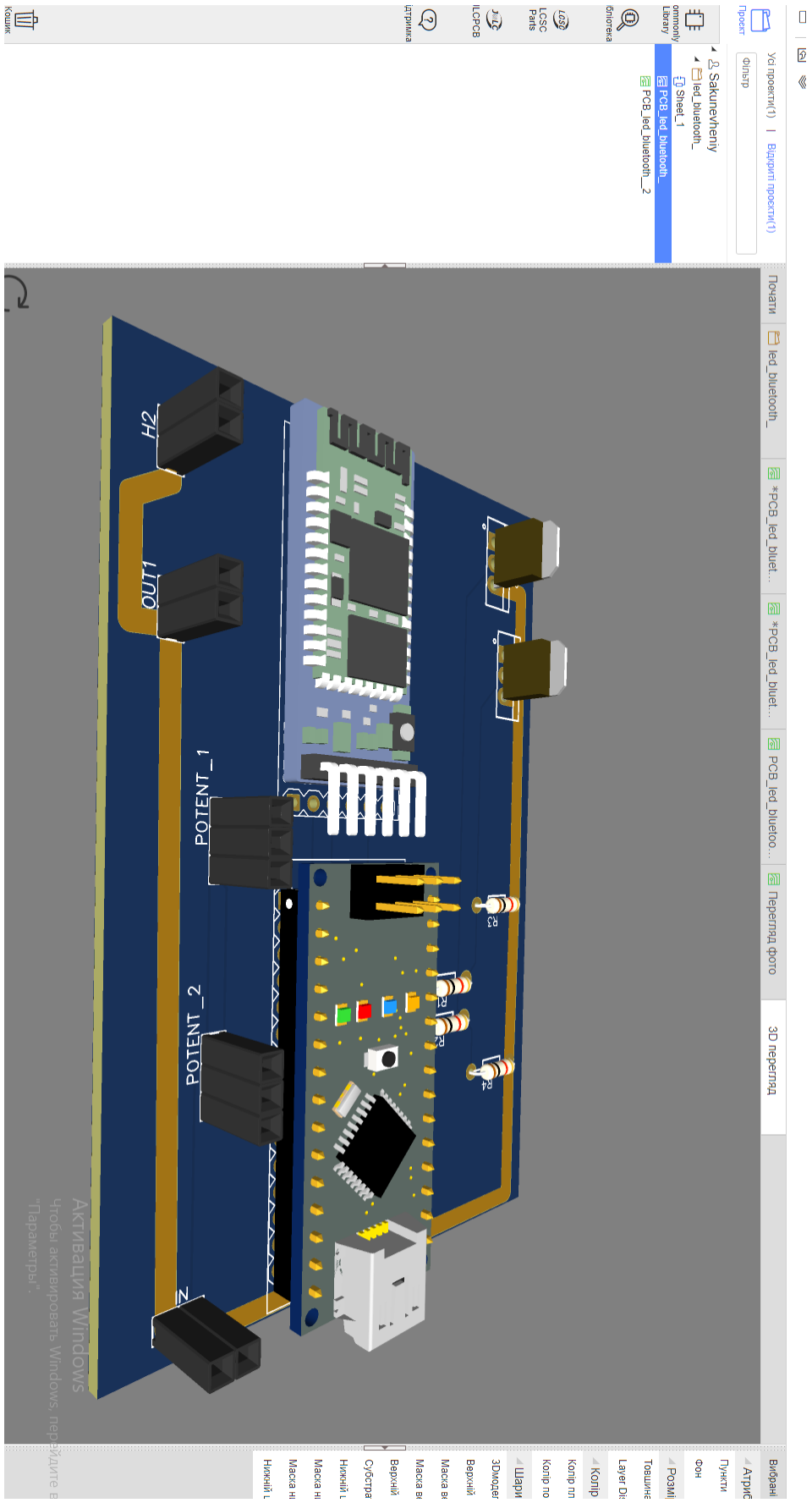


Рисунок В.9 3D Вигляд розведеного обладнання для печатної плати