

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «МІКРОКОНТРОЛЕРНА СИСТЕМА ОХОРОНИ ПЕРИМЕТРУ
ПІДПРИЄМСТВА»

Виконав: студент 2 курсу, групи 1КІ-23м
спеціальності 123 – Комп'ютерна інженерія
Рогозянський В.А.

Керівник д.т.н., проф.

Азаров О.Д.

Оцінювач д.т.н. доц. каф. МБІС

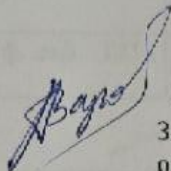
Шиян А.А.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.

«___» _____ 2024 р.

Вінниця 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 Комп'ютерна інженерія



ЗАТВЕРДЖУЮ
Завідувач кафедри
обчислювальної техніки
проф., д.т.н. О. Д. Азаров

« 18 » вересня 2024 р.

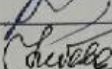
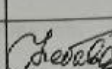
З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Рогозянському Вячеславу Андрійовичу

- 1 Тема роботи «Мікроконтролерна система охорони периметру підприємства», керівник роботи Азаров Олексій Дмитрович, д.т.н., професор, затверджені наказом вищого навчального закладу від 17.09.2024 року № 310.
- 2 Строк подання студентом роботи 16.12.2024 р.
- 3 Вихідні дані до роботи: мова програмування C++, бібліотеки для взаємодії з мережевими модулями такими як ESP8266, середовище розробки Arduino IDE.
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, огляд та аналіз автоматизованих систем охорони периметрів та об'єктів, проектування апаратного забезпечення системи, проектування програмного забезпечення системи, висновки.
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема системи, функціональна схема блоку управління системи, функціональна схема датчика системи, програмний код головного блоку системи, програмний код блоку датчика системи.
- 6 Консультантів розділів роботи представлено в табл. 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	Азаров О. Д., д.т.н., проф.		
5	Небава М.І., к.е.н., проф. каф. ЕП і ВМ		 -

7 Дата видачі завдання 18 . вересня . 2024 р.

8 Календарний план наведено в табл. 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Прим.
1	Постановка мети та задач роботи	21.10.24	вик.
2	Огляд і аналіз аналогів систем охорони периметрів та об'єктів	25.10-30.10.24	вик.
3	Проектування апаратного забезпечення системи	31.10-08.11.24	вик.
4	Складання структурної схеми системи, що розробляється	09.11-15.11.24	вик.
5	Визначення моделей компонентів системи	16.11-20.11.24	вик.
6	Проектування програмного забезпечення системи	21.11-25.11.24	вик.
7	Програмна реалізація рішень	26.11-31.11.24	вик.
8	Тестування системи	01.12-03.12.24	вик.
9	Розрахунок економічної частини роботи	04.12-07.12.24	вик.
10	Оформлення пояснювальної записки та ілюстративного матеріалу	08.12.24	вик.
11	Аналіз виконання роботи, висновки, додатки		вик.
12	Перевірка якості виконання магістерської роботи та усунення недоліків		вик.

Студент  Рогозянський В. А.Керівник роботи  Азаров О.Д.

АНОТАЦІЯ

УДК 004.9

Рогозянський В. А.

Мікроконтролерна система охорони периметру підприємства. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2024, 109 с.

На укр.мові. Бібліогр.: рис. , табл. .

Дана магістерська кваліфікаційна робота присвячена створенню мікроконтролерної системи охорони периметру підприємства.

У ході дипломної роботи досліджено методи виявлення порушень периметру та реагування на них, а також спроектовано і реалізовано систему спостереження за периметром підприємств, що є важливим інструментом для забезпечення безпеки об'єктів, що вимагають надійного захисту від несанкціонованого доступу. Система спроектована з метою своєчасного сповіщення персоналу або власників про будь-які порушення периметру, дозволяючи оперативно відреагувати на можливі вторгнення, незалежно від місця перебування відповідального персоналу або власника.

Ключові слова: система стеження за периметром, мікроконтролер, лазер, датчик світла, безпроводний зв'язок.

ABSTRACT

UDC 004.9

Rohozianskyi V. A.

Microcontroller system for enterprise perimeter protection. Master's qualification work in specialty 123 - computer engineering, educational program - computer engineering. Vinnytsia: VNTU, 2024, 109 p.

In Ukrainian. Bibliography: fig. , table. .

This master's qualification work is devoted to the creation of a microcontroller system for enterprise perimeter protection.

During the diploma work, methods for detecting perimeter violations and responding to them were investigated, and a perimeter surveillance system for enterprises was designed and implemented, which is an important tool for ensuring the security of facilities that require reliable protection against unauthorized access. The system is designed to promptly notify personnel or owners of any perimeter violations, allowing them to respond promptly to possible intrusions, regardless of the location of the responsible personnel or owner.

Keywords: perimeter surveillance system, microcontroller, laser, light sensor, wireless communication.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД І АНАЛІЗ АВТОМАТИЗОВАНИХ СИСТЕМ ОХОРОНИ ПЕРИМЕТРІВ ТА ОБ’ЄКТІВ	12
1.1 Автоматизовані системи охорони.....	12
1.2 Огляд аналогів	13
1.3 Проектування та вимоги до створюваної системи.....	16
2 ПРОЕКТУВАННЯ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	19
2.1 Розробка структурної схеми системи	19
2.2 Визначення моделей компонентів системи.....	20
2.2.1 Модуль датчика освітленості на базі фоторезистора GL5528	20
2.2.2 Модуль зв’язку NRF24L01	22
2.2.3 Мікроконтролер ATmega328.....	29
2.2.4 Мікроконтролер ATmega328P.....	34
2.2.5 Wi-Fi-модуль ESP8266 ESP-01.....	38
2.2.6 Екран LCD 1602 I2C	39
2.3 Функціональні схеми пристроїв системи	41
2.4 Розробка електричної принципової схеми	42
3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	45
3.1 Розробка основного алгоритму роботи системи	45
3.2 Розробка алгоритму зміни пароля	46
3.3 Вибір засобів для програмування мікроконтролера	48
3.4 Програмування мікроконтролерів	49
3.4.1 Програмування блоку датчиків	49
3.4.2 Програмування блоку управління системи	51
4 ТЕСТУВАННЯ СИСТЕМИ	62

					08-54.КМКР.016.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.		Рогозянський В.А.			Мікроконтролерна система охорони периметру підприємства Пояснювальна записка	Лім.	Арк.	Аркушів
Перевір.		Азаров О. Д.					6	106
Реценз.		Шиян А. А.				ВНТУ, гр. 1КІ-23м		
Н. Контр.		Швець С. І.						
Затверд.		Азаров О. Д.						

4.1 Тестування підключення до мережі Wi-Fi та сервера	62
4.2 Тестування реагування на порушення периметру	64
5 ЕКОНОМІЧНА ЧАСТИНА	66
5.1 Комерційний та технологічний аудит науково-технічної розробки	66
5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи	69
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	74
ВИСНОВКИ	80
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	81
ДОДАТОК А Технічне завдання	83
ДОДАТОК Б Лістинг коду головного блоку системи	87
ДОДАТОК В Лістинг коду блоку датчика.....	102
ДОДАТОК Г Структурна схема системи.....	104
ДОДАТОК Д Функціональна схема блоку управління	105
ДОДАТОК Е Функціональна схема датчика	106
ДОДАТОК Ж Протокол перевірки навчальної (кваліфікаційної) роботи.....	107

ВСТУП

Побудова систем пошуку, виділення та розпізнавання об'єктів із використанням засобів комп'ютерного зору — актуальна задача сьогодення. Рішення у цій області знаходять широке застосування у найрізноманітніших сферах життя людини: відстеження появи машин на стоянці, пошук забутого багажу в приміщеннях аеропортів та вокзалів, забезпечення захисту від несанкціонованого проникнення на певні об'єкти, розпізнавання облич людей, автомобільних номерів, написів, тощо. Етап розпізнавання об'єктів є одним із найважливіших у загальній послідовності вирішення задач по ідентифікації об'єктів. Ця задача, із якою людина легко впорядується кожного дня, є досить складною для обчислювальної техніки. Знаходження відповідності між двома

Системи охорони периметра є важливою складовою безпеки, яка забезпечує високий рівень захисту від проникнення. Ці системи дозволяють виявляти та протидіяти спробам несанкціонованого доступу до охоронюваної території, забезпечуючи надійний захист від потенційних загроз. Часто системи охорони периметра є частиною комплексної системи безпеки, яка може включати відеоспостереження, сигналізацію та інші засоби контролю.

Актуальність даної роботи полягає в необхідності розробки інноваційної системи контролю периметра. Зростаюча потреба в таких системах зумовлена зростанням криміногенної ситуації, що призводить до збільшення кількості крадіжок, вандалізму та інших протиправних дій. Тому створення ефективних систем безпеки є актуальним питанням сучасності.

Типи систем безпеки можна розділити на дві основні категорії: активні та пасивні.

Системи активної безпеки: ці системи постійно контролюють і аналізують навколишнє середовище за допомогою різних датчиків і датчиків. Вони можуть автоматично виявляти ненормальну поведінку або загрози та застосовувати заходи для

їх усунення. Системи активної безпеки включають такі технології, як системи контролю доступу, пожежної сигналізації, системи охорони периметра тощо.

Пасивні системи безпеки: на відміну від активних систем безпеки, ці системи не контролюють середовище в реальному часі. Зазвичай він використовується для зберігання інформації або свідчень про минулі події. До пасивних систем безпеки належать системи зберігання відео, аудіозапису, системи архівації даних тощо.

Системи пасивного та активного захисту часто використовуються разом для забезпечення комплексного захисту. Наприклад, система відеоспостереження може активно стежити за навколишнім середовищем і виявляти загрози в реальному часі (активний компонент), а також записувати відео для подальшого розслідування або збереження доказів (пасивний компонент).

Крім того, системи безпеки можна класифікувати за різними критеріями, такими як рівень технологічної складності (від простих до більш просунутих), масштабу (від локальних до глобальних) або методів виявлення загроз (від простих датчиків до біометричних систем).

Системи безпеки периметра можуть використовувати різні технології виявлення порушень чи загроз, включаючи RFID-мітки, брелоки, біометричні дані (відбитки пальців, розпізнавання обличчя), паролі та PIN-коди. Доступ контролюється на основі рівня доступу, наданого окремим користувачам.

Автоматизуючи контроль доступу, системи охорони периметра знижують ризики, пов'язані з людським фактором, і забезпечують високий рівень захисту. Він забезпечує точне розпізнавання загроз і швидке реагування на порушення.

Сьогодні технологічний прогрес створює нові можливості для вдосконалення систем охорони периметра. Наприклад, використання машинного навчання та штучного інтелекту дозволяє автоматично виявляти незвичну поведінку та розпізнавати різні загрози. Крім того, інтеграція з іншими системами безпеки (відеоспостереження, пожежна сигналізація, системи контролю доступу) створює умови для створення багаторівневої системи захисту.

Таким чином, системи охорони периметра є важливим елементом комплексної системи безпеки, що забезпечує захист від несанкціонованого доступу та протиправної діяльності. Вибір конкретної системи повинен ґрунтуватися на характеристиках, розмірі та характеристиках об'єкта, а також ресурсах, доступних для будівництва та обслуговування.

Розробка системи охорони периметру допоможе вирішити ряд важливих проблем:

- запобігання незаконному вторгненню може виявляти спроби незаконного проникнення на об'єкт чи територію та попереджати про спроби перекриття або знищення огорожі;

- реагування на загрози в режимі реального часу, тобто після виявлення потенційної загрози система може автоматично сповістити персонал безпеки або місцеві правоохоронні органи;

- миттєве виявлення спроби потрапити на підприємство, склад чи магазин невідомих осіб;

- покращення ефективності роботи працівників безпеки шляхом автоматизації повсякденних задач, що дозволяє працівникам безпеки сконцентруватися на більш складних задачах.

Задачі дослідження:

- проаналізувати системи охорони периметру та методи їх реалізації;

- проведення огляду та аналізу технологій охорони периметру;

- розробка проекту системи охорони периметру;

- розробка системи.

Об'єкт дослідження — процес забезпечення охорони периметру підприємства за допомогою автоматизованих технічних засобів.

Предмет дослідження — мікроконтролерна система охорони периметру, розроблена на базі платформи Arduino, яка включає сенсори, виконавчі пристрої та програмне забезпечення для моніторингу й реагування на загрози.

Практичне значення отриманих результатів полягає у тому, що була розроблена автоматизована система охорони периметру.

Апробація результатів магістерської роботи — участь у науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії 2025 доповіддю «Мікроконтролерна система охорони периметру підприємства» Тези опубліковано 13.11.24.

1 ОГЛЯД І АНАЛІЗ АВТОМАТИЗОВАНИХ СИСТЕМ ОХОРОНИ ПЕРИМЕТРІВ ТА ОБ'ЄКТІВ

1.1 Автоматизовані системи охорони

Сьогодні існує безліч автоматизованих систем, які дозволяють контролювати периметр об'єкта. Ці системи безпеки периметра використовують різні технології для захисту території, включаючи датчики руху, камери відеоспостереження та інфрачервоні датчики.

Одним з найпоширеніших варіантів є система охорони периметра на основі відеоспостереження. Ця система дозволяє не тільки спостерігати за об'єктом у режимі реального часу, а й записувати відеоматеріал для подальшого аналізу. Такі системи часто використовують відеоаналітику для автоматичного виявлення порушень і надсилання сповіщень оператору або охоронному агентству.

Інфрачервоні датчики також широко використовуються для охорони периметра. Ці датчики виявляють зміни інфрачервоного випромінювання, що випромінюється людьми та транспортними засобами. Вони в будь-який момент можуть знаходити предмети на відстані.

Для захисту периметра також використовується радіочастотна система. Вони складаються з передавача і приймача, які створюють навколо об'єкта невидиме поле. Якщо предмет перетинає поле, система виявляє порушення та надсилає сповіщення.

Останні розробки включають використання безпілотних літальних апаратів і роботизованих систем для спостереження за периметром. Ці технології можуть забезпечити високоякісне відео з висоти пташиного польоту або здійснювати наземне спостереження. Його також можна використовувати для виявлення руху або безпосереднього реагування на порушення.

Іншим напрямком розвитку є використання аналітики великих даних і штучного інтелекту для аналізу відеоданих і виявлення аномалій і потенційних загроз. Ці системи можуть автоматично аналізувати великі обсяги відеоданих, виявляти незвичайні моделі поведінки або порушення та надсилати сповіщення, коли виявлено

потенційні порушення.

Слід зазначити, що вибір системи спостереження за периметром залежить від низки факторів, включаючи розмір і характер об'єкта, бюджет, технічні можливості та конкретні потреби безпеки.

Електричні огорожі — це активні системи забезпечення безпеки, які використовують електричні імпульси для виявлення спроб проникнення на об'єкт. У разі спроби перетнути бар'єр система видає сигнал тривоги, який дозволяє негайно виявити загрозу безпеки.

Системи виявлення вібрації — використовують датчики, які реагують на фізичні зміни, такі як вібрація або зміни температури. Вони можуть виявити спроби порізати, зламати або пошкодити пристрій.

Електричні огорожі та системи виявлення вібрації часто використовуються для захисту від фізичних вторгнень, тоді як інфрачервоні бар'єри та радарні системи можуть виявити спроби вторгнення до того, як вони будуть виявлені.

1.2 Огляд аналогів

Вибір аналогу впав на компанію Optex. Вона пропонує різноманітні системи, що відповідають різним вимогам, мають різні функції та ціни.

Візьмемо, наприклад, систему безпеки Optex AX-200TFR. Optex AX-200TFR - це високоякісний детектор руху виробництва Optex. Сюди входять периметр і будівлі, склади, стоянки тощо. це активний інфрачервоний бар'єр, який використовується для захисту.

Optex AX-200TFR складається з двох основних компонентів: передавача (передавача) і приймача (приймача). Передавач і приймач встановлюються на протилежних сторонах від того, що ви хочете захистити.

Принцип роботи Optex AX-200TFR заснований на інфрачервоному детектуванні. Передавач випромінює інфрачервоні промені, спрямовані на приймач. Приймач приймає ці промені та контролює їх присутність. Якщо щось або хтось (наприклад,

людина або транспортний засіб) відрізано, система активується.

Основним призначенням Optex AX-200TFR є виявлення будь-якого руху або перетину ліній, створених інфрачервоним сигналом у зоні між передавачем і приймачем. Це означає, що кожен інфрачервоний промінь, який випромінює передавач, перехоплюється чимось або кимось, і сигнал не може досягти приймача.



Рисунок 1.1 — Система охорони периметру компанії Optex

Система має таку функціональність:

- пристрій здатний виявляти об'єкти на відстані до 60 метрів, що робить його ідеальним для великих відкритих територій або периметрів значної довжини;
- завдяки використанню інфрачервоної технології, Optex AX-200TFR стабільно функціонує навіть за несприятливих погодних умов, таких як дощ, туман чи сніг;

— вбудована система анти-маскування оперативно реагує на спроби блокування чи закриття променів, підвищуючи надійність пристрою та мінімізуючи ризик хибних спрацювань;

— чутливість детектора можна налаштувати відповідно до специфічних умов експлуатації, що дозволяє уникати спрацювань через незначні рухи чи шуми;

— пристрій також може мати додаткові можливості, як-от налаштування часу спрацювання, вибір конкретних зон виявлення та конфігурацію тривожних виходів;

— надійний детектор руху Optex AX-200TFR використовується в багатьох промислових і комерційних цілях, де потрібна система безпеки. Його можна використовувати для активації системи спостереження, запуску тривоги або ввімкнення світла при виявленні руху.

Недоліками цієї системи є:

— система чутлива до перешкод: високі рослини, гілки дерев чи інші предмети, які можуть заблокувати промінь, здатні спричинити некоректну роботу або хибні спрацювання;

— деякі погодні умови, зокрема сильний дощ, густий туман або снігопад, можуть впливати на інфрачервоне випромінювання, викликаючи його розсіювання або відбиття, що іноді призводить до зниження точності чи появи помилкових сигналів;

— максимальна дальність детекції Optex AX-200TFR становить 60 метрів, що може виявитися недостатнім для моніторингу дуже великих територій або довгих периметрів;

— висока вартість пристрою може стати стримуючим фактором, оскільки він дорожчий за деякі інші типи датчиків руху;

— система має невелике затримання між спрацюванням детектора та активацією пов'язаних систем, наприклад, сигналізації чи відеоспостереження, що може бути критичним у ситуаціях, де потрібна негайна реакція.

1.3 Проектування та вимоги до створюваної системи

Розробимо систему на основі трьох компонентів, які опишемо далі.

У загальному вигляді система охорони периметру представляється, як показано на рисунку 1.2.

Датчик — це пристрій, який виявляє зміни у лазерному промені, що виникають при перетині периметра, і передає інформацію про порушення до центрального блоку управління.

Лазерний модуль створює частково невидиму перешкоду у формі променя.

Робота датчика і лазерного модуля узгоджена: лазерний промінь спрямовується на чутливий елемент датчика, який фіксує його. У разі змін, спричинених перетином променя, датчик надсилає сигнал до центрального блоку.

Головний блок системи — основний елемент системи охорони який слугує для прийому та обробки даних з датчика шляхом безпроводної передачі інформації. Після чого блок управління виконує подальші дії щодо заходу безпеки в разі порушення периметру. Відбувається звукове повідомлення про тривогу, а також виведення інформації на екран та відправлення даних про порушника на сервер.

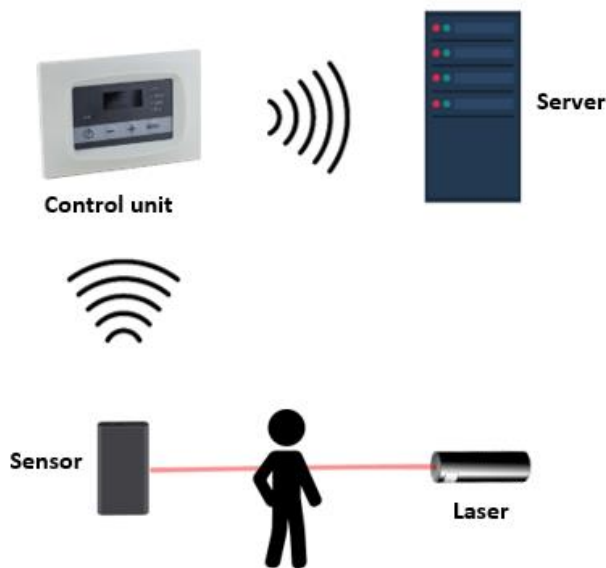


Рисунок 1.2 — Концепція системи охорони периметру

Система охорони периметру має забезпечити:

- миттєве спрацювання системи;
- можливість зміни паролю для вимкнення тривоги;
- збереження паролю в пам'яті системи;
- увімкнення тривоги у разі виявленні порушення периметру;
- надання інформації про місце порушення периметру;
- передавання інформації про порушення на сервер;
- можливість швидкого та дистанційного відключення тривоги.

Отже, на момент закінчення розробки приладу він повинен виконати усі зазначені вище дії.

Коли система фіксує порушення периметра, на екрані блоку управління з'являється відповідна інформація про точку перетину. Це супроводжується звуковим сигналом, який привертає увагу до тривоги. У разі активації тривоги система надсилає запит з детальною інформацією про інцидент, що дозволяє оперативно відреагувати.

Для скасування тривоги користувач повинен ввести пароль. Після правильного введення сигнал тривоги вимикається, а вказівка про порушення зникає. Пароль може бути змінений у будь-який час, але лише після введення чинного пароля. Це гарантує, що тільки авторизований користувач має право змінювати налаштування, що значно підвищує рівень безпеки системи.

Крім того, система може включати графік подій, який зберігає дані про всі тривоги, що дозволяє відстежувати історію порушень. Така функціональність забезпечує додатковий контроль та зручність у використанні.

2 ПРОЕКТУВАННЯ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Розробка структурної схеми системи

Для покращення розуміння влаштування системи було розроблено структурну схему пристрою (рисунку 2.1).

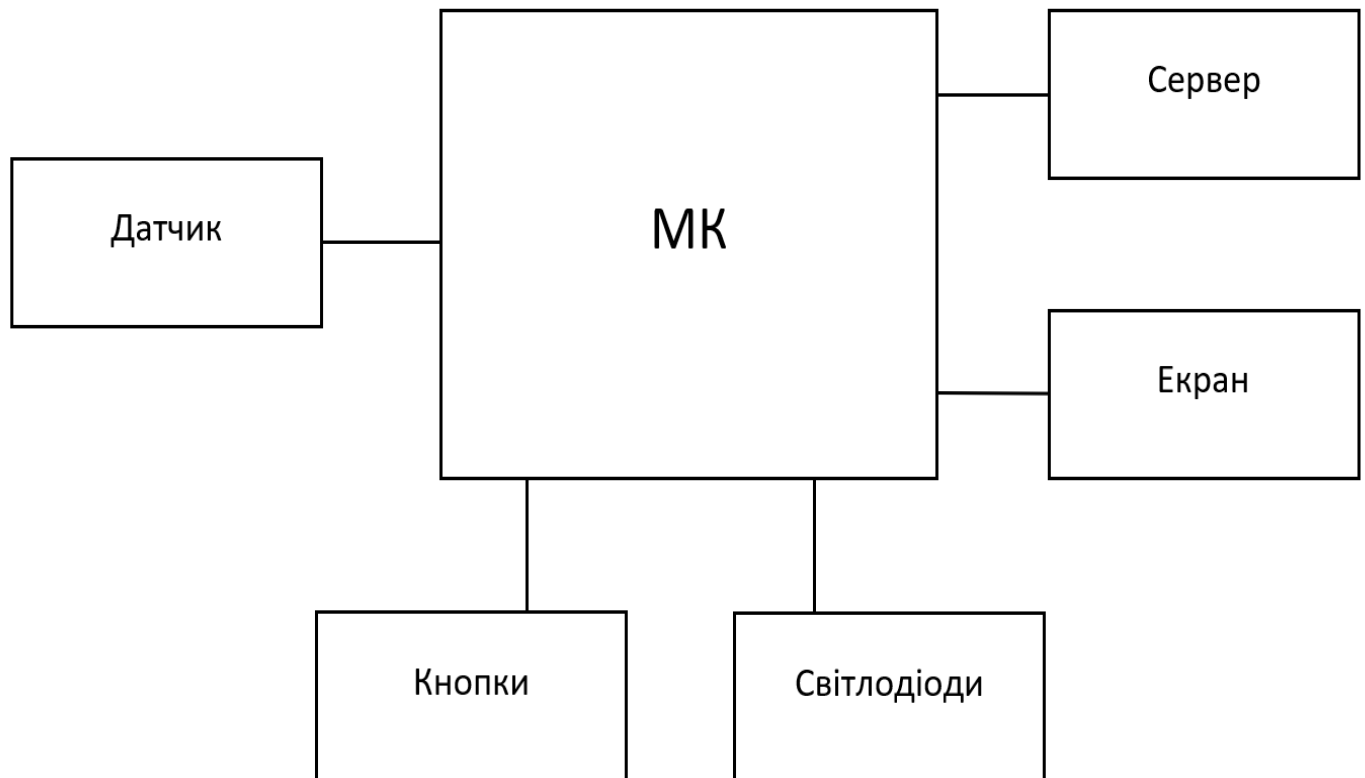


Рисунок 2.1 — Структурна схема системи

Схема має такі системні блоки:

- контролю і управління;
- зчитування;
- лазерний блок.

Блок контролю та управління є основним компонентом системи, що виконує роль "центру" управління, побудованого на базі мікроконтролера. Він отримує дані від датчика через бездротове з'єднання, обробляє їх і відображає на екрані

інформацію про місце порушення периметра. Одночасно мікроконтролер надсилає запит на сервер із деталями про порушення.

Після активації тривоги користувач може скасувати її, ввівши пароль за допомогою кнопок. Це дозволяє вимкнути звуковий сигнал та завершити обробку інциденту.

Світлодіоди слугують для індикації роботи пристрою. Якщо порушень немає то горить зелений світлодіод, а в разі порушення периметру — червоний. Жовтий світлодіод блимає лише тоді коли на сервер відправляється запит з інформацією про стан периметру.

2.2 Визначення моделей компонентів системи

За результатами дослідження існуючих систем охорони периметра підібрано всі необхідні компоненти для системи, спроектованої відповідно до вимог та конструктивної схеми.

Дві основні плати мікроконтролерів, що містять різні мікроконтролери, були прийняті для всієї системи.

Найважливішим МК є — ATmega328-AU на якому буде спроектований блок управління системи.

Для проектування зчитувального пристрою (датчика) обрано:

- плату Arduino Nano на базі мікроконтролера ATmega328;
- модуль датчика освітленості з фоторезистором GL5528;
- модуль зв'язку NRF24L01.

Для виведення інформації користувачеві було обрано символьний дисплей LCD 1602 з мікросхемою I2C.

Для відправлення запитів на сервер взято безпроводний WiFi-модуль ESP8266 ESP-01 за допомогою якого відбувається підключення до мережі WiFi та до сервера.

2.2.1 Модуль датчика освітленості на базі фоторезистора GL5528

Аналогово-цифровий модуль датчика світла на основі фоторезистора GL5528 є простим, але потужним інструментом для вимірювання рівнів освітленості за різних умов. Автоматичне керування освітленням, моніторинг росту рослин тощо. можна використовувати в різних проектах, таких як

GL5528 LDR (Light Dependent Resistor) або фоторезистор - це компонент, який змінює свій опір залежно від інтенсивності світла, що падає на нього. Зі зменшенням сили світла опір фоторезистора зростає і навпаки.

Аналого-цифровий модуль перетворює мінливий аналоговий сигнал, створений фоторезистором, у цифровий сигнал, який можна легко обробити мікроконтролером або комп'ютером. Цей процес передбачає використання АЦП (аналогово-цифрового перетворювача) для перетворення аналогового сигналу в цифровий.

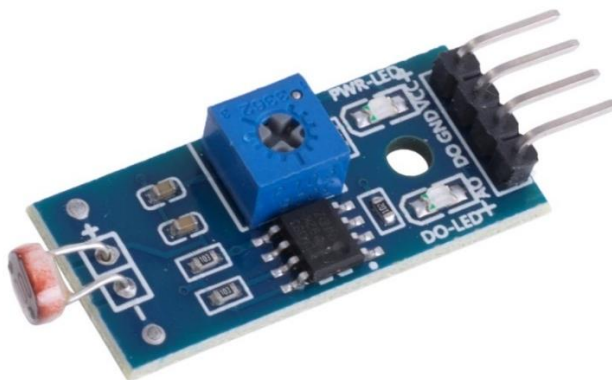


Рисунок 2.2 — Модуль датчика освітленості на базі фоторезистора GL5528

Загальний опис модуля:

- фоторезистор GL5528 є основним компонентом для визначення рівня освітленості, реагуючи на зміни інтенсивності світла;
- аналого-цифровий перетворювач (АЦП) перетворює аналоговий сигнал від фоторезистора у цифровий формат, що дозволяє його подальшу обробку;
- мікроконтролер отримує цифровий сигнал від АЦП, аналізує його та виконує задані дії залежно від рівня освітленості. Його можна запрограмувати для

автоматизації процесів, наприклад, включення або вимкнення освітлення при досягненні певного порогу;

— інтерфейс з'єднання забезпечує підключення модуля до інших пристроїв або систем. Він може включати клеми для проводів для підключення до мікроконтролера або підтримувати інтерфейси, такі як I2C, SPI чи UART, для комунікації з комп'ютером або іншими пристроями.

Таблиця 2.1 — Технічні характеристики модуля освітленості

Технічні характеристики	Значення
Напруга живлення	Від 3.3 В до 5.5 В
Споживаний струм	10 мА
Цифровий вихід	TTL (лог 1 або лог 0)
Аналоговий вихід	Від 0 В до 5 В (V_{cc})
Діаметр отвору для кріплення	2.5 мм
Вихідний струм	15 мА
Розмір	4.2см x 1.5см

2.2.2 Модуль зв'язку NRF24L01

Іноді найбільш зручним варіантом організації обміну даними між двома і більше платами Arduino є бездротовий зв'язок за допомогою радіомодулів. Бездротовий зв'язок відкриває багато можливостей, таких як віддалений моніторинг, збір даних із датчиків, домашня автоматизація, керування роботами тощо.

Існує багато різних типів чіпів приймача, і одним із найпопулярніших є nRF24L01. Таку високу популярність вони здобули завдяки низькій вартості, низькому енергоспоживанню, простоті експлуатації та високій гнучкості при побудові мереж різної топології та складності. На малюнку 2.3 показано модуль.

Вважається, що 126 (від 0 до 125) — це максимальна кількість модулів, які можуть передавати дані між частотними каналами. У деяких випадках кількість частотних каналів може визначати кількість модулів, але, як правило, це не пов'язано безпосередньо. Кількість модулів залежить від низки речей — яка пропускна здатність потрібна, яке посилення реалізується, топологія мережі тощо. Навіть якщо ви використовуєте апаратну адресацію (до п'яти байт адресного простору, це понад трильйон), тож nRF24L01 не так багато.

Зрозуміти, що таке частотні канали і для чого вони корисні, неважко. Модулі, налаштовані на одному каналі, можуть обмінюватися даними. Якщо модулі налаштовані на різні канали, приймач не зможе «почути» дані, надіслані іншим модулем. Щоб спростити, уявіть, що для передачі даних використовуються приймачі та передавачі з різними довжинами хвиль, скажімо, у видимому діапазоні, інфрачервоному та ультрафіолетовому. Якщо фотосенсор може «бачити» світло лише у видимому діапазоні, його можна використовувати для отримання інформації від джерела світла у видимому діапазоні, а не від інфрачервоного або ультрафіолетового.

Діапазон роботи модуля залежить від кількох факторів. Яку антену використовувати, чи є пряма видимість, як спрямовані антени модулів у просторі тощо. є важливим. Налаштування модуля впливають на бітрейт (для максимально можливої відстані 250 Кбіт), потужність передачі та чутливість приймача.

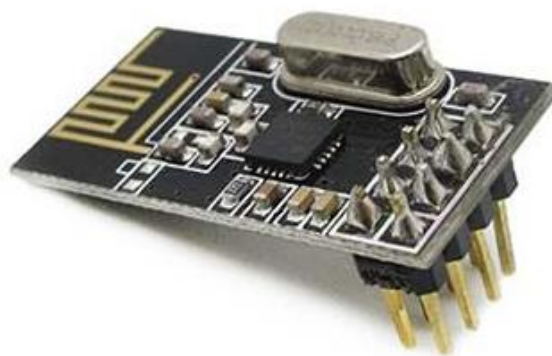


Рисунок 2.3 — Модуль зв'язку NRF24L01

Хоча частотний канал можна використовувати для псевдовипадкової настройки частоти, як правило, для цієї мети краще використовувати інші приймачі. Налаштування z nRF24l01 зазвичай використовує частотний канал, якщо немає певної частоти або перешкод від інших пристроїв. Це може бути WiFi, Bluetooth або інший пристрій. Для nRF24l01 також можуть використовуватися різні частоти, допустимо спрощення коду/алгоритму, протоколу, конфігурації, максимального бітрейту тощо. Два і більше модулів мають одну частоту, два і більше модулів мають іншу частоту тощо. Модулі, що працюють на різних частотах, не будуть заважати один одному.

Для зовнішніх модулів підсилювачів це зазвичай сотні метрів. Для згаданого раніше модуля міжрядковий інтервал може становити близько ста метрів. Якщо сигнал може відбиватися від стін або інших об'єктів у кімнаті, відстань (не по уявній прямій лінії від модуля до модуля, а по всьому шляху, по якому проходить радіосигнал) може становити десятки і більше метрів.

Оскільки не всі плати, екрани чи модулі Arduino мають живлення 3,3 В. Вищезгаданий модуль не має стабілізатора і може вийти з ладу при необхідності від 5В. Щоб цього не сталося, використовуйте додатковий стабілізатор потужності. Ви можете підключити його до наявного стабілізатора на макетній платі або використати спеціальний модуль, показаний на малюнку 2.4, який ми отримаємо для нашої системи.

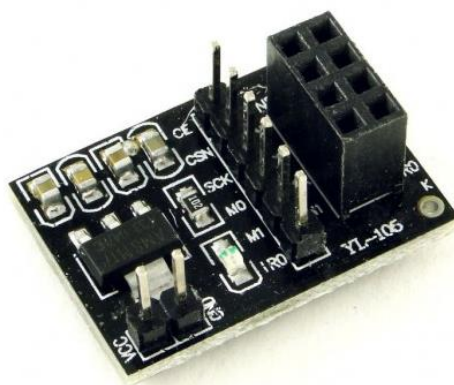


Рисунок 2.4 — Модуль живлення для nRF24L01

Використання інтерфейсу SPI може бути корисним, якщо вам потрібно використовувати програмне забезпечення інтерфейсу або якщо вам потрібно підключити кілька модулів nRF24L01 до однієї плати мікроконтролера.

Це може знадобитися в залежності від обраної топології мережі та кількості пристроїв у ній, характеристик реалізованого в коді протоколу та алгоритму роботи, пропускної здатності або кількості пристроїв в одномодульній мережі.

Використання кількох модулів може бути зручнішим для створення маршрутизатора/концентратора, якщо розділити його на підмережі для різних кімнат або завдань (збір даних із датчиків, індивідуальне керування освітленням тощо). У хабі/роутері самі модулі можуть бути різними — один модуль без підсилювача для збору даних з локальних датчиків або управління чимось, другий — це підсилювач для передачі локально отриманих даних або команд управління, спрямована антена і т.д. з віддаленого пристрою.

Окрім простоти та зручності, за допомогою кількох модулів можна створити можливість розподілу завдань на різні пристрої в майбутньому.

Цей модуль використовує протокол бездротової передачі сигналу ShockBurst, розроблений Northern Semiconductor. Цей протокол використовується для передачі даних у бездротових програмах, таких як бездротові датчики, системи збору даних і бездротові засоби керування.

Опис апаратного рівня обробки пакетів ShockBurst:

- формування пакета при його відправленні;
- виявлення та перевірка отриманих пакетів;
- налаштування джерела переривань для активації виходу IRQ;
- автоматичне відправлення зворотного повідомлення у відповідь на отриманий пакет;
- повторна передача втрачених пакетів (налаштовується кількість спроб у разі активації функції);

- гнучке налаштування довжини поля даних: від 0 до 32 байт у статичному режимі, або автоматичне визначення в динамічному;
- регульована довжина адреси від 3 до 5 байт;
- функція "MultiCeiver" для створення топології "зірка 1:6".

Налаштування ShockBurst є простим і гнучким. Захоплення частини функцій дозволяє спростити код і роботу з модулем, а також робить мікроконтролер/мікропроцесор плати управління дещо легшим і дозволяє модулю переходити в режим енергозбереження під час транспортування. очікування або обробка пакета.

Формат пакету:

- преамбула, 1 байт;
- адреса, 3-5 байт;
- поле контролю пакета, 9 біт;
- дані, 0-32 байт;
- контрольна сума, 1-2 байти.

Інформація в полі заголовка автоматично форматується на основі значення першої частини адреси.

Структура поля контролю пакета:

- поле даних: 6 біт, використовується лише в динамічному режимі. Приймає значення від 0 до 33 (де 33 означає, що довжина поля даних не враховується);
- ідентифікатор пакета: 2 біти. У передавачі значення цього поля автоматично збільшується після отримання нового пакета через SPI. У приймачі ідентифікатор дозволяє визначити, чи є пакет новим, чи це повторна відправка;
- прапор NO_ACK ("немає підтвердження"): 1 біт. Використовується лише при ввімкненій функції автоматичного підтвердження.

Поле даних може бути від 1 до 32 байтів. Довжина поля 0 може бути використана для перевірки наявності радіомодуля з певною адресою.

Прапорець використовується для визначення того, отримано пошкоджений пакет чи ні.

У режимі прийому модуль очікує спочатку заголовок, а потім адресу. Прийняття наступних полів пакета відбувається після успішної перевірки того, що заголовок відповідає першій частині адреси та що така адреса знайдена в регістрах `RX_ADDR`. Після отримання інших полів пакету перевіряється контрольна сума, і в разі успіху дані записуються в буфер FIFO одержувача, з якого контролер моніторингу вже може читати.

Якщо функцію автоматичної автентифікації ввімкнено, модуль у режимі отримання надішле пакет автентифікації після отримання пакета.

Якщо функцію надсилання ввімкнено, модуль намагатиметься надіслати пакет, доки не отримає пакет із підтвердженням повернення або кількість повторних спроб не збільшиться. У реєстрі `SETUP_RETR` ви можете налаштувати кількість повторних спроб (до 15 або 0, якщо потрібно вимкнути) і час очікування пакета підтвердження (від 250 до 4000 мкс, з кроком 250 мкс).

NRF24L01 може працювати в чотирьох режимах:

- Power Down;
- Standby;
- RX;
- TX.

У режимі вимкнення живлення, коли nRF24L01 знаходиться на мінімальній потужності, усі блоки відключені, крім SPI та регістрів.

Режими Standby-I та Standby-II є енергозберігаючими, при цьому тактовий генератор залишається активним, на відміну від режиму Power Down. Хоча це підвищує споживання енергії, такі режими дозволяють швидше переходити до режимів RX або TX. Standby-II відрізняється від Standby-I тим, що включає додаткові тактові буфери, що підвищують енергоспоживання.

У жодному з режимів не можна одночасно передавати та отримувати дані. Тип

зв'язку, що підтримує одночасну передачу та прийом, називається дуплексним. NRF24L01 використовує напівдуплексний режим з поділом часу для двостороннього зв'язку, де модуль передачі даних або приймає, або передає дані в кожен момент часу.

NRF24L01 можна налаштувати на 250 Кбіт/с, 1 Мбіт/с і 2 Мбіт/с. Це максимально можлива швидкість, тобто вона розподілена між усіма покупцями.

Зіткнення виникає, коли кілька вузлів передачі намагаються передавати одночасно. У разі конфлікту одержувач може отримати інформацію та має надіслати її повторно.

У режимі приймача ("первинний приймач", PRX, первинний приймач) можна лише отримувати дані. Коли пакет знайдено, він перевіряється, і в разі успіху дані записуються в приймальний буфер. Якщо ShockBurst вимкнено, режим PRX активний, доки контролер не перейде в інший режим.

У режимі передавача ("первинний передавач", PTX, первинний передавач) можна лише надсилати дані. Якщо на виводі CE низький рівень, nRF24L01 переходить в режим очікування. Якщо висновок CE високий, наступна дія залежить від стану буфера. Якщо в буфері є дані для надсилання, надсилається наступний пакет. Якщо буфер порожній, nRF24L01 переходить у режим очікування II.

Якщо функцію ShockBurst увімкнено, nRF24L01 може автоматично перемикається в інші режими. У PTX, якщо необхідно, він автоматично перемикається на PRX і чекає пакета підтвердження. PRX переключиться на PTX, якщо необхідно надіслати підтвердження або повторно надіслати пакет.

MultiCeiver можна використовувати в режимі PRX. MultiCeiver є скороченням від «Multiple Sender Single Receiver», що перекладається як «Один приймач із кількома відправниками».

Для прийому можна використовувати до 6 каналів, дані можуть прийматися тільки від одного відправника одночасно. Які канали вмикаються чи вимикаються, налаштовується в реєстрі EN_RXADDR (канали 0 і 1 увімкнено за замовчуванням). Кожна карта має унікальну адресу, яка налаштовується за допомогою реєстрів

RX_ADDR_P0-RX_ADDR_P5.

Таблиця 2.2 — Технічні характеристики nRF24L01

Технічні характеристики	Значення
Швидкість передачі даних:	до 2 Мбіт
Інтерфейс підключення:	SPI
Каналів зв'язку:	125
Живлення:	Від 1.9В до 3.6 В
Частота зв'язку:	2.4 ГГц
Модуляція:	GFSK
Вихідна потужність:	0, -6, -12 чи -18 dBm

2.2.3 Мікроконтролер ATmega328

Важливим компонентом на платі Arduino Nano є 8-розрядний мікроконтролер Atmega328. Він має вдосконалену архітектуру RISC і 131 інструкцію. Передбачається, що кожна окрема інструкція повинна бути виконана за мінімальний проміжок часу, тобто один або більше тактів мікропроцесора. Удосконалена архітектура RISC, запозичена з ARM для об'єднання простих інструкцій, які найчастіше використовуються, у складні інструкції. Від такого простого набору команд залежить висока продуктивність. Крім того, мікроконтролер здатний працювати на високих частотах, і програміст може легко розрахувати час виконання кожної частини коду під час написання програмного забезпечення.

Atmega328 має 32 8-розрядних регістри загального призначення для тимчасового зберігання даних. Старші 6 регістрів являють собою пари з трьох 16-розрядних регістрів, X, Y, Z, які представляють непряму адресу регістрів RAM під час запису даних.

Цікаво, що перша половина файлу регістрів (реєстри загального призначення) підтримує лише такі схеми введення/виведення для максимальної продуктивності:

- один 8-розрядний вхідний операнд і один 8-розрядний вихідний результат;
- два 8-розрядних вихідних операнди і один 8-розрядний вхідний;
- два 8-розрядних вихідних і один 16-розрядний вхідний результат;
- один 16-розрядний вихідний результат і один 16-розрядний вхідний операнд.

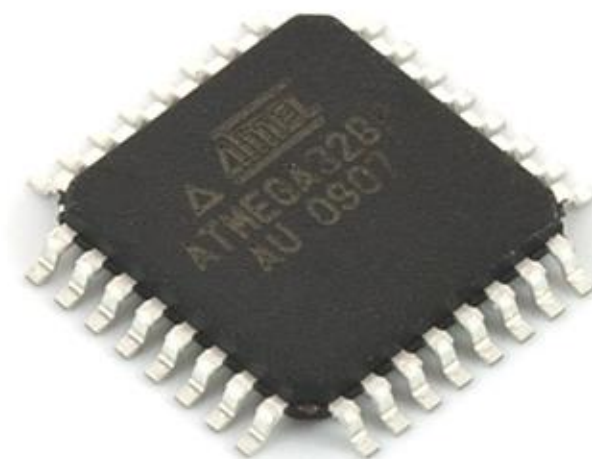


Рисунок 2.5 — Мікроконтролер Atmega328

ЦП містить 64 регістри вводу-виводу, які керують портами вводу-виводу, таймерами, перериваннями, стеком, сторожовим таймером і багатьма іншими функціями. Запис до регістрів вводу/виводу виконується за допомогою доступу до кодової адреси за допомогою команд SBI та CBI або команд IN, OUT для доступу до абсолютних адрес (адреса 2016, менша за абсолютну). 160 розширених регістрів введення/виведення та 2048 байт внутрішньої пам'яті SRAM підтримують непряме посилання. На малюнку 2.5 показано розташування внутрішньої пам'яті Atmega328.

1 кілобайт пам'яті даних EEPROM (EPROM) може витримувати до 100 000 циклів запису/перезапису.

Мікроконтролер має три внутрішніх таймера: два 8-бітних і один 16-бітний, внутрішні та зовнішні переривання на фронті та спаді входних імпульсів, а також дільники тактової частоти 2, 4, 8, 16, що забезпечує неймовірну гнучкість при проектуванні високоефективних детекторних систем. На рисунку 2.6 зображена структура робочих регістрів Atmega328.

	7	0	Addr.	
	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
General	R16		0x10	
Purpose	R17		0x11	
Working	...			
Registers	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Рисунок 2.6 — Структура робочих регістрів Atmega328

Data Memory	
32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Registers	0x0060 - 0x00FF
Internal SRAM (1048 x 8)	0x0100
	0x08FF

Рисунок 2.7 — Організація адрес пам'яті Atmega328

8 каналів, 10 біт АЦП і ЦАП, вбудований аналоговий компаратор, 6 каналів ШІМ, можливість роботи з послідовним інтерфейсом SPI, двопровідний інтерфейс I2C добре працює для обробки аналогових сигналів, кодування. кодування та взаємодія з іншими зовнішніми пристроями.

Завдяки діапазону робочих температур від мінус 40 °C ... плюс 125 °C цей мікроконтролер відповідає вимогам для нормальної роботи другого РЛС.

AVR використовують гарвардську архітектуру з окремими шинами пам'яті та даних для досягнення високої продуктивності та паралелізму. Поки виконується одна інструкція, наступна інструкція зчитується з пам'яті. Кожна інструкція переходить до наступного етапу виконання після того, як попередня вже завершена. Таким чином, як зазначалося вище, виконання інструкцій досягається практично за один цикл роботи мікропроцесора.

Але для зручності розробки передсерійного пристрою, варто скористатися вже готовою платою розширення через ряд параметрів:

— немає потреби в додаткових паяльних роботах для підключення зовнішнього кварцового резонатора (робоча частота 16 МГц для Atmega328 забезпечується тільки зовнішнім генератором) і в додатковому захисті від перепадів напруги живлення (що є важливим для вбудованих систем радіолокації), а також у використанні інших рекомендованих згідно з документацією до мікроконтролера «підтягуючих» резисторів;

— немає необхідності витрачати гроші на програматор, час на його підключення та встановлення драйверів;

— плата розширення є багатофункціональною, портативною платформою для експериментальної розробки пристроїв, що значно спрощує і прискорює роботу.

Серед платформ для пристроїв AVR, однією з найпопулярніших новинок є продукт торгової марки Arduino з мікроконтролером Atmega328 — Arduino Nano. Ця плата розширення повністю відповідає всім зазначеним вимогам.

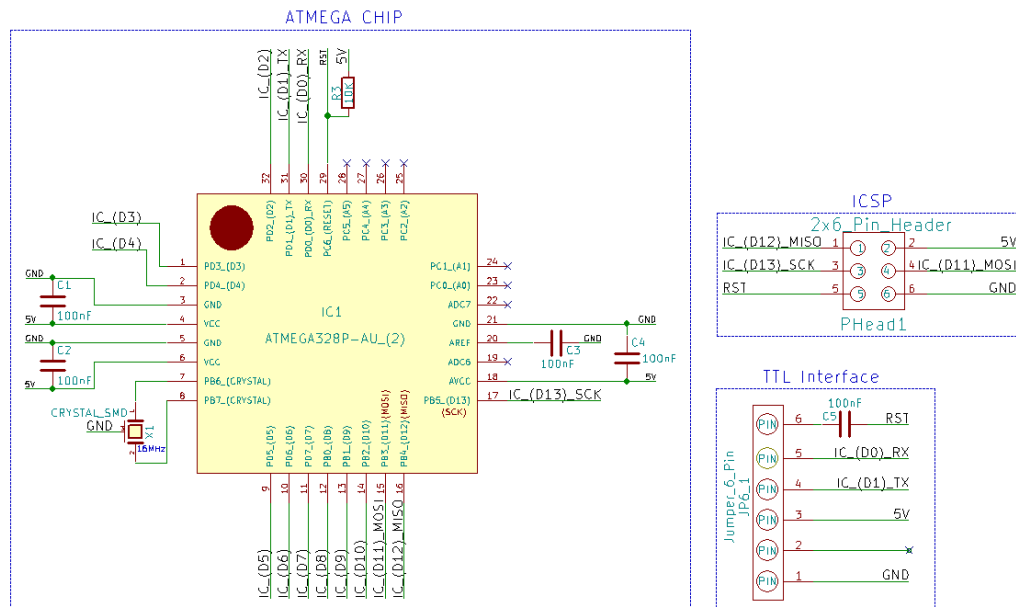


Рисунок 2.8 — Схема мікроконтролера Atmega328

Arduino Nano — одна з найменших плат Arduino. Повний аналог Arduino Uno - теж працює на чіпі ATmega328 (ще можна знайти варіанти з ATmega168), але меншого форм-фактора. Завдяки своїм загальним розмірам дошка часто використовується в проектах, де важлива компактність.

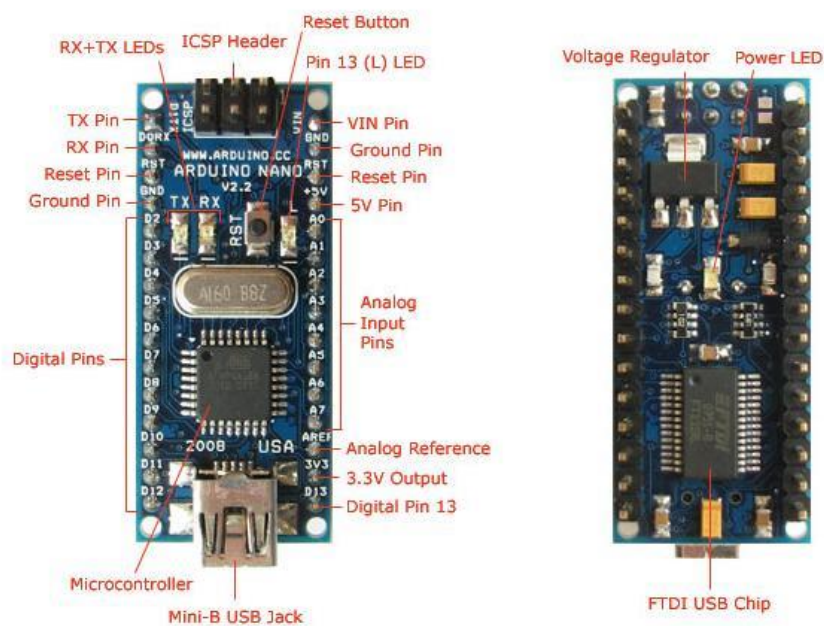


Рисунок 2.9 — Arduino Nano

Плата не має спеціальної зовнішньої розетки живлення, Arduino працює через USB (miniUSB або microUSB). В іншому параметри відповідають моделі Arduino Uno.

Таблиця 2.3 — Технічні характеристики Arduino Nano

Технічні характеристики	Значення
Мікроконтролер	Atmega328
Робоча напруга	5В
Напруга живлення (рекомендована)	Від 7В до 12В
Цифрові піни	14
Аналогові піни	8
Максимальний струм	40 мА
Flash-пам'ять	32 КБ
SRAM	2 КБ
EEPROM	1 КБ
Тактова частота	16 МГц
Розміри	1.85 см x 4.3 см

2.2.4 Мікроконтролер ATmega328P

ATmega328 і ATmega328P включають архітектуру мікроконтролера Atmel AVR, яка зазвичай використовується в популярних платах Arduino (таких як Arduino Uno).

Щодо основних відмінностей, Р у ATmega328P означає *power*, технологію виробництва Atmel, яка забезпечує високу енергоефективність. Це дозволяє мікроконтролерам "P" споживати менше енергії та бути більш придатними для акумулятора та пристроїв, що живляться від акумулятора.

Ще одна важлива відмінність полягає в тому, що деякі програми завантажувача Arduino можна налаштувати спеціально для ATmega328P і не працюватимуть з ATmega328.

Загалом, ATmega328 і ATmega328P дуже схожі та взаємозамінні для багатьох програм.

Опис порівняння мікроконтролерів:

- обидва мікроконтролери мають 32 КБ Flash пам'яті, 1 КБ EEPROM, 2 КБ SRAM, 23 лінії загального призначення I/O, 32 загального призначення регістри робочого простору, вбудований кварцевий генератор, а також таймери, АЦП, інтерфейси SPI та I2C тощо;

- ATmega328P оснащений технологією "picoPower", що робить його більш енергоефективним порівняно з ATmega328;

- ATmega328P підтримує режим BOD (Brown-out Detection), що дозволяє йому автоматично вимикатися при зниженні напруги, тоді як ATmega328 також має цю функцію, але вона може бути відсутньою в деяких версіях;

- ATmega328P може працювати в режимі з низьким споживанням енергії при зниженої швидкості, в той час як ATmega328 не підтримує таку можливість;

- обидва мікроконтролери можуть використовувати різні версії бутлоадера Arduino, тому ATmega328P може бути несумісним з деякими версіями програмного забезпечення, розробленими для ATmega328.

Враховуючи всі ці відмінності, ATmega328P вважається більш «преміальною» версією, ніж ATmega328, забезпечуючи більшу енергоефективність і гнучкість.

Arduino Uno є серцем екосистеми Arduino. Це основна плата розробки, яка використовується для більшості проектів Arduino. У порівнянні з Arduino Nano, Uno більший і пропонує більше місця для контактів та інших компонентів.

Uno має роз'єм USB-B для програмування та зв'язку з комп'ютером, і якщо Nano використовує Mini-USB, Arduino Uno можна жити від зовнішнього джерела живлення через цей USB-порт або роз'єм постійного струму, що робить його більш гнучким. різні сценарії живлення.

Плата може житись від USB або зовнішнього джерела живлення — тип джерела вибирається автоматично.

Адаптер змінного/постійного струму або батарею/акумулятор можна використовувати як зовнішнє джерело живлення (не USB). Штекер адаптера (діаметр

— 2,1 мм, центральний контакт — плюсовий полюс) необхідно вставити у відповідний роз'єм живлення на платі. У разі живлення від батареї дроти повинні бути підключені до клем Gnd і Vin роз'єму POWER.

Напруга зовнішнього джерела живлення може бути від 6 до 20 В, але зниження напруги живлення нижче 7 В призводить до зниження вихідної напруги, що може стати причиною нестабільної роботи пристрою. Використання напруги понад 12 В може призвести до перегріву стабілізатора напруги та несправності плати. Зважаючи на це, рекомендується використовувати джерело живлення з напругою від 7 В до 12 В.

Uno має велику міцну піну, яка витримує численні з'єднання та з'єднання. Це робить Uno вибором для проектів, які потребують частої заміни обладнання. Крім того, Uno має вбудований завантажувач, який спрощує процес завантаження програмного коду на плату.

Uno також може використовувати екрани, які є додатковими платами, які приєднуються до Uno, розширюючи можливості плати без необхідності паяти багато деталей або окремо.

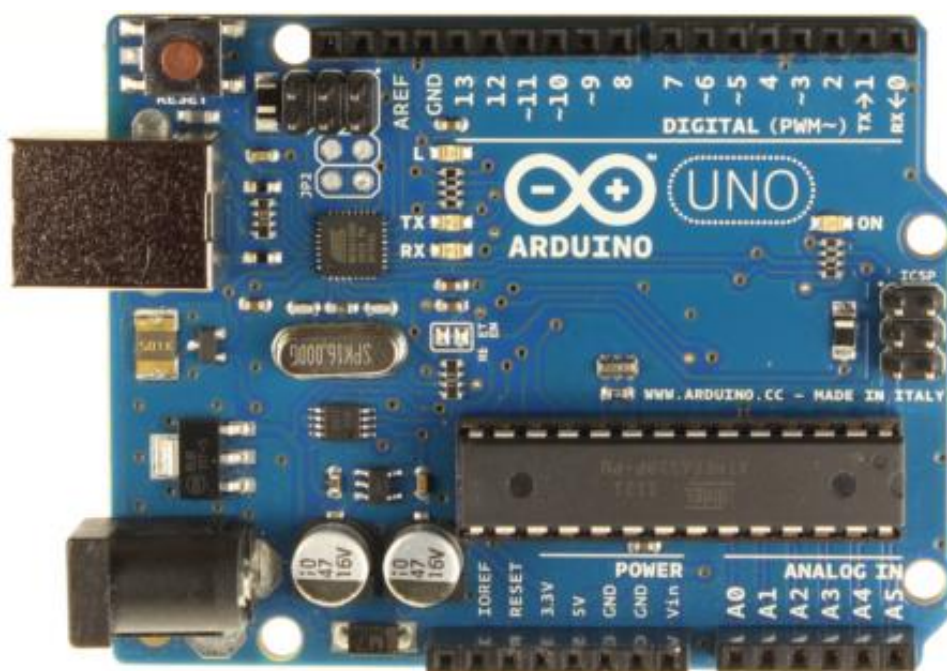


Рисунок 2.10 — Вигляд Arduino Uno зверху

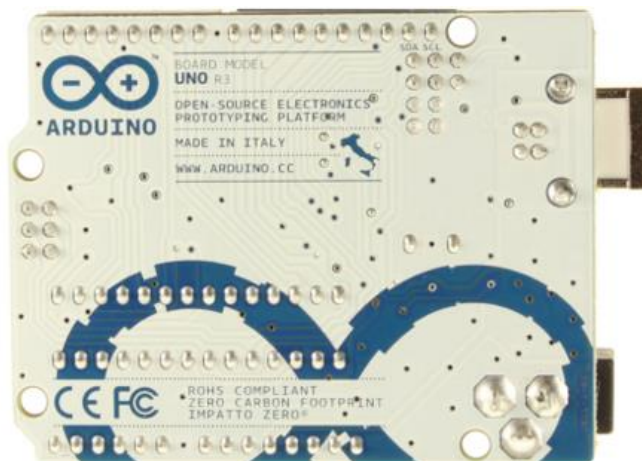


Рисунок 2.11 — Вигляд Arduino Uno знизу

Arduino Uno має змінні запобіжники, які захищають USB-порт комп'ютера від короткого замикання та перевантаження. Хоча більшість комп'ютерів мають власний захист, такі запобіжники забезпечують додатковий рівень захисту. Якщо через порт USB споживається струм понад 500 мА, запобіжник автоматично вимкнеться, доки не буде усунуто причину короткого замикання або перевантаження.

Таблиця 2.4 — Технічні характеристики Arduino Uno

Технічні характеристики	Значення
Мікроконтролер	Atmega328P
Напруга живлення	5В
Напруга живлення (рекомендована)	Від 7В до 12В
Цифрові піни	14 (з них 6 ШИМ-пінів)
Аналогові піни	6
Максимальний струм	40 мА
Максимальний струм 3.3V	50 мА
Flash-пам'ять	32 КБ
SRAM	2 КБ
EEPROM	1 КБ
Тактова частота	16 МГц
Розміри	6.9 см x 5.4 см

2.2.5 Wi-Fi-модуль ESP8266 ESP-01

ESP8266 ESP-01 — це модуль мікроконтролера Wi-Fi на основі мікросхеми ESP8266 від Espressif Systems. Цей чіп дозволяє пристроям IoT підключатися до бездротових мереж Wi-Fi і стає дуже популярним у спільноті розробників IoT. Модуль ESP-01 найбільш відомий своєю компактністю та низькою ціною, що робить його ідеальним для систем з обмеженим простором.

Однією з найбільших переваг ESP8266 ESP-01 є підтримка Wi-Fi, що робить його одним із найзручніших модулів для підключення пристроїв до Інтернету. Це відрізняє його від багатьох інших модулів, які не мають вбудованої підтримки Wi-Fi. ESP-01 також можна програмувати безпосередньо через Arduino IDE, що робить його доступним для широкого кола розробників.

Модуль ESP-01 має обмежену кількість контактів GPIO — лише два в порівнянні з багатьма іншими модулями ESP8266. Однак це робить його більш компактним і його легше встановити в обмеженому просторі.

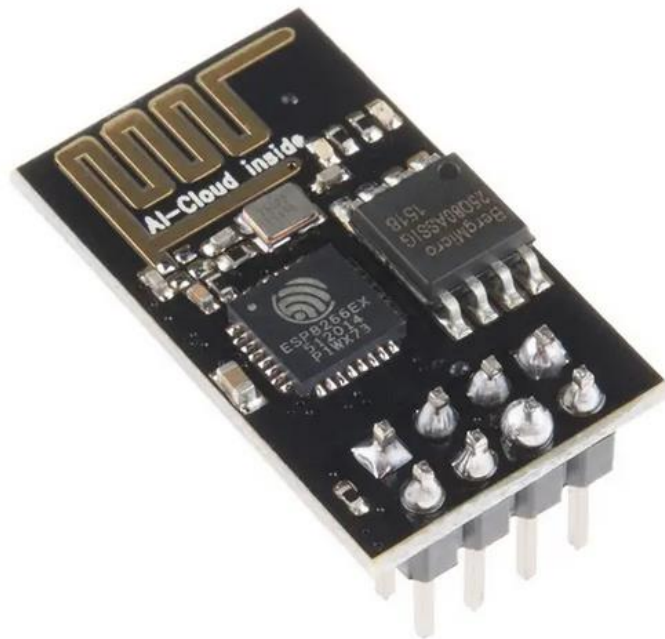


Рисунок 2.12 — ESP8266 ESP-01

Таблиця 2.5 — Технічні характеристики ESP8266 ESP-01

Технічні характеристики	Значення
Підтримка стандартів	802.11 b/g/n.
Діапазон частот:	Від 2.4 GHz до 2.5 GHz.
Мікроконтролер:	Tensilica L106
Пам'ять Flash:	1MB.
Робоча напруга:	Від 3.0V до 3.6V.
Режими роботи:	STA/AP/STA+AP.
Інтерфейси:	UART, GPIO.
Кількість GPIO	2.
Розміри:	1.43 см. x 2.48 см.

Tensilica L106 — це сімейство 32-розрядних мікроконтролерів, розроблених Tensilica, підрозділом Cadence Design Systems. Ці мікроконтролери відрізняються високою продуктивністю і низьким енергоспоживанням.

Мікроконтролер L106 використовує ядро Xtensa LX106, яке включає пряму керуючу шину та RISC-архітектуру, що дозволяє гнучко конфігурувати систему для конкретних потреб.

Цей мікроконтролер є ядром чіпа ESP8266 від Espressif Systems, який широко використовується в модулях Wi-Fi, таких як ESP-01. Він дозволяє модулям обробляти дані, отримувати та надсилати пакети даних через Wi-Fi, а також виконувати спеціальні сценарії та програми.

2.2.6 Екран LCD 1602 I2C

LCD 1602 I2C — це 16-символьний 2-рядковий буквено-цифровий модуль із підтримкою інтерфейсу I2C. I2C (інтегральна схема) — це двоканальний протокол зв'язку, який дозволяє підключати кілька пристроїв до мікроконтролера через два дроти (SDA — дані, SCL — годинник), що значно зменшує кількість необхідних контактів GPIO.



Рисунок 2.13 — Экран LCD 1602 та I2C

Таблиця 2.6 — Технічні характеристики LCD 1602 I2C

Технічні характеристики	Значення
Напруга живлення:	5V.
Інтерфейс підключення:	I2C.
Кількість символів:	16x2(32).
Підсвічування:	LED, синього кольору.
Контролер:	HD44780.
Кількість пінів:	4 (VCC, GND, SDA, SCL).
Розміри:	8 см x 3.6 см x 2 см.

Екран 1602 I2C LCD оснащений контролером HD44780 або еквівалентом, який є стандартним для цього типу дисплея. Він дозволяє відображати текст і прості символи,

що робить його широко використовуваним у всіх проектах, які вимагають простого виведення тексту, включаючи вбудовані системи, роботи, проекти DIY тощо.

У цьому дисплеї використовується підсвічування, яке керується ШІМ-сигналом, який дозволяє регулювати яскравість. Крім того, зазвичай є потенціометр для регулювання контрастності екрану.

2.3 Функціональні схеми пристроїв системи

Відповідно визначеним типам компонентів і спроектованої структурної схеми у підрозділах 2.1 та 2.2, було розроблено та спроектовано функціональну схему датчика (рисунок 2.15) та блоку управління системи (рисунок 2.14).

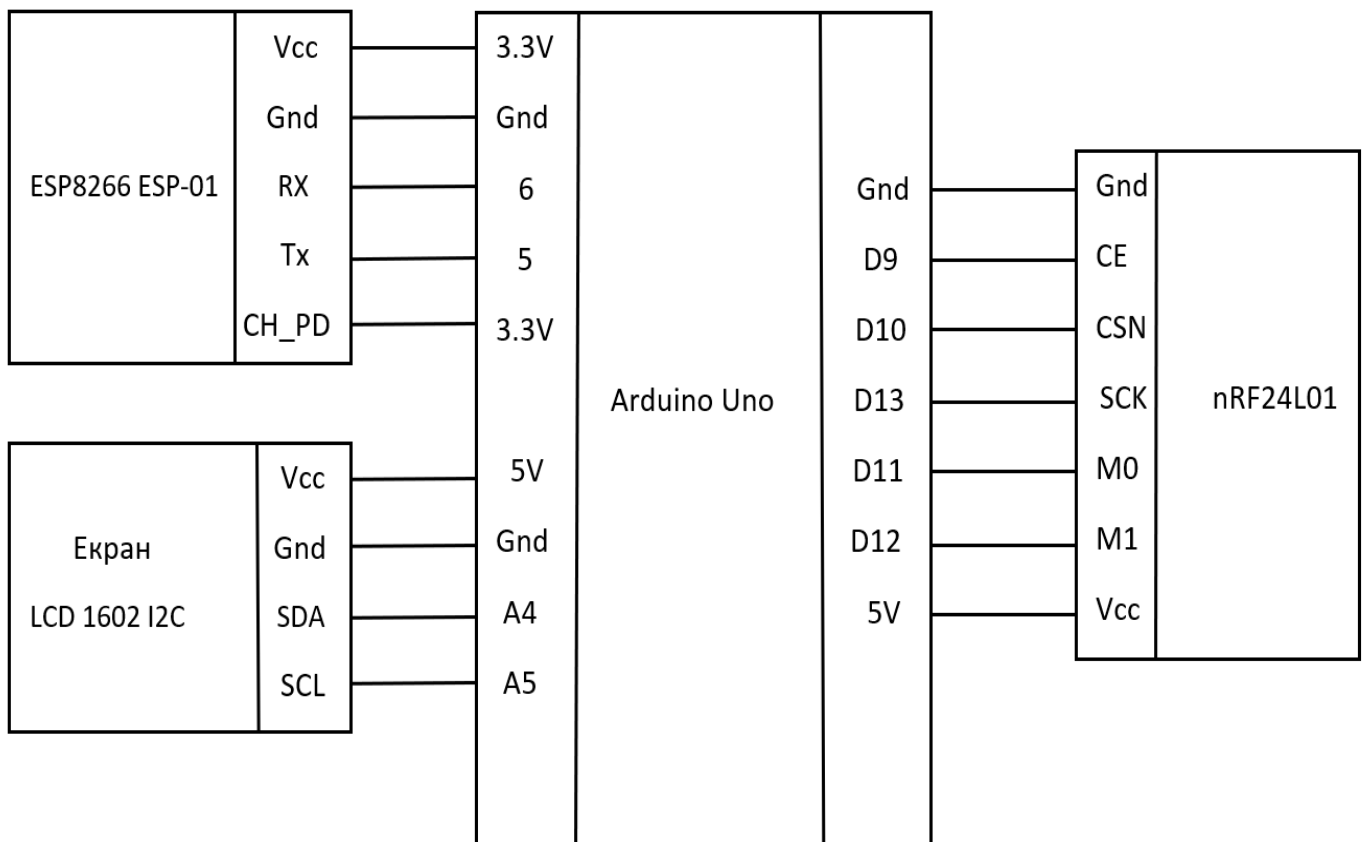


Рисунок 2.14 — Функціональна схема блоку управління системи

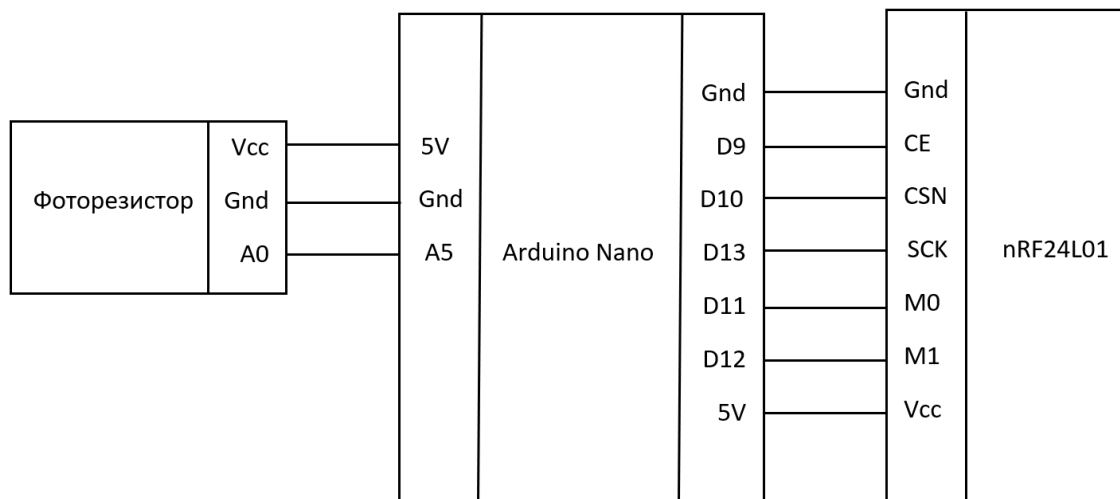


Рисунок 2.15 — Функціональна схема датчика системи

2.4 Розробка електричної принципової схеми

Оскільки головними платами в даній системі є Arduino Uno та Arduino Nano, то до них будуть підключатись модулі та елементи.

До Arduino Uno буде підключено:

- Wi-Fi-модуль ESP8266 ESP-01;
- модуль зв'язку nRF24L01;
- екран LCD 1602 I2C;
- бузер.

До Arduino Nano буде підключено:

- модуль зв'язку nRF24L01;
- модуль фоторезистора.

Оскільки для підключення дисплея до плати потрібна велика кількість вільних контактів, для економії місця на платі було вирішено вибрати дисплей з інтерфейсом I2C. Таким чином, для підключення дисплея потрібно всього чотири контакти - два для живлення і два для виходу, як показано в таблиці 2.9.

В таблицях 2.7 - 2.10 показано підключення елементів до плати Arduino Uno, а в таблицях 2.11 - 2.12 підключення елементів до плати Arduino Nano.

Таблиця 2.7 — Підключення Wi-Fi-модуля ESP8266 ESP-01

Контакти модуля	Контакти плати
RX	6
TX	5
CH_PD	3.3V
Gnd	Gnd
Vcc	3.3V

Таблиця 2.8 — Підключення модуля зв'язку nRF24L01

Контакти модуля	Контакти плати
CE	D9
CSN	D10
SCK	D13
M0	D11
M1	D12
Gnd	Gnd
Vcc	5V

Слід зазначити, що комунікаційний модуль підключається до джерела живлення комунікаційного модуля nRF24L01, що забезпечує зниження напруги для стабільної роботи. Сам модуль кріпиться до елемента живлення на захищених входах, що забезпечує надійну передачу сигналу. Таке розташування дозволяє швидко замінити модуль nRF24L01 без допоміжних інструментів і без пайки в разі поломки.

Підключення дисплея через згаданий вище інтерфейс I2C дозволяє зменшити кількість цифр, необхідних для підключення дисплея до плати, що дозволяє розширити можливості та функціональність проектів.

Підключення модуля зв'язку nRF24L01 до плати Arduino Nano відбувається за такою ж схемою та таблицею як і до Uno.

Таблиця 2.9 — Підключення екрану

Контакти модуля	Контакти плати
SDA	A4
SCL	A5
Gnd	Gnd
Vcc	5V

Таблиця 2.10 — Підключення бузера

Контакти модуля	Контакти плати
I/O	8
Gnd	Gnd
Vcc	5V

Таблиця 2.11 — Підключення модуля зв'язку nRF24L01

Контакти модуля	Контакти плати
CE	D9
CSN	D10
SCK	D13
M0	D11
M1	D12
Gnd	Gnd
Vcc	5V

Таблиця 2.12 — Підключення бузера

Контакти модуля	Контакти плати
A0	A5
Gnd	Gnd
Vcc	5V

3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Розробка основного алгоритму роботи системи

Відповідно до завдання та основного набору функцій модель матиме такий алгоритм функціонування, як показано на рисунку 3.1.



Рисунок 3.1 — Блок-схема роботи системи

Словесний алгоритм роботи системи по кроках:

1) отримання значень, отриманих від датчика;

2) порівняти ці значення з попередніми значеннями;

3) якщо значення відрізняються, але не перевищують вказаний поріг, ми повертаємося до кроку 1 цього словесного алгоритму, якщо значення відрізняються значно і перевищують вказаний поріг, ми переходимо до кроку 4;

4) відкриваємо звукові та візуальні сигнали попередження, тобто надсилаємо запит про порушення на сервер і виводимо інформацію про порушення на екран;

5) дочекатися введення пароля;

6) якщо користувач не ввів пароль, попередження продовжує працювати, якщо пароль було введено, попередження очищається шляхом відображення інформації на екрані та надсилання запиту на скасування на сервер.

3.2 Розробка алгоритму зміни пароля

Основною метою системи є забезпечення надійності та безпеки, і для досягнення цього в систему була додана можливість змінити пароль.

Коли система підключена до джерела живлення, її робота починається з підключення до мережі Wi-Fi і запуску всіх модулів. Відтепер, утримуючи кнопку входу протягом трьох секунд, ви зможете змінити пароль, але лише якщо ви знаєте поточний системний пароль.

Таким чином, він захищає систему від зломисників, які спостерігають за тим, як користувач вводить пароль.

Вигляд алгоритма словами по кроках:

1) натиснення та утримання кнопки підтвердження під час роботи системи;

2) виводиться запис про початок процесу зміни пароля;

3) з'являється та вводиться поле для введення пароля, введення пароля здійснюється кнопкою введення та підтвердження кнопкою підтвердження;

4) перевірка правильності поточного пароля.

- 5) якщо пароль неправильний, повернутися до введення зазначеного пароля, якщо пароль правильний, перейти до кроку 6;
- 6) введення нового пароля;
- 7) зберегти новий пароль і вивести інформацію про нього.



Рисунок 3.2 — Блок-схема алгоритму зміни пароля

3.3 Вибір засобів для програмування мікроконтролера

Середовищем для програмування мікроконтролера була обрана програма Arduino IDE.

Arduino IDE — це середовище програмування, яке використовує C++ для програмування всіх плат Arduino.

IDE означає інтегроване середовище розробки. За допомогою цього середовища програмісти пишуть програми швидше та зручніше, ніж за допомогою традиційних текстових редакторів, але його також можна використовувати для написання програмного коду.

Arduino IDE дозволяє писати програми для зручного текстового редактора, компілювати їх у машинний код і завантажувати на всі версії плати Arduino. Програма абсолютно безкоштовна і її можна завантажити з офіційного сайту Arduino Community.

Інтерфейс Arduino IDE легше вивчити, оскільки його основою є C++, функціональна мова програмування. Для програмування Arduino буде використовуватися спрощена версія мови C++. Як і в інших мовах програмування, таких як C, існує кілька правил написання коду.

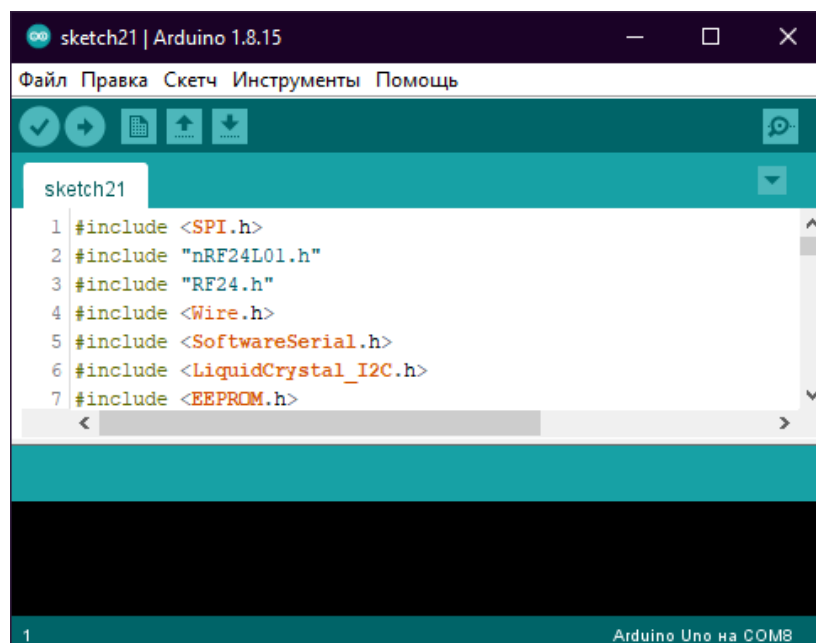


Рисунок 3.3 — Вікно програми

Arduino IDE також має широкий спектр підтримуваних плат: Arduino UNO, Arduino Nano, ESP32, ESP8266, сімейство мікроконтролерів Attiny тощо. Ви можете підключити будь-яку плату підтримки за допомогою вбудованого контролера плати Arduino IDE, показаного на малюнку 3.4.

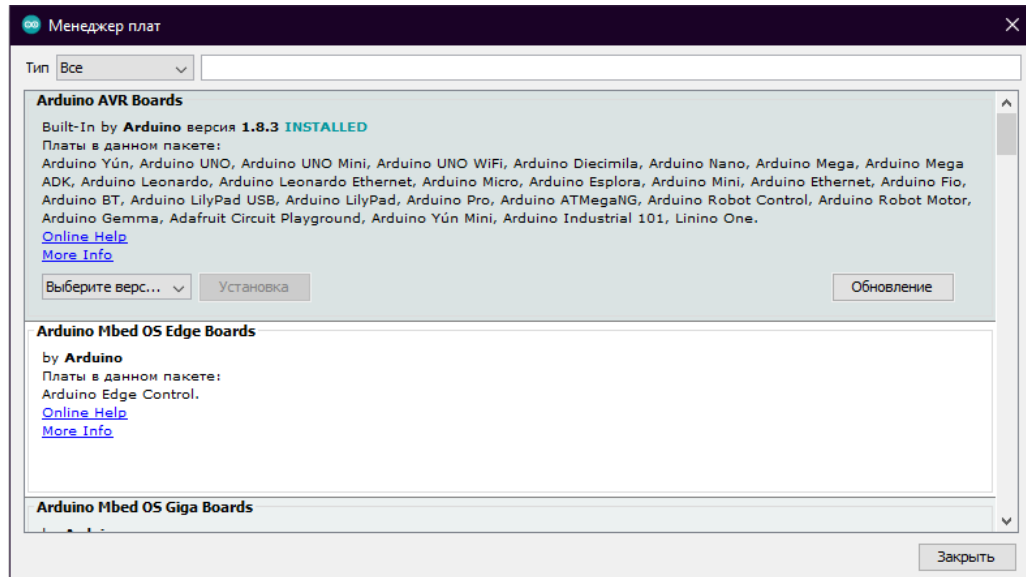


Рисунок 3.4 — Вікно менеджера плат для Arduino IDE

3.4 Програмування мікроконтролерів

3.4.1 Програмування блоку датчиків

При проектуванні та впровадженні електронних систем на основі мікроконтролерів одним із найважливіших аспектів є процес програмування. Цей процес передбачає не лише розробку програмного коду, а й розуміння роботи та архітектури мікроконтролерів, а також їх взаємодії з периферійними пристроями.

Далі ми детально розглянемо код програми для нашого мікроконтролера, включаючи його основні частини та функціональність.

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
```

У цих стрічках ми оголошуємо бібліотеки, які використовуватимуться для програмування мікроконтролера. А саме бібліотеки SPI, nRF24L01 і RF24. Бібліотека SPI використовується для полегшення програмування та зв'язку з пристроями, які використовують протокол SPI. Він надає набір функцій і методів, які програмісти можуть використовувати для надсилання та отримання даних від SPI. Дві інші бібліотеки призначені для інтерфейсу мікроконтролера з модулем nRF24L01.

```
const int sensorPin = A5; // пін для датчика світла
int sensorValue = 0; // зберігає останнє зчитане значення датчика світла
bool lastState = false; // зберігає стан світла з попереднього зчитування
RF24 radio(9, 10); // "створити" модуль на пінах 9 і 10
byte address[] = "6Node"; // адрес приймача
byte dataToSend = 1; // дані, які потрібно відправити
```

У стрічках вище ми оголошуємо змінні, які будуть використовуватися в кодуванні, точки підключення модулів або інших елементів, в нашому випадку це модуль фоторезистора і модуль nRF24L01.

```
void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.setAutoAck(1);
  radio.setRetries(0, 15);
  radio.enableAckPayload();
  radio.setPayloadSize(32);
  radio.openWritingPipe(address);
  radio.setChannel(0x6e);
  radio.setPALevel (RF24_PA_MAX);
  radio.setDataRate (RF24_250KBPS);
  radio.powerUp();
}
```

```
radio.stopListening();
}
```

У цій частині коду, тобто функції налаштування, ініціалізуються елементи та модулі, підключені до мікроконтролера, задаються параметри або параметри системи, її слід використовувати лише один раз під час запуску пристрою.

```
void loop() {
  radio.powerUp();
  sensorValue = analogRead(sensorPin); // зчитування значення з датчика світла
  bool currentState = (sensorValue < 300); // визначення поточного стану світла
  if (currentState != lastState) { // перевірка на зміну стану світла
    lastState = currentState;
    if (!currentState) { // якщо світло вимкнулось
      radio.write(&dataToSend, sizeof(dataToSend)); //відправка даних на інший
модуль
    }
  }
  delay(100); // затримка для стабільної роботи з датчиком
}
```

У головній функції циклу, крім програмування користувачем мікроконтролера на очікування сигналів і виконання дій, відбуваються основні процеси та дії. У нашому випадку в цій функції ми перевіряємо значення датчика світла, тобто фоторезистора, тоді, якщо значення датчика менше порогового значення, буде надіслано повідомлення зі змінною. до відділу адмін. Це буде служити вказівкою на те, що периметр перетнули.

3.4.2 Програмування блоку управління системи

В даному підрозділі ми оглянемо не весь код та його функції для мікроконтролера, а лише найголовніші його частини, які потрібні для основної задачі

пристрою.

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Wire.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
```

Ці стрічки містять усі необхідні бібліотеки для роботи нашої системи. Ми розглянули перші три бібліотеки в попередньому розділі, а також розглянемо наступні. Провід призначений для роботи з інтерфейсом I2C, який використовується для підключення до дисплея. У нашому випадку SoftwareSerial призначений для взаємодії мікроконтролерів з модулем ESP8266 ESP-01, який буде відправляти запити на сервер. Бібліотека LiquidCrystal_I2C використовується для роботи з дисплеями, які мають існуючий інтерфейс I2C. Остання EEPROM, ця бібліотека призначена для роботи з пам'яттю плати, де буде зберігатися пароль.

```
RF24 radio(9, 10);
byte address[6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"};
const int ledPin = 7;
const int ledPin1 = 3;
const int ledPin2 = 14;
const int inputButtonPin = 2;
const int confirmButtonPin = 4;
const int buzzerPin = 8;
SoftwareSerial esp8266(6, 5); // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Стрічки вище містять опції для всіх модулів і елементів нашого пристрою, таких як кнопки, світлодіодні індикатори, дисплей і зумер. Також встановлюється адреса для отримання повідомлень модуля nRF24L01.

```
byte message;
int password = 5; // пароль за замовчуванням
```

Далі йде опис змінної для отримання повідомлення від іншого модуля nRF24L01, яке служить сигналом для обрізання периметра. Потім оголошується змінна для значення системного пароля.

```
String server = "13.48.132.106";
String url = "/event/on/test11";
String postRequest = "POST " + url + " HTTP/1.1\r\n" +
    "Host: " + server + ":3000\r\n\r\n";
String urlp = "/event/off/test11";
String postRequestp = "POST " + urlp + " HTTP/1.1\r\n" +
    "Host: " + server + ":3000\r\n\r\n";
```

Ці фіди оголошують змінні типу фіду, значеннями яких є адреса сервера, на який буде відправлено запит, два url-адреси для надсилання запитів про стан системи відстеження та два вмісту запитів для надсилання інформації про системні збої або нормальна робота.

```
void setup() {
    initializeSerial();
    initializePins();
    initializeRadio();
    initializeLCD();
    password = EEPROM.read(0); // Read saved password from EEPROM
}
```

В функції `setup()` як і в попередньому підрозділі, відбувається ініціалізація всіх елементів пристрою та їх налаштувань які потрібні лише один раз.

```
void loop() {
  byte pipeNo;
  unsigned long currentMillis = millis();
  postmes();
  if (radio.available(&pipeNo)) {
    Serial.println("повідомлення");
    radio.read(&message, sizeof(message));
    if (message == 1 || message == 0) {
      firstTime = true;
      messageAlert = true;
      Serial.println("повідомлення OK");
      handleIntrusion(message, currentMillis);
    }
  }
  checkButtonInput();
}
```

У функції `loop()` відбувається увімкнення "труб" для прийому повідомлень з модулів `nRF24L01`, використання таймера `millis()`, виклику функції для відправки запиту на сервер `postmes()` та перевірки і читання повідомлення. Якщо значення повідомлення 1 або 0 то змінні `firstTime` та `messageAlert` встановлюються в значення `true` і викликається функція `handleIntrusion()`.

```
void postmes() { // відправка запиту на сервер
  unsigned long currentMillis = millis();
  if (firstTime) {
    previousPostMillis = currentMillis - postInterval;
```

```

    firstTime = false;
}
// Перевіряємо, чи пора відправити HTTP POST-запит
if (currentMillis - previousPostMillis >= postInterval) {
    previousPostMillis = currentMillis;
    // відправка HTTP POST-запиту
    digitalWrite(ledPin2, HIGH);
    esp8266.println("AT+CIPSTART=\"TCP\", \"" + server + "\",3000");
    delay(50);
}
if (messageAlert) {
    esp8266.println("AT+CIPSEND=" + String(postRequestp.length()));
    delay(50);
    esp8266.println(postRequestp);
    delay(50);
} else {
    esp8266.println("AT+CIPSEND=" + String(postRequest.length()));
    delay(50);
    esp8266.println(postRequest);
    delay(50);
}
esp8266.println("AT+CIPCLOSE");
delay(50);
digitalWrite(ledPin2, LOW);
}
}

```

Далі у функції `postmess()` оголошується таймер і перевіряється значення змінної під час першого надсилання запиту на сервер і через 30 секунд, якщо тривога не зникає.

Далі описується процес надсилання запиту, який складається з підключення до сервера, визначення довжини запиту, його відправлення та закриття каналу. Також для ідентифікації запиту жовтий світлодіод включається на початку і вимикається в кінці.

```
bool checkPassword() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("parol:");
    lcd.setCursor(0, 1);
    delay(250);
    int passwordCount = 0;
    bool inputComplete = false;
    bool confirmed = false;
    unsigned long lastButtonPressMillis = millis();
    while (!inputComplete) {
        blinkLedAndSoundBuzzer();
        if (digitalRead(inputButtonPin) == HIGH) {
            delay(20);
            lastButtonPressMillis = millis();
            passwordCount++;
            lcd.print("*");
            delay(500);
            Serial.print("Password count: ");
            Serial.println(passwordCount);
        }
    }
```

Тут відбувається визначення чи введено пароль, якщо кнопка натискається то це записується як введення пароля методом збільшення лічильника passwordCount++ та на екран виводиться "*" після кожного натискання при введенні пароля. Пароль у

цьому контексті - це кількість натиснень кнопки, яку користувач натискає.

```

if (digitalRead(confirmButtonPin) == HIGH) {
    delay(200);
    lastButtonPressMillis = millis(); // Update the last button press time
    Serial.println("Confirm button pressed");
    if (passwordCount == password) {
        confirmed = true;
        inputComplete = true;
        Serial.println("Password confirmed");
    } else {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Nevirnyi parol");
        unsigned long wrongPasswordMillis = millis();
        while (millis() - wrongPasswordMillis < 2000) {
            blinkLedAndSoundBuzzer();}
    }
}

```

Далі після натиснення кнопки для підтвердження введення якщо пароль правильний то відбувається відміна та тривога припиняється. Якщо пароль введено не правильно то на дисплей на 2 секунди виводиться напис "Nevirnyi parol" та надається можливість ввести пароль повторно.

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("parol:");
    lcd.setCursor(0, 1);
    passwordCount = 0;
    Serial.println("Incorrect password, resetting password count");
}

```

```

}
}
return confirmed;
}

```

Загалом у функції `checkPassword()` відбувається процес введення та перевірки пароля, і у випадку його коректності відбувається відміна тривоги. Також ця функція може повертати значення або `true` або `false`.

```

void changePassword() {
    digitalWrite(ledPin, LOW);
    digitalWrite(ledPin1, HIGH);
    digitalWrite(buzzerPin, HIGH);
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Novyi parol:");
    lcd.setCursor(0, 1);
    unsigned long resetMillis = millis();
    while (millis() - resetMillis < 1000) {
    }
    int newPasswordCount = 0;
    bool inputComplete = false;
    unsigned long lastButtonPressMillis = millis();
    while (!inputComplete) {
        if (digitalRead(inputButtonPin) == HIGH) {
            delay(20);
            newPasswordCount++;
            lcd.print("*");

```

```

    delay(500);
}

```

У цій функції відбувається процес зміни пароля. Одразу після деяких налаштувань на дисплей виводиться "Novyi parol:" встановлюються потрібні змінні newPasswordCount, inputComplete, lastButtonPressMillis та лічильник newPasswordCount++, щоб користувач міг ввести новий пароль.

```

if (digitalRead(confirmButtonPin) == HIGH) {
    delay(200);
    password = newPasswordCount;
    EEPROM.write(0, password); // збереження пароля
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Parol' zmineno");
    unsigned long resetMillis = millis();
}

```

Після того як користувач введе новий пароль, його значення переноситься зі змінної newPasswordCount в змінну password та потім зберігається в пам'яті та виводиться напис "Parol' zmineno".

```

while (millis() - resetMillis < 2000) {}
    inputComplete = true;
    lcd.clear();
    lcd.noBacklight();
}
}
}

void handleIntrusion(byte message, unsigned long currentMillis) {
    displayIntrusionWarning(message);
    halfHourMillis = currentMillis;
}

```

```

bool passwordCheck = false;
while (!passwordCheck && (currentMillis - halfHourMillis < halfHourInterval)) {
    currentMillis = millis();
    blinkLedAndSoundBuzzer();
    if (digitalRead(confirmButtonPin) == HIGH) {
        passwordCheck = checkPassword();
        if (!passwordCheck) {
            displayIntrusionWarning(message);
        }
    }
    if (currentMillis - halfHourMillis >= halfHourInterval) {
        displayIntrusionWarning(message);
    }
}
if (passwordCheck) {
    resetIntrusionState();
}
}

```

Функція `HandIntrusion()` є обробником умови вторгнення. Ця функція приймає два аргументи: повідомлення для відображення та `CurrentMillis`, що є поточним значенням у мілісекундах. Спочатку викликається функція `IntrusionWarning` (повідомлення), яка відображає на екрані повідомлення про вторгнення. Потім він продовжується, доки не буде введено правильний пароль або поки не пройде півгодини.

```

void displayIntrusionWarning(byte message) {
    lcd.clear();
    lcd.backlight();
}

```

```

lcd.setCursor(0, 0);
lcd.print("uvaga porushnyk");
lcd.setCursor(0, 1);
lcd.print((message == 1) ? "1 dveri" : "2 dveri");
}

```

У функції `displayIntrusionWarning(byte message)` відбувається відображення на дисплей інформації про порушення периметру. Зчитується значення повідомлення та відображається на яких з проходів порушення.

```

void blinkLedAndSoundBuzzer() {
    unsigned long currentMillis = millis();
    if (changePasswordMode) {
        return; }
    postmes();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        ledState = !ledState;
        digitalWrite(ledPin, ledState ? HIGH : LOW);
        buzzerState = !buzzerState;
        digitalWrite(buzzerPin, buzzerState ? LOW : HIGH);
    }
}

```

Наступна функція `blinkLedAndSoundBuzzer()` — це алгоритм, який блимає індикатором тривоги та чує звук. Додано перевірку, чи змінює користувач зараз пароль для належної роботи системи. Крім того, спочатку викликається функція `postmess()`, щоб надіслати запит на порушення периметру.

4 ТЕСТУВАННЯ СИСТЕМИ

Даний розділ в дипломній роботі містить детальний опис процесу перевірки працездатності та ефективності розробленої системи. У цьому розділі будуть представлені основні етапи тестування, які включають в себе перевірку кожного компонента системи та її функціональності в цілому.

На рисунку 4.1 зображено екран привітання при увімкненні головного блоку системи.

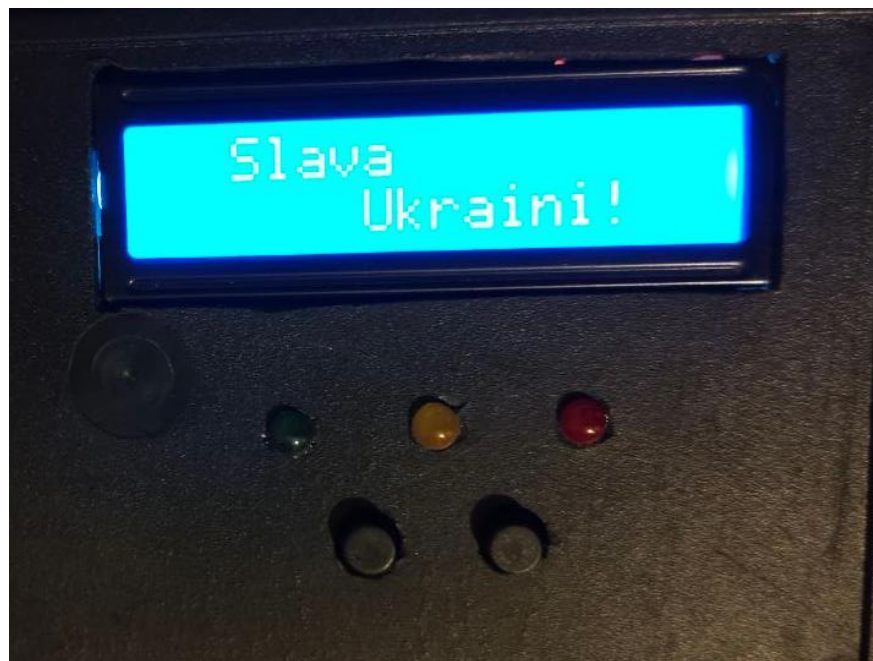


Рисунок 4.1 — Екран привітання

4.1 Тестування підключення до мережі Wi-Fi та сервера

На рисунку 4.2 показане повідомлення, що з'являється при процесі підключення до мережі Wi-Fi. Також загоряється жовтий світлодіод, що слугує індикатором мережевого підключення.

На рисунку 4.3 показано вивід повідомлення при вдалому підключенні до мережі Wi-Fi.

На наступному рисунку (Рис. 4.4) зображено повідомлення що виводиться при невдалому підключенні до мережі Wi-Fi чи серверу.



Рисунок 4.2 — Процес підключення до мережі



Рисунок 4.3 — Повідомлення про вдале підключення

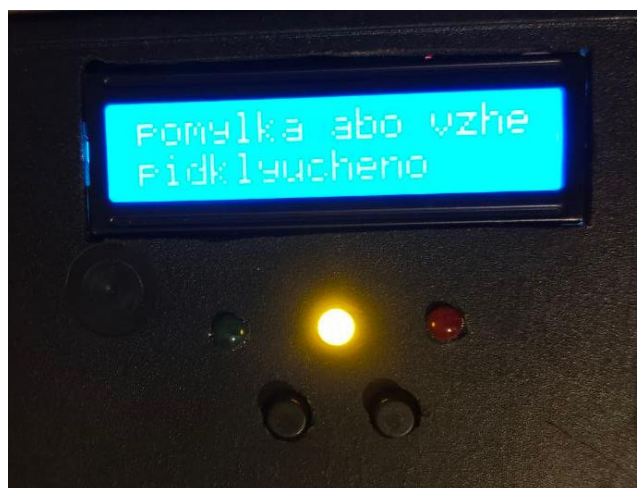


Рисунок 4.4 — Повідомлення про невдале підключення

4.2 Тестування реагування на порушення периметру

На рисунку 4.5 зображено повідомлення, що надходить на головний блок при виявленні порушення безпеки периметру об'єкту. При чому, якщо є декілька входів то в повідомленні буде вказано назву чи номер входу де відбулось порушення периметру.



Рисунок 4.5 — Повідомлення про порушення периметру

Далі показано повідомлення, що виводиться при введенні пароля для відключення сигналізації про порушника. При цьому, якщо пароль введено не правильно, сигналізація буде працювати і далі поки користувач не введе правильний пароль, або поки не пройде пів години. Наступну подію зображено на рисунку 4.6.



Рисунок 4.6— Повідомлення про відключення тривоги

Також можна зазначити, що сервер та бот в месенджері Telegram працює успішно та без помилок, що зображено на рисунку 4.7.

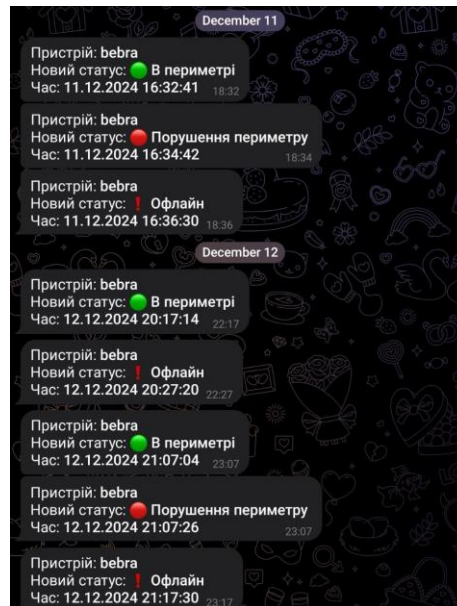


Рисунок 4.7 — Повідомлення в боті

В результаті проведеного тестування системи охорони периметру, яку було спроектовано та розроблено в ході виконання КМКР, можна зробити наступні висновки:

Система відповідає встановленим вимогам та успішно виконує основні завдання, зокрема виявлення порушень периметру в реальному часі, передача сигналу тривоги на сервер та в бот.

Система продемонструвала стабільну роботу в умовах реального середовища, включаючи вплив погодних умов та різні рівні освітленості.

Показники виявлення порушень периметру відповідають очікуванням. Відсоток хибних спрацьовувань був мінімальним і не перевищував допустимого рівня.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нової системи охорони периметру для запобігання проникнення сторонніх осіб на підприємство в невстановлений для цього час чи місце. Особливістю розробки є автоматизований контроль і моніторинг периметру, який забезпечує вчасне виявлення загроз за допомогою сенсорів і камер, під'єднаних до мікроконтролерів. Це сприяє ефективності охорони та зниженню людського фактора, що підвищує рівень безпеки підприємства.

Аналогом розробки може бути система охорони Optex AX-200TFR – 23292 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги					
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 нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження табл. 5.1

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
Ринкові перспективи					
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
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
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	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 більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження табл. 5.1

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 вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	3
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	3	3	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	4
Супровідна документація	3	3	3
Сума	41	40	41
Середньоарифметична сума балів	$(41+40+41) / 3 = 40,67$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є вище середнього, що досягається за рахунок автоматизованого контролю і моніторингу периметру, який забезпечує вчасне виявлення загроз за допомогою сенсорів і камер, під'єднаних до мікроконтролерів.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів за місяць, 21 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	45000	2142,86	32	68571,429

Програміст	42000	2000,00	32	64000,000
Всього				132571,43

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брали участь в розробці обладнання/програмного продукту

Додаткову заробітну плату прийнято розраховувати як 14 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 14 \% / 100 \% \quad (5.2)$$

$$З_д = (132571,43 \cdot 14 \% / 100 \%) = 18560,00 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників

Згідно діючого законодавства складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22 \% / 100\% \quad (5.3)$$

$$Н_з = (132571,43 + 18560,00) \cdot 22 \% / 100 \% = 33248,91 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12} \text{ [грн.]} \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 24000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,52 міс.

$$A_{\text{обл}} = \frac{24000}{2} \times \frac{1,52}{12} = 1523,81 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{нем.ак.}} = 11000$ грн.

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна периферія	24000	2	1,52	1523,810
Офісне обладнання (меблі)	28000	4	1,52	888,889
Приміщення	1850000	20	1,52	11746,032
Всього				14158,73

5.2.6 Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення

При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2024 році для Вінницької області за даними Енерга-Вінниця, $V = 10,42$ грн./кВт;

Π – встановлена потужність обладнання, кВт. $\Pi = 0,3$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,3 \cdot 8 \cdot 32 \cdot 10,42 = 720,2304 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі 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» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.6)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_e = 132571,43 \cdot 80\% / 100\% = 106057,1 \text{ (грн.)}$$

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» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 132571,43 * 150 \% / 100 \% = 198857 \text{ (грн.)}$$

5.2.9 Витрати на проведення науково-дослідної роботи

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 132571,43 + 18560,00 + 33248,91 + 14158,73 + 11000 + 720,23 + 106057,1 + \\ + 198857 = 515173,59 \text{ грн.}$$

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів

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

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$3B = 515173,59 / 0,5 = 1030347 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

— вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

— зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

— кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

— визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем

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

$$\Delta\P_i = (\pm\Delta\P_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.9)$$

де $\pm\Delta\P_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_0$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_0 = Ц_6 \pm \Delta Ц_0$;

$Ц_6$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2024 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 13300 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 3000 шт., протягом другого року – на 2000 шт., протягом третього року на 1500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\P_1 = (0 \cdot 500 + (13300 + 500) \cdot 3000) \cdot 0,8333 \cdot 0,23 \cdot (1 - 0,18) = 6270949,749 \text{ грн.}$$

$$\Delta\P_2 = (0 \cdot 500 + (13300 + 500) \cdot (3000 + 2000)) \cdot 0,8333 \cdot 0,23 \cdot (1 - 0,18) = 10844499,566 \text{ грн.}$$

$$\Delta\P_3 = (0 \cdot 500 + (13300 + 500) \cdot (3000 + 2000 + 1500)) \cdot 0,8333 \cdot 0,23 \cdot (1 - 0,18) = 14097849,436 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 31213298,75 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

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

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$ПП = (6270949,749/(1+0,1)^1) + (10844499,566/(1+0,1)^2) + (14097849,436/(1+0,1)^3) = 5700863,41 + 8962396,336 + 10591922,94 = 25255182,69 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * 3B, \quad (5.11)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку

приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{inv}=2...5$, але може бути і більшим;

$$PV = 2 * 1030347 = 2060694,36 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (5.12)$$

$$E_{abc} = 25255182,69 - 2060694,36 = 23194488,33 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.13)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 23194488,33/2060694,36) - 1} = 1,306$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,09 \dots 0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_v > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (5.15)$$

$$T_{ок} = 1 / 1,306 = 0,77 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,77 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 1030347 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,77 роки.

ВИСНОВКИ

У процесі виконання дипломної роботи було досліджено методи та технології виявлення порушень периметру та реагування на них. В результаті виконано всі заплановані завдання, спроектовано та реалізовано систему стеження за периметром на основі плат Arduino Uno (мікроконтролер Atmega328P) та Arduino Nano (мікроконтролер Atmega328).

Було розроблено концепцію побудови системи з урахуванням цілей та умов її використання. Порівняно різні аналоги подібних систем, їх характеристики, щоб досягти кращих результатів у створюваній системі. Також визначено вимоги до розроблюваного продукту.

У роботі було розглянуто та обрано компоненти системи, їх моделі та характеристики. Спроектовано і описано структурну схему системи стеження за периметром. Проаналізовано обрані плати та мікроконтролери, на основі яких побудована система. Розроблено функціональні схеми всіх блоків системи та їх електричні принципові схеми.

Були створені алгоритми роботи системи та її додаткових функцій. Оцінено та вибрано середовище програмування (Arduino IDE) та мова програмування (C++). Розроблено програмний код для блоку датчиків і описано його основні функції. Також створено програмний код для блоку управління системою стеження за периметром, з детальним описом основних функцій.

Розроблена система задовільняє задані вимоги та має переваги над аналогами, а саме: простота використання, модульність деяких елементів, компактність, ціна.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Баран В. С., Власюк Г. Г., Оникієнко Ю. О., Смоленська О. І. Основи мікропроцесорної техніки: лабораторний практикум КПІ ім. Ігоря Сікорського, 2019. 140 с.
2. Simon Monk, "Arduino Workshop: A Hands-On Introduction with 65 Projects". No Starch Press. San Francisco, CA . 2013. 392 с.
3. Massimo Banzi, Michael Shiloh, "Getting Started with Arduino: The Open Source Electronics Prototyping Platform". Maker Media, Inc. Sebastopol, CA. 2015. 262 с.
4. Paul McWhorter, "Arduino Programming in 24 Hours, Sams Teach Yourself". Sams Publishing. Indianapolis, IN. 2014. 432 с.
5. Charles Platt, "Make: Electronics: Learning Through Discovery". Maker Media, Inc. Sebastopol, CA. 2015. 352 с.
6. І. Е. Петруніна Доповнений довідковий посібник. 3-є видання. Під ред. «Довідник по пайці» Машинобудування, 2003 рік, 480 с.
7. Кіреєв П. С. Фізика напівпровідників. М. : Вища школа , 1975. 584 с
8. Крик В.В. Навчальний посібник «Мікропроцесорна техніка». Київський Політехнічний інститут. Мікроконтролери в електротехнічних системах. Київ «Політехніка». 2014. розд. 5, С. 24–25.
9. Я. В. Бобицький, Г. Л. Матвіїшин Лазерні технології: навч. посіб. Ч. 1 ; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». Львів: Вид-во Львів. політехніки, 2015. 316 с.
10. О. А. Бурий, С. Б. Убізський Моделювання та оптимізація твердотільних мікрочіпових лазерів: монографія ; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». Л. : Вид-во Львів. політехніки, 2013. 200 с.
11. Фредерік М. Бланкеншіп. "Security Engineering: A Guide to Building Dependable Distributed Systems" Addison-Wesley Professional, 2008, 848 с.
12. Система охорони периметру компанії Ajax [Електронний ресурс]. —Режим доступу: <https://worldvision.com.ua/laser-f2-mod-for->

ajax//%D0%BF%D0%BB%D0%B0%D0%BD%D0%B8%20%D0%BA%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D0%B8/19/19.htm.

13. Система охорони периметру компанії Optex [Електронний ресурс]. —Режим доступу: <https://worldvision.com.ua/ru/optex-ax-200tfr/>

14. Джозеф К. Дойелл. "Operational Security Management: A Complete Guide to Effective Security Programs" Auerbach Publications, 2011, 424 с.

15. ATmega328 [Електронний ресурс]. — Режим доступу: <https://www.microchip.com/en-us/search?searchQuery=ATmega328&category=ALL&fq=start%3D0%26rows%3D10>

16. ATmega328 Datasheet [Електронний ресурс]. — Режим доступу: <https://is.gd/zLkvLF>

17. Майкл Марголіс, «Arduino Cookbook». O'Reilly Media, 3-є видання, 2020. 796 с.

18. Джонатан Оксер, Г'ю Блемінгс, «Практичний Arduino: класні проекти для обладнання з відкритим кодом». Апрес, 2011. 456 с.

19. Ітан Торп, «Arduino: передові методи та стратегії використання Arduino». Independent Published, 2019. 160 с.

20. Джон Нассі, «Arduino для чайників». Для чайників, 2-є видання, 2018. 432 с.

21. Томас Скарборо, «Arduino Projects for Amateur Radio». АРПЛ, 2020. 352 с.

22. Саймон Монк, «Програмування Arduino: початок роботи зі скетчами». McGraw-Hill Education, 2nd Edition, 2016. 200 с.

23. Марк Геддес, «Посібник із проекту Arduino: 25 практичних проектів для початку». No Starch Press, 2016. 272 с

ДОДАТОК А

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

_____ д.т.н., проф. О. Д. Азаров

«___» _____ 20__ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Мікроконтролерна система охорони периметру підприємства»

08-54. КМКР.016.00.000 ПЗ

Науковий керівник

д.т.н., ст. проф.

_____ Азаров О.Д.

Виконав:

Магістрант 2 курсу

_____ Рогозянський В.А.

Вінниця 2024

1 Підстава виконання магістерської кваліфікаційної роботи

1.1 Наказ про затвердження теми магістерської кваліфікаційної роботи від 17.09.2024 р. №310.

2 Мета і призначення КМКР

2.1 Метою є розробка мікроконтролерної системи охорони периметру підприємства на базі платформи Arduino, яка забезпечує автоматизований моніторинг, виявлення порушень і оповіщення персоналу для підвищення рівня безпеки.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3 Вихідні дані для виконання КМКР

Вихідні дані: мова програмування C++, бібліотеки для взаємодії з мережевими модулями такими як ESP8266, середовище розробки Arduino IDE.

4 Вимоги до виконання КМКР

КМКР повинна задовільняти такі вимоги:

- провести аналіз існуючих аналогів;
- спроектувати систему та підібрати компоненти;
- забезпечити взаємодію апаратної частини з сервером;
- провести розробку та тестування системи.

5 Етапи КМКР та очікувані результати

Етапи роботи та очікувані результати наведені в таблиці А.1.

6 Матеріали, що подаються до захисту КМКР

До захисту КМКР подаються: пояснювальна записка КМКР, ілюстративні та графічні матеріали, протокол попереднього захисту КМКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до КМКР

українською та іноземною мовами, довідка про відповідність оформлення КМКР діючим вимогам.

Таблиця А.1 — Етапи КМКР

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Прим.
1	Постановка мети та задач роботи	21.10.24	
2	Огляд і аналіз аналогів систем охорони периметрів та об'єктів	25.10-30.10.24	
3	Проектування апаратного забезпечення системи	31.10-08.11.24	
4	Складання структурної схеми системи, що розробляється	09.11-15.11.24	
5	Визначення моделей компонентів системи	16.11-.20.11.24	
6	Проектування програмного забезпечення системи	21.11-25.11.24	
7	Програмна реалізація рішень	26.11-31.11.24	
8	Тестування системи	01.12-03.12.24	
9	Розрахунок економічної частини роботи	04.12-07.12.24	
10	Оформлення пояснювальної записки та ілюстративного матеріалу	08.12.24	
11	Аналіз виконання роботи, висновки, додатки		
12	Перевірка якості виконання магістерської роботи та усунення недоліків		

7 Порядок контролю виконання та захисту КМКР

Виконання етапів графічної та розрахункової документації КМКР контролюється науковим керівником згідно зі встановленими термінами. Захист КМКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення КМКР

8.1 При оформленні КМКР використовуються:

— ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022.

8.2 Порядок виконання КМКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21

ДОДАТОК Б

Лістинг коду головного блоку системи

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Wire.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>

RF24 radio(9, 10);

byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"};
const int ledPin = 7;
const int ledPin1 = 3;
const int ledPin2 = 14;
const int inputButtonPin = 2;
const int confirmButtonPin = 4;
const int buzzerPin = 8;
SoftwareSerial esp8266(6, 5); // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);

unsigned long lastIntrusionMillis = 0;
unsigned long previousMillis = 0;
unsigned long interval = 500;
unsigned long displayMillis = 0;
unsigned long displayInterval = 3000;
unsigned long observationMillis = 0;
unsigned long observationInterval = 5000;
unsigned long halfHourMillis = 0;
unsigned long halfHourInterval = 1800000;
unsigned long checkMessageMillis = 0;
unsigned long checkMessageInterval = 100;
```

```

unsigned long buttonPressMillis = 0;
unsigned long longPressInterval = 5000; // 5 seconds
// Час останнього POST-запиту
unsigned long previousPostMillis = 0;
// Інтервал між POST-запитами (30 секунд у мілісекундах)
const long postInterval = 30000;

byte message;
bool changePasswordMode = false;
bool buttonPressed = false;

bool ledState = false;
bool buzzerState = false;
bool intrusionActive = false;
int password = 5; // Default password
bool firstTime = true;
bool messageAlert = false;
String server = "13.48.132.106";
String url = "/event/on/test11";
String postRequest = "POST " + url + " HTTP/1.1\r\n" +
    "Host: " + server + ":3000\r\n\r\n";

String urlp = "/event/off/test11";
String postRequestp = "POST " + urlp + " HTTP/1.1\r\n" +
    "Host: " + server + ":3000\r\n\r\n";

void setup() {
    initializeSerial();
    initializePins();
    initializeRadio();
    initializeLCD();
    password = EEPROM.read(0); // Read saved password from EEPROM
}

```

```

void loop() {
    radio.setAutoAck(1);
    radio.setRetries(5, 15);
    radio.enableAckPayload();
    radio.setPayloadSize(128);
    radio.powerUp();
    byte pipeNo;
    unsigned long currentMillis = millis();
    postmes();
    if (radio.available(&pipeNo)) {
        Serial.println("повідомлення");
        radio.read(&message, sizeof(message));
        if (message == 5 || message == 0) {
            firstTime = true;
            messageAlert = true;
            Serial.println("повідомлення ОК");
            handleIntrusion(message, currentMillis);
        }
    }
    checkButtonInput();
}

void checkButtonInput() {
    unsigned long buttonPressMillis = millis();
    if (!radio.available() && digitalRead(inputButtonPin) == HIGH) {
        handleInfoButtonPress(buttonPressMillis);
    }
    if (!radio.available() && digitalRead(confirmButtonPin) == HIGH) {
        handleChangePasswordButtonPress(buttonPressMillis);
    }
}

void postmes() { // відправка запиту на сервер
    unsigned long currentMillis = millis();

```

```

    if (firstTime) {
        previousPostMillis = currentMillis - postInterval;
        firstTime = false;
    }
    // Перевіряємо, чи пора відправити HTTP POST-запит
    if (currentMillis - previousPostMillis >= postInterval) {
        previousPostMillis = currentMillis;
        digitalWrite(ledPin2, HIGH);
        esp8266.println("AT+CIPSTART=\"TCP\",\"" + server + "\",3000");
        delay(50);
        if (esp8266.find("OK")) {
            Serial.println("Connected to server");
        } else {
            Serial.println("Error");
        }
        if (messageAlert) {
            esp8266.println("AT+CIPSEND=" + String(postRequestp.length()));
            delay(50);
            if (esp8266.find(">")) {
                Serial.println("CIPSEND OK");
            } else {
                Serial.println("CIPSEND Error");
            }
            esp8266.println(postRequestp);
            delay(50);
            if (esp8266.find("SEND OK")) {
                Serial.println("SEND OK");
            } else {
                Serial.println("SEND Error");
            }
        } else {
            esp8266.println("AT+CIPSEND=" + String(postRequest.length()));
            delay(50);

```

```

    if (esp8266.find(">")) {
        Serial.println("CIPSEND OK");
    } else {
        Serial.println("CIPSEND Error");
    }
    esp8266.println(postRequest);
    delay(50);
    if (esp8266.find("SEND OK")) {
        Serial.println("SEND OK");
    } else {
        Serial.println("SEND Error");
    }
}
esp8266.println("AT+CIPCLOSE");
delay(50);
if (esp8266.find("OK")) {
    Serial.println("CLOSE");
} else {
    Serial.println("CLOSE Error");
}
digitalWrite(ledPin2, LOW);
radio.powerDown();
}
}

void handleInfoButtonPress(unsigned long buttonPressMillis) {
    while (digitalRead(inputButtonPin) == HIGH && millis() - buttonPressMillis < 3000) { }
    if (millis() - buttonPressMillis >= 3000) {
        displayElapsedTimeSinceLastIntrusion();
    }
}

void handleChangePasswordButtonPress(unsigned long buttonPressMillis) {

```

```

while (digitalRead(confirmButtonPin) == HIGH && millis() - buttonPressMillis < 3000) { }
if (millis() - buttonPressMillis >= 3000) {
    attemptPasswordChange();
}
}

```

```

void displayElapsedTimeSinceLastIntrusion() {
    // Display elapsed time since last intrusion
    unsigned long elapsedTime = millis() - lastIntrusionMillis;
    int hours = int(elapsedTime / 3600000);
    elapsedTime = elapsedTime % 3600000;
    int minutes = int(elapsedTime / 60000);
    elapsedTime = elapsedTime % 60000;
    int seconds = int(elapsedTime / 1000);
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Ostannya trivoga:");
    lcd.setCursor(0, 1);
    lcd.print(hours);
    lcd.print("h ");
    lcd.print(minutes);
    lcd.print("m ");
    lcd.print(seconds);
    lcd.print("s ");
    delay(5000);
    lcd.clear();
    lcd.noBacklight();
}

```

```

void attemptPasswordChange() {
    lcd.clear();
    changePasswordMode = true;
}

```

```

    lcd.backlight();
    lcd.print("Zmina parolya");
    delay(1000);
    if(checkPassword()) {
        changePassword();
    }
    changePasswordMode = false;
}

void initializeSerial() {
    Serial.begin(115200);
    esp8266.begin(115200);
}

void initializePins() {
    pinMode(ledPin, OUTPUT);
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(inputButtonPin, INPUT_PULLUP);
    pinMode(confirmButtonPin, INPUT_PULLUP);
    pinMode(buzzerPin, OUTPUT);
    digitalWrite(buzzerPin, HIGH);
}

void initializeRadio() {
    radio.begin();
    radio.setAutoAck(1);
    radio.setRetries(5, 15);
    radio.enableAckPayload();
    radio.setPayloadSize(128);
    radio.openReadingPipe(1, address[0]);
    radio.setChannel(0x6e);
    radio.setPALevel(RF24_PA_MAX);

```

```

    radio.setDataRate(RF24_250KBPS);
    radio.powerUp();
    radio.startListening();
}

void initializeLCD() {
    Wire.begin();
    lcd.init();
    lcd.backlight();
    lcd.setCursor(2, 0);
    lcd.print("Slava");
    lcd.setCursor(6, 1);
    lcd.print("Ukraini!");
    esp8266.println("AT+RST");
    delay(1000);
    digitalWrite(ledPin2, HIGH);
    esp8266.println("AT+CWMODE=1");
    delay(2000);
    if (esp8266.find("OK")) {
        Serial.println("CWMODE OK");
    } else {
        Serial.println("CWMODE Error");
    }
}

// Підключення до Wi-Fi
esp8266.println("AT+CWLAP=\"ASUSrog\", \" \");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("pidklyuchennya");
lcd.setCursor(0, 1);
lcd.print("do merezhi");
delay(8000);
if (esp8266.find("OK")) {
    Serial.println("Connected to WiFi");
}

```

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("pidklyucheno");
} else {
    Serial.println("Error connecting to WiFi");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("pomyłka abo vzhe");
    lcd.setCursor(0, 1);
    lcd.print("pidklyucheno");
}
// Налаштування ESP8266 на одноразову передачу даних
esp8266.println("AT+CIPMODE=0");
delay(1000);
if (esp8266.find("OK")) {
    Serial.println("CIPMODE OK");
} else {
    Serial.println("CIPMODE Error");
}
digitalWrite(ledPin2, LOW);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("vedetsya");
lcd.setCursor(0, 1);
lcd.print("sposterezhennya");
digitalWrite(ledPin1, HIGH);
delay(2000);
lcd.noBacklight();
}

bool checkPassword() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("parol:");

```

```

lcd.setCursor(0, 1);
delay(250);
int passwordCount = 0;
bool inputComplete = false;
bool confirmed = false;
unsigned long lastButtonPressMillis = millis();
while (!inputComplete) {
    blinkLedAndSoundBuzzer();
    if (digitalRead(inputButtonPin) == HIGH) {
        delay(20);
        lastButtonPressMillis = millis();
        passwordCount++;
        lcd.print("*");
        delay(500);
        Serial.print("Password count: ");
        Serial.println(passwordCount);
    }
    if (digitalRead(confirmButtonPin) == HIGH) {
        delay(200);
        lastButtonPressMillis = millis();
        Serial.println("Confirm button pressed");
        if (passwordCount == password) {
            confirmed = true;
            inputComplete = true;
            Serial.println("Password confirmed");
        } else {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Nevirnyi parol");
            unsigned long wrongPasswordMillis = millis();
            while (millis() - wrongPasswordMillis < 2000) {
                blinkLedAndSoundBuzzer();
            }
        }
    }
}

```

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("parol:");
    lcd.setCursor(0, 1);
    passwordCount = 0;
    Serial.println("Incorrect password, resetting password count");
}
}

if ( millis() - lastButtonPressMillis > 5000) {
    lcd.noBacklight();
    displayIntrusionWarning(message);
    buttonPressed = false;
    passwordCount = 0;
    return false;
}
}

return confirmed;
}

void changePassword() {
    digitalWrite(ledPin, LOW);
    digitalWrite(ledPin1, HIGH);
    digitalWrite(buzzerPin, HIGH);
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Novyi parol:");
    lcd.setCursor(0, 1);
    unsigned long resetMillis = millis();
    while (millis() - resetMillis < 1000) {
    }
    int newPasswordCount = 0;
    bool inputComplete = false;
    unsigned long lastButtonPressMillis = millis();

```

```

while (!inputComplete) {
  if (millis() - lastButtonPressMillis > 5000) {
    lcd.clear();
    lcd.noBacklight();
    return;
  }
  if (digitalRead(inputButtonPin) == HIGH) {
    delay(20);
    lastButtonPressMillis = millis();
    newPasswordCount++;
    lcd.print("*");
    delay(500);
  }
  if (digitalRead(confirmButtonPin) == HIGH) {
    delay(200);
    lastButtonPressMillis = millis();
    password = newPasswordCount;
    EEPROM.write(0, password);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Parol' zmineno");
    unsigned long resetMillis = millis();
    while (millis() - resetMillis < 2000) {
    }
    inputComplete = true;
    lcd.clear();
    lcd.noBacklight();
  }
}

void handleIntrusion(byte message, unsigned long currentMillis) {
  displayIntrusionWarning(message);
  halfHourMillis = currentMillis;
}

```

```

lastIntrusionMillis = currentMillis;
bool passwordCheck = false;
while (!passwordCheck && (currentMillis - halfHourMillis < halfHourInterval)) {
    currentMillis = millis();
    blinkLedAndSoundBuzzer();
    // Check if the confirm button is pressed
    if (digitalRead(confirmButtonPin) == HIGH) {
        passwordCheck = checkPassword();
        if (!passwordCheck) {
            displayIntrusionWarning(message);
        }
    }
    if (currentMillis - halfHourMillis >= halfHourInterval) {
        displayIntrusionWarning(message); // Re-display the intrusion warning
    }
}
if (passwordCheck) {
    resetIntrusionState();
}
}

void displayIntrusionWarning(byte message) {
    Serial.print("Порушник на ");
    Serial.print((message == 5) ? "1" : "2");
    Serial.println(" дверях");
    lcd.clear();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("uvaga porushnyk");
    lcd.setCursor(0, 1);
    lcd.print((message == 5) ? "1 dveri" : "2 dveri");
    digitalWrite(ledPin1, LOW);
}

void blinkLedAndSoundBuzzer() {

```

```

unsigned long currentMillis = millis();
postmes();
if (changePasswordMode) {
    return;
}
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    ledState = !ledState;
    digitalWrite(ledPin, ledState ? HIGH : LOW);
    buzzerState = !buzzerState;
    digitalWrite(buzzerPin, buzzerState ? LOW : HIGH);
}
}

void resetIntrusionState() {
    digitalWrite(ledPin, LOW);
    digitalWrite(ledPin1, HIGH);
    digitalWrite(buzzerPin, HIGH);
    buttonPressed = false;
    messageAlert = false;
    firstTime = true;
    postmes();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("vidmina");
    unsigned long resetMillis = millis();
    while (millis() - resetMillis < 3000) {
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("vedetsya");
    lcd.setCursor(0, 1);
    lcd.print("sposterezhennya");
    resetMillis = millis();
}

```

```
while (millis() - resetMillis < 5000) {  
  }  
  lcd.noBacklight();  
}
```

ДОДАТОК В

Лістинг коду блоку датчика

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"

const int sensorPin = A5; // пін для датчика світла
int sensorValue = 0; // зберігає останнє зчитане значення датчика світла
bool lastState = false; // зберігає стан світла з попереднього зчитування
RF24 radio(9, 10);

byte address[] = "6Node"; // адрес приймача
byte dataToSend = 5; // дані, які потрібно відправити

void setup() {
    Serial.begin(9600);
    radio.begin();
    radio.setAutoAck(1);
    radio.setRetries(0, 15);
    radio.enableAckPayload();
    radio.setPayloadSize(32);
    radio.openWritingPipe(address);
    radio.setChannel(0x6e);
    radio.setPALevel (RF24_PA_MAX);
    radio.setDataRate (RF24_250KBPS);
    radio.powerUp();
    radio.stopListening();
}

void loop() {
```

```
radio.powerUp();  
sensorValue = analogRead(sensorPin); // зчитування значення з датчика світла  
bool currentState = (sensorValue < 300); // визначення поточного стану світла  
if (currentState != lastState) { // перевірка на зміну стану світла  
    lastState = currentState;  
    if (!currentState) { // якщо світло вимкнулось  
        Serial.println("Порушник на 1 дверях");  
        radio.write(&dataToSend, sizeof(dataToSend)); // відправка даних на інший модуль  
    }  
}  
delay(100); // затримка для стабільної роботи з датчиком  
}
```

ДОДАТОК Г
Структурна схема системи

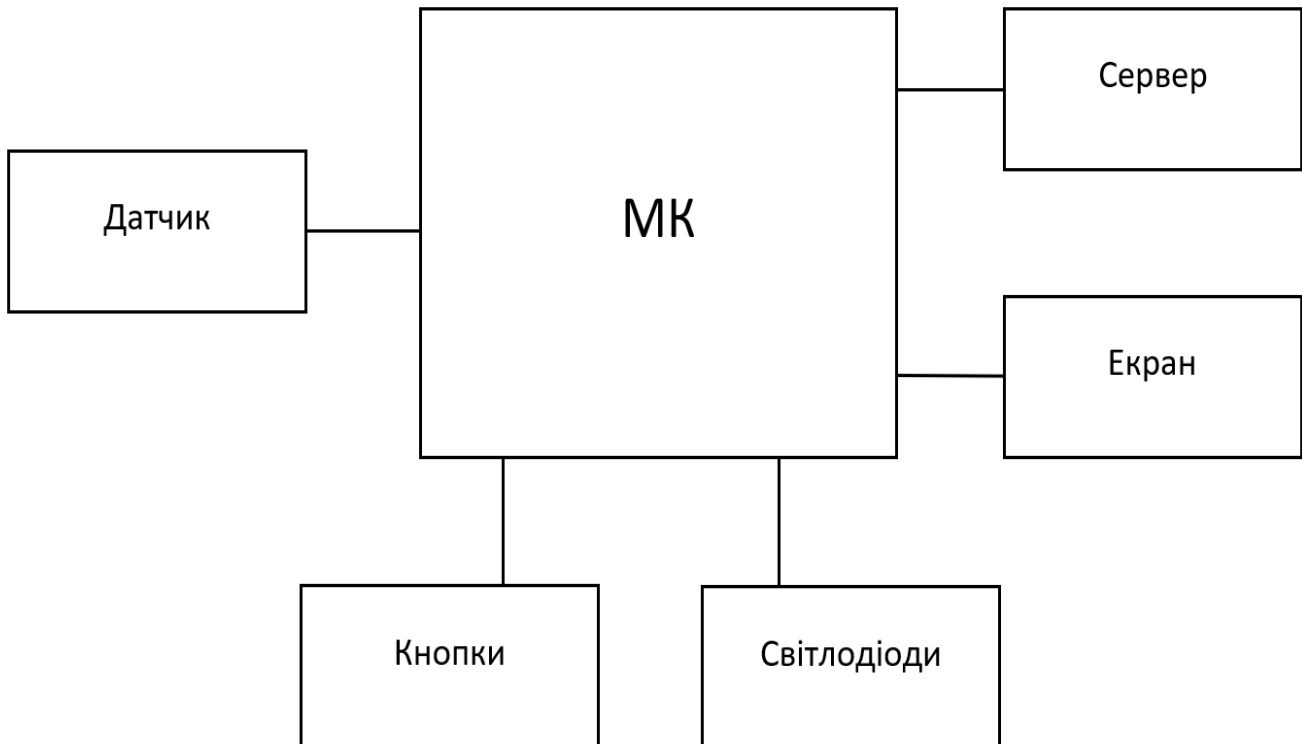


Рисунок Г.1 — Структурна схема системи

ДОДАТОК Д
Функціональна схема блоку управління

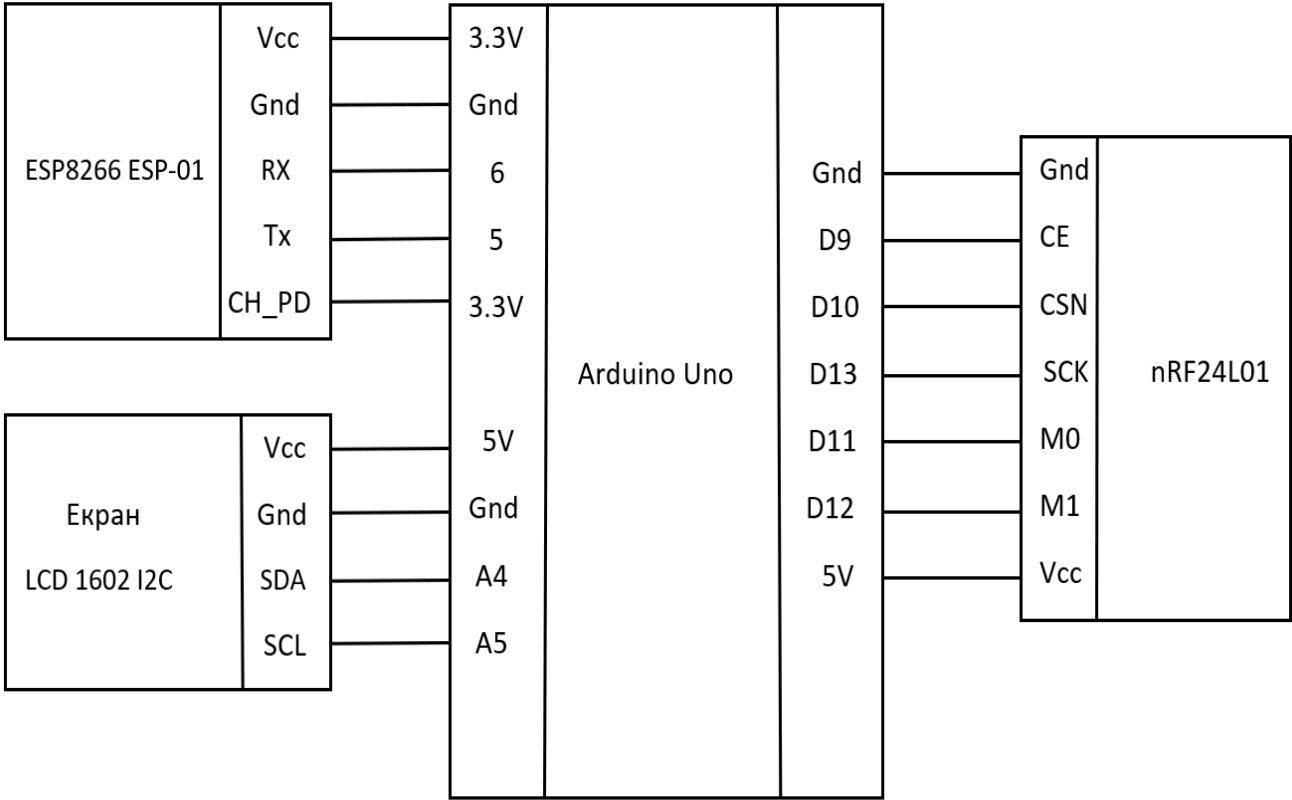


Рисунок Д.1 — Функціональна схема блоку управління системи

ДОДАТОК Е

Функціональна схема датчика

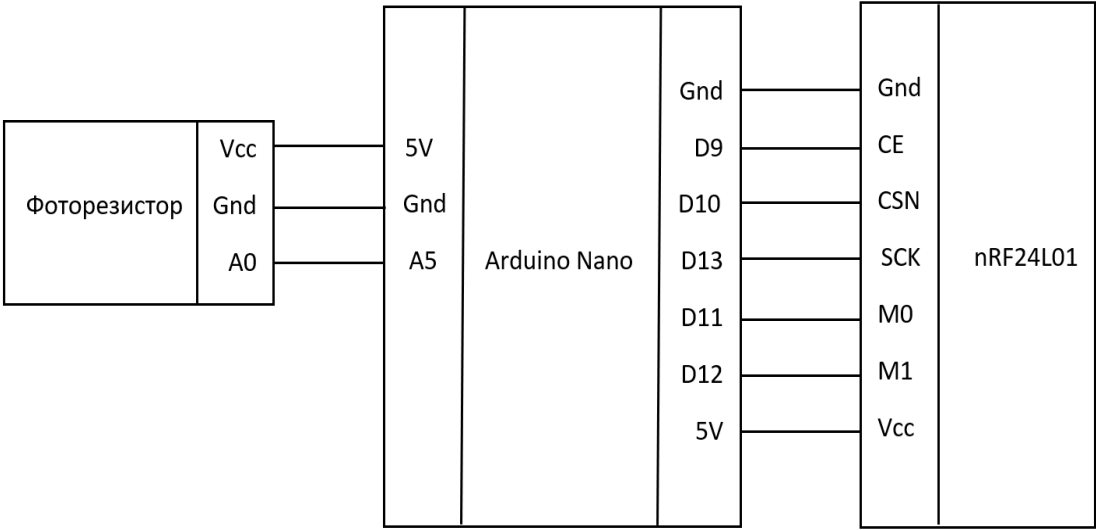


Рисунок Е.1 — Функціональна схема датчика системи

ДОДАТОК Ж

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Мікроконтролерна система охорони периметру підприємства

Тип роботи: Магістерська кваліфікаційна робота
(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (вказати))

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет (інститут), навчальна група)

Керівник Азаров О.Д. проф. каф. ОТ
(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	92%
Загальна схожість	8%

Аналіз звіту подібності (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор Рогозянський В.А.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

Особа, відповідальна за перевірку Захарченко С.М.
(підпис) (прізвище, ініціали)

Експерт _____
(за потреби) (підпис) (прізвище, ініціали, посада)