

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

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Мікропроцесорна система безпеки підприємства на платформі Arduino»

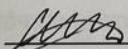
08-54.БКР.014.00.000 ПЗ

Виконав: студент 4 курсу, групи 1КІ-21Б
спеціальності 123 — «Комп'ютерна інженерія»
освітня програма — «Комп'ютерна інженерія»



Проценко О. Р.

Керівник phd., ст.викл. каф. ОТ

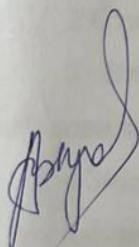


Обертюх М. Р.

Рецензент к.т.н., доцент каф.МБІС



Грицак А.В.



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

"__" _____ 2025 р.

Вінниця ВНТУ – 2025 рік

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

Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень бакалавр

Галузь знань – 12 – Інформаційні технології

Спеціальність – 123 – Комп'ютерна інженерія

Освітньо-професійна програма – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ проф., д.т.н.

Азаров О. Д.

"___" "___" 2025 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Проценку Олександру Руслановичу

1 Тема роботи: Мікропроцесорна система безпеки підприємства на платформі Arduino, керівник Обертюх Максим Романович, phd., ст.викл. каф. ОТ, затверджено наказом вищого навчального закладу «97» від 20 березня 2025 р.

2 Строк подання студентом роботи 13.05.2025р.

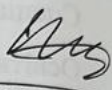
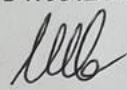
3 Вихідні дані до роботи: комплектуючі та технічне оснащення для побудови мікропроцесорної системи безпеки підприємства на платформі Arduino.

4 Зміст розрахунково-пояснювальної записки: вступ, аналіз концепції IoT та її застосування у системах безпеки, побудова архітектури системи на основі Arduino, реалізація прототипу та програмна інтеграція, висновки.

5 Перелік графічного матеріалу: схема підключення апаратних компонентів, макет системи.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	phd., ст.викл. каф. ОТ Обертюх М.Р	21.03.2025р. 	13.06.2025р. 
Нормоко нтроль	ас.каф.ОТ Швець С.І.	14.05.2025р. 	30.05.2025р.

7 Дата видачі завдання 21.03.2025 р.

8 Календарний план приведений в таблиці 2.

Таблиця 2— Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання	Підпис
1	Постановка задачі роботи	21.03.25	
2	Аналіз концепції IoT	21.03.25—29.03.25	
3	Застосування концепції IoT у системах безпеки	21.03.25—29.03.25	
4	Комплектуючі та технічне оснащення	30.03.25—11.04.25	
5	Програмування Arduino	12.04.25—26.04.25	
6	Тестування прототипу	27.04.25—07.05.25	
7	Оформлення пояснювальної записки та ілюстративного матеріалу	08.05.25	
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	14.05.25	

Студент

Керівник роботи

Олександр ПРОЦЕНКО

phd., ст.викл.

Максим

ОБЕРТЮХ

АНОТАЦІЯ

УДК 004.27

Проценко О. Р. Мікропроцесорна система безпеки підприємства на платформі Arduino. Бакалаврська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма Комп'ютерна Інженерія. Вінниця: ВНТУ, 2025. 72 с.

На укр. мові. Бібліогр.: 20 назв; рис.: 30; табл. 3.

Бакалаврська кваліфікаційна робота присвячена проектуванню та реалізації мікропроцесорної охоронної системи для підприємства з використанням технологій Інтернету речей. У роботі досліджено підходи до створення інтегрованих рішень на базі платформи Arduino, що дозволяють забезпечити виявлення порушень безпеки та своєчасне оповіщення відповідальних осіб у режимі реального часу.

В процесі дослідження проаналізовано методи підключення сенсорів та актуаторів до мікроконтролерів, принципи їх взаємодії, а також особливості побудови архітектури системи з урахуванням хмарних сервісів. Розглянуто протоколи зв'язку, які дозволяють передавати дані на віддалений сервер для подальшої обробки, зокрема реалізовано функціонал моніторингу та надсилання сповіщень через хмарну платформу.

Розроблена система була протестована в умовах, наближених до реальних, що дозволило оцінити її надійність, точність спрацювання та потенціал для масштабування. В результаті впровадження забезпечено підвищення рівня безпеки об'єкта, покращення якості моніторингу та зниження ризику небажаних інцидентів.

Ключові слова: Arduino, IoT, система безпеки, хмарна платформа, сенсори, моніторинг, реальний час.

ABSTRACT

Protsenko O. R. Microprocessor-Based Enterprise Security System on the Arduino Platform. Bachelor's Thesis in Specialty 123 — Computer Engineering, Educational Program Computer Engineering. Vinnytsia: VNTU, 2025. 72 p.

In Ukrainian. Bibliography: 20 sources; figures: 30; tables: 3.

The Bachelor's thesis is dedicated to the design and implementation of a microprocessor-based security system for an enterprise using Internet of Things technologies. The study explores approaches to creating integrated solutions based on the Arduino platform, enabling the detection of security breaches and timely notification of responsible personnel in real time.

The research analyzes methods for connecting sensors and actuators to microcontrollers, their interaction principles, and the specifics of system architecture design, considering cloud services. Communication protocols for transmitting data to a remote server for further processing were examined, with the implementation of monitoring and notification functionalities through a cloud platform.

The developed system was tested under conditions close to real-world scenarios, allowing an assessment of its reliability, accuracy, and scalability potential. The implementation resulted in enhanced object security, improved monitoring quality, and reduced risk of undesirable incidents.

Keywords: Arduino, IoT, security system, cloud platform, sensors, monitoring, real time.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ КОНЦЕПЦІЇ ІОТ ТА ЇЇ ЗАСТОСУВАННЯ У СИСТЕМАХ	
БЕЗПЕКИ.....	5
1.1 Складові систем технічної безпеки на підприємствах	5
1.2 Технологічні можливості Arduino в контексті IoT	10
1.3 Висновки щодо доцільності використання Arduino у проєктах безпеки	15
2 ПОБУДОВА АРХІТЕКТУРИ СИСТЕМИ НА ОСНОВІ ARDUINO	17
2.1 Сенсори руху та відкриття: принцип дії і підключення	17
2.2 Відеоспостереження як складова безпеки	23
2.3 Організація контролю доступу	26
2.4 Узагальнення побудови архітектури охоронної системи	30
3 РЕАЛІЗАЦІЯ ПРОТОТИПУ ТА ПРОГРАМНА ІНТЕГРАЦІЯ.....	32
3.1 Комплектуючі та технічне оснащення.....	32
3.2 Розробка програмної частини	42
3.3 Збірка і запуск тестового зразка системи	50
3.4 Тестування прототипу	53
3.5 Порівняння з комерційними системами безпеки.....	56
3.6 Підсумкова оцінка ефективності реалізованої системи.....	56
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А Технічне завдання	61
ДОДАТОК Б Протокол перевірки кваліфікаційної роботи	65
ДОДАТОК В Графічна частина	66
ДОДАТОК Г Ілюстративна частина.....	68
ДОДАТОК Д Лістинг програмного забезпечення	70

ВСТУП

Актуальність. На сьогодні технології Інтернету речей (IoT) відіграють важливу роль у трансформації багатьох сфер діяльності, включно з промисловістю, побутом і, зокрема, безпекою. Сучасні виклики, пов'язані зі зростанням загроз для бізнесу, потребують нових підходів до захисту об'єктів. Одним із ефективних рішень є впровадження IoT-платформ для створення інтелектуальних систем безпеки. Arduino, як доступна та гнучка платформа для мікроконтролерів, відкриває широкі можливості для розробки масштабованих, економічно вигідних рішень, які дозволяють виявляти загрози в реальному часі та оперативно реагувати на них.

Метою роботи є теоретичне обґрунтування та практична реалізація концепції створення системи безпеки підприємства на основі технологій IoT з використанням Arduino. Для досягнення поставленої мети необхідно виконати наступні завдання:

Завдання дослідження:

- провести аналіз сучасних IoT-технологій, що застосовуються у сфері безпеки;
- дослідити технічні характеристики та функціональні можливості мікроконтролерів Arduino для побудови охоронних систем;
- оцінити переваги та обмеження впровадження IoT-рішень на основі Arduino у системах безпеки підприємств;
- проаналізувати ефективність використання сенсорів, модулів та інших компонентів Arduino у комплексній системі контролю доступу та моніторингу.

Об'єкт дослідження функціонування системи безпеки підприємства на основі технологій IoT з використанням Arduino.

Предмет дослідження процеси інтеграції IoT-рішень для забезпечення безпеки підприємства на платформі Arduino.

Методи дослідження: системний аналіз, моделювання, порівняльний аналіз, а також емпіричні підходи до перевірки працездатності системи.

Практична цінність результатів.

Результати дослідження можуть бути використані для модернізації існуючих охоронних систем, розробки нових ефективних рішень для контролю доступу та моніторингу. Впровадження Arduino та IoT забезпечує зменшення витрат на охоронну інфраструктуру, підвищення рівня безпеки та можливість оперативної адаптації системи до потреб конкретного підприємства.

Апробація результатів бакалаврської кваліфікаційної роботи була проведена на конференції “Молодь в науці: дослідження, проблеми, перспективи” (МН-2025) [1]:

Проценко О.Р. Мікропроцесорна система безпеки підприємства на платформі Arduino // Матеріали всеукраїнської науково-практичної Інтернет-конференції студентів та молодих науковців — Режим доступу : <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/25451>

1 АНАЛІЗ КОНЦЕПЦІЇ ІОТ ТА ЇЇ ЗАСТОСУВАННЯ У СИСТЕМАХ БЕЗПЕКИ

1.1 Складові систем технічної безпеки на підприємствах

Інтернет речей (IoT) є однією з провідних інновацій у сучасних технологіях. Це концепція, яка передбачає створення мережі взаємопов'язаних фізичних пристроїв, здатних обмінюватися даними та взаємодіяти між собою через Інтернет. IoT відкриває нові можливості для багатьох галузей, таких як промисловість, транспорт, медицина та побутові технології [2].

Основною метою IoT є налагодження зв'язку між пристроями, які раніше працювали ізольовано. Така інтеграція дозволяє створювати автоматизовані системи, здатні спростити повсякденне життя та підвищити ефективність виробництва. Наприклад, в аграрному секторі IoT може бути використаний для моніторингу врожаю або автоматизації поливу, що дозволяє знижувати витрати та збільшувати врожайність.

Для реалізації цієї концепції використовуються різноманітні технології та пристрої. Сенсори та датчики здійснюють збір даних з навколишнього середовища, а мережеві технології забезпечують їх передачу через Інтернет. Управління і аналіз цих даних здійснюється за допомогою спеціального програмного забезпечення, яке дає змогу реалізовувати різні IoT-додатки.

IoT активно застосовується в різних галузях. У медицині він допомагає здійснювати віддалений моніторинг пацієнтів та підтримувати життєво важливі функції. У промисловості IoT автоматизує виробничі процеси і покращує системи контролю якості. У транспорті IoT дозволяє відстежувати рух транспортних засобів та вантажів. У побуті такі системи полегшують управління домом і підвищують безпеку [3].

Інтернет речей відкриває безліч можливостей для інтеграції нових технологій у повсякденне життя. Він є однією з основних технологічних тенденцій XXI століття, і в майбутньому очікується, що його вплив на різні сфери життєдіяльності лише зростатиме.

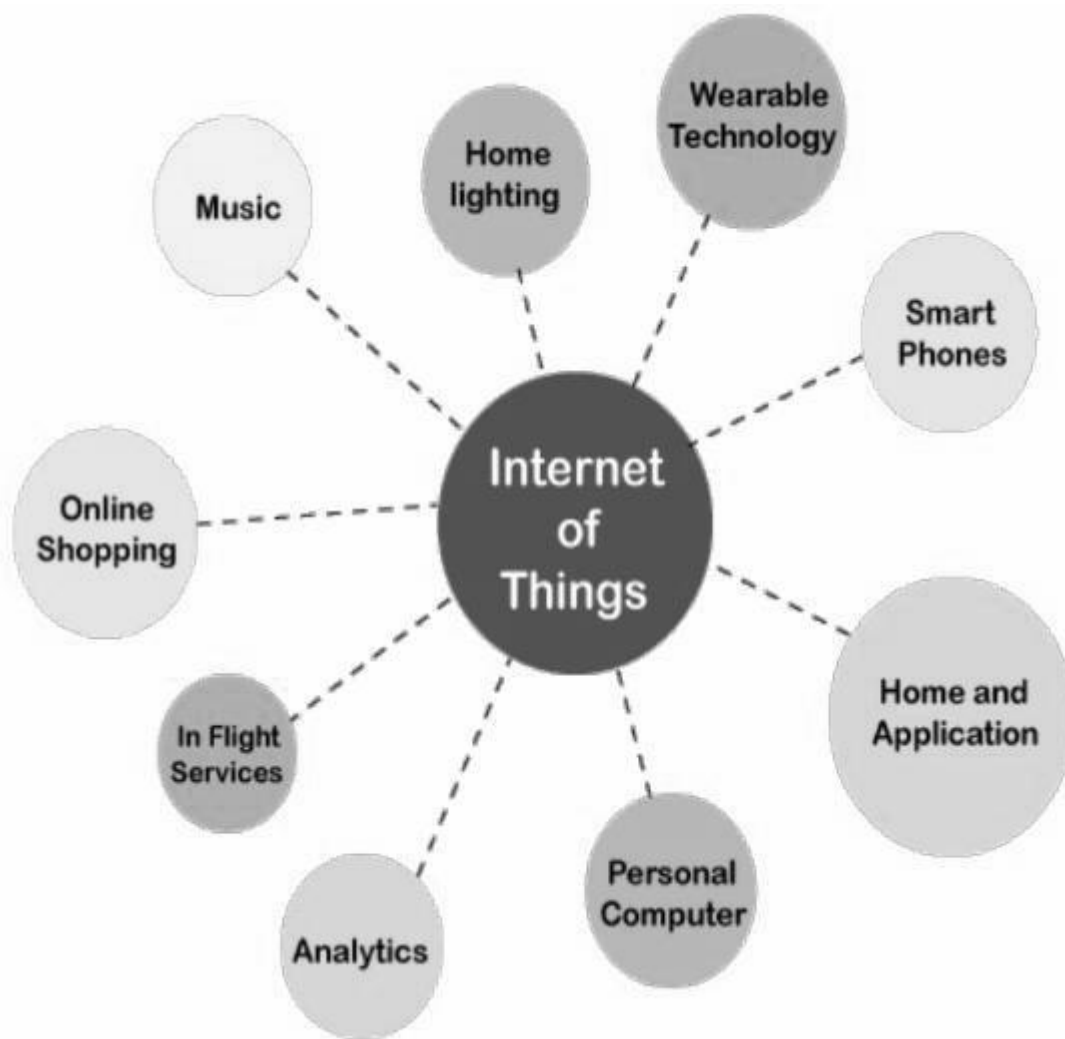


Рисунок 1.1 — Напрямки використання IoT

Системи безпеки підприємства включають низку важливих елементів, які забезпечують захист майна, персоналу та конфіденційної інформації. Ці компоненти інтегровані в єдину систему на базі Arduino (рис. 1.2), що покликана забезпечити підприємство захистом від різних загроз, таких як крадіжки, вандалізм, пожежі та кібератаки [4].

Однією з основних складових таких систем є фізична безпека. До цієї категорії належать системи контролю доступу (рис. 1.3), які обмежують доступ до певних об'єктів чи приміщень тільки для авторизованих осіб. Також важливими компонентами є системи відеоспостереження, що дозволяють охоронцям у реальному часі виявляти підозрілі події та реагувати на них.

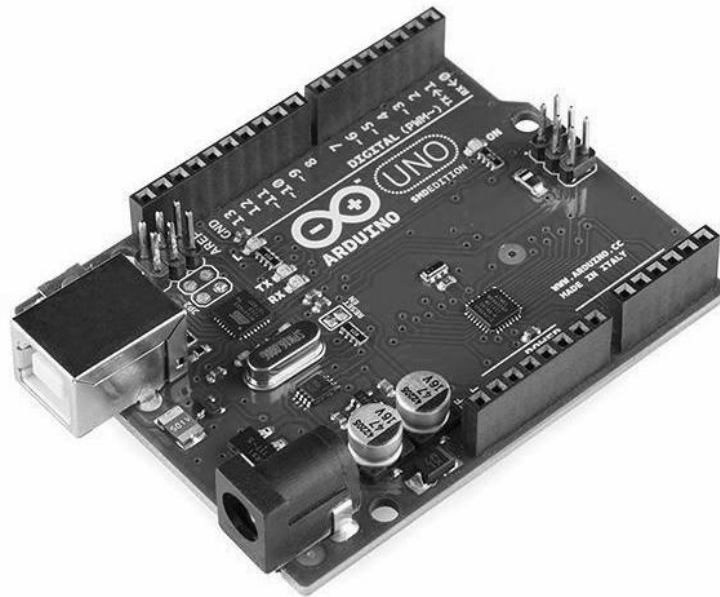


Рисунок 1.2 — Плата Arduino

Не менш важливою складовою є кібербезпека. В умовах зростання кількості кіберзагроз бізнесу необхідно забезпечити надійний захист інформації та мережевої інфраструктури від хакерських атак і витоків даних. Це включає встановлення спеціалізованого програмного забезпечення для виявлення та запобігання кіберзагрозам, а також регулярне навчання персоналу щодо правил кібербезпеки.

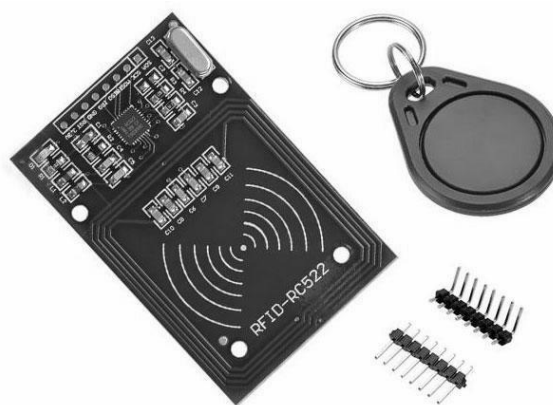


Рисунок 1.3 — Вигляд RFID-зчитувача

Ще одним важливим компонентом є системи пожежної безпеки, які виявляють загрози пожежі та негайно сповіщають про небезпеку, а також системи

оповіщення про надзвичайні ситуації, що забезпечують швидку та ефективну комунікацію з працівниками та охороною.

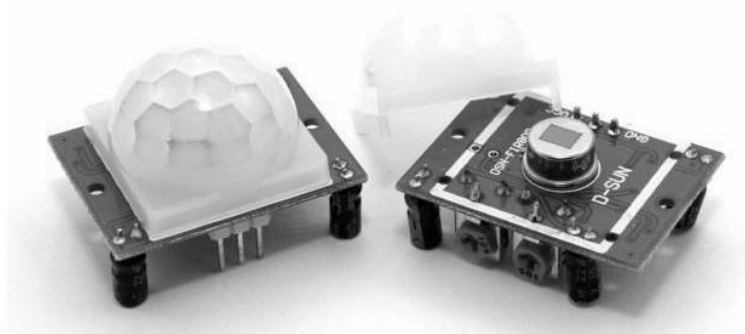


Рисунок 1.4 — Вигляд датчика руху та відкриття

Для високотехнологічних підприємств також доступні спеціалізовані системи моніторингу та контролю, які дозволяють в режимі реального часу відслідковувати стан різних систем і пристроїв, а також оперативно реагувати на будь-які несправності.

В цілому, ефективна система безпеки підприємства повинна інтегрувати різні аспекти фізичної та кібербезпеки для комплексного захисту від потенційних загроз. Така система має бути гнучкою, масштабованою і надійною, аби забезпечити безпеку в умовах постійних змін.

Існують різні види систем безпеки, які відрізняються за своїми характеристиками та призначенням, але їх загальна мета — забезпечення захисту різних аспектів діяльності підприємства або організації. Розглянемо деякі з них [5].

Фізична безпека: цей вид систем безпеки відповідає за захист фізичних об'єктів, таких як будівлі, приміщення, обладнання та майно, від несанкціонованого доступу, крадіжок і інших загроз. Системи фізичної безпеки включають в себе датчики руху, відеоспостереження, системи контролю доступу та сигналізації.

Інформаційна безпека: ця система фокусується на захисті конфіденційності, цілісності та доступності інформації. Вона охоплює заходи, спрямовані на захист комп'ютерних систем, мереж та даних від кібератак, вірусів, зловмисного

програмного забезпечення та інших цифрових загроз.

Психологічна безпека: орієнтована на створення безпечного та комфортного робочого середовища для працівників і відвідувачів. Включає в себе профілактичні заходи, психологічну підтримку, навчання з питань безпеки та процедури кризового управління для зняття стресу і забезпечення психологічного комфорту.

Розробка рішень Інтернету речей (IoT) на базі Arduino для охоронних систем підприємств може стати ефективним методом інтеграції різних видів безпеки. Завдяки використанню Arduino можна створювати пристрої, що працюють з різноманітними сенсорами та пристроями зв'язку, і програмувати їх для виконання таких функцій, як виявлення руху, зчитування даних, запис відео та аудіо сповіщення, а також віддалене управління системами безпеки через Інтернет. Такий підхід дозволяє розробити інтегровану систему для комплексного забезпечення безпеки підприємства.

Інтеграція систем безпеки в інфраструктуру підприємства є критично важливим аспектом для забезпечення захисту працівників, майна та конфіденційної інформації. Ось кілька ключових аспектів, які підкреслюють важливість такої інтеграції.

Забезпечення безпеки працівників та відвідувачів: системи контролю доступу та відеоспостереження допомагають запобігти несанкціонованому доступу до об'єктів і приміщень підприємства. Це включає контроль за входами та виходами, ідентифікацію осіб та моніторинг дій для забезпечення безпеки на території підприємства.

Захист майна та активів: системи сигналізації та виявлення вторгнень дозволяють оперативно реагувати на спроби злому або крадіжки, що допомагає запобігти втратам і пошкодженням майна та активів підприємства.

Захист конфіденційної інформації: інтеграція систем інформаційної безпеки дозволяє захистити конфіденційну інформацію від несанкціонованого доступу, витоку даних або кібератак. Це може включати захист комп'ютерних систем, шифрування даних, моніторинг мережевої активності для виявлення загроз.

Забезпечення безпеки виробничих процесів: системи безпеки можуть бути інтегровані в виробничі процеси для запобігання нещасним випадкам та забезпечення безпеки працівників, особливо під час роботи з небезпечним обладнанням і технікою.

Інтеграція цих систем дає змогу підприємствам ефективно реагувати на потенційні загрози, знижуючи ризики для безпеки та безперервності діяльності. Це, в свою чергу, сприяє покращенню репутації підприємства, зменшенню витрат та забезпеченню стабільної і продуктивної робочої атмосфери.

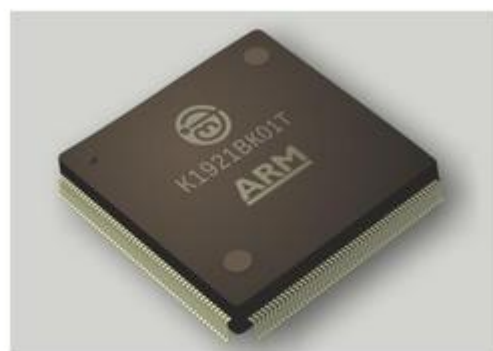
1.2 Технологічні можливості Arduino в контексті IoT

Мікроконтролери Arduino є популярними завдяки своїй простоті, доступності та широкому спектру можливостей, що робить їх ідеальним вибором для створення систем безпеки підприємства на основі Інтернету речей (IoT).

Платформи Arduino використовують мікроконтролери AVR (рис. 1.5, а) або ARM (рис. 1.5, б), які мають різні характеристики та можливості. Ці мікроконтролери оснащені вбудованим USB-портом, що полегшує підключення до комп'ютера та програмування.



а)



б)

Рисунки 1.5 — Мікроконтролери: а) AVR; б) ARM

Arduino також має велику спільноту користувачів та розробників, які активно

обмінюються досвідом і створюють нові бібліотеки та проекти.

Нижче наведено основні відмінності між мікроконтролерами AVR та ARM.

Архітектура: AVR — це 8-розрядні мікроконтролери, тоді як ARM — 32-розрядні. Завдяки більшій кількості робочих бітів, ARM здатні обробляти більші обсяги даних і виконувати складніші операції.

Продуктивність: мікроконтролери ARM зазвичай мають кращу продуктивність завдяки більшій ширині слова і кількості робочих бітів, що дозволяє ефективніше обробляти дані.

Енергоефективність: AVR є більш енергоефективними, оскільки 8-розрядні архітектури зазвичай споживають менше енергії порівняно з 32-розрядними архітектурами ARM. Це може бути важливим для застосувань з обмеженим джерелом живлення.

Вартість: мікроконтролери AVR зазвичай дешевші через простішу архітектуру і менші вимоги до ресурсів для їх виробництва.

Вибір між AVR і ARM залежить від конкретних вимог проекту, таких як продуктивність, енергоефективність, вартість і наявність необхідних ресурсів.

Головною перевагою Arduino є її простота використання, що робить платформу доступною навіть для новачків в електроніці та програмуванні. Інтуїтивно зрозумілий інтерфейс програмування дозволяє швидко освоїти основи без необхідності спеціальних знань.

Arduino підтримує велику кількість датчиків, модулів і розширень, що робить її універсальним інструментом для розробки систем безпеки. Це дозволяє розширювати функціональність пристроїв, додавати нові можливості та адаптувати їх до конкретних потреб.

Крім того, Arduino має велику кількість бібліотек та вбудованих функцій, що спрощує розробку додатків. Ці бібліотеки надають доступ до різноманітних функцій для взаємодії з датчиками, перетворювачами і механізмами управління, що мінімізує зусилля програміста.

Історія та розвиток мікроконтролерів Arduino є важливим етапом у розвитку

електроніки та робототехніки. Ось кілька основних моментів, що відображають цей шлях.

Arduino було створено в 2005 році групою інженерів: Массімо Банзі, Девідом Кушіє та Девідом Меллісом із Масачусетського технологічного інституту. Спочатку вони розробили прототип плати для студентського проекту, назвавши її Arduino. Плата була розроблена з метою спрощення процесу програмування для створення інтерактивних пристроїв. Arduino відразу привернула увагу своєю простотою та доступністю.

Випуск перших моделей також відбувся у 2005 році, тоді була випущена перша модель — Arduino Uno. Ця плата швидко здобула популярність серед хобістів, студентів і професійних розробників завдяки своєму зручному інтерфейсу програмування, доступності і можливості працювати з різноманітними електронними компонентами.

Протягом наступних років Arduino продовжував еволюціонувати, випускаючи нові моделі, що надавали ще більше можливостей. Кожна нова версія плати була покращена для більшої функціональності, підтримки нових компонентів і забезпечення кращої сумісності. Розробники створювали безліч додатків та розширень, що дозволяли використовувати мікроконтролери Arduino в різних сферах, від робототехніки до Інтернету речей.

Arduino також став важливим елементом у русі відкритого програмного забезпечення, оскільки розробники платформи активно підтримують обмін кодом, ідеями та проектами. Це стимулювало створення спільноти користувачів і розробників, що в подальшому сприяло інноваціям та поширенню технологій у різних галузях.

На сьогоднішній день Arduino залишається однією з найбільш популярних платформ для розробки прототипів електронних пристроїв. Платформа широко використовується у численних проектах, від навчальних до комерційних, і залишається важливим інструментом для розробників по всьому світу.

Історія Arduino не тільки свідчить про технологічний прогрес, але й

демонструє важливість спільноти в створенні та розвитку інноваційних ідей у галузі електроніки та програмування.

Платформа Arduino є однією з найпоширеніших і найбільш ефективних для розробки рішень у галузі Інтернету речей (IoT). Ось деякі ключові функції цієї платформи для створення IoT рішень.

Arduino відома своєю доступністю та простотою для користувачів, навіть тих, хто не має досвіду в програмуванні. Інтерфейс програмування є інтуїтивно зрозумілим, що дозволяє швидко освоїти основи та розпочати роботу з платформою.

Arduino підтримує великий асортимент датчиків, модулів та розширень, що дає змогу створювати різноманітні проекти. Ви можете підключати датчики для вимірювання різних параметрів, таких як температура, вологість, освітлення, рух тощо, і використовувати їх у своїх IoT системах.

Arduino здатна виконувати різноманітні завдання, від зчитування даних з датчиків до управління пристроями. Ви можете програмувати мікроконтролер для передачі даних через мережу, контролювати виконавчі механізми та організовувати взаємодію з іншими пристроями за допомогою різних комунікаційних інтерфейсів.

Arduino має активну та велику спільноту, яка постійно ділиться досвідом, документацією, прикладами коду та пропонує допомогу в разі виникнення питань. Це дозволяє швидко знаходити рішення для будь-яких проблем і забезпечує підтримку під час розробки проектів.

Arduino підтримує кілька мов програмування, таких як C та C++, що дає змогу створювати складні програми з багатим функціоналом і ефективно реалізовувати навіть складні IoT рішення.

Загалом, Arduino є потужним і зручним інструментом для розробки IoT проектів, поєднуючи простоту використання з великим вибором можливостей і підтримкою великої спільноти користувачів. Це робить платформу ідеальним вибором для будь-якого IoT проекту.

Порівняння Arduino з іншими мікроконтролерами для Інтернету речей (IoT) допомагає визначити, який вибір найкраще підходить для конкретних потреб вашого проекту. Ось деякі важливі аспекти цього порівняння:

Arduino відомий своєю простотою у використанні, що робить його чудовим вибором для новачків. Його інтерфейс програмування є інтуїтивно зрозумілим, і навіть люди без досвіду програмування можуть швидко освоїти основи. У порівнянні з іншими платформами, такими як Raspberry Pi або STM32, Arduino виграє за рахунок своєї доступності для початківців.

Arduino надає велику кількість бібліотек, датчиків та модулів для розширення функціональності. Завдяки широкому асортименту доступних аксесуарів, ви можете створювати різноманітні проекти, від простих до складних. У той же час ESP32 або STM32 також мають розширені можливості, зокрема, підтримку Wi-Fi, Bluetooth і складніші апаратні можливості.

Arduino — це доступний варіант для багатьох, особливо для освітніх проектів і хобістів. Однак ESP8266 і ESP32 можуть бути навіть дешевшими, а Raspberry Pi пропонує додаткову обчислювальну потужність за невелику різницю в ціні.

Якщо ваш проект вимагає більшої обчислювальної потужності або швидкості, то Raspberry Pi або STM32 будуть кращими варіантами. Вони здатні обробляти складніші алгоритми або запускати більш потужні програми, ніж Arduino, яке обмежене простими задачами.

Arduino має величезну спільноту користувачів, що робить його привабливим варіантом для тих, хто шукає підтримку та готові приклади коду. Хоча інші платформи, як Raspberry Pi та ESP32, також мають великі спільноти, Arduino все ж залишається одним з лідерів завдяки своїй популярності серед хобістів і освітніх проектів.

Альтернативи Arduino наведені нижче.

Raspberry Pi: це одноплатний комп'ютер, що підходить для більш складних проектів, які потребують операційної системи та великих обчислювальних потужностей. Raspberry Pi підтримує різні мови програмування, такі як Python,

C++, і більше, але є дорожчим варіантом для простих IoT рішень.

Мікроконтролери PIC мають велику різноманітність моделей, від простих до високопродуктивних, і можуть бути хорошим вибором для проектів, що вимагають численних входів/виходів. Програмуються в C чи Basic.

STM32: ці мікроконтролери з архітектурою ARM часто використовуються для складних проектів, що потребують високої швидкості та потужності. Підтримують кілька мов програмування, включаючи C та Python.

ESP8266 та ESP32: ці мікроконтролери ідеально підходять для IoT проектів завдяки вбудованим можливостям Wi-Fi та Bluetooth. Вони є економічними варіантами для підключення пристроїв до мережі, підтримують C++ та MicroPython.

AVR: мікроконтролери AVR, такі як ті, що використовуються в Arduino, є чудовим вибором для простих і доступних проектів. Вони мають добре розвинену підтримку та доступність бібліотек для програмування.

1.3 Висновки щодо доцільності використання Arduino у проектах безпеки

Після ретельного аналізу концепції Інтернету речей (IoT) та огляду можливостей мікроконтролерів Arduino для застосування в системах безпеки підприємств можна зробити кілька важливих висновків, які відображають ключові переваги та перспективи використання цих технологій [6].

По-перше, IoT виявляється потужним інструментом для забезпечення безпеки підприємства. Його впровадження дозволяє здійснювати віддалений моніторинг і контроль системи безпеки, швидко реагувати на небезпечні ситуації та мінімізувати ризик збитків. Це особливо актуально в сучасних умовах, коли безпека підприємств набуває все більшого значення через зростання кіберзагроз та інших потенційних ризиків.

Мікроконтролери Arduino займають центральне місце в реалізації IoT-рішень для безпеки підприємств. Вони мають ряд переваг, серед яких простота використання, гнучкість і доступність. Мікроконтролери Arduino дають

можливість швидко створювати різноманітні охоронні пристрої та системи, реалізуючи різні сценарії — від датчиків руху та відкриття до систем відеоспостереження та контролю доступу.

Однією з найбільших переваг мікроконтролерів Arduino є їх активна спільнота користувачів і розробників. Ця спільнота забезпечує постійний потік нових ідей, бібліотек і розширень, що робить Arduino ще більш потужним і універсальним інструментом для розробки систем безпеки підприємств.

Таким чином, розробка IoT-рішень для систем безпеки підприємств з використанням Arduino є перспективним напрямком, який може значно підвищити рівень безпеки і знизити ризик збитків для підприємств у сучасному цифровому світі.

2 ПОБУДОВА АРХІТЕКТУРИ СИСТЕМИ НА ОСНОВІ ARDUINO

2.1 Сенсори руху та відкриття: принцип дії і підключення

Розглядаючи концепцію Інтернету речей (IoT) і можливість використання мікроконтролерів Arduino в системах безпеки, можна визначити ключові аспекти архітектури таких систем.

По-перше, архітектура системи безпеки на основі Arduino зазвичай складається з набору компонентів, які включають датчики (наприклад, датчики руху та відкриття), камери відеоспостереження, сигналізацію та системи контролю доступу. Ці компоненти працюють разом, щоб виявляти потенційні загрози та реагувати на них.

Датчики відіграють ключову роль у системі та збирають інформацію про стан навколишнього середовища. Наприклад, датчики руху можуть фіксувати присутність непроханих осіб або визначати незвичну активність в охоронюваній зоні. Датчики відкриття, такі як сенсори на дверях чи вікнах, можуть сигналізувати про їхнє незаконне відкриття.

Для візуального спостереження за об'єктами та реєстрації подій використовуються камери відеоспостереження. Вони допомагають отримувати зображення потенційних загроз, що дозволяє здійснити точну ідентифікацію та прийняти відповідні заходи.

Системи сигналізації служать для сповіщення користувачів про потенційну небезпеку або вторгнення. Це можуть бути звукові або світлові сигнали, що миттєво привертають увагу, а також повідомлення на мобільні пристрої відповідальних осіб, що дозволяє оперативно реагувати.

Системи контролю доступу регулюють доступ до об'єктів, таких як корпоративні приміщення чи важливі зони. Вони можуть використовувати різноманітні методи ідентифікації та автентифікації, зокрема ключ-карти, біометричні дані (відбитки пальців, розпізнавання обличчя) або персональні PIN-коди.

Основними перевагами цієї архітектури є складність та інтегрованість.

Кожен компонент системи виконує свою специфічну роль, але разом вони забезпечують повний захист об'єкта. Крім того, використання Arduino як базової платформи дозволяє легко розширювати та оновлювати систему за потреби, додавати нові функції чи пристрої, що робить систему гнучкою і адаптованою до змінних умов.

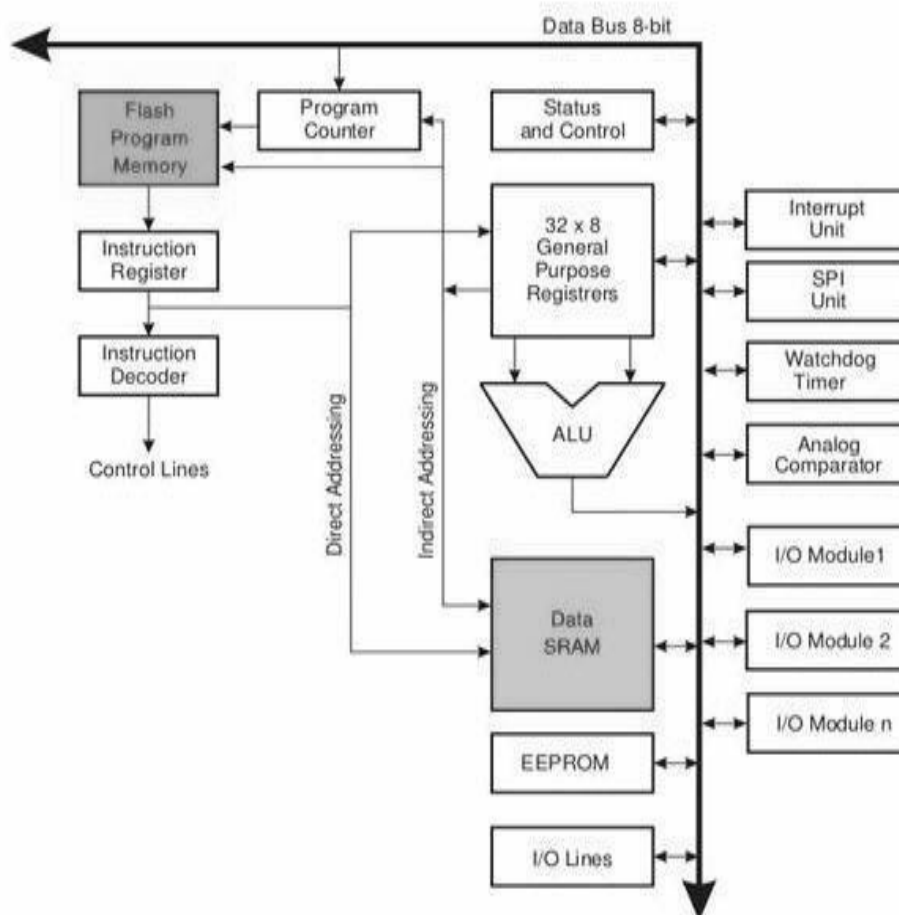


Рисунок 2.1 — Архітектура ядра AVR

Системи безпеки підприємств не можуть обійтися без датчиків руху та відкривання, оскільки вони є ключовими елементами виявлення загроз та несанкціонованих проникнень. Їхнє призначення та принципи функціонування мають велике значення для забезпечення надійного захисту об'єктів.

Датчик руху — це пристрій, який фіксує переміщення в заданій зоні дії. Він використовується для виявлення присутності людей або об'єктів у контрольованому просторі. Існує кілька типів таких пристроїв, серед яких —

інфрачервоні (рис. 2.2) та ультразвукові (рис. 2.3) датчики. У разі виявлення руху сигнал передається на мікроконтролер Arduino, який аналізує дані та виконує запрограмовані дії системи безпеки [7].



Рисунок 2.2 — Інфрачервоний датчик руху



Рисунок 2.3 — Ультразвуковий датчик HC-SR04 SR602

Ще одним важливим компонентом охоронної системи є датчики відкривання. Вони виявляють факт відкриття дверей, вікон або інших потенційних точок доступу (рис. 2.4). Принцип дії таких пристроїв зазвичай базується на зміні магнітного поля. При відкритті охоронюваного об'єкта датчик надсилає сигнал до мікроконтролера на платформі Arduino, який у свою чергу активує відповідні заходи реагування.



Рисунок 2.4 — Герконовий датчик відкривання дверей²²

Основною перевагою подібних датчиків є їхня здатність працювати в режимі реального часу, що забезпечує негайне реагування на зміну ситуації. Завдяки цьому забезпечується безперервний контроль над станом об'єкта та своєчасне виявлення загроз.

Підсумовуючи, можна сказати, що датчики руху та відкривання є незамінними складовими системи безпеки на основі Arduino. Вони відіграють вирішальну роль у швидкому виявленні порушень і сприяють запобіганню можливим небезпекам. Їхня ефективність, швидкодія та надійність відповідають вимогам сучасних безпекових систем.

Аналіз принципів дії сенсорів руху та відкриття у межах безпекових систем на базі платформи Arduino має важливе значення для формування адекватної реакції на загрози та події, що виникають у контрольованому середовищі. Ці пристрої відіграють ключову роль у моніторингу доступу та переміщення в межах охоронюваної території, забезпечуючи своєчасне виявлення потенційних порушень і запуск відповідних механізмів реагування.

Датчики руху працюють на основі реєстрації змін інфрачервоного (ІЧ) випромінювання, яке відбивається від об'єктів у зоні дії. Вони реагують на присутність рухомих тіл, таких як люди або тварини. При виявленні переміщення пристрій генерує сигнал, який надходить на мікроконтролер Arduino для подальшої обробки та запуску певних дій у системі.

До поширених типів сенсорів також належать ультразвукові датчики, які

активно використовуються в сучасних безпекових рішеннях, зокрема в системах на основі Arduino. Їхній принцип роботи передбачає випромінювання ультразвукових імпульсів і визначення часу, за який сигнал повертається після відбиття від перешкод.

Основні етапи функціонування ультразвукового сенсора включають:

- генерація імпульсу — пристрій випромінює короткий звуковий сигнал ультразвукового діапазону;
- відбиття хвилі — сигнал відбивається від навколишніх об'єктів (наприклад, стін, меблів, людей);
- прийом сигналу — відбитий імпульс фіксується приймачем сенсора;
- обчислення часу проходження — фіксується часовий інтервал між передачею та отриманням сигналу;
- розрахунок відстані — використовуючи дані про швидкість звуку, система визначає відстань до об'єкта.

Ультразвукові сенсори мають низку переваг, що робить їх актуальними для застосування в корпоративних охоронних системах:

- точність вимірювання — забезпечують високий рівень чутливості та здатність до точного визначення положення об'єкта;
- широкий діапазон дії — можуть охоплювати значну зону спостереження;
- швидкодія — дозволяють оперативно реагувати на зміну ситуації в охоронюваному середовищі.

Такі датчики виступають ефективним доповненням до стандартних засобів контролю руху та відкриття, підвищуючи рівень моніторингу та захисту будівель на основі платформи Arduino.

Щодо сенсорів відкриття, то вони застосовуються для визначення факту відкривання або закривання дверей, вікон, шухляд тощо. Їхній механізм роботи базується на виявленні змін у магнітному полі, що свідчить про зміну стану охоронюваного об'єкта. У разі зміни положення (відкриття чи закриття) формується електричний сигнал, який надсилається до контролера системи.

Інтеграція обох типів датчиків у єдину систему безпеки підприємства дозволяє ефективно контролювати пересування та доступ до приміщення, фіксувати несанкціоновані дії, а також своєчасно інформувати адміністрацію або активувати інші захисні засоби. Такі можливості забезпечують стабільну та надійну роботу системи безпеки, яка розробляється з використанням Arduino.

Однією з ключових складових при створенні системи безпеки на базі Arduino є правильне підключення сенсорів та ефективна обробка даних, які вони генерують. Нижче розглянемо основні етапи підключення датчиків і подальшу взаємодію з мікроконтролером.

Етапи підключення сенсорів:

- вибір відповідного датчика — перед підключенням необхідно обрати тип сенсора залежно від функціональних потреб системи: це можуть бути сенсори руху, відкриття, диму тощо;

- підготовка мікроконтролера — для забезпечення сумісності обраного сенсора з Arduino слід правильно підібрати пін (контакт) на платі для підключення, у деяких випадках можуть знадобитися додаткові елементи — резистори, модулі-реле або адаптери;

- фізичне підключення до портів Arduino — сенсор з'єднується з мікроконтролером за допомогою відповідних дротів або конекторів; його виходи підключаються до цифрових або аналогових входів плати залежно від типу сигналу;

- програмне забезпечення для зчитування даних — після фізичного з'єднання необхідно реалізувати програмну частину — код для читання та обробки даних із сенсорів, програмування здійснюється в середовищі Arduino IDE з використанням мови C/C++.

У прикладі на рисунку 2.4 показано, як можна налаштувати читання стану з двох сенсорів — датчика руху та датчика відкривання:

У цьому коді два входи призначені для сенсорів, які конфігуруються як вхідні в `setup()`. У циклі `loop()` постійно зчитується їхній стан за допомогою `digitalRead()`

та передається до послідовного монітора для виведення. Далі відбувається обробка даних, отриманих від сенсорів, наступний крок — зчитування сигналів: Arduino отримує вхідні сигнали, які можуть мати аналогову або цифрову форму, залежно від конструкції сенсора. В подальшому зібрана інформація аналізується.

Зчитані значення аналізуються у програмному середовищі. Наприклад, якщо сенсор фіксує рух, система може класифікувати подію як потенційно небезпечну.

На основі результатів аналізу можуть виконуватись різні дії: запуск звукової чи світлової сигналізації, запис події до логів або передача повідомлення на віддалений пристрій.

```
#include <Arduino.h>
#include <Wire.h>
const int motionSensorPin = 2;
const int openSensorPin = 3;
void setup() {
  pinMode(motionSensorPin, INPUT);
  pinMode(openSensorPin, INPUT);
  Serial.begin(9600);}

void loop() {
  int motionState = digitalRead(motionSensorPin);
  int openState = digitalRead(openSensorPin);
  Serial.print("Motion Sensor: ");
  Serial.println(motionState);
  Serial.print("Open Sensor: ");
  Serial.println(openState);
  delay(1000);}
```

2.2 Відеоспостереження як складова безпеки

Камери відеоспостереження, інтегровані в корпоративні системи безпеки на платформі Arduino, є важливим компонентом для забезпечення контролю за об'єктами та охорони. Вони дозволяють здійснювати моніторинг подій у реальному часі, стежити за діяльністю персоналу та інших осіб, а також записувати відео для подальшого аналізу (рис. 2.5).



Рисунок 2.5 — OV7670 камера для Arduino

Камери відеоспостереження відрізняються різними технічними характеристиками, що впливають на їх ефективність і функціональність. Одним із основних параметрів є роздільна здатність, яка визначає чіткість і детальність зображення. Чим вища роздільна здатність, тим якісніше зображення, що, в свою чергу, сприяє більш точній ідентифікації об'єктів і подій.

Деякі моделі камер мають вбудовані системи нічного бачення, що дозволяють здійснювати відеоспостереження в умовах поганого освітлення або повної темряви. Це особливо важливо для безперервного контролю об'єкта незалежно від часу доби.

Ще одним важливим параметром є кут огляду камери. Широкий кут дозволяє охоплювати більшу площу та забезпечує ефективніший моніторинг простору.

Вибір типу камери для системи безпеки є важливим етапом у розробці ефективного рішення на основі Arduino. Розглянемо основні види камер, їх характеристики та особливості.

Аналогова камера — це традиційний тип камери, який використовує аналогове з'єднання для передачі відео на монітор або відеореєстратор. Ці камери прості в установці і експлуатації, однак вони мають обмежену роздільну здатність і функціональні можливості.

ІР-камери — ці камери підключаються до мережі через Ethernet або Wi-Fi та передають відео по мережі на комп'ютер або сервер. Вони пропонують високу роздільну здатність, додаткові функції, такі як нічний режим і детектори руху, а

також дозволяють дистанційно керувати пристроєм через Інтернет.

Камери HD-SDI — ці камери забезпечують передачу високоякісного відео з роздільною здатністю від 720p до 4K через коаксіальний кабель. Вони часто використовуються в професійних системах відеоспостереження, де важлива висока якість зображення.

PTZ-камери (панорамні, нахилиючі, масштабовані) дозволяють панорамувати, нахилити та змінювати масштаб зображення за допомогою пульта дистанційного керування або програмного забезпечення. Це дає змогу точно відстежувати та фіксувати об'єкти з високою деталізацією.

Кожен тип камери має свої переваги та обмеження. Вибір залежить від специфічних вимог вашої системи безпеки, таких як охоплення території, якість зображення та необхідність додаткових функцій. При розробці системи на базі Arduino важливо врахувати сумісність камер з мікроконтролерами, можливість підключення додаткових модулів та збереження й обробку відеоінформації.

Для інтеграції камер у систему безпеки на основі Arduino можна використовувати різноманітні модулі камер і спеціалізовані бібліотеки для управління ними. Основним принципом є підключення камери до мікроконтролера Arduino, програмування її для захоплення та зберігання відеоданих.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Camera.h>
const char* ssid = "Назва_мережі";
const char* password = "Пароль_мережі";
Camera cam;
void setup() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);}
  cam.begin();}
void loop() {
  cam.capture();
  delay(1000);}
```

В цьому прикладі використовується мікроконтролер Arduino з модулем Wi-Fi ESP8266 і камерою. Після підключення до мережі Wi-Fi необхідно ініціалізувати

камеру та знімати кадри в нескінченному циклі в функції `loop()`. Отримані відеодані можна обробляти або зберігати на віддалених серверах за допомогою протоколів передачі даних, таких як HTTP або MQTT.

В системі безпеки на базі Arduino камери відеоспостереження можуть бути підключені до мікроконтролера через різні інтерфейси, такі як USB, Ethernet або бездротові з'єднання. Мікроконтролер може керувати камерами, записувати відео, зберігати його і аналізувати активність для виявлення подій або загроз.

2.3 Організація контролю доступу

Системи контролю доступу є важливим елементом для обмеження та моніторингу доступу до окремих зон або будівель. Вони забезпечують конфіденційність та захист ресурсів і інформації, контролюючи доступ користувачів. Системи можуть бути побудовані з використанням різних технологій та компонентів, таких як електромагнітні замки, сенсорні карти, біометричні системи ідентифікації та інші, що дозволяють ефективно ідентифікувати користувачів та авторизувати їх перед наданням доступу до об'єктів (рис. 2.6).



Рисунок 2.6 — Зовнішній вигляд зчитувача відбитків пальців

Однією з основних переваг таких систем є їхня гнучкість і можливість адаптації до конкретних потреб бізнесу. Системи можуть бути інтегровані з іншими компонентами безпеки для створення комплексної системи захисту. Мікроконтролери Arduino часто використовуються для керування процесами автентифікації та прийняття рішень щодо доступу на основі інформації від

різноманітних датчиків та пристроїв. Це дозволяє ефективно контролювати доступ до об'єктів та реагувати на потенційні загрози.

Системи контролю доступу, створені на основі Arduino, надають широкі можливості для реалізації різних сценаріїв безпеки, адаптуючи їх до потреб підприємства. Це дозволяє забезпечити високий рівень безпеки і захисту об'єктів, створюючи ефективну систему контролю доступу.

Системи контролю доступу можна класифікувати на автономні та мережеві.

Автономна система є оптимальним і доступним варіантом для об'єктів з однією точкою проходу. Вона не потребує підключення до комп'ютера, що робить її зручною для використання в віддалених або ізольованих місцях, наприклад, на складах. В таких системах можна застосовувати різні типи зчитувачів для контролю доступу.

Системи контролю доступу є важливою складовою корпоративних систем безпеки, створених на основі мікроконтролерів Arduino. Вони забезпечують контроль за входом у певні зони або будівлі, обмежуючи доступ лише авторизованим користувачам.

Система контролю доступу має на меті обмеження доступу до певних ресурсів або територій лише для користувачів, що мають відповідні права. Для цього використовуються різноманітні методи ідентифікації, такі як картки доступу, RFID мітки чи біометричні дані (наприклад, відбитки пальців чи розпізнавання обличчя).

Основні складові системи контролю доступу:

— система ідентифікації: це компонент, який відповідає за визначення користувача, який намагається отримати доступ, ідентифікація може здійснюватися за допомогою зчитувачів карток, біометричних сканерів або систем RFID;

— контролер доступу: це пристрій або програмне забезпечення, яке обробляє інформацію, отриману від системи ідентифікації, і приймає рішення щодо надання

доступу, як контролер можна використовувати мікроконтролер Arduino, який інтегрується з іншими компонентами системи;

— електронні замки: ці механізми фізично контролюють доступ до приміщень, це можуть бути електромагнітні замки, електромеханічні засуви чи інші механізми, що відкривають або закривають двері на основі команди, отриманої від контролера доступу;

— система моніторингу та журналювання: цей компонент фіксує всі спроби доступу до системи, включаючи відмови в доступі, журнали доступу допомагають в аналізі безпеки та виявленні потенційних вразливостей;

— сповіщення: ці системи надають негайне сповіщення про несанкціоновані спроби доступу або інші інциденти безпеки, сповіщення можуть бути в формі звукових сигналів, повідомлень на мобільний телефон або надісланих на центральний сервер безпеки.

Загалом, система контролю доступу працює за принципом ідентифікації користувача, перевірки його прав доступу та фізичного блокування або надання доступу залежно від результату цієї перевірки. Це робить системи контролю доступу важливими елементами безпеки в будівлях та на підприємствах.

Підключення та програмування електронних замків для системи контролю доступу на основі Arduino є важливою складовою розробки корпоративної безпеки. Цей процес включає кілька етапів, від вибору замка до тестування та моніторингу. Розглянемо основні етапи.

Вибір електронного замка: перш ніж вибрати електронний замок, необхідно ретельно проаналізувати потреби вашої системи безпеки. Важливо враховувати тип замка (електромагнітний, електромеханічний тощо), сумісність з Arduino та можливість програмування. Замок має відповідати вимогам безпеки та бути легко інтегрованим з іншими компонентами системи.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Camera.h>
```

```

const char* ssid = "Назва_мережі";
const char* password = "Пароль_мережі";
Camera cam;
void setup() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000); }
  cam.begin();}
void loop() {
  cam.capture();
  delay(1000);}

```

Підключення до Arduino: після вибору замка необхідно правильно підключити його до мікроконтролера Arduino. Це включає підключення проводів до відповідних цифрових або аналогових портів Arduino, а також забезпечення належного живлення замка, якщо це необхідно. Варто враховувати необхідність додаткових компонентів, таких як реле, для керування високими навантаженнями замка.

Програмування замка: після підключення замка до Arduino потрібно розробити програмне забезпечення для управління ним. Це включає написання коду, що дозволяє відкривати або закривати замок залежно від сигналів, що надходять від системи контролю доступу. Програма має забезпечити належну обробку подій, наприклад, підтвердження або відмову в доступі. Ось приклад базового коду для відкриття та закриття електронного замка. У цьому коді використовується пін Arduino для керування замком. Функції `openLock()` та `closeLock()` дозволяють відповідно відкрити та закрити замок.

Тестування та налагодження: після написання програмного забезпечення важливо провести тестування замка для перевірки його роботи. Тестування дозволяє виявити можливі проблеми в коді або підключенні компонентів. Важливо переконатися, що замок працює правильно в усіх сценаріях (наприклад, при наданні або відмові в доступі). Тестування також повинно забезпечити сумісність замка з іншими компонентами системи.

Впровадження та моніторинг: після успішного тестування замок можна інтегрувати у корпоративну систему контролю доступу. Важливо забезпечити

регулярний моніторинг роботи замків, щоб оперативно виявляти несправності та забезпечити безперебійну роботу системи безпеки.

Програмування електронних замків для системи контролю доступу за допомогою Arduino дозволяє створити ефективні та надійні системи безпеки. Для досягнення оптимальних результатів необхідно враховувати всі етапи підключення, налаштування та тестування, щоб забезпечити високий рівень безпеки на підприємстві.

2.4 Узагальнення побудови архітектури охоронної системи

Аналіз архітектури системи безпеки, побудованої на платформі Arduino, дозволяє зробити кілька важливих висновків щодо її функціональності, ефективності та потенційних можливостей.

По-перше, використання датчиків руху та відкриття дверей в складі системи сигналізації дає змогу системі оперативно реагувати на зміни в навколишньому середовищі об'єкта. Це дозволяє своєчасно виявляти потенційні загрози безпеці, наприклад, несанкціоновані вторгнення або технічні несправності. Зокрема, використання таких сенсорів допомагає підвищити рівень безпеки підприємства та оперативно реагувати на будь-які надзвичайні ситуації.

По-друге, інтеграція камер відеоспостереження в систему забезпечує можливість візуального моніторингу об'єкта в реальному часі. Відеокамери не тільки фіксують інциденти, але й служать важливим інструментом для аналізу ситуацій, визначення рівня загрози і розробки відповідних реакцій. Крім того, відеозаписи можуть бути збережені як докази для подальшого аналізу або використання в розслідуваннях інцидентів, що дозволяє підвищити правову та оперативну ефективність системи.

По-третє, системи контролю доступу забезпечують належний рівень безпеки, обмежуючи доступ до критичних або заборонених зон об'єкта лише для авторизованих користувачів. Цей компонент дозволяє ефективно управляти конфіденційністю, запобігаючи несанкціонованому доступу і знижуючи ризик

витоку інформації або пошкодження майна. За допомогою технологій, таких як RFID, біометричні сканери та електронні замки, ці системи забезпечують високий рівень захисту [17].

Загалом, архітектура системи безпеки на основі Arduino відображає сучасний підхід до організації комплексної безпеки об'єкта. Інтеграція різних компонентів, таких як датчики, камери відеоспостереження та системи контролю доступу, створює ефективний механізм моніторингу та захисту, який відповідає актуальним вимогам безпеки.

3 РЕАЛІЗАЦІЯ ПРОТОТИПУ ТА ПРОГРАМНА ІНТЕГРАЦІЯ

3.1 Комплектуючі та технічне оснащення

Розробка та впровадження захищеної системи безпеки на платформі Arduino включає кілька ключових етапів — від вибору компонентів до програмування мікроконтролера та тестування прототипу. Нижче наведено детальний огляд кожного етапу цього процесу.

Перший етап — вибір комплектуючих та обладнання. Визначення вимог до системи безпеки: першочергово необхідно проаналізувати потреби клієнта, тип об'єкта, що охороняється, та специфікації безпеки, щоб визначити вимоги до системи.

Вибір датчиків: на основі вимог вибираються відповідні датчики руху, відкриття дверей, диму, температури та інші сенсори, що дозволяють оперативно реагувати на зміни в навколишньому середовищі.

Вибір камер відеоспостереження: при виборі камер враховуються такі параметри, як роздільна здатність, кут огляду, можливості нічного бачення та інші функції, що впливають на якість відеоспостереження.

Вибір модулів зв'язку: необхідно вибрати відповідний модуль для мережевого зв'язку — Wi-Fi, Bluetooth чи Ethernet, в залежності від необхідної швидкості передачі даних та умов установки системи.

Другий етап — програмування мікроконтролерів Arduino. Програмне забезпечення для обробки сигналів від датчиків: необхідно написати програмне забезпечення для зчитування та обробки сигналів, що надходять від датчиків руху та відкриття. Це дозволяє реалізувати реакцію системи на зміну стану в зоні спостереження.

Розробка алгоритмів реагування: необхідно створити алгоритми для спрацьовування системи сигналізації, відеоспостереження та інші механізми реагування на події, що відбуваються.

Реалізація повідомлень про збій: для забезпечення оперативної реакції потрібно запрограмувати мікроконтролер на надсилання повідомлень

адміністраторам або відповідальним особам у разі збою системи або виявлення підозрілих подій.

Останній, третій етап — складання та тестування прототипів системи. Розробка системи безпеки на базі Arduino є складним процесом, що вимагає глибоких знань у галузях електроніки, програмування та безпеки. У порівнянні з традиційними програмними додатками, розробка систем безпеки на основі мікроконтролерів вимагає уваги до фізичних аспектів взаємодії апаратного забезпечення, а також адаптації програмного забезпечення до специфічних вимог безпеки.

Складання компонентів системи: після вибору комплектуючих та підготовки мікроконтролера до роботи, слід налаштувати фізичне з'єднання датчиків, камер та інших компонентів з Arduino, щоб забезпечити коректну взаємодію між усіма частинами системи.

Тестування прототипу: важливим етапом є тестування системи в реальних умовах. Це дозволяє виявити проблеми з точністю роботи датчиків, збої в програмному забезпеченні або помилки в апаратних компонентах. Тестування також допомагає перевірити надійність з'єднань та стабільність роботи системи в умовах реальної експлуатації.

Аналіз результатів тестування: після тестування важливо проаналізувати отримані результати та порівняти їх з існуючими системами безпеки. Це дасть змогу оцінити ефективність розробленої системи, виявити можливості для вдосконалення і підвищити надійність системи в подальших версіях.

Розробка та впровадження системи безпеки на платформі Arduino — це складний, але важливий процес, що вимагає ретельного підходу до вибору комплектуючих, програмування та тестування. Тільки після ретельного тестування і вдосконалення можна досягти високої ефективності та надійності системи, яка відповідає вимогам безпеки та захисту об'єктів.

Правильний вибір компонентів і обладнання є критично важливим етапом при розробці системи безпеки на платформі Arduino. Для цього необхідно

враховувати технічні характеристики, сумісність з іншими компонентами та можливість інтеграції в єдину систему. Ось основні аспекти, на які слід звернути увагу при виборі компонентів.

Датчики руху та відкриття використовуються для виявлення руху та відкриття дверей чи вікон можна використовувати різноманітні датчики, зокрема:

- інфрачервоні (PIR) датчики — найбільш поширені для виявлення руху, вони працюють на основі змін температури, викликаних рухом людини;
- ультразвукові датчики використовуються для вимірювання відстані та виявлення руху в конкретних зонах.

При виборі датчиків важливо враховувати:

- радіус дії — як далеко може виявляти датчик рух;
- точність — здатність правильно розпізнавати рух чи зміну в навколишньому середовищі;
- надійність і вартість — чи відповідають ці параметри вашим вимогам та бюджету.

Камери відеоспостереження є важливим компонентом системи відеоспостереження. Вони допомагають не тільки записувати інциденти, але й здійснювати візуальний моніторинг у реальному часі. Вибір камер залежить від таких параметрів:

- якість зображення — роздільна здатність та здатність до зйомки в умовах поганого освітлення;
- швидкість передачі даних — здатність камери передавати зображення в реальному часі, що є критичним для моніторингу;
- функціональність — можливості руху камери (PTZ, фіксовані), підтримка інтерфейсів для підключення до Arduino (USB, IP).

Електронні замки та системи контролю доступу використовуються для обмеження доступу до певних приміщень використовуються різні типи електронних замків:

- електромагнітні замки — прості в установці і надійні;

— електромеханічні замки — можуть бути більш складними в налаштуванні, але мають високий рівень безпеки;

— сенсорні панелі — для введення пін-кодів або використання біометричних даних (відбитків пальців чи сканування обличчя);

— RFID-картки — для ідентифікації користувачів на основі карт або міток, які зчитуються зчитувачами.

Ці замки інтегруються з мікроконтролером Arduino, щоб забезпечити автоматичне відкриття або закриття замків на основі верифікації користувача.

Мікроконтролер Arduino є основною платформою для централізованого управління системою безпеки. Він дозволяє інтегрувати різні модулі та компоненти. Ось деякі з них:

— Ethernet або Wi-Fi shield — для підключення до Інтернету і забезпечення можливості віддаленого моніторингу;

— РК-екран — для виведення інформації або статусів системи;

— реле та серво-механізми — для управління електричними замками та іншими механізмами;

— сенсори температури та вологості — для додаткового контролю навколишнього середовища.

Важливі аспекти при виборі компонентів: інтеграція компонентів, сумісність і простота програмування, тестування компоєкетів.

Інтеграція компонентів: усі частини системи повинні бути сумісними та легко інтегруватися в єдину систему.

Сумісність і простота програмування: Arduino має велику спільноту та бібліотеки для різних модулів, що полегшує програмування.

Тестування: перед впровадженням важливо протестувати всі компоненти на практиці, щоб переконатися в їхній працездатності та ефективності.

Цей підхід дозволяє створити систему, яка буде надійною, гнучкою та масштабованою для різних сценаріїв безпеки. Підбір правильних компонентів є основою для успішної реалізації системи безпеки на базі Arduino.

Вибір компонентів для системи безпеки на базі Arduino є важливим етапом, оскільки він визначає ефективність і надійність всієї системи. У цьому розділі ми розглянемо ключові аспекти, які варто врахувати при виборі датчиків, камер, систем контролю доступу та інших компонентів.

Вибір датчиків руху та відкриття передбачає аналіз датчиків при якому потрібно звертати увагу на кілька важливих характеристик, таких як діапазон дії, кут огляду, точність і чутливість. В залежності від вимог безпеки, можна використовувати різні типи датчиків:

- інфрачервоні датчики (PIR) — для виявлення руху;
- ультразвукові датчики — для вимірювання відстані або додаткового контролю за рухом;

- магнітні датчики — для виявлення відкриття дверей або вікон;

Також важливо враховувати ступінь захисту датчиків:

- IP54, IP55, IP65 — для зовнішнього використання, де датчик повинен бути захищений від пилу та води;

- IP44 — для використання під навісами або в умовах, де не передбачається сильний вплив вологості;

- IP20 — підходить для закритих приміщень з низьким рівнем вологості, але може бути пошкоджений від попадання пилу;

- IP41 — захист від водяних бризок, що дозволяє використовувати датчик в умовах підвищеної вологості.

Також важливо враховувати максимальну потужність, яку можуть використовувати датчики для включення освітлення або інших електричних пристроїв.

Для інфрачервоних датчиків важливо пам'ятати, що вони можуть не працювати через перешкоди, наприклад, звичайне скло не пропускає інфрачервоні хвилі. У таких випадках доцільно використовувати кілька датчиків, розміщених у різних зонах спостереження.

Подальша інформація стосується різних типів камер відеоспостереження.

Оцінка камер відеоспостереження передбачає врахування таких параметрів, як:

- роздільна здатність — для чіткого зображення необхідно вибрати камери з високою роздільною здатністю;
- кут огляду — це важливий параметр для покриття максимальної площі;
- нічне бачення — необхідно для забезпечення зйомки в умовах поганого освітлення;
- вбудована відеоаналітика — можливість автоматичного виявлення руху або осіб;
- зберігання відеоданих — важливо мати можливість запису і зберігання відео для подальшого аналізу.

Вибір системи контролю доступу насамперед повинен враховувати ефективність контролю доступу в системі безпеки. Для цього слід включити такі типи пристроїв:

- RFID зчитувачі — для безконтактної ідентифікації;
- біометричні системи — для ідентифікації осіб за відбитками пальців або за допомогою сканування обличчя;
- електронні замки — для управління доступом до приміщень.

Вибір системи контролю доступу вимагає оцінки таких характеристик:

- надійність — важливо, щоб система працювала без збоїв;
- швидкість роботи — швидке реагування на запити користувачів;
- зручність використання — система повинна бути простою в експлуатації.

Проаналізуємо можливості мікроконтролера Arduino та додаткових модулів.

При виборі компонентів для інтеграції з мікроконтролером Arduino потрібно враховувати:

- сумісність з іншими модулями — наприклад, Ethernet або Wi-Fi shield для з'єднання з мережею;
- можливість підключення додаткових пристроїв — таких як LCD екрани для відображення інформації, реле для управління пристроями тощо;

— розширення функціональних можливостей — використання Bluetooth модулів та інших технологій для покращення ефективності системи.

Проведений аналіз дозволяє вибрати оптимальні компоненти для побудови системи безпеки на основі Arduino. Оцінка характеристик датчиків, камер, систем контролю доступу і сумісності з Arduino допомагає створити надійну та ефективну систему, що відповідає потребам конкретного підприємства.

Розробка системи безпеки на базі Arduino для підприємства

Для побудови системи безпеки на основі Arduino, яка задовольнятиме вимоги підприємства, можна обрати наступні компоненти:

Датчики руху та відкриття:

Датчик руху HC-SR501 (рис. 3.1) — високочутливий датчик з великою дальністю дії для виявлення руху.

Магнітний датчик відкриття дверей або вікон (рис. 3.2) — для фіксації відкриття дверей або вікон.



Рисунок 3.1 — Вигляд датчика руху HC-SR501



Рисунок 3.2 — Вигляд магнітного датчика відкриття дверей МС-38

Камера Arducam 64 Мп з автофокусом (рис. 3.3) — камера з високою роздільною здатністю, здатна до чіткого зображення та автофокусу.

IP-камера Hikvision DS-2CD1321-I (рис. 3.4) — камера для потокової передачі відео в реальному часі, яка забезпечує стабільне відеоспостереження.



Рисунок 3.3 — Вигляд камери Arducam 64 Мп

Електронний замок SEVEN LOCK SL-7737S (рис. 3.5) — замок з підтримкою RFID-карт та кодових замків для контролю доступу.

RFID-зчитувач RC-522 (рис. 3.6) — пристрій для зчитування RFID-карт, що

забезпечує доступ користувачів.



Рисунок 3.4 — Вигляд камера Hikvision



Рисунок 3.5 — Вигляд електронного замка

Arduino Mega (рис. 3.7) — потужний мікроконтролер, який дозволяє

обробляти великі обсяги даних та підключати велику кількість пристроїв для розширених систем [16].

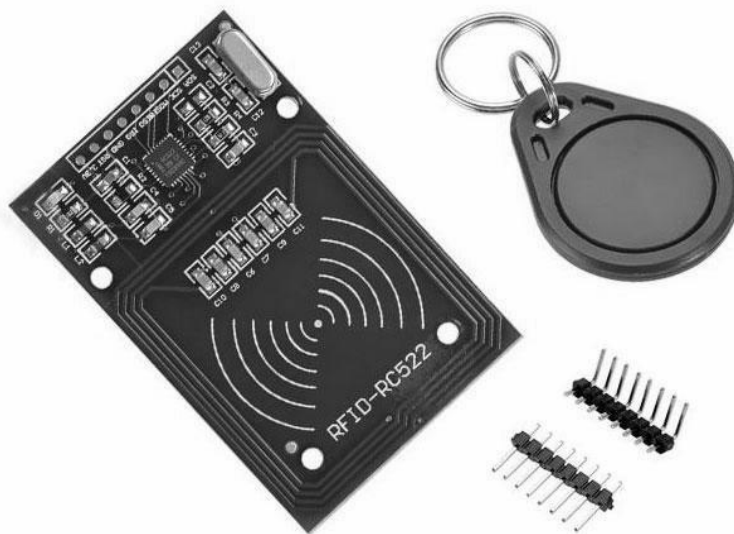


Рисунок 3.6 — Вигляд RFID-зчитувач RC-522

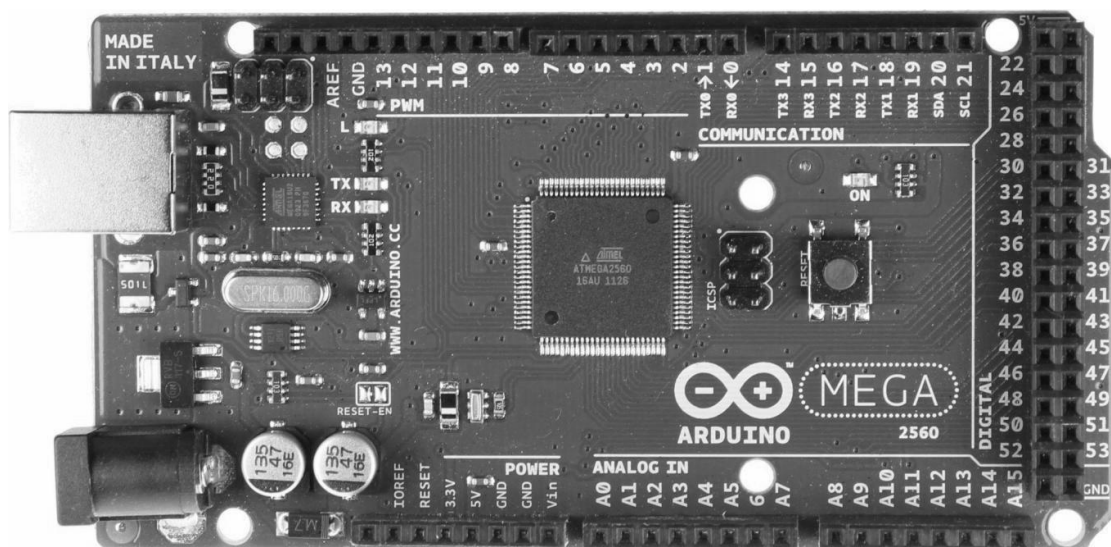


Рисунок 3.7 — Мікроконтролер Arduino Mega

Реле ITV U-Prox Relay AC (рис. 3.8) — для управління замками та іншими електронними пристроями.

LCD-дисплей YM0802A-1-Good Display (рис. 3.9) — дисплей для відображення стану системи та подій у реальному часі.

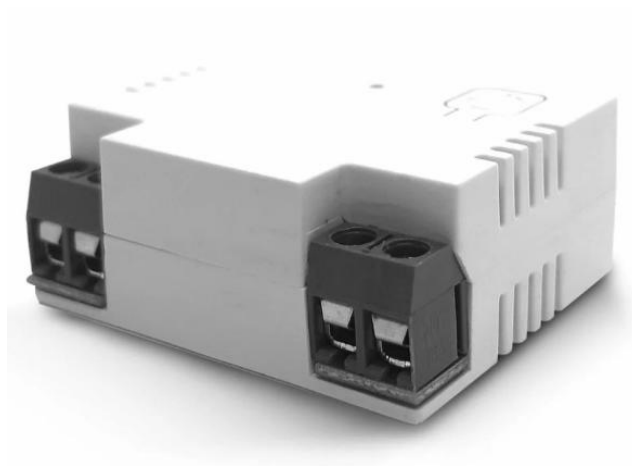


Рисунок 3.8 — Вигляд реле



Рисунок 3.9 — Вигляд LCD-дисплея YM0802A

3.2 Розробка програмної частини

Алгоритм обробки даних із сенсорів [18]:

- задається порогове значення, при якому спрацює датчик;
- визначаються умови для аналізу отриманої інформації з метою виявлення руху або відкриття дверей/вікон.

Алгоритм керування відеоспостереженням:

- встановлюються правила щодо початку та тривалості запису після спрацювання системи;

- реалізується періодичне ввімкнення/вимкнення камери для зменшення енергоспоживання.

Алгоритм перевірки ідентифікаційних даних:

- для авторизації використовуються RFID-картки або кодові комбінації, які зв'язуються з базою даних користувачів;

- після введення ідентифікатора здійснюється перевірка на відповідність збереженим даним.

Алгоритм керування електронними замками:

- залежно від прав доступу користувача замок переводиться у стан «відкрито» або «закрито»;

- перед відкриттям виконується перевірка наявності відповідних дозволів.

Алгоритм виведення інформації на дисплей:

- створюється інтерфейс, який відображає основну інформацію про стан системи та параметри керування;

- можна реалізувати анімації або ефекти переходів для покращення зручності взаємодії з користувачем.

Алгоритм роботи у надзвичайних ситуаціях

- у разі виявлення загрози безпеці активуються всі критичні модулі системи та передається сигнал тривоги;

- система може автоматично надсилати сповіщення охоронним службам або активувати додаткові захисні механізми.

Зазначені алгоритми становлять основу для програмного забезпечення системи безпеки, що працює на платформі Arduino. Їх практична реалізація включає написання коду, тестування та оптимізацію для забезпечення максимальної надійності та стабільної роботи всієї системи.

Програмування мікроконтролерів Arduino можна здійснювати за допомогою середовища Arduino IDE — це офіційне середовище розробки з відкритим вихідним кодом, створене спеціально для роботи з платами Arduino (рис. 3.10).



Рисунок 3.10 — Вигляд головного вікна Arduino IDE

Нижче наведено послідовність дій для написання та завантаження програмного коду на мікроконтролер.

Спочатку відбувається інсталяція Arduino IDE:

- завантаження актуальної версії Arduino IDE з офіційного сайту [19];
- встановлення програми на комп'ютер і відкриття її після завершення інсталяції.

Далі вибирається тип плати, для чого у розділі «Інструменти» вибирається модель мікроконтролера, з якою будете працювати, наприклад Arduino Uno або Arduino Mega.

Після чого пишеться код на мові C або C++ безпосередньо в середовищі Arduino IDE, програма повинна реалізовувати функціональність вашої системи — це може бути зчитування даних із сенсорів, управління пристроями, обробка сигналів, взаємодія з камерами тощо [20].

Наступний крок — завантаження коду на плату:

- з'єднується плата Arduino з комп'ютером за допомогою USB-кабелю;
- встановлюється відповідний COM-порт у меню «Інструменти»;
- натискається кнопка «Завантажити», щоб передати програму на пристрій.

Тестування та від лагодження:

- для моніторингу та налагодження використовується вбудований послідовний монітор Arduino IDE;

- також застосовуються додаткові інструменти відлагодження, щоб виявити і усунути можливі помилки.

Оптимізація та вдосконалення

- ефективність роботи мікроконтролера покращується шляхом оптимізації коду;

- за потреби додаються нові функції для розширення можливостей та підвищення стабільності системи.

Виконавши вказані кроки, зможемо створити і завантажити програму, яка забезпечить коректну роботу системи безпеки на базі Arduino. Її слід періодично оновлювати та перевіряти функціональність програмного забезпечення, щоб гарантувати стабільність і надійність його роботи.

Після вибору компонентів можна приступити до розробки програмного забезпечення для Arduino, яке буде відповідати за взаємодію з датчиками, камерами та системою контролю доступу. Програмний код поділено на дві частини:

- 1) Код для Arduino Mega який відповідає за обробку подій від сенсорів, управління замком, дисплеєм, GSM-модулем і передачу подій у ESP через UART
- 2) Код для ESP8266: отримує події через Serial і виконує HTTPS-запити до хмарного сервісу IFTTT для надсилання push-нотифікацій

Ось основні етапи програмування Arduino:

На початку розробки програмного коду системи безпеки необхідно оголосити всі бібліотеки, визначити логічні назви пінів мікроконтролера та створити об'єкти для роботи з периферійними пристроями.

Ініціалізація: налаштування портів вводу/виводу та підключення всіх компонентів

```
#include <SPI.h>           // Для SPI-підключення
```

```

#include <MFRC522.h> // Бібліотека для роботи з RC522 RFID-зчитувачем
#include <LiquidCrystal.h> // Бібліотека для LCD 16x2
#define PIR_PIN 2 // Датчик руху
#define DOOR_SENSOR 3 // Герконовий сенсор
#define RELAY_PIN 4 // Реле замка
#define BUZZER_PIN 5 // Сирена
#define RST_PIN 6 // RST пін RFID-зчитувача
#define SS_PIN 53 // SS пін RFID
//Об'єкти для роботи з дисплеєм та RFID
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); // LCD
MFRC522 mfrc522(SS_PIN, RST_PIN); // RFID-зчитувач
//Дані для авторизації
const int maxUIDs = 10;
byte validUID[10][4] = {
  {0xDE, 0xAD, 0xBE, 0xEF},
  {0x11, 0x22, 0x33, 0x44}
};
int numUIDs = 2;
//Номер телефону для GSM-повідомлень
const String PHONE_NUMBER = "+3809833333333";
//Змінна для debounce подій
unsigned long lastEventTime = 0;

```

Функція `setup()` є обов'язковою у всіх програмах Arduino. Вона виконується лише один раз при старті програми, тому потрібно виконати всі необхідні ініціалізації.

```

void setup() {
  Serial.begin(9600); // Головний UART для налагодження
  Serial1.begin(9600); // GSM-модуль
  Serial3.begin(9600); // ESP8266
  SPI.begin(); // Ініціалізація SPI-шини
  if (!mfrc522.PCD_PerformSelfTest()) {
    Serial.println("RFID module failed to initialize"); }
  mfrc522.PCD_Init(); // Ініціалізація RFID-зчитувача RC522
  lcd.begin(16, 2); // Ініціалізація LCD-дисплея
  lcd.print("Waiting card..."); // Виведення стартового повідомлення
  pinMode(PIR_PIN, INPUT); // PIR-сенсор як вхід
  pinMode(DOOR_SENSOR, INPUT_PULLUP); // Геркон
  pinMode(RELAY_PIN, OUTPUT); // Реле замка як вихід
  pinMode(BUZZER_PIN, OUTPUT); // Сирена як вихід
  // Початковий стан замка і сирени — вимкнено
  digitalWrite(RELAY_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);}

```

Після одноразового виконання функції `setup()`, мікроконтролер переходить до виконання функції `loop()`. Це головна частина програми, яка виконується

нескінченно — на кожному циклі система аналізує стан сенсорів, зчитувачів та виконує відповідні дії у реальному часі.

```
void loop() {
  handleSMSCommands(); // Обробка SMS-команд
  if (alarmEnabled) {
    handleMotion(); // Реакція на спрацювання PIR-датчика
    handleDoor(); // Реакція на відкриття дверей (геркон) }
    handleRFID(); // Обробка зчитування RFID-карток}
```

Зчитування даних: організовано зчитування з RFID-зчитувача для ідентифікації користувача. У разі успішного зчитування дозволено відкриття електрозамка, в іншому випадку генерується коротка тривога.

```
void handleRFID() {
  if (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) return;
  if (isValidCard(mfrc522.uid.uidByte)) {
    lcd.clear();
    lcd.print("Access granted");
    sendSMS("Access granted");
    sendToESP("access_granted");
    openLock();
  } else {
    lcd.clear();
    lcd.print("Access denied");
    sendSMS("Access denied");
    sendToESP("access_denied");
    triggerAlarm(500); }
  delay(2000); // Коротка пауза після події
  lcd.clear();
  lcd.print("Waiting card...");
  mfrc522.PICC_HaltA(); // Завершення сесії зчитування}
void openLock() {
  unsigned long startTime = millis();
  digitalWrite(RELAY_PIN, HIGH);
  while (millis() - startTime < 3000) { }
  digitalWrite(RELAY_PIN, LOW);}
void triggerAlarm(int duration) {
  unsigned long startTime = millis();
  digitalWrite(BUZZER_PIN, HIGH);
  while (millis() - startTime < duration) { }
  digitalWrite(BUZZER_PIN, LOW);}
```

Функція `handleMotion()` викликається у кожному циклі `loop()` і перевіряє, чи спрацював PIR-датчик руху, підключений до цифрового входу `PIR_PIN`.

```

void handleMotion() {
  if (digitalRead(PIR_PIN) == HIGH && millis() - lastEventTime > 2000) {
    lastEventTime = millis();
    lcd.clear();
    lcd.print("Motion detected");
    sendSMS("Motion detected");
    sendToESP("motion");
    triggerAlarm(1000);
    lcd.clear();
    lcd.print("Waiting card..."); }}

```

Герконовий датчик використовується для фіксації фізичного відкриття дверей чи вікон. У разі спрацювання система негайно повідомляє про це користувача і виконує комплекс захисних дій.

```

void handleDoor() {
  if (digitalRead(DOOR_SENSOR) == HIGH && millis() - lastEventTime > 2000) {
    lastEventTime = millis();
    lcd.clear();
    lcd.print("Door opened!");
    sendSMS("Door opened");
    sendToESP("door_opened");
    triggerAlarm(2000);
    lcd.clear();
    lcd.print("Waiting card..."); }}

```

Для реалізації функціоналу віддалених push-сповіщень через Інтернет, у системі використовується модуль ESP8266.

Надсилення повідомлень: при кожній тригерній події (рух, спроба входу, відкрита двері) GSM-модуль відправляє відповідне SMS на запрограмований номер телефону.

```

void sendSMS(String message) {
  Serial1.println("AT");
  delay(100);
  if (Serial1.find("OK")) {
    Serial1.println("AT+CMGF=1");
    delay(100);
    Serial1.println("AT+CMGS=\"" + 38098333333333 + "\"");
    delay(100);
    Serial1.print(message);
    Serial1.write(26); }}

```

Порівняння UID зчитаної RFID-картки з переліком дозволених

ідентифікаторів. Якщо UID входить до списку, доступ дозволено якщо ні то заборонено.

```
bool isValidCard(byte *uid) {
    if (numUIDs > maxUIDs) numUIDs = maxUIDs; // Захист від переповнення
    for (int i = 0; i < numUIDs; i++) {
        bool match = true;
        for (int j = 0; j < 4; j++) {
            if (uid[j] != validUID[i][j]) {
                match = false;
                break;
            }
        }
        if (match) return true; // Знайдено збіг
    }
    return false; // UID не знайдено у списку
}
```

ESP8266 у складі системи використовується як IoT-модуль для передавання подій, зафіксованих Arduino Mega, у хмару. Він забезпечує повідомлення користувачу через сервіси. На відміну від Arduino, який працює з фізичними датчиками, ESP8266 займається мережевою частиною: підключенням до Wi-Fi, обробкою подій та виконанням HTTPS-запитів.

При отриманні події ESP викликає функцію, яка формує HTTPS GET-запит

```
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
const char* iftttKey = "bDg1FfA7Jx12KlmZ";
const String baseUrl = "https://maker.ifttt.com/trigger/";
void sendEventToIFTTT(String eventName) {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClientSecure client;
        client.setInsecure(); // Пропуск валідації сертифіката
        HTTPClient http;
        String url = baseUrl + eventName + "/with/key/" + iftttKey;
        if (http.begin(client, url)) {
            http.GET();
            http.end();
        }
    }
}
```

Після написання та тестування програмного забезпечення, систему можна встановлювати на об'єкті для ефективного управління безпекою.

3.3 Збірка і запуск тестового зразка системи

Після підбору необхідних компонентів для створення системи безпеки на базі Arduino, наступним кроком є збирання прототипу та перевірка його

функціональності. Нижче наведено детальний опис основних етапів монтажу та налаштування кожного елементу.

Датчики руху та відкривання: два датчики HC-SR501 підключаються до входів мікроконтролера: один вихід передає сигнал (OUT), другий — забезпечує живлення (VCC). OUT підключається до цифрового входу Arduino.

Магнітний сенсор MC-38 підключається до Arduino аналогічним чином. Один контакт іде до GND або VCC (залежно від логіки), інший — до входу контролера.

Відеоспостереження: камеру Arducam (64MP) та IP-камеру Hikvision необхідно підключити до локальної мережі або через відповідні інтерфейси, сумісні з Arduino.

Система доступу: електронний замок SEVEN LOCK підключається через реле ITV U-Prox Relay AC, яке керується Arduino.

Зчитувач RFID RC-522 встановлюється для введення ідентифікаційної інформації, підключаючись до цифрових входів/виходів Arduino.

Arduino Mega: встановлюється мікроконтролер на макетну плату або корпус, підключаються всі елементи системи до відповідних пінів Arduino згідно зі схемою.

Додаткові компоненти: реле ITV U-Prox Relay AC керує електронними механізмами (замки, сирени тощо).

LCD-дисплей (модель YM0802A-1-Good) використовується для відображення поточного стану системи та повідомлень.

Після повного з'єднання всіх модулів (рис. 3.11) слід перейти до етапу тестування. Тест проводиться за допомогою імітації руху перед сенсором, система повинна реагувати на зміну стану (рис. 3.12).

Відкриваються/закриваються двері або вікно з встановленим магнітним сенсором — перевіряється правильність спрацювання.

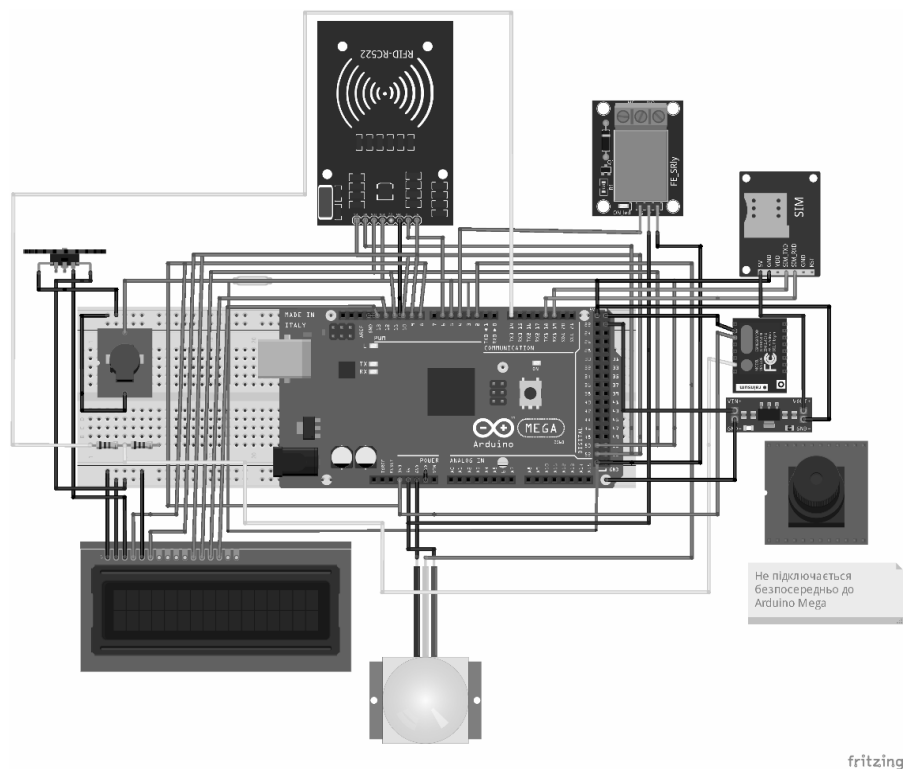


Рисунок 3.11 — Макет системи

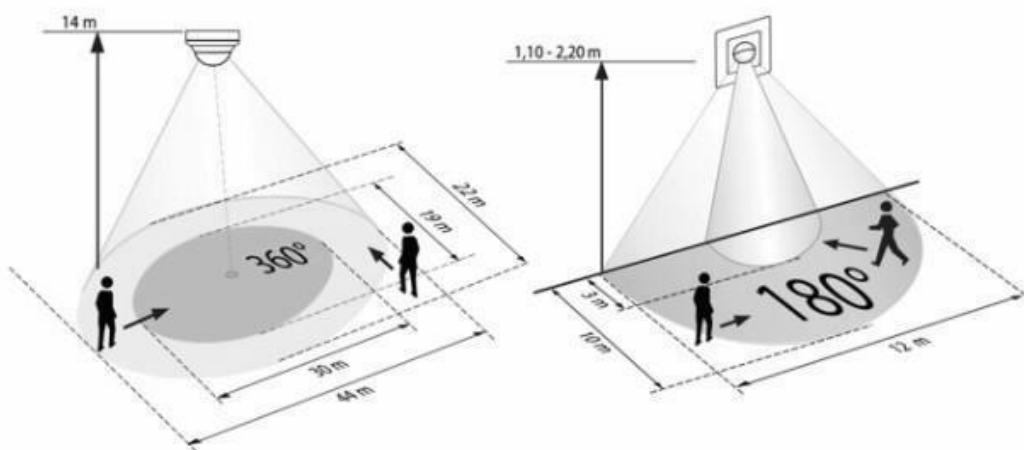


Рисунок 3.12 — Принципова схема роботи датчика руху

При тестуванні камери відеоспостереження записується відео при спрацюванні події, перевіряється, чи правильно функціонує відеозапис. Перевіряється прямий стрім із IP-камери та підключення до мережі (рис. 3.13).



Рисунок 3.13 — Прототип додатку для перегляду відео з камер у режимі реального часу

Що стосується контролю доступу, то за допомогою RFID-зчитувача відбувається перевірка зчитування картки та правильність визначення системою користувача.

Відповідь системи тестується на валідні та невалідні ідентифікатори.

При тестуванні мікроконтролера перевіряється справність цифрових та аналогових входів/виходів Arduino, перевіряється правильність обміну даними між мікроконтролером та всіма підключеними модулями.

Перевірка допоміжних елементів передбачає перевірку роботи реле під час активації замка або інших виконавчих пристроїв, користувач переконується, що інформація на РК-дисплеї відображається коректно та оновлюється в режимі реального часу (рис. 3.14).



Рисунок 3.14 — Зразок виведення інформації на дисплей системи безпеки

У процесі тестування звертається увага на точність реакції кожного модуля, синхронізацію між ними та можливі відхилення в роботі. У разі виявлення помилок необхідно внести корективи до схеми підключень або програмного забезпечення, щоб забезпечити надійну та стабільну роботу всієї системи.

3.4 Тестування прототипу

Тестування системи безпеки здійснюється за допомогою стандартних інструментів середовища Arduino IDE. Для виведення діагностичної інформації у серійному режимі були створені спеціалізовані скетчі, які дозволяють відстежити роботу кожного компонента.

Спочатку відбувається перевірка реакції на рух (рис. 3.15):

```
14:11:48.601 -> Alarm is activated
14:12:03.321 -> Motion detected
14:12:13.340 -> AT+CMGS="+380980000000"
14:12:14.372 ->
14:12:14.372 -> >
14:12:14.372 -> Motion detected! Alarm had been activated!
14:12:14.419 -> □
```

Motion detected! Alarm had been activated!

Рисунок 3.15 — Вивід результатів перевірки роботи датчика руху

На зображенні видно, що система фіксує рух і реагує на нього протягом 10 секунд. Це підтверджує стабільну роботу модуля виявлення руху в межах заданого часового інтервалу. Нижче приведено тест команди «Turn off» (рис. 3.16):

```

14:14:47.398 -> +CMTI: "SM",1
14:14:47.398 -> AT+CMGR=1
14:14:48.519 ->
14:14:48.519 -> +CMGR: "REC UNREAD","+38098[REDACTED]", "", "21/05/25,14:14:38+12"
14:14:48.613 -> Turn off
14:14:48.613 ->
14:14:48.613 -> OK
14:14:48.613 ->
14:14:48.613 -> Phone: +38098[REDACTED]
14:14:48.660 -> Message: Turn off
14:14:48.660 -> Alarm deactivation
14:14:48.660 -> Alarm is deactivated
14:14:48.707 -> AT+CMGDA="DEL ALL"
14:14:49.735 ->
14:14:49.735 -> OK

```

Рисунок 3.16 — Результат обробки команди Turn off

Система отримує повідомлення від користувача (власника), після чого коректно виконує команду вимкнення сигналізації. Це свідчить про правильну обробку вхідних SMS-команд або сигналів.

Тест команди «Turn on» (рис. 3.17):

```

14:15:29.724 -> +CMTI: "SM",1
14:15:29.771 -> AT+CMGR=1
14:15:30.852 ->
14:15:30.852 -> +CMGR: "REC UNREAD","+38098[REDACTED]", "", "21/05/25,14:15:20+12"
14:15:30.944 -> Turn on
14:15:30.944 ->
14:15:30.944 -> OK
14:15:30.944 ->
14:15:30.944 -> Phone: +38098[REDACTED]
14:15:30.991 -> Message: Turn on
14:15:30.991 -> Alarm activation
14:15:30.991 -> Alarm is activated
14:15:31.037 -> AT+CMGDA="DEL ALL"
14:15:32.060 ->
14:15:32.060 -> OK

```

Рисунок 3.17 — Результат обробки команди Turn on

Як видно з результатів, система отримала команду ввімкнення та активувала сигналізацію. Це свідчить про надійну роботу алгоритмів дистанційного керування.

Тестування команди «Status» (рис. 3.18):

```

14:20:21.436 -> +CMTI: "SM",2
14:20:21.436 -> AT+CMGR=2
14:20:22.557 ->
14:20:22.557 -> +CMGR: "REC UNREAD","+38098[REDACTED]", "", "21/05/25,14:20:12+12"
14:20:22.602 -> Status
14:20:22.602 ->
14:20:22.602 -> OK
14:20:22.649 ->
14:20:22.649 -> Phone: +38098[REDACTED]
14:20:22.649 -> Message: Status
14:20:22.649 -> AT+CMGS="+38098[REDACTED]"
14:20:23.672 ->
14:20:23.672 -> >
14:20:23.672 -> Signalization is inactive
14:20:23.719 -> □
14:20:24.757 -> Signalization is inactive
14:20:24.804 -> >
14:20:24.804 -> >
14:20:24.804 -> AT+CMGDA="DEL ALL"
14:20:27.515 ->
14:20:27.515 -> +CMGS: 47
14:20:27.515 ->
14:20:27.515 -> OK

```

Рисунок 3.18 — Результат виконання запиту Status

На основі запиту користувача система надає відповідь про свій поточний стан. Це демонструє стабільну роботу механізму моніторингу.

Загальний аналіз результатів тестування підтвердив, що система функціонує стабільно та відповідає вимогам безпеки. Основні результати:

- реакція на зміну умов середовища: датчики руху та відкривання показали швидку та точну реакцію на події, затримка між активацією сенсорів і відповіддю системи була в межах допустимих норм, що забезпечує високу ефективність;

- керування сигналізацією: система чітко розпізнає команди користувача, як-от увімкнення чи вимкнення сигналізації; робота системи не порушується навіть при кількох послідовних командах, що свідчить про її надійність;

- інформація про стан: відповіді на команду «Status» були точними та своєчасними, це дозволяє користувачеві в режимі реального часу отримувати актуальні дані про роботу всієї системи.

Таким чином, проведене тестування засвідчує успішну інтеграцію всіх складових частин системи безпеки. Усі функціональні модулі коректно реагують на події, а логіка управління підтверджує свою ефективність під час практичного застосування.

3.5 Порівняння з комерційними системами безпеки

Порівняльний аналіз систем безпеки на основі платформи Arduino та традиційних або альтернативних рішень дозволяє краще оцінити переваги й недоліки кожного варіанта. Такий підхід важливий для прийняття обґрунтованого рішення під час вибору оптимальної технології для конкретних завдань.

Arduino порівняно з комерційними (пропрієтарними) системами безпеки: Arduino зазвичай є економічно вигіднішим рішенням, оскільки не потребує витрат на ліцензування програмного забезпечення або придбання дорогого обладнання. Платформа також вирізняється високою гнучкістю — завдяки відкритому коду користувач може адаптувати функціональність системи під конкретні потреби. Крім того, активна спільнота користувачів забезпечує доступ до великої кількості прикладів, рішень і порад, що спрощує розробку й усунення несправностей.

Arduino порівняно з іншими відкритими системами безпеки: Arduino вважається зручнішим у використанні, особливо для новачків або тих, хто не має значного досвіду в електроніці. Його інтерфейс та середовище розробки інтуїтивно зрозумілі, що дозволяє швидко розпочати роботу. Крім того, система легко інтегрується з іншими модулями екосистеми Arduino, що дає змогу реалізовувати комплексні рішення. Однак варто зазначити, що інші відкриті платформи, як-от Raspberry Pi або ESP32, можуть мати вищу продуктивність і краще підходити для масштабованих або ресурсомістких проєктів.

Загалом вибір між Arduino, комерційними рішеннями або іншими відкритими платформами залежить від специфіки завдань, технічних вимог, бюджету та рівня підготовки розробника. Arduino є оптимальним вибором для побудови недорогих, адаптивних та ефективних систем безпеки для побутового або локального використання.

3.6 Підсумкова оцінка ефективності реалізованої системи

Під час роботи було розглянуто всі аспекти створення системи безпеки — від вибору компонентів і обладнання до розробки програмного забезпечення та

тестування прототипу. Основні етапи реалізації проєкту:

Вибір комплектуючих та обладнання включає аналіз доступних датчиків, камер та інших модулів з урахуванням специфіки вимог системи безпеки. Обрані компоненти, зокрема датчик руху HC-SR501, магнітний датчик відкриття MC-38, камери Arducam і Hikvision, електронний замок SEVEN LOCK та зчитувач RFID, забезпечують високу чутливість, швидку реакцію та можливість подальшого розширення функціоналу системи.

Програмування мікроконтролера Arduino здійснювалося з використанням мови C/C++ з використанням Arduino IDE. Код забезпечує зчитування даних із сенсорів, керування відеоспостереженням та доступом. У логіку роботи входить обробка сигналів датчиків, активація камер під час виявлення руху, ідентифікація користувачів за допомогою RFID-карток та управління електронними замками.

Складання та тестування прототипу полягало в тому, що усі компоненти було фізично зібрано та підключено до Arduino. Прототип системи пройшов комплексне тестування, яке охоплювало перевірку роботи сенсорів руху та відкривання, камер та системи контролю доступу. Тестові результати показали стабільну роботу всіх елементів: система своєчасно реагує на рух, відкриття дверей чи вікон, ідентифікує користувачів і виконує управління доступом.

Аналіз результатів підтвердив відповідність системи встановленим вимогам безпеки. Вона забезпечує ефективний моніторинг, контроль доступу та має високий потенціал для подальшого розширення. Завдяки гнучкій архітектурі можлива інтеграція нових функцій і модулів, що робить систему придатною для використання в середовищі Інтернету речей (IoT).

У порівнянні з традиційними комерційними рішеннями, система на базі Arduino вирізняється доступністю, простотою розгортання, легкістю у програмуванні та можливістю налаштування під конкретні потреби. Це робить її привабливим варіантом для підприємств, що потребують персоналізованих рішень безпеки.

ВИСНОВКИ

У ході виконання роботи було розроблено мікропроцесорну систему безпеки для підприємства з використанням апаратної платформи Arduino. Результати дослідження підтвердили, що Arduino є ефективною основою для створення доступних, гнучких та надійних охоронних рішень, що відповідають сучасним вимогам до автоматизованих систем безпеки.

В процесі реалізації системи було проведено відбір і налаштування апаратних компонентів: датчиків руху, контактів відкриття, відеокамер, RFID-зчитувача та електронного замка. Обрана апаратна база забезпечує необхідний рівень чутливості, швидку реакцію на події та підтримує можливість подальшого розширення функціональності.

Програмне забезпечення, розроблене на мові C/C++, забезпечує взаємодію між мікроконтролером та компонентами системи. Алгоритми обробки подій дозволяють ефективно реагувати на загрози, зчитувати дані з датчиків, керувати доступом та активувати камери у відповідь на рух.

Тестування прототипу в умовах, наближених до реального використання, засвідчило стабільність роботи системи, своєчасність спрацювання на вхідні сигнали та надійність комунікації з користувачем. Це підтверджує доцільність застосування Arduino для створення локальних систем охорони на підприємствах.

Порівняно з традиційними комерційними рішеннями, система на основі Arduino демонструє такі переваги, як нижча вартість, відкритість архітектури, легкість у модернізації та інтеграції з іншими IoT-рішеннями.

Отже, розроблена мікропроцесорна система безпеки є перспективною як для використання на невеликих об'єктах, так і як основа для подальших досліджень у сфері автоматизації безпеки, енергозбереження та цифровізації підприємств.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Проценко О.Р. Мікропроцесорна система безпеки підприємства на платформі Arduino // Матеріали всеукраїнської науково-практичної Інтернет-конференції студентів та молодих науковців — Режим доступу :
<https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/25451>
2. Lake D., Rayes A., Morrow M. The Internet of Things // The Internet Protocol Journal. 2012. Vol. 15, No. 3.
3. ITU-T. Common Requirements and Capabilities of a Gateway for Internet of Things Applications. Recommendation Y.2067. June 2014.
4. Cisco Systems. The Internet of Things Reference Model. White Paper. 2014. URL: <http://www.iotwf.com/>
5. Frahim J., et al. Securing the Internet of Things: A Proposed Framework. Cisco White Paper. March 2015.
6. ITU-T. Overview of the Internet of Things. Recommendation Y.2060. June 2012.
7. Погребенник В. Д., Політило Р. В. Ультразвукові сенсори системи охоронної сигналізації // Вісник НТУУ “КПІ”. Серія «Приладобудування». 2008. Вип. 36. С. 68–76.
8. Кугір А. В. Автоматизована система охоронної сигналізації для промислового підприємства. 2021. С. 75–76.
9. Çavaş M., Ahmad M. B. A review advancement of security alarm system using Internet of Things (IoT) // International Journal of New Computer Architectures and their Applications (IJNCAA). 2019. Vol. 9, No. 2. P. 38–49.
10. Михальчук Д. О., Яворська О. М. Аналіз ринку систем охоронної сигналізації // Матеріали 75-ї науково-технічної конференції професорсько-викладацького складу, науковців, аспірантів та студентів. 2020. С. 49–50.
11. Ahmad M. B., Abdullahi A. A., Muhammad A. S., Saleh Y. B., Usman U. B. The Various Types of Sensors Used in the Security Alarm System // International Journal of New Computer Architectures and their Applications (IJNCAA). 2019. Vol. 9, No. 2.

Р. 50–59.

12. Розумний будинок. URL: https://www.smarthouse.ua/ua/umnyj_dom.html (дата звернення: 08.03.2025)

13. Афзель С. С., Березанська М. О. Огляд сучасного стану та перспективи розвитку датчиків руху // Ефективність інженерних рішень у приладобудуванні: матеріали доповідей XIV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених. 2018. С. 16.

14. Крошкін С. С., Півторак Д. А. Порівняльний аналіз датчиків руху, що застосовуються в охоронних системах // Нові напрямки розвитку приладобудування: матеріали 12-ї Міжнародної науково-технічної конференції молодих вчених та студентів. 2019. С. 128–133.

15. Arduino Nano. URL: <http://arduino-nano.ua> (дата звернення: 08.03.2025).

16. Arduino Mega. URL: <http://arduino-mega.ua> (дата звернення: 08.03.2025).

17. Іванов А. О. Теорія автоматичного керування: підручник. Дніпро: Національний гірничий університет, 2014. 250 с.

18. Енциклопедія кібернетики: в 2 т. — К.: Головна редакція УРЕ, 2016.

19. Офіційний сайт Arduino. URL: <http://www.arduino.cc> (дата звернення: 08.03.2025).

20. Language Reference URL: <https://docs.arduino.cc/language-reference/> (дата звернення: 08.03.2025).

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____ проф., д.т.н. О.Д. Азаров

«__» _____ 2025р.

ТЕХНІЧНЕ ЗАВДАННЯ

Мікропроцесорна система безпеки підприємства на платформі Arduino

08-54.БКР.014.00.000 ПЗ

Керівник роботи: phd., ст.викл. каф. ОТ

_____ Обертюх М.Р.

Виконав: студент гр. 1КІ-216

_____ Проценко О.Р.

Вінниця 2025

1 Підстава для виконання кваліфікаційної роботи (БКР)

— Тема бакалаврської кваліфікаційної роботи «Мікропроцесорна система безпеки підприємства на платформі Arduino» актуальна у зв'язку з тим, що на сьогодні технології Інтернету речей (IoT) відіграють важливу роль у трансформації багатьох сфер діяльності, включно з промисловістю, побутом і, зокрема, безпекою. Сучасні виклики, пов'язані зі зростанням загроз для бізнесу, потребують нових підходів до захисту об'єктів. Одним із ефективних рішень є впровадження IoT-платформ для створення інтелектуальних систем безпеки.

проаналізувати ефективність використання сенсорів, модулів та інших компонентів Arduino у комплексній системі контролю доступу та моніторингу.

— наказ про затвердження теми бакалаврської кваліфікаційної роботи.

2 Мета і призначення БКР

Метою роботи є теоретичне обґрунтування та практична реалізація концепції створення системи безпеки підприємства на основі технологій IoT з використанням Arduino. Для досягнення поставленої мети необхідно виконати наступні завдання:

— призначення розробки — виконання бакалаврської кваліфікаційної роботи із подальшою підтримкою та розвитком.

3 Джерела розробки

1. Ahmad M. B., Abdullahi A. A., Muhammad A. S., Saleh Y. B., Usman U. B. The Various Types of Sensors Used in the Security Alarm System // International Journal of New Computer Architectures and their Applications (IJNCAA). 2019. Vol. 9, No. 2. P. 50–59.

2. Arduino Nano. URL: <http://arduino-nano.ua> (дата звернення: 08.03.2025).

3. Arduino Mega. URL: <http://arduino-mega.ua> (дата звернення: 08.03.2025).

4 Технічні вимоги до виконання БКР

— наявність середовища Arduino IDE;

— наявність комплектуючих та технічного оснащення.

5 Етапи роботи та очікуванні результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи БКР

№ з/п	Назва етапів дипломної роботи	Строк виконання	Примітка
1	Постановка задачі роботи	21.03.25	
2	Аналіз концепції IoT	21.03.25—29.03.25	
3	Застосування концепції IoT у системах безпеки	21.03.25—29.03.25	
4	Комплектуючі та технічне оснащення	30.03.25—11.04.25	
5	Програмування Arduino	12.04.25—26.04.25	
6	Тестування прототипу	27.04.25—07.05.25	
7	Оформлення пояснювальної записки та ілюстративного матеріалу	08.05.25	
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	14.05.25	

6 Матеріали, що подаються до захисту БКР

Пояснювальна записка, графічні та ілюстративні матеріали, протокол попереднього захисту роботи на кафедрі, відзив наукового керівника, рецензія опонента, анотації до українською та іноземною мовами, нормоконтроль про відповідність оформлення діючим вимогам.

7 Порядок контролю виконання та захисту БКР

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

8 Вимоги до оформлювання та порядок виконання БКР

8.1 При оформлювання БКР 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 бакалаврських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія» (освітня програма «Комп'ютерна інженерія»). Кафедра обчислювальної техніки ВНТУ 2022;

— документами на які посилаються у вище вказаних.

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21»

Проценко О. Р.
(прізвище, ініціали)

ДОДАТОК В

Графічна частина

Мікропроцесорна система безпеки підприємств на базі Arduino

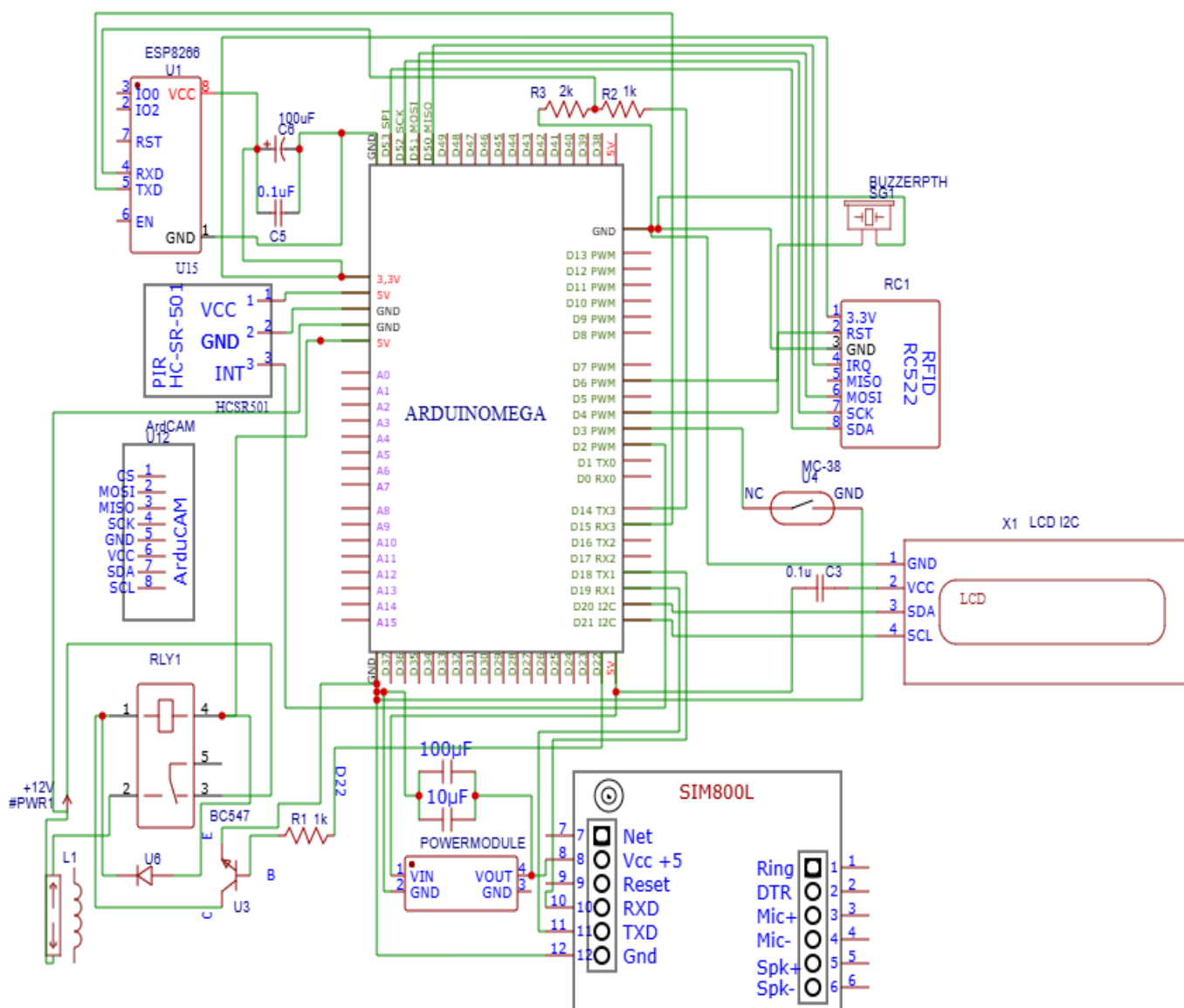


Рисунок В.1 — Схема підключення апаратних компонентів

ДОДАТОК Г

Ілюстративна частина

Мікропроцесорна система безпеки підприємств на базі Arduino

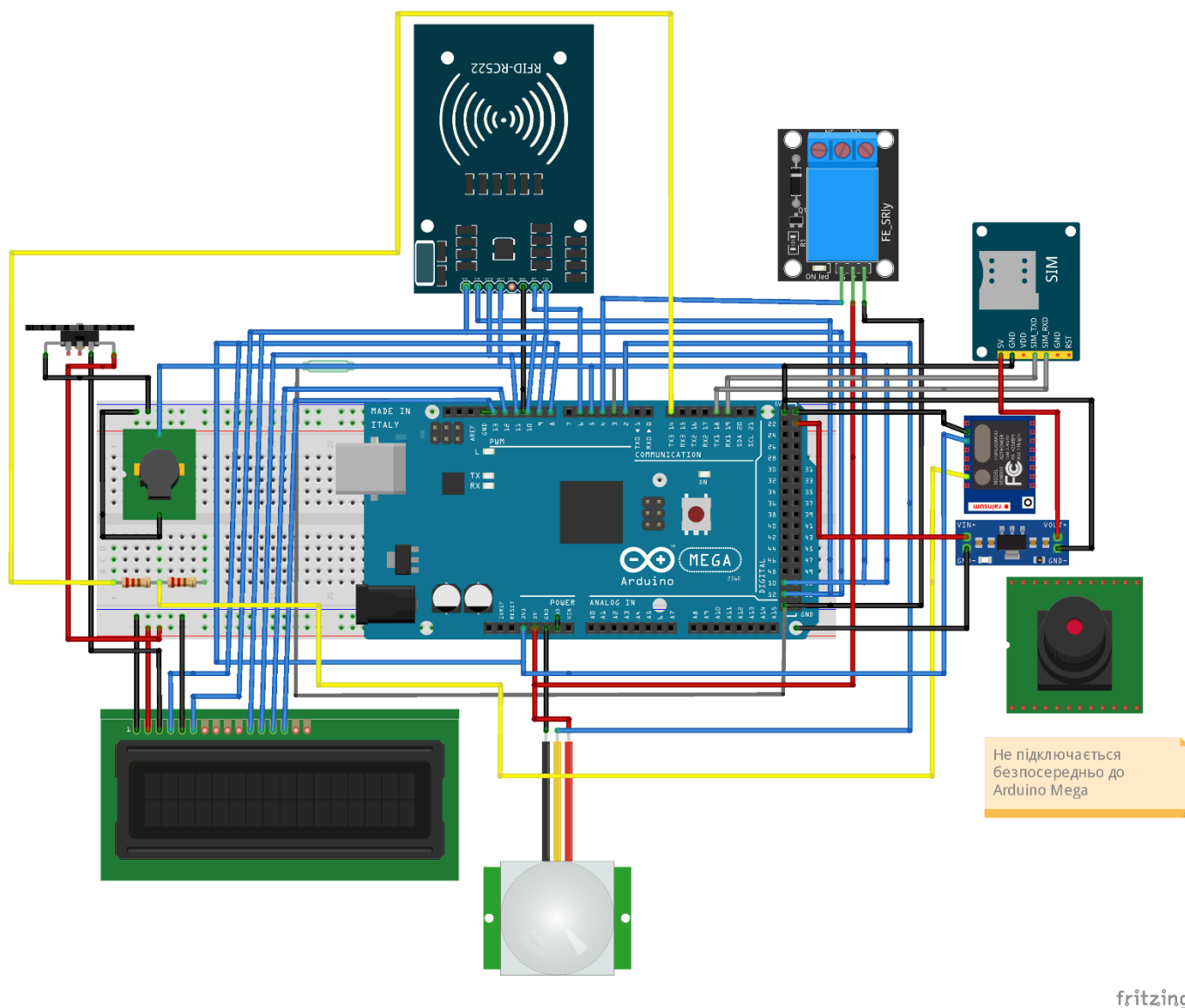


Рисунок Г.1 — Макет системи

ДОДАТОК Д

Лістинг програмного забезпечення

```
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Arducam.h> // Додана бібліотека для Arducam
#define PIR_PIN      2
#define DOOR_SENSOR  3
#define RELAY_PIN    4
#define BUZZER_PIN   5
#define RST_PIN      6
#define SS_PIN       53
#define ARDUCAM_SS   50 // Вибір піна SS для Arducam
#define ARDUCAM_RST   6 // Спільний RST з RFID для спрощення
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
MFRC522 mfrc522(SS_PIN, RST_PIN);
Arducam arducam(ARDUCAM_SS, ARDUCAM_RST); // Ініціалізація Arducam
const int maxUIDs = 10;
byte validUID[10][4] = {
    {0xDE, 0xAD, 0xBE, 0xEF},
    {0x11, 0x22, 0x33, 0x44}};
int numUIDs = 2;
const String PHONE_NUMBER = "+380983333333";
unsigned long lastEventTime = 0;
unsigned long lastSMSTime = 0;
bool alarmEnabled = true;
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600); // GSM
    Serial3.begin(9600); // ESP8266
    SPI.begin();
    if (!mfrc522.PCD_PerformSelfTest()) {
        Serial.println("RFID module failed to initialize"); }
    mfrc522.PCD_Init();
    arducam.begin(); // Ініціалізація Arducam
    lcd.begin(16, 2);
    lcd.print("Waiting card...");
    pinMode(PIR_PIN, INPUT);
    pinMode(DOOR_SENSOR, INPUT_PULLUP);
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    digitalWrite(RELAY_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);}
void loop() {
    handleSMSCommands();
    if (alarmEnabled) {
        handleMotion();
        handleDoor(); }
    handleRFID();
    handleArducam(); // Додана обробка камери}
void sendSMS(String message) {
```

```

Serial1.println("AT");
delay(100);
if (Serial1.find("OK")) {
    Serial1.println("AT+CMGF=1");
    delay(100);
    Serial1.println("AT+CMGS=\"\" + PHONE_NUMBER + \"\");
    delay(100);
    Serial1.print(message);
    Serial1.write(26); }
void sendToESP(String event) {
    Serial3.println(event);}
void openLock() {
    unsigned long startTime = millis();
    digitalWrite(RELAY_PIN, HIGH);
    while (millis() - startTime < 3000) {}
    digitalWrite(RELAY_PIN, LOW);}
void triggerAlarm(int duration) {
    unsigned long startTime = millis();
    digitalWrite(BUZZER_PIN, HIGH);
    while (millis() - startTime < duration) {}
    digitalWrite(BUZZER_PIN, LOW);}
bool isValidCard(byte *uid) {
    if (numUIDs > maxUIDs) numUIDs = maxUIDs;
    for (int i = 0; i < numUIDs; i++) {
        bool match = true;
        for (int j = 0; j < 4; j++) {
            if (uid[j] != validUID[i][j]) {
                match = false;
                break;    }    }
        if (match) return true;    }
    return false;}
void handleRFID() {
    if (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) return;
    if (isValidCard(mfrc522.uid.uidByte)) {
        lcd.clear(); lcd.print("Access granted");
        sendSMS("Access granted");
        sendToESP("access_granted");
        openLock();
        captureArducam(); // Зйомка при доступі
    } else {
        lcd.clear(); lcd.print("Access denied");
        sendSMS("Access denied");
        sendToESP("access_denied");
        triggerAlarm(500);    }
    delay(2000);
    lcd.clear(); lcd.print("Waiting card...");
    mfrc522.PICC_HaltA();}
void handleMotion() {
    if (digitalRead(PIR_PIN) == HIGH && millis() - lastEventTime > 2000) {
        lastEventTime = millis();
        lcd.clear(); lcd.print("Motion detected");
        sendSMS("Motion detected");
    }
}

```

```

    sendToESP("motion");
    triggerAlarm(1000);
    captureArducam(); // Зйомка при русі
    lcd.clear(); lcd.print("Waiting card..."); }}
void handleDoor() {
    if (digitalRead(DOOR_SENSOR) == HIGH && millis() - lastEventTime > 2000) {
        lastEventTime = millis();
        lcd.clear(); lcd.print("Door opened!");
        sendSMS("Door opened");
        sendToESP("door_opened");
        triggerAlarm(2000);
        captureArducam(); // Зйомка при відкритті дверей
        lcd.clear(); lcd.print("Waiting card..."); }}
void handleSMSCommands() {
    // Логіка обробки SMS (залишається без змін)}
void captureArducam() {
    arducam.takePicture(); // Знімок із камери
    lcd.clear(); lcd.print("Photo captured");
    delay(1000); // Коротка затримка для відображення}

```