

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
(повна назва)

Кафедра _____ Автоматизації проектування обчислювальної техніки _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший(бакалаврський) _____
Система для управління освітленням, музикою та дверима в _____
«Розумному будинку» на базі Arduino _____
(тема)

Виконав:
здобувач _____ 4 _____ року навчання,
групи _____ КІУКІ-21-9 _____
Ломако В. О. _____
(власне ім'я, прізвище)

Спеціальність 123 – Комп'ютерна інженерія _____
(код і повна назва спеціальності)
Тип програми _____ освітньо-професійна _____
Освітня програма Комп'ютерна інженерія _____
(повна назва освітньої програми)

Керівник _____ ас. Кулак Г. К. _____
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____ _____
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ломако Владлену Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Система для управління освітленням, музикою та дверима в
«Розумному будинку» на базі Arduino

затверджена наказом університету від 21 05 2025 р. № 403 Ст 1

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 06 2025 р.

3. Вихідні дані до роботи Arduino,

Плата Arduino Nano, RFID-модуль RC522

Дисплей LCD 1602 з модулем I2C, Джойстик KY-023

Світлодіодна стрічка, Активний зумер

Середовище розробки Arduino IDE 2.3.2.

Мова програмування C++

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз та огляд існуючих систем

Постановка задачі

Розробка структурної схеми пристрою

Розробка функціональної схеми програми

Розробка макету

Тестування пристрою

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
13 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Видача теми проекту, узгодження і затвердження теми	06.05.2025 – 09.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	09.05.2025 – 14.05.2025	
3	Розробка структурної схеми пристрою, вибір апаратної платформи	14.05.2025 – 16.05.2025	
4	Розробка функціональної схеми пристрою	16.05.2025 – 17.05.2025	
5	Розробка програмних модулів. Проведення тестування	17.05.2025 – 23.05.2025	
6	Оформлення пояснювальної записки	23.05.2025 – 25.05.2025	
7	Перевірка виконаного проекту керівником, допуск до захисту	25.05.2025 – 14.06.2025	

Дата видачі завдання 06 05 2025 р.

Здобувач _____



(підпис)

Керівник роботи _____



(підпис)

ас. Кулак Г. К.

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 72 сторінки, 16 рисунків, 2 додаток, 12 джерел посилання.

ARDUINO NANO, СИСТЕМА «РОЗУМНИЙ БУДИНОК»,
УПРАВЛІННЯ ОСВІТЛЕННЯМ, УПРАВЛІННЯ МУЗИКОЮ, УПРАВЛІННЯ
ДВЕРИМА, МІКРОКОНТРОЛЕР, АВТОМАТИЗАЦІЯ ЖИТЛА

Метою даної роботи є розробка та реалізація багатфункціональної системи для управління освітленням, музичним супроводом та доступом (дверима) в рамках концепції «Розумного будинку» на базі платформи Arduino.

Проект включатиме розробку схемотехнічних рішень для інтеграції освітлювальних приладів, аудіосистеми та механізмів керування дверима, програмування мікроконтролера Arduino для обробки команд користувача та стану підключених пристроїв, а також інтеграцію відповідних модулів для забезпечення автоматизованого керування. Результатом буде працюючий прототип системи «Розумний будинок», який дозволить користувачеві ефективно та зручно керувати освітленням, відтворенням музики та доступом до приміщення, підвищуючи рівень комфорту, безпеки та енергоефективності житла.

ABSTRACT

The explanatory note of the qualification work contains: 72 pages, 16 figures, 2 appendices, 12 sources.

ARDUINO NANO, «SMART HOME» SYSTEM, LIGHTING CONTROL, MUSIC CONTROL, DOOR CONTROL, MICROCONTROLLER, HOME AUTOMATION

The aim of this work is the development and implementation of a multifunctional system for controlling lighting, music playback, and access (doors) within the «Smart Home» concept based on the Arduino platform.

The project will include the development of schematic solutions for integrating lighting fixtures, an audio system, and door control mechanisms, programming the Arduino microcontroller to process user commands and the status of connected devices, as well as the integration of relevant modules to ensure automated control. The result will be a working prototype of a «Smart Home» system that will allow the user to efficiently and conveniently control lighting, music playback, and access to the premises, thereby increasing the level of comfort, security, and energy efficiency of the home.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	11
1.1 Загальні поняття	11
1.2 Різновиди мікроконтролерів	13
1.3 Мікроконтролери сімейства AVR	16
1.4 ARM-сумісні мікроконтролери	18
1.5 Мікроконтролер ATmega328P	20
1.6 Мета та постановка завдання	22
2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ПРОЄКТУ	24
2.1 Вибір мови програмування та середовища розробки	24
2.1.1 Arduino IDE для Arduino	24
2.1.2 Мультиплатформенність PlatformIO	25
2.1.3 Visual Studio Code з розширенням для Arduino	26
2.1.4 Обґрунтування програмно-технологічних рішень проєкту ...	27
2.2 Призначення та базові функції проєкту	27
3 СПЕЦИФІКАЦІЯ АПАРАТНИХ КОМПОНЕНТІВ СИСТЕМИ	29
3.1 Технологічна база мікроелектромеханічних систем	29
3.2 LCD-монітор	30
3.3 Сервомотор	31
3.4 Активний зумер	32
3.5 RFID-модуль	33
3.6 Модуль керування двоосьовий ХУ джойстик KY-023.....	34
4 ПРОЄКТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ	35
4.1 Проєктування системи «Розумний будинок»	35
4.2 Тестування розроблених функцій системи «Розумний будинок»	40

ВИСНОВКИ	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47
ДОДАТОК А	48
ДОДАТОК Б	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ARM (Advanced RISC Machines) – Архітектура 32-бітних та 64-бітних процесорів, що широко використовується у мобільних пристроях та вбудованих системах завдяки енергоефективності та продуктивності.

AVR (Alf-Egil Bogen Vegard Wollan RISC) – Сімейство 8-бітних мікроконтролерів, розроблених Atmel (тепер Microchip), популярних завдяки платформі Arduino.

CAN (Controller Area Network) – Інтерфейс для обміну даними між мікроконтролерами, особливо у транспортних системах.

CISC (Complex Instruction Set Computer) – Архітектура процесора з розширеним набором інструкцій.

DSP-мікроконтролери (Digital Signal Processing) – Мікроконтролери з підтримкою обробки цифрових сигналів.

EEPROM (Electrically Erasable Programmable Read-Only Memory) – Енергонезалежна пам'ять, яку можна електрично перезаписувати.

ESP (Espressif Systems) – Сімейство мікроконтролерів з підтримкою Wi-Fi/Bluetooth, зокрема ESP8266, ESP32.

GPIO (General-Purpose Input/Output) – Цифрові контакти мікроконтролера, які можна програмно налаштувати на вхід або вихід сигналу.

I2C (Inter-Integrated Circuit) – Послідовна шина для зв'язку між інтегральними мікросхемами, що використовує дві лінії.

IoT (Internet of Things) – Концепція з'єднання фізичних пристроїв через інтернет для автоматизації й управління.

PIC (Peripheral Interface Controller) – Сімейство мікроконтролерів від Microchip Technology.

PWM (Pulse Width Modulation) – Широтно-імпульсна модуляція, метод керування потужністю шляхом зміни тривалості імпульсів.

RISC (Reduced Instruction Set Computer) – Архітектура процесора, що використовує простий набір команд, які виконуються швидко.

RTOS (Real-Time Operating System) – Операційна система реального часу, що забезпечує передбачувану реакцію на події.

SPI (Serial Peripheral Interface) – Інтерфейс для синхронного обміну даними між мікроконтролером та периферійними пристроями.

SRAM (Static Random-Access Memory) – Тип енергонезалежної оперативної пам'яті з високою швидкістю доступу.

UART (Universal Asynchronous Receiver/Transmitter) – Інтерфейс для послідовного асинхронного обміну даними між пристроями.

АЛП (Арифметико-логічний пристрій) – Блок процесора, який виконує арифметичні та логічні операції.

АЦП (Аналого-цифровий перетворювач) – Пристрій, що перетворює аналоговий сигнал у цифровий формат для обробки мікроконтролером.

МЕМС (Мікроелектромеханічні системи) – Технологія створення мініатюрних пристроїв, що поєднують механічні та електричні компоненти на одному кристалі.

ОЗП (Оперативний запам'ятовувальний пристрій) – Пам'ять для тимчасового зберігання даних під час роботи пристрою.

ПЗП (Постійний запам'ятовувальний пристрій) – Пам'ять для зберігання даних, яка не стирається після вимкнення живлення.

ЦАП (Цифро-аналоговий перетворювач) – Пристрій, що перетворює цифрові сигнали в аналогові.

ЦПП (Центральний процесорний пристрій) – Основний обчислювальний блок комп'ютера або мікроконтролера.

ШІМ (Широтно-імпульсна модуляція) – Метод керування потужністю шляхом зміни тривалості імпульсів при постійній частоті.

ВСТУП

Сучасний світ характеризується неперервним технологічним прогресом, який проникає в усі сфери людського життя, трансформуючи наше повсякдення, роботу та відпочинок. Одним із найяскравіших проявів цієї трансформації є концепція «Розумного будинку» (Smart Home), яка перетворює традиційне житло на інтерактивне, автоматизоване та високоефективне середовище. Прагнення до підвищеного комфорту, безпеки, енергоефективності та персоналізації житлового простору стимулює стрімкий розвиток ринку пристроїв та систем домашньої автоматизації. Інтернет речей (IoT) виступає технологічною основою, що дозволяє об'єднувати різноманітні побутові прилади, датчики та виконавчі механізми в єдину керовану мережу, відкриваючи нові горизонти для взаємодії людини з її домом.

Актуальність даної роботи зумовлена зростаючим попитом на інтегровані рішення для управління домашніми системами. Якщо раніше автоматизація обмежувалася окремими функціями, то сьогодні користувачі прагнуть до комплексного контролю над освітленням, кліматом, мультимедійними системами та безпекою з єдиного центру або навіть дистанційно. Керування освітленням дозволяє не лише створювати бажану атмосферу та економити електроенергію, але й імітувати присутність господарів для підвищення безпеки. Інтеграція музичних систем у загальну інфраструктуру «Розумного будинку» забезпечує новий рівень комфорту, дозволяючи насолоджуватися улюбленими композиціями в будь-якій кімнаті або синхронізувати аудіосупровід з різними сценаріями (наприклад, пробудження чи вечірка). Автоматизоване управління дверима та доступом є критично важливим елементом безпеки, що забезпечує контроль над входом до приміщення, можливість дистанційного відкриття/закриття та моніторинг подій.

Незважаючи на велику кількість комерційних рішень для «Розумного

будинку», вони часто є дорогими, закритими для модифікацій або вимагають складного налаштування. У цьому контексті особливої значущості набувають відкриті апаратні платформи, такі як Arduino. Завдяки своїй доступності, гнучкості, великій кількості сумісних модулів та активній спільноті розробників, Arduino стала популярним інструментом для створення кастомізованих систем автоматизації, в тому числі й для освітніх та дослідницьких цілей. Розробка системи управління основними функціями будинку на базі Arduino дозволяє не лише отримати практичний досвід у галузі мікроконтролерної техніки та програмування, але й створити функціональний прототип, адаптований до конкретних потреб користувача.

Проте, реалізація такої багатофункціональної системи пов'язана з низкою завдань. Необхідно забезпечити надійну взаємодію між різними компонентами: датчиками, виконавчими пристроями (динаміки, сервоприводи, екрани) та центральним мікроконтролером. Важливим аспектом є розробка інтуїтивно зрозумілого інтерфейсу користувача для управління системою, а також програмного забезпечення, здатного обробляти команди, аналізувати дані з датчиків та реалізовувати задані сценарії автоматизації. Особливої уваги потребують питання стабільності роботи системи, її енергоефективності та можливостей для подальшого масштабування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні поняття

Мікроконтролер є фундаментальним будівельним блоком для численних сучасних електронних систем, і він відіграє центральну роль у розробці системи управління освітленням, музикою та дверима в «Розумному будинку» на базі Arduino. По суті, мікроконтролер – це цілий комп'ютер, мініатюризований до розмірів однієї інтегральної мікросхеми. Його головне призначення полягає не в виконанні широкого спектру завдань, як у персональних комп'ютерів, а в керуванні конкретними пристроями та процесами в межах вбудованої системи [1].

В основі кожного мікроконтролера лежить центральний процесорний пристрій (ЦПП), який виступає його «мозком». ЦПП відповідає за обробку інструкцій, закладених у програмному кодї, виконання арифметичних та логічних операцій, а також за координацію роботи всіх інших складових мікроконтролера. Невід'ємною частиною є пам'ять, яка буває декількох типів. Пам'ять програм, зазвичай реалізована як Flash-пам'ять або ПЗП (постійний запам'ятовуючий пристрій), зберігає саму програму (прошивку), яку мікроконтролер буде виконувати [2]. Ця пам'ять є енергонезалежною, тобто її вміст зберігається навіть після вимкнення живлення. Оперативна пам'ять (ОЗП) слугує для тимчасового зберігання даних, з якими процесор працює безпосередньо під час виконання програми – це можуть бути змінні, проміжні результати обчислень.

Ключовою особливістю, що відрізняє мікроконтролери від мікропроцесорів загального призначення, є наявність на одному кристалі різноманітних периферійних пристроїв. Ці пристрої забезпечують взаємодію мікроконтролера із зовнішнім світом. Порти вводу-виводу загального призначення (GPIO) дозволяють підключати датчики, кнопки, світлодіоди,

реле та інші компоненти, зчитуючи з них сигнали або керуючи ними. Таймери та лічильники необхідні для генерації точних часових затримок, вимірювання тривалості подій або формування ШІМ-сигналів, які, наприклад, можуть використовуватися для регулювання яскравості світла. Аналого-цифрові перетворювачі (АЦП) дають змогу мікроконтролеру «розуміти» аналогові сигнали від сенсорів (наприклад, температури або рівня освітленості), перетворюючи їх у цифровий формат. Для зворотного перетворення існують цифро-аналогові перетворювачі (ЦАП). Не менш важливими є комунікаційні інтерфейси, такі як UART, SPI, I2C, які дозволяють мікроконтролеру обмінюватися даними з іншими мікросхемами, комп'ютерами або модулями бездротового зв'язку [3].

Принцип роботи мікроконтролера полягає в послідовному виконанні команд програми, що зберігається в його пам'яті. Процесор зчитує команду, декодує її, виконує відповідну операцію (яка може включати взаємодію з периферійними пристроями або внутрішньою пам'яттю) і переходить до наступної команди. Цей цикл повторюється безперервно, доки на мікроконтролер подається живлення. Механізми переривань дозволяють мікроконтролеру ефективно реагувати на асинхронні зовнішні події без необхідності постійно опитувати стан всіх можливих джерел таких подій.

Платформа Arduino значно спрощує процес роботи з мікроконтролерами, надаючи готові плати з мікроконтролером (зазвичай сімейства AVR від Microchip, наприклад, ATmega328P на популярній платі Arduino Nano) та необхідною обв'язкою, а також зручне середовище розробки (Arduino IDE) з простою мовою програмування, заснованою на C/C++. Тобто, можна сказати що, Arduino гарний вибір для створення прототипів та реалізації проєктів, подібних до системи «Розумний будинок», дозволяючи зосередитися на логіці роботи системи, а не на низькорівневих аспектах програмування мікроконтролера.

1.2 Різновиди мікроконтролерів

Ринок мікроконтролерів надзвичайно різноманітний, і вибір конкретного типу залежить від специфіки завдання, вимог до продуктивності, енергоспоживання, вартості та набору необхідної периферії. Існує декілька ключових ознак, за якими класифікують ці універсальні мікросхеми, що дозволяє розробникам підібрати оптимальне рішення для своїх проєктів.

Однією з фундаментальних характеристик є розрядність мікроконтролера. Цей параметр визначає, скільки біт даних центральний процесор може обробляти одночасно, а також ширину його шин даних та адрес. Найбільш поширеними є:

– 8-розрядні мікроконтролери – це часто найпростіші та найдешевші мікроконтролери. Вони ідеально підходять для нескладних завдань керування, де не потрібна висока обчислювальна потужність, наприклад, для керування окремими побутовими приладами, зчитування даних з простих датчиків або виконання базових логічних операцій. Мікроконтролери AVR від Microchip (раніше Atmel), що лежать в основі багатьох популярних плат Arduino (як-от Arduino Uno з ATmega328P), є яскравими представниками цього класу. Вони вирізняються простотою програмування та достатньою для багатьох проєктів «Розумного будинку».

– 16-розрядні мікроконтролери – пропонують компроміс між 8-бітними та 32-бітними аналогами, забезпечуючи більшу продуктивність та точність обчислень порівняно з 8-бітними, при цьому часто маючи менше енергоспоживання, ніж 32-бітні. Вони знаходять застосування в промисловій автоматизації, автомобільній електроніці та більш складних системах керування.

– 32-розрядні мікроконтролери – на сьогодні є найбільш динамічно зростаючим сегментом. Вони пропонують значну обчислювальну потужність, великий адресний простір, що дозволяє працювати з більшими обсягами даних та складнішими програмами. Мікроконтролери на базі архітектури ARM (наприклад, сімейства Cortex-M) домінують у цьому сегменті. Вони широко

використовуються у смартфонах, планшетах, мережевому обладнанні, а також у більш просунутих системах «Розумного будинку», які вимагають обробки мультимедіа, реалізації складних мережевих протоколів або алгоритмів штучного інтелекту. Плати Arduino, такі як Arduino Due (на базі SAM3X8E ARM Cortex-M3) або плати на базі мікроконтролерів ESP32 (які поєднують 32-розрядний процесор з Wi-Fi та Bluetooth), також належать до цієї категорії.

– 64-розрядні мікроконтролери – хоча менш поширені у традиційному розумінні «мікроконтролерів для вбудованих систем», вони починають з'являтися у високопродуктивних додатках, де потрібна обробка дуже великих обсягів даних [4].

Наступною важливою ознакою класифікації мікроконтролерів є архітектура пам'яті, основними є:

– Архітектура фон Неймана, використовує спільний простір пам'яті та спільну шину для команд і даних. Це може обмежувати швидкість виконання, оскільки команди та дані не можуть зчитуватися одночасно.

– Гарвардська архітектура, має роздільні простори пам'яті та шини для команд і даних, що дозволяє одночасно вибирати наступну команду та отримувати доступ до даних поточної команди, підвищуючи таким чином швидкодію. Більшість сучасних мікроконтролерів, включаючи AVR та багато ARM-мікроконтролерів, використовують модифіковану Гарвардську архітектуру, яка поєднує переваги обох підходів.

Також мікроконтролери можна розрізнити за набором команд процесора:

– CISC (Complex Instruction Set Computer) – мікроконтролери з повним набором команд. Кожна команда може виконувати складну послідовність операцій. Це може спростити програмування на низькому рівні, але ускладнює сам процесор.

– RISC (Reduced Instruction Set Computer) – мікроконтролери зі скороченим набором команд. Команди простіші та виконуються швидше (зазвичай за один такт). Складні операції реалізуються послідовністю простих

команд. Архітектура ARM є типовим прикладом RISC-архітектури. RISC-процесори часто більш енергоефективні.

За сферою застосування та інтегрованою периферією мікроконтролери можуть бути:

- Універсальними (General-purpose) – мають збалансований набір периферійних пристроїв і підходять для широкого кола завдань.

- Спеціалізованими – тобто, оптимізовані для конкретних застосувань. Наприклад, мікроконтролери для керування двигунами (з великою кількістю ШІМ-каналів та АЦП), для обробки сигналів (DSP-мікроконтролери), для автомобільних систем (з CAN-інтерфейсами та розширеним температурним діапазоном),

На ринку існує багато сімейств мікроконтролерів від різних виробників. Найбільш відомі з них:

- AVR (Microchip/Atmel) – популярні 8-розрядні мікроконтролери, відомі своєю простотою, надійністю та широким використанням у платформі Arduino.

- PIC (Microchip) – ще одне велике сімейство, що включає 8-, 16- та 32-розрядні мікроконтролери, відомі своєю гнучкістю та низьким енергоспоживанням.

- ARM (Advanced RISC Machines) – мікроконтролери на базі ARM домінують у 32-розрядному сегменті завдяки високій продуктивності, енергоефективності та широкому спектру периферійних можливостей [5].

- ESP (Espressif Systems) – сімейство мікроконтролерів (ESP8266, ESP32), що здобуло величезну популярність завдяки інтегрованим модулям Wi-Fi та Bluetooth, низькій вартості та сумісності з середовищем Arduino. Вони є чудовим вибором для IoT-проектів та систем «Розумного будинку», що потребують бездротового зв'язку.

1.3 Мікроконтролери сімейства AVR

Мікроконтролери сімейства AVR (Alf-Egil Bogen Vegard Wollan RISC) займають важливе місце у світі вбудованих систем завдяки вдалому поєднанню продуктивності, енергоефективності та багатого набору периферійних пристроїв. Розроблені спочатку компанією Atmel (згодом придбаною Microchip Technology), ці мікроконтролери здобули величезну популярність, особливо в освітньому середовищі та серед ентузіастів електроніки, значною мірою завдяки їх широкому використанню в якості основи для платформи Arduino. Саме ця платформа стала каталізатором, що вивів мікроконтролери AVR на новий рівень доступності для широкого загалу.

В основі мікроконтролерів AVR лежить 8-бітна RISC-архітектура (Reduced Instruction Set Computer). Це означає, що вони використовують спрощений набір команд, більшість з яких виконуються за один тактовий цикл, що сприяє високій швидкодії при відносно низьких тактових частотах. Ключовою особливістю архітектури AVR є її модифікована Гарвардська структура, яка передбачає роздільні шини та простори пам'яті для програм (Flash) та даних (SRAM). Такий підхід дозволяє одночасно вибирати наступну команду програми та отримувати доступ до даних, необхідних для поточної операції, що усуває «вузьке місце фон Неймана» і підвищує загальну продуктивність системи. Процесорне ядро містить значну кількість (зазвичай 32) 8-бітних регістрів загального призначення, які безпосередньо підключені до арифметико-логічного пристрою (АЛП), забезпечуючи швидке виконання операцій [6].

Мікроконтролери AVR оснащені широким спектром інтегрованих периферійних модулів, що робить їх універсальними для багатьох завдань керування, включаючи компоненти систем «Розумного будинку»:

– Порти вводу-виводу загального призначення (GPIO) - це одна з ключових переваг. Кожен пін порту може бути індивідуально налаштований на ввід або вивід, дозволяючи легко підключати кнопки, світлодіодні

індикатори, реле (для керування освітленням чи силовими колами дверей), датчики стану (наприклад, геркони для дверей) тощо.

– Аналого-цифрові перетворювачі (АЦП) - більшість AVR мікроконтролерів мають вбудований багатоканальний АЦП, зазвичай з роздільною здатністю 10 біт. Це дозволяє вимірювати аналогові сигнали від різноманітних сенсорів, таких як датчики освітленості, температури, вологості або потенціометри.

– Таймери/лічильники - це універсальні апаратні блоки, що можуть використовуватися для генерації точних часових інтервалів, підрахунку зовнішніх імпульсів, а також для формування сигналів широтно-імпульсної модуляції (ШИМ, PWM). ШИМ є надзвичайно корисною для плавного регулювання яскравості освітлення, керування швидкістю обертання двигунів або навіть для простої генерації звукових сигналів.

Сімейство AVR включає різні підродини, такі як ATtiny (для найпростіших завдань з обмеженими ресурсами), ATmega (найбільш поширені, з хорошим балансом ресурсів, наприклад, ATmega328P, ATmega2560) та ATxmega (більш продуктивні з розширеною периферією). Вибір конкретного мікроконтролера AVR для плати Arduino зазвичай визначається балансом між кількістю портів вводу-виводу, обсягом пам'яті, наявністю специфічної периферії та вартістю.

Хоча мікроконтролери AVR можуть програмуватися мовами низького рівня, такими як Асемблер, або більш універсальною мовою С, їхня популярність значно зросла завдяки платформі Arduino. Середовище розробки Arduino IDE надає спрощений варіант мови С++ (відомий як Wiring) та велику кількість готових бібліотек. Ці бібліотеки є ключовим елементом екосистеми Arduino, оскільки вони інкапсулюють складні низькорівневі операції з регістрами мікроконтролера AVR у прості для розуміння та використання функції. Наприклад, для ініціалізації та використання ШИМ-каналу на AVR безпосередньо мовою С потрібно налаштувати декілька регістрів таймера, тоді як в Arduino це досягається викликом однієї функції `analogWrite(pin, value)`.

Таке абстрагування робить розробку програм для AVR значно доступнішою для широкого кола користувачів, включаючи тих, хто не має глибоких знань в електроніці чи програмуванні мікроконтролерів [7]. Ця синергія між апаратною простотою та потужністю AVR та програмною доступністю Arduino стала запорукою їхнього успіху.

1.4 ARM-сумісні мікроконтролери

Мікроконтролери, що базуються на архітектурі ARM на сьогодні є одними з найпоширеніших та найдинамічніше зростаючих у світі вбудованих систем. Наразі у світі є величезне різноманіття ARM-сумісних мікроконтролерів на ринку, які знаходять застосування у найширшому спектрі пристроїв – від побутової електроніки та мобільних телефонів до складних промислових систем керування та автомобільних компонентів. Ключовими принципами архітектури ARM є використання RISC-концепції, що забезпечує високу продуктивність при виконанні простих, оптимізованих команд, а також значний акцент на енергоефективності.

Для сегмента мікроконтролерів компанія ARM розробила спеціалізоване сімейство процесорних ядер Cortex-M. Це сімейство включає низку ядер, таких як Cortex-M0, Cortex-M0+, Cortex-M3, Cortex-M4, Cortex-M7, Cortex-M23, Cortex-M33 та інші, кожне з яких оптимізоване для певного балансу між продуктивністю, енергоспоживанням та набором функцій. Така масштабованість дозволяє розробникам підбирати оптимальне ядро для конкретного завдання. Всі мікроконтролери на базі Cortex-M є 32-розрядними, що надає їм переваги над 8-бітними аналогами у вигляді більшого адресного простору, ширших шляхів даних та, як правило, вищої продуктивності при виконанні складних обчислень та обробці великих обсягів даних.

Велика кількість відомих виробників напівпровідників випускають мікроконтролери на базі ядер ARM Cortex-M. Серед них варто відзначити:

- STMicroelectronics із надзвичайно популярною серією STM32, яка

включає величезну кількість моделей на базі різних ядер Cortex-M (від M0 до M7 та новіших), з різноманітними периферійними наборами та обсягами пам'яті.

- NXP Semiconductors з лінійками LPC та Kinetis.
- Texas Instruments, що пропонує серії Tiva C та MSP432.
- Microchip Technology (яка придбала Atmel) з серіями SAM D, SAM L, SAM C та іншими.
- Infineon Technologies з мікроконтролерами PSoC та XMC.
- Nordic Semiconductor, відома своїми мікроконтролерами серії nRF, що часто мають інтегровані радіомодулі для Bluetooth Low Energy, Thread, Zigbee.

Така конкуренція та різноманітність виробників забезпечують широкий вибір мікроконтролерів з різними комбінаціями периферійних пристроїв, обсягів пам'яті, корпусів та цінових категорій [8].

Периферійні можливості ARM-сумісних мікроконтролерів зазвичай значно перевершують можливості 8-бітних аналогів. Окрім стандартного набору, такого як GPIO, АЦП, таймери, UART, SPI, I2C, вони часто пропонують: більшу кількість цих ресурсів, вищу продуктивність та роздільну здатність, а також розширені периферійні модулі.

Програмування ARM-мікроконтролерів переважно здійснюється мовами C та C++. Для розробки використовуються як професійні комерційні середовища (наприклад, Keil MDK, IAR Embedded Workbench), так і безкоштовні інструменти на базі GCC (GNU Compiler Collection) у поєднанні з різними редакторами коду та системами збирання (наприклад, STM32CubeIDE від STMicroelectronics, MCUXpresso IDE від NXP, або VS Code з відповідними плагінами) [9].

Для складних проєктів з великою кількістю паралельних завдань часто використовуються операційні системи реального часу (RTOS), такі як FreeRTOS, Zephyr OS або Mbed OS. Важливо зазначити, що платформа Arduino також розширилася для підтримки ARM-мікроконтролерів; плати, як-от Arduino Due, Portenta, Nano 33 IoT/BLE, а також Giga R1 WiFi, базуються на

ядрах ARM Cortex-M. Це робить потужність ARM-архітектури доступнішою для широкого кола розробників.

Перевагами ARM-сумісних мікроконтролерів є їхня висока продуктивність, енергоефективність, масштабованість рішень, великий вибір від різних виробників, а також потужна екосистема програмних інструментів та підтримки. У контексті систем «Розумного будинку» ARM-мікроконтролери дозволяють реалізовувати складну логіку керування, обробляти дані з численних датчиків, підтримувати мережеві протоколи (TCP/IP, MQTT), забезпечувати бездротовий зв'язок, а в деяких випадках навіть виконувати локально завдання базового машинного навчання або обробки голосових команд [10]. Їхні можливості роблять їх чудовим вибором для створення багатофункціональних та інтелектуальних пристроїв домашньої автоматизації.

1.5 Мікроконтролер ATmega328P

Центральним обчислювальним елементом класичної та найпоширенішої версії плати Arduino Nano, є 8-бітний мікроконтролер ATmega328P, розроблений компанією Microchip (раніше Atmel). Ця мікросхема є «мозком», тим же що й у популярній платі Arduino Uno, проте для забезпечення мініатюрних розмірів Nano, ATmega328P зазвичай монтується у компактному корпусі.

В основі ATmega328P лежить 8-бітна RISC-архітектура AVR (можемо побачити на рисунку 2.1), відома своєю ефективністю. На платі Arduino Nano цей мікроконтролер стандартно працює на тактовій частоті 16 МГц, яка генерується за допомогою зовнішнього кварцового резонатора. Його ресурси пам'яті добре збалансовані для широкого кола вбудованих завдань: він має 32 КБ Flash-пам'яті для зберігання програмного коду (скетчів), причому близько 2 КБ з цього обсягу зазвичай відведено під завантажувач Arduino (bootloader), що спрощує процес прошивки. Для оперативних даних та змінних під час

виконання програми доступно 2 КБ SRAM, а для довготривалого зберігання налаштувань або важливих даних, які повинні зберігатися після вимкнення живлення, передбачено 1 КБ EEPROM [11].

Для взаємодії із зовнішнім світом Arduino Nano через мікроконтролер ATmega328P надає 14 цифрових пінів (маркованих D0-D13). Кожен із них може бути програмно налаштований як вхід або вихід. Важливою особливістю є те, що 6 із цих цифрових пінів (D3, D5, D6, D9, D10, D11) підтримують апаратну широтно-імпульсну модуляцію. Це дозволяє генерувати аналогоподібні сигнали, що є критично важливим, наприклад, для плавного регулювання яскравості світлодіодів чи керування швидкістю обертання двигунів. Окрім цифрових, плата Nano пропонує 8 аналогових входів (A0-A7), що на два більше, ніж у стандартної Arduino Uno. Ці входи підключені до вбудованого 10-бітного аналого-цифрового перетворювача (АЦП) мікроконтролера, який здатен вимірювати напругу в діапазоні від 0 до 5 Вольт з роздільною здатністю 1024 рівні.

Комунікаційні можливості ATmega328P реалізовані через декілька апаратних інтерфейсів. На платі Nano для послідовного зв'язку (UART) використовуються піни D0 (RX) та D1 (TX). Цей інтерфейс, як правило, з'єднаний з інтегральною схемою USB-to-Serial конвертера (наприклад, FTDI або CH340 на різних версіях плат), що забезпечує легке підключення до комп'ютера через Mini-USB порт як для завантаження програм, так і для обміну даними [12]. Величезна популярність Arduino Nano та її мікроконтролера ATmega328P значною мірою зумовлена екосистемою Arduino. Завантажувач (bootloader), попередньо прошитий в мікроконтролер, дозволяє користувачам легко завантажувати скетчі через USB без потреби у зовнішніх апаратних програматорах. Інтуїтивно зрозуміле середовище розробки Arduino IDE разом з мовою програмування Wiring (яка базується на C++) та великою кількістю готових бібліотек значно спрощують програмування, абстрагуючи складні апаратні деталі. Це робить ATmega328P на платі Nano доступним навіть для початківців.

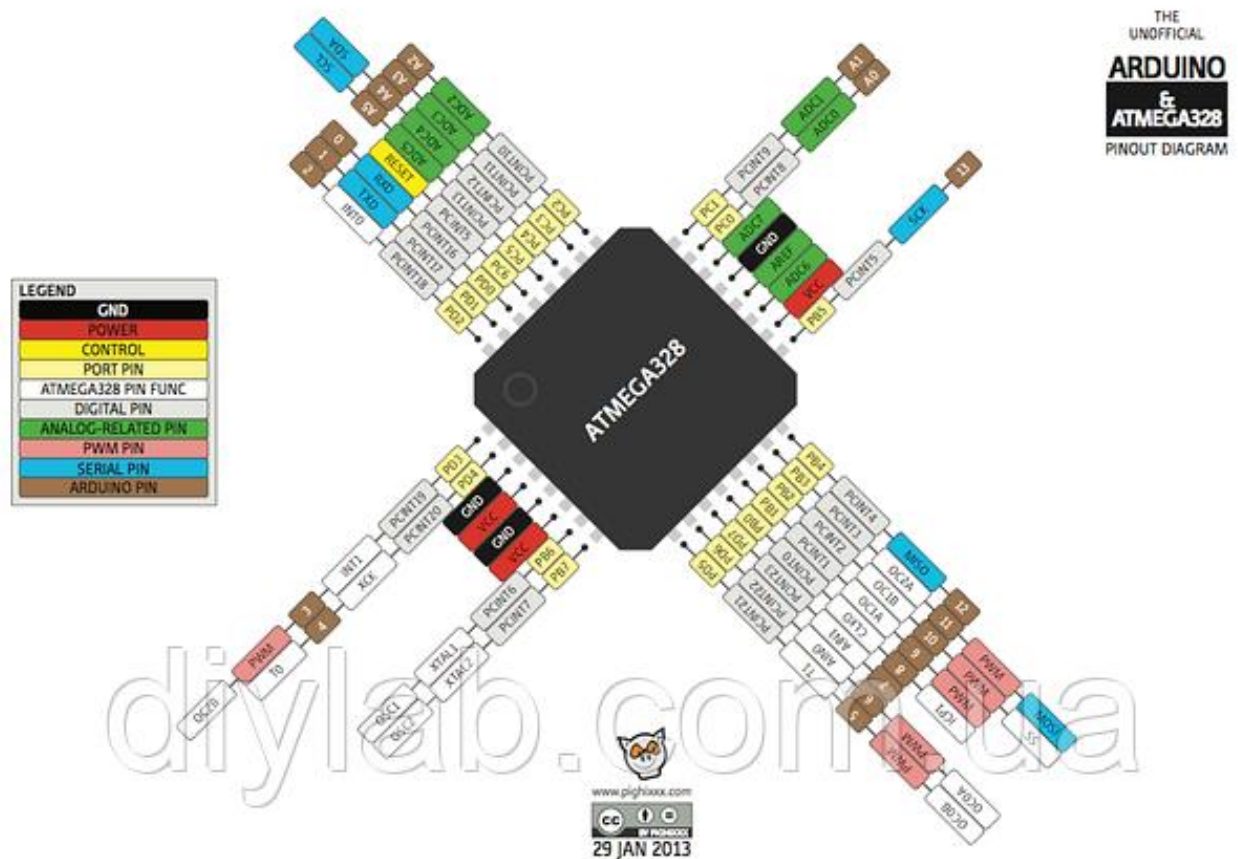


Рисунок 2.1 – Розпіновка мікроконтролера ATMEGA328

1.6 Мета та постановка завдання

Метою даної кваліфікаційної роботи є розробка та реалізація багатофункціональної системи для управління освітленням, музичним супроводом та доступом (дверима) в рамках концепції «Розумного будинку» на базі платформи Arduino.

Для досягнення поставленої мети необхідно вирішити комплекс завдань, що включає розробку схемотехнічних рішень, необхідних для інтеграції освітлювальних приладів, аудіосистеми та механізмів керування дверима в єдину систему. Також потрібно запрограмувати мікроконтролер Arduino для ефективної обробки команд, що надходять від користувача, та для моніторингу стану всіх підключених до системи пристроїв.

Важливою частиною є інтеграція відповідних апаратних модулів, таких

як RFID-модуль, сервомотор, LCD-дисплей, джойстик, світлодіодна стрічка та зумер, з метою забезпечення автоматизованого керування ключовими функціями «Розумного будинку». Зокрема, це передбачає реалізацію автоматизованого керування доступом до приміщення за допомогою RFID-міток та пульта керування, створення системи інтелектуального керування освітленням, а також забезпечення централізованого керування аудіосистемою для відтворення музики.

Результатом виконання цих завдань має стати працюючий прототип системи «Розумний будинок». Цей прототип дозволить користувачеві ефективно та зручно керувати освітленням, відтворенням музики та доступом до приміщення, що сприятиме підвищенню рівня комфорту, безпеки та енергоефективності житла. Головним призначенням розроблюваного проєкту є створення автоматизованої системи керування основними побутовими функціями, спрямованої на покращення умов проживання.

2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ПРОЄКТУ

2.1 Вибір мови програмування та середовища розробки

Вибір мови програмування та відповідного середовища розробки є одним із фундаментальних та стратегічно важливих етапів на початковій стадії будь-якого програмного проєкту. Це рішення суттєво впливає на весь подальший життєвий цикл розробки, ефективність роботи, а також на якість, продуктивність та можливості масштабування кінцевого програмного продукту. Неправильно обрана мова програмування може призвести до ускладнення процесу розробки, збільшення часу, необхідного для реалізації функціоналу, та виникнення труднощів при подальшій підтримці й модифікації системи.

Аналогічно, середовище розробки (IDE) відіграє ключову роль у продуктивності програміста. Сучасні IDE пропонують широкий набір інструментів – від інтелектуального автодоповнення коду та потужних засобів налагодження до інтеграції з системами контролю версій та інструментами для рефакторингу. Ефективне середовище значно скорочує час на рутинні операції, допомагає швидше виявляти та виправляти помилки, а також сприяє написанню більш чистого та структурованого коду.

2.1.1 Arduino IDE для Arduino

Arduino IDE є надзвичайно популярним середовищем розробки для платформи Arduino, передусім завдяки своїй винятковій простоті та інтуїтивності, що робить його ідеальним вибором для початківців та швидкого прототипування. Легкість освоєння підкріплюється величезною кількістю доступних бібліотек, які значно спрощують роботу з різноманітними сенсорами та модулями, а також потужною підтримкою великої світової

спільноти. Кросплатформеність, безкоштовність та пряма інтеграція з апаратним забезпеченням Arduino ще більше посилюють його привабливість. Хоча для дуже великих та складних проєктів його функціоналу, зокрема можливостей редактора коду та відсутності повноцінного апаратного відладчика, може бути недостатньо. Отже переваги Arduino IDE для навчання, хобі та створення більшості типових проєктів на Arduino є беззаперечними, забезпечуючи низький поріг входження та ефективну реалізацію ідей.

2.1.2 Мультиплатформенність PlatformIO

PlatformIO є потужною кросплатформенною екосистемою для розробки вбудованих систем, що кардинально розширює можливості порівняно з більш простими інструментами. Його ключова перевага полягає у винятковій мультиплатформенності: PlatformIO підтримує тисячі різноманітних мікроконтролерних плат, від популярних Arduino до значно потужніших ESP32, STM32 та інших архітектур, дозволяючи розробникам легко перемикатися між ними. Крім того, він інтегрує підтримку численних програмних фреймворків, таких як Arduino, ESP-IDF, Mbed, надаючи гнучкість у виборі програмної основи проєкту. Найчастіше PlatformIO використовується як розширення для професійних середовищ розробки, наприклад, Visual Studio Code, надаючи доступ до їхніх розширених функцій.

Однак, така універсальність та широкий набір можливостей обумовлюють значно вищий поріг входження та складність освоєння, особливо для новачків, вимагаючи глибшого розуміння процесів налаштування та управління проєктами. Ця складність є платою за надзвичайну гнучкість, потужний контроль над процесом розробки та можливість ефективно працювати над складними, багатокomпонентними проєктами з використанням найсучасніших технологій у сфері вбудованих систем.

2.1.3 Visual Studio Code з розширенням для Arduino

Visual Studio Code, розроблений Microsoft, є надзвичайно популярним, безкоштовним та потужним редактором коду, який завдяки своїй гнучкості та великій екосистемі розширень може бути адаптований для широкого спектру завдань програмування, включаючи розробку для мікроконтролерів Arduino. Замість використання окремого, більш простого середовища як Arduino IDE, розробники можуть інтегрувати функціональність для роботи з Arduino безпосередньо у VS Code за допомогою спеціальних розширень, таких як офіційне розширення «Arduino» від Microsoft.

Використання VS Code з таким розширенням надає низку суттєвих переваг. По-перше, розробники отримують доступ до всіх потужних функцій сучасного редактора: значно краще автодоповнення коду (IntelliSense), зручну навігацію по великих проєктах, вбудовану підтримку системи контролю версій Git, інтегрований термінал та широкі можливості для кастомізації робочого простору за допомогою тем та інших розширень. Це створює більш професійне та продуктивне середовище, особливо для складніших проєктів Arduino, де структура коду та управління файлами відіграють важливу роль.

Крім того, деякі розширення для Arduino у VS Code пропонують покращені можливості для компіляції, завантаження прошивки та моніторингу послідовного порту, а також іноді інтеграцію з відладчиками, що є кроком вперед порівняно з базовими можливостями Arduino IDE. Це дозволяє швидше знаходити та виправляти помилки, а також ефективніше керувати процесом розробки.

Звісно, початкове налаштування VS Code та встановлення необхідних розширень для роботи з Arduino може вимагати трохи більше часу та зусиль, ніж просто запуск стандартної Arduino IDE. Користувачеві може знадобитися розібратися з конфігураційними файлами (наприклад, `arduino.json` для розширення Microsoft) для вказівки типу плати, порту та інших параметрів.

2.1.4 Обґрунтування програмно-технологічних рішень проєкту

Для реалізації програмної частини проєкту, зокрема для програмування мікроконтролерної складової, було проведено аналіз відповідних мов програмування та інтегрованих середовищ розробки. Враховуючи специфіку роботи з апаратним забезпеченням, вимоги до взаємодії з сенсорами та виконавчими пристроями, як основну мову програмування я обрав Arduino Sketch (C++). Цей вибір зумовлений великою кількістю готових бібліотек для роботи з різноманітними апаратними компонентами та можливістю низькорівневого доступу до ресурсів мікроконтролера.

Для розробки, компіляції, завантаження програмного коду на мікроконтролер та його налагодження було використано інтегроване середовище розробки Arduino IDE. Дане середовище забезпечує зручний інтерфейс для роботи з проєктами Arduino, включає необхідні інструменти для компіляції коду під цільову платформу та засоби для моніторингу роботи пристрою, що суттєво оптимізувало процес створення вбудованого програмного забезпечення.

2.2 Призначення та базові функції проєкту

Головним призначенням проєкту «Розумний будинок» є створення автоматизованої системи керування основними побутовими функціями житла, спрямованої на підвищення рівня комфорту, зручності, безпеки та ефективності використання ресурсів для мешканців. Проєкт має на меті продемонструвати можливість інтеграції сучасних технологій у повсякденне життя, забезпечуючи інтуїтивно зрозуміле та гнучке управління ключовими елементами домашнього середовища. Система розробляється з акцентом на модульність та можливість подальшого розширення функціоналу.

Для досягнення поставленої мети, на поточному етапі реалізації проєкт «Розумний будинок» включає декілька базових функцій:

– По-перше, це автоматизоване керування доступом, що забезпечує можливість дистанційного або автоматизованого контролю входних дверей. Реалізація передбачає відкриття дверей за допомогою пульта керування або через використання RFID міток. Це підвищує зручність, усуваючи необхідність у традиційних ключах, та може покращити безпеку шляхом реєстрації подій доступу, а також надавати можливість тимчасового доступу гостям.

– Наступною важливою функцією є інтелектуальне керування освітленням. Система дозволяє користувачам гнучко керувати освітленням, вмикаючи/вимикаючи світло за допомогою того ж самого пульта керування. Така функція не лише створює комфортні умови освітлення відповідно до потреб мешканців.

– Третьою базовою можливістю є централізоване керування аудіосистемою, що надає змогу відтворювати музику в будинку. Керування цим процесом здійснюється через пульт керування домом, з можливістю відтворення локальних аудіофайлів.

Реалізація цих базових функцій є першим кроком до створення комплексної та інтелектуальної системи управління житловим простором, що робить його більш адаптивним до потреб сучасної людини.

3 СПЕЦИФІКАЦІЯ АПАРАТНИХ КОМПОНЕНТІВ СИСТЕМИ

3.1 Технологічна база мікроелектромеханічних систем

Технологія МЕМС, що розшифровується як мікроелектромеханічні системи, є передовою галуззю інженерії, яка зосереджена на проектуванні та виробництві мініатюрних пристроїв, де механічні елементи, такі як мікроважелі, мембрани, шестерні чи дзеркала, нерозривно поєднані з електронними компонентами, наприклад, схемами обробки сигналів чи керування, на єдиній інтегральній підкладці, найчастіше кремнієвій. Ці пристрої мають характерні розміри в діапазоні від декількох мікрометрів до кількох міліметрів. Процес їх виготовлення значною мірою запозичує методи з напівпровідникової промисловості, що дозволяє здійснювати масове паралельне виробництво, забезпечуючи високу точність, відтворюваність та відносно низьку вартість одиниці при великих обсягах.

Сьогодні технологія МЕМС є невід'ємною частиною величезної кількості пристроїв, що нас оточують. Їх можна знайти у смартфонах та планшетах (акселерометри, гіроскопи, мікрофони), в автомобільній електроніці (датчики тиску в шинах, сенсори для систем безпеки), у медичному обладнанні (мініатюрні діагностичні сенсори, імплантовані пристрої), в промислових системах моніторингу та керування, а також у побутовій техніці, такій як струменеві принтери. Важливість технології МЕМС полягає у її здатності забезпечувати значну мініатюризацію пристроїв, зниження їх енергоспоживання та вартості, водночас розширюючи їх функціональні можливості та відкриваючи шлях до створення принципово нових продуктів та рішень у найрізноманітніших сферах.

3.2 LCD-монітор

LCD 1602A є дуже поширеним символьним рідкокристалічним дисплеєм, призначеним для відображення текстової інформації, цифр та простих символів у проектах на мікроконтролерах, зокрема з Arduino (рисунок 3.1). Його назва вказує на формат, тобто він може відображати 16 символів у двох рядках. Такі дисплеї базуються на стандартному контролері HD44780 або сумісному з ним, що забезпечує легку інтеграцію та програмування за допомогою стандартних бібліотек, як-от LiquidCrystal для Arduino.

Підключення може здійснюватися паралельно, використовуючи 4 або 8 ліній даних, або значно простіше через спеціальний модуль-адаптер I2C, який дозволяє керувати дисплеєм лише по двох дротах. Популярність LCD 1602A зумовлена його низькою ціною, широкою доступністю та простотою використання, це робить цей монітор ідеальним для відображення базової інформації та створення простих користувацьких меню.



Рисунок 3.1 – LCD-монітор 1602A

3.3 Сервомотор

Сервомотор – це спеціальний тип двигуна, що забезпечує дуже точне керування кутовим положенням свого вала завдяки вбудованій системі зворотного зв'язку (рисунок 3.2). Він складається з двигуна постійного струму, редуктора для збільшення крутного моменту, датчика положення (наприклад, потенціометра) та електронної схеми керування. Ця схема отримує керуючий сигнал, зазвичай у вигляді ШІМ-імпульсів, де тривалість імпульсу визначає бажаний кут. Електроніка постійно порівнює задане положення з поточним, яке зчитується з датчика, і коригує роботу двигуна, доки вал не досягне потрібної позиції та не зафіксується в ній, активно утримуючи положення навіть під навантаженням. Завдяки простоті керування мікроконтролерами, високому крутному моменту та точності позиціонування, сервомотори знаходять широке застосування в робототехніці, радіокерованих моделях, анімації та різноманітних автоматизованих системах.



Рисунок 3.2 – Сервомотор

3.4 Активний зумер

Активний зумер – це простий звуковипромінюючий пристрій, який зазвичай генерує звук фіксованої частоти при поданні на нього постійної напруги живлення, оскільки має вбудований генератор коливань (рисунок 3.3). У контексті проєктів на Arduino, його підключення є дуже простим: достатньо подати живлення з цифрового виходу мікроконтролера. Хоча активні зумери за своєю природою призначені для видачі одного тону, у даному проєкті для Arduino, завдяки використанню спеціальної бібліотеки, вдалося реалізувати відтворення послідовностей звукових сигналів. Ця бібліотека дозволяє керувати ввімкненням та вимкненням зумера з точною синхронізацією, створюючи таким чином прості ритмічні структури або елементарні мелодії, використовуючи характерний для даного активного зумера тон. Це розширює можливості його застосування від звичайних сигналів попередження до більш інформативних звукових сповіщень.



Рисунок 3.3 – Активний зумер

3.5 RFID-модуль

RFID-модуль для платформи Arduino є компактним електронним компонентом, який інтегрує технологію радіочастотної ідентифікації у мікроконтролерні проєкти. Його ключове призначення - забезпечити бездротовий обмін даними з RFID-мітками, дозволяючи системі розпізнавати їх на невеликій відстані, можемо побачити на рисунку 3.4. Коли мітка потрапляє в електромагнітне поле, що генерується антеною модуля, вона активується і передає свій унікальний ідентифікатор або інші збережені дані. Модуль, в свою чергу, приймає цей сигнал, обробляє його та передає інформацію на Arduino для подальшого використання.

Одним з найпоширеніших прикладів є модуль RC522, що працює на частоті 13.56 МГц і підтримує популярні стандарти міток. Підключення до Arduino зазвичай здійснюється через стандартний інтерфейс SPI, що вимагає з'єднання кількох цифрових пінів. Для розробників доступні численні бібліотеки програмного забезпечення, які значно спрощують процес програмування, надаючи готові функції для ініціалізації модуля, виявлення міток та зчитування їхніх даних.

Це робить RFID-модуль цінним інструментом для додавання функціональності безконтактної ідентифікації до різноманітних пристроїв та систем, керованих Arduino.



Рисунок 3.4 – RFID-модуль RC522

3.6 Модуль керування двоосьовий XY джойстик KY-023

Модуль KY-023 є поширеним компонентом для проєктів на базі Arduino, що слугує для ручного введення керуючих сигналів (рисунок 3.5). Він являє собою двоосьовий аналоговий джойстик, функціонально подібний до мініатюрних маніпуляторів, які можна знайти в ігрових контролерах, та додатково оснащений інтегрованою тактильною кнопкою. Основний принцип його роботи полягає у перетворенні фізичного відхилення руків'я по горизонтальній (вісь X) та вертикальній (вісь Y) осях в аналогові електричні сигнали. Це досягається за допомогою двох вбудованих потенціометрів, опір яких змінюється залежно від положення руків'я, що, в свою чергу, змінює напругу на відповідних вихідних контактах (VRx та VRy). Натискання на руків'я зверху активує окрему кнопку (SW), яка видає цифровий сигнал.

Для підключення до плати Arduino модуль KY-023 використовує аналогові входи для зчитування значень з осей X та Y, а також один цифровий вхід для обробки сигналу від кнопки. Окрім цього, необхідне підключення до джерела живлення (зазвичай +5В) та землі (GND).

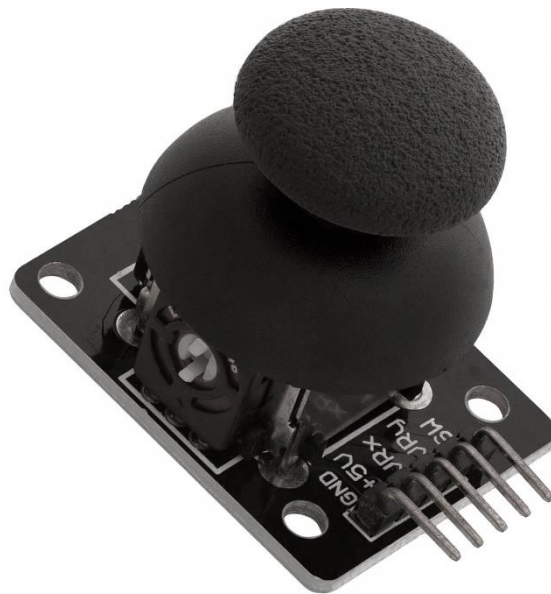


Рисунок 3.5 - Модуль керування KY-023

4 ПРОЄКТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

4.1 Проєктування системи «Розумний будинок»

Проєктування системи «Розумний будинок» базувалося на меті створення багатофункціонального пристрою для управління освітленням, музичним супроводом та дверима на основі платформи Arduino. Ключовим принципом була модульність та можливість інтеграції доступних апаратних компонентів для реалізації заданого функціоналу. Система спроектована для розміщення у макеті будинку, що наочно демонструє її роботу (рисунок 4.1). Також на рисунку 4.2 можемо побачити електричну функціональну схему самого проєкту.

Центральним елементом системи є плата Arduino Nano, яка містить мікроконтролер ATmega328P. Цей мікроконтролер відповідає за обробку команд користувача, зчитування даних з сенсорів та керування виконавчими пристроями.

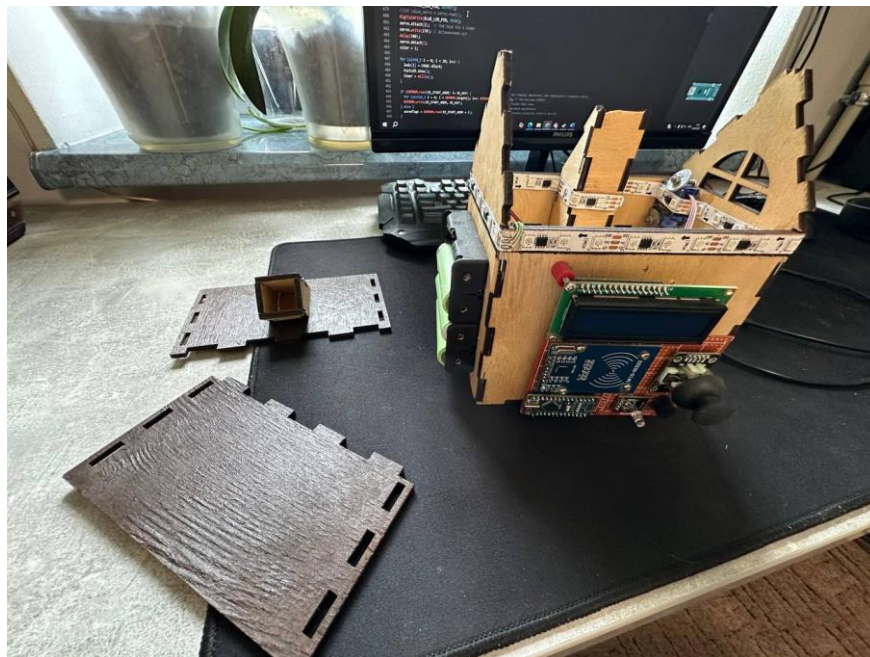


Рисунок 4.1 – Зовнішній вигляд макету системи «Розумний будинок»

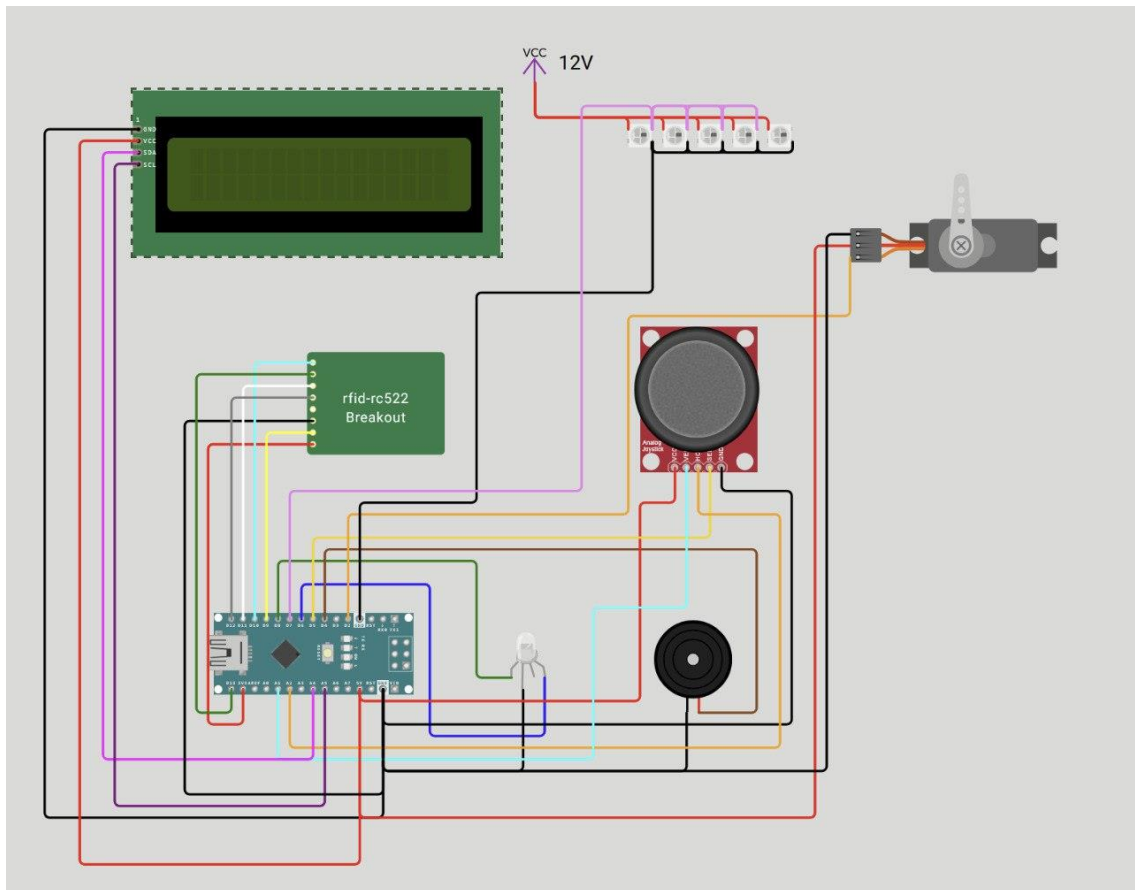


Рисунок 4.2 – Електрична функціональна схема проекту

Для взаємодії користувача з системою передбачено:

Модуль керування (джойстик KY-023) - це двоосьовий аналоговий джойстик з кнопкою використовується для навігації по меню та вибору функцій системи. Він підключений до аналогових (для осей X та Y) та цифрового (для кнопки) входів Arduino.

LCD-монітор (1602A) - це символний рідкокристалічний дисплей формату 16x2 слугує для відображення поточної інформації, пунктів меню та стану системи. Він підключений до Arduino через інтерфейс I2C за допомогою спеціального модуля-адаптера, що спрощує підключення.

На рисунку 4.2 продемонстровано панель керування системи, де розміщені джойстик, LCD-дисплей, RFID-модуль та плата Arduino Nano.

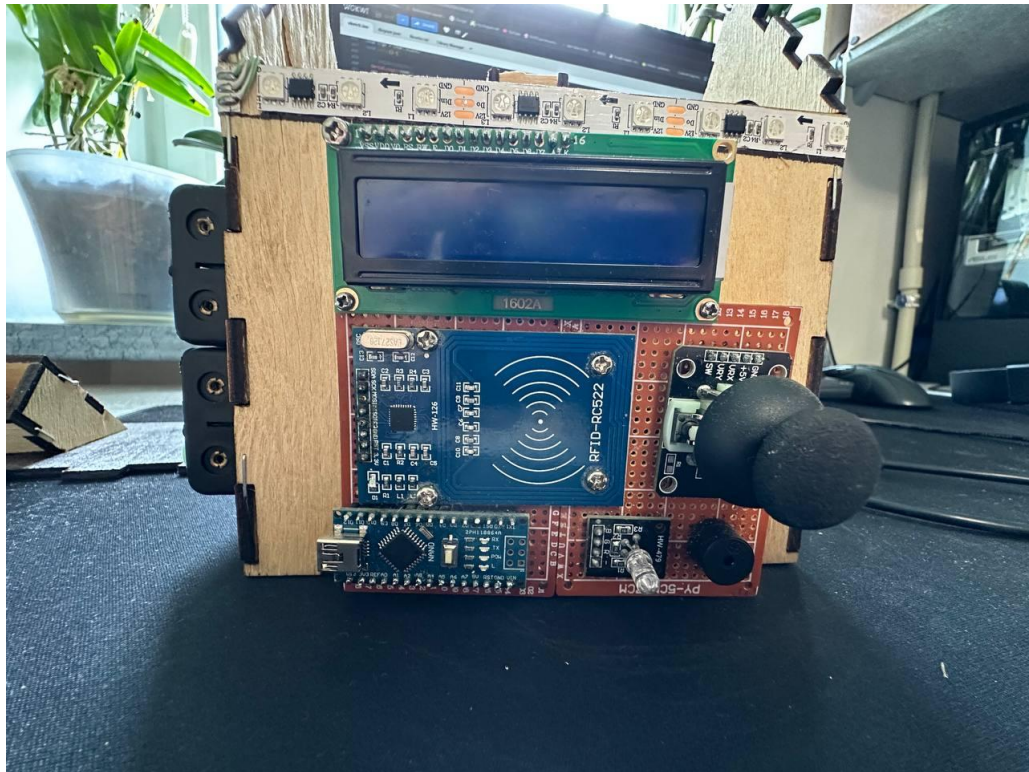


Рисунок 4.2 – Панель керування системи «Розумний будинок»

Реалізація керування доступом здійснюється за допомогою:

RFID-модуля (RC522) – використовується для безконтактної ідентифікації користувачів за допомогою RFID-міток. Модуль підключений до Arduino через інтерфейс SPI. При піднесенні зареєстрованої мітки система надає доступ. Система дозволяє зберігати та видаляти RFID-мітки.

Сервомотора, котрий забезпечує фізичне відкриття та закриття механізму дверей. Керується ШІМ-сигналом з цифрового виходу Arduino. Станом дверей (відчинено/зачинено) можна керувати через меню.

Для керування освітленням використовується:

Світлодіодна стрічка, звичайна адресна світлодіодна стрічка дозволяє гнучко керувати кольором освітлення. Вона підключена до цифрового піну Arduino (пін 7) та керується за допомогою бібліотеки FastLED. Користувач може обирати колір світіння (наприклад, синій, зелений, червоний) та вмикати/вимикати освітлення через меню системи.

На рисунку 4.3 видно реалізацію підсвітки на канті макету будинку.



Рисунок 4.3 – Реалізація підсвітки на макеті будинку

Підсистема звукового супроводу реалізована за допомогою активного зумера, котрий використовується для відтворення простих мелодій та звукових сигналів сповіщення. Підключений до цифрового виходу Arduino (пін 4). Система дозволяє обирати та відтворювати попередньо запрограмовані мелодії (наприклад, «Merry Christmas», «Rickroll», «Doom»).

Система живиться від зовнішнього джерела, що забезпечує необхідну напругу для Arduino Nano та всіх підключених модулів. На макеті передбачено використання портативного джерела живлення, акумуляторного блоку, здебільшого для того щоб забезпечити електрикою LED-стрічку, що видно на рисунку 4.4. Так як стрічка бере 12 Вольт, а ми через Arduino отримуємо лише 5 Вольт, то було зроблено такий акумуляторний блок з 3-ма акумуляторами типу 18650, який забезпечує необхідну напругу.

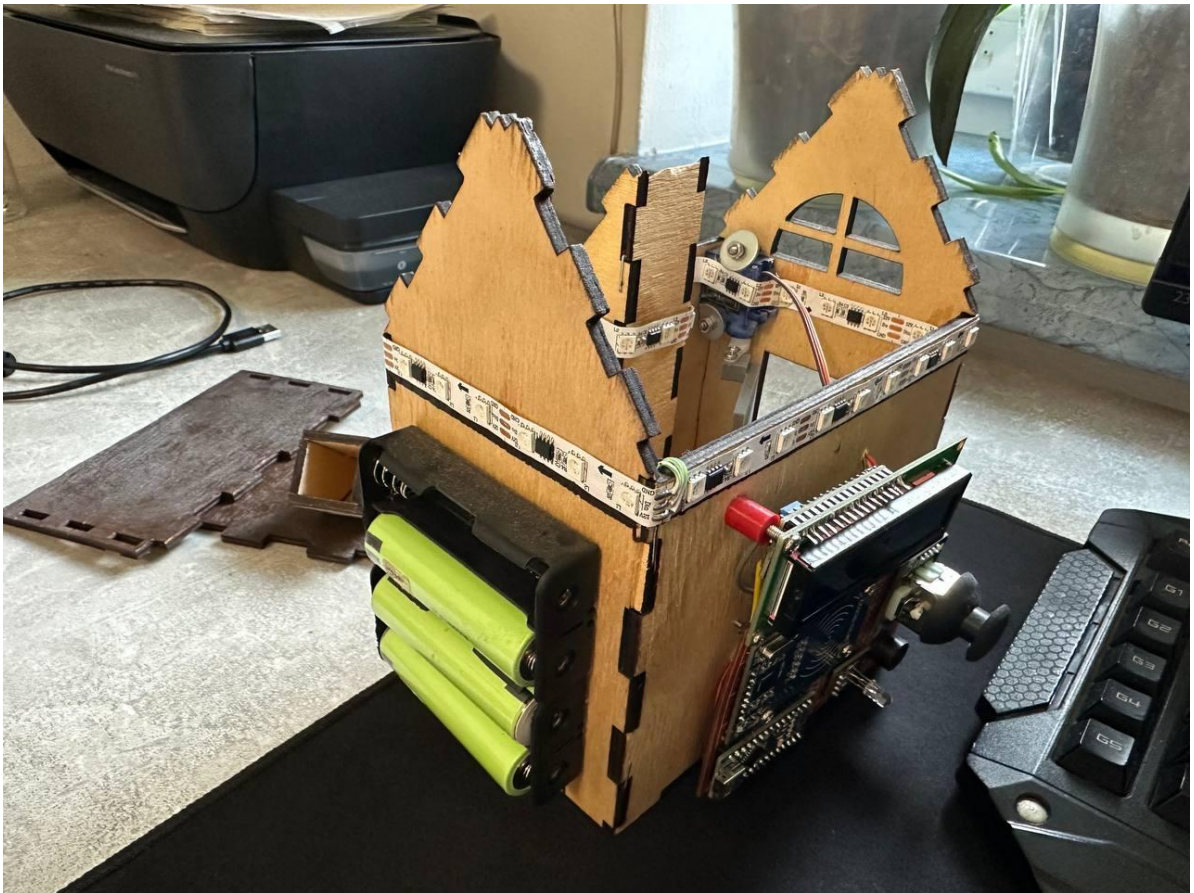


Рисунок 4.4 – Живлення LED-стрічки

Програмне забезпечення розроблено в середовищі Arduino IDE мовою Arduino Sketch (C++). Логіка програми включає обробку сигналів з джойстика, вивід інформації на LCD-дисплей, взаємодію з RFID-модулем, керування сервомотором, світлодіодною стрічкою та зумером. Реалізовано меню для вибору функцій системи: керування музикою (вибір мелодії, старт/стоп), освітленням (вибір кольору, вкл/викл), та дверима (відкрити, закрити, зберегти/видалити RFID-картку). Головне меню надає доступ до підменю «Music», «Light», «Door», «ON/OFF».

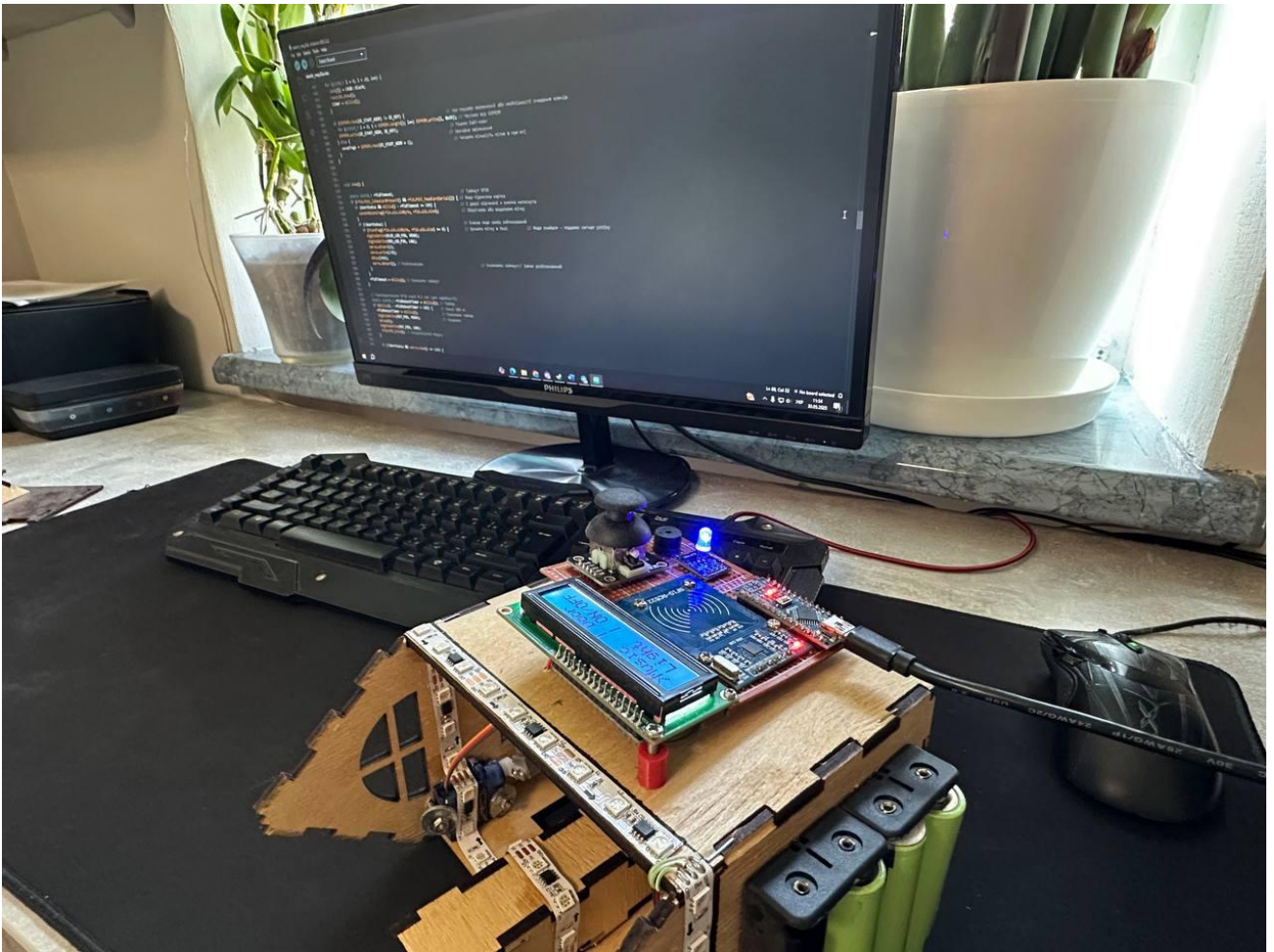


Рисунок 4.5 – Процес налагодження системи

4.2 Тестування розроблених функцій системи «Розумний будинок»

Після завершення етапів проєктування та програмної реалізації, було проведено всебічне тестування системи «Розумний будинок» для перевірки її коректної роботи та відповідності поставленим вимогам. Тестування охоплювало кожен функціональний блок системи окремо та їхню інтегровану роботу. Перевірка інтерфейсу користувача розпочалася з тестування засобів введення та відображення інформації. Детально аналізувалася коректність роботи джойстика KY-023, який є основним елементом для навігації по меню, що відображається на LCD-дисплеї 1602A. Тести включали перевірку точного переміщення курсора по всіх пунктах меню – як горизонтально (зміна

активного стовпця опцій), так і вертикально (зміна активного рядка опцій) – при відповідному відхиленні руків'я джойстика по осях X та Y. Також перевірялося чітке спрацювання вибору опції при натисканні на вбудовану кнопку джойстика (рисунок 4.6). Контролювалася швидкість відгуку системи на дії користувача та чіткість відображення інформації на екрані, зокрема правильність виведення назв головного меню («Music», «Light», «Door», «ON/OFF») та відповідних підменю, згідно з логікою, закладеною у функціях PrintMenu та CursorSetting.

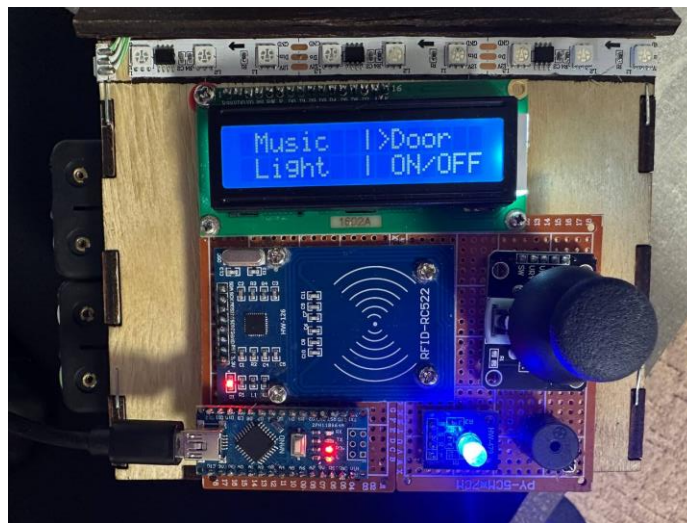


Рисунок 4.6 – Процес тестування програмних функцій проєкту

Після підтвердження справності інтерфейсу, детально тестувалася функціональність керування доступом. По-перше, перевірялося керування дверима через відповідні пункти меню «Door»: опція «Open» мала відчиняти двері, а «Close» – зачиняти, що супроводжувалося рухом сервомотора, підключеного до піна 2, у визначені положення 170 градусів для відчинення та 105 для зачинення (рисунок 4.7).

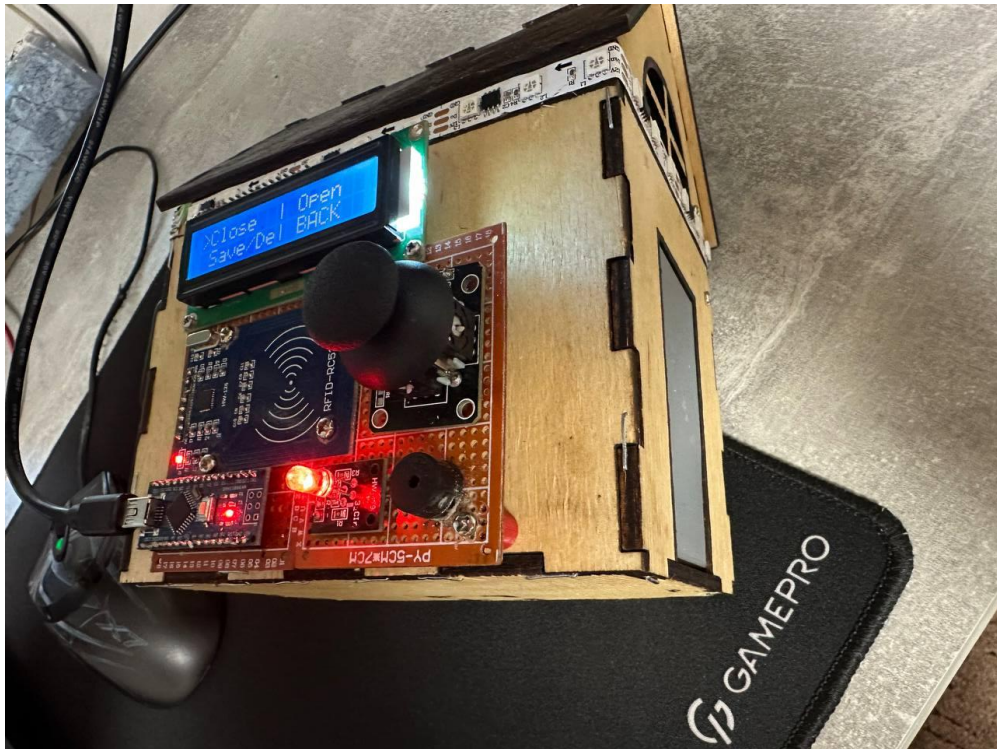


Рисунок 4.7 – Тестування відкриття дверей через пункт керування

По-друге, ретельно тестувалася робота RFID-системи на базі модуля MFRC522. Процедура збереження нової RFID-мітки включала вибір опції «Save/Del» в меню «Door», піднесення нової картки до зчитувача, після чого її унікальний ідентифікатор (UID) мав бути успішно записаний в енергонезалежну пам'ять EEPROM; при цьому лічильник збережених карток `savedTags` мав збільшитися, що також зберігається в EEPROM. Аналогічно, тестувалася процедура видалення RFID-мітки: при повторному піднесенні вже збереженої картки до зчитувача, коли активна опція «Save/Del», картка мала видалитися з пам'яті (шляхом затирання її даних та зсуву останньої картки на її місце, якщо це не остання картка), а лічильник `savedTags` – зменшуватися. Критичними тестами були перевірка надання доступу – піднесення зареєстрованої картки до зчитувача повинно було призводити до спрацювання сервомотора та імітації відчинення дверей, що супроводжувалося світінням синього світлодіода та згасанням червоного (рисунок 4.8). Навпаки, піднесення незареєстрованої картки не повинно було викликати жодної

реакції з боку механізму дверей, а червоний світлодіод мав продовжувати світитися, вказуючи на зачинений стан, що підтверджувало коректну логіку безпеки та роботу змінної `doorStatus`.

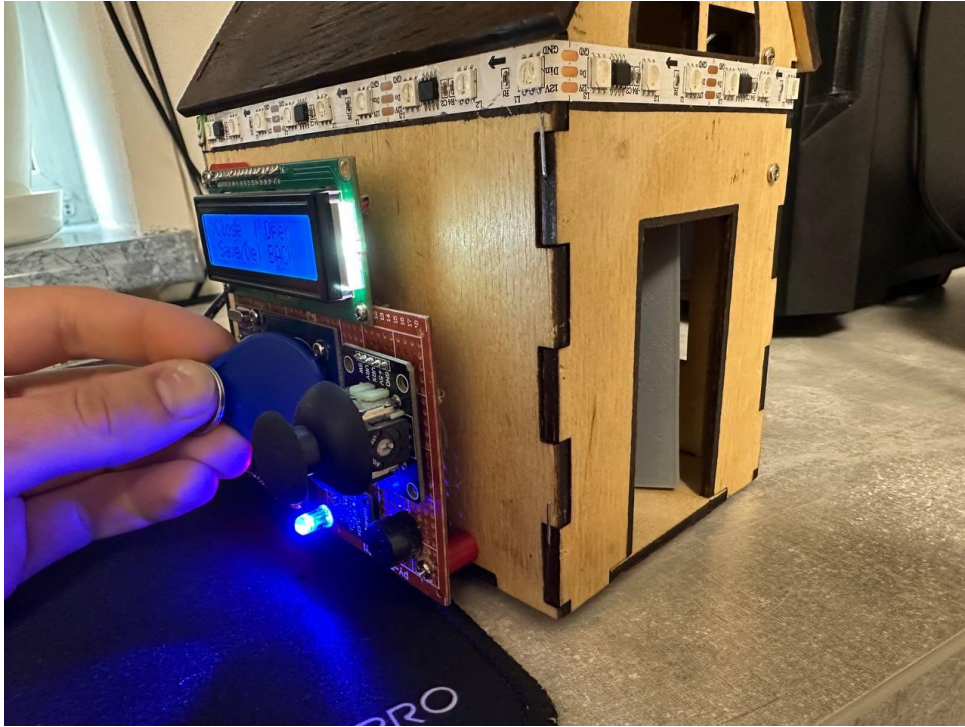


Рисунок 4.8 – Тестування відкривання дверей за допомогою RFID-мітки

Далі тестування системи керування освітленням передбачало перевірку всіх закладених можливостей. Було підтверджено, що через підменю «Light», доступне з головного меню, користувач може вмикати та вимикати світлодіодну стрічку, стан освітлення контролювався змінною `status_lights`. Особлива увага приділялася функції зміни кольору освітлення: послідовний вибір опцій «Red», «Green», «Blue» у відповідному підменю призводив до негайної зміни кольору світіння всієї адресної стрічки, підключеної до пін 7 та керованої бібліотекою `FastLED` (рисунок 4.9).

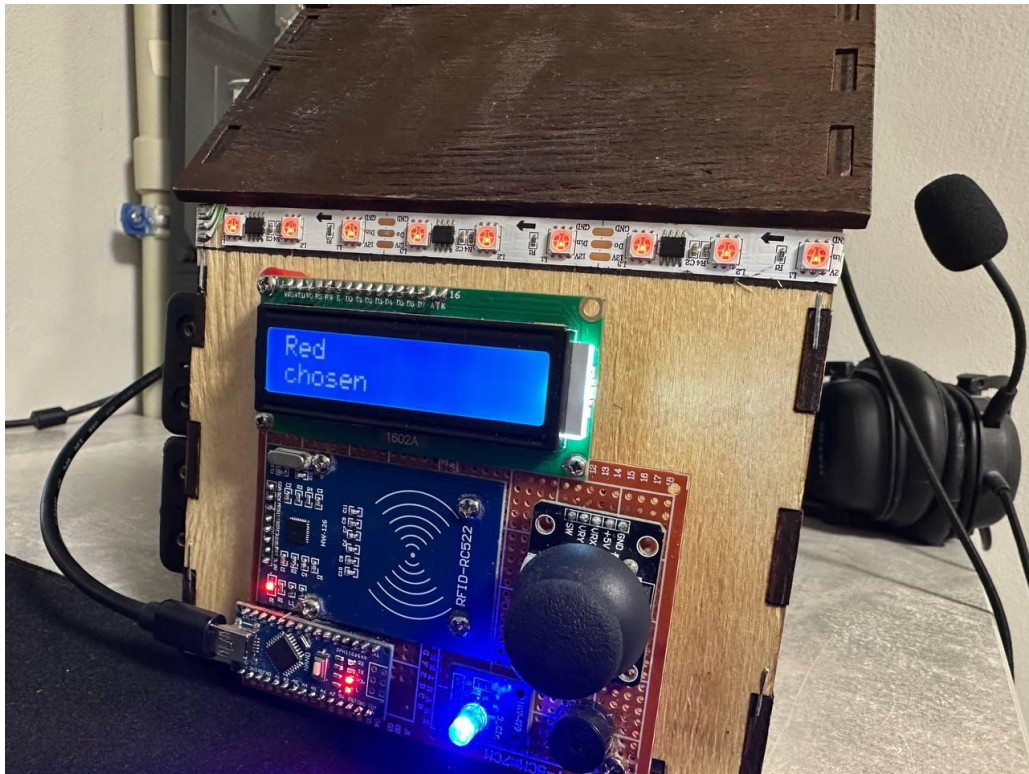


Рисунок 4.9 – Тестування системи освітлення

Перевірка функцій звукового супроводу полягала у тестуванні відтворення мелодій через активний зумер, підключений до піна 4. З підменю «Music», доступного з головного меню, послідовно обиралися доступні мелодії, такі як «MerryChr», «Rickroll» або «Doom» (рисунок 4.10). Після вибору кожної мелодії перевірялося її коректне та повне відтворення зумером згідно з визначеними нотами та їх тривалістю. Також тестувалися функції керування відтворенням через екран «ON/OFF»: опція «MUSIC» мала запускати або продовжувати відтворення обраної мелодії (змінна `song_status` встановлювалася в `true`), а опція «STOP» – негайно припиняти будь-яке звучання зумера (`noTone(BUZZER_PIN)`).



Рисунок 4.10 – Тестування звукового супроводу

На завершальному етапі було проведено інтеграційне тестування, спрямоване на перевірку узгодженої роботи всіх підсистем. Контролювалося, що активація однієї функції (наприклад, відтворення музики) не призводить до непередбачуваних змін у роботі інших (наприклад, системи освітлення чи доступу). Оцінювалася загальна стабільність роботи системи «Розумний будинок» при тривалій експлуатації, одночасному використанні кількох функцій (наприклад, світло горить під час відтворення музики) та багаторазовому виконанні різних команд через меню. Результати тестів підтвердили, що всі розроблені функції працюють належним чином, а система є стабільною та відповідає поставленим завданням.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно розроблено та реалізовано прототип багатофункціональної системи для управління основними елементами «Розумного будинку» на базі платформи Arduino Nano. Проєкт продемонстрував можливість ефективної інтеграції апаратних компонентів, таких як LCD-дисплей, джойстик, RFID-модуль, сервомотор, світлодіодна стрічка та зумер, для забезпечення керування освітленням, музичним супроводом та доступом до приміщення.

Було досліджено різновиди мікроконтролерів та обґрунтовано вибір ATmega328P як оптимального рішення для даного проєкту, враховуючи його доступність, функціональність та широку підтримку з боку екосистеми Arduino. Розроблено програмне забезпечення в середовищі Arduino IDE мовою C++, що включає логіку обробки команд користувача, взаємодію з периферійними пристроями та реалізацію меню для інтуїтивного керування системою.

Проведене тестування підтвердило коректну роботу всіх заявлених функцій: навігації по меню, керування дверима за допомогою RFID-міток та команд з меню, зміни кольорів освітлення, а також відтворення попередньо заданих мелодій. Система показала стабільну роботу та відповідність поставленим завданням щодо підвищення комфорту, зручності та елементів безпеки житла.

Результатом роботи є функціонуючий макет системи «Розумний будинок», який може слугувати основою для подальшого розвитку та розширення функціоналу, наприклад, шляхом додавання нових датчиків, виконавчих пристроїв або інтеграції з бездротовими технологіями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аронець О. Arduino для початківців. Івано-Франківськ, 2018. 190 с.
2. Цирульник С. М., Лисенко Г. Л. Програмування мікроконтролерів AVR : навчальний посібник. Вінниця : ВНТУ, 2018. 111 с.
3. Поджаренко В. О., Кучерук В. Ю., Севастьянов В. М. Основи мікропроцесорної техніки : навчальний посібник. Вінниця : ВНТУ, 2006. 226 с.
4. Матвієнко М., Розен В. Комп'ютерна схемотехніка. Навчальний посібник / Матвієнко М., Розен В. – К. : Ліра-К, 2014. – 192 с.
5. Sarah Harris. Digital Design and Computer Architecture: ARM Edition, 1st edition / S. Harris, D. Harris. – Morgan Kaufmann, 2015. – 584 с.
6. Arduino основи програмування. *Geekmatic* : веб-сайт. URL: https://geekmatic.in.ua/ua/arduino_osnovyi_programmirovaniya (дата звернення: 10.05.2025).
7. Wayne Wolf. FPGA-Based System Design, Prentice Hall Modern Semiconductor Design Series / Wolf W. – Prentice Hall, 2004. – 530 с.
8. Noergaard T. Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers. 2nd ed. Elsevier Inc., 2013. 600 p.
9. Розумний будинок AJAX [Електронний ресурс] URL: <https://ajax.systems.ua/> (дата звернення: 08.05.2025)
10. Розумний будинок Broadlink [Електронний ресурс] URL: <https://broadlink.com.ua/> (дата звернення: 09.05.2025)
11. Документація Arduino Nano [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.arduino.cc/hardware/nano>
12. Patrascu M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and MultiAgent Manufacturing and Robotics. / Patrascu., 2014. – 544 с.