

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

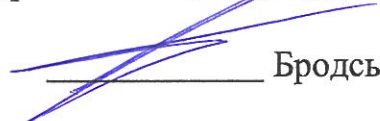
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект системи Smart Home для приватного
домоволодіння на базі мікроконтролерів MCU та SBC»

Виконав студент 4-го курсу групи КІ-4
спеціальності 123 «Комп'ютерна інженерія»

 Андрусенко О. М.

Керівник доцент кафедри КІ та КБ, к. т. н., доцент

 Бродський Ю. Б.

Рецензент професор кафедри КІ та КБ, д. т. н., доцент

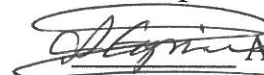
 Воротніков В. В.

Житомир – 2022

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко

1 березня 2022 р.

ЗАВДАННЯ

на випускн кваліфікаційну роботу бакалавра

Студент: Андрусенко Олег Михайлович

Тема роботи: «Проект системи Smart Home для приватного домоволодіння на базі мікроконтролерів MCU та SBC»

затверджена Наказом університету від 1 березня 2022 р. № 127/с

Термін здачі студентом закінченої роботи 10 червня 2022 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):
Предметом дослідження є використання програмованих контролерів в організації системи «Smart Home», а об'єктом дослідження – процес розробки та реалізації даної системи.

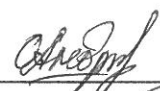
Обладнання та програмне забезпечення:

1. Мікроконтролери ESP8266 NodeMCU V3.
2. Одноплатний комп'ютер Raspberry Pi.
3. Platformio IDE.

4. IoT сервіс Blynk.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

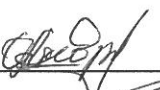
Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Бродський Ю. Б.	1.03.22	1.03.22
2	Єфіменко А. А.	1.03.22	1.03.22
3	Головня О. С.	1.03.22	1.03.22

Студент  Андрусенко О.М.

Керівник  Бродський Ю. Б.

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	01.03.2022	
2	Аналіз існуючих рішень системи «Smart Home»	02.03 – 16.03.2022	
3	Розробка структури системи	17.03 – 17.04.2022	
4	Розробка структурних та функціональних схем системи	18.04 – 25.04.2022	
5	Підбір компонентів для реалізації розробленої системи	26.04 – 02.05.2022	
6	Розробка програмного забезпечення для мікроконтролерів	03.05 – 03.06.2022	
7	Тестування розробленої системи	04.06 – 11.06.2022	
8	Оформлення пояснювальної записки	12.06 – 22.06.2022	
9	Захист дипломної роботи	24.06.2022	

Студент  Андрусенко О.М.

Керівник  Бродський Ю. Б.

АНОТАЦІЯ

Текстова частина викладена на 72 сторінках друкованого тексту. Пояснювальна записка має 13 сторінок додатків. Список використаних джерел містить 20 найменувань та займає 2 сторінки. В роботі наведено 102 рисунки. Загальний обсяг роботи – 85 сторінок.

Мета кваліфікаційної роботи полягає у розробці системи «Smart Home» на основі програмованих мікроконтролерів.

У кваліфікаційній роботі досліджено та налаштовано апаратні засоби та розроблено програмне забезпечення, яке використовується для функціонування системи «Smart Home». Розглянуто особливості побудови таких систем та проаналізовано їх переваги та недоліки.

Проведено проектування підсистем, наведено схеми їх роботи та взаємодії між собою. Створено алгоритми роботи мікроконтролерів підсистем та на їх основі розроблено відповідне програмне забезпечення.

Після розробки програмного забезпечення наведено процес налаштування та тестування системи «Smart Home»

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, МІКРОКОНТРОЛЕР, ОДНОПЛАТНИЙ КОМП'ЮТЕР, ЛОКАЛЬНИЙ СЕРВЕР, BLYNK.

					КІК.КР.Б – 123 – 01 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект системи Smart Home для приватного домоволодіння на базі мікроконтролерів MCU та SBC	Літ.	Арк.	Аркушів
Розроб.		Андрусенко О.М.		10.06				
Керівник		Бродський Ю. Б.		20.06				
Реценз.		Воротніков В.В.		20.06				
Н. контр.		Бродський Ю. Б.		20.06				
Затверд.		Єфіменко А.А.		20.06	ФІКТ, Гр. КІ-4			

ANNOTATION

The text part is laid out on 72 pages of printed text. The explanatory note has 13 pages of annexs. The list of used sources contains 20 names and occupies 2 pages. 102 figures are given in the paper. The total volume of work is 85 pages.

The goal of qualification work is to develop the system "Smart Home" on the basis of programd microcontrollers.

In the qualification work the hardware tools were studied and adjusted and the software was developed, which is used for the functioning of the system "Smart Home". Features of such systems are considered and analyzed.

The subsystems were designed, the schemes of their work and interaction among themselves were presented. Algorithms of microcontrollers of sub-systems are created and on their basis the corresponding software is developed.

After development of the software, the process of configuration and testing of the system "Smart Home" is given.

Keywords: INTERNET OF THINGS, MICROCONTROLLER, SINGLEBOARD COMPUTER, LOCAL SERVER, BLYNK.

		Андрусенко О.М.			<i>KIK.KP.B – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ «SMART HOME»	10
1.1 Історія створення та концепція Smart Home.	10
1. 2 Загальна структура (архітектура) існуючих систем Smart Home.	12
1. 3 Аналіз технічних рішень.	13
1. 4 Висновки до розділу 1	17
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ «SMART HOME»	18
2. 1 Розробка структури системи.....	18
2. 2 Схемні рішення: структурні та функціональні схеми підсистем	20
2.3 Управління системою. Модель взаємодії підсистем в системі «Smart Home»	34
2.4 Висновки до розділу 2	35
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ СИСТЕМИ «SMART HOME».....	36
3.1 Обґрунтування технічних рішень вибору компонентів забезпечення функціонування підсистем в системі «Smart Home»	36
3.2 Розробка програмної частини системи. Інтерфейс користувача	44
3.3 Тестування системи «Smart Home».....	57
3.4 Висновки до розділу 3	69
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення.

MCU – MicroController Unit (блок мікроконтролера).

SBC – Single Board Computer (одноплатний комп'ютер).

GPIO – General-Purpose Input/Output (інтерфейс введення/виведення загального призначення).

I2C – Inter-Integrated Circuit (послідовна асиметрична шина зв'язку).

UART – Universal Asynchronous Receiver/Transmitter (універсальний асинхронний приймач-передавач).

SPI – Serial Peripheral Interface (послідовний периферійний інтерфейс).

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Термін «розумний» будинок можна інтерпретувати як «будинок, що побудовано розумно», тобто його спроектовано так, щоб всі складові його частини та сервіси могли б поєднуватися одне з одним з мінімальними затратами, а керування ними та їх обслуговування було зручним і простим. Така система має безліч переваг. За допомогою системи керування власники будинку можуть створювати складні функції управління системами будинку, оскільки всі вони здатні працювати злагоджено і спільно.

На сьогоднішній день в світі існує багато шляхів для побудови системи керування «Smart Home», але всі вони розроблені та орієнтовані під екосистеми певного виробника. Використання програмованих мікроконтролерів при розробці такої системи робить поєднання компонентів різних виробників в одну систему можливим, що дозволяє підібрати найкращий з точки зору ціни та якості набір приладів для створення системи «Smart Home». Тому тематика дипломної роботи є актуальною і потребує подальшого дослідження.

Метою кваліфікаційної роботи є розробка системи «Smart Home» на основі програмованих мікроконтролерів.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- *проаналізувати існуючі системи «Smart Home»;*
- *створити структуру системи, що базується на програмованих мікроконтролерах, описати алгоритм її роботи;*
- *розробити систему «Smart Home», що базується на програмованих мікроконтролерах.*

Об'єктом дослідження є процес розробки та реалізації системи «Smart Home».

Предметом дослідження є використання програмованих контролерів в організації системи «Smart Home».

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ «SMART HOME»

1.1 Історія створення та концепція Smart Home.

Постійне наростання кількості побутової техніки в приватних будинках, впровадження охоронних систем та сигналізації в них привело до необхідності створення єдиної надійної та зручної системи для управління всіма вищеперерахованими пристроями.

Ідея «розумних» будинків, так як і більшості сучасних пристроїв, початково з'явилася на сторінках творів жанру фантастики. Проте матеріалізація даної ідеї розпочалася лише у XX столітті після початку широкого застосування електрики в приватних будинках та розвитку інформаційних технологій. До перших згадок про віддалені пристрої контролю можна віднести розробки Ніколи Тесли щодо дистанційного керування транспортними засобами та кораблями у 1898 році.

Термін «розумний будинок» було введено американською асоціацією «Housebuilders» у 1984 році.

Існує думка, що «розумний» будинок це просто житло, в якому встановлено безліч електроніки. Проте це не так. «Розумний» будинок це ціла інфраструктура, яка перетворює житло в автономну екосистему.

На початку свого розвитку таких систем їх основним призначенням була економія енергоресурсів за допомогою відключення електроприладів, що не використовувалися, тим самим система регулювала споживання електроенергії на основі потреб користувача в них.

Чому ж ця система така затребувана і що в неї входить? Система «Smart Home» здатна:

- керувати системами опалення та водопостачання;
- реагувати на нештатні ситуації, такі як потоп, або ж пожежа;

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

- *регулювати рівень вологості повітря в приміщеннях;*
- *забезпечувати охорону приміщень;*
- *керувати побутовими електроприладами.*

І це лише мала частина того, що може система «розумний» будинок.

Загалом, концепція системи має на меті створення зони комфорту в місці, де людина проводить більшу частину життя. В основу системи покладено ієрархічну структуру, головною частиною якої є контролер, який відповідає за налагодження роботи пристроїв системи, забезпечення зв'язку та обміну даними між ними, а також стежить за виконанням складних алгоритмів. Для контролю ситуації в приміщенні в підпорядкуванні контролера є різні датчики, що сигналізують про зміни параметрів середовища: збільшення рівня задимленості, виникнення пожежі, або іншої проблеми, що робить приміщення небезпечним для життя та здоров'я людини.

В загальному випадку виділяють наступні частини розумного будинку:

- *виконавці – пристрої, що відповідають за виконання команд, сформованих на основі даних датчиків;*
- *пристрої управління – пристрої, що надають можливість просто та зручно керувати розумним будинком;*
- *інтегроване обладнання та веб-сервіси – такі пристрої як відеокамери, периферія, пульти дистанційного управління, що полегшують процес взаємодії користувача з пристроями розумного будинку.*

Кожен пристрій налаштовано як на передавання, так і на приймання сигналу, тобто якщо якесь обладнання вийде з ладу, то його функції візьме на себе інший прилад системи, який здатен їх виконувати.

Сучасні технології дозволяють збирати систему розумного будинку покомпонентно, тобто обирати лише ті прилади, функції яких вам потрібні.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Сьогодні навіть речі, які раніше були лише предметами інтер'єру можуть виконувати ряд корисних функцій.

1. 2 Загальна структура (архітектура) існуючих систем Smart Home.

Система «Smart Home» складається з трьох типів пристроїв:

- *контролер — пристрій керування, що з'єднує всі пристрої в єдину систему;*
- *датчики — пристрої системи, які отримують інформацію про зовнішні умови та передають їх на опрацювання контролеру;*
- *актуатори — пристрої системи, які є безпосередніми виконавцями команд контролера, а також найчисленніша група пристроїв, до якої належать розумні побутові прилади, розетки, вимикачі і таке інше.*

У більшості випадків контролер з'єднується та обмінюється даними з іншими пристроями системи за допомогою радіосигналу, найпоширенішими стандартами якого є Z-Wave та ZigBee, технології Wi-Fi та Bluetooth.

Майже всі ці стандарти підтримують шифрування даних за допомогою стандарту AES-128, а в технології Wi-Fi застосовується шифрування WPA, WPA2, WPA3 або WEP.

Концепція «Smart Home» містить в собі наступні складові:

- *Створення вбудованої системи керування будинком, тобто системи, в якій забезпечується робота всіх інженерних систем будинку, таких як опалення, освітлення та інших, як комплексу.*
- *Надання функцій контролю та прийняття рішень підсистемам комплексу управління будинком, в які закладено логіку будинку, тобто алгоритм дій при зміні параметрів датчиків та інші події.*
- *Забезпечення роботи окремих підсистем комплексу у разі виходу з ладу окремих підсистем, або ж керуючого пристрою.*

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Використання загальних стандартів при побудові комплексу, що дозволить мінімізувати вартість його обслуговування та модернізації.
- Створення механізму негайного відключення та, при необхідності, можливості ручного керування людиною будь-якою підсистемою комплексу.
- Наявність в будинку комунікаційного середовища для розширення комплексу шляхом підключення нових пристроїв та модулів, а також наявність вибору різних типів фізичних каналів в цьому середовищі: радіоканали, лінії слабого струму і т. д.

1.3 Аналіз технічних рішень.

Сучасний ринок пропонує великий вибір пристроїв для автоматизації житла, але далі ми ознайомимося з найпопулярнішими з них.

Першою в списку буде українська система автоматизації Ажах. Дана система автоматизації в повній мірі справляється одразу з двома важливими задачами:

- забезпечує комфорт та зручність в управлінні приміщенням;
- гарантує безпеку житла в повній мірі, контролюючи межі об'єкта на предмет взлому, а також інших можливих загроз для будинку.

Пристрої даної системи працюють на надійно зашифрованому та захищеному двосторонньому радіозв'язку Jeweller власної розробки, має повну автономність від електромережі завдяки резервному джерелу живлення – хабу, характеризується стильним дизайном всіх своїх пристроїв.

Дана система має наступні переваги:

- простий монтаж;
- безпроводний канал зв'язку між елементами системи;
- обширна зона дії сигналу;

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

- наявність захисту від зняття будь-якого датчика;
- автономна робота хабу від акумулятора;
- підключення до 100 пристроїв;
- невелика вартість комплекту (від 200\$).

А також певні недоліки:

- функціонування тільки з роботою центрального контролера, тобто відсутність автономності датчиків;
- управління тільки через телефон.

Підводячи підсумки можна сказати, що система Ajax багатофункціональна, надійна, зручна та компактна. Установка такого комплексу спрощена до мінімуму та доволі доступна навіть для технічно не підготованих користувачів. Також важливою перевагою є доволі демократична ціна на девайс з урахуванням його функціоналу.

Наступною системою, яку ми розглянемо буде система «Broadlink». Обладнання розумного будинку BroadLink — набір сучасних цифрових пристроїв, створених для раціонального управління побутовою технікою, а також освітленням, енергією, безпекою та іншими системами будинку. Кожен елемент такого комплексу може працювати незалежно або взаємодіяти один з одним.

Переваги:

- швидке встановлення, під'єднання та налаштування;
- широкий спектр датчиків (вологість, температура, освітлення, шум, забруднення повітря);
- може легко додавати і видаляти різні пристрої;
- працює без центрального хаба;

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

- *бездротова взаємодія між пристроями;*
- *контролюється Wi-Fi через Інтернет з будь-якого місця в світі;*
- *демократична вартість обладнання (від \$ 200).*

Недоліки:

- *малий діапазон сигналу (до 50 м);*
- *відсутнє джерело резервного живлення;*

У рейтингу систем розумного будинку BroadLink займає міцне друге місце завдяки широкій функціональності, якісному програмному забезпеченню, простоті установки і використання, а також доступній вартості. Такий комплекс не потребує центрального контролера, оскільки всі його пристрої, хоч і взаємопов'язані, але можуть працювати повністю автономно. Робота побутової техніки в будинку налаштовується за сценаріями в додатку на смартфоні.

На третьому місці розташувалася система розумного будинку від китайської компанії Xiaomi. Розумний будинок від даної компанії відноситься до бюджетного класу обладнання, яке дозволяє зробити управління різними пристроями та побутовою технікою в домі максимально простим та зручним.

Переваги:

- *повна автономність пристроїв;*
- *можливість масштабування;*
- *бездротовий протокол ZigBee;*
- *зручне керування за допомогою смартфона через Wi-Fi;*
- *наявність сценаріїв;*
- *компактність та стильний дизайн;*
- *низька вартість базового комплексу(всього 90\$);*

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Недоліки:

- *дуже мала зона дії сигналу(до 10 метрів);*
- *скромний набір сенсорів та виконуючих пристроїв в базовому наборі;*
- *на різні датчики потрібне своє положення;*
- *відсутність резервного живлення хабу.*

Таким чином комплект Xiaomi є гарною стартовою платформою для підключення інших сенсорів та пристроїв, в тому числі і сторонніх виробників. Її елементи працюють як самостійно, так і в якості єдиного цілого. З їх допомогою можна створити досить функціональну систему контролю житлового простору, включаючи також забезпечення безпеки.

Проаналізувавши системи «розумного» дому від різних виробників можна встановити те, що вони працюють за схожими алгоритмами. Тобто в них існує центральний хаб, або ж контролер, до якого під'єднуються датчики та актуатори та через який здійснюється контроль та управління системою. Загальна схема такого підходу до побудови системи наведена на рисунку 1. 1.

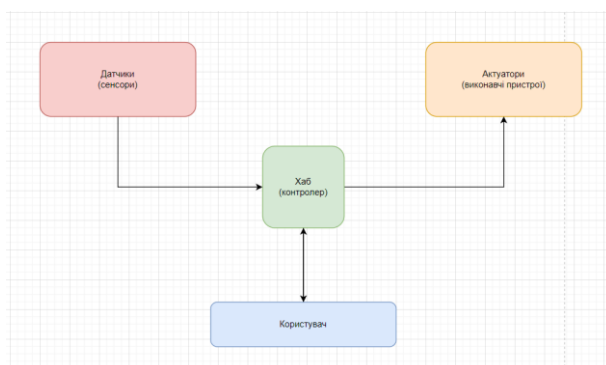


Рисунок 1. 1 – Реалізація системи «розумного» будинку із використанням центрального хаба

Проте існує й інший підхід реалізації такої системи. В ньому відсутній хаб та все керування відбувається через хмару та домашній Wi-Fi маршрутизатор. За допомогою додатку на смартфоні здійснюється додавання нових пристроїв,

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

налаштування та моніторинг даних. Загальна схема такого підходу наведена на рисунку 1. 2.

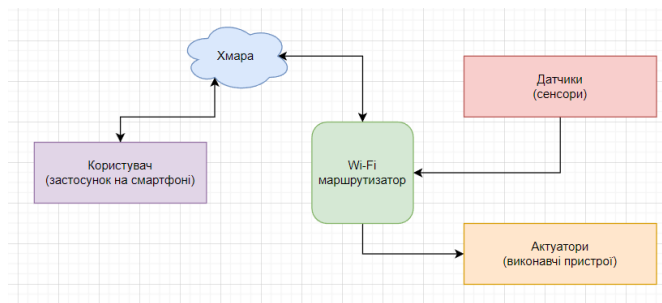


Рисунок 1. 2 – Реалізація системи «розумного» будинку без використання центрального хаба

1. 4 Висновки до розділу 1

В першому розділі було розглянуто такі питання:

- історію створення та концепцію систем «Smart Home»;
- архітектуру та основні компоненти систем «Smart Home»;
- аналіз існуючих рішень та підходів до побудови систем, наведено їх переваги та недоліки.

Аналіз існуючих рішень та підходів до побудови системи «розумного» будинку показав, що досить широко використовуються у технічних рішеннях різних виробників системи з використанням центрального хаба як керуючого пристрою, проте системи без його використання також реалізуються досить часто. Кожна з цих систем має свої переваги та недоліки, які було описано в даному розділі і враховано в процесі проектування нашої системи «Smart Home».

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ «SMART HOME»

2. 1 Розробка структури системи

Після аналізу існуючих проектів «розумного» будинку можна перейти до розробки структури нашої системи. Для початку оберемо який підхід буде використано в нашій системі – з хабом, чи без нього.

Проаналізувавши переваги та недоліки кожного з цих підходів було прийнято рішення спроектувати власний підхід до вирішення даного питання шляхом модернізації підходу без використання хаба.

Оскільки центрального хаба, як керуючого пристрою в нашій системі не буде, роль пристрою для зв'язку між елементами системи буде виконувати локальний сервер Blynk. За його допомогою буде відбуватися обмін даними між мікроконтролерами в підсистемах, а також буде налагоджуватися зв'язок з мобільним додатком, через який можна буде здійснювати керування системою та моніторинг її елементів. Відмінністю від звичайного підходу реалізації системи без хаба є те, що управління актуаторами здійснюється всередині самих підсистем, а не на рівні мобільного додатку, або сервера. Загальна схема системи, на якій зображено описаний вище алгоритм роботи, наведена на рисунку 2. 1.

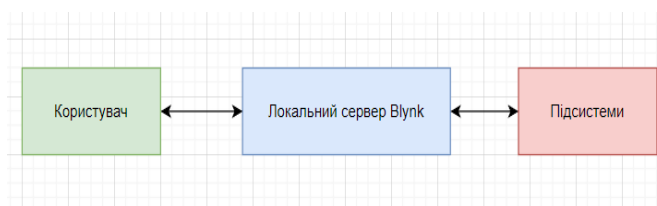


Рисунок 2. 1 – Загальна схема системи

Розібравшись із підходом до побудови системи необхідно розробити алгоритм її роботи. Для початку здійснюється запуск локального сервера, який слугуватиме як елемент зв'язку між пристроями в системі, після чого всі пристрої за допомогою системи Blynk під'єднуються до запущеного сервера за його IP-адресою.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Далі починається робота системи. Щосекунди відповідні мікроконтролери здійснюють збір даних з датчиків, відправляють їх на сервер, оброблюють та виконують необхідні дії на основі даних з датчиків та знову надсилають оновлення на сервер для відображення їх в мобільному додатку. Алгоритм, описаний вище, зображений на рисунку 2. 2.

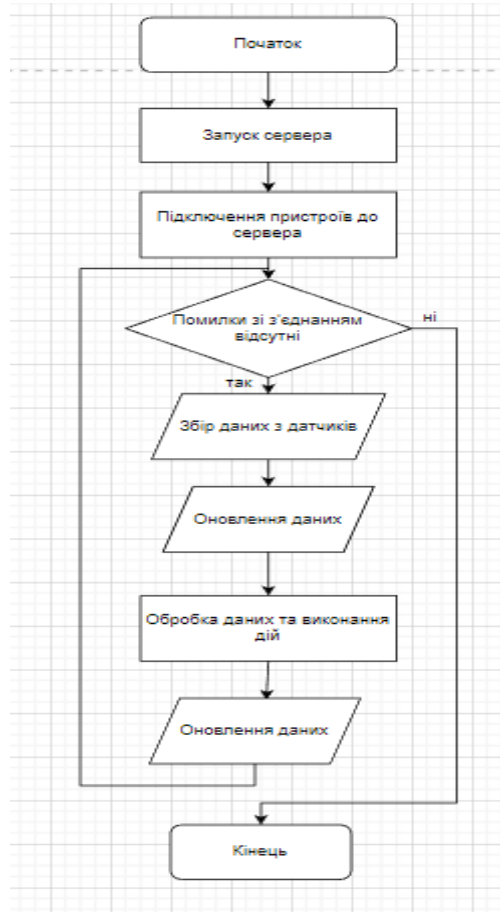


Рисунок 2. 2 – Алгоритм роботи системи

Після того, як систему було спроектовано та розібрано алгоритм її роботи, необхідно визначитися з яких підсистем вона складатиметься. Наша система «Smart Home» складатиметься з таких частин:

- *локальний сервер системи blynk;*
- *підсистема регулювання температури в будинку;*
- *підсистема безпеки;*
- *підсистема контролю доступу;*

- підсистема контролю якості повітря;
- підсистема контролю водопостачання;
- підсистема моніторингу пожежі;
- підсистема пожежогасіння;
- підсистема керування освітленням.

Вся система працюватиме на базі локального сервера системи Blynk, який буде забезпечувати користувачу доступ до інформації та керування підсистемами, а також забезпечить зв'язок між ними. Загальна структура системи представлена на рисунку 2. 3.

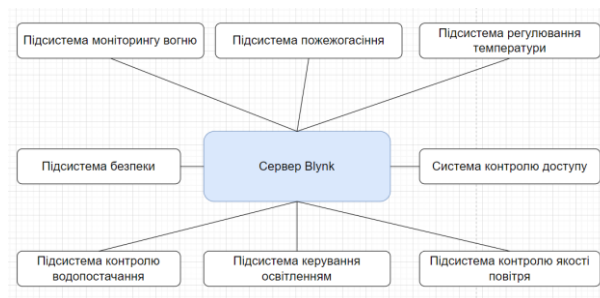


Рисунок 2. 3 – Загальна структура системи

2. 2 Схемні рішення: структурні та функціональні схеми підсистем

Розглянемо детальніше структурні та функціональні схеми кожної підсистеми. Першою розглянемо підсистему регулювання температури.

Дана підсистема складатиметься з двох мікроконтролерів, які здійснюватимуть обмін даними через локальний сервер за допомогою віртуального мосту. Загальна схема підсистеми представлена на рисунку 2. 4.

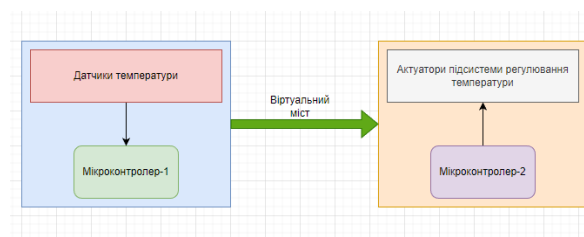


Рисунок 2. 4 – Загальна схема підсистеми регулювання температури

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Логіка роботи даної підсистеми наступна: перший мікроконтролер збирає дані з датчиків температури з кімнат та за допомогою віртуального мосту передає їх на другий мікроконтролер, який обробляє ці дані та відповідно до умов подає сигнали на блок актуаторів системи.

Блок актуаторів підсистеми, схема якого представлена на рисунку 2. 5, складається з 5 модулів подвійних реле, до яких під'єднані живлення та логічні виходи з мікроконтролера, нагрівальні та охолоджувальні пристрої в кімнатах.

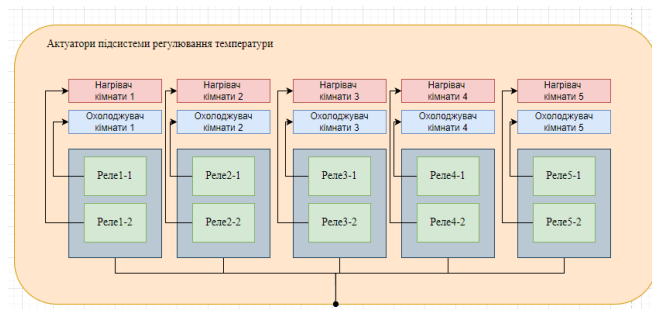


Рисунок 2. 5 – Блок актуаторів підсистеми регулювання температури

Блок датчиків температури, схема якого зображена на рисунку 2. 6, складається з 5 датчиків температури, які знаходяться в кімнатах та щосекунди передають свої дані на перший мікроконтролер, який отримує їх, обробляє та відправляє на мікроконтролер.

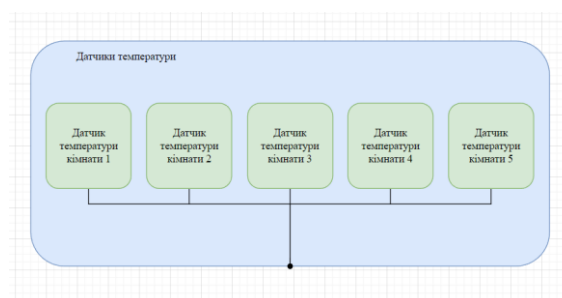


Рисунок 2. 6 – Блок датчиків температури підсистеми регулювання температури

Розглянемо схеми підключення пристроїв до мікроконтролерів підсистеми регулювання температури.

До першого мікроконтролера буде підключено 5 датчиків температури (рисунок 2. 7). Всі вони контактами живлення будуть під'єднані до логічних цифрових виходів D1-D5 відповідно, через які дані з датчиків будуть відправлятися на мікроконтролер та оброблятися ним.

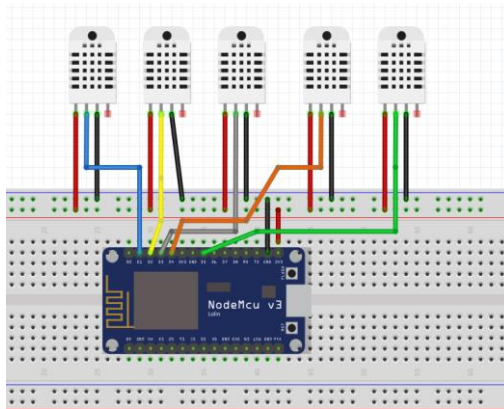


Рисунок 2. 7 – Схема підключення датчиків до першого мікроконтролера підсистеми регулювання температури

До другого мікроконтролера буде підключено 5 модулів двоканальних реле, які керуватимуться ним (рисунок 2. 8). В свою чергу до кожного такого модуля буде підключено нагрівальний та охолоджуючий елементи, які будуть підключені до напруги живлення 220 В через ці модулі. Живлення модулів реле відбуватиметься безпосередньо від мікроконтролера. Управління нагрівальними елементами буде здійснюватися виходами D0, D2, D4, D6, D8, а охолоджувальними – D1, D3, D5, D7, D9 відповідно.

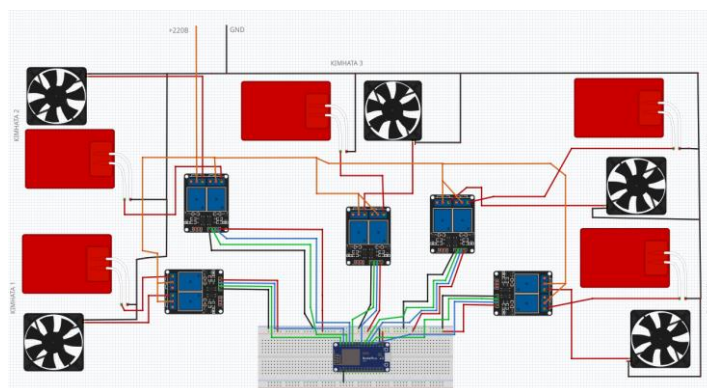


Рисунок 2. 8 – Схема підключення актуаторів до другого мікроконтролера підсистеми регулювання температури

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступною розглянемо підсистему контролю якості повітря, яка буде контролювати рівень шкідливих газів в повітрі та реагуватиме на зміну їх концентрації. Схема даної підсистеми, яка складатиметься з одного мікроконтролера представлена на рисунку 2. 9.

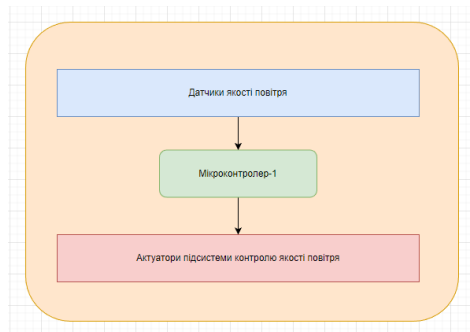


Рисунок 2. 9 – Загальна схема підсистеми контролю якості повітря

Логіка роботи даної підсистеми наступна: мікроконтролер отримує дані з датчиків якості повітря, оброблює їх та відповідно до змін посилає команди на актуатори даної підсистеми.

Блок актуаторів, схема якого представлена на рисунку 2. 10, складається з 4 реле, до яких під'єднані живлення та логічні виходи з мікроконтролера, а також кулери вентиляційних систем у кімнатах.

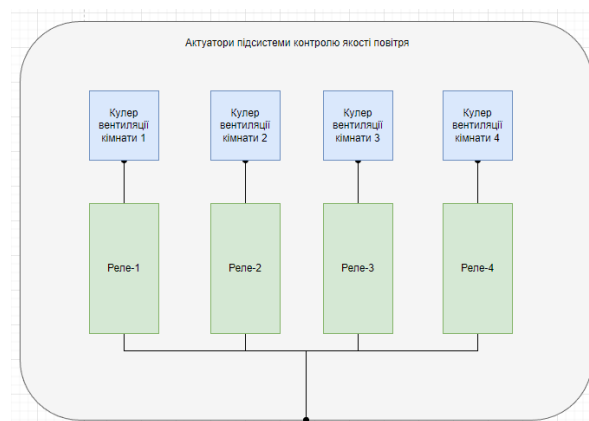


Рисунок 2. 10 – Блок актуаторів підсистеми контролю якості повітря

Далі детально розглянемо блок датчиків якості повітря підсистеми, схема якого представлена на рисунку 2. 11. Даний блок складається з 4 датчиків якості

повітря, які вимірюють значення концентрації шкідливих газів у повітрі та передають їх на мікроконтролер.

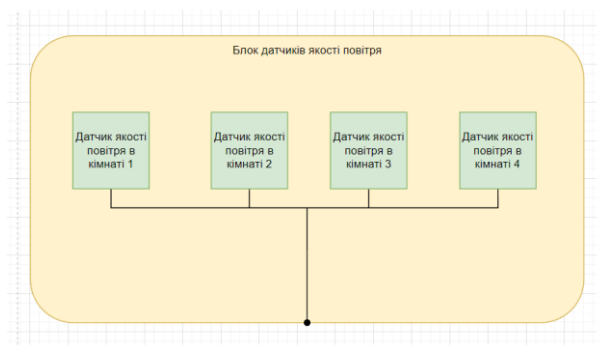


Рисунок 2. 11 – Блок датчиків якості повітря підсистеми контролю якості повітря

Тепер розглянемо схему підключення пристроїв до мікроконтролера. Дана схема наведена на рисунку 2. 12. Датчики якості повітря та реле будуть живитися безпосередньо від мікроконтролера, а кулери системи вентиляції – від джерела живлення напругою 12 В. Один з датчиків буде підключено до аналогового виходу А0, а решта три до виходів D0, D1 та D2 відповідно. Реле в свою чергу керуючими виходами будуть підключені до виходів D3, D4, D5 та D6 мікроконтролера та комутуватимуть підключення кулерів до живлення.

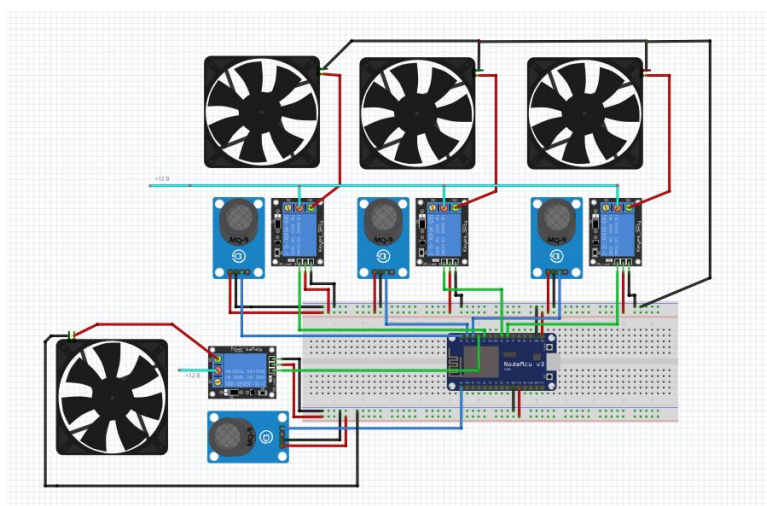


Рисунок 2. 12 – Схема підключення пристроїв до мікроконтролера підсистеми контролю якості повітря

Перейдемо до підсистеми моніторингу пожежі, схема якої наведена на рисунку 2. 13. Дана підсистема слідкуватиме за наявністю загорянь в кімнатах та складатиметься з одного мікроконтролера.

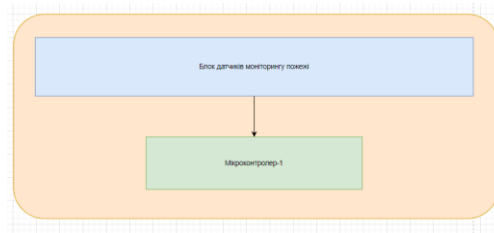


Рисунок 2. 13 – Загальна схема підсистеми моніторингу пожежі

Логіка роботи даної підсистеми проста: мікроконтролер збирає дані з датчиків моніторингу пожежі та за допомогою віртуального мосту відправляє сигнали до підсистеми пожежогасіння.

Блок датчиків моніторингу пожежі детально зображений на рисунку 2. 14. Він складається з шести датчиків вогню, які реагують на загоряння та передають ці дані на мікроконтролер.

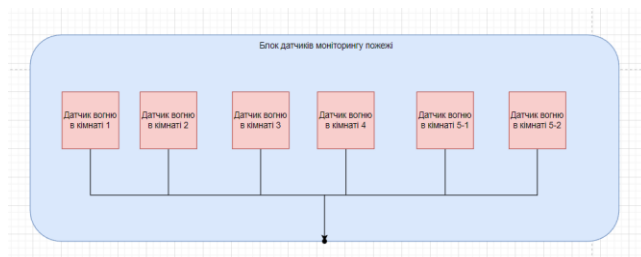


Рисунок 2. 14 – Блок датчиків моніторингу пожежі підсистеми моніторингу пожежі

Тепер розглянемо схему підключення пристроїв до мікроконтролера даної підсистеми, яка зображена на рисунку 2. 15. Всі датчики вогню будуть живитися від мікроконтролера та цифровими виходами під'єднані до виходів D0 - D5 відповідно.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

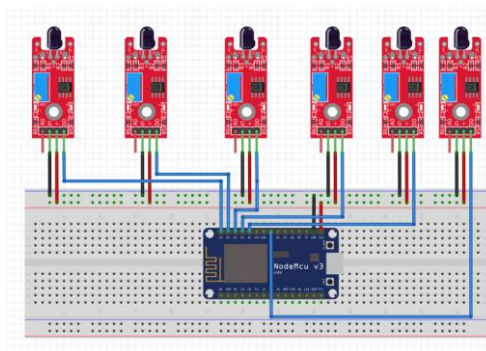


Рисунок 2. 15 – Схема підключення пристроїв до мікроконтролера підсистеми моніторингу пожежі

Наступною розглянемо підсистему пожежогасіння. Вона складається з одного мікроконтролера та блока актуаторів підсистеми. Схема підсистеми представлена на рисунку 2. 16.

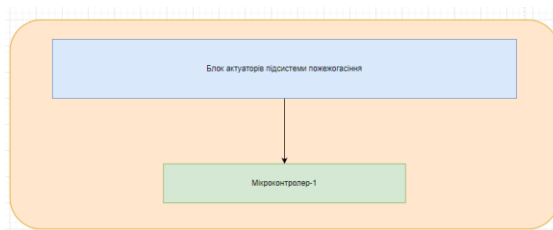


Рисунок 2. 16 – Загальна схема підсистеми пожежогасіння

Блок актуаторів складається з шести реле та шести соленоїдних клапанів, через які здійснюватиметься подача води на розбризкувачі в кімнатах. Схема даного блоку наведена на рисунку 2. 17.

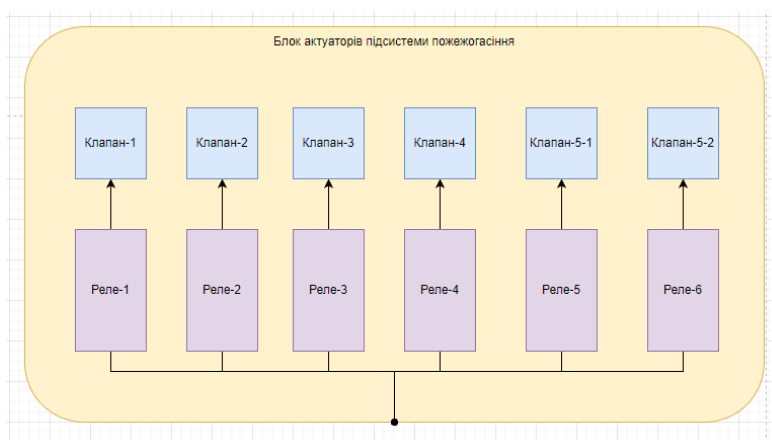


Рисунок 2. 17 – Блок актуаторів підсистеми пожежогасіння

Дана підсистема має наступну логіку роботи: мікроконтролер через віртуальний міст отримує дані з підсистеми моніторингу та відповідним чином реагує на них надсилаючи керуючі сигнали на реле, які в свою чергу відкриватимуть клапани подачі води.

Пристрої до мікроконтролера даної підсистеми будуть підключені за схемою, зображеною на рисунку 2. 18. Нормально закриті соленоїдні клапани підключені до живлення в 220 В через реле, які в свою чергу живлення отримують безпосередньо від мікроконтролера, а їх виходи керування підключені до виходів D0 – D5 відповідно.

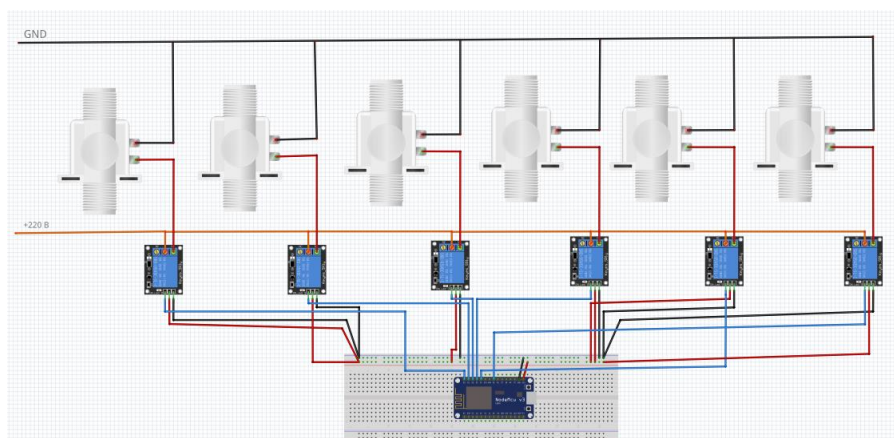


Рисунок 2. 18 – Схема підключення пристроїв до мікроконтролера підсистеми пожежогашіння

Далі розглянемо підсистему контролю водопостачання. Дана підсистема складається з одного мікроконтролера, блоку датчиків та блоку актуаторів. Загальна схема підсистеми наведена на рисунку 2. 19.

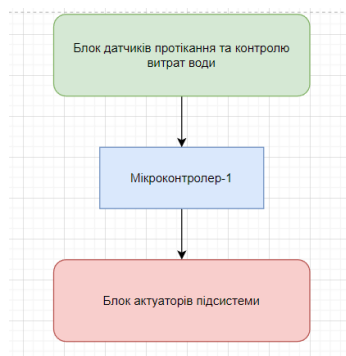


Рисунок 2. 19 – Загальна схема підсистеми контролю водопостачання

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Логіка роботи даної підсистеми наступна: блок датчиків передає інформацію про наявність протікань та кількість витраченої води на мікроконтролер, який обробляє цю інформацію і, за наявності протікання, передає команду на блок актуаторів, які закривають клапан та усувають протікання.

Блок датчиків складається з двох датчиків дощу, які реагують на краплі, та двох датчиків контролю витрат води. Схема даного блоку представлена на рисунку 2. 20.

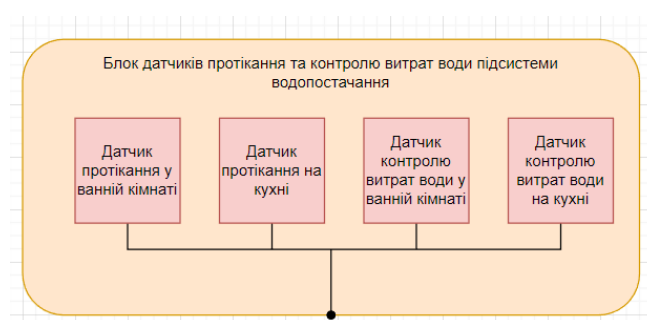


Рисунок 2. 20 – Блок датчиків підсистеми контролю водопостачання

Блок актуаторів містить в собі два реле, які керують двома нормально відкритими соленоїдними клапанами. Схема блоку наведена на рисунку 2. 21.

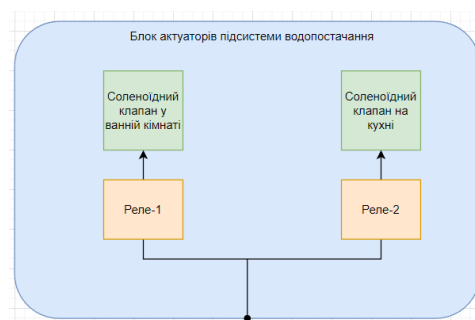


Рисунок 2. 21 – Блок актуаторів підсистеми водопостачання

Далі розглянемо схему підключення пристроїв до мікроконтролера даної підсистеми. Датчики дощу, контролю витрат води та реле живитимуться безпосередньо від мікроконтролера, а соленоїдні клапани – від 220 В через відповідні реле. Датчики дощу будуть підключені до цифрових виходів D0 та D1,

датчики контролю витрат води – до D2 та D3, а реле, які керують клапанами до виходів D4 та D5 відповідно. Схема підключення представлена на рисунку 2. 22.

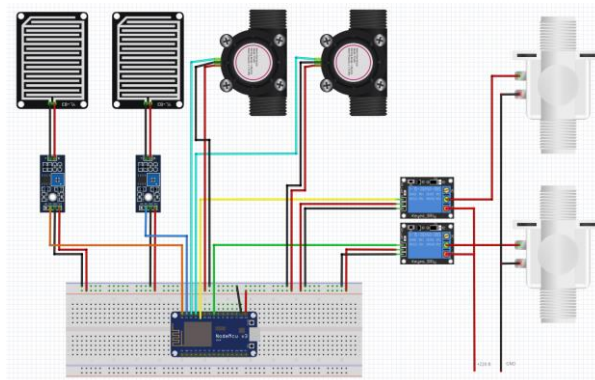


Рисунок 2. 22 – Схема підключення пристроїв до мікроконтролера підсистеми контролю водопостачання

Підсистема безпеки складатиметься з трьох мікроконтролерів, блоку датчиків руху та блоку датчиків відкриття дверей та вікон. Схема даної підсистеми наведена на рисунку 2. 23.

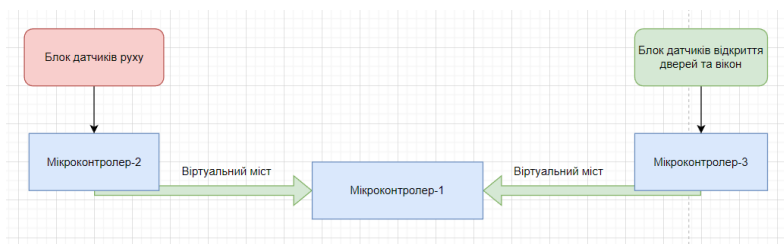


Рисунок 2. 23 – Загальна схема підсистеми безпеки

Дана підсистема працює за наступною логікою: мікроконтролери 2 та 3 збирають інформацію з датчиків руху та відкриття дверей і вікон відповідно та передають ці дані через віртуальні мости на мікроконтролер 1, який в свою чергу здійснює сповіщення господарів будинку про відповідні ситуації.

Блок датчиків руху складається із семи датчиків руху, які підключені до мікроконтролера 2 підсистеми безпеки. Схема блоку наведена на рисунку 2. 24.

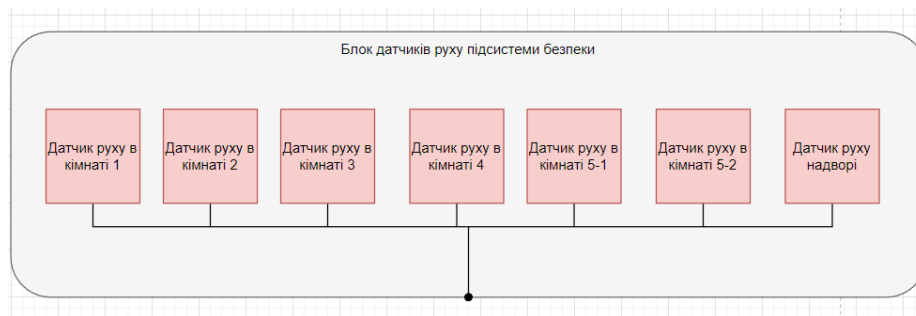


Рисунок 2. 24 – Блок датчиків руху підсистеми безпеки

Блок датчиків відкриття вікон та дверей, підключений до мікроконтролера 3 підсистеми безпеки, складається з п'яти датчиків відкриття вікон та дверей. Схема даного блоку наведена на рисунку 2. 25.



Рисунок 2. 25 – Блок датчиків відкриття вікон та дверей

Розглянемо схему підключення пристроїв до мікроконтролера 2. До цього мікроконтролера буде підключено 7 датчиків руху, які житимуться безпосередньо від мікроконтролера, а цифровими виходами будуть підключені до виходів D0 – D6 мікроконтролера відповідно. Схема підключення наведена на рисунку 2. 26.

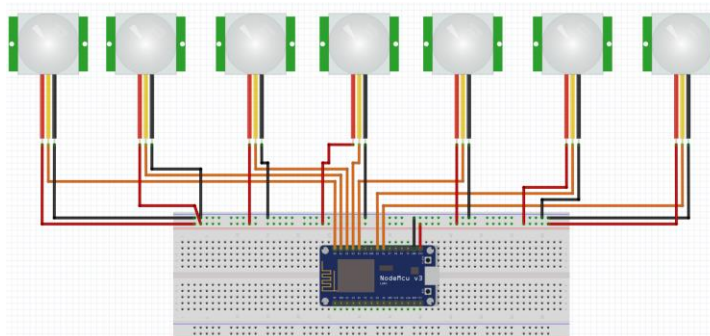


Рисунок 2. 26 – Схема підключення пристроїв до мікроконтролера 2 підсистеми безпеки

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

До мікроконтролера 3 буде підключено 5 датчиків відкриття вікон та дверей, які одним контактом підключені до цифрових виходів D0 – D4 мікроконтролера, а іншим – до землі. Схема підключення наведена на рисунку 2. 27.

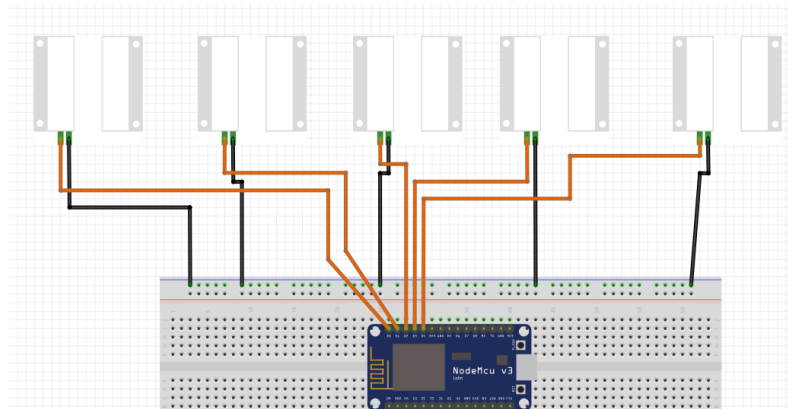


Рисунок 2. 27 – Схема підключення пристроїв до мікроконтролера 3 підсистеми безпеки

Підсистема керування освітленням складатиметься з одного мікроконтролера та лише блока актуаторів, оскільки дані з датчиків він отримуватиме з блоку датчиків руху підсистеми безпеки. Загальна схема підсистеми наведена на рисунку 2. 28.

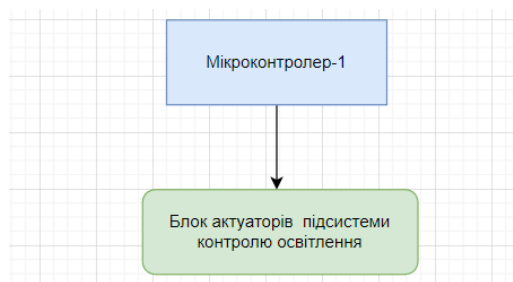


Рисунок 2. 28 – Загальна схема підсистеми керування освітленням

Логіка роботи підсистеми наступна: підсистема отримуватиме дані з датчиків руху підсистеми безпеки та за наявності руху вмикатиме освітлення у відповідній кімнаті, або частині будинку. Також система відповідатиме за ручне віддалене керування освітленням.

Блок актуаторів підсистеми складатиметься з шести реле, які керуватимуть освітленням у відповідних частинах будинку. Схема даного блоку наведена на рисунку 2. 29.

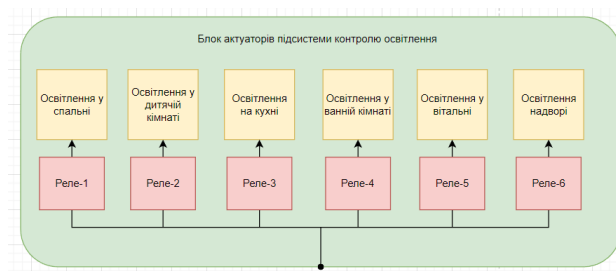


Рисунок 2. 29 – Блок актуаторів підсистеми керування освітленням

Схема підключення пристроїв до мікроконтролера підсистеми керування освітленням представлена на рисунку 2. 30. Живлення реле відбувається безпосередньо від мікроконтролера, а систем освітлення від мережі 220 В. Керуючими виходами реле підключені до виходів D0 – D5 мікроконтролера.

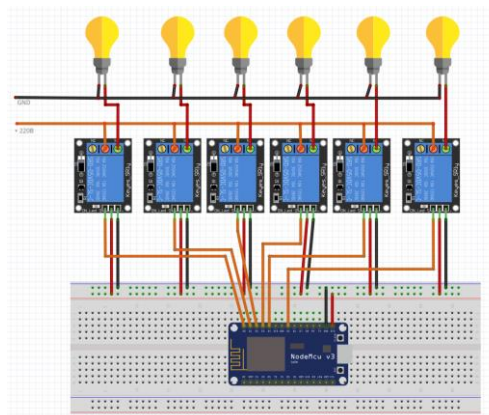


Рисунок 2. 30 – Схема підключення пристроїв до мікроконтролера підсистеми керування освітленням

Підсистема контролю доступу складається з одного мікроконтролера, блоку зчитувача електронного замка та кнопок керування ним, а також блоку актуаторів електронного замка. Загальна схема підсистеми представлена на рисунку 2. 31.

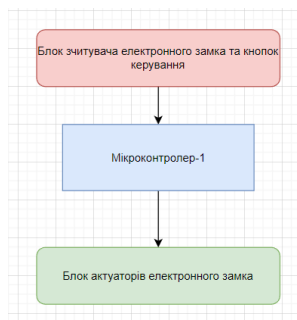


Рисунок 2. 31 – Загальна схема підсистеми контролю доступу

Логіка роботи цієї підсистеми наступна: до мікроконтролера через керуюче реле під'єднано нормально закритий соленоїдний замок та RFID-зчитувач. Якщо піднесена до зчитувача картка є в списку дозволених, то замок відкриється, в інакшому випадку – залишиться закритим. Також замок можна відкрити зсередини за допомогою відповідної кнопки.

Блок зчитувача та кнопок керування, зображений на малюнку 2. 32, складається з RFID-зчитувача та кнопки керування.

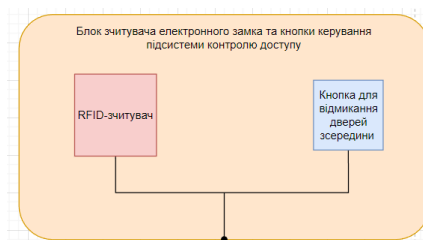


Рисунок 2. 32 – Блок зчитувача та кнопки керування підсистеми контролю доступу

Блок актуаторів складається із соленоїдного нормально закритого замка, підключеного до мікроконтролера через керуюче реле. Схема даного блоку представлена на рисунку 2. 33.

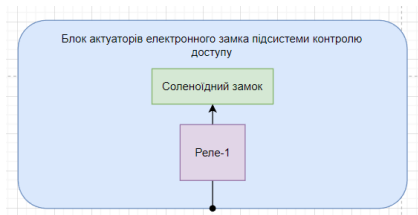


Рисунок 2. 33 – Блок актуаторів підсистеми контролю доступу

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

Зчитувач буде підключений відповідними своїми виходами до виходів D0, D5, D6, D7, D8 мікроконтролера, а також до контактів його живлення. Реле керування замком буде підключено до виходу D3, а кнопка керування до D1. Живлення замка відбуватиметься від джерела напругою 12 В. Схема підключення представлена на рисунку 2. 34.

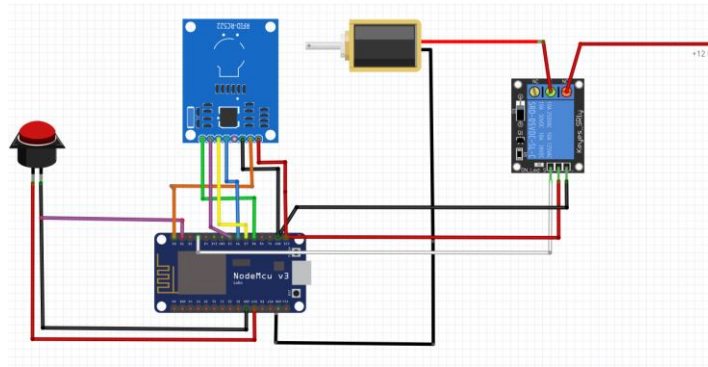


Рисунок 2. 34 – Схема підключення пристроїв до мікроконтролера підсистеми контролю доступу

2.3 Управління системою. Модель взаємодії підсистем в системі «Smart Home»

Після проектування та розробки системи необхідно розібратись яким чином буде здійснюватись управління нею.

Оскільки наша система зроблена на основі локального сервера системи Blynk, то керування нею буде здійснюватись за допомогою відповідного мобільного додатку. Додаток Blynk має великий асортимент віджетів та елементів моніторингу, дозволяє налаштувати інтерфейс дуже гнучко та інформативно, так як забажає користувач.

Далі розберемо як взаємодіятимуть різні підсистеми між собою в нашій системі. Модель взаємодії представлена на рисунку 2. 35.

Підсистема моніторингу вогню буде взаємодіяти з підсистемою пожежогасіння наступним чином: при виявленні пожежі підсистема передає дані на підсистему пожежогасіння, яка вмикає відповідні актуатори. Також при

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

виявленні пожежі дана підсистема передасть сигнал підсистемі контролю доступу яка відімкне двері.

Підсистема безпеки та контролю доступу взаємодіятимуть між собою при відкриванні та закриванні замка – під час цих дій вимикатиметься та, відповідно, вмикатиметься підсистема безпеки.

Підсистема безпеки в свою чергу взаємодіятиме з підсистемою контролю освітлення шляхом передачі на неї даних з датчиків руху.

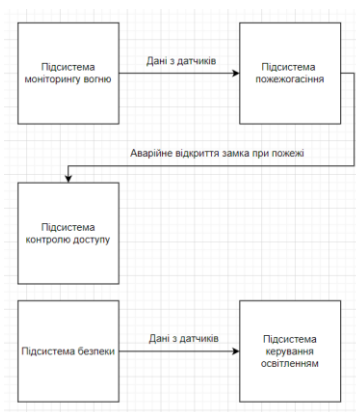


Рисунок 2.35 – Модель взаємодії підсистем між собою

2.4 Висновки до розділу 2

В другому розділі було спроектовано систему «Smart Home». Після проведення аналізу підходів до побудови систем було обрано підхід без використання центрального хаба в ролі керуючого пристрою для організації роботи системи, що дозволить розділити нашу систему на підсистеми, які взаємодіятимуть між собою за допомогою локального сервера.

Кожна підсистема керуватиметься відповідними їх мікроконтролерами, що дозволить розподілити навантаження на систему рівномірно. Також при проектуванні підсистем було реалізовано можливість використання компонентів одних підсистем іншими підсистемами. Прикладом того є використання датчиків руху, які належать до підсистеми безпеки, підсистемою контролю освітлення.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ СИСТЕМИ «SMART HOME»

3.1 Обґрунтування технічних рішень вибору компонентів забезпечення функціонування підсистем в системі «Smart Home»

Після того як систему та її підсистеми було спроектовано необхідно обрати компоненти для забезпечення їх повного функціонування.

Розпочнемо з вибору рішення для розміщення локального сервера. Це повинен бути комп'ютер, який матиме невеликі розміри, зможе підключатися до домашнього маршрутизатора по дротовій, або ж бездротовій мережі та матиме достатню обчислювальну потужність.

Під такий опис ідеально підходять одноплатні комп'ютери Raspberry Pi, а саме, в нашому випадку Raspberry Pi Compute Module 4, яка зображена на рисунку 3.1. Дана плата має досить невеликі розміри, необхідну для роботи нашої системи обчислювальну потужність та можливість підключення до маршрутизатора через FastEthernet, що дозволить мінімізувати затримки при передачі даних.



Рисунок 3.1 – Плата Raspberry Pi Compute Module 4

Наступними компонентами для вибору стануть мікроконтролери. Вони повинні мати достатню кількість контактів для підключення необхідних пристроїв, мати вбудований WiFi-адаптер, достатню обчислювальну потужність та малі розміри. Під такий запит підходить мікроконтролер ESP8266 від виробника Amіса, зображений на рисунку 3.2. Даний мікроконтролер має достатню кількість виходів, вбудований безпроводний мережний адаптер,

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

невеликі розміри та низьку вартість(близько 5\$), а також достатню для функціонування вузлів підсистем обчислювальну потужність.



Рисунок 3.2 – Мікроконтролер ESP8266

Далі розберемо елементи керування приладами, які присутні в більшості підсистем, наприклад, реле. Вони повинні бути високого рівня, тобто спрацьовувати при надходженні логічної 1 на вхід керування, та мати достатню напругу комутації. Для таких цілей підійде реле SRD-05, зображене на рисунку 3.3. Даний модуль реле високого рівня має необхідну нам напругу живлення в діапазоні від 3.3 до 5 В та достатню напругу комутації в 250 В при струмі в 10 А.



Рисунок 3.3 – Модуль реле SRD-05

Наступним елементом, який буде використовуватись відразу у декількох підсистемах є соленоїдні клапани.

Нормально відкриті клапани використовуються в підсистемі контролю водопостачання та мають бути надійні та стримувати доволі великий потік рідини при закриванні. Гарним варіантом буде електромагнітний клапан фірми Pollena моделі 2W-15K, зображений на рисунку 3.4. Він має латунний корпус,

діаметр ½ дюйма, що буде достатньо для нашої підсистеми та надійний клапан всередині механізму.



Рисунок 3.4 – Електромагнітний нормально відкритий клапан Pollena 2W-15K

Підсистема пожежогасіння теж містить у своєму складі соленоїдні клапани, проте нормально закриті. Ідеальним варіантом стане клапан фірми AquaWorld моделі KL-NX-0102, зображений на рисунку 3.5. Цей клапан також має металевий корпус, що дозволить збільшити його час служби та високий максимальний тиск в 10 бар, що буде достатнім тиском для підсистеми пожежогасіння.



Рисунок 3.5 – Електромагнітний нормально закритий клапан AquaWorld KL-NX-0102

Далі розберемо елементи кожної підсистеми окремо. Розпочнемо з підсистеми регулювання температури. Датчики температури даної підсистеми мають бути доволі точними, компактними, коректно працювати з обраними мікроконтролерами та мати невисоку вартість. В такому випадку гарним варіантом стане модуль датчика вологості та температури DHT11, зображений

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

на рисунку 3.6. Даний модуль має невеликі похибки вимірювань, невеликі розміри та підтримує потрібну нам напругу живлення 3.3 В. Частота опитування даного датчика становить 1 Гц, тобто дані оновлюються кожну секунду, що дозволить нашій підсистемі із мінімальним затримками реагувати на зміни температури.



Рисунок 3.6 – Модуль датчика температури DHT11

В ролі нагрівальних елементів підсистеми мають бути пристрої, якими можна було б керувати за допомогою реле. Гарним варіантом такого пристрою буде керамічний обігрівач Emby СН-300(рисунок 3.7), який має потужність 300 Вт та чудово справляється з швидким збільшенням температури в кімнаті.



Рисунок 3.7 – Керамічний обігрівач Emby СН-300

В ролі охолоджувальних елементів мають бути компактні, досить потужні охолоджувачі повітря, керувати якими можна теж за допомогою реле, з високою якістю та невисокою ціною. Під ці параметри підходить охолоджувач повітря Arctic Air (рисунок 3.8). Він має доволі велику потужність в 10 Вт, невеликі розміри та резервуар для води, який дозволить не лише понизити температуру в кімнаті, а й підвищити вологість в ній.

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				39
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.8 – Охолоджувач повітря Arctic Air

Наступними підберемо елементи підсистеми контролю якості повітря. Дана підсистема містить в своєму складі датчики якості повітря та кулери, які розміщуються в системі вентиляції.

В якості датчиків якості повітря мають бути точні та надійні пристрої, які дозволятимуть отримувати не лише аналоговий, а й цифровий сигнал, що полегшить його обробку мікроконтролером. Гарним варіантом буде датчик MQ-9 (рисунок 3.9). Цей пристрій стабільний, з довгим терміном служби, з простою схемою підключення та малим часом відгуку, а також має потенціометр для налаштування чутливості датчика.



Рисунок 3.9 – Датчик якості повітря MQ-9

В якості кулерів мають бути швидкі та надійні кулери з невеликими розмірами та живленням від напруги 12 В. Під такі параметри підходить вентилятор MX-9025, зображений на рисунку 3.10. Він має невеликі розміри, що дозволить встановити його майже у всі вентиляційні отвори, високу швидкість обертання та великий термін служби.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				40
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.10 – Вентилятор MX-9025

Підсистема моніторингу пожежі складається з датчиків моніторингу вогню. Ці датчики мають бути достатньо чутливі, не реагувати на інфрачервоне випромінювання та денне світло, а лише на вогонь, а також мати досить великий радіус дії. Датчик KY-026, зображений на рисунку 3.11, має доволі точне налаштування довжини хвилі, на яку реагує датчик, що дозволяє йому реагувати лише на полум'я на відстані до 1.5 метра з кутом спрацювання в 60 градусів.

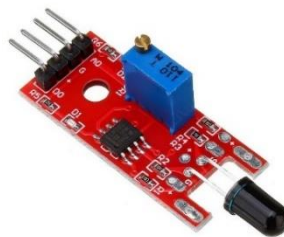


Рисунок 3.11 – Датчик вогню KY-026

Підсистема безпеки складається з блоку датчиків руху та блоку датчиків відкриття дверей та вікон.

Датчики руху в даній підсистемі мають бути надійними, з достатнім кутом спрацювання та мати можливість налаштування чутливості та часу, протягом якого датчик буде видавати на своєму виході логічну одиницю. Під ці параметри підходить інфрачервоний датчик руху HC-SR501, зображений на рисунку 3.12. Він має досить велику дальність спрацювання, широкий кут огляду та можливості налаштування чутливості та часу спрацювання.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				41
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.12 – Датчик руху HC-SR501

Датчики відкриття вікон та дверей повинні бути простими та надійними, безпомилково відпрацьовувати будь-які дії з дверима, або ж вікнами. Для таких задач чудово підійде датчик MC-38 (рисунок 3.13), який має невеликі розміри, досить чутливий та малу відстань спрацювання.



Рисунок 3.13 – Датчик відкриття вікон та дверей MC-38

В підсистемі контролю водопостачання для виявлення протікань використовуватимуться датчики дощу, а для контролю кількості витраченої води – датчик витрат води.

Датчик дощу має надійно і швидко реагувати на скопичення крапель води при протіканні. Для цих задач підійде датчик дощу DAT137 (рисунок 3.14), який в своєму складі має плату з підсилювачем, який обробляє сигнал та видає його в аналоговому та цифровому вигляді. Чутливість даного датчика регулюється підлаштувальним резистором.



Рисунок 3.14 – Датчик дощу DAT137

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

Контроль кількості води буде здійснюватись датчиками витрат води YF-S201 (рисунок 3.15), який працює на ефекті Холла, тобто кожен оберт ротора всередині датчика фіксується за допомогою датчика Холла і передається відповідний сигнал на вихід.



Рисунок 3.15 – Датчик витрат води YF-S201

Підсистема контролю доступу складатиметься з RFID-зчитувача та соленоїдного замка.

В якості зчитувача було обрано RFID модуль RC-522 (рисунок 3.16), який працює на частоті 13.56 МГц по SPI інтерфейсу, що робить його використання досить простим та зрозумілим.



Рисунок 3.15 – RFID модуль RC-522

Соленоїдний замок було обрано від компанії Sparkfun (рисунок 3.16). Він є нормально закритим та живиться від джерела живлення напругою 12 В.



Рисунок 3.16 – Соленоїдний замок від Sparkfun

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Розробка програмної частини системи. Інтерфейс користувача

Підібравши необхідні елементи підсистем та з'єднавши їх за складеними схемами можна перейти до розробки програмного забезпечення мікроконтролерів.

Підсистема регулювання температури складається з двох мікроконтролерів. Алгоритм роботи першого мікроконтролера представлений на рисунку 3. 17.



Рисунок 3.17 – Алгоритм роботи першого мікроконтролера підсистеми регулювання температури

Розберемо детальніше цей алгоритм. На початку своєї роботи мікроконтролер виконує ініціалізацію датчиків, тобто встановлення режиму введення на контактах, до яких підключені датчики, а також створення відповідних об'єктів цих датчиків для подальшої роботи з ними.

Далі виконується створення таймера для періодичної обробки даних, встановлюється з'єднання з сервером та відбувається запуск сервісу Blynk, а також створеного таймера.

Якщо при виконанні програми не виникло жодних критичних помилок, то з періодичністю, вказаною раніше при створенні таймера, відбуватимуться наступні дії: зчитування та відправка, за допомогою віртуального мосту між мікроконтролерами, даних з датчиків температури на другий мікроконтролер підсистеми, який здійснюватиме їх обробку. Фрагмент коду, який відповідає за збір та відправку даних з датчиків наведений на рисунку 3.18.

```
void sendDHTData()
{
    for(int i = 0; i < (int)(sizeof(dhtArray)/sizeof(dhtArray[0])); i++){
        float value = (int)dhtArray[i].readTemperature();
        bridge.virtualWrite(dhtVirtual[i], value);
    }
}
```

Рисунок 3.18 – Функція збору та відправки даних з датчиків

Даний фрагмент виконуватиметься циклічно до тих пір, коли його виконання не перерветься критичною помилкою.

Далі розглянемо алгоритм роботи та відповідне програмне забезпечення другого мікроконтролера підсистеми регулювання температури, рис. 3.19.

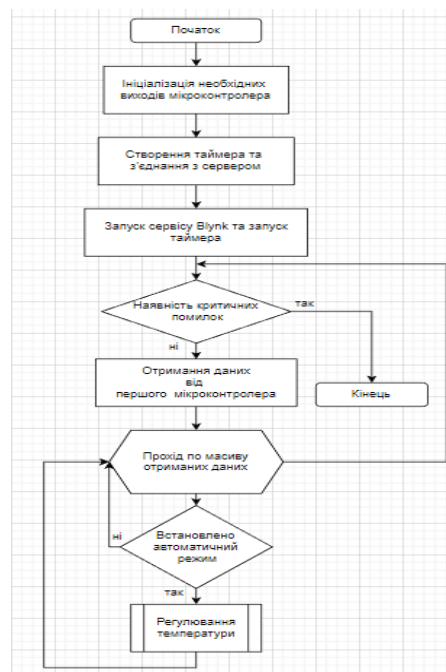


Рисунок 3.19 – Алгоритм роботи другого мікроконтролера підсистеми регулювання температури

Аналогічного з першим мікроконтролером, на початку відбувається ініціалізація потрібних виходів мікроконтролера, створення таймера та з'єднання з сервером.

Якщо критичних помилок не виявлено, то здійснюється прохід по масиву отриманих з датчиків значень. Далі здійснюється перевірка на те, чи встановлено користувачем автоматичний режим регулювання температури: якщо так, то відбуватиметься регулювання температури функцією, представленою на рисунку 3.20, якщо ж ні, то програма перейде на наступну ітерацію циклу.

```
void autoLevel(int room)
{
    if (temps[room] < tempLevels[room])
    {
        heat(room);
    }
    else if (temps[room] > tempLevels[room])
    {
        cool(room);
    }
    else
        offAtRoom(room);
}
```

Рисунок 3.20 – Функція автоматичного регулювання температури

Повний код програмного забезпечення мікроконтролерів підсистеми регулювання температури наведений в додатках А.1 та А.2.

Керування підсистемою контролю якості повітря здійснюється лише одним мікроконтролером, алгоритм роботи якого наведений на рисунку 3.21.

		Андрушенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

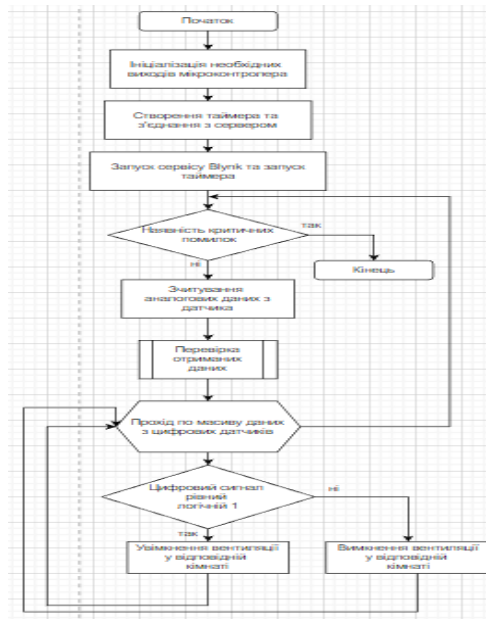


Рисунок 3.21 – Алгоритм роботи мікроконтролера підсистеми контролю якості повітря

Після ініціалізації необхідних виходів, створення таймера та з'єднання з сервером за відсутності критичних помилок відбувається зчитування аналогових даних з датчика якості повітря, після чого відбувається їх перевірка. Фрагмент коду, який відповідає за перевірку наведений на рисунку 3.22.

```
void checkKitchen(int value)
{
    if (value > CO2_WARNING_LEVEL)
    {
        Blynk.virtualWrite(V4, 1);
        Blynk.notify("Витік газу на кухні!!!");
    }
    else
    {
        Blynk.virtualWrite(V4, 0);
    }
}
```

Рисунок 3.22 – Функція перевірки даних з датчика

Після виконання перевірки аналогових даних відбувається циклічна перевірка цифрових даних з датчиків якості повітря, що лишилися. Якщо на виході даних датчиків з'являється логічна одиниця, то у відповідній кімнаті увімкнеться система вентиляції. Фрагмент коду, який відповідає за дану перевірку наведений на рисунку 3.23.

```

for (int i = 0; i < 3; i++)
{
    data[i] = digitalRead(rooms[i]);
    Blynk.virtualWrite(virtualRooms[i], data[i]);
    if (data[i] == 1)
    {
        Blynk.virtualWrite(leds[i], 1);
        digitalWrite(fans[i], HIGH);
    }
    else
    {
        Blynk.virtualWrite(leds[i], 0);
        digitalWrite(fans[i], LOW);
    }
}

```

Рисунок 3.23 – Фрагмент коду, який відповідає за перевірку цифрових даних з датчиків якості повітря

Виконання програми відбуватиметься циклічно до виникнення критичної помилки, яка унеможливить продовження функціонування. Повний код програмного забезпечення мікроконтролера підсистеми контролю якості повітря наведений у додатку Б.

Підсистема моніторингу пожежі також містить у своєму складі лише один мікроконтролер. Алгоритм його роботи наведений на рисунку 3.24.

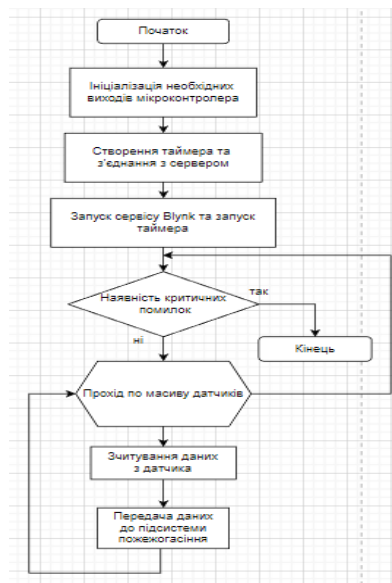


Рисунок 3.24 – Алгоритм роботи мікроконтролера підсистеми моніторингу пожежі

Після ініціалізації виходів, створення таймера, з'єднання з сервером та за відсутності критичних помилок відбувається прохід по масиву датчиків, під час якого здійснюється зчитування даних та передача їх на мікроконтролер

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

підсистеми пожежогасіння. Фрагмент коду, який відповідає за дану процедуру наведений на рисунку 3.25.

```
for (int i = 0; i < 6; i++)
{
    int tmp = digitalRead(fireDetectorPins[i]);
    if (tmp == 1)
    {
        firefight.virtualWrite(virtualFireDetectors[i], 1);
        fireDetectorLeds[i].on();
    }
    else
    {
        firefight.virtualWrite(virtualFireDetectors[i], 0);
        fireDetectorLeds[i].off();
    }
}
```

Рисунок 3.25 – Фрагмент коду, що відповідає за зчитування даних з датчиків вогню та їх передачу до підсистеми пожежогасіння

Повний код програмного забезпечення даного мікроконтролера наведений у додатку В.

Підсистема пожежогасіння складається з одного мікроконтролера, алгоритм його роботи представлений на рисунку 3.26.

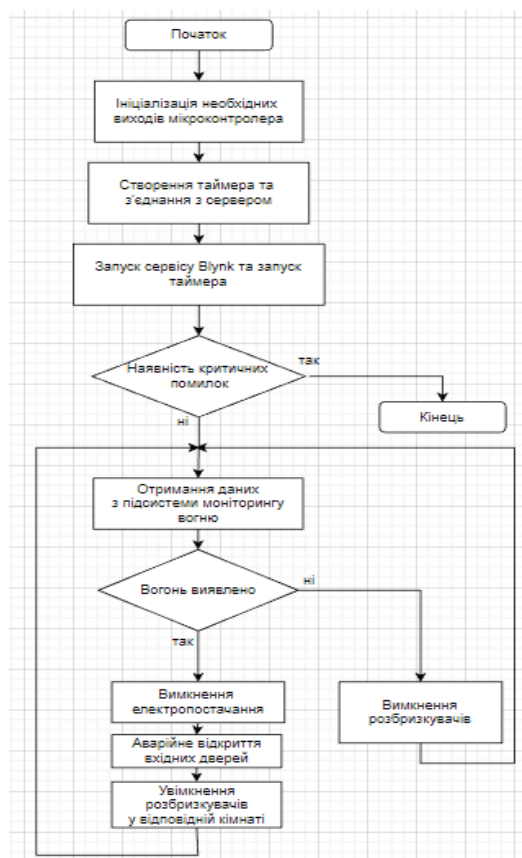


Рисунок 3.26 – Алгоритм роботи мікроконтролера підсистеми пожежогасіння

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

Після ініціалізації необхідних виходів, створення таймера та з'єднання з сервером за відсутності критичних помилок відбувається отримання даних від підсистеми моніторингу вогню та обробка цих даних.

Якщо у деякій кімнаті буде виявлено вогонь, то у всьому будинку буде вимкнено електропостачання, здійснено аварійне відкриття замка входних дверей, а також у відповідній кімнаті увімкнеться розбризкувач води, який гаситиме пожежу. У інакшому випадку розбризкувач буде вимкнено. Фрагмент коду, який відповідає за ввімкнення та вимкнення розбризкувачів, наведено на рисунку 3.27.

```
void sprinklerOn(int room)
{
  turnOffElectricity();
  Blynk.notify("В будинку пожежа! Будинок знеструмлено та увімкнено систему пожежогасіння!");
  sprinklerLeds[room].on();
  digitalWrite(sprinklerPins[room], HIGH);
}

void sprinklerOff(int room)
{
  sprinklerLeds[room].off();
  digitalWrite(sprinklerPins[room], LOW);
}
```

Рисунок 3.27 – Фрагмент коду з функціями ввімкнення та вимкнення розбризкувачів

Повний код програмного забезпечення даного мікроконтролера наведений у додатку Г.

Підсистема безпеки складається з трьох мікроконтролерів. Алгоритм роботи першого мікроконтролера, який відповідає за збір даних з датчиків руху, наведений на рисунку 3.28.

		Андрушенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

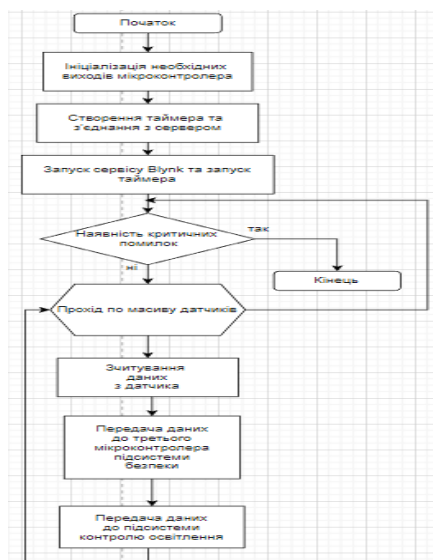


Рисунок 3.28 – Алгоритм роботи першого мікроконтролера підсистеми безпеки

Програмне забезпечення даного мікроконтролера, окрім стандартних дій з ініціалізації виходів, створення таймерів та з'єднання з сервером, здійснює збір даних з датчиків руху та їх передачу на третій мікроконтролер підсистеми безпеки та на мікроконтролер підсистеми контролю освітлення за допомогою відповідних віртуальних мостів. Фрагмент коду, який відповідає за збір та відправку даних наведений на рисунку 3.29.

```

for (int i = 0; i < 7; i++)
{
  if (digitalRead(sensorPins[i]) == 1)
  {
    motionLeds[i].on();
    securityBridge.virtualWrite(bridgePins[i], 1);
    lightBridge.virtualWrite(bridgePins[i], 1);
  }
  else
  {
    motionLeds[i].off();
    securityBridge.virtualWrite(bridgePins[i], 0);
    lightBridge.virtualWrite(bridgePins[i], 0);
  }
}

```

Рисунок 3.29 – Фрагмент коду, який відповідає за збір та відправку даних з датчиків руху

Алгоритм роботи другого мікроконтролера підсистеми безпеки, який відповідає за збір даних з датчиків відкриття вікон та дверей, наведений на рисунку 3.30.

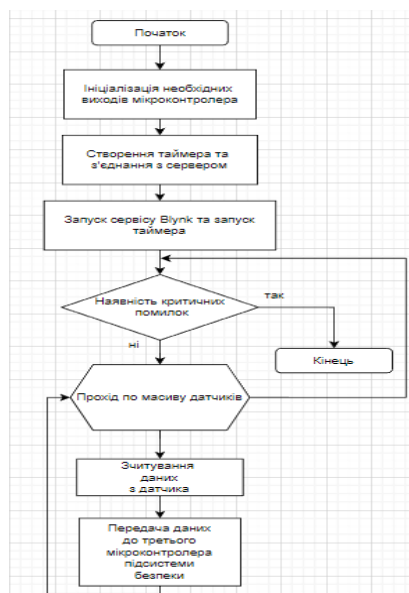


Рисунок 3.30 – Алгоритм роботи другого мікроконтролера підсистеми безпеки

Даний мікроконтролер після ініціалізації виходів, створення таймера та налагодження з'єднання з сервером здійснює прохід по масиву датчиків відкриття вікон та дверей, збирає дані з них та передає їх на третій мікроконтролер підсистеми безпеки за допомогою віртуального мосту. Фрагмент коду, який відповідає за описані вище дії наведено на рисунку 3.31.

```

for (int i = 0; i < 5; i++)
{
  if (digitalRead(sensorPins[i]) == 1)
  {
    dwLeds[i].on();
    securityBridge.virtualWrite(bridgePins[i], 1);
  }
  else
  {
    dwLeds[i].off();
    securityBridge.virtualWrite(bridgePins[i], 0);
  }
}

```

Рисунок 3.31 – Фрагмент коду, який відповідає за збір та передачу даних з датчиків

Третій мікроконтролер підсистеми безпеки відповідає за обробку отриманих даних з датчиків руху та відкриття вікон і дверей. Алгоритм його роботи наведений на рисунку 3.32.

Робота даного мікроконтролера, аналогічно з попередніми, розпочинається з ініціалізації виходів, створення таймерів та з'єднання з сервером. Після цього відбувається отримання даних з датчиків, а також їх обробка.

Якщо підсистема знаходиться в режимі охорони, то при виявленні руху, або ж відкритті двере, чи вікон, власникам будинку буде надіслано відповідне сповіщення на телефон, якщо режим охорони вимкнено, то система сповіщень не надсилатиме. Фрагмент коду, який відповідає за обробку даних з датчиків наведений на рисунку 3.33.

```
void checkMotion()
{
    for (int i = 0; i < 7; i++)
    {
        if(security){
            if(motion[i] == 1){
                Blynk.notify("В будинку виявлено рух! Можливий несанкціонований доступ!");
            }
        }
    }
};

void checkOpen()
{
    for (int i = 0; i < 5; i++)
    {
        if(security){
            if(opening[i] == 1){
                Blynk.notify("Виявлено відкриття вікна/дверей! Можливий несанкціонований доступ!");
            }
        }
    }
};
```

Рисунок 3.33 – Функції, що відповідають за обробку даних з датчиків

Повний код програмного забезпечення мікроконтролерів підсистеми безпеки наведений в додатках Д.1, Д.2 та Д.3 відповідно.

Підсистема керування освітленням складається з одного мікроконтролера, алгоритм роботи якого представлений на рисунку 3.34.

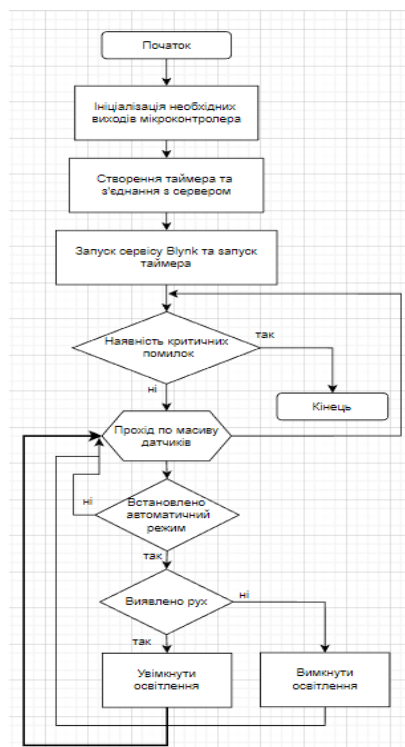


Рисунок 3.34 – Алгоритм роботи мікроконтролера підсистеми контролю освітлення

Робота даного мікроконтролера, як і всіх інших в нашій системі, розпочинається з ініціалізації виходів, створення таймера та налагодження зв'язку з сервером. Після цього здійснюється прохід по масиву отриманих від мікроконтролера підсистеми безпеки даних з датчиків руху. Якщо встановлений автоматичний режим освітлення, то при виявленні руху у відповідній кімнаті вмикатиметься освітлення. Функція, яка відповідає за цю перевірку предстала на рисунку 3.35.

```

void checkMotion()
{
    for (int i = 0; i < 6; i++)
    {
        if (autoMode)
        {
            if (motion[i] == 1)
            {
                digitalWrite(relayPins[i], HIGH);
            }
            else
            {
                digitalWrite(relayPins[i], LOW);
            }
        }
    }
};

```

Рисунок 3.35 – Функція керування освітленням

Повний код програмного забезпечення мікроконтролера підсистеми керування освітленням наведений у додатку Е.

Підсистема керування водопостачанням містить у своєму складі один мікроконтролер. Алгоритм його роботи представлений на рисунку 3. 36.

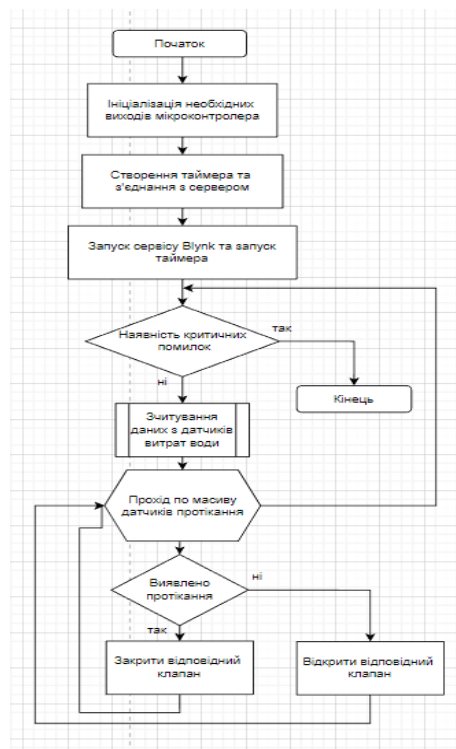


Рисунок 3.36 – Алгоритм роботи мікроконтролера системи контролю водопостачання

Після створення всіх необхідних таймерів, з'єднання з сервером та ініціалізації відповідних виходів відбувається зчитування даних з датчиків витрат води. Функція, яка відповідає за зчитування наведена на рисунку 3.37.

```

void update()
{
    currentTime = millis();
    if (currentTime >= (loopTime + 1000))
    {
        loopTime = currentTime;
        count_imp_allB = count_imp_allB + count_impB;
        count_imp_allK = count_imp_allK + count_impK;
        Blynk.virtualWrite(V10, count_imp_allB / 450000);
        Blynk.virtualWrite(V11, count_imp_allK / 450000);

        count_impB = 0;
        count_impK = 0;
    }
}

```

Рисунок 3.37 – Функція зчитування даних з датчиків витрат води

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

Після зчитування даних з датчиків витрат води відбувається прохід по масиву датчиків протікання. Якщо протікання буде виявлено, то у відповідному місці буде перекрито клапан і водопостачання припиниться. Фрагмент коду, що виконує дану перевірку зображений на рисунку 3.38.

```
void readData()
{
  for (int i = 0; i < 2; i++)
  {
    if (digitalRead(sensorPins[i]) == 1)
    {
      leakLeds[i].on();
      closeValve(i);
      Blynk.notify("Виявлено протікання! Водопостачання до місця протікання перекрито!");
    }
    else
    {
      leakLeds[i].off();
      if (!flags[i])
      {
        openValve(i);
      }
    }
  }
};
```

Рисунок 3.38 – Фрагмент коду, що відповідає за перевірку датчиків протікання

Повний код програмного забезпечення мікроконтролера системи контролю водопостачання наведений в додатку К.

Підсистема контролю доступу теж складається лише з одного мікроконтролера, алгоритм роботи якого наведений на рисунку 3.39.

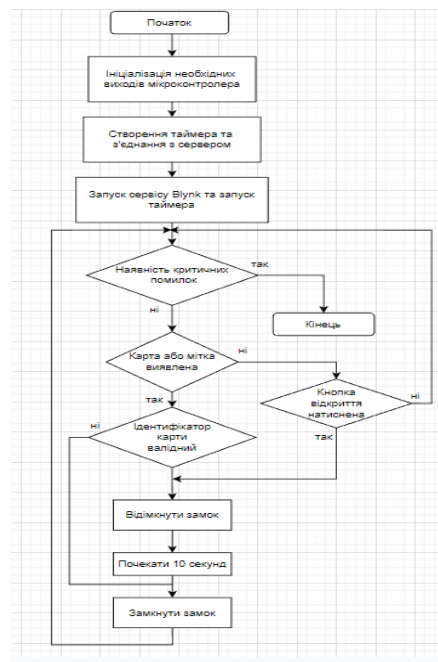


Рисунок 3.39 – Алгоритм роботи мікроконтролера підсистеми контролю доступу

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

На початку своєї роботи мікроконтролер ініціалізує необхідні виходи, встановлює з'єднання з сервером, створює необхідні таймери та запускає їх. Після цього відбувається очікування піднесення картки, або ж мітки. Якщо її виявлено, то відбувається перевірка на наявність ідентифікатора в списку дозволених. Якщо перевірку пройдено успішно – замок відкривається. Фрагмент коду, який відповідає за описану перевірку наведено на рисунку 3.40.

```
void waitCard()
{
    if (!mfr522.PICC_IsNewCardPresent())
    {
        return;
    }
    if (!mfr522.PICC_ReadCardSerial())
    {
        return;
    }
    String content = "";
    for (byte i = 0; i < mfr522.uid.size; i++)
    {
        content.concat(String(mfr522.uid.uidByte[i] < 0x10 ? "0" : " "));
        content.concat(String(mfr522.uid.uidByte[i], HEX));
    }
    for (int i = 0; i < CARDS; i++)
    {
        if (content.substring(1) == cards[i])
        {
            unlockDoor();
        }
        else
        {
            lockDoor();
        }
    }
}
```

Рисунок 3.40 – Функція, яка відповідає за перевірку зчитаної карти, або ж мітки

Також замок можна відкрити зсередини за допомогою кнопки. Замок автоматично закривається кожних 15 секунд.

Повний код програмного забезпечення мікроконтролера наведено у додатку Л.

3.3 Тестування системи «Smart Home»

Перед початком тестування системи необхідно запустити локальний сервер системи Blynk, рис. 3.41.

```
PS F:\Diplom\Blynk_Server> java -jar .\server-0.41.16-java8.jar -dataFolder data
Blynk Server 0.41.17-SNAPSHOT successfully started.
All server output is stored in folder 'F:\Diplom\Blynk_Server\.\logs' file.
```

Рисунок 3.41 – Запуск локального сервера Blynk

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

Далі необхідно перейти в мобільний додаток, авторизуватися там та створити новий проект, вказавши його назву та обравши мікроконтролер ESP8266 (рис. 3. 42).

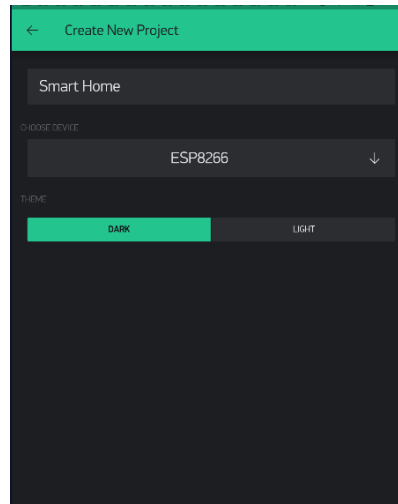


Рисунок 3. 42 – Створення нового проекту

Далі в налаштуваннях проекту у розділі «Пристрої» необхідно додати необхідну для функціонування системи кількість мікроконтролерів. В нашому випадку таких пристроїв буде 11 (рисунок 3.43).

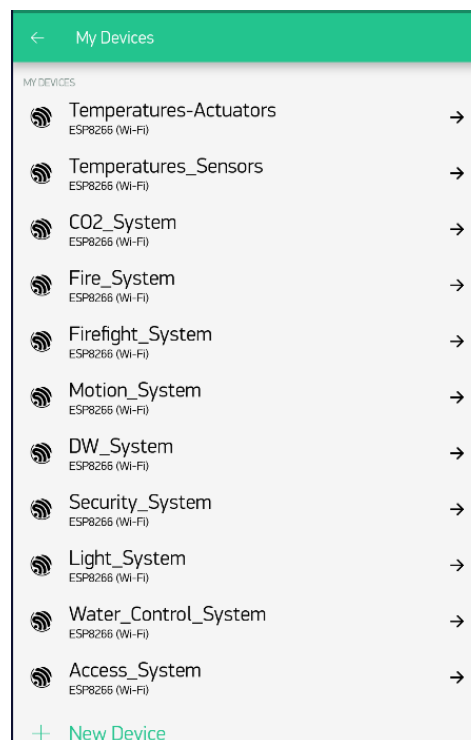


Рисунок 3.43 – Список пристроїв проекту

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

Після створення проекту та додавання до нього всіх пристроїв необхідно у мобільному додатку розмістити віджети для керування підсистемами. Розпочнемо з підсистеми регулювання температури. На відповідній вкладці розмістимо всі необхідні для керування віджети, рис. 3.44.



Рисунок 3.44 – Панель керування підсистемою регулювання температури

При ручному режимі керування можна здійснювати ввімкнення та вимкнення приладів власноруч за допомогою кнопок (рис. 3. 45).

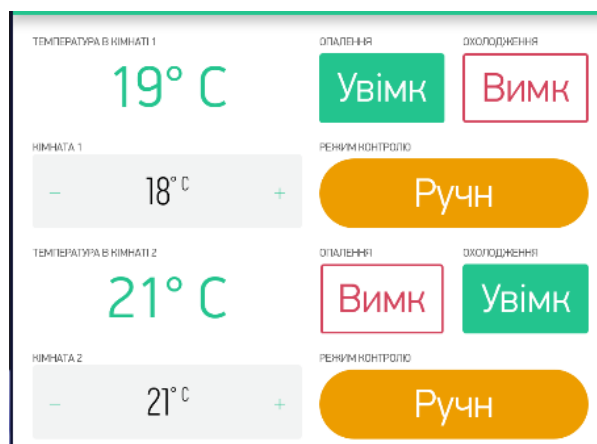


Рисунок 3. 45 – Ручне керування приладами

Для автоматичного підтримання заданої температури необхідно виставити бажану температуру та перевести систему в режим «Авто» натисканням на відповідну кнопку (рис. 3. 46).

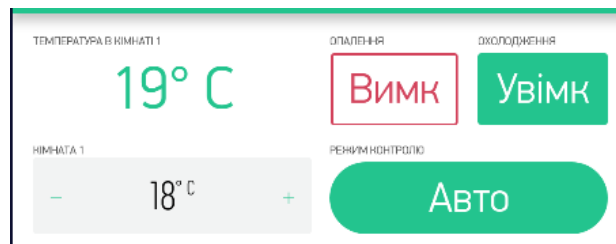


Рисунок 3. 46 – Автоматичний режим регулювання температури

При автоматичному режимі прилади будуть вмикатися залежно від поточної температури в кімнаті. Якщо температура менша за встановлену, то вмикатиметься обігрівач (рис. 3. 47), якщо більша, то охолоджувач (рис. 3. 48), а якщо температура рівна встановленій, то обидва прилади вимикатимуться (рис. 3. 49).

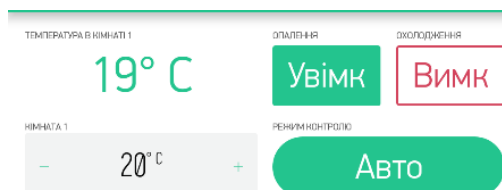


Рисунок 3. 47 – Робота підсистеми при температурі, меншій за встановлену

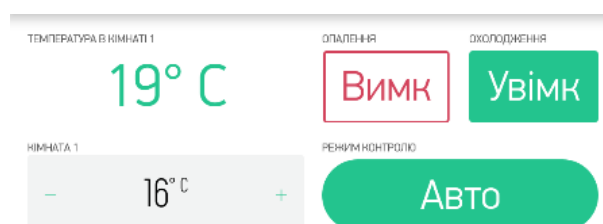


Рисунок 3. 48 – Робота підсистеми при температурі, більшій за встановлену

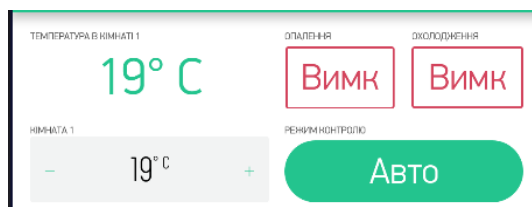


Рисунок 3. 49 – Робота підсистеми при температурі рівній встановленій

Як бачимо підсистема коректно реагує на зміни температури та правильно вмикає необхідні для правильного регулювання прилади. З цього можна зробити висновок, що програмне забезпечення для неї створено та працює правильно.

Наступною налаштуємо панель керування системою контролю якості повітря, розмістивши на відповідні вкладці елементи керування, рис. 3.50.



Рисунок 3.50 – Панель керування підсистемою контролю якості повітря

Запустивши систему ми бачимо рівень вмісту газу в повітрі на кухні та в трьох кімнатах, рис. 3.51.

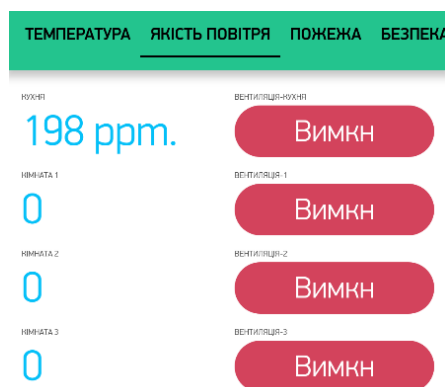


Рисунок 3.51 – Відображення даних з датчиків підсистеми контролю якості повітря

		Андрушенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо рівень концентрації шкідливих газів буде перебільшено, то у відповідній кімнаті буде увімкнено систему вентиляції (рис. 3.52), а також буде надіслано сповіщення про небезпеку (рис. 3.53).

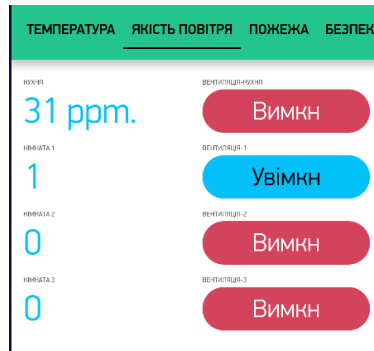


Рисунок 3.52 – Увімкнення системи вентиляції при перевищенні концентрації шкідливих газів

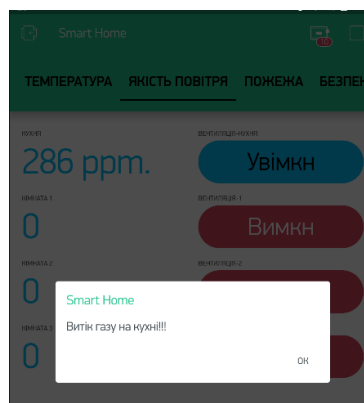


Рисунок 3.53 – Сповіщення про небезпеку

Як бачимо при перевищенні рівня шкідливих газів у повітрі система вентиляції автоматично вмикається, та вимикається при його нормалізації, отже програмне забезпечення працює правильно.

Оскільки підсистеми моніторингу вогню та пожежогасіння є автоматичними і не потребують ручного керування, то налаштуємо панель спостереження за ними. Розмістимо на панелі необхідні віджети для контролю ситуації з пожежною безпекою в будинку, рис 3.54.



Рисунок 3.54 – Панель спостереження за підсистемами моніторингу вогню та пожежогасіння

При виявленні загорання у відповідній кімнаті буде увімкнено розбризкувач води (рис. 3.55), а також вимкнено електропостачання будинку, про що користувача буде сповіщено відповідним повідомленням (рис. 3.56).

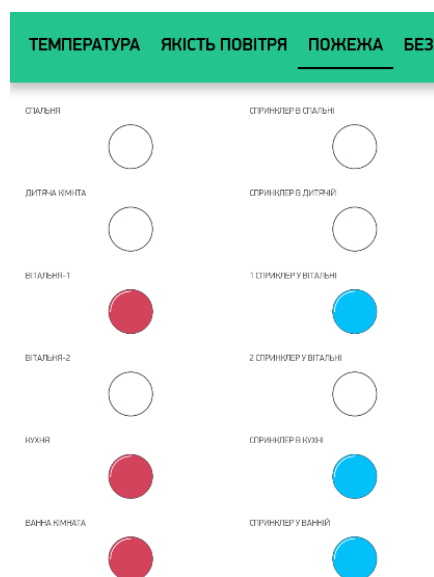


Рисунок 3.55 – Увімкнення розбризкувачів при загоранні

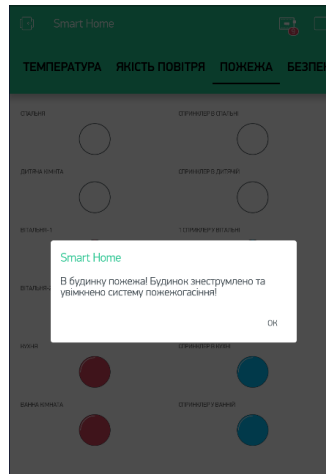


Рисунок 3.56 – Сповіщення про загорання та знеструмлення будинку

На панелі керування підсистемою безпеки буде розміщено лише одну кнопку, яка дозволить ставити будинок під охорону та індикаторів датчиків руху та відкриття вікон і дверей (рис. 3.57).

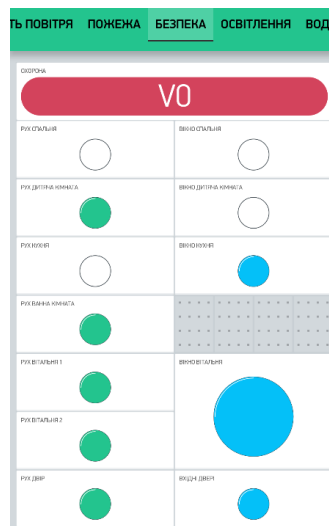


Рисунок 3.57 – Панель керування підсистемою безпеки

При постановці будинку під охорону будь-який рух всередині будинку, або ж відкриття дверей та вікон будуть зафіксовані відповідними датчиками та про несанкціонований доступ буде повідомлено господарям будинку (рис. 3.58).

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

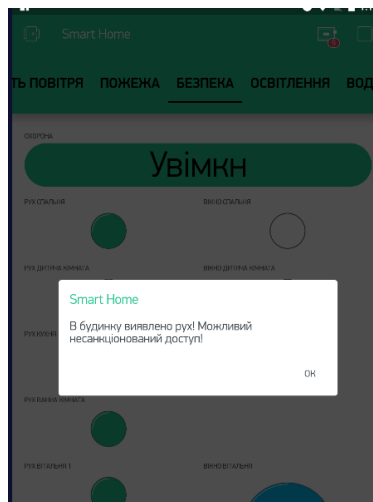


Рисунок 3. 58 – Сповіщення про можливий несанкціонований доступ
При вимкненій охороні зафіксовані дані з датчиків ігноруються (рис. 3. 59).

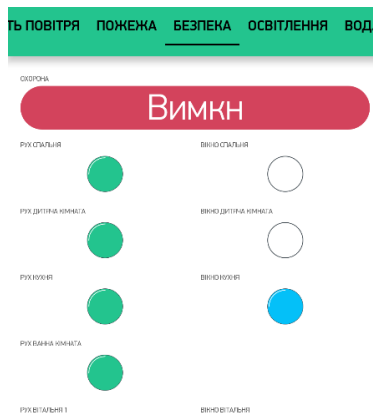


Рисунок 3.59 – Ігнорування показів датчиків при вимкненій охороні
будинку

Панель керування підсистемою освітлення міститиме на собі 7 кнопок: одну для встановлення режиму освітлення та 6 для керування освітленням в окремих місцях будинку (рис. 3.60).

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

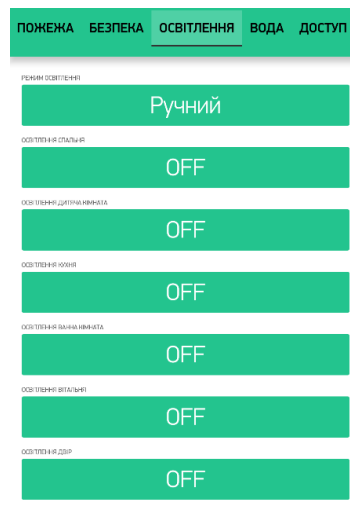


Рисунок 3.60 – Панель керування підсистемою освітлення

Автоматичний режим дозволяє керувати освітленням за допомогою датчиків руху, тобто вмикати освітлення лише в кімнатах, де наявний рух. Увімкнення автоматичного режиму не виключає можливість ручного керування освітленням.

Панель керування підсистемою контролю водопостачання міститиме на собі індикатори протікань, кнопки керування клапанами водопостачання, а також дані з лічильників витрат води (рис. 3.61).

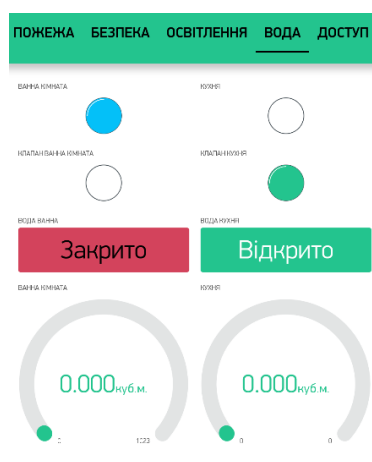


Рисунок 3. 61 – Панель керування підсистемою контролю водопостачання

При виявленні протікання у відповідному місці буде перекрито клапан водопостачання, а також господарю будинку буде надіслано відповідне

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

сповіщення (рис. 3. 62). Також перекривання клапанів можна здійснювати за допомогою відповідних кнопок.

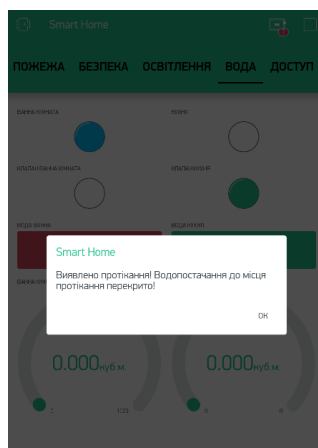


Рисунок 3. 62 – Сповіщення про виявлення протікання та призупинення водопостачання

Датчики витрат води показують кількість води, яку було витрачено у відповідному місці (рис. 3.62).

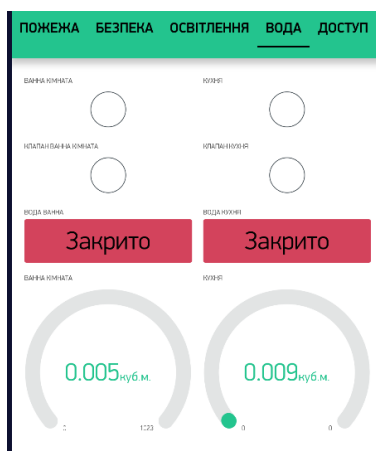


Рисунок 3.62 – Відображення кількості витраченої води

Панель керування підсистою контролю доступу міститиме лише дві кнопки для відмикання замка та встановлення режиму охорони будинку та індикатор стану замка (рис. 3. 63).

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

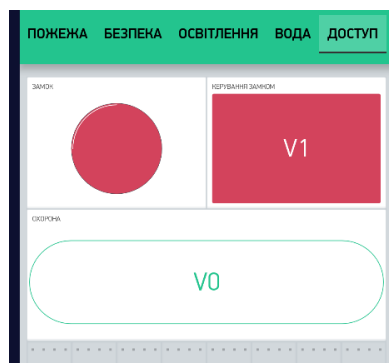


Рисунок 3.63 – Панель керування підсистемою контролю доступу

За допомогою цієї панелі можна відкрити, або ж закрити замок (рис. 3.64) та поставити будинок під охорону (рис. 3. 65).

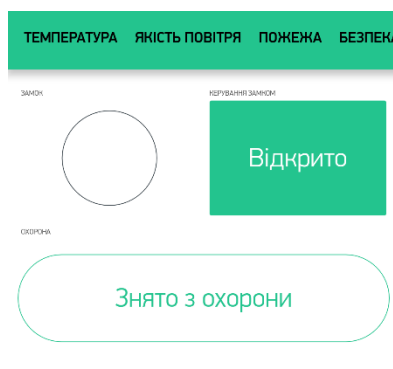


Рисунок 3. 64 – Відкриття замка через панель керування

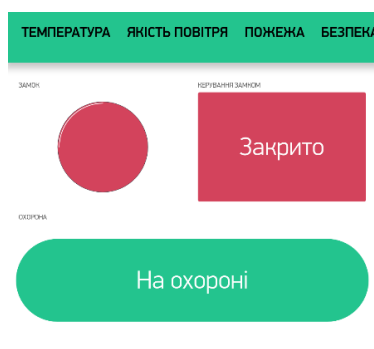


Рисунок 3.65 – Постановка будинку під охорону через панель керування

Як ми бачимо все працює справно, отже програмне забезпечення для мікроконтролерів розроблено правильно.

3.4 Висновки до розділу 3

В розділі 3 було підібрано та обґрунтовано вибір відповідних компонентів для організації роботи системи, а також розроблено алгоритми роботи мікроконтролерів підсистем.

Відповідні реалізовані технічні рішення та пропозиції були пояснені та представлені ключові фрагменти коду програмного забезпечення мікроконтролерів.

Систему було реалізовано на базі локального сервера Vlnk, створені панелі керування для кожної підсистеми та показано принципи їх роботи та основні технічні можливості.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній роботі було проведено аналіз існуючих систем «Smart Home», підходів до їх побудови, описано переваги та недоліки кожної з них, які було враховано при подальшій розробці системи.

Також в даній роботі було спроектовано систему «Smart Home». Після проведення аналізу підходів до побудови систем було обрано підхід без використання центрального хаба в ролі керуючого пристрою для організації роботи системи, що дозволить розділити нашу систему на підсистеми, які взаємодіятимуть між собою за допомогою локального сервера. Кожна підсистема керуватиметься відповідними їх мікроконтролерами, що дозволить розподілити навантаження на систему рівномірно. Також при проектуванні підсистем було реалізовано можливість використання компонентів одних підсистем іншими підсистемами. Прикладом того є використання датчиків руху, які належать до підсистеми безпеки, підсистемою контролю освітлення.

А також було проаналізовано апаратне забезпечення, підібрано та обґрунтовано вибір відповідних компонентів для організації роботи системи, а також розроблено алгоритми роботи мікроконтролерів підсистем. Відповідні реалізовані технічні рішення та пропозиції були пояснені та представлені ключові фрагменти коду програмного забезпечення мікроконтролерів.

Систему було реалізовано на базі локального сервера Blynk, а також створено панелі керування для кожної підсистеми та показано принципи їх роботи, їх основні можливості.

Розроблена у ході виконання кваліфікаційної роботи система може використовуватися у приватних будинках а також може бути покращена шляхом використання хмарного сервера Blynk замість локального.

Для демонстрації роботи системи було розроблено модель системи у середовищі Cisco Packet Tracer, яка дозволить показати роботу всієї системи вцілому.

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Документація Blynk. Blynk Documentation. URL: <https://docs.blynk.io/en/>.
2. Програмування мовою C/C++. METANIT.COM. URL: <https://metanit.com/cpp/c>.
3. Репозиторій локального сервера Blynk. GitHub. URL: <https://github.com/blynkkk/blynk-server>.
4. Спільнота ESP8266. Спільнота ESP8266. URL: <https://esp8266.ru/>.
5. blynk. (LEGACY) Tutorial: Control multiple devices with Blynk app (Arduino Mega and nodeMCU), 2017. YouTube. URL: <https://www.youtube.com/watch?v=FhS44hGk1Lc>.
6. ESP8266 – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/ESP8266>.
7. Kirill Ivanychev. Як налаштувати локальний сервер Blynk, 2020. YouTube. URL: <https://www.youtube.com/watch?v=N5W58hBq-Io>.
8. Учасники проєктів Вікімедіа. Розумний дім – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Розумний_дім.
9. Учасники проєктів Вікімедіа. Raspberry Pi – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Raspberry_Pi.
10. IoT Explained — How Does an IoT System Actually Work? // електрон. текст. дані URL: <https://medium.com/iotforall/iot-explained-how-does-an-iot-systemactually-work-e90e2c435fe7>.
11. RFID-технологія // електрон. текст. дані URL: <https://www.idexpert.ru/technology/121/>.
12. RFID-технології и магнітні мітки // електрон. текст. дані URL: <http://allta.com.ua/what-is-rfid>.
13. Ардуїно реле. ArduinoMaster все об Ардуино. URL: <https://arduinomaster.ru/datchiki-arduino/podklyuchenie-rele-k-arduino/>.

		Андрусенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

- 14.DHT11 и DHT22 – підключення до Ардуїно датчиків температури та вологості. URL: <https://arduino-master.ru/datchiki-arduino/datchiki-temperature-i-vlazhnosti-dht11-dht22/>.
- 15.Seneviratne P. Hands-On Internet of Things with Blynk: Build on the power of Blynk to configure smart devices and build exciting IoT projects. Packt Publishing, 2018. 266 p.
- 16.Margolis M. Arduino cookbook. 2nd ed. Sebastopol, Calif : O'Reilly, 2012. 699 p.
- 17.Шварц М. Интернет речей з ESP8266. 2-ге вид. Санкт Петербург : БХВ-Петербург, 2019. 224 с.
- 18.Using Node MCU / P. Chati et al. International Journal of Innovations in Engineering and Science. 2020. Vol. 5, no. 10. P. 41. URL: <https://doi.org/10.46335/ijies.2020.5.10.9>.
- 19.Weather Monitoring System using Blynk Application. International Journal of Innovative Technology and Exploring Engineering. 2019. Vol. 9, no. 1. P. 1315–1319. URL: <https://doi.org/10.35940/ijitee.l3666.119119>.
- 20.Stroustrup B. C++ Programming Language. Addison Wesley, 2013. 1368 p.

ДОДАТКИ

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А.1 – Код програмного забезпечення першого мікроконтролера підсистеми регулювання температури

```
#define BLYNK_AUTH_TOKEN "c00QDv1evdc0v6w7r3yr-VzxAhwKN0w5"
#define DHTTYPE DHT11
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "DHT.h"
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
uint8_t dhtPins[] = {D1, D2, D3, D4, D5};
uint8_t dhtVirtual[] = {V39, V40, V41, V42, V43};

DHT dhtArray[] = {DHT(dhtPins[0], DHTTYPE),
                  DHT(dhtPins[1], DHTTYPE),
                  DHT(dhtPins[2], DHTTYPE),
                  DHT(dhtPins[3], DHTTYPE),
                  DHT(dhtPins[4], DHTTYPE)};

BlynkTimer timer;
WidgetBridge bridge(V10);
BLYNK_CONNECTED()
{
  bridge.setAuthToken("qT9Uv1R1B97oK112F31VBnr5jRCywPXq");
}

void sendDHTData()
{
  for(int i = 0; i < (int)(sizeof(dhtArray)/sizeof(dhtArray[0])); i++){
    float value = (int)dhtArray[i].readTemperature();
    bridge.virtualWrite(dhtVirtual[i], value);
  }
}

void initializeDHT()
{
  for (int i = 0; i < 5; i++)
  {
    pinMode(dhtPins[i], INPUT);
    dhtArray[i].begin();
  }
}

void setup()
{
  Serial.begin(115200);
  initializeDHT();
  Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080);
  timer.setInterval(1000L, sendDHTData);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

		Андрусенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А.2 – Код програмного забезпечення другого мікроконтролера підсистеми регулювання температури

```
#define BLYNK_AUTH_TOKEN "qT9Uv1R1B97oK112F31VBnr5jRCywPXq"
#define BLYNK_PRINT Serial
#define ROOMS 5 //кількість підключених кімнат
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
int temps[ROOMS];
uint8_t heaterPins[] = {16, 4, 2, 12, 15},
        coolerPins[] = {5, 0, 14, 13, 3};
uint8_t heatersVirtual[] = {V11, V13, V15, V17, V19},
        coolersVirtual[] = {V12, V14, V16, V18, V20},
        switchModePins[] = {V21, V22, V23, V24, V25};
BlynkTimer timer;
int modes[ROOMS] = {0, 0, 0, 0, 0},
        tempLevels[ROOMS] = {15, 15, 15, 15, 15};
void initializePins()
{
    for (int i = 0; i < ROOMS; i++)
    {
        pinMode(heaterPins[i], OUTPUT);
        pinMode(coolerPins[i], OUTPUT);
    }
}
BLYNK_WRITE(V30)
{tempLevels[0] = param.asInt();}
BLYNK_WRITE(V31)
{tempLevels[1] = param.asInt();}
BLYNK_WRITE(V32)
{tempLevels[2] = param.asInt();}
BLYNK_WRITE(V33)
{tempLevels[3] = param.asInt();}
BLYNK_WRITE(V34)
{tempLevels[4] = param.asInt();}
BLYNK_WRITE(V39)
{temps[0] = param.asInt();Blynk.virtualWrite(V45, param.asInt());}
BLYNK_WRITE(V40)
{temps[1] = param.asInt();Blynk.virtualWrite(V46, param.asInt());}
BLYNK_WRITE(V41)
{temps[2] = param.asInt();Blynk.virtualWrite(V47, param.asInt());}
BLYNK_WRITE(V42)
{temps[3] = param.asInt();Blynk.virtualWrite(V48, param.asInt());}
BLYNK_WRITE(V43)
{temps[4] = param.asInt();Blynk.virtualWrite(V49, param.asInt());}
BLYNK_WRITE(V21)
```

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{ modes[0] = param.asInt();}
BLYNK_WRITE(V22)
{ modes[1] = param.asInt();}
BLYNK_WRITE(V23)
{modes[2] = param.asInt();}
BLYNK_WRITE(V24)
{modes[3] = param.asInt();}
BLYNK_WRITE(V25)
{ modes[4] = param.asInt();}

void heatOff(int room)
{digitalWrite(heaterPins[room], 0);Blynk.virtualWrite(heatersVirtual[room], 0);}

void coolOff(int room)
{digitalWrite(coolerPins[room], 0);Blynk.virtualWrite(coolersVirtual[room], 0);}

void offAtRoom(int room)
{heatOff(room);coolOff(room);}

void cool(int room)
{heatOff(room);digitalWrite(coolerPins[room], 1);Blynk.virtualWrite(coolersVirtual[room], 1);}

void heat(int room)
{coolOff(room);digitalWrite(heaterPins[room], 1);Blynk.virtualWrite(heatersVirtual[room], 1);}

BLYNK_WRITE(V11)
{param.asInt() == 1 ? heat(0) : heatOff(0)}
BLYNK_WRITE(V12)
{ param.asInt() == 1 ? cool(0) : coolOff(0);}
BLYNK_WRITE(V13)
{param.asInt() == 1 ? heat(1) : heatOff(1);}
BLYNK_WRITE(V14)
{ param.asInt() == 1 ? cool(1) : coolOff(1);}
BLYNK_WRITE(V15)
{param.asInt() == 1 ? heat(2) : heatOff(2);}
BLYNK_WRITE(V16)
{param.asInt() == 1 ? cool(2) : coolOff(2);}
BLYNK_WRITE(V17)
{param.asInt() == 1 ? heat(3) : heatOff(3);}
BLYNK_WRITE(V18)
{param.asInt() == 1 ? cool(3) : coolOff(3);}
BLYNK_WRITE(V19)
{param.asInt() == 1 ? heat(4) : heatOff(4);}
BLYNK_WRITE(V20)
{ param.asInt() == 1 ? cool(4) : coolOff(4);}

void autoLevel(int room)
{if (temps[room] < tempLevels[room]){heat(room)} else if (temps[room] > tempLevels[room]) {cool(room);}
else offAtRoom(room);}

void checkModes()
{for (int i = 0; i < ROOMS; i++)
{if (modes[i] == 1){autoLevel(i);}}}

void setup()
{Serial.begin(115200);initializePins();
Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080); timer.setInterval(1000L, checkModes); }

void loop()
{Blynk.run();timer.run();}

```

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Б – Код програмного забезпечення мікроконтролера підсистеми контролю якості повітря

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "MgkXDyDEwVWIQJ6M49AAk7zw9zgJZ7US"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
uint8_t kitchenPin = A0,
        kitchenFan = D3;
BlynkTimer timer;
int kitchenCO2level;
#define CO2_WARNING_LEVEL 250
uint8_t rooms[] = {D0, D1, D2}, fans[] = {D4, D5, D6}, virtualRooms[] = {V1, V2, V3}, leds[] = {V5,
V6, V7};
int data[3] = {};
BLYNK_WRITE(V4)
{if (param.asInt() == 1){digitalWrite(kitchenFan, HIGH);}else digitalWrite(kitchenFan, LOW);
}
void initializePins()
{pinMode(kitchenPin, INPUT);pinMode(kitchenFan, OUTPUT);
for (int i = 0; i < 3; i++)
{
pinMode(rooms[i], INPUT);pinMode(fans[i], OUTPUT);}}
void checkKitchen(int value)
{
if (value > CO2_WARNING_LEVEL)
{Blynk.virtualWrite(V4, 1);Blynk.notify("Витік газу на кухні!!!");}
else Blynk.virtualWrite(V4, 0);
}
void readData()
{
kitchenCO2level = analogRead(kitchenPin);
Blynk.virtualWrite(V0, kitchenCO2level);
checkKitchen(kitchenCO2level);
for (int i = 0; i < 3; i++)
{
data[i] = digitalRead(rooms[i]);
Blynk.virtualWrite(virtualRooms[i], data[i]);
if (data[i] == 1)
{ Blynk.virtualWrite(leds[i], 1);digitalWrite(fans[i], HIGH);
}
else
{Blynk.virtualWrite(leds[i], 0);digitalWrite(fans[i], LOW);
}
}
}
void setup()
{initializePins();Serial.begin(115200); Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0,
108), 8080);
timer.setInterval(1000L, readData);
}

void loop()
{
Blynk.run();
timer.run();
}
```

		Андрусенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток В – Код програмного забезпечення мікроконтролера підсистеми моніторингу пожежі

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "yf56D2F_5oFmc7MY0tF_0YV9eZGpAtG0"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
BlynkTimer timer;
WidgetBridge firefight(V10);
uint8_t fireDetectorPins[] = {16, 5, 4, 0, 2, 14},
      virtualFireDetectors[] = {V0, V1, V2, V3, V4, V5};
WidgetLED fireDetectorLeds[] = {WidgetLED(V10),
                                WidgetLED(V11),
                                WidgetLED(V12),
                                WidgetLED(V13),
                                WidgetLED(V14),
                                WidgetLED(V15)};

BLYNK_CONNECTED()
{
  firefight.setAuthToken("c3ssoIT5ntOS-gmJrfpMhF_NI4rHiW");
}
void initializePins()
{
  for (int i = 0; i < 6; i++)
  {
    pinMode(fireDetectorPins[i], INPUT);
  }
}
void readData()
{
  for (int i = 0; i < 6; i++)
  {
    int tmp = digitalRead(fireDetectorPins[i]);
    if (tmp == 1)
    {
      firefight.virtualWrite(virtualFireDetectors[i], 1);
      fireDetectorLeds[i].on();
    }
    else
    {
      firefight.virtualWrite(virtualFireDetectors[i], 0);
      fireDetectorLeds[i].off();
    }
  }
}
void setup()
{
  initializePins();
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080);
  timer.setInterval(1000L, readData);
}
void loop()
{
  Blynk.run();
  timer.run();
}
```

		Андрусенко О.М.			<i>KIK.KP.B – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Г – Код програмного забезпечення мікроконтролера підсистеми пожежогасіння

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "c3ssoIT5ntOS-gmJrfpHeMhF_NI4rHiW"
#define BLYNK_PRINT Serial
#define BLYNK_MAX_SENDBYTES 256
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
bool electricity = true;
uint8_t sprinklerPins[] = {16, 5, 4, 0, 2, 14};
WidgetLED sprinklerLeds[] =
{WidgetLED(V20),WidgetLED(V21),WidgetLED(V22),WidgetLED(V23),WidgetLED(V24),WidgetLED(V
25)};
void initializePins()
{for (int i = 0; i < 6; i++)
  {pinMode(sprinklerPins[i], OUTPUT);}pinMode(13, OUTPUT);}
void turnOffElectricity()
{if (electricity){
  electricity = false;digitalWrite(13, LOW);}}
void sprinklerOn(int room)
{turnOffElectricity();
  Blynk.notify("В будинку пожежа! Будинок знеструмлено та увімкнено систему
пожежогасіння! Перевірте застосунок для локалізації загоряння та викличте пожежних!");
  sprinklerLeds[room].on();digitalWrite(sprinklerPins[room], HIGH);
}
void sprinklerOff(int room)
{sprinklerLeds[room].off();digitalWrite(sprinklerPins[room], LOW);
}
BLYNK_WRITE(V0){if (param.asInt() == 1){sprinklerOn(0);}else{sprinklerOff(0);}}
BLYNK_WRITE(V1){if (param.asInt() == 1){sprinklerOn(1);}else{sprinklerOff(1);}}
BLYNK_WRITE(V2){if (param.asInt() == 1){sprinklerOn(2);}else{sprinklerOff(2);}}
BLYNK_WRITE(V3){if (param.asInt() == 1){sprinklerOn(3);}else{sprinklerOff(3);}}
BLYNK_WRITE(V4){if (param.asInt() == 1){sprinklerOn(4);}else{sprinklerOff(4);}}
BLYNK_WRITE(V5){if (param.asInt() == 1){sprinklerOn(5);}else{sprinklerOff(5);}}
void setup()
{initializePins();Serial.begin(115200);
  Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080);}
void loop()
{
  Blynk.run();
}
```

		Андрусенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Д.1 – Код програмного забезпечення першого мікроконтролера підсистеми безпеки

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "N3QwQGGxN0gqnt75Lb5Uz62QcSMLIUXh"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
uint8_t sensorPins[] = {D0, D1, D2, D3, D4, D5, D6},
        bridgePins[] = {V11, V12, V13, V14, V15, V16, V17};
WidgetLED motionLeds[] = {WidgetLED(V0),WidgetLED(V1),WidgetLED(V2),WidgetLED(V3),
        WidgetLED(V4),WidgetLED(V5),WidgetLED(V6)};

BlynkTimer timer;
WidgetBridge securityBridge(V10), lightBridge(V9);
BLYNK_CONNECTED()
{securityBridge.setAuthToken("uuShZLAEnxlmfCGxFyc_EPVX0o1hSsc");}
void initializePins()
{
    for (int i = 0; i < 7; i++)
    {
        pinMode(sensorPins[i], INPUT);
    }
}
void readData()
{
    for (int i = 0; i < 7; i++)
    {
        if (digitalRead(sensorPins[i]) == 1)
        {
            motionLeds[i].on();
            securityBridge.virtualWrite(bridgePins[i], 1);
            lightBridge.virtualWrite(bridgePins[i], 1);
        }
        else
        {
            motionLeds[i].off();
            securityBridge.virtualWrite(bridgePins[i], 0);
            lightBridge.virtualWrite(bridgePins[i], 0);
        }
    }
};
void setup()
{
    Serial.begin(115200);
    initializePins();
    Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080);
    timer.setInterval(1000L, readData);
}

void loop()
{
    Blynk.run();
    timer.run();
}
```

		Андрущенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Д.2 – Код програмного забезпечення другого мікроконтролера підсистеми безпеки

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "9eNp1tWJcaTjSt_k1pDFt-p2xBXeNLft"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
uint8_t sensorPins[] = {D0, D1, D2, D3, D4}, bridgePins[] = {V20, V21, V22, V23, V24};
WidgetLED dwLeds[] = {
    WidgetLED(V0), WidgetLED(V1), WidgetLED(V2), WidgetLED(V3), WidgetLED(V4)};
BlynkTimer timer; WidgetBridge securityBridge(V9);
BLYNK_CONNECTED()
{securityBridge.setAuthToken("uuShZLAEnximlfCGxFyc_EPVX0o1hSsc");}
void initializePins()
{
    for (int i = 0; i < 5; i++)
        {pinMode(sensorPins[i], INPUT);}
}
void readData()
{for (int i = 0; i < 5; i++)
    {if (digitalRead(sensorPins[i]) == 1)
        {
            dwLeds[i].on();
            securityBridge.virtualWrite(bridgePins[i], 1);
        }else
        {
            dwLeds[i].off();
            securityBridge.virtualWrite(bridgePins[i], 0);
        }
    }
};
void setup()
{
    Serial.begin(115200);
    initializePins();
    Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080);
    timer.setInterval(1000L, readData);
}

void loop()
{
    Blynk.run();
    timer.run();
}
```

		Андрусенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Д.3 – Код програмного забезпечення третього мікроконтролера підсистеми безпеки

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "uuShZLAEnximlFCGxFyc_EPVX0o1hSsc"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
bool security = false;
int motion[7] = {},
    opening[5] = {};
BlynkTimer timerMotion, timerOpen;
BLYNK_WRITE(V0){if (param.asInt() == 1){security = true;}
    Else security = false;}
BLYNK_WRITE(V11){motion[0] = param.asInt();}
BLYNK_WRITE(V12){motion[1] = param.asInt();}
BLYNK_WRITE(V13){motion[2] = param.asInt();}
BLYNK_WRITE(V14){motion[3] = param.asInt();}
BLYNK_WRITE(V15){motion[4] = param.asInt();}
BLYNK_WRITE(V16){motion[5] = param.asInt();}
BLYNK_WRITE(V17){motion[6] = param.asInt();}
BLYNK_WRITE(V20){opening[0] = param.asInt();}
BLYNK_WRITE(V21){opening[1] = param.asInt();}
BLYNK_WRITE(V22){opening[2] = param.asInt();}
BLYNK_WRITE(V23){opening[3] = param.asInt();}
BLYNK_WRITE(V24){opening[4] = param.asInt();}
void checkMotion(){
    for (int i = 0; i < 7; i++)
        {if(security){if(motion[i] == 1){
            Blynk.notify("В будинку виявлено рух! Можливий несанкціонований доступ!");}}}};
void checkOpen()
{for (int i = 0; i < 5; i++)
    {if(security){if(opening[i] == 1){
        Blynk.notify("Виявлено відкриття вікна/дверей! Можливий несанкціонований доступ!");
        }}}};
void setup()
{Serial.begin(115200); Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108),
8080); timerMotion.setInterval(1000L, checkMotion);
    timerOpen.setInterval(2000L, checkOpen);}
void loop()
{
    Blynk.run();
    timerMotion.run();
    timerOpen.run();
}
```

		Андрусенко О.М.			<i>КІК.КР.Б – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Е – Код програмного забезпечення мікроконтролера підсистеми керування освітленням

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "76WvLxEDG7znywHtDwvuK0RcDXay3IVT"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
int motion[6] = {};
bool autoMode = false;
BlynkTimer timer;
uint8_t relayPins[] = {D0, D1, D2, D3, D4, D5};
BLYNK_WRITE(V0){if (param.asInt() == 1){autoMode = true;}else autoMode = false;}
BLYNK_WRITE(V11){motion[0] = param.asInt();}
BLYNK_WRITE(V12){motion[1] = param.asInt();}
BLYNK_WRITE(V13){motion[2] = param.asInt();}
BLYNK_WRITE(V14){motion[3] = param.asInt();}
BLYNK_WRITE(V15){motion[4] = param.asInt();}
BLYNK_WRITE(V16){motion[5] = param.asInt();}
BLYNK_WRITE(V20){if (!autoMode){if (param.asInt() == 1){digitalWrite(relayPins[0], HIGH);}else digitalWrite(relayPins[0], LOW);}}
BLYNK_WRITE(V21){if (!autoMode){if (param.asInt() == 1){digitalWrite(relayPins[1],HIGH);}else digitalWrite(relayPins[1], LOW);}}
BLYNK_WRITE(V22){if (!autoMode){if (param.asInt() == 1){digitalWrite(relayPins[2], HIGH);}elsedigitalWrite(relayPins[2], LOW);}}
BLYNK_WRITE(V23){if (!autoMode){if (param.asInt() == 1){digitalWrite(relayPins[3], HIGH);}else digitalWrite(relayPins[3], LOW);}}
BLYNK_WRITE(V24){if (!autoMode){if (param.asInt() == 1){digitalWrite(relayPins[4],HIGH);}else digitalWrite(relayPins[4], LOW);}}
BLYNK_WRITE(V25){if (!autoMode){if (param.asInt() == 1){digitalWrite(relayPins[5],HIGH);}else digitalWrite(relayPins[5], LOW);}}
void initializePins()
{for (int i = 0; i < 6; i++)
{pinMode(relayPins[i], OUTPUT);}}
void checkMotion()
{ for (int i = 0; i < 6; i++)
{if (autoMode){if (motion[i] == 1){digitalWrite(relayPins[i], HIGH);}else digitalWrite(relayPins[i], LOW);}}}}
void setup()
{
Serial.begin(115200);
initializePins();
Blynk.begin(auth, ssid, pass, IPAddress(192, 168, 0, 108), 8080);
timer.setInterval(1000L, checkMotion);
}

void loop()
{
Blynk.run();
timer.run();
}
```

		Андрусенко О.М.			KIK.KP.B – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

Додаток К – Код програмного забезпечення мікроконтролера підсистеми контролю водопостачання

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "2pkwU9qcaWwW2BqYd3URrk0unU8ozjnt"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
const byte InterruptPinBathroom PROGMEM = D2;
const byte InterruptPinKitchen PROGMEM = D3;
volatile uint16_t count_impB, count_impK;
float count_imp_allB, count_imp_allK;
uint32_t currentTime, loopTime;
float literB = 0, literK = 0;
uint8_t sensorPins[] = {D0, D1}, relayPins[] = {D4, D5}, virtPins[] = {V7,V8};
WidgetLED leakLeds[] = {
    WidgetLED(V0),
    WidgetLED(V1),
},
    valveLeds[] = {WidgetLED(V2), WidgetLED(V3)};
BlynkTimer timer, cTimer;
bool flags[] = {true, true};
IRAM_ATTR void getFlowB()
{ count_impB++;}
IRAM_ATTR void getFlowK()
{count_impK++;}
void closeValve(int room)
{ digitalWrite(relayPins[room], LOW);
  valveLeds[room].off();
  Blynk.virtualWrite(virtPins[room], 1);}
void openValve(int room){ digitalWrite(relayPins[room], HIGH);
  valveLeds[room].on();
  Blynk.virtualWrite(virtPins[room], 0);}
BLYNK_WRITE(V7){if (param.asInt() == 1){flags[0] = true;closeValve(0);}else{flags[0] =
false;openValve(0);} }
BLYNK_WRITE(V8){if (param.asInt() == 1){flags[1] = true;closeValve(1);}else{flags[1] =
false;openValve(1);} }
void initializePins(){for (int i = 0; i < 2; i++){pinMode(sensorPins[i],
INPUT);pinMode(relayPins[i], OUTPUT);} pinMode(InterruptPinBathroom,
INPUT);pinMode(InterruptPinKitchen, INPUT);}
void readData(){ for (int i = 0; i < 2; i++){if (digitalRead(sensorPins[i]) == 1)
{ leakLeds[i].on();closeValve(i);Blynk.notify("Виявлено протікання! Водопостачання
до місця протікання перекрито!");}else{leakLeds[i].off();if (!flags[i]){openValve(i);}
} } }
void update(){ currentTime = millis();
  if (currentTime >= (loopTime + 1000))
  {
    loopTime = currentTime;count_imp_allB = count_imp_allB + count_impB;
    count_imp_allK = count_imp_allK + count_impK;Blynk.virtualWrite(V10, count_imp_allB
/ 450000);Blynk.virtualWrite(V11, count_imp_allK / 450000);count_impB = 0;count_impK =
0;}}
void setup(){
  Serial.begin(115200);initializePins(); Blynk.begin(auth, ssid, pass, IPAddress(192,
168, 0, 108), 8080); attachInterrupt(InterruptPinBathroom, getFlowB, FALLING);
  attachInterrupt(InterruptPinKitchen, getFlowK, FALLING); timer.setInterval(1000L,
readData); cTimer.setInterval(100L, update); currentTime = millis(); loopTime =
currentTime;}
void loop(){Blynk.run(); timer.run(); cTimer.run();
}
```

		Андрусенко О.М.			КІК.КР.Б – 123 – 01 – ПЗ	Арк.
		Бродський Ю. Б.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Л – Код програмного забезпечення мікроконтролера підсистеми контролю доступу

```
#include <Arduino.h>
#define BLYNK_AUTH_TOKEN "j4ATsshr2i79e0h9l7AXZkTgyuN9ibUS"
#define BLYNK_PRINT Serial
#define SS_PIN 16
#define RST_PIN 15
#define CARDS 1
#define BUTTON 5
#define LOCK 0
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <MFRC522.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "TP-Link_AF52";
char pass[] = "17078078";
MFRC522 mfrc522(SS_PIN, RST_PIN);
String cards[] = {"83 23 38 BB"};
BlynkTimer timer, lockDoorTimer;
WidgetLED doorLed(V0);
void unlockDoor()
{
    doorLed.off();
    Blynk.virtualWrite(V1, 1);
    digitalWrite(LOCK, HIGH);
}
void lockDoor()
{
    doorLed.on();
    Blynk.virtualWrite(V1, 0);
    digitalWrite(LOCK, LOW);
}
BLYNK_WRITE(V1){
    if(param.asInt() == 1){
        unlockDoor();
    }else lockDoor();
}
void waitCard()
{
    if (!mfrc522.PICC_IsNewCardPresent()){return;}
    if (!mfrc522.PICC_ReadCardSerial()){return;}
    String content = "";
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
        content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
        content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
    for (int i = 0; i < CARDS; i++){if (content.substring(1) == cards[i]){
        unlockDoor();}else{lockDoor();}}
}
void setup(){ Serial.begin(115200); SPI.begin(); mfrc522.PCD_Init();
    pinMode(BUTTON, INPUT); pinMode(LOCK, OUTPUT); Blynk.begin(auth, ssid, pass, IPAddress(192,
168, 0, 108), 8080); timer.setInterval(1000L, waitCard);
    lockDoorTimer.setInterval(15000L, lockDoor);}
void loop()
{
    Blynk.run();
    timer.run();
    lockDoorTimer.run();
}
```

		Андрусенко О.М.			<i>KIK.KP.B – 123 – 01 – ПЗ</i>	Арк.
		Бродський Ю. Б.				85
Змн.	Арк.	№ докум.	Підпис	Дата		