

КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

бакалавр

Освітній рівень

Система «Розумний будинок» з віддаленим керуванням

Назва теми

КПТР. 021013.01.02 ПЗ

Шифр

Галузь знань 17 «Електроніка та телекомунікації»

Шифр, назва

Спеціальність 172 «Телекомунікації та радіотехніка»

Шифр, назва

Освітня програма «Телекомунікації, медійні технології та інтелектуальні мережі»

Назва

Виконав: студент IV курсу, група TR2-21-1

du
Підпис

М. С. Зданевич

Ініціали, прізвище

Керівник

С. К. Підченко
Підпис, дата

С. К. Підченко

Ініціали, прізвище

Нормоконтролер

В. І. Стецюк
Підпис, дата

В. І. Стецюк

Ініціали, прізвище

До захисту допускаю:
Зав. кафедри телекомунікацій,
медійних та інтелектуальних
технологій

С. К. Підченко
Підпис

С. К. Підченко

Ініціали, прізвище

« 11 » 06 2025 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра телекомунікацій, медійних та інтелектуальних технологій

Освітній рівень: бакалавр

Галузь знань: 17 «Електроніка та телекомунікації»

Спеціальність: 172 «Телекомунікації та радіотехніка»

Освітня програма: «Телекомунікації, медійні технології та інтелектуальні мережі»

ЗАТВЕРДЖУЮ

Зав. кафедри С. К. Підченко

«10» 02 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНИЙ ПРОЄКТ БАКАЛАВРА

Зданевичу Михайлу Сергійовичу

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи): Система «Розумний будинок» з віддаленим керуванням

Назва теми

Керівник проєкту (роботи) С.К. Підченко, докт. техн. наук., професор

Ініціали, прізвище, науковий ступінь, вчене звання

Затверджена наказом ректора університету від «7» лютого .2025 р. № 90

2. Термін подання студентом проєкту (роботи) на кафедру: «1» 06 .2025 р.

3. Вихідні дані до проєкту (роботи): завдання на кваліфікаційний проєкт

4. Зміст пояснювальної записки (перелік завдань на проєктування):

4.1. Огляд існуючих проєктно-технологічних рішень згідно теми проєкту

4.2. Розробка структури системи «Розумний будинок» та вибір її компонентів



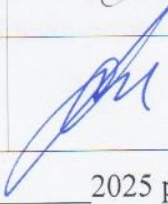

4.3. Апаратно-програмна реалізація системи «Розумний будинок»

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень):

5.1. Схема електрична структурна блоку керування та підсистеми контролю температури і вологості «розумного будинку».

5.2. Схема електрична принципова блоку керування та підсистеми контролю температури і вологості «розумного будинку».

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	В.І. Стецюк, доцент кафедри ТМІТ		
Антиплагіат	О.С. Пивовар, доцент кафедри ТМІТ		

7. Дата видачі завдання «10» 02 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи «Розумний будинок»	01.04.2025	виконано
5	Робота над розділом 3 – проектування системи «Розумний будинок»: розробка схеми електричної структурної, схеми електричної принципової та програмного забезпечення.	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКП	29.05.2025	виконано
8	Захист ВКП на засіданні ЕК	червень 2025 року	

Студент


Підпис

М. С. Зданевич

Ініціали, прізвище

Керівник роботи


Підпис

С. К. Підченко

Ініціали, прізвище

№ рядка	Формат	Позначення	Найменування	Кл-сть. листів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КПТР. 021013.01.02 ПЗ	Пояснювальна записка	56		
			<u>Графічні матеріали</u>			
2	A2	КПТР. 021013.01.02 Е1	Структурна схема блоку керування та підсистеми контролю температури та вологості	1		
3	A2	КПТР. 021013.01.02 Е3	Принципова схема блоку керування та підсистеми контролю температури та вологості	1		
4	A4		Копія схема електричної структурної	1		Копія А2
5	A4		Копія схема електричної принципової	1		Копія А2

КПТР. 021013.01.02 ВП

Зм.	Арк.	№ докум.	Підпис	Дата	Літера	Аркуш	Аркушів
Розробив		М.С.Зданевич	<i>М.С.Зданевич</i>	10.06.25	н	1	1
Перевірив		С.К. Підченко	<i>С.К. Підченко</i>	10.06.25			
Н. контр.		В.І. Стецюк	<i>В.І. Стецюк</i>	12.08.25			
Затвердив		С.К. Підченко	<i>С.К. Підченко</i>	11.06.25			

Система
«Розумний будинок»
з віддаленим керуванням
Відомість проекту

ХНУ, ТР2-21-1

АНОТАЦІЯ

Тема кваліфікаційного проєкту: Система «Розумний будинок» з віддаленим керуванням

Автор роботи: Зданевич Михайло Сергійович

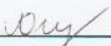
Керівник роботи: Підченко Сергій Костянтинович

Пояснювальна записка: 56 с., 38 рис., 6 табл., 3 дод., 28 джерел

Графічна частина: 2 креслення

РОЗУМНИЙ БУДИНОК, СИСТЕМА, ВІДДАЛЕНЕ КЕРУВАННЯ, ТЕЛЕКОМУНІКАЦІЇ, АВТОМАТИКА

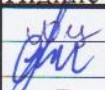
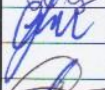

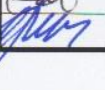
Розроблено структуру системи «Розумний будинок», яка складається з блоку керування, мережевого інтерфейсу з безпроводовим каналом зв'язку та сенсорно-актуаторною частиною для контролю та керування температурою та вологістю в кімнаті «Розумного будинку». Розроблено схему електричну принципову блоку керування та підсистеми моніторингу і контролю вологості і температури на базі мікроконтролера ESP32, цифрового сенсора DHT22 та релейно-комутаційної частини з оптронною розв'язкою для керування вентиляторами та зволожувачем повітря. Розроблено програмне забезпечення для блоку керування, модуля віддаленого доступу на базі протоколу MQTT, а також серверну частину веб-орієнтованого інтерфейсу користувача для налаштування параметрів системи та моніторингу даних сенсорів. Розроблений прототип системи «Розумний будинок» з віддаленим керуванням для Інтернету речей може бути використаний для тестування, демонстрації та подальшої розробки подібних систем, а також в навчальному процесі для здобувачів освіти за спеціальністю 172 – Телекомунікації та радіотехніка.


Підпис студента

09.06.25
Дата

ЗМІСТ

Скорочення та умовні позначення.....	3
Вступ.....	4
1 Огляд існуючих проектно-технологічних рішень згідно теми проекту	5
1.1 Загальні відомості про автоматизовані домашні системи на прикладі системи «Розумний будинок»	5
1.2 Огляд відкритих літературних джерел та доступного обладнання	8
1.3 Постановка технічного завдання	17
1.4 Висновки до 1-го розділу	17
2 Розробка структури системи «Розумний будинок» та вибір її компонентів	18
2.1 Розробка структури системи «Розумний будинок»	18
2.2 Обґрунтування вибору компонентів системи	20
2.3 Формалізація процесу функціонування системи та розробка алгоритму програмного забезпечення.....	30
2.4 Висновки до 2-го розділу	34
3 Апаратно-програмна реалізація системи «Розумний будинок».....	35
3.1 Розробка принципової схеми пристроїв системи «Розумний будинок»	35
3.2 Реалізація віддаленого керування на базі протоколу MQTT.....	36
3.3 Реалізація програмних компонентів системи «Розумний будинок»	39
3.4 Висновки до 3-го розділу	52
Загальні висновки до проекту	53
Перелік джерел посилання	54
Додаток А Копія схеми електричної структурної.....	57
Додаток Б Копія схеми електричної принципової.....	58
Додаток В Код програмного забезпечення.....	59

КПТР. 021013.01.02 ПЗ								
Зм.	Арк.	Недокум.	Підпис	Дата	Система «Розумний будинок» з віддаленим керуванням Пояснювальна записка	Літера	Арк.шл.	Арк.шпів
Виконав		М.С. Зданевич		10.06.25		Н		55
Перевір.		С.К. Підченко		10.06.25				
Н.контр.		В.І. Стецюк		11.06.25				
Затвер.		С.К. Підченко		11.06.25			ХНУ, ТР2-21-1	

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API	– application programming interface / інтерфейс програмування
IOT	– Internet of things / Інтернет речей
MQTT	– message queuing telemetry transport
MQTT	– message queuing telemetry transport
MVC	– model-view-controller / модель-представлення-контролер
ORM	– object-relational mapping
SOC	– system on a chip / система на кристалі
UART	– universal asynchronous receiver/transmitter
ULP	– ultralow power / наднизьке енергоспоживання
АЦП	– аналого-цифровий перетворювач
БЖ	– блок живлення
БК	– блок керування
ЕОМ	– електронно обчислювальна машина
ІКТ	– інформаційно-комунікаційні технології
МК	– мікроконтролер
МП	– мікропроцесор
ОЗП	– оперативно запам'ятовуючий пристрій
ПЗ	– програмне забезпечення
ПЗП	– постійний запам'ятовуючий пристрій
САВ	– сенсорно-актуаторний вузол
ТТХ	– тактико-технічні характеристики
ЦАП	– цифро-аналоговий перетворювач
ШИМ	– широтно-імпульсна модуляція

ВСТУП

Впровадження систем автоматизації в домашні приміщення сприяє підвищенню безпеки та комфорту проживання людей. Сучасні інформаційні технології та елементна база вимірювальних та обчислювальних компонентів сприяють ефективній та економічно доцільній реалізації інтелектуальних кіберфізичних систем за типом «Розумний будинок» для Інтернету речей.

В рамках виконання кваліфікаційного проєкту бакалавра за спеціальність 172 – Телекомунікації та радіотехніка було обрано тему проєктування, що стосується розробки системи «Розумний будинок» з віддаленим керуванням для Інтернету речей.

Метою проєкту є розробка прототипу системи «Розумний будинок», що складається з мікроконтролерного блоку керування та підсистеми контролю температури і вологості в приміщенні з можливістю віддаленого моніторингу і керування через веб-орієнтований інтерфейс користувача.

Задачі проєктування:

1. Розробка структури системи «Розумний будинок», яка включає апаратну частину, ПЗ, а також та комунікаційні канали з можливістю гнучкого розширення за рахунок підключення нових підсистем в процесі експлуатації. В якості базової підсистем розробити: підсистему контролю температури та вологості в приміщенні.

2. Розробка схеми електричної принципової блоку керування на базі МК для системи «Розумний» будинок, а також підсистеми керування температурою та вологістю, яка включає в себе сенсорну частину та комутаційні механізми для керування вентиляцією та зволожувачем повітря.

3. Розробка алгоритму керування та ПЗ бля компонентів системи «Розумний будинок» та підсистеми контролю температури і вологості. Розробити ПЗ для блоку керування та проміжних обчислювальних вузлів, базу даних (БД) вимірюваних величин та верхній рівень інтерфейсу користувача для віддаленого керування системою.

									Арк.
									4
Зм.	Арк.	№ докум.	Підпис	Дата	КПТР. 021013.01.02 ПЗ				

1 ОГЛЯД ІСНУЮЧИХ ПРОЄКТНО-ТЕХНОЛГІЧНИХ РІШЕНЬ ЗГІДНО ТЕМИ ПРОЄКТУ

1.1 Загальні відомості про автоматизовані домашні системи на прикладі системи «Розумний будинок»

Автоматизовані технології використовуються все частіше в домашніх приміщеннях з метою покращення умов життя людей, підвищення безпеки та загального комфорту тощо. Розвиток інформаційно-комунікаційних технологій (ІКТ) сприяв проникненню розумних малопотужних електронних обчислювальних машин (ЕОМ) та безпроводових засобів передачі інформації у сфери домогосподарства. З іншого боку, розвиток комп'ютерної техніки та мережевих технологій, на прикладі глобальної системи Інтернет, посприяв виникненню концепції «Інтернету речей – Internet Of Things, (IoT) – мережі розумних пристроїв на базі мікропроцесорів (МП) та мікроконтролерів (МК), сенсорно-актуаторних вузлів (САВ) та мережевих інтерфейсів. Відповідно до прийнятої архітектури IoT [1-2], виділяються такі рівні ієрархії:

1. Рівень «розумних» пристроїв, що забезпечуються збір та обробку даних об'єктів вимірювання та контролю в реальному часі, а також керування об'єктом – актуаторна функція.

2. Мережевий рівень та шлюз для інтеграції із протоколами Інтернет.

3. Рівень послуг щодо віддаленого керування пристроями IoT, засоби інтеграції x прикладним програмним забезпеченням (ПЗ), прикладні інтерфейси програмування – Application programming interface (API).

4. Прикладний рівень – верхній рівень додатків для використання кінцевим користувачем (оператором) системи.

На Рисунку 1.1 показані деякі головні компоненти IoT та взаємні потоки інформації між ними відповідно до вищеописаної ієрархії. Так. Потоки даних надходять на Контролер, де обробляються в реальному часі (попередня обробка),

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

потім оброблені та стиснені дані передаються Серверу обробки для здійснення, наприклад, інтелектуального аналізу даних; дані акумулюються у Сховищі.



Рисунок 1.1 – Схема інформаційних процесів між компонентами мережі IoT [2]

Розглянемо в якості прикладу IoT систему «Розумний будинок» відповідно до тематики кваліфікаційного проєкту. Відомо [3], що даний термін вперше був запропонований асоціацією забудовників у США в 1984 році в для позначення замиського будинку, відмінною особливістю якого є ефективне використання житлового середовища. Відповідно до визначення, наведеного у бібліографічному покажчику [3], «Розумний будинок» – це сучасне житлове приміщення, яке інтегроване із сучасними засобами автоматизації, інфокомунікацій та інтелектуальної підтримки. Типовими компонентами для такої системи є модуль кліматконтролю, засоби безпеки (наприклад, відеоспостереження), підсистема контролю освітлення, засоби захисту від

Зм.	Арк.	№ докум.	Підпис	Дата

несанкціонованого проникнення (розумні замки), а також засоби аудіовізуальної підтримки. Концепція системи «Розумного будинку» [4] показана на Рисунку 1.2.

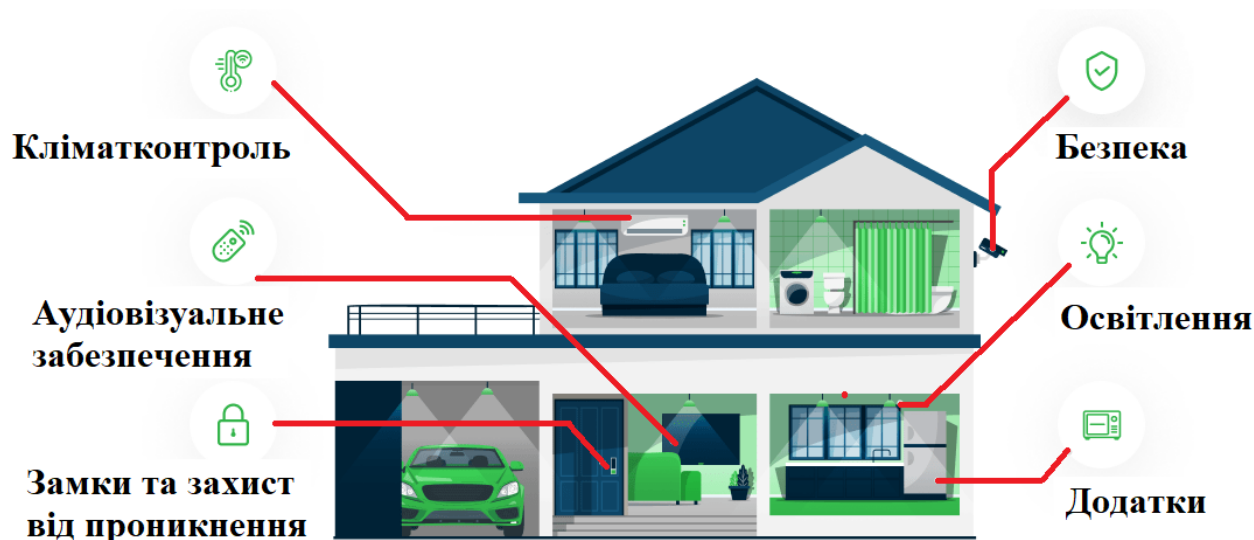


Рисунок 1.2 – Концепція «Розумного будинку» [4]

Підсистема керування кліматом в приміщенні функціонує з метою підтримки оптимального рівня температури повітря, вологості, рівня притоку свіжого повітря тощо.

До головних функцій підсистеми освітлення належить підтримка бажаного рівня освітленості в житловому приміщенні для заданого часу доби, балансування між штучними та природними джерелами світла, керування рівнем освітленості.

Важливим компонентом «Розумного будинку» є підсистема безпеки, яку умовно можна розділити на дві складові: систему розумних замків та захисту від проникнення, а також систему відеоспостереження та контролю за людьми в приміщенні, зокрема за допомогою сенсорів руху.

Підсистема аудіовізуалізації призначена для підвищення комфорту мешканців, забезпечення дозвілля, а також розвитку дітей.

Окремо варто відзначити набір додатків для керування системою оператором (головним авторизованим користувачем системи з відповідними правами доступу).

1.2 Огляд відкритих літературних джерел та доступного обладнання

Тематиці «Розумного будинку» присвячено досить багато відкритих публікацій. Так, наприклад в роботі [5] проведено огляд існуючих проектів даного типу та дано прогноз тенденції розвитку концепції.

Проект системи «Розумний будинок» з реалізацією підсистеми контролю температури на базі сенсора DHT22 з блоком керування на базі МК ATmega328 запропоновано в роботі [6]. Також в роботі представлено програмне забезпечення високого рівня в середовищі Microsoft Visual Studio на мові програмування C#.

В статті [7] авторкою представлена власна концепція побутових складових «Розумного будинку» на прикладі систем керування в квартирі, офісі і в будинку. Виділено методи керування: автоматичне керування на базі сенсорів, керування «Розумним будинком» за допомогою пульта дистанційного керування, а також віддалене керування. Також виділяються декілька реалізацій концепції «Розумного будинку» з різними рівнями інтеграції, а саме: вбудовані системи з центральним МК та без нього, інтегровані системи до налаштування [7].

В роботі [8] авторами представлена розробка «Розумного будинку» з підвищеною надійністю компонентів системи на базі платформи Arduino. Запропонована архітектура побудована на базі МК Arduino, який виконує функції пристрою-посередника між апаратною частиною на ПЗ на відконфігурованому для віддаленого керування персональному комп'ютері (ПК) або сервері (див. Рисунок 1.3). Авторами також запропоновано декілька методів підвищення надійності системи, а саме «гаряче», «холодне» та «тепле» резервування, резервування ліній живлення та передачі даних. Показана необхідність «холодного» резервування у випадку віддаленого керування – коли головний МК виходить з ладу його функції покладаються на резервний МК. Високі показники надійності показані для «гарячого резерву», коли головний та

									Арк.
									8
Зм.	Арк.	№ докум.	Підпис	Дата	КПТР. 021013.01.02 ПЗ				

резервний МК працюють одночасно, що в свою чергу збільшує загальне енергоспоживання системи. Використання «теплого резервування» також відбувається затримка при умиканні резервного МК. Таким чином, авторами роботи [8] запропонована комбінація вищеназваних методів резервування критичних компонентів системи «Розумний будинок».

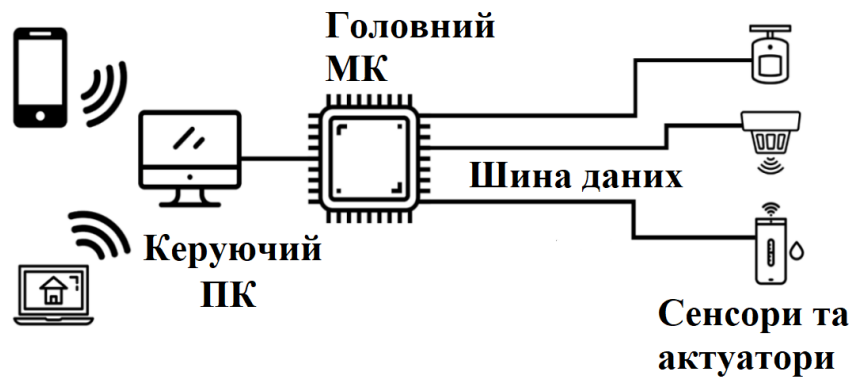


Рисунок 1.3 – Архітектура системи «Розумник будинок» на базі МК [8]

В статті [9] розглядається сучасні системи «Розумних будинків» для квартир, офісів та готельних комплексів, а також основні напрямки автоматизації, керування підсистемами освітленням, кліматом, шторами, управління аудіо/відео, а також відповідне ПЗ. Запропоновано схеми централізованої, децентралізованої та комбінованої автоматизації (Рисунки 1.4-1.6). Відповідно до схеми на Рисунку 1.4, централізована автоматизація полягає у використанні єдиного МК з максимальною кількістю виводів для підключення пристроїв. Перевагами централізованої схеми є можливість підключення обладнання різного типу та керування пристроями через загальний інтерфейс, а також можливість розробки циклічних сценаріїв. Недоліком такої схеми є недостатня надійність та підвищена вразливість системи до виходу з ладу головного МК. Децентралізована схема (Рисунок 1.5) частково вирішує цю проблему за рахунок впровадження мережі периферійних МК, кожен з яких виконує свою функцію, або набір функції відповідно до задач підсистеми, яку він представляє. Збільшення кількості МК звичайно призводить до ускладнення

системи, збільшення її вартості, а також підвищення енергоспоживання. Тому оптимальним вважається використання комбінованої схеми автоматизації, яка полягає у поєднанні двох вищеописаних схем [9].

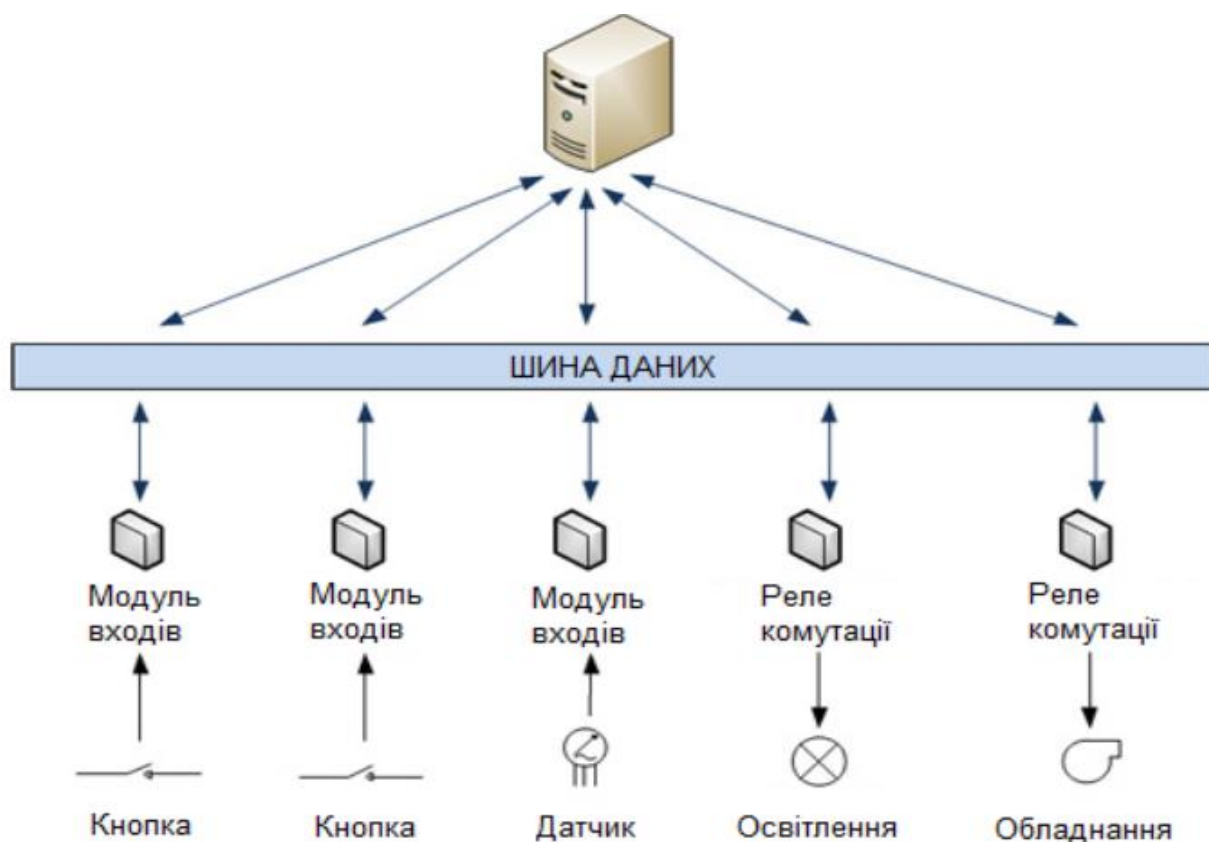


Рисунок 1.4 – Схема централізованої автоматизації «Розумного будинку» [9]

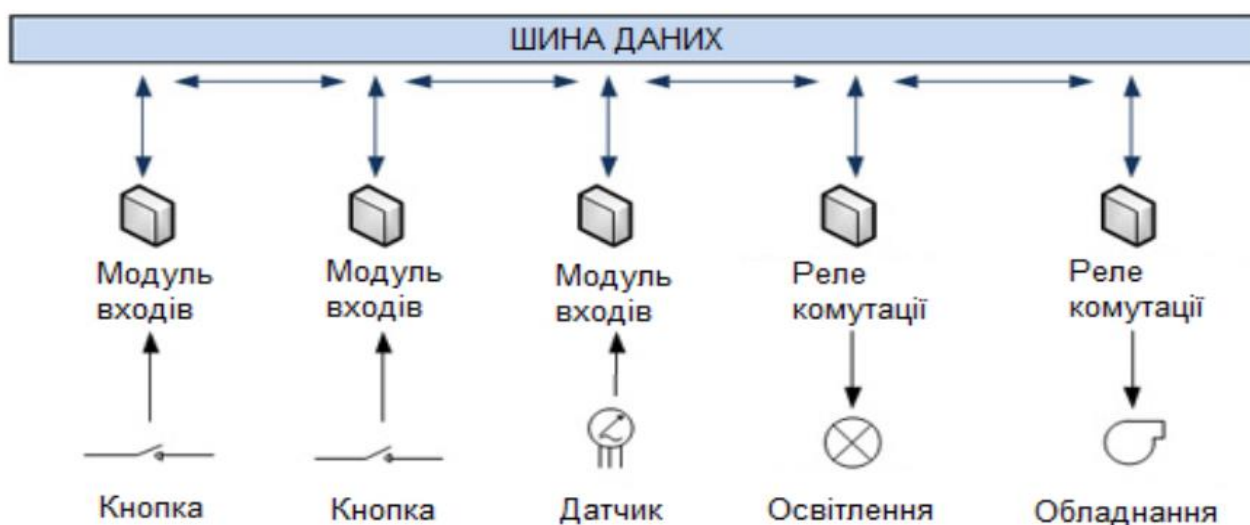


Рисунок 1.5 – Схема децентралізованої автоматизації «Розумного будинку» [9]

Зм.	Арк.	№ докум.	Підпис	Дата

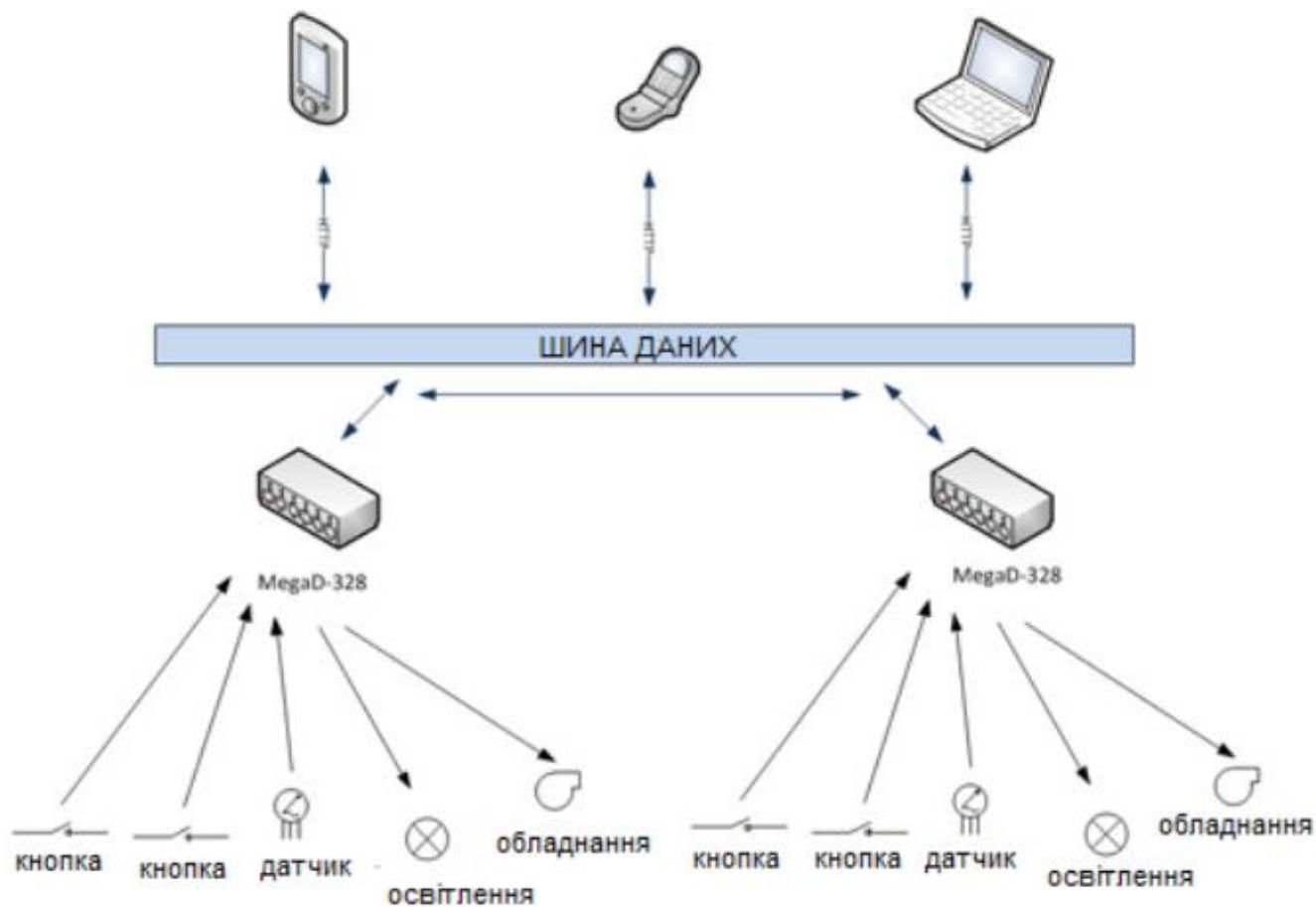


Рисунок 1.6 – Схема комбінованої автоматизації «Розумного будинку» [9]

В статті [9] також наводяться приклади типового сучасного обладнання, яке використовується для побудови систем «Розумного будинку». До таких пристроїв, зокрема, належать [9]

- сенсори руху,
- модулі релейних контактів,
- сенсори протікання води,
- сенсори диму та вогню,
- пристрій регулювання потужності (диммер),
- електронний блок керування.

Наведемо приклади обладнання для використання в системах «Розумних будинків», доступного на ринку України станом на травень 2025 року.

На Рисунку 1.7 показано інфрачервоний сенсор руху Євросвітло SB-01 360° [10] з кутом огляду 360°, призначений для автоматичного вмикання

освітлення при виявленні руху в зоні дії. Він встановлюється на стелю та ефективно працює в приміщеннях або на відкритому повітрі під захистом від вологи. Модель SB-01 дозволяє регулювати рівень освітленості для активації, а також час затримки вимкнення світла, що робить її зручною для енергозбереження й автоматизації освітлення. Технічні характеристики сенсора наведені в Таблиці 1.1.



Рисунок 1.7 – Сенсор руху Євросвітло SB-01

Таблиця 1.1 – Технічні характеристики сенсора руху SB-01 360°

Характеристика	Значення
Тип	Інфрачервоний
Тип монтажу	Вбудований
Відстань виявлення	До 6 м
Клас захисту IP	IP20
Кут виявлення	360°
Монтажні розміри	6 на 6 см
Вартість	369 грн.

Модуль на 8 релейних виходів МРЛ-8 із перекидними контактами реле та інтерфейсом RS-485 [11] показаний на Рисунку 1.8. Це пристрій із вісьмома

незалежними релейними виходами з перекидними контактами, призначений для керування різними електричними навантаженнями в системах автоматизації. Кожне реле може працювати як перемикач, що забезпечує гнучкість у підключенні. Модуль оснащений інтерфейсом RS-485 і підтримує протокол Modbus RTU, що дозволяє інтегрувати його в різноманітні системи управління. Завдяки компактному виконанню та надійності, МРЛ-8 широко використовується в промислових і будівельних автоматизованих комплексах. Основні технічні характеристики модуля МРЛ-8 подані в Таблиці 1.2.



Рисунок 1.8 – Релейний модуль на 8 виходів МРЛ-8 з інтерфейсом RS-485

Таблиця 1.2 – Технічні характеристики релейного модуля МРЛ-8

Характеристика	Значення
Кількість контактів	8
Змінний струм/напруга комутації	не більше 5 А / 240 В
Постійний струм/напруга комутації	не більше 5 А / 24 В.
Напруга живлення	9,5-14,5 В
Струм споживання в режимі	не більше 310 мА.
Монтажні розміри	230 мм на 145 мм на 35мм.
Маса	1,2 кг
Вартість	2 379 грн.

Сенсор контролю протікання води Mastino WS2 [12] показаний на Рисунку 1.9. Це компактний пристрій, призначений для виявлення витoku води в системах водопостачання та опалення. Він реагує на появу вологи в зоні встановлення та передає сигнал на контролер або систему безпеки для своєчасного реагування. Модель WS2 відзначається простотою монтажу, високою чутливістю та сумісністю з більшістю систем «Розумного будинку». Основні технічні характеристики сенсора WS2 подані в Таблиці 1.3.



Рисунок 1.9 – Сенсор контролю протікання води Mastino WS2

Таблиця 1.3 – Технічні характеристики сенсора протікання води WS2

Характеристика	Значення
Габаритні розміри	53 на 12 мм
Довжина дроту	10 м
Напруга живлення	До 12 В
Максимальна відстань до МК	100 м
Матеріал корпусу	нержавіюча сталь AISI 304
Температурний діапазон середовища	+5С...+40 °С
Вартість	758 грн.

Розглянемо в якості прикладу сенсора диму універсальний спеціальний сенсор диму в корпусі з міцного білого пластику СПД-3.2 (Рисунок 1.10). Прилад представляє собою провідний оптичний точковий димовий пожежний сповіщувач, призначений для раннього виявлення загорянь у закритих приміщеннях, що супроводжуються появою диму. Він працює на основі інфрачервоного розсіювання світла: при потраплянні диму в камеру змінюється інтенсивність світлового сигналу, що фіксується фотодатчиком, після чого формується сигнал тривоги [13].



Рисунок 1.10 – Сенсор диму СПД-3.2

Таблиця 1.4 – Технічні характеристики сенсора диму СПД-3.2

Характеристика	Значення
Індикатор режиму очікування	Так
Кількість контактів бази	4
Напруга живлення	10-30 В
Розміри	100 на 48 мм
Струм споживання в режимі очікування	До 0.15 мА
Інерційність	До 10 с
Вартість	230 грн.

Розглянемо в якості прикладу промислового блоку керування пристрій CoolMasterNet (Рисунок 1.11). Це універсальний інтерфейс для інтеграції VRF/VRV систем кондиціонування від провідних виробників (Daikin, Mitsubishi, LG тощо) з системами автоматизації будівель (BMS), розумними будинками та хмарними сервісами. Пристрій забезпечує двосторонній зв'язок за протоколами BACnet, Modbus, REST API та MQTT, що дозволяє централізовано керувати кліматом, моніторити стан обладнання, отримувати коди помилок і здійснювати віддалений доступ через хмарну платформу CoolAutomation.



Рисунок 1.11 – Блок керування системи кліматконтролю CoolMasterNet

Отже, в результаті аналізу ринку комплектуючих для системи «Розумний будинок», можна зробити висновок щодо достатньої кількості готових повнофункціональних модулів, що використовують уніфіковані інтерфейси взаємодії і можуть бути використані для організації таких систем. Однак, для спеціалізованих задач доцільним є розробка нових пристроїв із покращеними тактико-технічними характеристиками (ТТХ), окрема підвищеними показниками надійності та економічної доцільності.

1.3 Постановка технічного завдання

Для досягнення поставленої мети проектування до вирішення приймаються такі задачі:

1. Розробка структури системи «Розумний будинок», яка включає апаратну частину, ПЗ, а також та комунікаційні канали з можливістю гнучкого розширення за рахунок підключення нових підсистем в процесі експлуатації. В якості базової підсистем розробити: підсистему контролю температури та вологості в приміщенні.

2. Розробка схеми електричної принципової блоку керування на базі МК для системи «Розумний» будинок, а також підсистеми керування температурою та вологістю, яка включає в себе сенсорну частину та комутаційні механізми для керування вентиляцією та зволожувачем повітря.

3. Розробка алгоритму керування та ПЗ бля компонентів системи «Розумний будинок» та підсистеми контролю температури і вологості. Розробити ПЗ для блоку керування та проміжних обчислювальних вузлів, базу даних (БД) вимірюваних величин та верхній рівень інтерфейсу користувача для віддаленого керування системою.

1.4 Висновки до 1-го розділу

В 1-му розділі кваліфікаційного проєкту було викладено загальну концепцію систем типу «Розумний будинок», проведено аналіз відкритих джерел на дану тематику, а також розглянуто обладнання, яке може бути використане для реалізації компонентів «Розумного будинку».

Відповідно до поставленої мети, сформульовані головні задачі проектування.

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

2 РОЗРОБКА СТРУКТУРИ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» ТА ВИБІР ЇЇ КОМПОНЕНТІВ

2.1 Розробка структури системи «Розумний будинок»

Відправним кроком розробки структури системи «Розумний будинок» є проектування блоку її керування (БК). Це головний компонентом системи, який базується на МК з можливістю обробки даних, прийняття рішень та комунікації з іншими пристроями. Побудований на базі МК, наприклад, ESP32 або STM32, БК використовується для зчитування інформації з сенсорів, аналізу отриманих даних та керування виконавчими пристроями.

З метою завантаження та налагодження ПЗ необхідно передбачити інтерфейс програмування, наприклад UART або JTAG. В той же час наявність мережевого інтерфейсу TCP/IP необхідна для підключення до локальної мережі та Інтернету, для здійснення віддаленого моніторингу і керування через додаток або веб-інтерфейс. Для реалізації віддаленого доступу необхідно також передбачити безпроводовий канал зв'язку, який реалізований, наприклад, за допомогою радіомодуля Wi-Fi, що також забезпечує інтеграцію з іншими IoT-пристроями.

Отже, БК системи складається з таких елементів: МК, інтерфейс програмування, інтерфейсу програмування та налагодження, мережевого інтерфейсу TCP/IP та радіомодуля Wi-Fi.

Наступним кроком проектування структури системи є розробка підсистема контролю температури та вологості, яка відповідає за моніторинг кліматичних параметрів і автоматичне регулювання. Така підсистема включає сенсори температури та вологості, наприклад DHT22 або SHT31, що вимірюють поточні значення та передають їх до МК. На основі отриманих даних система керує релейно-комутаційним модулем, який активує або деактивує вентилятори та зволожувач повітря для підтримки оптимального мікроклімату. Отже,

підсистема контролю температури та вологості буде складатись з таких компонентів: сенсор вологості та сенсор температури, релейно-комутаційний інтерфейс для керування вентиляторами та зволожувачем повітря.

Для живлення схеми необхідно реалізувати блок живлення (БЖ), що може бути побудований на базі лінійного стабілізатора, наприклад, AMS1117-3.3.

З урахуванням вищесказаного, представимо на Рисунку 2.1 структурну схему БК системи «Розумний будинок», а також підсистему контролю температури та вологості.

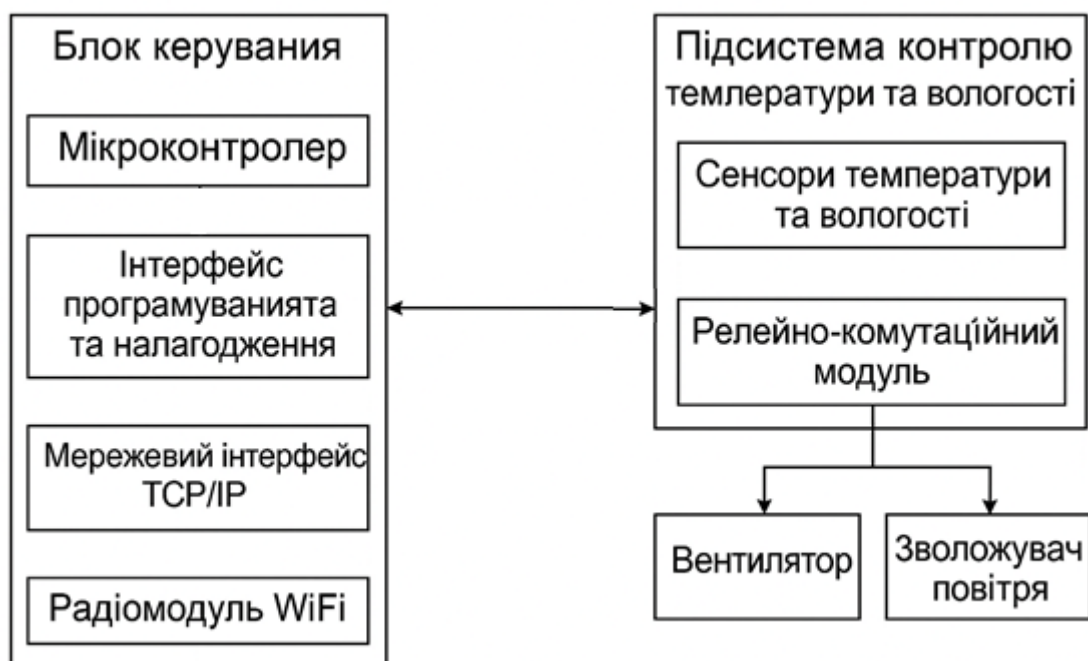


Рисунок 2.1 – Структурна схема блоку керування «Розумного будинку» та підсистеми контролю температури та вологості

З метою розширення функціоналу система також може бути доповнена функціями віддаленого моніторингу через протокол Message Queuing Telemetry Transport (MQTT) або спеціалізований API, інтеграцією з голосовими помічниками та функцією журналювання даних для подальшого аналізу. Завдяки модульній архітектурі її можливо потенційно розширювати, додаючи нові сенсори або виконавчі пристрої (актуатори).

2.2 Обґрунтування вибору компонентів системи

Для практичної реалізації системи «Розумний будинок» згідно розробленої в попередньому розділі структурної схеми необхідно вибрати компоненти системи серед наявної елементної бази. Доцільним вважається використання МК для розробки на його базі БК системи. Відповідно до структури, що показана на Рисунку 2.1, необхідно передбачити також наявність мережевого інтерфейсу, радіомодуля, а також інтерфейсу програмування. Враховуючи це, для реалізації БК включно із компонентами МК, мережевого інтерфейсу та модуля Wi-Fi було обрано платформу ESP32, що представляє собою систему на кристалі – System on a Chip (SoC) з інтегрованим безпроводовим зв'язком [15]. Блок-схема МК ESP32 показана на Рисунку 2.2.

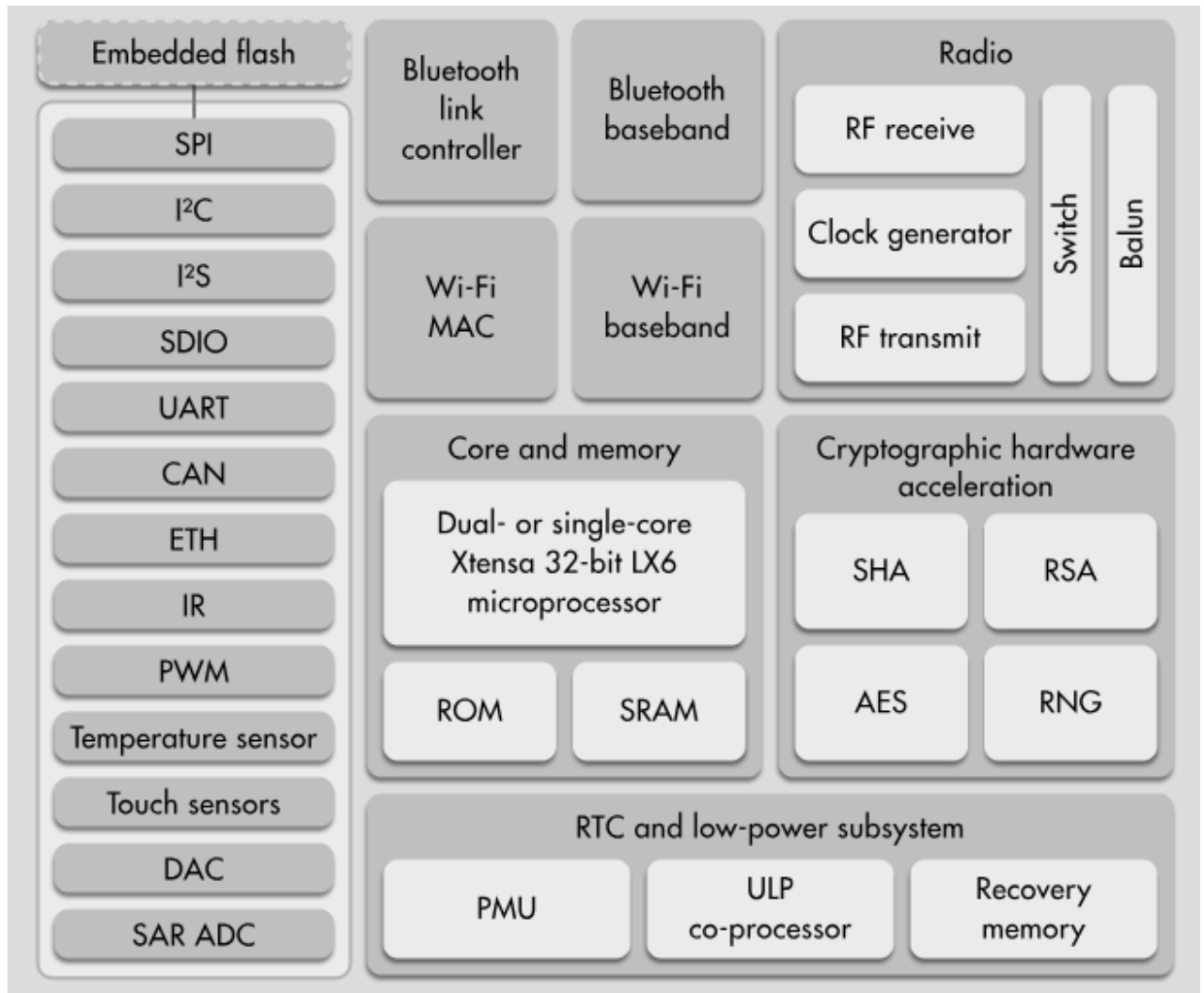


Рисунок 2.2 – Блок-схема мікроконтролера ESP32 [16]

Відповідно до структури, ESP32 є високоінтегрованою SoC – системою, яка поєднує потужні обчислювальні можливості з енергоефективністю та безпроводовим зв'язком. Його архітектура базується на двох 32-бітних ядрах Xtensa LX6 з тактовою частотою до 240 МГц, кожне з яких має власну кеш-пам'ять для інструкцій та даних. Ці ядра можуть працювати незалежно або спільно, забезпечуючи гнучкість у різних сценаріях використання. Для зберігання завантажувача та базової прошивки МК містить 448 Кб вбудованої ПЗП-пам'яті, а також 520 Кб ОЗП, що поділяється на основну оперативну пам'ять об'ємом 328 Кб та RTC-пам'ять об'ємом 192 Кб, яка зберігає дані режимі зниженого енергоспоживання. Також МК підтримує підключення зовнішньої flash-пам'яті через інтерфейси SPI або QSPI. Важливою складовою ESP32 є співпроцесор з наднизьким енергоспоживанням (ULP), який залишається активним навіть у режимі глибокого сну, споживаючи при цьому близько 150 мкА. Він може виконувати прості задачі, такі як опитування датчиків або перевірка таймерів, і пробуджувати основні ядра лише за необхідністю, що робить ESP32 ідеальним вибором для пристроїв із батарейним живленням. Безпроводові комунікації забезпечуються вбудованими модулями Wi-Fi 802.11 b/g/n у діапазоні 2,4 ГГц та Bluetooth 4.2 з підтримкою класичного режиму та BLE. Отже, МК може працювати як точка доступу, клієнт або в двох режимах одночасно. Крім того ESP32 має розширений набір периферійних інтерфейсів, включаючи до 34 програмованих GPIO, 12-бітний АЦП з 18 каналами, 8-бітний ЦАП на 2 канали, 4 64-бітних таймери, інтерфейси SPI, I2C, UART, CAN 2.0, Ethernet MAC, а також спеціалізовані модулі для керування двигунами з широтно-імпульсною модуляцією (ШИМ) та світлодіодами (LED PWM). Додатково доступні 10 сенсорних входів для ємнісних сенсорів. Система керування живленням дозволяє перемикатися між різними режимами енергоспоживання: активним, модемним сном, легким сном та глибоким сном. У режимі глибокого сну споживання струму знижується до 5 мкА завдяки роботі лише RTC-блока та ULP-співпроцесора. Функції безпеки ESP32 включають

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

апаратне прискорення шифрування AES, SHA-2, RSA, генератор випадкових чисел, захищене завантаження та шифрування флеш-пам'яті. Це робить його придатним для застосувань, де важлива конфіденційність даних. Архітектура ESP32 забезпечує оптимальний баланс між продуктивністю, енергоефективністю та функціональністю, що обґрунтовує вибір даного МК в якості бази для БК системи «Розумний будинок».

В якості інструменту для програмування та налагодження використовується інтерфейс JTAG [17], що є стандартним механізмом для програмування, налагодження та тестування МК, включаючи ESP32. Він забезпечує низькорівневий доступ до внутрішніх компонентів системи, таких як процесорні ядра, пам'ять та периферійні пристрої, через спеціальний 5-контактний інтерфейс (TMS, TCK, TDI, TDO, TRST). Структура JTAG-сумісного пристрою показана на Рисунку 2.3.

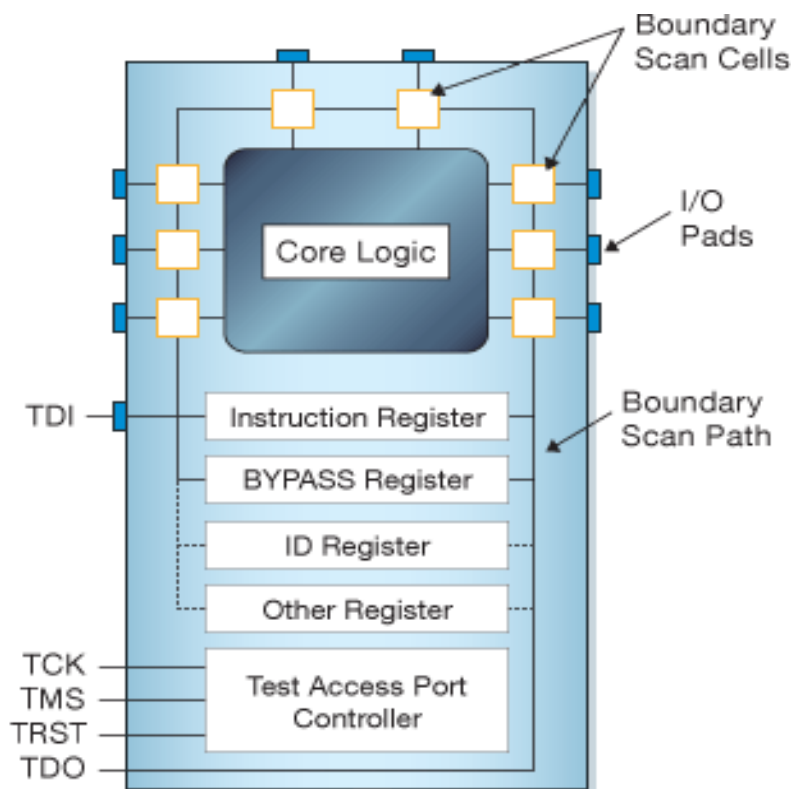


Рисунок 2.3 – Структурна схема інтерфейсу JTAG [17]

У випадку з ESP32, JTAG дозволяє розробникам виконувати покрокове виконання коду, аналізувати стан регістрів, встановлювати точки зупини та

читати/записувати дані у пам'ять без необхідності завантажувати додатковий код. Для роботи з JTAG на ESP32 необхідно підключити спеціальний пристрій налагодження, наприклад, Segger J-Link [18] або ESP-Prog [19], до відповідних виводів МК.

В якості сенсорного вузла для вимірювання та контролю температури і вологості в кімнаті скористаємось цифровим сенсором DHT22 [20], який також відомий як AM2302 і широко використовується в різноманітних проектах IoT, системах моніторингу клімату та домашній автоматизації. Він відрізняється високою точністю вимірювань: діапазон визначення вологості становить від 0 до 100% з похибкою $\pm 2\%$, а температура вимірюється в межах від -40 до $+80^{\circ}\text{C}$ з точністю $\pm 0,5^{\circ}\text{C}$. Сенсор працює за принципом ємнісного визначення вологості та термістора для вимірювання температури, а вбудований АЦП забезпечує цифровий вихідний сигнал, що спрощує його інтеграцію з МК. На рисунку 2.4 зображено фото сенсора DHT22 з позначенням виводів (Рисунок 2.4, а) та сенсорного модуля на його базі (Рисунок 2.4, б).

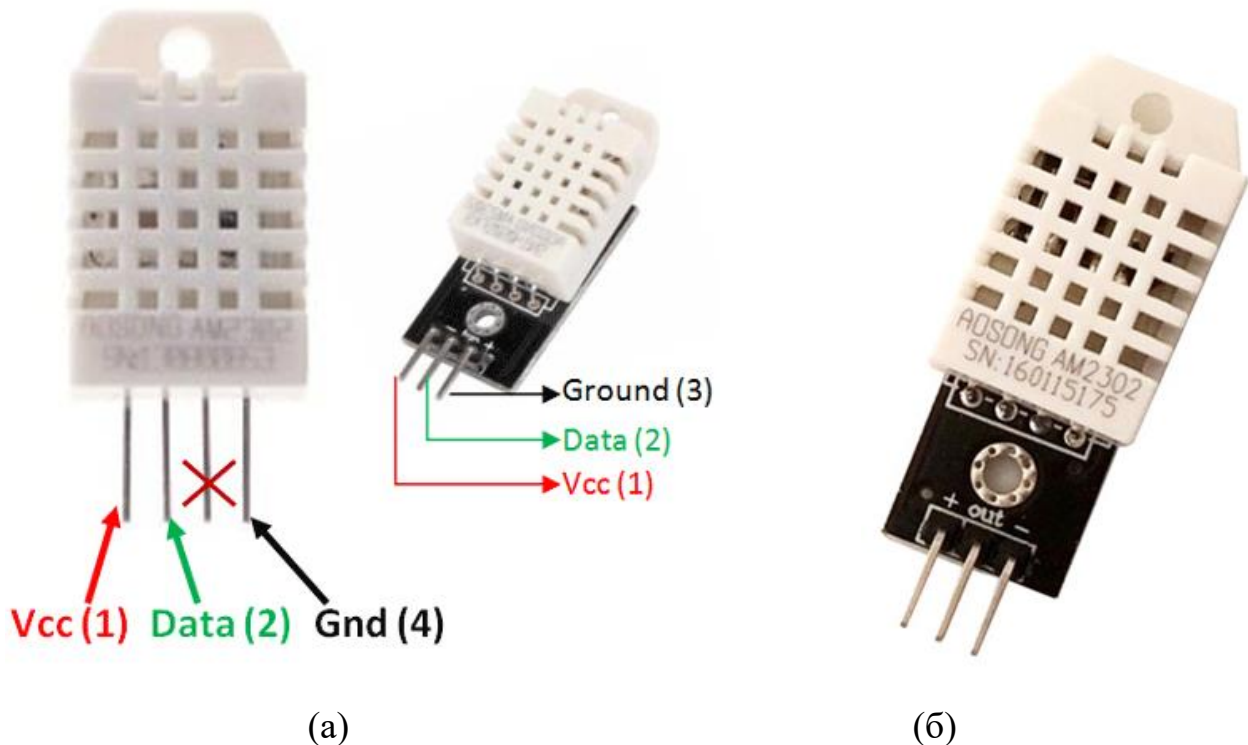


Рисунок 2.4 – Сенсор температури та вологості: окремо сенсор та позначення виводів (а), у вигляді модуля розширення (б)

Як видно з Рисунку 2.4, сенсор DHT22 використовує три виводи: живлення Vcc, загальний вивід Gnd та інформаційний вивід Data (цифровий). На Рисунку 2.5 показана функціональна схема підключення DHT22 до МК.

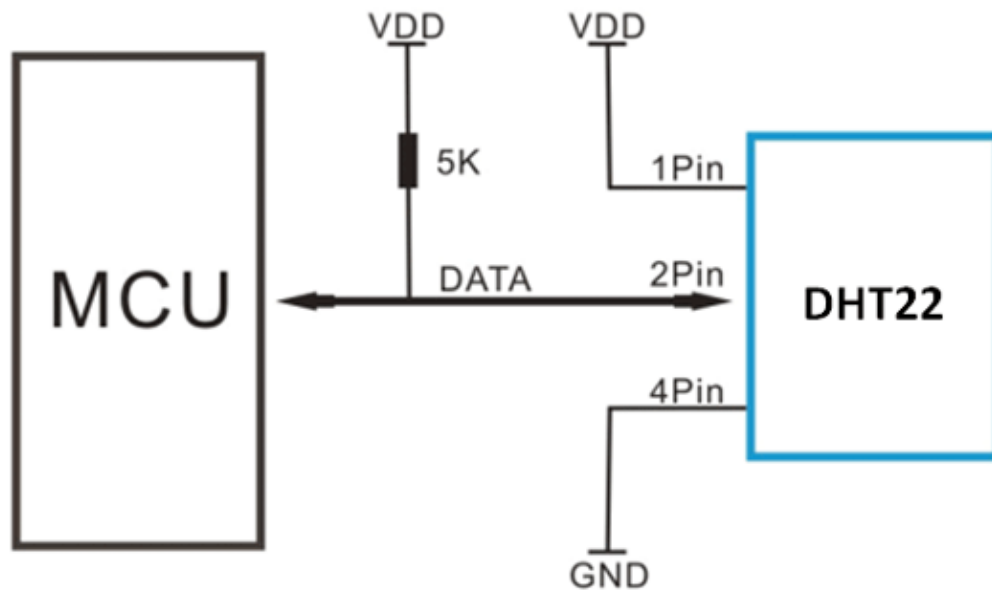


Рисунок 2.5 – Підключення DHT22 до МК [20]

Для виконання релейно-комутаційного модуля необхідно реалізувати перемикання силових реле сигналами з виводів МК. Для цього необхідно передбачити гальванічну розв'язку із забезпеченням окремого живлення для силових компонентів. Для досягнення цієї мети обрано схему, яка зображена на Рисунку 2.6. Згідно цієї схеми сигнал логічного керування реле з рівнем напруги 3,3 В від МК (який через резистор R1 підтягнутий до напруги живлення МК Vcc) підключається через оптрон (до якого також підключено силове живлення JDVcc) для активації реле шляхом подання напруги на транзисторний ключ T1. За такої схеми підключення вхідні контакти модуля працюють в зворотному порядку, тобто реле активується тоді, коли вхідний контакт знаходиться в низькому рівні; при високому рівні вхідного виводу реле активується.

Схема підключення МК до модуля силового реле через оптрон для комутації силової лінії живлення показана на Рисунку 2.7.

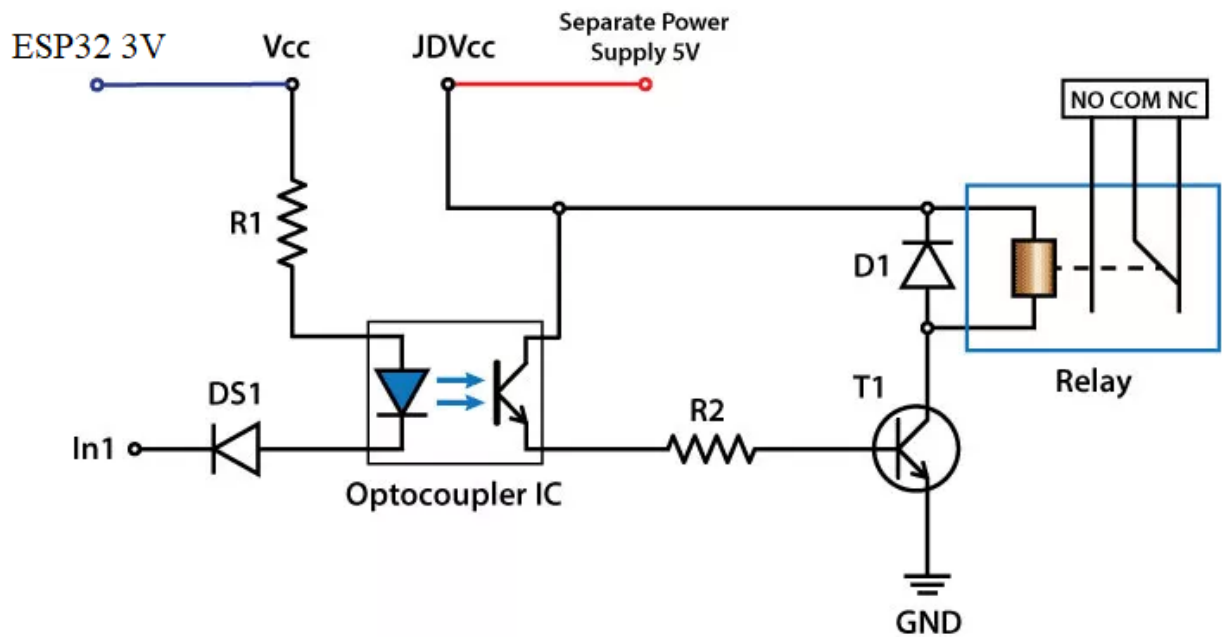


Рисунок 2.6 – Принципова схема підключення МК до релейно-комутаційного модуля

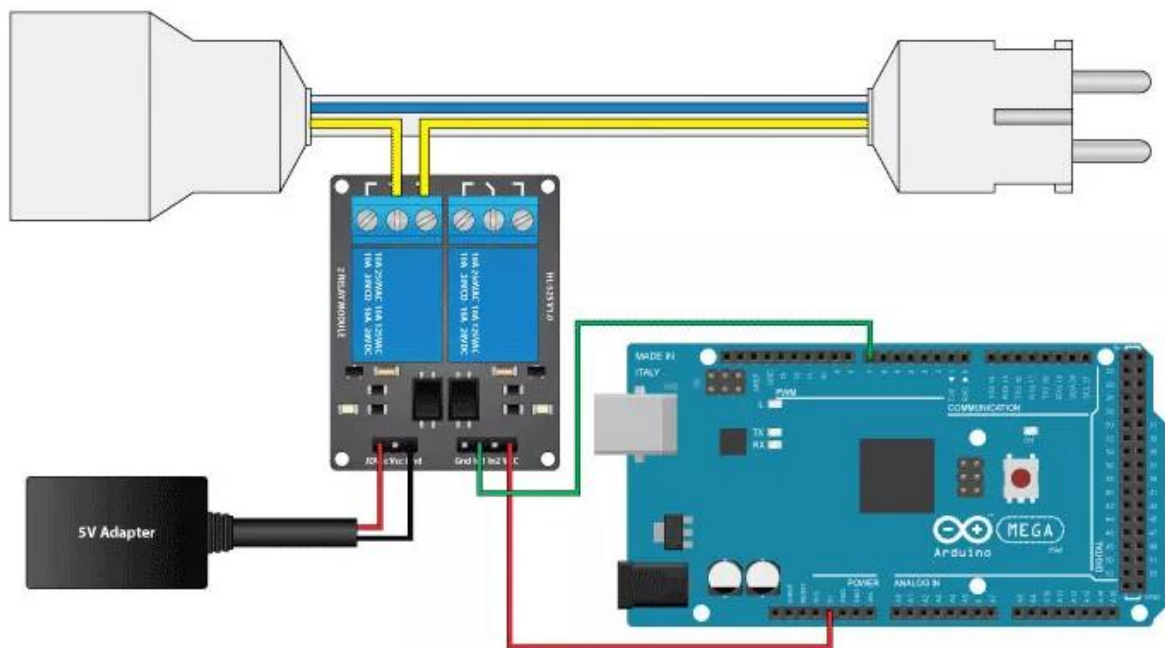


Рисунок 2.7 – Схема комутації силової лінії живлення

Для реалізації схеми живлення цифрових вузлів МК використано лінійний стабілізатор LM1117 [21], який забезпечує стабільну вихідну напругу. Він випускається у різних модифікаціях з фіксованим (наприклад, 3,3 В або 5 В) або

Зм.	Арк.	№ докум.	Підпис	Дата

регульованою напругою на виході, що робить його універсальним для живлення МК, сенсорів та іншої електроніки. Основні характеристики LM1117MPX включають роботу у широкому діапазоні вхідної напруги (до 15 В), максимальний вихідний струм до 800 мА та низьке падіння напруги (близько 1.2 В при повному навантаженні). Стабілізатор має вбудований захист від перегріву та перевантаження, що підвищує його надійність у різних умовах експлуатації. Корпус TO-252 забезпечує ефективне розсіювання тепла, що дозволяє використовувати його в компактних пристроях без додаткового радіатора при помірних навантаженнях. Застосування LM1117MPX охоплює живлення плат розробки (наприклад, ESP32, Arduino), периферійних пристроїв, а також систем, де потрібна стабільна напруга з мінімальними пульсаціями. На рисунку 2.8 показана принципова схема включення стабілізатора LM1117.

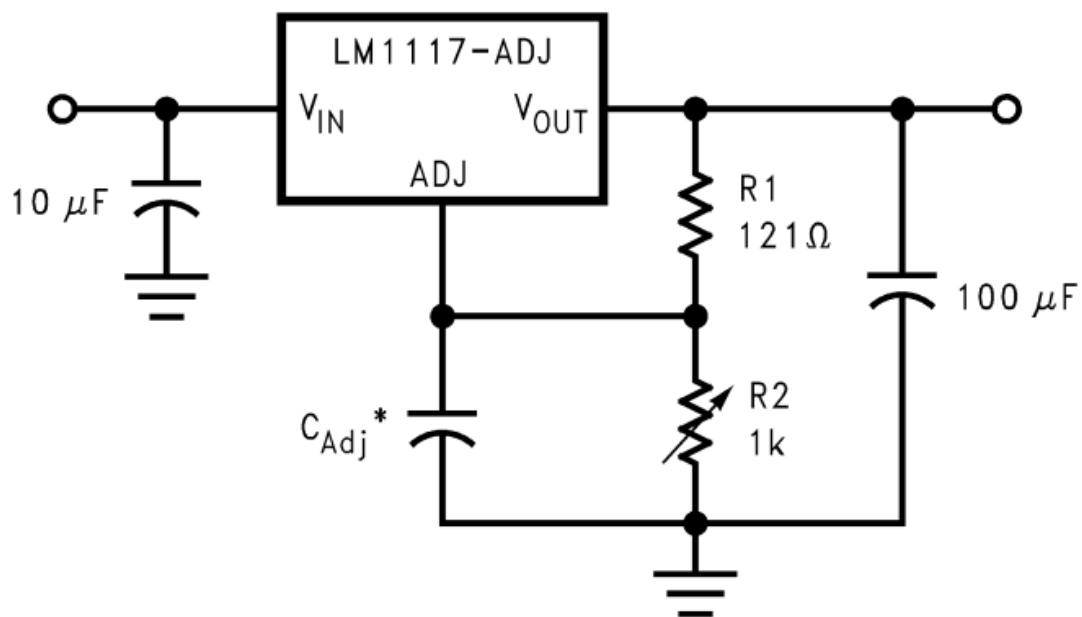


Рисунок 2.8 – Приклад схеми включення стабілізатора LM1117

Для реалізації релейно-комутаційного модуля згідно схеми. Яка зображена на Рисунку 2.6, обрано реле HJR-4102-L-12VDC [22] з комутаційними можливостями до 10А, що підходить для керування високим навантаженням у низьковольтних колах постійного струму. Реле працює від напруги котушки 12В постійного струму та має компактний корпус, що дозволяє легко інтегрувати

його в різноманітні електронні пристрої. Контактна група реле забезпечує надійне перемикання з мінімальними втратами, а вбудована ізоляція між котушкою та комутуючими контактами підвищує безпеку експлуатації. Корпус реле зазвичай має стандартні кріплення для монтажу на DIN-рейку або плату, що спрощує його встановлення в різних типах електронних щитів. Геометричні розміри реле показані на Рисунку 2.9, а зовнішній вигляд – на Рисунку 2.10.

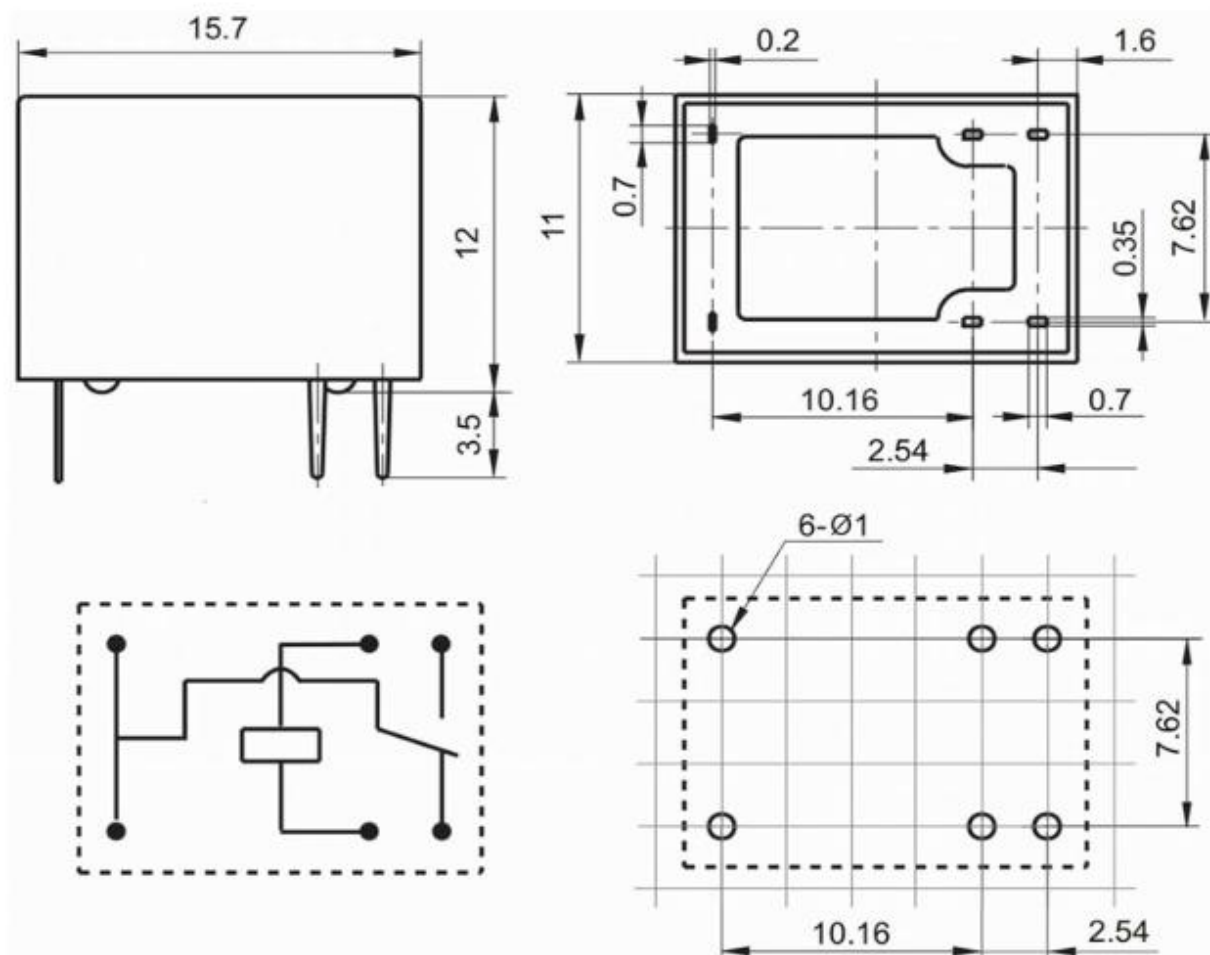


Рисунок 2.9 – Розміри реле HJR-4102-L-12VDC



Рисунок 2.10 – Зовнішній вигляд реле HJR-4102-L-12VDC в корпусі

Зм.	Арк.	№ докум.	Підпис	Дата

Для розв'язки силових ліній та ліній живлення МК використано оптрон PC817. Він складається з інфрачервоного світлодіода та інтегрованого фототранзистора, що забезпечує передачу сигналу без електричного контакту між вхідним і вихідним колами. Основні характеристики включають коефіцієнт передачі струму (CTR) 50-600%, напругу ізоляції до 5 кВ та максимальний вхідний струм 50 мА. Пристрій використовується для захисту мікроконтролерів, ізоляції сигналів у блоках живлення та промислових системах автоматизації [23]. Фото оптрона PC817 та опис його виводів показано на Рисунку 2.11.

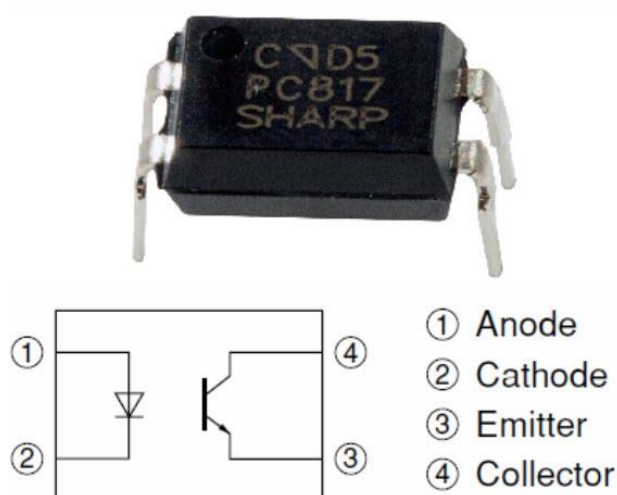


Рисунок 2.11 – Оптрон PC817 корпусі DIP та опис його виводів

В якості силового ключа для реле згідно схеми на Рисунку 2.6 використано транзистор TIP35C NPN-типу [24], який призначений для роботи в підсилювачах потужності та комутаційних схемах. Він має максимальну колекторну напругу 100В, струм до 25А у імпульсному режимі та потужність розсіювання до 125Вт. Корпус TO-247 (Рисунок 2.12) забезпечує ефективне охолодження, що дозволяє використовувати його у високонавантажених пристроях, таких як аудіопідсилювачі, джерела живлення та системи керування двигунами. Коефіцієнт посилення струму (h_{FE}) становить 15-75 залежно від робочого

Зм.	Арк.	№ докум.	Підпис	Дата

режиму. Транзистор має вбудований захисний діод між колектором та емітером, що підвищує його надійність при роботі з індуктивним навантаженням.

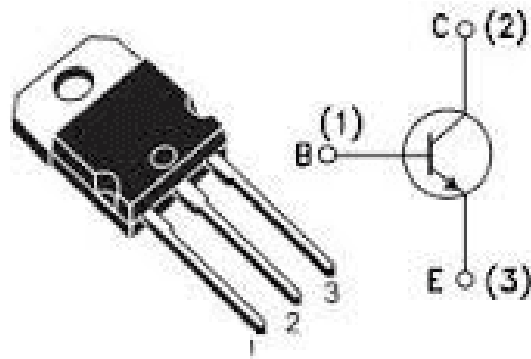


Рисунок 2.12 – Транзистор TIP35C NPN

Для посилення захисту використано додатково діод Шотткі SS14 [25], що характеризується низьким падінням напруги ($\sim 0.5\text{В}$ при 1А) та високою швидкістю. Основні параметри: максимальна зворотна напруга – 40В , прямий струм – 1А , імпульсний струм – 20А . Корпус SMA (DO-214AC) забезпечує компактність та легкий монтаж на поверхню плати (див. Рисунок 2.13).

Катод (-)



Анод (+)

$I_{F(AV)}$	1,0	A
V_{RRM}	40	V
T_j	125	$^{\circ}\text{C}$
$V_F(\text{max})$	0,5	V

DO-214AC



Анод (+)

Катод (-)

Рисунок 2.13 Характеристики виводів та зовнішній вигляд діода Шотткі SS14 в корпусі SMA (DO-214AC)

Зм.	Арк.	№ докум.	Підпис	Дата

КПТР. 021013.01.02 ПЗ

Арк.

29

Таким чином, усі головні компоненти та їх ТТХ були описані в даному підрозділі кваліфікаційного проєкту. Для більшої наочності в Таблиці 2.1 подано відповідність між елементами структурної схеми та обраними компонентами для практичної реалізації пристрою.

Таблиця 2.1 – Співставлення елементів структурної схеми та компонентів реалізації блоку керування та підсистеми контролю вологості і температури «Розумного будинку»

Елемент структурної схеми		Обраний компонент для реалізації
Блок керування	Мікроконтролера	ESP32 SoC
	Мережевий інтерфейс	
	Радіомодуль Wi-Fi	
	Інтерфейс програмув.	JTAG Connector 6 pin
Підсистема контролю температури та вологості	Сенсор температури	DHT22
	Сенсор вологості	
	Релейно-комутаційний модуль	Реле HJR-4102-L-12VDC
		Оптрон PC817
		Транзистор напівпров. TIP35C NPN
Діод Шотткі SS14		

2.3 Формалізація процесу функціонування системи та розробка алгоритму програмного забезпечення

Представимо формальний опис функціонування системи «Розумний будинок» у вигляді узагальненої задачі керування:

$$Y_i = F_i\{P_i, X_i\}, \quad (2.1)$$

де Y_i – реакції системи на стимули X_i за відомими функціоналами F_i та параметрами P_i .

За означенням задачі керування, відомими вважаються також реакції Y_i – цільові значення, яких треба досягти шляхом зміни стимулів X_i – невідомих значень.

Так, вважатимемо $Y_{t,h} \in Y_i$ – цільові значення кліматичних параметрів температури (що вимірюється в $^{\circ}\text{C}$) та відсоток вологості повітря (%).

Згідно вітчизняних норм згідно ДБН В.2.5-67:2022 – Опалення, вентиляція та кондиціювання і СанПіН. Для житлових і офісних приміщень у літній період рекомендовані значення [27]:

- температура: $+23\dots+25^{\circ}\text{C}$ (допустимий діапазон $+22\dots+26^{\circ}\text{C}$);
- відносна вологість: 40-60% (максимум до 65%).

Для дитячих закладів і медичних установ:

- температура: $+21\dots+23^{\circ}\text{C}$;
- вологість: 50-55%.

Ці норми вважатимемо параметрами системи керування: $P_{T,h} \in P_i$, які задаються при налаштуванні системи.

Таким чином, необхідно розробити функціонали (алгоритми) системи керування, відповідно алгоритм керування температурою F_t та відносною вологістю F_h в приміщенні «Розумного будинку».

Блок-схеми алгоритмів керування температурою та рівнем вологості показані на Рисунках 2.14 та 2.15 відповідно. В основі цих алгоритмів лежить спрощений функціонал контролю заданого значення відповідної величини з урахуванням допустимої похибки та періодичне опитування сенсорів. Алгоритм передбачає встановлення цільового значення шляхом віддаленого керування згідно мети розробки проєкту. Згідно технічного завдання на проєктування керування температурою здійснюється за допомогою двох вентиляторів з метою підвищення ефективності їх одночасної роботи в літній період, в той час як для контролю рівня вологості передбачено використання одного зволожувача повітря.

									Арк.
									31
Зм.	Арк.	№ докум.	Підпис	Дата	КПТР. 021013.01.02 ПЗ				

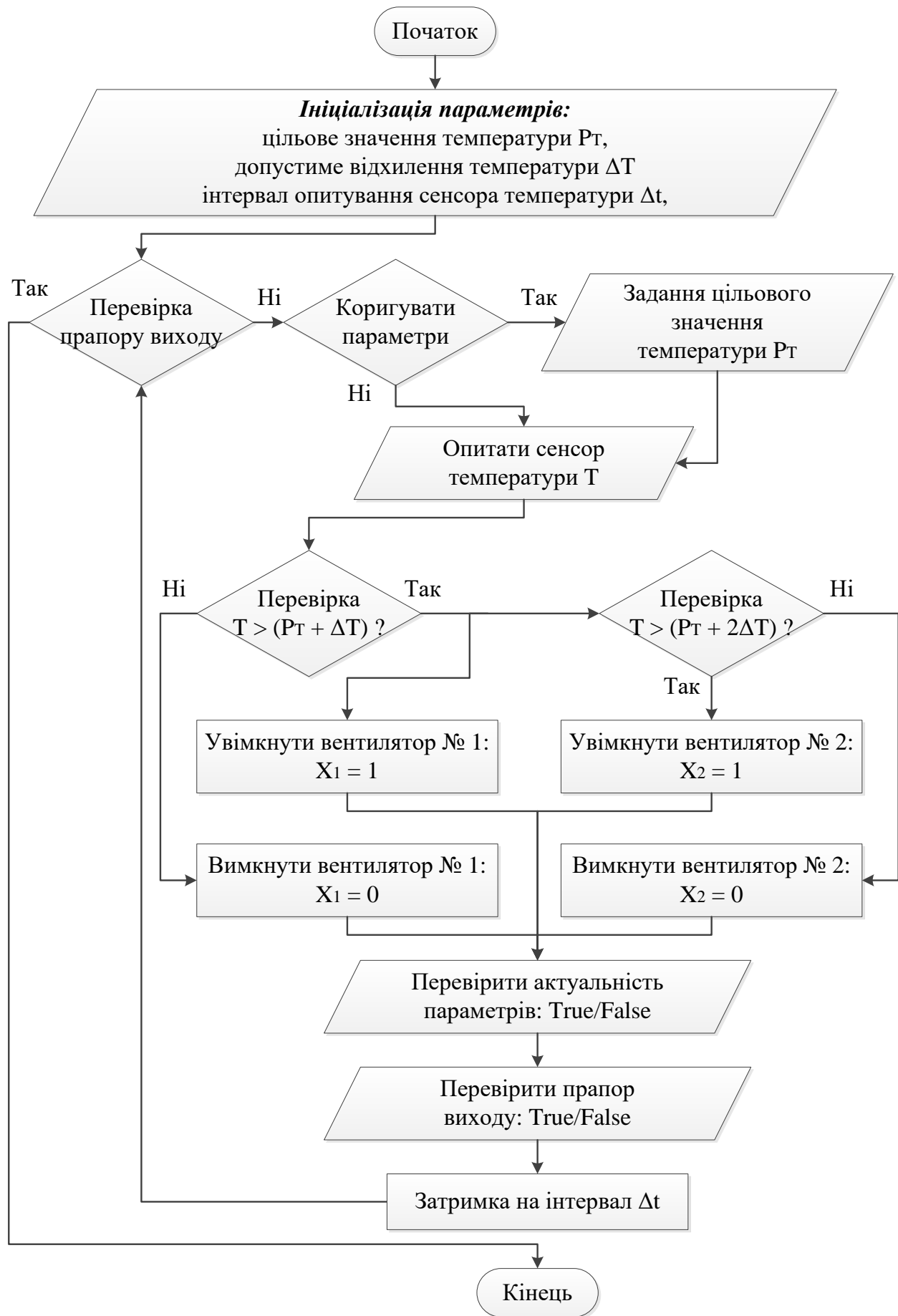


Рисунок 2.14 – Алгоритм керування температурою в «Розумному будинку»

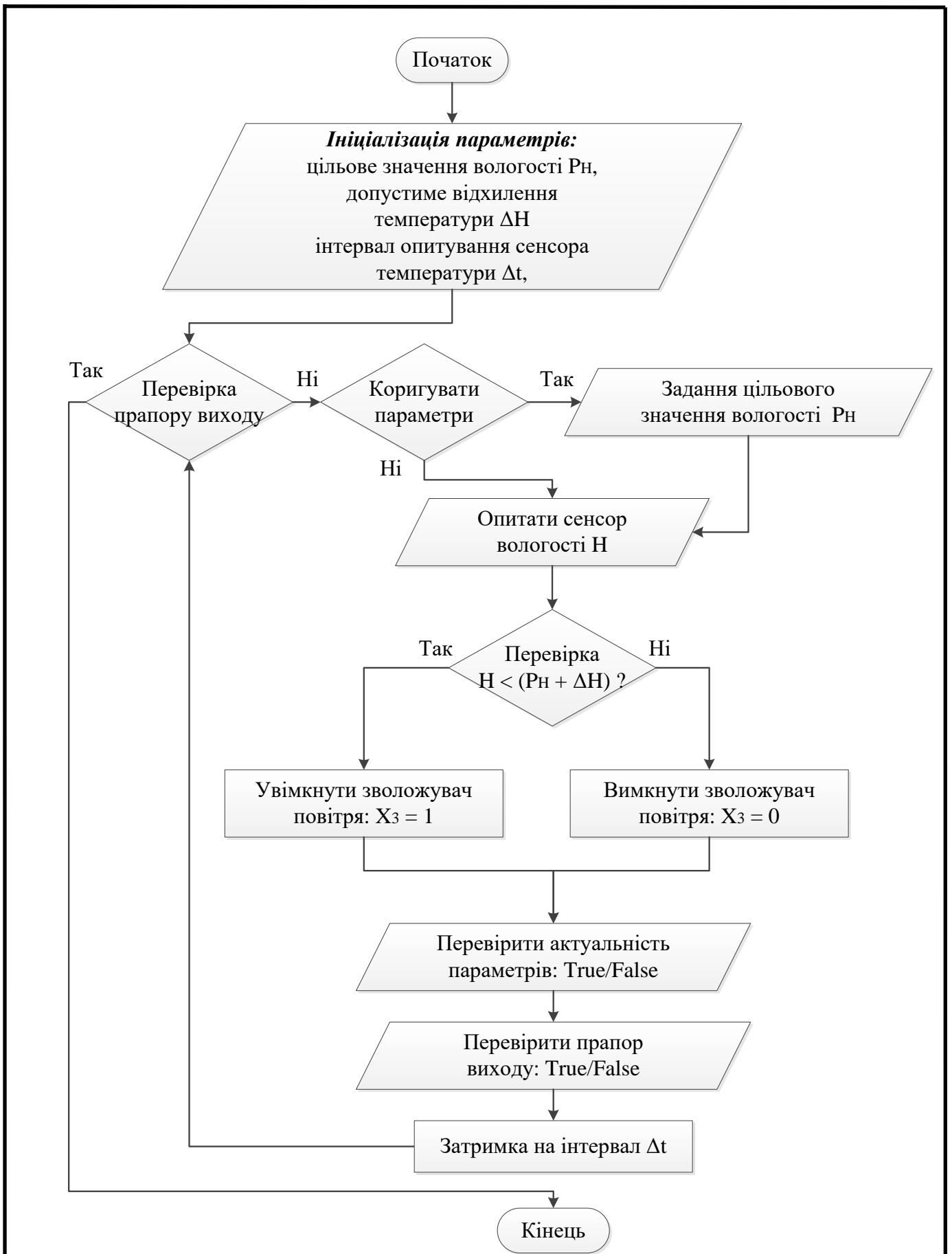


Рисунок 2.15 – Алгоритм керування рівнем вологості в «Розумному будинку»

2.4 Висновки до 2-го розділу

В 2-му розділі кваліфікаційного проекту було розроблено структурну схему БК та підсистеми контролю вологості і температури «Розумного будинку». Креслення схеми електричної структурної додається до проекту. В Додатку А подану копію креслення схеми електричної структурної блоку керування та підсистеми контролю температури і вологості «Розумного будинку».

Також було виконано вибір елементної бази для реалізації складових структурної схеми. В таблиці 2.1 наведено співставлення елементів структурної схеми та обраних компонентів для реалізації.

На основі загальної задачі керування було розроблено спрощені алгоритми керування підсистемою контролю вологості та температури «Розумного будинку».

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

3 АПАРАТНО-ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

«РОЗУМНИЙ БУДИНОК»

3.1 Розробка принципової схеми пристроїв системи «Розумний будинок»

Схема електрична принципова виконана на форматі А2 і додається до матеріалів кваліфікаційного проєкту як обов'язковий елемент конструкторської документації. Копія схеми у повному обсязі з усіма позначеннями та поясненнями подана в Додатку Б. В основі схеми лежить мікроконтролер ESP32 (позначення DD1 на схемі), який виконує функції центрального обчислювального вузла системи.

Для забезпечення гнучкості розробки та налагодження програмного забезпечення до плати виведені інтерфейси Universal Asynchronous Receiver/Transmitter (UART) та JTAG. Інтерфейс (UART) реалізований на виводах 34 (RXD0) та 35 (TXD0) МК, які через захисні елементи заведені на роз'єм XS1 (контакти 5 та 6). Це дозволяє підключати плату до ПК для програмування та моніторингу через стандартний послідовний інтерфейс. Інтерфейс JTAG для низькорівневого відлагодження виведений на виводи 13 (IO14), 14 (IO12), 17 (IO13) та 23 (IO15) МК через струмообмежуючі резистори R2-R5 номіналом 100 Ом і також підключений до роз'єму XS1 (контакти 1-4).

Сенсор температури та вологості DHT22 (позначення DD2 на схемі) підключений за схемою з трьома виводами. Живлення сенсора (вивід 1 VCC) підключено до стабілізованого джерела живлення VCC +3,3В. Інформаційний вивід 2 (SDA) через підтягуючий резистор R7 номіналом 2.7 кОм підключений до виводу 33 (IO21) мікроконтролера DD1, що забезпечує надійний цифровий інтерфейс для передачі даних. Для захисту від перешкод лінія даних додатково фільтрується керамічним конденсатором С3 ємністю 100 нФ.

									Арк.
									35
Зм.	Арк.	№ докум.	Підпис	Дата	КПТР. 021013.01.02 ПЗ				

Релейно-комутаційний модуль реалізований на базі трьох незалежних каналів керування, кожен з яких включає оптронну розв'язку (елементи U1-U3), біполярні транзистори (VT1-VT3), електромагнітне реле (K1-K3) та елементи захисту. Бази транзисторів VT1-VT3 через обмежувальні резистори R12-R14 номіналом 10 кОм підключені до виходів оптронів, що забезпечує гальванічну розв'язку між МК і силовою частиною. Керуючі сигнали з виводів МК 31 (IO19), 30 (IO18) та 23 (IO5) через обмежувальні резистори R8-R10 номіналом 2.7 кОм подаються на входи оптронів. Для захисту від викидів самоіндукції на обмотках реле встановлені діоди Шоттки VD1-VD3. Індикація стану кожного реле здійснюється за допомогою світлодіодів HL2-HL4.

Система живлення побудована на базі лінійного стабілізатора LM1117 (позначення DA1 на схемі) у стандартному включенні з вхідною напругою +5В та вихідною +3,3В. Для фільтрації пульсацій на вході та виході стабілізатора встановлені відповідні фільтри, що виконані на базі конденсаторів C4-C5 та C6-C15 відповідно. Індикація наявності живлення реалізована на світлодіоді HL1 з обмежувальним резистором R6. Підключення живлення +5В здійснюється через роз'єми X1-X2.

3.2 Реалізація віддаленого керування на базі протоколу MQTT

Архітектура програмного компоненту проекту «Розумний будинок» для керування вентиляторами, зволожувачем повітря та моніторингу даних сенсорів побудована на основі протоколу Message Queuing Telemetry Transport (MQTT), який забезпечує ефективний обмін повідомленнями між пристроями в режимі реального часу. Система складається з кількох головних компонентів: МК ESP32, MQTT-брокера, серверного ПЗ та БД під керуванням MS SQL Server.

Відомо [28], що протокол MQTT – це легкий, ефективний та масштабований протокол обміну повідомленнями, розроблений для пристроїв з обмеженими ресурсами, таких як IoT-пристрої, сенсори та МК. Протокол працює

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

за принципом «видавець-підписник» (publish-subscribe), де учасники системи не зв'язані напряму, а спілкуються через центральний вузол – MQTT-брокер.

Згідно архітектури системи БК на базі МК ESP32 виконує роль клієнта MQTT, який підключено до сенсору температури та вологості, а також до актуаторних вузлів, таких як вентилятори та зволожувач повітря. МК ESP32 зчитує дані сенсорів у реальному часі та публікує їх у відповідні MQTT-теми брокера, наприклад, `sensors/temperature/room1` та `sensors/humidity/room1`. Одночасно ESP32 підписані на теми керування, такі як `control/fan/room1` та `control/humidifier/room1`, через які отримуються команди вмикання/вимикання від серверного ПЗ. При отриманні повідомлення МК аналізує його вміст і відправляє відповідні сигнали до виконавчих пристроїв.

MQTT-брокер є центральним вузлом системи, який відповідає за маршрутизацію повідомлень між підписниками та видавцями. Він приймає дані від ESP32 і перенаправляє їх серверному ПЗ, а також передає керуючі команди від сервера до МК. Брокер може бути розгорнутий локально (наприклад, Mosquitto) або використовувати хмарні рішення (AWS IoT, HiveMQ).

Серверне ПЗ також є MQTT-клієнтом і підписане на теми сенсорів для отримання оновлень температури та вологості. При отриманні нових даних ПЗ здійснює їх обробку (наприклад, перевірку на межі допустимих значень) та зберігає у БД MS SQL Server для подальшого аналізу та візуалізації. Якщо показники виходять за задані межі, серверне ПЗ публікує команди керування у відповідні топіки, щоб активувати або деактивувати вентилятори та зволожувач. Для реалізації серверного компонента може використано платформу .NET та мову програмування C#.

В таблиці 3.1 подані основні характеристики модуля віддаленого керування на база протоколу MQTT, зокрема рівні якості обслуговування - Quality of Service (QoS), опис вузлів системи в термінах MQTT та додаткові можливості.

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Таблиця 3.1 – Основні характеристики модуля віддаленого керування

Рівні якості обслуговування QoS	
QoS 0 – лише 1 раз	Повідомлення відправляється один раз без підтвердження. Найшвидший, але ненадійний варіант.
QoS 1 – принаймні 1 раз	Брокер гарантує отримання, але можливі дублікати.
QoS 2 – гарантовано 1 раз	Найнадійніший рівень, де повідомлення доставляється строго один раз за рахунок додаткового етапу підтвердження.
Функціонування в системі «Розумний будинок»	
ESP32 (видавець)	Відправляє показники температури/вологості у відповідні теми.
ESP32 (підписник)	Отримує команди з тем керування та виконує дії.
Сервер (підписник)	Отримує дані, аналізує їх та публікує команди керування (наприклад, увімкнути вентилятор).
Додаткові можливості	
Зберегти останнє / Retained Messages	Брокер зберігає останнє повідомлення в темі та автоматично відправляє його новим підписникам.
Остання команда / Last Will and Testament (LWT)	Якщо клієнт несподівано від'єднується, брокер публікує заздалегідь визначене повідомлення.
Очистка сесії / Clean Session	Визначає, чи зберігатиме брокер підписки клієнта після відключення

Таким чином, система працює у замкнутому циклі: сенсори передають дані, сервер аналізує їх і відправляє команди, виконавчі пристрої регулюють клімат у кімнаті. Протокол MQTT забезпечує швидкий та надійний обмін даними, а MS SQL Server дозволяє зберігати та обробляти інформацію для подальшого використання.

3.3 Реалізація програмних компонентів системи «Розумний будинок»

В даному розділі кваліфікаційного проєкту було виконано розробку ПЗ для системи «Розумний будинок», а також таких її підсистем:

- підсистема контролю температури в кімнаті,
- підсистема контролю вологості.

Відповідно до декомпозиції системи, серверна частина та інтерфейс взаємодії програми з користувачем представлені такими основними компонентами:

1. Інтерфейс користувача (верхній рівень ПЗ) – представлено у вигляді веб-орієнтованого додатку, який побудований згідно шаблону Модель-Представлення-Контролер (Model-View-Controller, MVC).

2. Сервер-координатор кінцевих пристроїв (проміжне ПЗ).

Елементи інтерфейсу користувача представлені набором HTML-сторінок – View-компонентів, в той час як логіка програми та взаємодія із БД реалізовані у компонентах Controllers (контролери) та Models (моделі даних). Крім того, серверний модуль включає інтерфейс програмної взаємодії (Application Programming Interface, API) для комунікації із кінцевими пристроями системи, керування та акумулювання сенсорних даних.

Нижче проілюстровано процес розробки ПЗ в середовищі Microsoft Visual Studio, зокрема показано процедуру створення та налаштування проєкту, підключення до бази даних та початкову міграцію БД. Авторизація користувача системи відбувається на рівні БД за логіном та паролем.

Створення проєкту (Рисунок 3.1 – 3.3) ASP.NET MVC у Microsoft Visual Studio починається з запуску середовища розробки та вибору відповідного шаблону. Після відкриття Visual Studio обирається шаблон ASP.NET Web Application (.NET Framework). або ASP.NET Core Web App, залежно від версії, яка використовується. Для розробки ПЗ проєкту обрано шаблон MVC, цільова платформа – .NET Core.

Після створення проєкту автоматично генерується базова структура папок, яка відповідає шаблону MVC. У корені проєкту з'являються каталоги, такі як Controllers (для класів-контролерів), Models (для моделей даних), Views (для HTML-представлень) та Scripts (для JavaScript-файлів). Також створюються файли конфігурації, зокрема Web.config (для .NET Framework) або appsettings.json (для .NET Core), які містять налаштування додатку. У папці App_Start знаходиться конфігурацію маршрутизації (RouteConfig.cs), яка визначає, як URL-адреси співвідносяться з контролерами та їх методами.

У шаблоні ASP.NET MVC з підтримкою автентифікації, який генерується Visual Studio за замовчуванням, БД створюється на основі технології Entity Framework та Identity Framework. Ця структура включає основні таблиці для керування користувачами, ролями та пов'язаними даними авторизації. Головною таблицею є AspNetUsers, де зберігаються основні відомості про користувачів, такі як ідентифікатор, ім'я, електронна пошта, хеш пароля, сіль для шифрування, номер телефону та статус підтвердження облікового запису.

Для роботи з ролями використовується таблиця AspNetRoles, яка містить ідентифікатори та назви ролей, таких як Admin або User. Зв'язок між користувачами та ролями реалізований через проміжну таблицю AspNetUserRoles, де зберігаються пари ідентифікаторів користувача та ролі. Це дозволяє призначати одному користувачеві кілька ролей.

Додаткові таблиці, такі як AspNetUserClaims та AspNetUserLogins, відповідають за зберігання додаткових претензій (claims) користувачів, які можуть використовуватися для розширеної авторизації, а також інформації про

									Арк.
									40
Зм.	Арк.	№ докум.	Підпис	Дата	КПТР. 021013.01.02 ПЗ				

зовнішні провайдери входу, якщо додаток підтримує автентифікацію через соціальні мережі або інші сервіси.

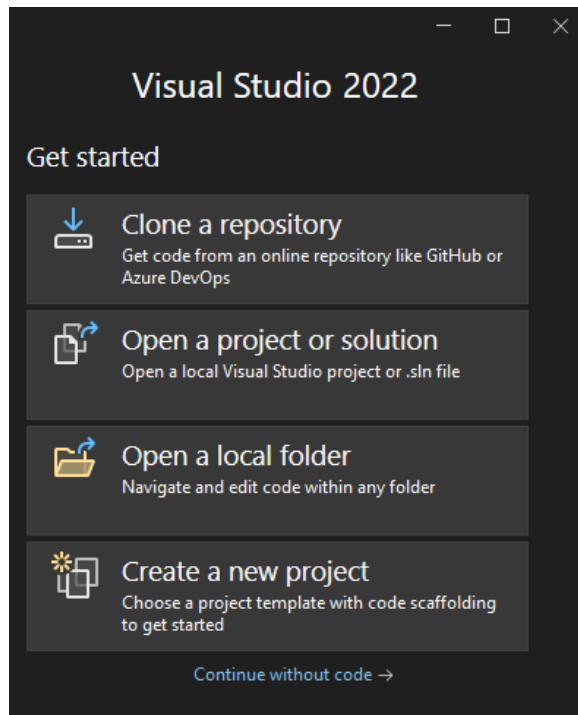


Рисунок 3.1 – Створення нового проекту у IDE MS Visual Studio

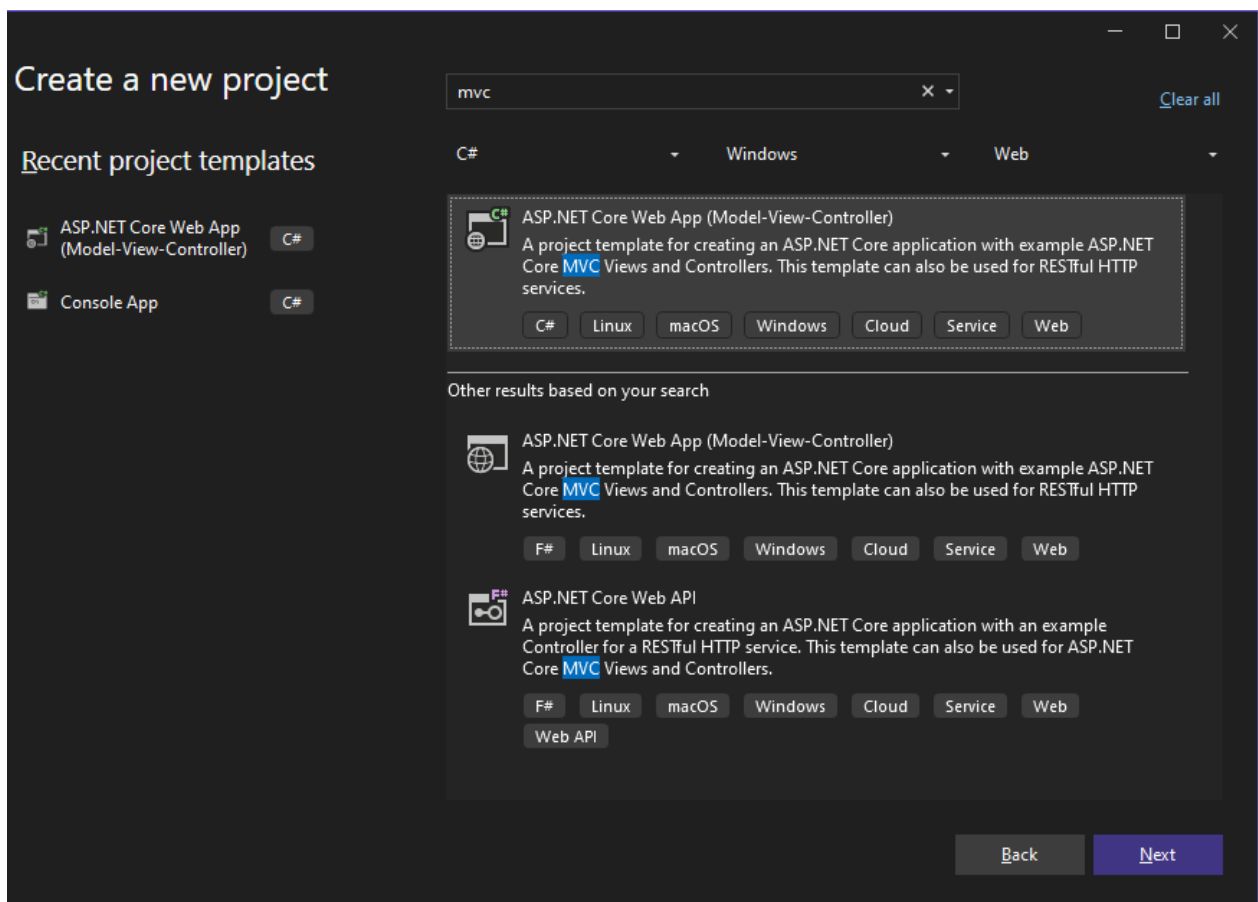


Рисунок 3.2 – Вибір шаблону проекту MVC

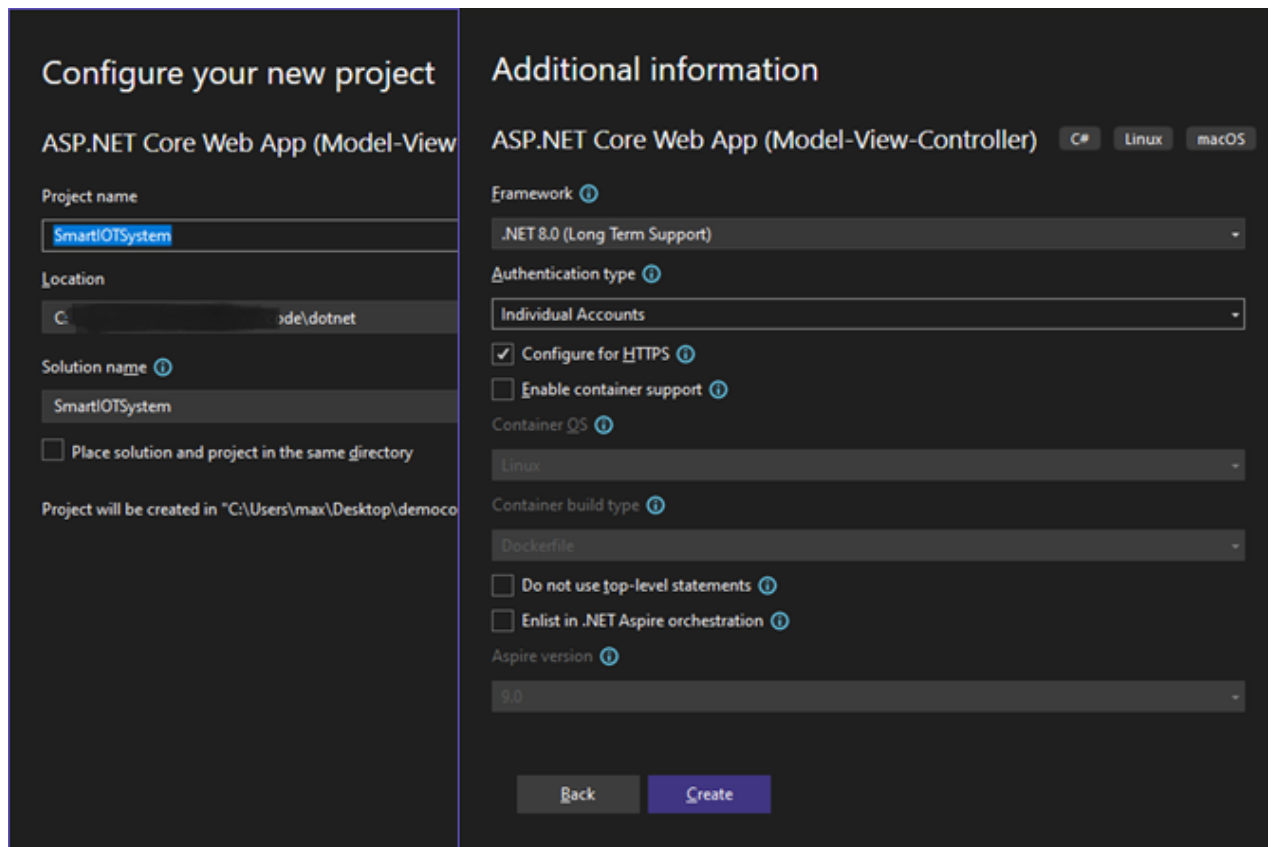


Рисунок 3.3 – Налаштування нового проекту

Для керування токенами підтвердження електронної пошти, скидання паролів та інших тимчасових даних використовується таблиця `AspNetUserTokens`. Вона дозволяє зберігати токени у зв'язку з конкретними користувачами, що необхідно для функціоналу відновлення пароля або двофакторної автентифікації. Структура БД також може включати додаткові таблиці, пов'язані зі специфікою додатку, які розробник може додати вручну або через міграції Entity Framework. Загалом, ця схема забезпечує гнучке керування користувачами та їх правами, що є основою для більшості сучасних веб-додатків із системою авторизації.

Для взаємодії з БД на рівні програмного коду, а також для міграції БД виконано такі додаткові пакети (Рисунок 3.4):

- Microsoft.EntityFrameworkCore,
- Microsoft.EntityFrameworkCore.SqlServer,
- Microsoft.EntityFrameworkCore.Tools.

1) Entity Framework Core (EF Core) – це сучасний засіб відображення об’єктної бази даних, який дає змогу створювати чистий, портативний і високорівневий рівень доступу до даних за допомогою .NET (C#) для різноманітних баз даних, зокрема SQL Server (локальна та Azure), SQLite, MySQL, PostgreSQL, Oracle та Azure Cosmos DB. Він підтримує запити LINQ, відстеження змін, оновлення та міграції схем.

2) Microsoft.EntityFrameworkCore.SqlServer – це пакет постачальника бази даних EF Core для Microsoft SQL Server і Azure SQL.

3) Інструменти Entity Framework Core допомагають із завданнями розробки під час проектування. Вони в основному використовуються для керування міграціями та створення каркасу DbContext і типів сутностей шляхом зворотного проектування схеми бази даних.

Пакет Microsoft.EntityFrameworkCore.Tools призначений для інструментів PowerShell, які працюють у консолі диспетчера пакетів Visual Studio (PMC).

В якості системи керування БД обрано MS SQL Server Express. На Рисунку 3.5 показано налаштування підключення до СКБД. Дані вимірювання системи містяться в двох головних таблицях: IoTUnit та DataRecord, які пов’язані зв’язком «один до багатьох» – див. Рисунок 3.6.

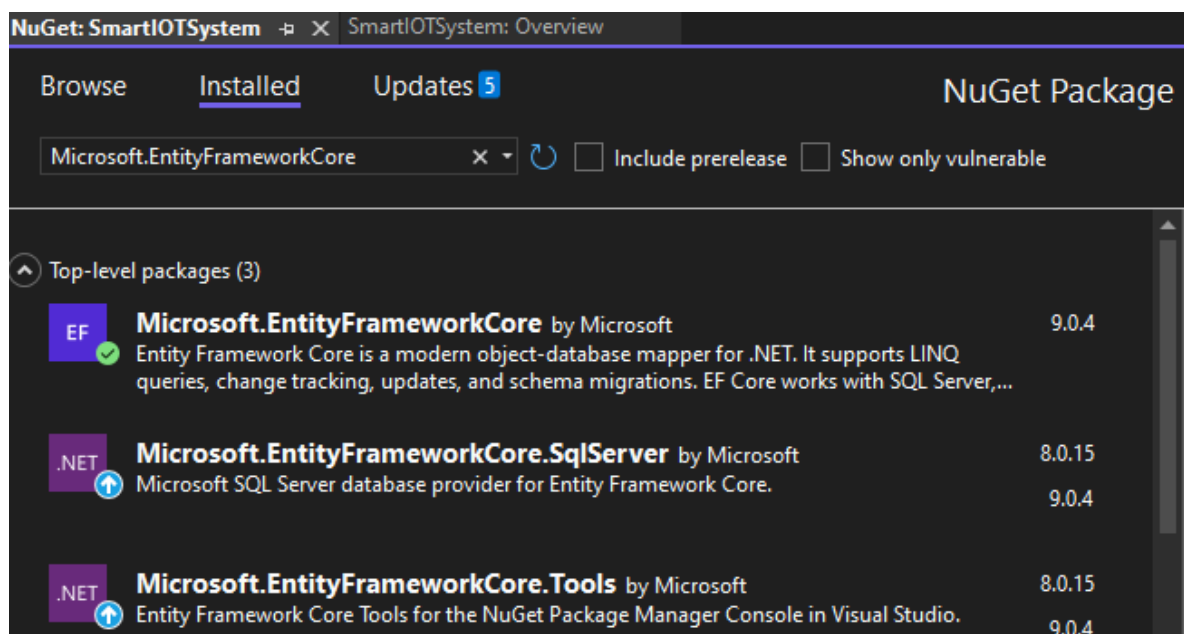


Рисунок 3.4 – Підключення додаткових пакетів для роботи з БД

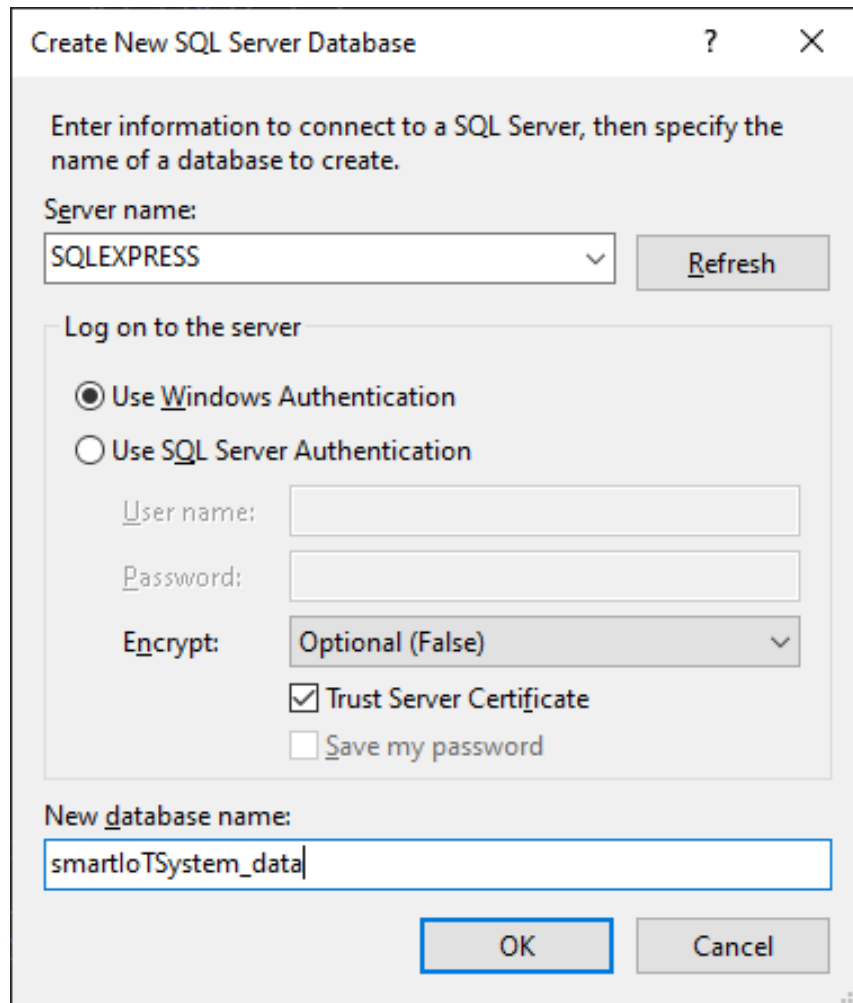


Рисунок 3.5 – Налаштування підключення до MS SQL Server Express

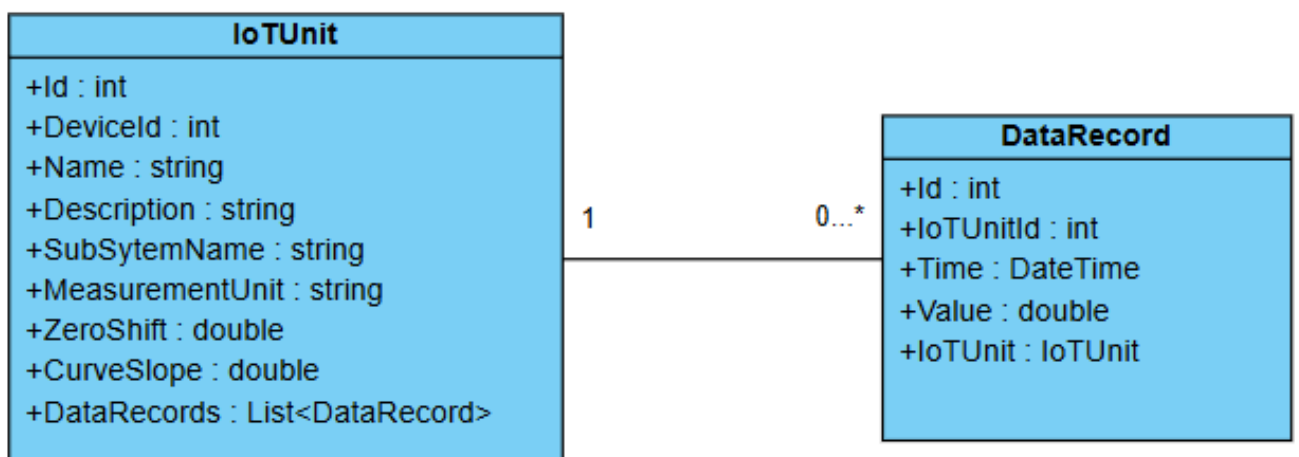


Рисунок 3.6 – Схема бази даних

Нижче наведено код моделі даних.

```
namespace SmartIOTSystem.Models
{
    public class IoTUnit
    {
        public int Id { get; set; } // Primary key

        public int DeviceId { get; set; }

        public string Name { get; set; } = null!;

        public string Description { get; set; } = null!;

        public string SubSystemName { get; set; } = null!;

        public string MeasurementUnit { get; set; } = null!;

        // Calibration coefficient  $x = value * CurveSlope + ZeroShift$ 

        public double ZeroShift { get; set; }

        public double CurveSlope { get; set; } = 1.0;

        // Navigation property - 1 IoTUnit can have many DataRecords
        public ICollection<DataRecord> DataRecords { get; set; } = new List<DataRecord>();
    }
}

namespace SmartIOTSystem.Models
{
    public class DataRecord
    {
        public int Id { get; set; } // Primary key

        public DateTime Time { get; set; }

        public double Value { get; set; }

        public int IoTUnitId { get; set; } // Foreign key

        public IoTUnit IoTUnit { get; set; } = null!; // Navigation property

        public double GetCalibratedValue()
        {
            double slope = 1, shift = 0;
            if (IoTUnit != null)

```

Зм.	Арк.	№ докум.	Підпис	Дата

КПТР. 021013.01.02 ПЗ

Арк.

45

```

        {
            slope = IoTUnit.CurveSlope;
            shift = IoTUnit.ZeroShift;
        }
        return Value * slope + shift;
    }
}
}
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;
using SmartIOTSystem.Models;

namespace SmartIOTSystem.Data
{
    public class ApplicationDbContext : IdentityDbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }

        public DbSet<IoTUnit> IoTUnits { get; set; }

        public DbSet<DataRecord> DataRecords { get; set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            base.OnModelCreating(modelBuilder);

            // Configure the 1-to-many relationship
            modelBuilder.Entity<DataRecord>()
                .HasOne(record => record.IoTUnit)
                .WithMany(unit => unit.DataRecords)
                .HasForeignKey(record => record.IoTUnitId)
                .OnDelete(DeleteBehavior.Cascade); // Delete records when unit is deleted
        }
    }
}

```

Класи моделей даних проєкту ASP.NET MVC у просторі імен SmartIOTSystem.Models представляють структуру для роботи з IoT-пристроями та їхніми даними. Основним класом є IoTUnit, який описує одиницю IoT-обладнання та містить властивості для ідентифікації пристрою, його назви, опису, підсистеми, одиниць вимірювання, а також коефіцієнти калібрування

					КПТР. 021013.01.02 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

(ZeroShift та CurveSlope). Клас включає навігаційну властивість DataRecords, що реалізує зв'язок «один-до-багатьох» з класом DataRecord, який відповідає за зберігання окремих записів даних.

Клас DataRecord призначений для зберігання показників, отриманих від IoT-пристроїв, у певний момент часу. Він містить ідентифікатор запису, час фіксації даних, значення показника та зовнішній ключ IoTUnitId, що зв'яже запис із конкретним пристроєм. Навігаційна властивість IoTUnit дозволяє отримувати доступ до пов'язаного пристрою, а метод GetCalibratedValue() виконує коригування значення з урахуванням калібрувальних коефіцієнтів пристрою, повертаючи точніший результат.

Клас ApplicationDbContext, успадкований від IdentityDbContext, забезпечує інтеграцію системи автентифікації з базою даних. Він містить набори даних DbSet<IoTUnit> та DbSet<DataRecord>, які відображають таблиці бази даних для IoT-пристроїв та записів відповідно. У методі OnModelCreating налаштовується зв'язок між цими сутностями, де визначається поведінка при видаленні (каскадне видалення записів при видаленні пристрою).

Модель даних реалізує принципи Object-Relational Mapping (ORM) через Entity Framework Core. Класи IoTUnit та DataRecord відображаються на таблиці БД, а їхні властивості – на стовпці. Навігаційні властивості забезпечують зручний доступ до пов'язаних даних без необхідності ручного написання SQL-запитів. Калібрувальні коефіцієнти в IoTUnit дозволяють програмно коригувати отримані значення, що є важливим для забезпечення точності вимірювань у IoT-системах.

Отже, наведена вище структура моделей є типовою для ASP.NET MVC-додатків, які працюють із реляційними БД. Вона поєднує бізнес-логіку (калібрування даних) з механізмами зберігання та автентифікації, що робить її придатною для розробки систем моніторингу IoT-пристроїв із підтримкою користувацьких ролей та прав доступу. Використання Identity Framework

дозволяє легко додати функціонал реєстрації, входу та керування користувачами без необхідності створювати відповідні моделі з нуля.

На Рисунку 3.7 показано таблиці БД після початкової міграції даних.

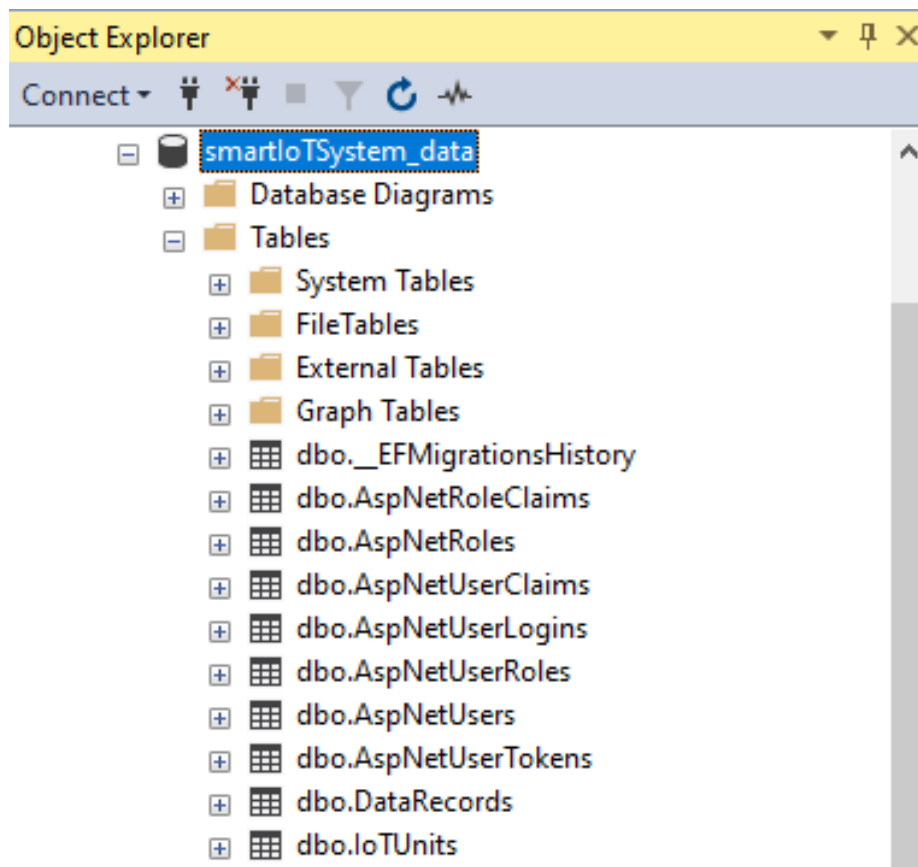


Рисунок 3.7 – Результат початкової міграції у вигляді створеної схеми даних на SQL Server

Інтерфейс користувача системи SmartIoTSystem починається з форм авторизації та реєстрації (Рисунок 3.8 – 3.9), які забезпечують захищений доступ до функціоналу. Форма входу містить поля для електронної пошти та пароля, а також посилання на відновлення пароля для забутих облікових даних. Форма реєстрації включає додаткові поля для підтвердження пароля та, за необхідності, інші відомості про користувача. Після успішної автентифікації система перенаправляє користувача на головну панель керування, де відображаються основні інструменти для роботи з IoT-пристроями.

Register

Create a new account.

Use another service to register.

Email
user@email.com

Password
%Cb%S?C_Vc4HEdR

The Password Field is required.

Confirm Password
%Cb%S?C_Vc4HEdR

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).

Google Password Manager created a strong password for this website

You won't need to remember this password. It will be saved to Google Password Manager for maxslbn9@gmail.com.

Рисунок 3.8 – Реєстрація користувача в системі

Log in

Use a local account to log in.

Email
user@email.com

Password
.....

Remember me?

[Forgot your password?](#)

[Register as a new user](#)

[Resend email confirmation](#)

Hello user@email.com!

Рисунок 3.9 – Авторизація в системі.

Головна панель керування є центральним елементом інтерфейсу, де користувач може швидко отримати доступ до всіх функцій. Тут розміщені елементи керування зі статусом підключених пристроїв, графіки останніх вимірювань та кнопки швидкого керування (включення/виключення пристроїв, налаштування параметрів). Панель також містить навігаційне меню з

посиланнями на сторінки зі списком пристроїв, історією даних та налаштуваннями системи. Інтерфейс адаптований для різних типів користувачів, тому адміністратори бачать додаткові інструменти для керування обліковими записами та глобальними налаштуваннями.

Сторінка зі списком розумних пристроїв організована за підсистемами (Рисунок 3.10), що дозволяє зручно групувати обладнання за функціональним призначенням. Кожен пристрій відображається у вигляді картки з основним описом, поточним статусом та значенням останнього вимірювання. Користувач може фільтрувати пристрої за підсистемою, типом або статусом, а також швидко переходити на сторінку детальних параметрів конкретного пристрою. Для зручності на цій сторінці реалізовано пошук за назвою або ідентифікатором, що особливо корисно в системах з великою кількістю підключених пристроїв.

Index

[Create New](#)

Device Id	Name	Description	Subsystem	
1	Temperature Sensor	Temperature control in the room	Temperature Control	Edit Details Delete
2	Fan 1	Cooling Fan 1	Temperature Control	Edit Details Delete
3	Fan 2	Cooling Fan 2	Temperature Control	Edit Details Delete
4	Humidity Sensor	Humidity control in the room	Humidity Control	Edit Details Delete
5	Air Humidifier	Air Humidifier 1	Humidity Control	Edit Details Delete
6	Motion sensor 1	Motion sensor to turn on lights	Light Control	Edit Details Delete
7	Motion sensor 2	Motion sensor to turn on lights	Light Control	Edit Details Delete
8	Motion sensor 3	Motion sensor to turn on lights	Light Control	Edit Details Delete
9	Turn On Lights	Turn On/Off the Lights	Light Control	Edit Details Delete
101	Test Unit 1	Test Unit	Test	Edit Details Delete
102	Test Unit 2	Test Unit	Test	Edit Details Delete

Рисунок 3.10 – Список Розумних пристроїв в системі, розділених на підсистеми

Сторінка параметрів пристрою (Рисунок 3.11) містить детальну інформацію про обраний IoT-пристрій, включаючи калібрувальні коефіцієнти та тестові інструменти для налагодження. Тут користувач може змінювати назву,

опис, одиниці вимірювання та інші метадані пристрою. Розділ тестування дозволяє відправляти пробні команди до пристрою та перевіряти його відповіді, що є важливим етапом при інтеграції нового обладнання. Для пристроїв із підтримкою калібрування на цій сторінці доступні інструменти для налаштування ZeroShift та CurveSlope з можливістю перегляду результатів калібрування у реальному часі.

Details

IoTUnit

Deviceld	101
Name	Test Unit 1
Description	Test Unit
SubSystemName	Test
MeasurementUnit	none
ZeroShift	0
CurveSlope	1

Рисунок 3.11 – Параметри пристрою (тестовий елемент для налагодження)

Сторінка візуалізації даних (Рисунок 3.12) демонструє історію вимірювань обраного пристрою за останні 24 години у вигляді графіка. Для тестового режиму використовується генератор випадкових даних, який імітує роботу реального датчика. Користувач може змінювати масштаб графіка, вибирати періоди відображення та порівнювати дані з різних пристроїв. Під графіком розміщена таблиця з детальними значеннями, яку можна експортувати у формати CSV або Excel для подальшого аналізу. Ця сторінка особливо корисна для оперативного моніторингу та виявлення аномалій у роботі підключених пристроїв.

									Арк.
									51
Зм.	Арк.	№ докум.	Підпис	Дата					

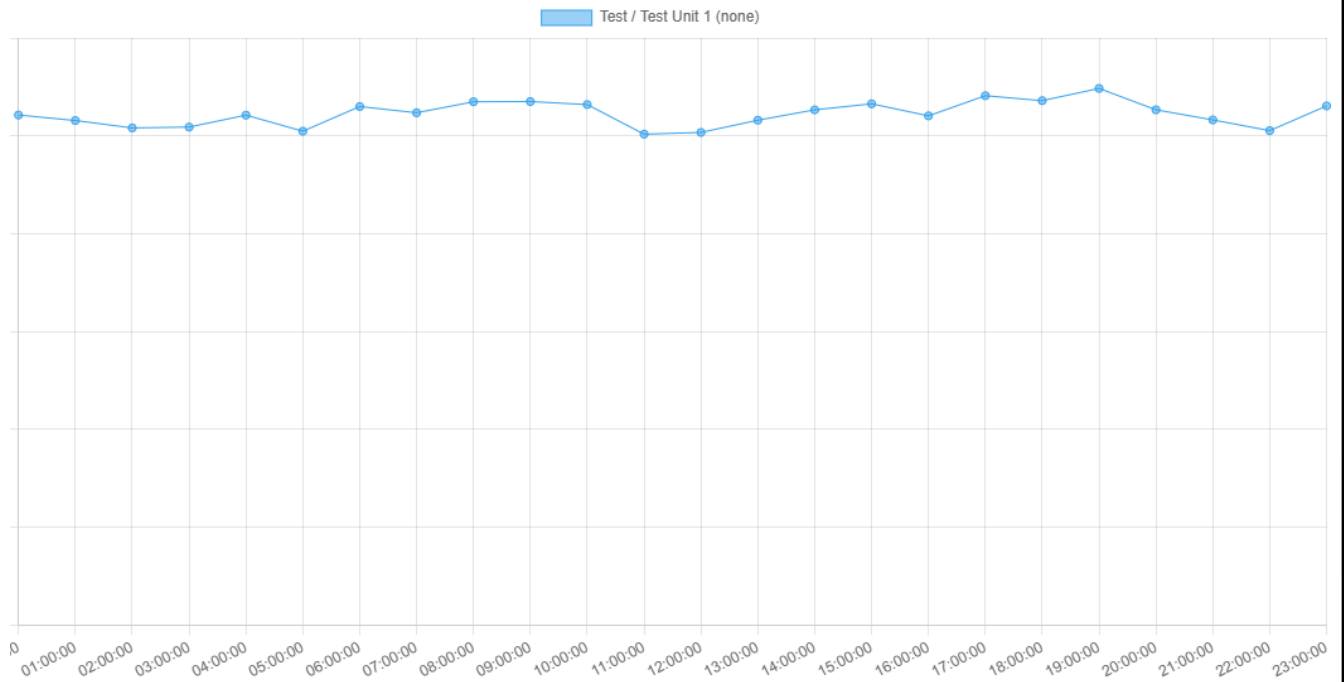


Рисунок 3.12 – Приклад відображення даних вимірювання за 24 години
(тестовий набір даних для налагодження програми)

3.4 Висновки до 3-го розділу

В 3-му розділі кваліфікаційного проєкту виконано розробку схеми електричної принципової БК та системи контролю температури та вологості в кімнаті «Розумного будинку». Схема електрична принципова виконана на форматі А2 і додається до матеріалів проєкту. В Додатку Б подано копію схеми.

Також в розділі описано процес розробки та наведено демонстрацію роботи ПЗ інтерфейсу для віддаленого керування системою «Розумний будинок».

ЗАГАЛЬНІ ВИСНОВКИ ДО ПРОЄКТУ

1. Виконано аналіз сучасних компонентів систем «Розумного будинку» з приведенням їх тактико-технічних характеристик та економічної рентабельності. Проведено аналіз існуючих технологій та рішень щодо проєктування систем «Розумний будинок» для Інтернету речей.

2. Розроблено структуру системи «Розумний будинок», яка складається з блоку керування, мережевого інтерфейсу з безпроводовим каналом зв'язку та сенсорно-актуаторною частиною для контролю та керування температурою та вологістю в кімнаті «Розумного будинку».

3. Розроблено схему електричну принципову блоку керування та підсистеми моніторингу і контролю вологості і температури на базі мікроконтролера ESP32, цифрового сенсора DHT22 та релейно-комутаційної частини з оптронною розв'язкою для керування вентиляторами та зволожувачем повітря.

4. Розроблено програмне забезпечення для блоку керування, модуля віддаленого доступу на базі протоколу MQTT, а також серверну частину веб-орієнтованого інтерфейсу користувача для налаштування параметрів системи та моніторингу даних сенсорів.

Розроблений прототип системи «Розумний будинок» з віддаленим керуванням для Інтернету речей може бути використаний для тестування, демонстрації та подальшої розробки подібних систем, а також в навчальному процесі для здобувачів освіти за спеціальністю 172 – Телекомунікації та радіотехніка.

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. M. Weyrich, C. Ebert, «Reference architectures for the internet of things», IEEE Software, 2016, Vol. 33, № 1, P. 112–116.
2. Ю. В. Пиріг, М. В. Кайдан, О. В. Гордійчук-Бублівська. Аналіз концепції інтернету речей та динаміки її розвитку у різних галузях, Вісник Університету «Україна», 2020, № 1 (24), С. 172–185.
3. «Розумний будинок» : бібліографічний покажчик / КЗ «ЗОУНБ» ЗОР, Від. наук. інформації та бібліографії ; [уклад. М. Маслова]. – Запоріжжя : [ЗОУНБ], 2021. – 76 с.
4. O. Roberman. Smart Home: A Complete Guide for a Connected House [Online resource]. Access: <https://easternpeak.com/blog/smart-home-guide-for-a-connected-house/>.
5. І. О. Дужак. «Розумний будинок», Автоматизація технологічних і бізнес-процесів, 2013, № 13(14), С.31–33.
6. В. В. Лавриненко. Система «Розумний будинок», XI НТК Радіоелектроніка у XXI столітті, 2017, С. 71–73.
7. М. А. Чижевська. Параметри інформаційної системи «Розумний будинок», Наукові записки УНДІЗ, 2019, № 4(56), С. 61–67.
8. А. О. Федько, І. Я. Зеленьова, М. Ю. Тягунова. Розробка та моделювання системи «Розумний будинок» з підвищеною надійністю на базі Arduino, 2019, Наукові праці Донецького національного технічного університету, серія «Проблеми моделювання та автоматизації проектування» №1 (14), С. 16–25.
9. С. О. Макєєв, С. В. Вировець. Сучасна апаратура та обладнання систем «Розумний будинок», Вісник Національного технічного університету «ХПІ». Серія: Проблеми удосконалювання електричних машин і апаратів. Теорія і практика, 2020, № 2, С. 16–20.
10. Характеристики Датчик руху Євросвітло SB-01 360° Білий [Електронний ресурс]. Режим доступу: https://rozetka.com.ua/ua/evrosvet_sb-01_360_belyj_40945/p166306176/characteristics/.

									Арк.
									54
Зм.	Арк.	№ докум.	Підпис	Дата	КПТР. 021013.01.02 ПЗ				

11. Модуль на 8 релейних виходів МРЛІ-8 із перекидними контактами реле та інтерфейсом RS-485 [Електронний ресурс]. Режим доступу: <https://www.elvis.com.ua/product/mrl-8-modul-relejnyh-linij.html>.

12. Сенсор контролю протікання води Mastino WS2 White, 2м [Електронний ресурс]. Режим доступу: <https://smart-systems.com.ua/water-leakage-control-sensor-mastino-ws2-white>.

13. Сенсор диму СПД-3.2 [Електронний ресурс]. Режим доступу: <https://security-shop.com.ua/ua/datchik-dimu-spd-3-2>.

14. CoolMasterNet HVAC Interface [Online resource]. Access: <https://www.smarthomelabs.com.au/products/coolmasternet-hvac-interface>.

15. ESP32 Technical Reference Manual [Online resource] / Espressif Systems. – Version of Document 4.7. – 2022. – Access: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.

16. The Internet Of Things With ESP32 – Features & Specifications [Online resource]. Access: <http://esp32.net/>.

17. Technical Guide to JTAG [Online resource]. Access: <https://www.xjtag.com/about-jtag/jtag-a-technical-overview/>.

18. J-Link PRO – The ultra-fast debug probe with Ethernet [Online resource]. Access: <https://www.segger.com/products/debug-probes/j-link/models/j-link-pro/>.

19. Introduction to the ESP-Prog Board [Online resource]. Access: https://docs.espressif.com/projects/esp-iot-solution/en/latest/hw-reference/ESP-Prog_guide.html.

20. DHT22 – Temperature and Humidity Sensor [Online resource]. Access: <https://components101.com/sensors/dht22-pinout-specs-datasheet>.

21. LM1117MPX 800mA Low-Dropout Linear Regulator [Online resource]. Access: <https://www.ti.com/lit/ds/symlink/lm1117.pdf>.

22. HJR-4102-L-12VDC Signal Relay Datasheet [Online resource]. Access: <https://cdn.hongjingrelay.com/upload/file/HJR-4102-L-12VDC.pdf>.

					КПТР. 021013.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

23. PC817x Series Photocoupler [Online resource]. Access: https://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/pc817xnsz_e.pdf.

24. TIP35C TIP36C Power Transistors [Online resource]. Access: <https://www.st.com/resource/en/datasheet/tip35c.pdf>.

25. SS14 Surface Mount Schottky Barrier Diode [Online resource]. Access: <https://www.alldatasheet.com/datasheet-pdf/pdf/25784/SURGE/SS14.html>.

26. Моделювання і оптимізація радіотехнічних засобів електронних комунікацій : методичні рекомендації до виконання курсової роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 172 «Електронні комунікації та радіотехніка» / О. С. Пивовар. Хмельницький : ХНУ, 2024, 38 с.

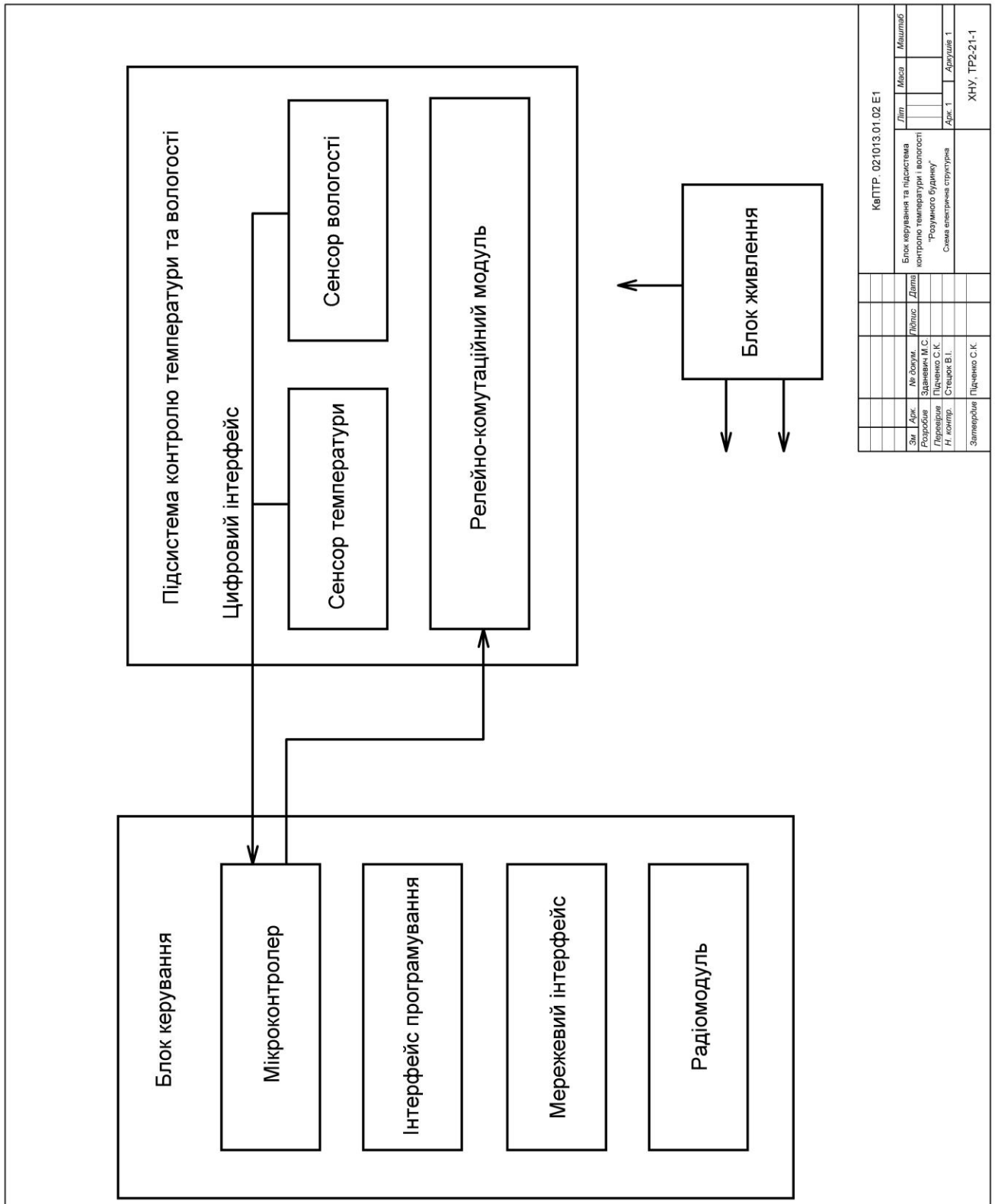
27. Державні будівельні норми України В.2.5-67:2022 [Електронний ресурс] / Мінрегіон України. – Київ, 2022. – Режим доступу: <https://dbn.co.ua/load/normativy/dbn/1>.

28. Офіційна специфікація протоколу MQTT в. 5.0: MQTT Version 5.0. OASIS Standard, 07 March 2019 [Електронний ресурс]. Режим доступу: URL: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.

ДОДАТОК А

(обов'язковий)

Копія схеми електричної структурної

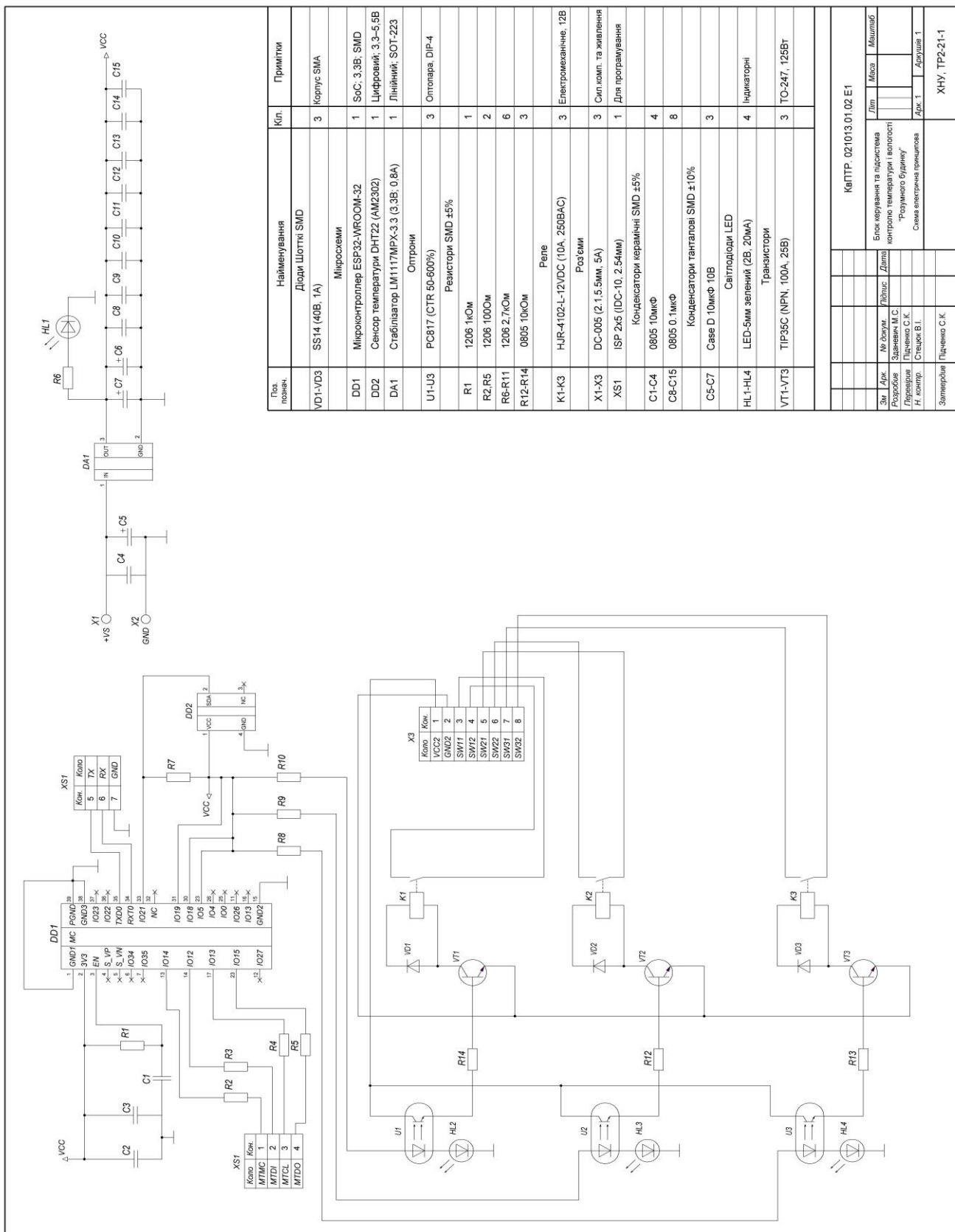


КвЛТР: 021013.01.02 Е1											
Блок керування та підсистема контролю температури і вологості "Роумного Будинку"				Лип		Месо		Маштаб			
Схеми електрична структура				Арс. 1		Аркушів 1					
								ЖНУ, ТР2-21-1			
Зм	Арх	М. Дорж	Поліс	Датів							
Розробив	Затвердив	М. С.									
Проєктував	Підписав	С. К.									
Н. Копир	Стецько	В. І.									
Затвердив	Підписав	С. К.									

ДОДАТОК Б

(обов'язковий)

Копія схеми електричної принципової



ДОДАТОК В

Код програмного забезпечення

```
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;
using SmartIOTSystem.Data;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
var connectionString = builder.Configuration.GetConnectionString("DefaultConnectionString") ??
    throw new InvalidOperationException("Connection string 'DefaultConnectionString' not found.");
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(connectionString));
builder.Services.AddDatabaseDeveloperPageExceptionFilter();

builder.Services.AddDefaultIdentity<IdentityUser>(options =>
    options.SignIn.RequireConfirmedAccount = true)
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddControllersWithViews();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseMigrationsEndPoint();
}
else
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see
    https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.MapRazorPages();

app.Run();
```

```

using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using SmartIOTSystem.Models;

namespace SmartIOTSystem.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using SmartIOTSystem.Data;
using SmartIOTSystem.Models;

namespace SmartIOTSystem.Controllers
{
    public class IoTUnitsController : Controller

```

```

{
    private readonly ApplicationDbContext _context;

    public IoTUnitsController(ApplicationDbContext context)
    {
        _context = context;
    }

    // GET: IoTUnits
    [Authorize]
    public async Task<IActionResult> Index()
    {
        return View(await _context.IoTUnits.ToListAsync());
    }

    // GET: IoTUnits/Details/5
    [Authorize]
    public async Task<IActionResult> Details(int? id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var ioTUnit = await _context.IoTUnits
            .FirstOrDefaultAsync(m => m.Id == id);
        if (ioTUnit == null)
        {
            return NotFound();
        }

        return View(ioTUnit);
    }

    // GET
    [Authorize]
    public IActionResult GetChartData(int? id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var ioTUnit = _context.IoTUnits.Include(m => m.DataRecords)
            .FirstOrDefault(m => m.Id == id);
        var records = ioTUnit.DataRecords.ToList().Select(r=>r.Value);

        var result = new
        {
            Name = ioTUnit.Name,
            Subsystem = ioTUnit.SubSystemName,

```

```

        Units = ioTUnit.MeasurementUnit,
        Values = ioTUnit.DataRecords.ToList().Select(r => r.Value),
        TimeStamps = ioTUnit.DataRecords.ToList().Select(r => r.Time)
    };

    return Json(result);
}

// GET: IoTUnits/Create
[Authorize]
public IActionResult Create()
{
    return View();
}

// POST: IoTUnits/Create
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize]
public async Task<IActionResult>
Create([Bind("Id,DeviceId,Name,Description,SubSystemName,MeasurementUnit,ZeroShift,CurveSlope")] IoTUnit ioTUnit)
{
    if (ModelState.IsValid)
    {
        _context.Add(ioTUnit);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(ioTUnit);
}

// GET: IoTUnits/Edit/5
[Authorize]
public async Task<IActionResult> Edit(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var ioTUnit = await _context.IoTUnits.FindAsync(id);
    if (ioTUnit == null)
    {
        return NotFound();
    }
    return View(ioTUnit);
}

```

```

// POST: IoTUnits/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize]
public async Task<IActionResult> Edit(int id,
[Bind("Id,DeviceId,Name,Description,SubSystemName,MeasurementUnit,ZeroShift,CurveSlope")]
IoTUnit ioTUnit)
{
    if (id != ioTUnit.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(ioTUnit);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!IoTUnitExists(ioTUnit.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(ioTUnit);
}

// GET: IoTUnits/Delete/5
[Authorize]
public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var ioTUnit = await _context.IoTUnits
        .FirstOrDefaultAsync(m => m.Id == id);
    if (ioTUnit == null)
    {

```

```

        return NotFound();
    }

    return View(ioTUnit);
}

// POST: IoTUnits/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
[Authorize]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    var ioTUnit = await _context.IoTUnits.FindAsync(id);
    if (ioTUnit != null)
    {
        _context.IoTUnits.Remove(ioTUnit);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool IoTUnitExists(int id)
{
    return _context.IoTUnits.Any(e => e.Id == id);
}
}
}

```

```

using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;
using SmartIOTSystem.Models;

```

```

namespace SmartIOTSystem.Data
{
    public class ApplicationDbContext : IdentityDbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }

        public DbSet<IoTUnit> IoTUnits { get; set; }

        public DbSet<DataRecord> DataRecords { get; set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            base.OnModelCreating(modelBuilder);
        }
    }
}

```

```

// Configure the 1-to-many relationship
modelBuilder.Entity<DataRecord>()
    .HasOne(record => record.IoTUnit)
    .WithMany(unit => unit.DataRecords)
    .HasForeignKey(record => record.IoTUnitId)
    .OnDelete(DeleteBehavior.Cascade); // Delete records when unit is deleted
    }
}
}

using System;
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace SmartIOTSystem.Data.Migrations
{
    /// <inheritdoc />
    public partial class initial_migration : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "IoTUnits",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    DeviceId = table.Column<int>(type: "int", nullable: false),
                    Name = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Description = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    SubSystemName = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    MeasurementUnit = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    ZeroShift = table.Column<double>(type: "float", nullable: false),
                    CurveSlope = table.Column<double>(type: "float", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_IoTUnits", x => x.Id);
                });

            migrationBuilder.CreateTable(
                name: "DataRecords",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),

```

```

        Time = table.Column<DateTime>(type: "datetime2", nullable: false),
        Value = table.Column<double>(type: "float", nullable: false),
        IoTUnitId = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_DataRecords", x => x.Id);
        table.ForeignKey(
            name: "FK_DataRecords_IoTUnits_IoTUnitId",
            column: x => x.IoTUnitId,
            principalTable: "IoTUnits",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });

    migrationBuilder.CreateIndex(
        name: "IX_DataRecords_IoTUnitId",
        table: "DataRecords",
        column: "IoTUnitId");
}

/// <inheritdoc />
protected override void Down(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropTable(
        name: "DataRecords");

    migrationBuilder.DropTable(
        name: "IoTUnits");
}
}
}

```

```

namespace SmartIOTSystem.Models
{
    public class DataRecord
    {
        public int Id { get; set; } // Primary key

        public DateTime Time { get; set; }

        public double Value { get; set; }

        public int IoTUnitId { get; set; } // Foreign key

        public IoTUnit IoTUnit { get; set; } = null!; // Navigation property

        public double GetCalibratedValue()
        {

```

```

    double slope = 1, shift = 0;
    if (IoTUnit != null)
    {
        slope = IoTUnit.CurveSlope;
        shift = IoTUnit.ZeroShift;
    }
    return Value * slope + shift;
}
}
}

```

```

namespace SmartIOTSystem.Models
{
    public class ErrorViewModel
    {
        public string? RequestId { get; set; }

        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}

```

```

namespace SmartIOTSystem.Models
{
    public class IoTUnit
    {
        public int Id { get; set; } // Primary key

        public int DeviceId { get; set; }

        public string Name { get; set; } = null!;

        public string Description { get; set; } = null!;

        public string SubSystemName { get; set; } = null!;

        public string MeasurementUnit { get; set; } = null!;

        // Calibration coefficient x = value * CurveSlope + ZeroShift

        public double ZeroShift { get; set; }

        public double CurveSlope { get; set; } = 1.0;

        // Navigation property - 1 IoTUnit can have many DataRecords
        public ICollection<DataRecord> DataRecords { get; set; } = new List<DataRecord>();
    }
}

```

```
}
```

```
@{  
    ViewData["Title"] = "Home Page";  
}
```

```
<div class="text-center">  
    <h1 class="display-4">Welcome</h1>  
    <p>Learn about <a href="https://learn.microsoft.com/aspnet/core">building Web apps with  
ASP.NET Core</a>.</p>  
</div>
```

```
@model IEnumerable<SmartIOTSystem.Models.IoTUnit>
```

```
@{  
    ViewData["Title"] = "Index";  
}
```

```
<h1>Index</h1>  
  
<p>  
    <a asp-action="Create">Create New</a>  
</p>  
<table class="table">  
    <thead>  
        <tr>  
            <th>Device Id</th>  
            <th>Name</th>  
            <th>Description</th>  
            <th>Subsystem</th>  
        </tr>  
    </thead>  
    <tbody>  
        @foreach (var item in Model) {  
            <tr>  
                <td>  
                    @Html.DisplayFor(modelItem => item.DeviceId)  
                </td>  
                <td>  
                    @Html.DisplayFor(modelItem => item.Name)  
                </td>  
                <td>  
                    @Html.DisplayFor(modelItem => item.Description)  
                </td>  
                <td>  
                    @Html.DisplayFor(modelItem => item.SubSystemName)  
                </td>  
            </tr>  
        }  
    </tbody>  
</table>
```

```

        <td>
            <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |
            <a asp-action="Details" asp-route-id="@item.Id">Details</a> |
            <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
        </td>
    </tr>
}
</tbody>
</table>

```

```
@model SmartIOTSystem.Models.IoTUnit
```

```

@{
    ViewData["Title"] = "Details";
}

```

```
<h1>Details</h1>
```

```

<div>
    <h4>IoTUnit</h4>
    <hr />
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.DeviceId)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.DeviceId)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Name)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Name)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Description)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Description)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.SubSystemName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.SubSystemName)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.MeasurementUnit)
        </dt>
    </div>

```

```

<dd class = "col-sm-10">
  @Html.DisplayFor(model => model.MeasurementUnit)
</dd>
<dt class = "col-sm-2">
  @Html.DisplayNameFor(model => model.ZeroShift)
</dt>
<dd class = "col-sm-10">
  @Html.DisplayFor(model => model.ZeroShift)
</dd>
<dt class = "col-sm-2">
  @Html.DisplayNameFor(model => model.CurveSlope)
</dt>
<dd class = "col-sm-10">
  @Html.DisplayFor(model => model.CurveSlope)
</dd>
</dl>
</div>
<div>
  <a asp-action="Edit" asp-route-id="@Model?.Id">Edit</a> |
  <a asp-action="Index">Back to List</a>
</div>

<div class="container" id="item-container" data-item-id="@Model?.Id">
  <div>
    <canvas style="max-width: 100%; max-height: 650px" id="myChart"></canvas>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

  <script>
    const ctx = document.getElementById('myChart');
    const itemId = document.getElementById('item-container').dataset.itemId;
    const url = `/IoTUnits/GetChartData/${itemId}`;
    fetch(url)
      .then(response => response.json())
      .then(data => {
        console.log(data);
        new Chart(ctx, {
          type: 'line',
          data: {
            labels: data.timeStamps.map(d => (
              (new Date(d)).toTimeString().split(' ')[0]
            )),
            datasets: [{
              label: `${data.subsystem} / ${data.name} (${data.units})`,
              data: data.values,
              borderWidth: 1
            }]
          },
          options: {
            scales: {

```

```

        y: {
            beginAtZero: true
        }
    }
});
});
</script>
</div>

```

```
@model SmartIOTSystem.Models.IoTUnit
```

```
@{
    ViewData["Title"] = "Edit";
}
```

```
<h1>Edit</h1>
```

```
<h4>IoT Unit</h4>
```

```
<hr />
```

```
<div class="row">
```

```
    <div class="col-md-4">
```

```
        <form asp-action="Edit">
```

```
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
            <input type="hidden" asp-for="Id" />
```

```
            <div class="form-group">
```

```
                <label asp-for="DeviceId" class="control-label">Device ID</label>
```

```
                <input asp-for="DeviceId" class="form-control" />
```

```
                <span asp-validation-for="DeviceId" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Name" class="control-label">Name</label>
```

```
                <input asp-for="Name" class="form-control" />
```

```
                <span asp-validation-for="Name" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Description" class="control-label">Description</label>
```

```
                <input asp-for="Description" class="form-control" />
```

```
                <span asp-validation-for="Description" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="SubSystemName" class="control-label">Subsystem</label>
```

```
                <input asp-for="SubSystemName" class="form-control" />
```

```
                <span asp-validation-for="SubSystemName" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="MeasurementUnit" class="control-label">Units</label>
```

```
                <input asp-for="MeasurementUnit" class="form-control" />
```

```
                <span asp-validation-for="MeasurementUnit" class="text-danger"></span>
```

```

</div>
<div class="form-group">
  <label asp-for="ZeroShift" class="control-label">Zero Shift</label>
  <input asp-for="ZeroShift" class="form-control" />
  <span asp-validation-for="ZeroShift" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="CurveSlope" class="control-label">Curve Slope</label>
  <input asp-for="CurveSlope" class="form-control" />
  <span asp-validation-for="CurveSlope" class="text-danger"></span>
</div>
<div class="form-group">
  <input type="submit" value="Save" class="btn btn-primary" />
</div>
</form>
</div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

```

```

@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

```

@model SmartIOTSystem.Models.IoTUnit

```

```

@{
  ViewData["Title"] = "Create";
}

```

```

<h1>Create</h1>

```

```

<h4>IoT Unit</h4>

```

```

<hr />

```

```

<div class="row">
  <div class="col-md-4">
    <form asp-action="Create">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="DeviceId" class="control-label">Device ID</label>
        <input asp-for="DeviceId" class="form-control" />
        <span asp-validation-for="DeviceId" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Name" class="control-label">Name</label>
        <input asp-for="Name" class="form-control" />
        <span asp-validation-for="Name" class="text-danger"></span>
      </div>
    </form>
  </div>
</div>

```

```

</div>
<div class="form-group">
  <label asp-for="Description" class="control-label">Description</label>
  <input asp-for="Description" class="form-control" />
  <span asp-validation-for="Description" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="SubSystemName" class="control-label">Subsystem</label>
  <input asp-for="SubSystemName" class="form-control" />
  <span asp-validation-for="SubSystemName" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="MeasurementUnit" class="control-label">Units</label>
  <input asp-for="MeasurementUnit" class="form-control" />
  <span asp-validation-for="MeasurementUnit" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="ZeroShift" class="control-label">Zero Shift</label>
  <input asp-for="ZeroShift" class="form-control" />
  <span asp-validation-for="ZeroShift" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="CurveSlope" class="control-label">Curve Slope</label>
  <input asp-for="CurveSlope" class="form-control" />
  <span asp-validation-for="CurveSlope" class="text-danger"></span>
</div>
<div class="form-group">
  <input type="submit" value="Create" class="btn btn-primary" />
</div>
</form>
</div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

```

```

@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - SmartIOTSystem</title>
  <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
  <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

```

```

<link rel="stylesheet" href="~/SmartIOTSystem.styles.css" asp-append-version="true" />
</head>
<body>
  <header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
      <div class="container-fluid">
        <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">SmartIOTSystem</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
          <ul class="navbar-nav flex-grow-1">
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-controller="IoTUnits" asp-action="Index">IoT Units</a>
            </li>
          </ul>
          <partial name="_LoginPartial" />
        </div>
      </div>
    </nav>
  </header>
  <div class="container">
    <main role="main" class="pb-3">
      @RenderBody()
    </main>
  </div>

  <footer class="border-top footer text-muted">
    <div class="container">
      &copy; 2025 - SmartIOTSystem - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
    </div>
  </footer>
  <script src="~/lib/jquery/dist/jquery.min.js"></script>
  <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
  <script src="~/js/site.js" asp-append-version="true"></script>
  @await RenderSectionAsync("Scripts", required: false)
</body>

```

</html>

/* Please see documentation at <https://learn.microsoft.com/aspnet/core/client-side/bundling-and-minification>

for details on configuring this project to bundle and minify static web assets. */

```
a.navbar-brand {  
  white-space: normal;  
  text-align: center;  
  word-break: break-all;  
}
```

```
a {  
  color: #0077cc;  
}
```

```
.btn-primary {  
  color: #fff;  
  background-color: #1b6ec2;  
  border-color: #1861ac;  
}
```

```
.nav-pills .nav-link.active, .nav-pills .show > .nav-link {  
  color: #fff;  
  background-color: #1b6ec2;  
  border-color: #1861ac;  
}
```

```
.border-top {  
  border-top: 1px solid #e5e5e5;  
}
```

```
.border-bottom {  
  border-bottom: 1px solid #e5e5e5;  
}
```

```
.box-shadow {  
  box-shadow: 0 .25rem .75rem rgba(0, 0, 0, .05);  
}
```

```
button.accept-policy {  
  font-size: 1rem;  
  line-height: inherit;  
}
```

```
.footer {  
  position: absolute;  
  bottom: 0;  
  width: 100%;  
  white-space: nowrap;  
  line-height: 60px;  
}
```

Система «Розумний будинок» з віддаленим керуванням

Матеріали до захисту
кваліфікаційного проекту

Виконав: студент IV курсу, група TP2-21-1 М. С. Зданевич

Керівник: д.т.н., професор С.К. Підченко

Вступ - Загальні відомості

Автоматизовані технології використовуються все частіше в домашніх приміщеннях з метою покращення умов життя людей, підвищення безпеки та загального комфорту тощо. Розвиток інформаційно-комунікаційних технологій (ІКТ) сприяв проникненню розумних малопотужних електронних обчислювальних машин (ЕОМ) та безпроводових засобів передачі інформації у сфері домогосподарства. З іншого боку, розвиток комп'ютерної техніки та мережевих технологій, наприкладі глобальної системи Інтернет, посприяв виникненню концепції «Інтернету речей – Internet Of Things, (IoT) – мережі розумних пристроїв на базі мікропроцесорів (МП) та мікроконтролерів (МК), сенсорно-актуаторних вузлів (САВ) та мережевих інтерфейсів.

Ієрархія Інтернету речей

- Відповідно до прийнятої архітектури IoT [1-2], виділяються такі рівні ієрархії:
- 1. Рівень «розумних» пристроїв, що забезпечуються збір та обробку даних об'єктів вимірювання та контролю в реальному часі, а також керування об'єктом – актуаторна функція.
- 2. Мережевий рівень та шлюз для інтеграції із протоколами Інтернет.
- 3. Рівень послуг щодо віддаленого керування пристроями IoT, засоби інтеграції х прикладним програмним забезпеченням (ПЗ), прикладні інтерфейси програмування – Application programming interface (API).
- 4. Прикладний рівень – верхній рівень додатків для використання кінцевим користувачем (оператором) системи.

Ієрархія Інтернету речей (продовження)

На Рисунку 1 показані деякі головні компоненти IoT та взаємні потоки інформації між ними відповідно до вищеописаної ієрархії. Так. Потоки даних надходять на Контролер, де обробляються в реальному часі (попередня обробка), потім оброблені та стиснені дані передаються Серверу обробки для здійснення, наприклад, інтелектуального аналізу даних; дані акумулюються у Сховищі.



Рисунок 1 – Схема інформаційних процесів між компонентами мережі IoT [2]

Концепція «Розумного будинку»

Розглянемо в якості прикладу IoT систему «Розумний будинок» відповідно до тематики кваліфікаційного проєкту. Відомо [3], що даний термін вперше був запропонований асоціацією забудовників у США в 1984 році в для позначення заміського будинку, відмінною особливістю якого є ефективне використання житлового середовища.

Відповідно до визначення, наведеного у бібліографічному покажчику [3], «Розумний будинок» – це сучасне житлове приміщення, яке інтегроване із сучасними засобами автоматизації, інфокомунікацій та інтелектуальної підтримки.

Типовими компонентами для такої системи є модуль кліматконтролю, засоби безпеки (наприклад, відеоспостереження), підсистема контролю освітлення, засоби захисту від несанкціонованого проникнення (розумні замки), а також засоби аудіовізуальної підтримки.

Концепція «Розумного будинку» (продовження)

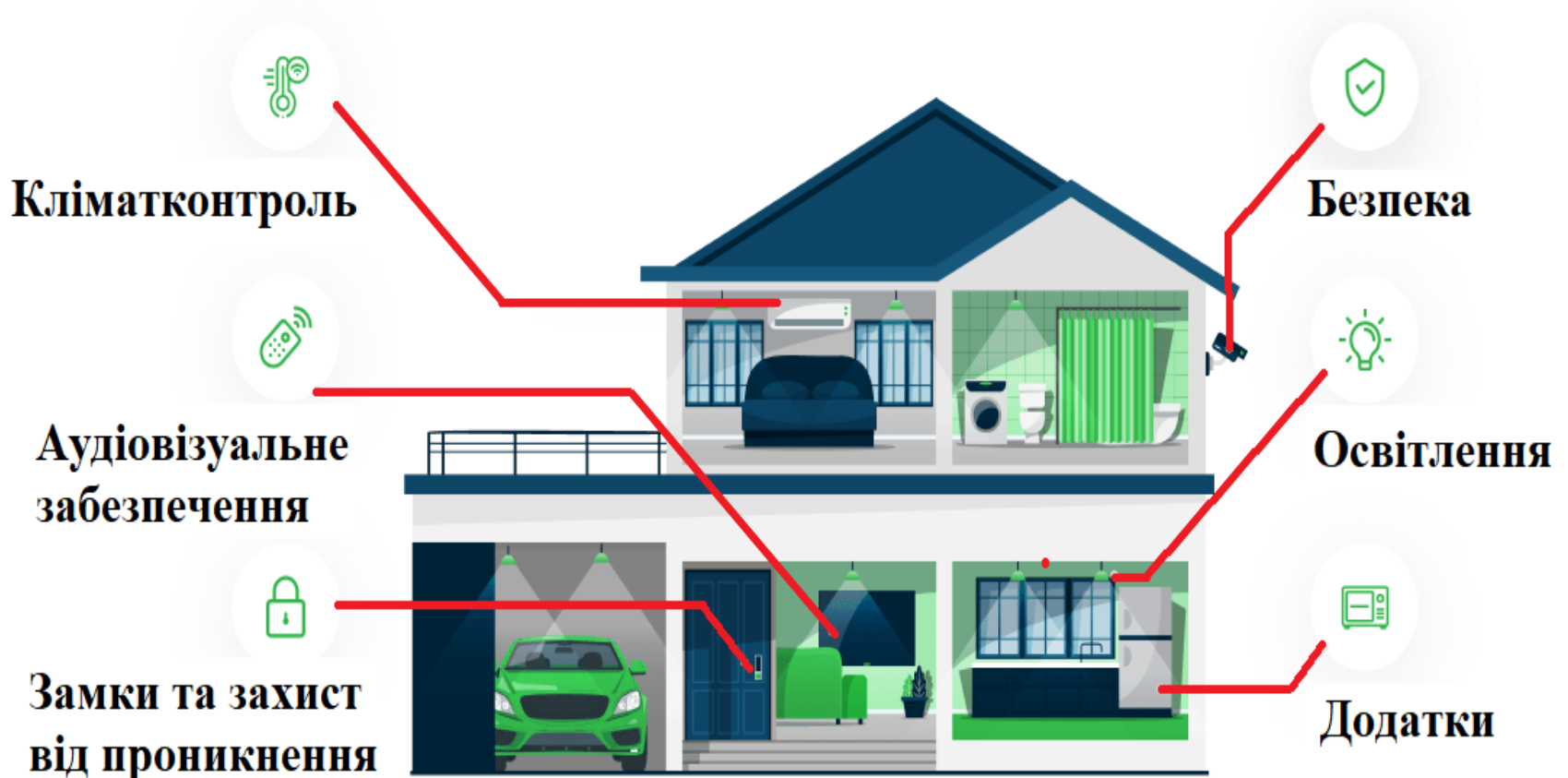


Рисунок 2 – Концепція «Розумного будинку» [4]

Концепція «Розумного будинку» (продовження)

Підсистема керування кліматом в приміщенні функціонує з метою підтримки оптимального рівня температури повітря, вологості, рівня притоку свіжого повітря тощо.

До головних функцій підсистеми освітлення належить підтримка бажаного рівня освітленості в житловому приміщенні для заданого часу доби, балансування між штучними та природними джерелами світла, керування рівнем освітленості.

Важливим компонентом «Розумного будинку» є підсистема безпеки, яку умовно можна розділити на дві складові: систему розумних замків та захисту від проникнення, а також систему відеоспостереження та контролю за людьми в приміщенні, зокрема за допомогою сенсорів руху.

Підсистема аудіовізуалізації призначена для підвищення комфорту мешканців, забезпечення дозвілля, а також розвитку дітей.

Окремо варто відзначити набір додатків для керування системою оператором (головним авторизованим користувачем системи з відповідними правами доступу).

Постановка технічного завдання

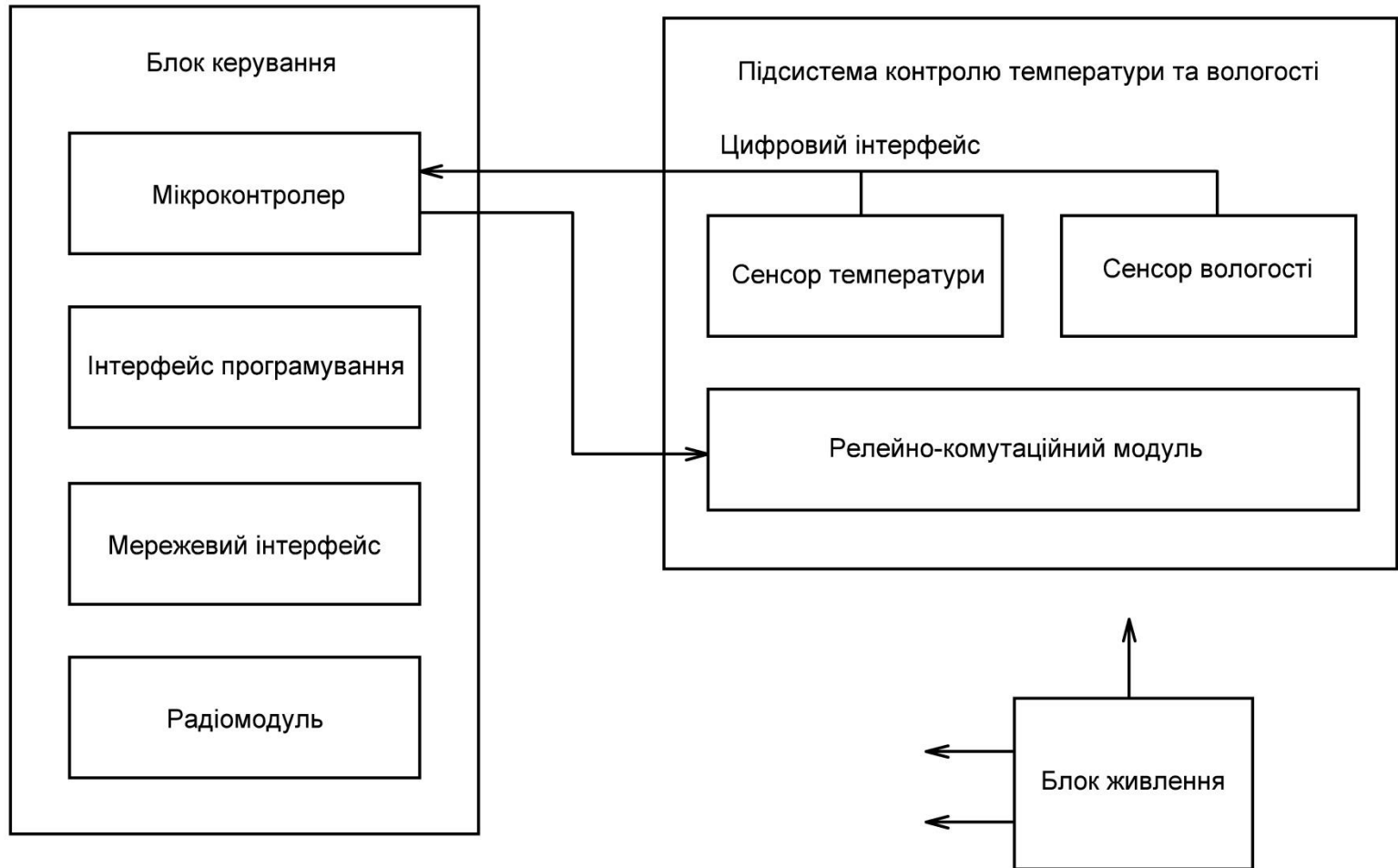
Для досягнення поставленої мети проєктування до вирішення приймаються такі задачі:

1. Розробка структури системи «Розумний будинок», яка включає апаратну частину, ПЗ, а також та комунікаційні канали з можливістю гнучкого розширення за рахунок підключення нових підсистем в процесі експлуатації. В якості базової підсистем розробити: підсистему контролю температури та вологості в приміщенні.

2. Розробка схеми електричної принципової блоку керування на базі МК для системи «Розумний» будинок, а також підсистеми керування температурою та вологістю, яка включає в себе сенсорну частину та комутаційні механізми для керування вентиляцією та зволожувачем повітря.

3. Розробка алгоритму керування та ПЗ для компонентів системи «Розумний будинок» та підсистеми контролю температури і вологості. Розробити ПЗ для блоку керування та проміжних обчислювальних вузлів, базу даних (БД) вимірюваних величин та верхній рівень інтерфейсу користувача для віддаленого керування системою.

Схема електрична структурна



КвПТР. 021013.01.02 Е1						Лист	Маса	Маштаб
Зм.	Арк.	№ докум.	Годпис	Дата	Блок керування та підсистема контролю температури і вологості "Розумного Будинку" Схема електрична структурна			
Розробив		Зданевич М.С.						
Перевірив		Підченко С.К.				Арк. 1		Аркуше 1
Н. контр.		Стецюк В.І.						
Затвердив		Підченко С.К.						

Опис структурної схеми

Відправним кроком розробки структури системи «Розумний будинок» є проектування блоку її керування (БК). Це головний компонент системи, який базується на МК з можливістю обробки даних, прийняття рішень та комунікації з іншими пристроями. Побудований на базі МК, наприклад, ESP32 або STM32, БК використовується для зчитування інформації з сенсорів, аналізу отриманих даних та керування виконавчими пристроями.

З метою завантаження та налагодження ПЗ необхідно передбачити інтерфейс програмування, наприклад UART або JTAG. В той же час наявність мережевого інтерфейсу TCP/IP необхідна для підключення до локальної мережі та Інтернету, для здійснення віддаленого моніторингу і керування через додаток або веб-інтерфейс. Для реалізації віддаленого доступу необхідно також передбачити безпроводовий канал зв'язку, який реалізований, наприклад, за допомогою радіомодуля Wi-Fi, що також забезпечує інтеграцію з іншими IoT-пристроями.

Отже, БК системи складається з таких елементів: МК, інтерфейс програмування, інтерфейсу програмування та налагодження, мережевого інтерфейсу TCP/IP та радіомодуля Wi-Fi.

Опис структурної схеми (продовження)

Наступним кроком проектування структури системи є розробка підсистема контролю температури та вологості, яка відповідає за моніторинг кліматичних параметрів і автоматичне регулювання. Така підсистема включає сенсори температури та вологості, наприклад DHT22 або SHT31, що вимірюють поточні значення та передають їх до МК. На основі отриманих даних система керує релейно-комутаційним модулем, який активує або деактивує вентилятори та зволожувач повітря для підтримки оптимального мікроклімату. Отже, підсистема контролю температури та вологості буде складатись з таких компонентів: сенсор вологості та сенсор температури, релейно-комутаційний інтерфейс для керування вентиляторами та зволожувачем повітря.

Для живлення схеми необхідно реалізувати блок живлення (БЖ), що може бути побудований на базі лінійного стабілізатора, наприклад, AMS1117-3.3.

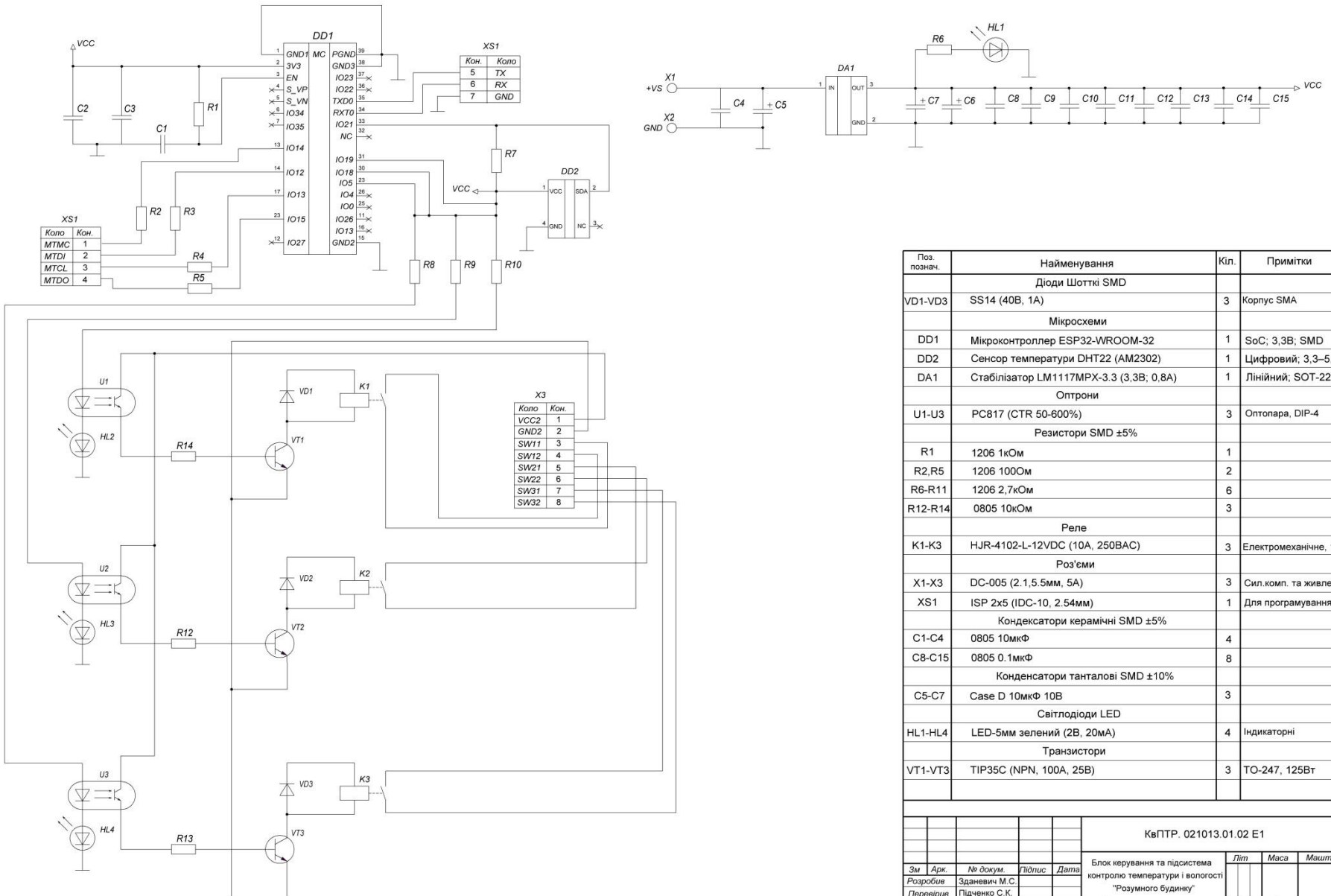
З урахуванням вищесказаного, представимо на Рисунку 2.1 структурну схему БК системи «Розумний будинок», а також підсистему контролю температури та вологості.

Вибір компонентів для реалізації

Таким чином, усі головні компоненти та їх ТТХ були описані в підрозділі 2.2 кваліфікаційного проекту «Обґрунтування вибору компонентів системи». Для більшої наочності в Таблиці 2.1 подано відповідність між елементами структурної схеми та обраними компонентами для практичної реалізації пристрою.

Елемент структурної схеми		Обраний компонент для реалізації
Блок керування	Мікроконтролера	ESP32 SoC
	Мережевий інтерфейс	
	Радіомодуль Wi-Fi	
	Інтерфейс програмув.	JTAG Connector 6 pin
Підсистема контролю температури та вологості	Сенсор температури	DHT22
	Сенсор вологості	
	Релейно-комутаційний модуль	Реле HJR-4102-L-12VDC
		Оптрон PC817
		Транзистор напівпров. TIP35C NPN
	Діод Шоттки SS14	

Схема електрична принципова



Поз. познач.	Найменування	Кіл.	Примітки
	Діоди Шотткі SMD		
VD1-VD3	SS14 (40B, 1A)	3	Корпус SMA
	Мікросхеми		
DD1	Мікроконтролер ESP32-WROOM-32	1	SoC; 3,3В; SMD
DD2	Сенсор температури DHT22 (AM2302)	1	Цифровий; 3,3-5,5В
DA1	Стабілізатор LM1117MPX-3.3 (3,3В; 0,8А)	1	Лінійний; SOT-223
	Оптрони		
U1-U3	PC817 (CTR 50-600%)	3	Оптопара, DIP-4
	Резистори SMD ±5%		
R1	1206 1кОм	1	
R2,R5	1206 100Ом	2	
R6-R11	1206 2,7кОм	6	
R12-R14	0805 10кОм	3	
	Реле		
K1-K3	HJR-4102-L-12VDC (10A, 250VAC)	3	Електромеханічне, 12В
	Роз'єми		
X1-X3	DC-005 (2,1,5,5мм, 5А)	3	Сил. комп. та живлення
XS1	ISP 2x5 (IDC-10, 2,54мм)	1	Для програмування
	Конденсатори керамічні SMD ±5%		
C1-C4	0805 10мкФ	4	
C8-C15	0805 0,1мкФ	8	
	Конденсатори танталові SMD ±10%		
C5-C7	Case D 10мкФ 10В	3	
	Світлодіоди LED		
HL1-HL4	LED-5мм зелений (2В, 20мА)	4	Індикаторні
	Транзистори		
VT1-VT3	TIP35C (NPN, 100А, 25В)	3	ТО-247, 125Вт

КвПТР. 021013.01.02 Е1				
Зм.	Арж.	№ докум.	Підпис	Дата
Розробив		Заданевич М.С.		
Перевірив		Підченко С.К.		
Н. контр.		Стецюк В.І.		
Затвердив		Підченко С.К.		
Блок керування та підсистема контролю температури і вологості "Розумного будинку"				
Схема електрична принципова				
		Лит	Маса	Маштаб
		Арж. 1	Аркуше 1	
ХНУ, ТР2-21-1				

Опис принципової схеми

Схема електрична принципова виконана на форматі А2 і додається до матеріалів кваліфікаційного проєкту як обов'язковий елемент конструкторської документації. Копія схеми у повному обсязі з усіма позначеннями та поясненнями подана в Додатку Б. В основі схеми лежить мікроконтролер ESP32 (позначення DD1 на схемі), який виконує функції центрального обчислювального вузла системи.

Для забезпечення гнучкості розробки та налагодження програмного забезпечення до плати виведені інтерфейси Universal Asynchronous Receiver/Transmitter (UART) та JTAG. Інтерфейс (UART) реалізований на виводах 34 (RXD0) та 35 (TXD0) МК, які через захисні елементи заведені на роз'єм XS1 (контакти 5 та 6). Це дозволяє підключати плату до ПК для програмування та моніторингу через стандартний послідовний інтерфейс. Інтерфейс JTAG для низькорівневого відлагодження виведений на виводи 13 (IO14), 14 (IO12), 17 (IO13) та 23 (IO15) МК через струмообмежуючі резистори R2-R5 номіналом 100 Ом і також підключений до роз'єму XS1 (контакти 1-4).

Сенсор температури та вологості DHT22 (позначення DD2 на схемі) підключений за схемою з трьома виводами. Живлення сенсора (вивід 1 VCC) підключено до стабілізованого джерела живлення VCC +3,3В. Інформаційний вивід 2 (SDA) через підтягуючий резистор R7 номіналом 2.7 кОм підключений до виводу 33 (IO21) мікроконтролера DD1, що забезпечує надійний цифровий інтерфейс для передачі даних. Для захисту від перешкод лінія даних додатково фільтрується керамічним конденсатором C3 ємністю 100 нФ.

Опис принципової схеми (продовження)

Релейно-комутаційний модуль реалізований на базі трьох незалежних каналів керування, кожен з яких включає оптронну розв'язку (елементи U1-U3), біполярні транзистори (VT1-VT3), електромагнітне реле (K1-K3) та елементи захисту. Бази транзисторів VT1-VT3 через обмежувальні резистори R12-R14 номіналом 10 кОм підключені до виходів оптронів, що забезпечує гальванічну розв'язку між МК і силовою частиною. Керуючі сигнали з виводів МК 31 (IO19), 30 (IO18) та 23 (IO5) через обмежувальні резистори R8-R10 номіналом 2.7 кОм подаються на входи оптронів. Для захисту від викидів самоіндукції на обмотках реле встановлені діоди Шоттки VD1-VD3. Індикація стану кожного реле здійснюється за допомогою світлодіодів HL2-HL4.

Система живлення побудована на базі лінійного стабілізатора LM1117 (позначення DA1 на схемі) у стандартному включенні з вхідною напругою +5В та вихідною +3,3В. Для фільтрації пульсацій на вході та виході стабілізатора встановлені відповідні фільтри, що виконані на базі конденсаторів C4-C5 та C6-C15 відповідно. Індикація наявності живлення реалізована на світлодіоді HL1 з обмежувальним резистором R6. Підключення живлення +5В здійснюється через роз'єми X1-X2.

Реалізація віддаленого керування на базі протоколу MQTT

Архітектура програмного компоненту проєкту «Розумний будинок» для керування вентиляторами, зволожувачем повітря та моніторингу даних сенсорів побудована на основі протоколу Message Queuing Telemetry Transport (MQTT), який забезпечує ефективний обмін повідомленнями між пристроями в режимі реального часу. Система складається з кількох головних компонентів: МК ESP32, MQTT-брокера, серверного ПЗ та БД під керуванням MS SQL Server.

Відомо [28], що протокол MQTT – це легкий, ефективний та масштабований протокол обміну повідомленнями, розроблений для пристроїв з обмеженими ресурсами, таких як IoT-пристрої, сенсори та МК. Протокол працює за принципом «видавець-підписник» (publish-subscribe), де учасники системи не зв'язані напряму, а спілкуються через центральний вузол – MQTT-брокер.

Згідно архітектури системи БК на базі МК ESP32 виконує роль клієнта MQTT, який підключено до сенсору температури та вологості, а також до актуаторних вузлів, таких як вентилятори та зволожувач повітря. МК ESP32 зчитує дані сенсорів у реальному часі та публікує їх у відповідні MQTT-теми брокера, наприклад, `sensors/temperature/room1` та `sensors/humidity/room1`. Одночасно ESP32 підписані на теми керування, такі як `control/fan/room1` та `control/humidifier/room1`, через які отримуються команди вмикання/вимикання від серверного ПЗ. При отриманні повідомлення МК аналізує його вміст і відправляє відповідні сигнали до виконавчих пристроїв.

MQTT-брокер є центральним вузлом системи, який відповідає за маршрутизацію повідомлень між підписниками та видавцями. Він приймає дані від ESP32 і перенаправляє їх серверному ПЗ, а також передає керуючі команди від сервера до МК. Брокер може бути розгорнутий локально (наприклад, Mosquitto) або використовувати хмарні рішення (AWS IoT, HiveMQ).

Реалізація віддаленого керування на базі протоколу MQTT (продовження)

Серверне ПЗ також є MQTT-клієнтом і підписане на теми сенсорів для отримання оновлень температури та вологості. При отриманні нових даних ПЗ здійснює їх обробку (наприклад, перевірку на межі допустимих значень) та зберігає у БД MS SQL Server для подальшого аналізу та візуалізації. Якщо показники виходять за задані межі, серверне ПЗ публікує команди керування у відповідні топіки, щоб активувати або деактивувати вентилятори та зволожувач. Для реалізації серверного компоненту може використано платформу .NET та мову програмування C#.

В таблиці 3.1 подані основні характеристики модуля віддаленого керування на база протоколу MQTT, зокрема рівні якості обслуговування - Quality of Service (QoS), опис вузлів системи в термінах MQTT та додаткові можливості.

Рівні якості обслуговування QoS	
QoS 0 – лише 1 раз	Повідомлення відправляється один раз без підтвердження. Найшвидший, але ненадійний варіант.
QoS 1 – принаймні 1 раз	Брокер гарантує отримання, але можливі дублікати.
QoS 2 – гарантовано 1 раз	Найнадійніший рівень, де повідомлення доставляється строго один раз за рахунок додаткового етапу підтвердження.
Функціонування в системі «Розумний будинок»	
ESP32 (видавець)	Відправляє показники температури/вологості у відповідні теми.
ESP32 (підписник)	Отримує команди з тем керування та виконує дії.
Сервер (підписник)	Отримує дані, аналізує їх та публікує команди керування (наприклад, увімкнути вентилятор).
Додаткові можливості	
Зберегти останнє / Retained Messages	Брокер зберігає останнє повідомлення в темі та автоматично відправляє його новим підписникам.
Остання команда / Last Will and Testament (LWT)	Якщо клієнт несподівано від'єднується, брокер публікує заздалегідь визначене повідомлення.
Очистка сесії / Clean Session	Визначає, чи зберігатиме брокер підписки клієнта після відключення

Алгоритми програмно забезпечення

Блок-схеми алгоритмів керування температурою та рівнем вологості показані на Рисунках 2.14 та 2.15 відповідно. В основі цих алгоритмів лежить спрощений функціонал контролю заданого значення відповідної величини з урахуванням допустимої похибки та періодичне опитування сенсорів. Алгоритм передбачає встановлення цільового значення шляхом віддаленого керування згідно мети розробки проєкту. Згідно технічного завдання на проєктування керування температурою здійснюється за допомогою двох вентиляторів з метою підвищення ефективності їх одночасної роботи в літній період, в той час як для контролю рівня вологості передбачено використання одного зволожувача повітря.

Таким чином, система працює у замкнутому циклі: сенсори передають дані, сервер аналізує їх і відправляє команди, виконавчі пристрої регулюють клімат у кімнаті. Протокол MQTT забезпечує швидкий та надійний обмін даними, а MS SQL Server дозволяє зберігати та обробляти інформацію для подальшого використання.

Алгоритми програмно забезпечення (продовження)

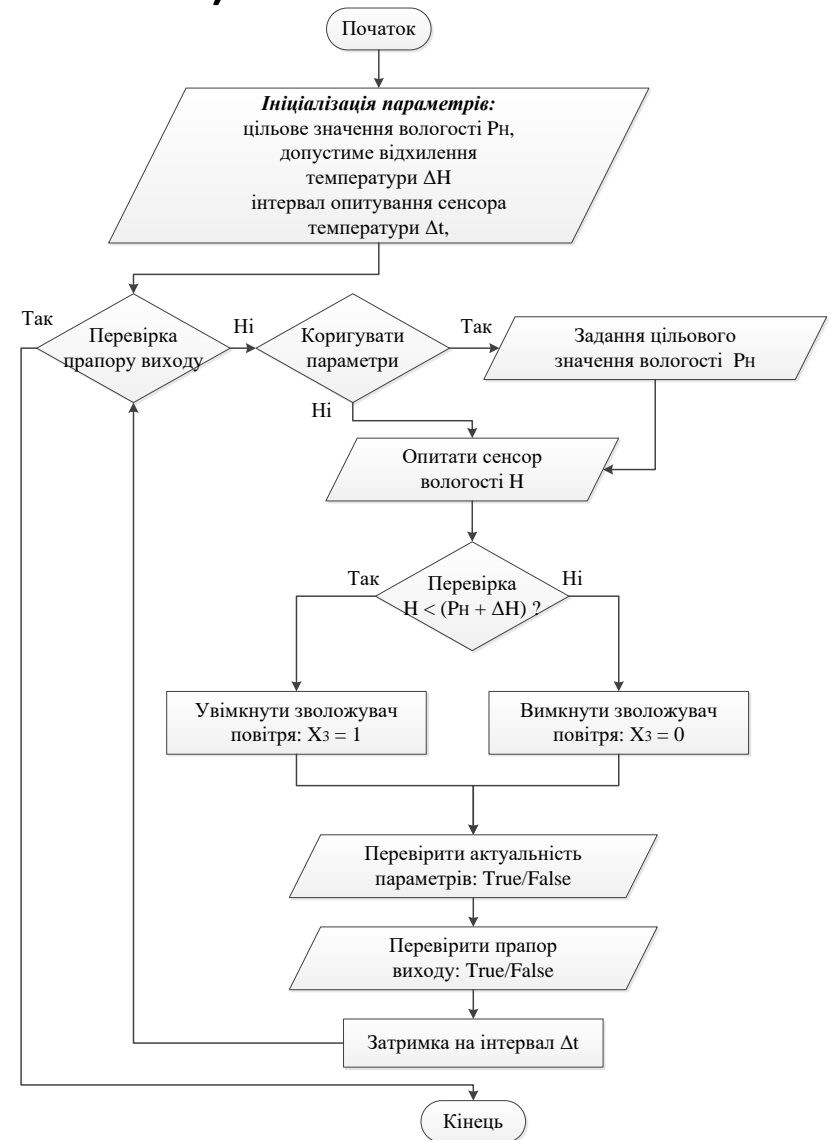
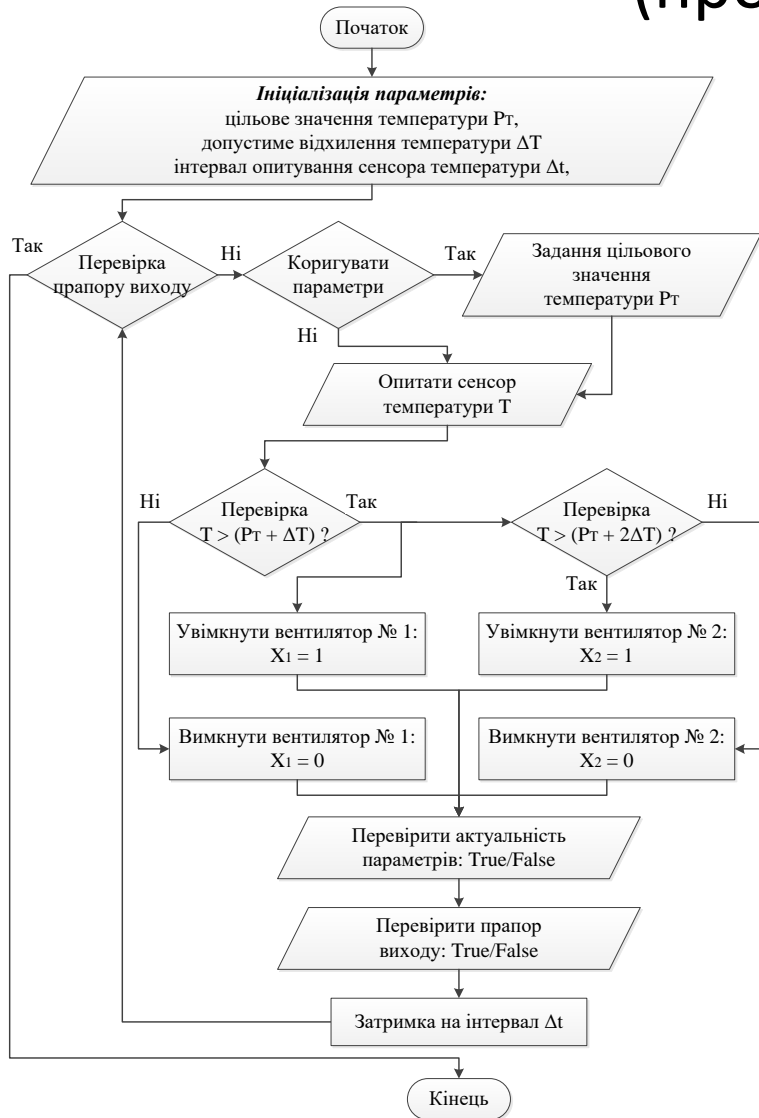


Рисунок 2.14 – Алгоритм керування температурою в «Розумному будинку»

Рисунок 2.15 – Алгоритм керування рівнем вологості в «Розумному будинку»

Програмне забезпечення

Відповідно до декомпозиції системи, серверна частина та інтерфейс взаємодії програми з користувачем представлені такими основними компонентами:

1. Інтерфейс користувача (верхній рівень ПЗ) – представлено у вигляді веб-орієнтованого додатку, який побудований згідно шаблону Модель-Представлення-Контролер (Model-View-Controller, MVC).

2. Сервер-координатор кінцевих пристроїв (проміжне ПЗ).

Елементи інтерфейсу користувача представлені набором HTML-сторінок – View-компонентів, в той час як логіка програми та взаємодія із БД реалізовані у компонентах Controllers (контролери) та Models (моделі даних). Крім того, серверний модуль включає інтерфейс програмної взаємодії (Application Programming Interface, API) для комунікації із кінцевими пристроями системи, керування та акумулювання сенсорних даних.

Робота з базою даних

Для взаємодії з БД на рівні програмного коду, а також для міграції БД виконистано такі додаткові пакети (Рисунок 3.4):

`Microsoft.EntityFrameworkCore,`

`Microsoft.EntityFrameworkCore.SqlServer,`

`Microsoft.EntityFrameworkCore.Tools.`

Entity Framework Core (EF Core) – це сучасний засіб відображення об'єктної бази даних, який дає змогу створювати чистий, портативний і високорівневий рівень доступу до даних за допомогою .NET (C#) для різноманітних баз даних, зокрема SQL Server (локальна та Azure), SQLite, MySQL, PostgreSQL, Oracle та Azure Cosmos DB. Він підтримує запити LINQ, відстеження змін, оновлення та міграції схем.

`Microsoft.EntityFrameworkCore.SqlServer` – це пакет постачальника бази даних EF Core для Microsoft SQL Server і Azure SQL.

Інструменти Entity Framework Core допомагають із завданнями розробки під час проектування. Вони в основному використовуються для керування міграціями та створення каркасу DbContext і типів сутностей шляхом зворотного проектування схеми бази даних.

Пакет `Microsoft.EntityFrameworkCore.Tools` призначений для інструментів PowerShell, які працюють у консолі диспетчера пакетів Visual Studio (PMC).

В якості системи керування БД обрано MS SQL Server Express. На Рисунку 3.5 показано налаштування підключення до СКБД. Дані вимірювання системи містяться в двох головних таблицях: IoTUnit та DataRecord, які пов'язані зв'язком «один до багатьох» – див. Рисунок 3.6.

Робота з базою даних (продовження)

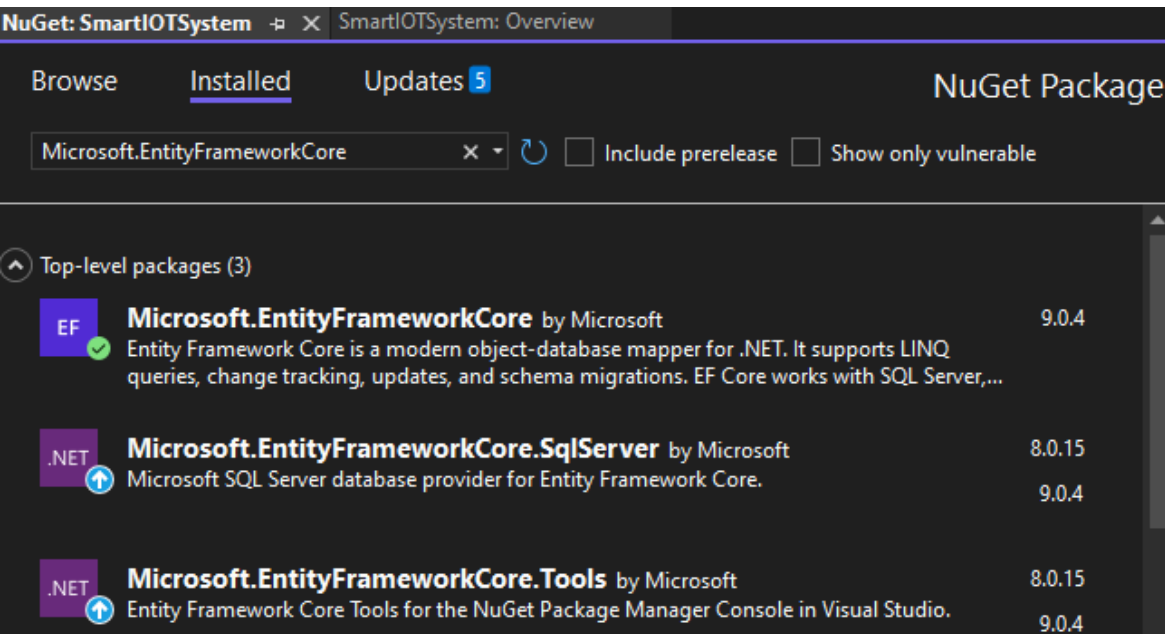


Рисунок 3.4 – Підключення додаткових пакетів для роботи з БД

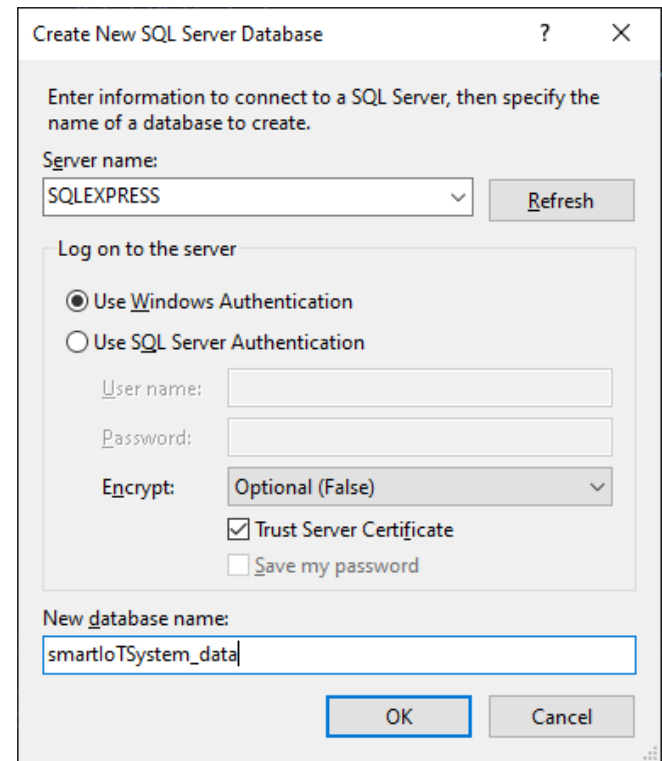


Рисунок 3.5 – Налаштування підключення до MS SQL Server Express

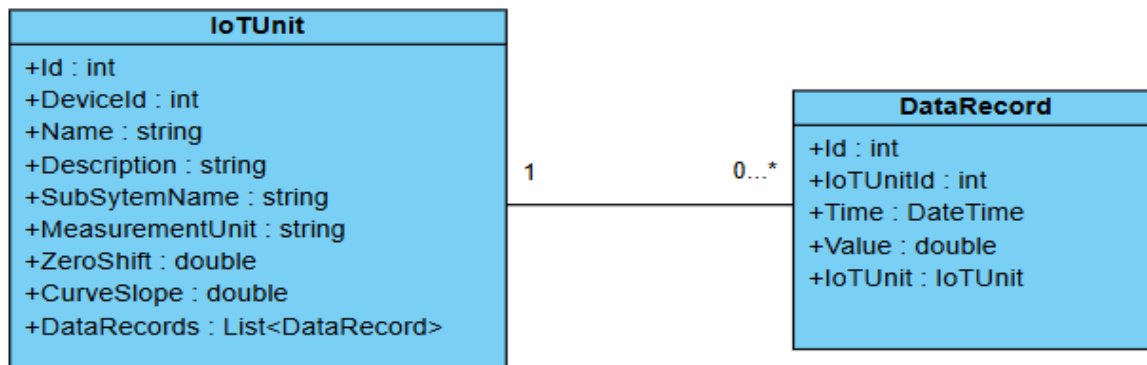


Рисунок 3.6 – Схема бази даних

Програмне забезпечення (продовження)

Головні елементи інтерфейсу користувача:

- Форми авторизації та реєстрації.
- Панель керування та розумними пристроями.

Register

Create a new account.

Email
user@email.com

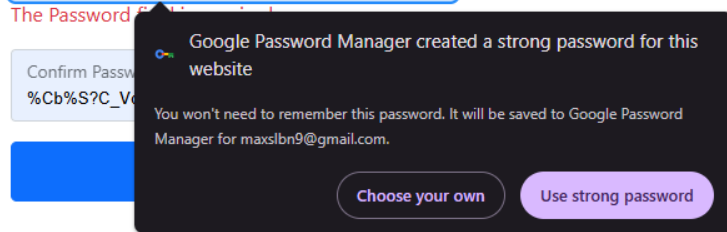
Password
%Cb%S?C_Vc4HEdR

Confirm Password
%Cb%S?C_Vc4HEdR

Register

Use another service to register.

There are no external authentication services configured
[about setting up this ASP.NET application to support log services.](#)



Реєстрація користувача в системі

Log in

Use a local account to log in.

Email
user@email.com

Password
.....

Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)

Hello user@email.com! Logout

Авторизація в системі

Програмне забезпечення (опис інтерфейсу користувача)

Головна панель керування є центральним елементом інтерфейсу, де користувач може швидко отримати доступ до всіх функцій. Тут розміщені елементи керування зі статусом підключених пристроїв, графіки останніх вимірювань та кнопки швидкого керування (включення/виключення пристроїв, налаштування параметрів). Панель також містить навігаційне меню з посиланнями на сторінки зі списком пристроїв, історією даних та налаштуваннями системи. Інтерфейс адаптований для різних типів користувачів, тому адміністратори бачать додаткові інструменти для керування обліковими записами та глобальними налаштуваннями.

Сторінка зі списком розумних пристроїв організована за підсистемами (Рисунок 3.10), що дозволяє зручно групувати обладнання за функціональним призначенням. Кожен пристрій відображається у вигляді картки з основним описом, поточним статусом та значенням останнього вимірювання. Користувач може фільтрувати пристрої за підсистемою, типом або статусом, а також швидко переходити на сторінку детальних параметрів конкретного пристрою. Для зручності на цій сторінці реалізовано пошук за назвою або ідентифікатором, що особливо корисно в системах з великою кількістю підключених пристроїв.

Сторінка параметрів пристрою (Рисунок 3.11) містить детальну інформацію про обраний IoT-пристрій, включаючи калібрувальні коефіцієнти та тестові інструменти для налагодження. Тут користувач може змінювати назву, опис, одиниці вимірювання та інші метадані пристрою. Розділ тестування дозволяє відправляти пробні команди до пристрою та перевіряти його відповіді, що є важливим етапом при інтеграції нового обладнання. Для пристроїв із підтримкою калібрування на цій сторінці доступні інструменти для налаштування ZeroShift та CurveSlope з можливістю перегляду результатів калібрування у реальному часі.

Сторінка візуалізації даних (Рисунок 3.12) демонструє історію вимірювань обраного пристрою за останні 24 години у вигляді графіка. Для тестового режиму використовується генератор випадкових даних, який імітує роботу реального датчика. Користувач може змінювати масштаб графіка, вибирати періоди відображення та порівнювати дані з різних пристроїв. Під графіком розміщена таблиця з детальними значеннями, яку можна експортувати у формати CSV або Excel для подальшого аналізу. Ця сторінка особливо корисна для оперативного моніторингу та виявлення аномалій у роботі підключених пристроїв.

Програмне забезпечення (продовження)

Index

[Create New](#)

Device Id	Name	Description	Subsystem	
1	Temperature Sensor	Temperature control in the room	Temperature Control	Edit Details Delete
2	Fan 1	Cooling Fan 1	Temperature Control	Edit Details Delete
3	Fan 2	Cooling Fan 2	Temperature Control	Edit Details Delete
4	Humidity Sensor	Humidity control in the room	Humidity Control	Edit Details Delete
5	Air Humidifier	Air Humidifier 1	Humidity Control	Edit Details Delete
6	Motion sensor 1	Motion sensor to turn on lights	Light Control	Edit Details Delete
7	Motion sensor 2	Motion sensor to turn on lights	Light Control	Edit Details Delete
8	Motion sensor 3	Motion sensor to turn on lights	Light Control	Edit Details Delete
9	Turn On Lights	Turn On/Off the Lights	Light Control	Edit Details Delete
101	Test Unit 1	Test Unit	Test	Edit Details Delete
102	Test Unit 2	Test Unit	Test	Edit Details Delete

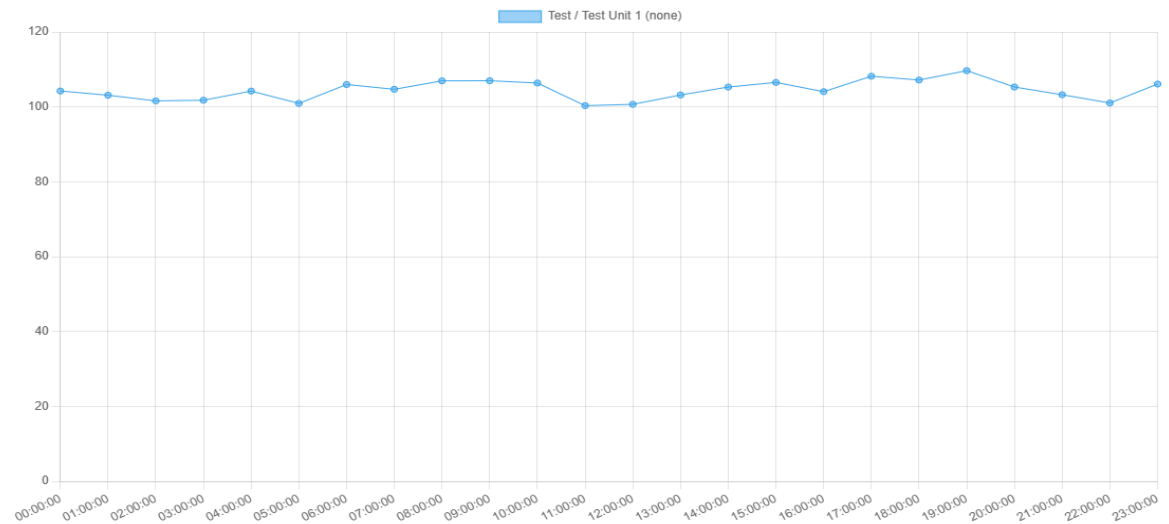
Список Розумних пристроїв в системі, розділених на підсистеми

Програмне забезпечення (продовження)

Details

IoTUnit

Deviceld	101
Name	Test Unit 1
Description	Test Unit
SubSystemName	Test
MeasurementUnit	none
ZeroShift	0
CurveSlope	1



Параметри пристрою (тестовий елемент для налагодження)

Приклад відображення даних вимірювання за 24 години (тестовий набір даних для налагодження програми).

Висновки

1. Виконано аналіз сучасних компонентів систем «Розумного будинку» з приведенням їх тактико-технічних характеристик та економічної рентабельності. Проведено аналіз існуючих технологій та рішень щодо проєктування систем «Розумний будинок» для Інтернету речей.

2. Розроблено структуру системи «Розумний будинок», яка складається з блоку керування, мережевого інтерфейсу з безпроводовим каналом зв'язку та сенсорно-актуаторною частиною для контролю та керування температурою та вологістю в кімнаті «Розумного будинку».

3. Розроблено схему електричну принципову блоку керування та підсистеми моніторингу і контролю вологості і температури на базі мікроконтролера ESP32, цифрового сенсора DHT22 та релейно-комутаційної частини з оптронною розв'язкою для керування вентиляторами та зволожувачем повітря.

4. Розроблено програмне забезпечення для блоку керування, модуля віддаленого доступу на базі протоколу MQTT, а також серверну частину веб-орієнтованого інтерфейсу користувача для налаштування параметрів системи та моніторингу даних сенсорів.

Розроблений прототип системи «Розумний будинок» з віддаленим керуванням для Інтернету речей може бути використаний для тестування, демонстрації та подальшої розробки подібних систем, а також в навчальному процесі для здобувачів освіти за спеціальністю 172 – Телекомунікації та радіотехніка.

Дякую за увагу!

Завідувачу кафедри
телекомунікацій, медійних та
інтелектуальних технологій (ТМІТ)
Сергію ПІДЧЕНКУ
студента 4 курсу, гр. ТР2-21-1
Михайла ЗДАНЕВИЧА

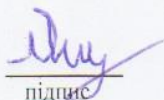
ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщена та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

29.05.25.
дата


підпис

Зданевич М.С.

Anti-Plagiarism (UA) v-15.281 Educational**The maximum coincidence with one document 2.0%**Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 19%**

ID: 244959 Title: Система «Розумний будинок» з віддаленим керуванням Added in a DB: 2025-06-11 Authors: Зданевич Михайло Сергійович Heads: Підченко Сергій Костянтинівич Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	66654	587	5229 (8%)	77 (13%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Михайло ЗДАНЕВИЧ

Співавтор:

Назва: Система «Розумний будинок» з віддаленим керуванням

Експерт: Підченко С.К., д.т.н., проф

Підрозділ: Кафедра телекомунікацій, медійних та інтелектуальних технологій

Коефіцієнт подібності 1:19.5%

Коефіцієнт подібності 2:6.8%

Мікропробіли: 0

Заміна букв: 11

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-11 20:12:02.0

Після аналізу Звіту подібності констатую наступне:


Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

11.06.25р.
Дата

 Підченко С.К.
експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва: Система «Розумний будинок» з віддаленим керуванням

Автор: **Зданевич Михайло Сергійович**

Освітня програма: **Телекомунікації, медійні технології та інтелектуальні мережі**

Рівень вищої освіти перший (бакалаврський) рівень

Спеціальність: **172 Телекомунікації та радіотехніка**

Науковий керівник: **д.т.н., проф. Підченко Сергій Костянтинович**

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія збрала такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	+
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порешень академічної доброчесності	

Підтвердження:

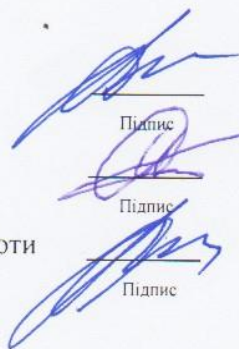
Виявленні запозичення не є плагіатом, так як розміщені у розділах, які не описують безпосередньо авторське дослідження (є власні терміни, визначення тощо), коефіцієнти подібності складають 19,5% та 2,0%, співпадіння мають посилання на приведений список літературних джерел.

11 червня 2025 р.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Сергій ПІДЧЕНКО

Віктор СТЕЦЮК

Сергій ПІДЧЕНКО

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Зданевич Михайло Сергійович на захист дипломного проєкту (роботи)

(прізвище, ім'я, по батькові)

за спеціальністю 172 - Телекомунікації та радіотехніка

На тему: Система «Розумний будинок» з віддаленим керуванням

Дипломний проєкт (робота), рецензія і довідка про перевірку на плагиат додаються.

Декан факультету



(підпис)

(ім'я, прізвище)

Тетяна Товарищенко

ДОВІДКА УСПІШНОСТІ

Зданевич М. С. за період навчання на факультеті інформаційних технологій з 2021 по 2025 роки повністю виконав навчальний план спеціальності з таким розподілом оцінок за: національною шкалою: відмінно 0,00 %, добре 13,04 %, задовільно 86,96 %. шкалою ЄКТС: А 0,00 %, В 9,30 %, С 9,30 %, D 18,60 %, E 62,79 %.

Методист факультету

(підпис)

(ім'я, прізвище)

Тетяна Товарищенко

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЄКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент

Зданевич Михайло Сергійович

кваліфікаційний проєкт відповідно до завдання на проєктування. Під час виконання кваліфікаційного проєкту Зданевич Михайло Сергійович відіграв відповідальну роль у виконанні проєкту та зарекомендував себе кваліфікованим фахівцем.

Оцінка дипломного проєкту (роботи)

відмінно

Керівник дипломного проєкту

(підпис)

(ім'я, прізвище)

Михайло С. Д.

" 11 " 06 2025 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)

Дипломний проєкт (роботу) розглянуто. Студент Зданевич М. С. допускається до захисту цього проєкту (роботи) в екзаменаційній комісії.

Завідувач кафедри

ТМІТ

(назва)

Михайло С. Д.

(підпис, ім'я, прізвище)

" 11 " 06

2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

Дипломник: Зданевич Михайло Сергійович

Тема роботи: Система «Розумний будинок» з віддаленим керуванням

Спеціальність 172 «Телекомунікації та радіотехніка»

Обсяг кваліфікаційного проекту

Кількість листів креслень 2 Кількість сторінок записки 76

1. Короткий зміст роботи та прийнятих рішень в результаті виконаного наукового дослідження Розроблено структуру системи «Розумний будинок», яка складається з блоку керування, мережевого інтерфейсу з безпроводовим каналом зв'язку та сенсорно-актуаторною частиною для контролю та керування температурою та вологістю в кімнаті «Розумного будинку». Розроблено схему електричну принципову блоку керування та підсистеми моніторингу і контролю вологості і температури на базі мікроконтролера ESP32, цифрового сенсора DHT22 та релейно-комутаційної частини з оптронною розв'язкою для керування вентиляторами та зволожувачем повітря. Розроблено програмне забезпечення для блоку керування, модуля віддаленого доступу на базі протоколу MQTT, а також серверну частину веб-орієнтованого інтерфейсу користувача для налаштування параметрів системи та моніторингу даних сенсорів. Запропонований прототип системи «Розумний будинок» з віддаленим керуванням для Інтернету речей може бути використаний для тестування, демонстрації та подальшої розробки подібних систем.

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки та техніки і передових методів роботи: Кваліфікаційний проект виконаний на достатньо високому науково-технічному рівні з використанням сучасних досягнень у галузі інтелектуальних систем на основі технології Internet of things/Інтернет речей. У вступній частині обґрунтовано актуальність обраної теми. У теоретичному розділі детально проаналізовано існуючі конструкції та вибрано оптимальну топологію. У проектній частині запропоновано структуру системи, обґрунтовано вибір компонентів та розроблено алгоритм роботи системи. У третьому розділі розглянуті питання апаратної реалізації системи та її програмного забезпечення. Усі етапи виконані послідовно, з дотриманням технічних вимог, із застосуванням сучасних методів проектування телекомунікаційних систем віддаленого керування.

4. Позитивні сторони роботи: Робота присвячена актуальному напрямку телекомунікаційних систем — розробці системи «Розумний будинок». Автор демонструє глибоке розуміння принципів роботи мікропроцесорних систем та сучасних технологій керування сенсорами на основі протоколу Message Queuing Telemetry Transport. Робота має практичну цінність і може бути використана для подальших розробок систем для Інтернету речей.

5. Негативні сторони роботи: Робота в цілому виконана на достатньо високому рівні. Доцільно було б провести експериментальні випробування системи з використанням запропонованих компонентів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: відповідає ЕСКД

7. Відгук про роботу в цілому: В цілому кваліфікаційний проєкт виконаний якісно, має чітку структуру, логічну послідовність викладення матеріалу та містить достатній рівень технічної деталізації. Робота демонструє вміння здобувача застосовувати сучасні інженерні методи в галузі електроніки та телекомунікацій. Отримані результати є обґрунтованими та мають практичну значущість. Робота відповідає вимогам до кваліфікаційних проєктів освітнього рівня «бакалавр».

8. Інші зауваження: немає

9. Оцінка дипломної роботи: Кваліфікаційний проєкт відповідає встановленим вимогам і заслуговує оцінки «відмінно» (4.75/А), а її автор Зданевич Михайло, присвоєння кваліфікації бакалавра з телекомунікацій та радіотехніки

10. Рецензент (прізвище, ім'я, по батькові, місце роботи) Кльоц Юрій Павлович, завідувач кафедри кібербезпеки Хмельницького національного університету

« 11 » червня 2025р.



підпис