

БАКАЛАВРСЬКА РОБОТА

БР.КІ-12.00.00.000 ПЗ

Група КІ-21-2

Мужик Олександр

2025

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Інститут інформаційних технологій
Кафедра комп'ютерних систем і мереж

Мужик Олександр Мирославович
(прізвище, ім'я, по батькові)

УДК 004.415
(індекс)

БАКАЛАВРСЬКА РОБОТА

*Розробка мікроконтролерної системи підтримання
заданої температури та вологості у приміщенні на основі STM32F*
(назва роботи)

Комп'ютерна інженерія
(назва освітньої програми)

123 – комп'ютерна інженерія
(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього ступеня _____
(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник *Кропивницька Віталія Богданівна, к.т.н., доцент*
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри КСМ _____ *С. І. Мельничук*
(посада) (підпис) (дата) (ініціали та прізвище)

6. Консультанти по дипломній роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 24 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів бакалаврської роботи	Термін виконання етапів роботи	Примітка
1	Визначення теми дипломної роботи	24.02.25 – 28.02.25	Виконано
2	Збір інформації. Дослідження предметної області	01.03.25 – 31.03.25	Виконано
3	Формування першого розділу пояснювальної записки	01.04.25 – 15.04.25	Виконано
4	Формування другого розділу пояснювальної записки	16.04.25 – 30.04.25	Виконано
5	Формування третього розділу пояснювальної записки	01.05.25 – 31.05.25	Виконано
6	Оформлення роботи	01.06.25 – 10.06.25	Виконано

Студент

_____ (підпис)

Мужик О. М.
(прізвище та ініціали)

Керівник роботи

_____ (підпис)

Кропивницька В.Б.
(прізвище та ініціали)

АНОТАЦІЯ

Зі зростанням інтересу до концепції розумного житла та стрімким розвитком Інтернету речей, системи, що забезпечують стабільний рівень температури й вологості в помешканні, почали відігравати важливу роль у структурі розумних будинків, здобуваючи широку популярність. Попри те, що класичні методи регулювання температурних та вологісних показників можуть забезпечувати базові функції керування цими параметрами, вони все ще мають низку обмежень – зокрема, у зручності використання, ефективності споживання енергії та здатності до адаптивного реагування.

Системи, побудовані на базі мікроконтролерів, привертають дедалі більше уваги завдяки поєднанню високої ефективності та низького рівня енергоспоживання.

У межах цієї роботи була створена система, що підтримує стабільну температуру й вологість у домашньому середовищі, реалізована на базі мікроконтролера STM32. Він працює в парі з відповідними датчиками, що зчитують інформацію про стан повітря в реальному часі. Отримані дані виводяться на екран типу OLED, а також передаються через бездротовий модуль ESP8266, завдяки чому забезпечується передача інформації до хмарного сервера та можливість дистанційного контролю за допомогою спеціального мобільного застосунку.

Такий підхід дозволяє подолати типові недоліки традиційних систем регулювання клімату в оселі, сприяє поширенню сучасних технологій розумного житла, покращує умови побуту та рівень здоров'я мешканців, водночас знижуючи споживання енергії та втілюючи принципи екологічного способу життя.

МІКРОКОНТРОЛЕР, РОЗУМНИЙ БУДИНОК

ABSTRACT

With the growing interest in the concept of smart homes and the rapid development of the Internet of Things, systems that ensure a stable level of temperature and humidity in the home have begun to play an important role in the structure of smart homes, gaining wide popularity. Despite the fact that classical methods of regulating temperature and humidity indicators can provide basic functions of controlling these parameters, they still have a number of limitations - in particular, in ease of use, energy efficiency and the ability to adapt.

Systems built on the basis of microcontrollers are attracting more and more attention due to the combination of high efficiency and low power consumption.

As part of this work, a system was created that maintains a stable temperature and humidity in the home environment, implemented on the basis of the STM32 microcontroller. It works in conjunction with appropriate sensors that read information about the air condition in real time. The received data is displayed on an OLED screen and transmitted via the ESP8266 wireless module, which ensures the transfer of information to a cloud server and the possibility of remote control using a special mobile application.

This approach allows you to overcome the typical shortcomings of traditional home climate control systems, promotes the spread of modern smart home technologies, improves living conditions and the health of residents, while reducing energy consumption and implementing the principles of an ecological lifestyle.

MICROCONTROLLER, SMART HOUSE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	4
ВСТУП.....	5
1 ОГЛЯД ТЕХНОЛОГІЙ «РОЗУМНОГО БУДИНКУ»	7
1.1 Аналіз теми досліджень.....	7
1.2 Огляд існуючих рішень	9
1.3 Мікроконтролери та сенсори у системах "розумного будинку"	12
1.4 Постановка завдання.....	19
2 РОЗРОБЛЕННЯ СТРУКТУРИ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ	20
2.1 Структурна схеми системи.....	20
2.2 Проектування та реалізація структури схеми	22
2.3 Налаштування програмного забезпечення	29
2.4 UML-діаграми системи.....	35
2.5 Висновки до розділу	37
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	38
3.1 Налаштування апаратного та програмного забезпечення	38
3.2 Висновки до розділу	45
ВИСНОВКИ.....	47
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	48

					БР.КІ-12.00.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розроб.		<i>Мужик О. М.</i>			Літ.	Арк.	Аркушів
Перевір.		<i>Кропивницька</i>			3	50	
Реценз.		<i>Заячук Я.І.</i>			ІФНТУНГ, КІ-21-2		
Н. Контр.		<i>Лазорів А.М.</i>					
Затверд.		<i>Мельничук С.І.</i>					
					Розробка мікроконтролерної системи підтримання заданої температури та вологості у приміщенні на основі STM32F		

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

IDE – Integrated Development Environment (вбудоване середовище розробки)

IoT – Internet of Things

IP – Internet Protocol

GPIO – General-Purpose Input/Output (інтерфейс зв'язку компонентів)

TCP – Transmission Control Protocol

SH – Smart House (розумний дім)

Wi-Fi – Wireless Fidelity (бездротова передача)

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

ВСТУП

Для підтримання температури та вологості в будинку в межах розумного діапазону основним способом досягнення автоматизації є впровадження раціональних заходів. Необхідно розробити науково обґрунтовану та практичну систему моніторингу та методи інтелектуального керування, щоб забезпечити ефективну і раціональну роботу кожного компонента. Функціональність може бути виконана автоматично пристроєм або вручну увімкнена користувачем для підтримання температури та вологості в допустимих межах.

Актуальність теми дослідження. На сьогодні на ринку відчувається нестача продуктів, які можуть одночасно контролювати як температуру, так і вологість, хоча саме ці показники є важливими для оцінки комфорту людини. Ефективним способом розв'язання цієї проблеми є проектування та створення системи контролю температури та вологості.

Метою дипломного проектування є мікроконтролерної системи контролю та регулювання температури і вологості у приміщенні з використанням технології безпроводних мереж.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- створення структурної схеми;
- створення схеми електричної принципової;
- створення алгоритмічного та програмного забезпечення.

Об'єктом дослідження є система підтримання постійної температури та вологості в умовах розумного будинку.

Предметом дослідження є процес вимірювання температури й вологості в приміщенні та процес ухвалення рішень системою.

Методи дослідження. У роботі використовуються теоретичні методи та експериментальні дослідження для аналізу досягнення порогових значень шляхом вимірювання температури та вологості середовища. Основою дослідження є метод визначення параметрів та розробка систем моніторингу та керування.

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

Практичне значення одержаних результатів полягає у реалізації системи контролю температури та вологості в домашніх умовах, яка дає змогу забезпечити підтримання комфортного мікроклімату шляхом налаштування та регулювання цих параметрів. Така система придатна для реального застосування та має потенціал до подальшого розширення.

З огляду на загальні тенденції моніторингу та автоматизації було розроблено доволі просту модель бездротового датчика температури та вологості. Завдяки програмуванню та налаштуванню реалізується система моніторингу змін цих параметрів. Результатом є реальна модель пристрою та системи, яка може бути використана на практиці та має потенціал для подальшого вдосконалення..

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		6

1 ОГЛЯД ТЕХНОЛОГІЙ «РОЗУМНОГО БУДИНКУ»

1.1 Аналіз теми досліджень

З поширенням концепції розумного будинку та стрімким розвитком технологій Інтернету речей, розробка та дослідження системи підтримки постійної температури та вологості в будинку, як важливої складової розумного дому, привертає дедалі більше уваги. Хоча традиційні системи підтримки температури та вологості певною мірою задовольняють потреби людей у контролі температури, вони все ж мають недоліки щодо інтелектуальності, енергоефективності та зручності у використанні. Саме тому розробка домашньої системи підтримки постійної температури та вологості на основі однокристального мікроконтролера набула актуальності й стала гарячою темою сучасних досліджень [1-2,27].

У повсякденному житті серед поширених пристроїв для контролю температури та вологості є кондиціонери, водяне опалення, тепла підлога, обігрівачі, зволожувачі та інші прилади. Але кожен із них виконує лише одну функцію. Їхня функціональність є неповною. Вони не забезпечують хорошого користувацького досвіду. Наприклад, коли ми влітку вмикаємо кондиціонер, часто відчуваємо сухість. А взимку, коли вмикаємо водяне опалення, деякі люди можуть відчувати задуху через неправильний рівень вологості. На ринку є багато продуктів, які можуть змінювати температуру, а також багато пристроїв, які змінюють вологість. Проте бракує пристрою, який міг би ефективно поєднати регулювання температури й вологості та керувати ними одночасно за допомогою одного контролера. Автоматичним способом підтримувати температуру й вологість у приміщенні в комфортному діапазоні. Тому в роботі буде досліджено пристрій, який може одночасно контролювати температуру та вологість.

Як високопродуктивний мікроконтролер із низьким енергоспоживанням, мікроконтролер STM32 має потужні обчислювальні можливості та багатий набір

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		7

периферійних інтерфейсів, що відкриває можливості для інтелектуального керування системами підтримки постійної температури та вологості в будинку. Така система здатна в режимі реального часу збирати дані про стан внутрішнього середовища, здійснювати контроль температури та забезпечувати підтримку постійного мікроклімату в приміщенні. Окрім цього, система дозволяє користувачам здійснювати персоналізовані налаштування та керування через інтерфейс людина-машина, покращуючи тим самим користувацький досвід. Вона усуває недоліки традиційних систем постійної температури та вологості в аспектах інтелектуальності та енергоефективності, а також сприяє розвитку і впровадженню технологій "розумного дому".

Завдяки системі постійної температури та вологості, що базується на мікроконтролері STM32, люди можуть зручніше керувати внутрішнім середовищем та покращувати комфорт і здоров'я життя. Водночас інтелектуальні та енергоощадні властивості системи сприяють зниженню споживання енергії в побуті та реалізації концепції екологічного способу життя.

Значущість цього дослідження полягає в інноваційній взаємодії різного апаратного забезпечення та усуненні недоліків контролю температури і вологості в домашньому середовищі. Високий рівень інтеграції системи відповідає вимогам для виконання повсякденних функцій. Поряд із підвищенням інтеграції, система стає зручнішою для спостереження та керування. Інтелектуальне керування з інтерфейсом людина-машина відповідає потребам нового часу та закладає основу для розвитку Інтернету речей у домашньому середовищі. Вдосконалюючи досвід користувача, система також підвищує комфорт, заощаджує енергію, сприяє захисту довкілля та покращує якість і здоров'я життя користувачів.

Розвинені країни, такі як Європа та Сполучені Штати, почали дослідження в галузі розумного дому раніше і вже сформували відносно повну технічну систему та ринкову структуру. Зарубіжні дослідники зосереджуються на інноваціях та проривних досягненнях, постійно впроваджуючи нові алгоритми та технічні засоби для досягнення контролю за внутрішнім середовищем. Для більш точного та інтелектуального управління деякі відомі компанії з розумного дому випустили

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

різноманітні продукти систем терморегулювання на основі мікроконтролерів і здобули хорошу репутацію на ринку. Ці продукти не тільки мають високий рівень інтелекту та персоналізації, але й орієнтовані на досвід користувача, енергозбереження та охорону навколишнього середовища, а також відповідають потребам сучасних сімей [6,27].

В останні роки, з постійним розширенням ринку розумних будинків в країні та поступовим удосконаленням технологій, системи терморегулювання будинку отримали широке визнання в Україні. Багато науково-дослідних установ та університетів почали вкладати кошти в дослідження цієї галузі й досягли низки важливих результатів. Вітчизняні дослідники досягли точного контролю за внутрішнім середовищем завдяки оптимізації алгоритмів і вдосконаленню апаратних засобів. Завдяки високій продуктивності та низькому споживанню енергії, мікроконтролери стали переважним рішенням для проектування систем терморегулювання будинку. Одночасно вітчизняні дослідники активно досліджують керування з іншими пристроями розумного дому для досягнення більш інтелектуального та зручного життєвого середовища [8,27]. Деякі компанії розумного дому вже почали випускати продукти систем терморегулювання і поступово займають свою частку на ринку. Ці продукти не тільки мають високу продуктивність контролю температури, але відзначаються високою ефективністю та простотою в користуванні, що робить їх популярними серед споживачів.

1.2 Огляд існуючих рішень

1.2.1 Побутовий кондиціонер

Одним із найпоширеніших рішень для контролю температури є кондиціонер, який має перевагу в зручності та швидкості. Кондиціонер проходить через компресор, який конденсує повітря. Це дозволяє швидко знижувати температуру в кімнаті влітку. Взимку можна використовувати принцип нагрівання за допомогою нагрівального дроту для обігріву повітря та викиду

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		9

гарячого повітря для підвищення температури в приміщенні. Але водночас є й очевидні недоліки, зокрема неможливість ефективно контролювати вологість повітря в кімнаті. Кожного разу, коли ви використовуєте кондиціонер, він забирає вологу з приміщення. Це знижує вологість повітря та створює відчуття сухості. Хоча отримуємо швидке охолодження, це відбувається за рахунок втрати вологи.



Рисунок 1.1 – Побутовий кондиціонер

Водночас система контролю кондиціонера складається з мікроконтролера, компресора, нагрівального дроту, вентилятора тощо. Мікроконтролер керує роботою компресора та вентилятора, щоб викидати холодне повітря. За допомогою керування нагрівальним дротом і феном викидається гаряче повітря. Однак ця система не може одночасно контролювати температуру та вологість повітря. Хоча є можливість контролювати температуру, немає способу контролювати вологість. Домашній кондиціонер показано на рисунку 1.1.

1.2.2 Побутовий повітряний обігрівач

Як одне з поширених рішень проблем з тепловою енергією взимку, обігрівачі можуть досягти робочого стану за дуже короткий час і постійно подавати тепле повітря. Переваги полягають у тому, що вони зручні, швидкі та легкі для переміщення. Їх можна використовувати на столах або підлогах для подачі теплого повітря туди, де це необхідно. Недоліки також очевидні. Для великих приміщень ефект обігріву всього простору дуже незначний, а тривале впливання тепла на шкіру людей може спричинити її місцеву сухість і навіть викликати захворювання. Принцип його роботи також полягає в обігріві повітря

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

за рахунок втрати вологи. Обігриваючи, він також робить кімнату ще більш сухою взимку. Побутовий обігривач повітря зображений на рисунку 1.2.



Рисунок 1.2 – Електричний обігривач

Система цього пристрою є простішою, і контролер лише керує потужністю вентилятора та нагрівального дроту, але, хоча вона й проста, вона також втрачає функціональність. Він все ще не може контролювати вологість.

1.2.3 Підігрів підлоги

Як одна з поширених систем опалення підлогове опалення (рис. 1.3) значно покращує комфорт у приміщенні. Воно рівномірно розподіляє тепло по всій площі підлоги, підтримуючи в кімнаті стабільний температурний режим. Цей метод обігріву знизу вгору може зменшити дискомфорт людей у приміщенні.



Рисунок 1.3 – "Тепла" підлога

Однак його недоліки також очевидні: велике споживання енергії, повільне нагрівання та необхідність ручного регулювання.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		11

1.2.4 Зволожувачі повітря

Зволожувач повітря – це пристрій, який часто використовується протягом року. Його перевага полягає в тому, що він може швидко та зручно поповнити вологість у повітрі. Він використовує принцип атомізації для розпилення чистої води на маленькі краплі, тим самим збільшуючи вологість повітря. Його недолік також очевидний – він потребує ручного контролю. І через конструкційні особливості він не може точно розрахувати середню вологість у приміщенні. Кожного разу, коли ми думаємо про його використання, часто виявляється, що повітря вже надмірно сухе.



Рисунок 1.4 – Зволожувач повітря

Система зволожувача повітря складається з простого атомайзера, який вмикається чи вимикається через контрольний вимикач. При цьому вона не здатна вимірювати вологість або змінювати робочий режим залежно від вологості. Все це здійснюється тільки за допомогою людського контролю. Зволожувач повітря показаний на рисунку 1.4.

1.3 Мікроконтролери та сенсори у системах "розумного будинку"

Мікроконтролер STC89C52RC (покращений варіант I8051) має переваги, такі як низька робоча напруга, низьке енергоспоживання, висока частота роботи, низька ціна тощо. Він може широко використовуватися в різноманітному низькопотужному, високоточному, мініатюрному електронному обладнанні [27].

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		12

Через свій більший розмір, нижчу швидкість обробки та меншу кількість виводів, мікроконтролер 8051 поступово витісняється більш сучасними мікропроцесорами. Мікроконтролер 8051 зображено на рисунку 1.5.

Мікроконтролер STM32 має характеристики потужної функціональності, численних моделей і широкого використання. Він інтегрує багато функцій, таких як перетворення аналогового сигналу, інтерфейс SPI, послідовна комунікація тощо.



Рисунок 1.5 – Мікроконтролер STC89C52RC (I8051)

Завдяки високій тактовій частоті має високу продуктивність обробки та низьке енергоспоживання, що робить його придатним для потреб проектування бездротових систем [10,27]. Мікроконтролер STM32 зображено на рисунку 1.6.

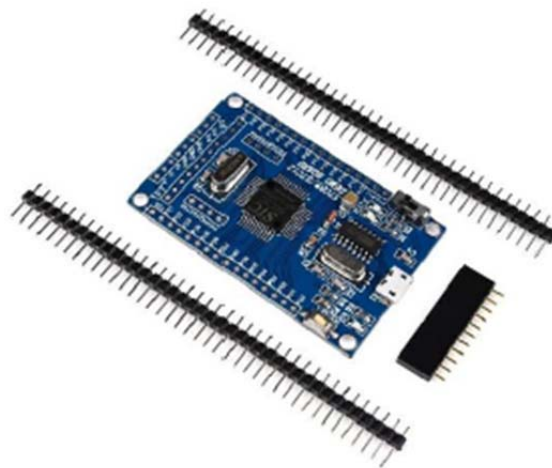


Рисунок 1.6 – Мікроконтролер STM32

Мікроконтролер Arduino використовує мікроконтролер серії Atmel AVR, заснований на RISC-процесорі архітектури Harvard, і має такі характеристики, як низьке енергоспоживання, низька вартість та простота програмування. Він використовує стандартизовані виводи та інтерфейси (наприклад, GPIO), а також різноманітні плати розширення (Shields), що робить його масштабованішим для задоволення потреб застосування. Зазвичай використовується в малих вбудованих системах. Мікроконтролер Arduino зображено на рисунку 1.7.

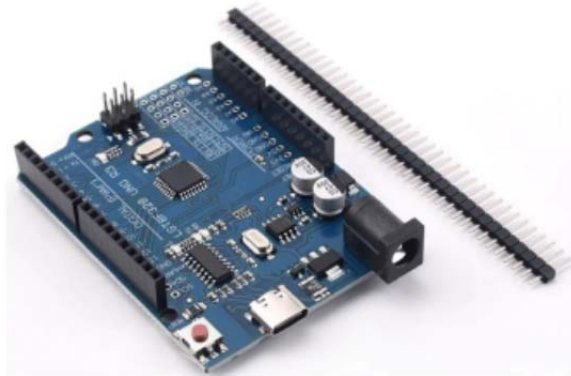


Рисунок 1.7 – Мікроконтролер Arduino

Плата розробки Raspberry Pi заснована на ARM-процесорі й має вищу обчислювальну продуктивність і більший обсяг пам'яті в порівнянні з подібними продуктами. Вона може запускати складні додатки та виконувати багатозадачність, а також має можливості підключення до мережі й забезпечує графічний інтерфейс. Але по суті, це потужний одноплатний комп'ютер з повноцінною обчислювальною потужністю та операційною системою. Плату Raspberry Pi зображено на рисунку 1.8.



Рисунок 1.8 – Плата розробки Raspberry Pi

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

Хоча I/O-виводи мікроконтролера 51 прості у використанні, вони не мають вихідної здатності при високому рівні. При обробці подвійних вказівників даних він працює занадто повільно. Крім того, захисні властивості мікроконтролера 51 дуже слабкі, і чип легко виводиться з ладу.

Незважаючи на те, що мікроконтролер 51 є одним із найпоширеніших 8-бітних мікроконтролерів, з огляду на майбутню масштабованість, 32-бітна архітектура STM32 більше відповідає сучасним тенденціям розвитку.

Arduino зазвичай використовує мікроконтролер з низьким енергоспоживанням, але з обмеженою обчислювальною потужністю, підходить для обробки простих керуючих задач і даних від сенсорів. Він має нижчу тактову частоту та менший обсяг пам'яті, тому не може обробляти складні обчислювальні задачі. З точки зору масштабованості, STM32 має більш широкий спектр підтримуваних пристроїв і гнучкіші методи програмування, що краще підходить для реалізації даної теми.

Raspberry Pi – це повноцінний одноплатний комп'ютер з потужною продуктивністю й багатим інтерфейсом. Він підтримує велику кількість модулів і сенсорів, розширюється через USB, GPIO тощо. Він може працювати під управлінням операційної системи Linux і запускати різне прикладне програмне забезпечення. Але за його високою продуктивністю стоїть висока вартість, а його можливості перевищують вимоги даної теми. З економічної точки зору – не найкращий варіант.

Для реалізації системи підтримки сталої температури й вологості в домі обрано мікроконтролер STM32, що має високу інтеграцію та малий розмір. Він задовольняє вимоги проекту та відповідає тенденціям майбутнього розвитку.

HC-05 – це класичний Bluetooth-модуль, що використовує стандартний протокол Bluetooth V2.0, працює в діапазоні частот 2.4 ГГц, підтримує послідовну комунікацію та режим прозорості передачі даних, і має відносно низьку ціну. Максимальна швидкість передачі даних становить 2.1 Мбіт/с. Він використовує послідовну комунікацію і може застосовуватись для бездротового з'єднання пристроїв. HC-05 зображений на рисунку 1.9.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		15

Бездротовий модуль ESP8266 – це недорогий бездротовий пристрій. Він інтегрує функцію Wi-Fi і може широко застосовуватись у розумних будинках, продуктах Інтернету речей та інших галузях.



Рисунок 1.9 – Bluetooth модуль HC-05

Має вбудований процесор і може виконувати власні програми, що робить його придатним для розробки IoT-пристроїв та вбудованих систем. Користувачі можуть реалізовувати функції збору даних із сенсорів, дистанційного керування та передачі даних через програмування [27]. ESP8266 зображено на рисунку 1.10.



Рисунок 1.10 – WiFi модуль ESP8266

Ця розробка призначена для моніторингу температури та вологості в приміщеннях. Wi-Fi модуль може бути підключений до роутера для організації мережевого підключення. Bluetooth є пристроєм ближнього зв'язку з точковим з'єднанням, і для віддаленого керування потребує додаткового обладнання. Тому для проекту було обрано Wi-Fi модуль.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		16

Семисегментний індикатор – це простий пристрій для відображення символів, який показує цифри шляхом підсвічування різних сегментів (рис. 1.11). Часто використовується для таймерів, лічильників та інших випадків, коли потрібне цифрове відображення. Має перевагу високої надійності.

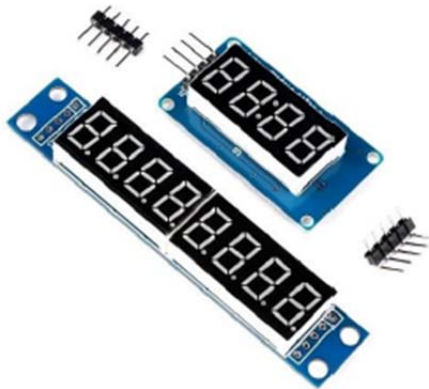


Рисунок 1.11 – Семисегментний індикатор

OLED-дисплей невеликий за розміром, має велику кількість пікселів і досить насичене зображення (рис. 1.12). Має власне підсвічування, а зображення чітке та яскраве. Може використовуватися в промислових і медичних сферах.



Рисунок 1.12 – OLED-дисплей

Цей проєкт потребує відображення інформації, такої як температура, вологість навколишнього середовища та встановлений час. Оскільки зміст виводу є досить різноманітним, для цього проєкту обрано OLED-дисплей.

Датчик температурного опору PT100 має широкий діапазон вимірювання температури, швидку реакцію, малий розмір, високу точність, а також захищену конструкцію (рис. 1.13).

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		17



Рисунок 1.13 – Сенсор РТ100

Модуль датчика DHT11 може одночасно збирати дані про температуру та вологість і здійснювати передавання даних та управління за допомогою одного сигнального проводу. Простий у використанні, широко застосовується у сфері моніторингу температури та вологості. DHT11 зображено на рисунку 1.14.

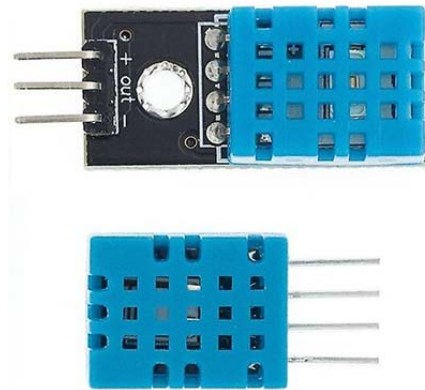


Рисунок 1.14 – Сенсор DHT11

Його компактний корпус дозволяє легко встановлювати модуль у різноманітні пристрої. Він стабільний і надійний, забезпечує точну інформацію в режимі реального часу про температуру та вологість, що є корисною для користувача в контексті екологічного моніторингу [27].

У цьому проєкті використовується датчик DHT11 для збору інформації про температуру та вологість у приміщенні.

1.4 Постановка завдання

Було виконано аналіз різних способів контролю температури та вологості, які є на ринку, і кожен з них має свої переваги. Однак на ринку відсутній комплексний продукт, який міг би одночасно автоматично та розумно контролювати й регулювати температуру та вологість у приміщенні. Зі змінами часу ми вступили в епоху Інтернету речей. Можливо, нам слід адаптуватися до розвитку часу. У майбутньому, з розвитком розумного дому та Інтернету всього, це стане сферою з величезним ринком. Нам слід адаптуватися до розвитку ринку та зробити продукти більш конкурентоспроможними.

Після аналізу переваг і недоліків кожного з методів прийняте рішення поєднати їхні переваги та розробити мікроконтролерну систему з можливістю підключення до мережі, яка буде контролювати температуру та вологість у кімнаті та постійно підтримувати діапазон, що підходить для людського організму.

Таким чином, метою даної роботи є розроблення системи з можливістю підключення до мережі, яка регулюватиме температуру та вологість у кімнаті, а також зможе завантажувати дані в хмару. Користувачі можуть керувати системою через мікроконтролер або дистанційного завдяки мобільному терміналу.

Для досягнення вказаної мети потрібно вирішені такі завдання:

- створення структурної схеми;
- створення схеми електричної принципової;
- створення алгоритмічного та програмного забезпечення.

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		19

2 РОЗРОБЛЕННЯ СТРУКТУРИ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ

2.1 Структурна схеми системи

В основі мікроконтролерної системи використовує однокристальний мікрокомп'ютер як основний пристрій керування, щоб точно отримувати дані про температуру та вологість домашнього середовища за допомогою зовнішніх датчиків і відображати їх у режимі реального часу на РК-дисплеї, що робить зручним для користувачів перевірку температури та вологості поточного середовища у будь-який час.

Система також оснащена простою функцією керування за допомогою кнопок, що дозволяє користувачам легко встановлювати порогові значення температури та вологості. Коли температура та вологість навколишнього середовища виходять за межі встановлених значень, система автоматично активує механізм регулювання, щоб підтримувати температуру та вологість у приміщенні в межах бажаного для користувача комфортного діапазону.

Показники температури та вологості в режимі реального часу можуть передаватися на хмарну платформу через бездротовий зв'язок, а параметри налаштування системи можна легко змінювати за допомогою мобільного телефону.

Схема регулювання температури. Коли користувач встановлює поріг температури через інтерфейс або хмарну платформу, система отримує команду для встановлення цього значення. Якщо датчик температури виявляє, що температура нижча за встановлене значення, серводвигун відкриває водяний клапан, а датчик потоку контролює швидкість потоку рідини в системі опалення для підвищення температури в приміщенні. Якщо датчик фіксує, що температура вища за встановлене значення, вмикається кондиціонер для зниження температури в приміщенні [27].

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		20

Схема регулювання вологості. Коли користувач встановлює поріг вологості через інтерфейс або хмарну платформу, система отримує команду для встановлення цього значення. Якщо датчик вологості виявляє, що вологість нижча за встановлене значення, вмикається зволожувач повітря для підвищення вологості в приміщенні. Якщо вологість перевищує встановлене значення, вмикається вентилятор, який виводить повітря з приміщення для зниження вологості.

Структура системи представлена на рисунку 2.1.



Рисунок 2.1 – Структурна схема системи

Компоненти системи:

- Microcontroller (мікроконтролер): збирає та аналізує дані, керує роботою системи;
- Temperature and humidity sensor (датчик температури та вологості): збирає дані про температуру та вологість у приміщенні;
- LCD (рідкокристалічний дисплей): відображає температуру, вологість, час тощо;

- Wireless module (бездротовий модуль): передає дані про температуру та вологість на хмарну платформу для віддаленого моніторингу в реальному часі;
- Fan (вентилятор/кондиціонер): видаляє вологе повітря з приміщення та знижує вологість;
- Flow rate sensor (датчик потоку): визначає швидкість потоку рідини в системі опалення;
- Steering gear (серводвигун): керує потоком рідини в обігрівачі для підвищення температури в приміщенні;
- Indicator light (індикаторна лампа): вказує на робочий стан сервомеханізмів, реле тощо;
- Button (кнопка): використовується для встановлення часу, температурних та вологісних порогів через панель керування мікроконтролера;
- Relay (реле): керує живленням зовнішніх пристроїв, таких як кондиціонери.

2.2 Проектування та реалізація структури схеми

У цій розробці використовується мікроконтролер STM32F103C8T6. Мікроконтролер побудований на базі ядра ARM і використовує Гарвардську архітектуру, що дозволяє працювати з підвищеною тактовою частотою. Чіп має 49 універсальних портів введення/виведення та низку багатих периферійних ресурсів, включаючи модулі AD/DA, таймери, інтерфейси зв'язку, вихід PWM тощо, що дозволяє задовольнити потреби широкого спектра застосувань. Тактова частота системи мікроконтролера STM32 може досягати 72 МГц за відповідної конфігурації. Він має вбудовану функцію годинника реального часу (RTC), зовнішній кварцовий резонатор на 32,768 кГц і може живитися від батарейки CR1220 [12,27]. Ядро мікроконтролера показано на рисунку 2.2.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		22

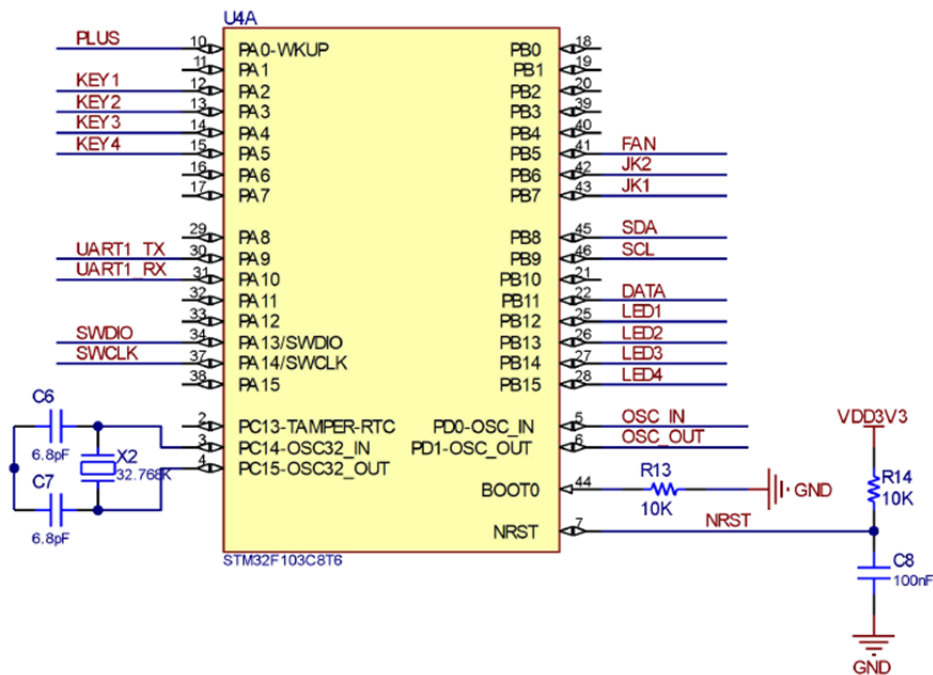


Рисунок 2.2 – Схема виводів плати мікроконтролера

У розробці використовується монохромний LCD-дисплей із роздільною здатністю 128×64 пікселів. Дисплей виготовлений за технологією органічних світлодіодів (OLED) і має такі характеристики, як високий контраст, швидкий відгук, широкий кут огляду та низьке енергоспоживання. Для послідовного зв'язку використовується інтерфейс I2C або SPI, що дозволяє легко під'єднувати його до різних пристроїв керування [14,27]. У цій розробці застосовано з'єднання за методом ІС, що характеризується меншою кількістю контактів і високою швидкістю оновлення. OLED-екран живиться від зовнішнього джерела 5В. Схема апаратного інтерфейсу показана на рисунку 2.3.

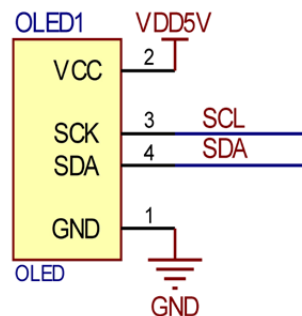


Рисунок 2.3 – Схема інтерфейсу РК-екрана

Модуль датчика DHT11 має два вбудовані чутливі елементи, які дозволяють одночасно вимірювати температуру і вологість. Він використовує однопровідний інтерфейс зв'язку, який дозволяє як передавати, так і приймати дані. Можна отримати цілі та десяткові значення температури і вологості. Мінімальна робоча напруга може досягати 3В [16,27]. Схема апаратного інтерфейсу показана на рисунку 2.4.

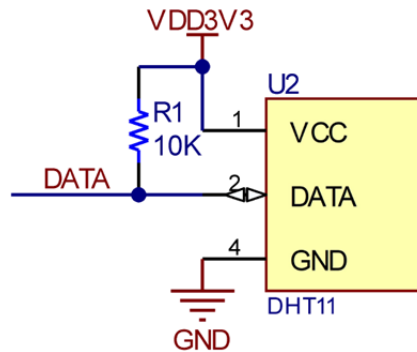


Рисунок 2.4 – Схема підключення контактів датчика температури та вологості DHT11

Бездротовий модуль ESP8266 дозволяє передавати дані та здійснювати зв'язок з іншими пристроями через бездротові з'єднання. Його принцип роботи полягає в інтеграції бездротових модулів і процесорів у складі апаратного забезпечення, наявності прошивки, яка реалізує протокол бездротового зв'язку, зв'язку з іншими пристроями через бездротові з'єднання та передачі даних за допомогою стеку протоколів TCP/IP [18,20,27]. Схема апаратного з'єднання між мікроконтролером і бездротовим модулем показана на рисунку 2.5. Модуль ESP8266 здатний функціонувати як у режимі точки доступу, так і в режимі клієнта мережі. Під час стандартної роботи в локальній Wi-Fi мережі він зазвичай налаштовується як клієнт. Для цього потрібно ввести назву мережі (SSID) і, якщо мережа захищена, – пароль. На початковому етапі налаштування зручно використовувати режим точки доступу. У цьому режимі ESP8266 з'являється в переліку доступних мереж на комп'ютерах чи мобільних пристроях. Після підключення до нього можна відкрити веб-інтерфейс і ввести необхідні мережеві параметри.

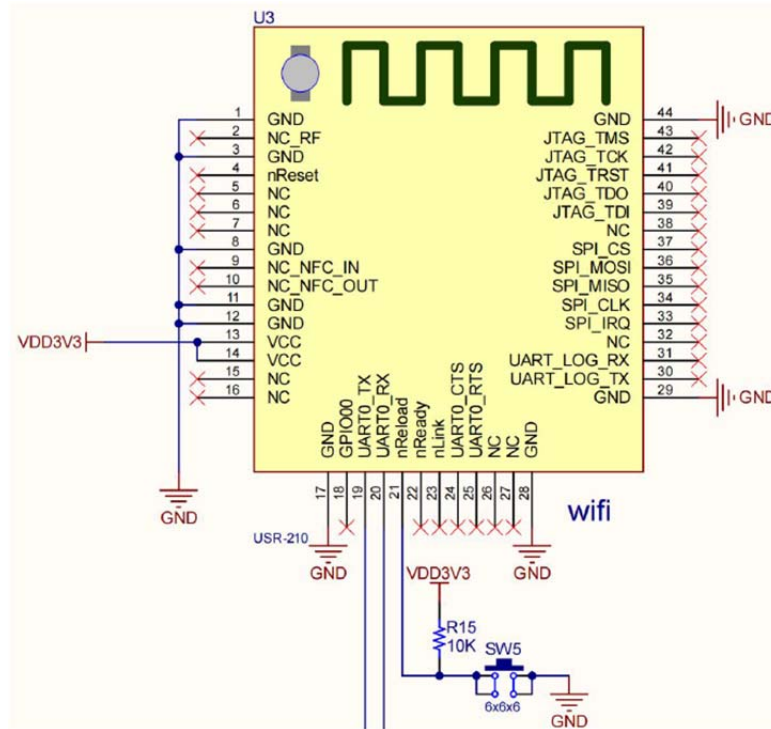


Рисунок 2.5 – Виводи ESP8266

У проєкті використано датчик витрати S201, який містить всередині елемент Холла. Його розроблено на основі принципу перетворення магнітного поля та електричного струму. Коли змінюється зовнішнє магнітне поле – змінюється напруга. Внутрішня схема підсилення містить підсилювач для посилення змінного сигналу напруги. Також всередині розміщено компаратор, який формує імпульсний вихідний сигнал. Схема підключення датчика витрати зображена на рисунку 2.6.

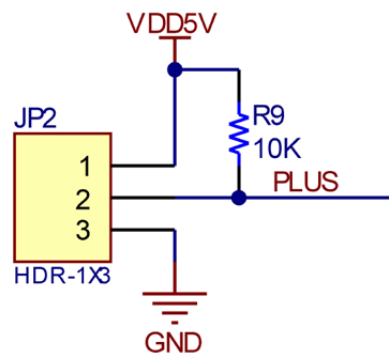


Рисунок 2.6 – Принципова схема датчика швидкості потоку S201

Збільшення вологості та охолодження внутрішнього середовища контролюються двома реле, які працюють як вимикачі. Реле працює за

принципом магнітної індукції. Коли рівень на базі транзистора знижується, транзистор відкривається, і котушка створює магнітне поле, яке притягує пластину. Це замикає контакти реле. У цей момент зволожувач або кондиціонер отримують живлення та починають працювати. Схема апаратної частини зображена на рисунку 2.7.

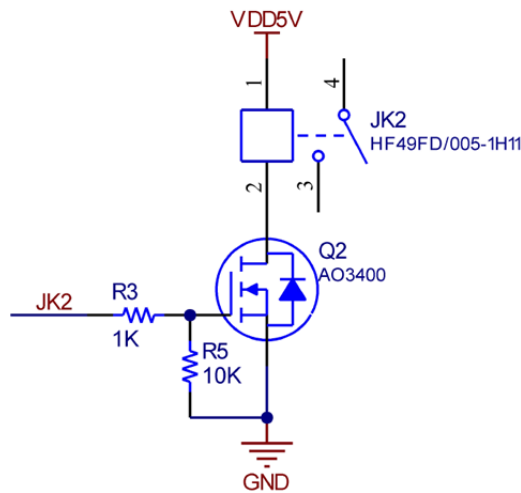


Рисунок 2.7 – Принципова схема керування реле

У запропонованій системі вентилятор виконує функцію осушувача. Коли досягається заданий поріг вологості, вмикається витяжний вентилятор. Він живиться від 5 В і керується транзистором, що працює як перемикач. Коли керуючий сигнал має високий рівень, транзистор відкривається, і вентилятор починає працювати. Схема апаратної частини зображена на рисунку 2.8.

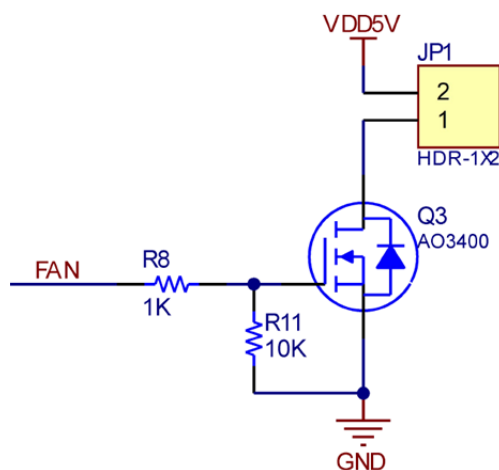


Рисунок 2.8 – Принципова схема керування вентилятором

Схема кнопок передбачає використання чотирьох окремих кнопок. Порт зчитування кнопок підключено до виводів введення/виведення мікроконтролера. Мікроконтролер зчитує стан низького рівня. У цьому дизайні кнопки використовуються для додавання, віднімання, встановлення порогових значень і налаштування годинника. Це зображено на рисунку 2.9.

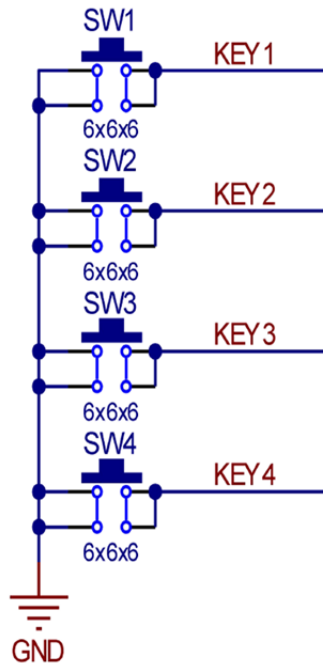


Рисунок 2.9 – Виконавчий пристрій з використанням реле

Сервопривід SG90 має лише три проводи: лінію живлення, заземлення та лінію керуючого сигналу. Керуючий сигнал є імпульсним сигналом із частотою 50 Гц. Кут обертання вала можна регулювати, змінюючи ширину імпульсу [22,27]. Апаратний інтерфейс зображено на рисунку 2.10.

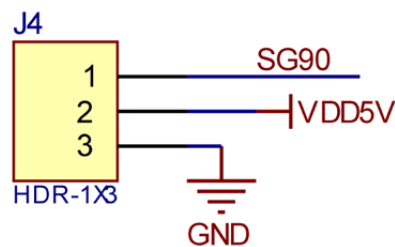


Рисунок 2.10 – Інтерфейс сервоприводу SG90

Проектування апаратної схеми є невід’ємною частиною реалізації всього проекту. Було чітко визначено умови використання обраних компонентів, щоб уникнути можливих функціональних відхилень на пізніших етапах.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		27

Для побудови принципових схем кожного компонента було використано програмне забезпечення Altium Designer, що дозволило забезпечити правильне й логічне з'єднання всіх елементів. Також були враховані питання цілісності сигналу, захисту від завад, відведення тепла та інші аспекти. Своєчасне усунення таких проблем значно скорочує витрати часу та коштів на подальше налагодження. Загальну схему підключення подано на рисунку 2.11.

2.3 Налаштування програмного забезпечення

Програмна частина цього проекту для мікроконтролера реалізується за допомогою середовища компіляції KEIL, а програмне забезпечення налаштовується відповідно до функцій модуля температури та вологості DHT11, модуля WIFI та інших периферійних пристроїв. Після увімкнення живлення мікроконтролера основна програма починає виконання: спочатку ініціалізується системний тактовий сигнал та порти периферії. За допомогою мікроконтролера здійснюється керування датчиком температури та вологості, а також датчиком витрати, зчитуються поточні дані та виводяться в реальному часі на OLED-дисплей. Одночасно отримані дані надсилаються до модуля WIFI і передаються у хмару. Користувачі можуть отримувати ці дані через мобільний додаток. Схема основного алгоритму програми показана на рисунку 2.12.

На основі призначення виводів OLED LCD-дисплея та технічної документації, було написано відповідний програмний код (Додаток А). Під час написання коду переважно використовуються дві допоміжні функції – функція запису команди та функція запису даних. Процес виводу інформації на LCD-дисплей передбачає попередню ініціалізацію драйвера-контролера SSD1306, встановлення його робочих параметрів та режиму відображення. Далі активується функція дисплею та готується прийом та обробка зображення. Після очищення відеопам'яті, необхідні дані записуються до неї та надсилаються до контролера. Контролер виводить зображення на екран LCD на основі отриманих даних. Процес керування OLED LCD-дисплеєм зображено на рисунку 2.13.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

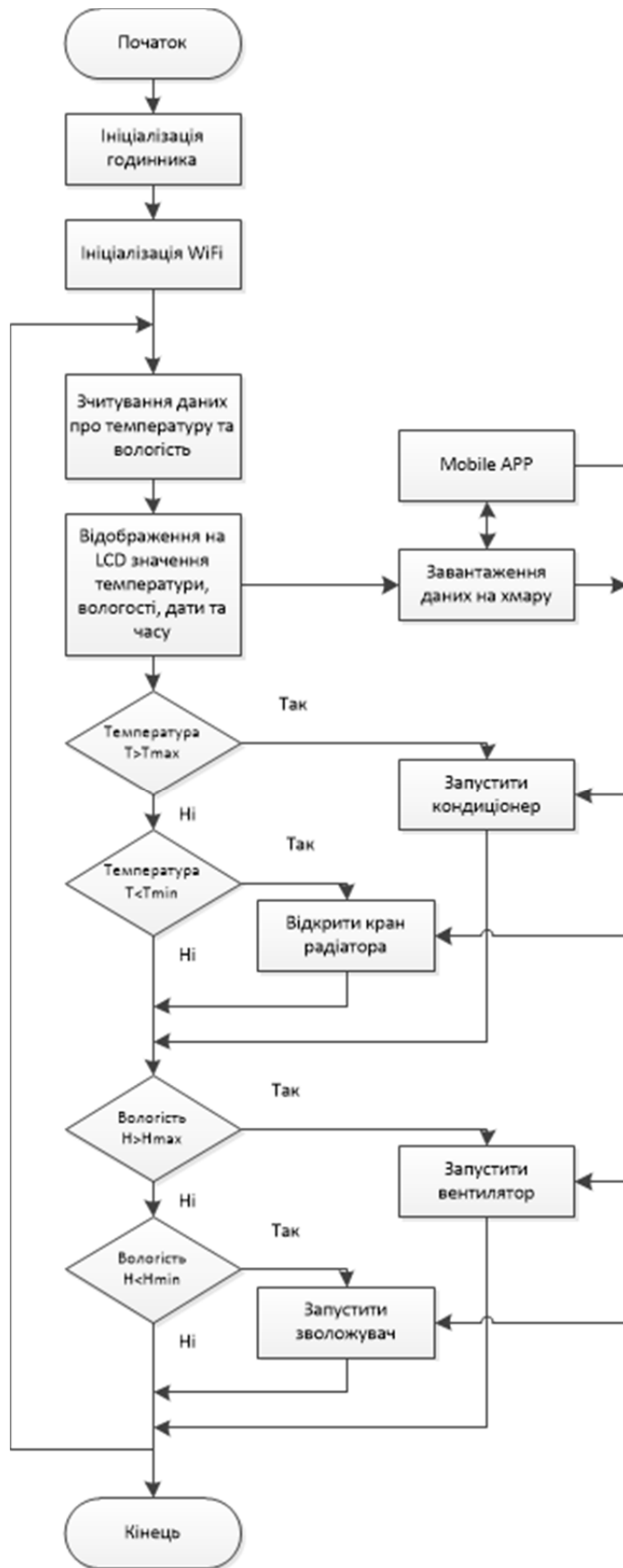


Рисунок 2.12 – Блок-схема алгоритму основної програми

Зм.	Арк.	№ докум.	Підп.	Дата



Рисунок 2.13 – Блок-схема алгоритму керування OLED LCD

У даній розробці модуль ESP8266 керується через послідовний порт мікроконтролера. Згідно з інструкцією до модуля, за допомогою AT-команд налаштовується серверний порт для передавання даних на сервер. Користувач може переглядати дані в реальному часі через мобільний додаток.



Рисунок 2.14 – Блок-схема алгоритму роботи модуля ESP8266

Дані, які надсилає мікроконтролер, мають певний формат згідно з протоколом. Модуль ESP8266 передає дані на хмарну платформу у режимі прозорої передачі, а мобільний додаток приймає та відображає цю інформацію. Блок-схема процесу зображена на рисунку 2.14.

Датчик температури та вологості DHT11 має конструкцію з одним сигнальним проводом. Під час зчитування даних з DHT11 можливо програмно отримати повний пакет даних від сенсора та одночасно обробити їх для досягнення більш точного результату вимірювання. Блок-схема процесу зчитування наведена на рисунку 2.15.

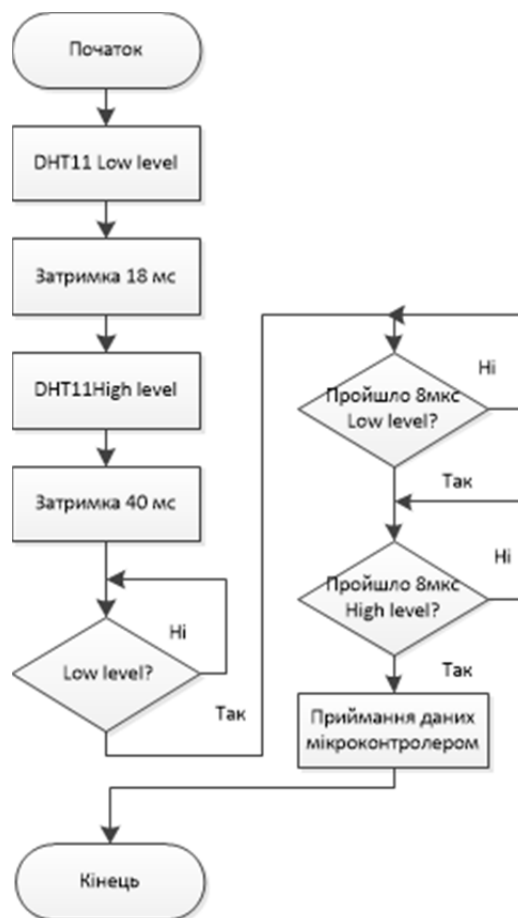


Рисунок 2.15 – Блок-схема алгоритму зчитування даних з модуля DHT11

Оскільки в цій системі потрібно керувати відкриттям і закриттям водопровідної труби з рідиною для підігріву, вхідний імпульс має період ширини 20 мс, що може бути реалізовано за допомогою налаштування ШІМ (PWM). Необхідно лише налаштувати два стани перемикача. Алгоритм роботи програми показано на рисунку 2.16.



Рисунок 2.16 – Блок-схема алгоритму керування сервоприводом

Модуль кнопок використовує чотири фізичні кнопки. Модуль реагує на спадний фронт сигналу. Коли кнопка натискається і мікроконтролер фіксує низький рівень на вході, починається обробка натискання.

Процес обробки передбачає стабілізацію сигналу. Для уникнення помилкових спрацьовувань у програмі використовується затримка для вибірки сигналу. Ця система має чотири кнопки: кнопки збільшення та зменшення, кнопки встановлення порогу температури та вологості, а також кнопка налаштування часу. Алгоритм розпізнавання натискань зображено на рисунку 2.17.

Використовується дві пари двоколірних індикаторів для визначення рівня температури та вологості. Зелений сигналізує про роботу в межах допустимого значення, червоний – про перевищення порогу. Алгоритм керування індикаторами зображено на рисунку 2.18.

Детально розроблені блок-схеми алгоритмів роботи програмного забезпечення дають змогу інтуїтивно зрозуміти структуру та функції системи. Завдяки детальному проектуванню кожного компонента було визначено шлях реалізації кожного модуля, що дозволяє зменшити невизначеність і ризики переробки під час подальшого написання програм.



Рисунок 2.17 – Блок-схема алгоритму розпізнавання кнопок



Рисунок 2.18 – Блок-схема алгоритму керування LED-індикаторами

Цей процес відіграє ключову роль упродовж усього проєкту, забезпечуючи відповідність загальної функціональності очікуванням.

У цілому, проєктування та реалізація програмної системи — це систематичний і детальний процес. Лише за допомогою ретельного аналізу та проєктування можна закласти основу для розробки та забезпечити зручність обслуговування й масштабованість усього проєкту.

2.4 UML-діаграми системи

Діаграми варіантів використання слугують інструментом для дослідження загальних вимог до проєктованої системи. Ці вимоги подаються у вигляді різних сценаріїв взаємодії з системою.

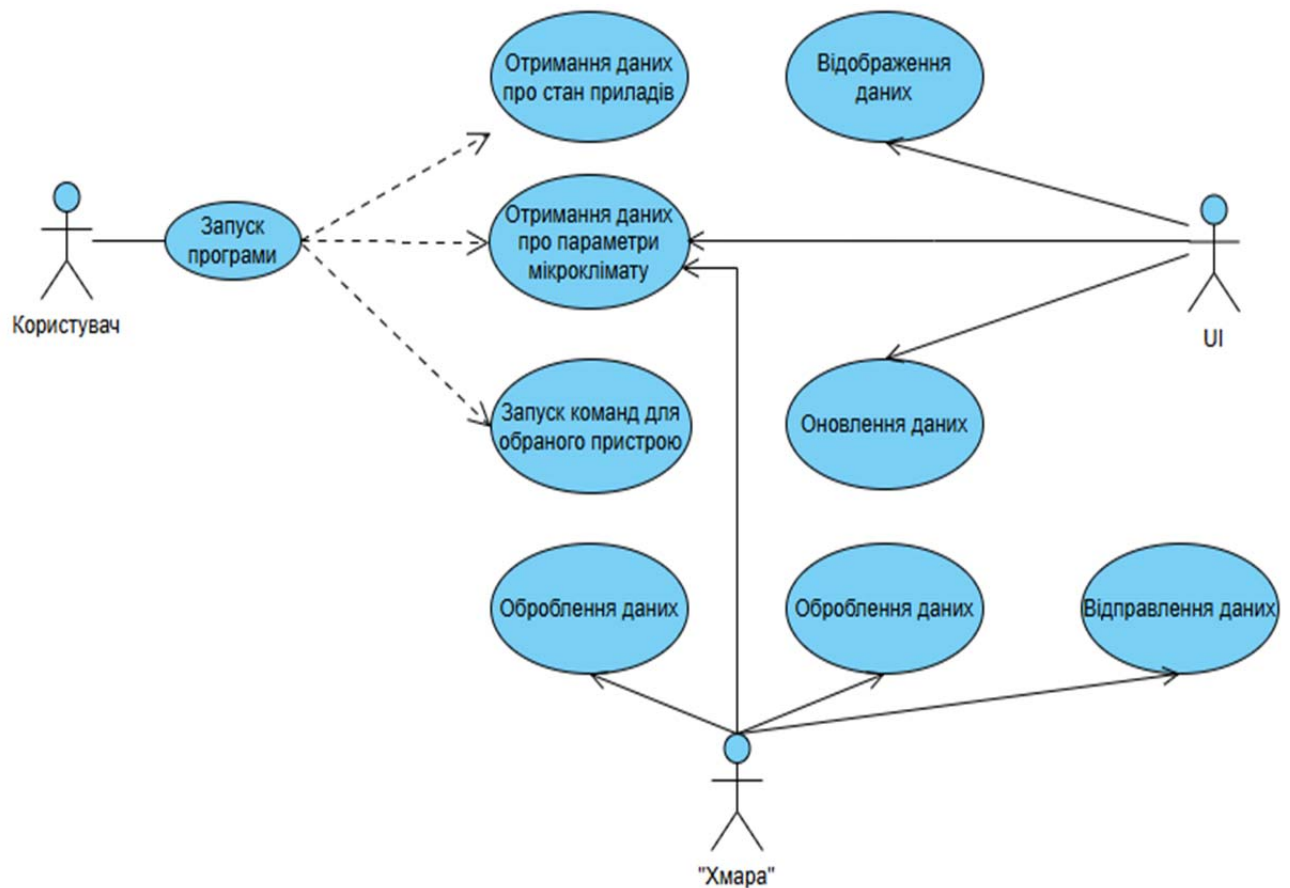


Рисунок 2.19 – UseCase діаграма

У такій діаграмі можна виокремити три ключові елементи:

- функціональні вимоги, що відображаються у формі сценаріїв використання;
- актори, які вступають у взаємодію з системою (ними можуть бути як люди, так і внутрішні чи зовнішні програмні компоненти);
- зв'язки між акторами та сценаріями, які позначаються стрілками.

Приклад діаграми варіантів використання для застосунку керування розумним будинком наведено на рисунку 2.19.

Схеми послідовностей демонструють порядок обміну повідомленнями та взаємодій між об'єктами й акторами. Актори або об'єкти вступають у дію лише тоді, коли це потрібно або коли з ними ініціює зв'язок інший об'єкт. Вся взаємодія відображається у часовій послідовності.

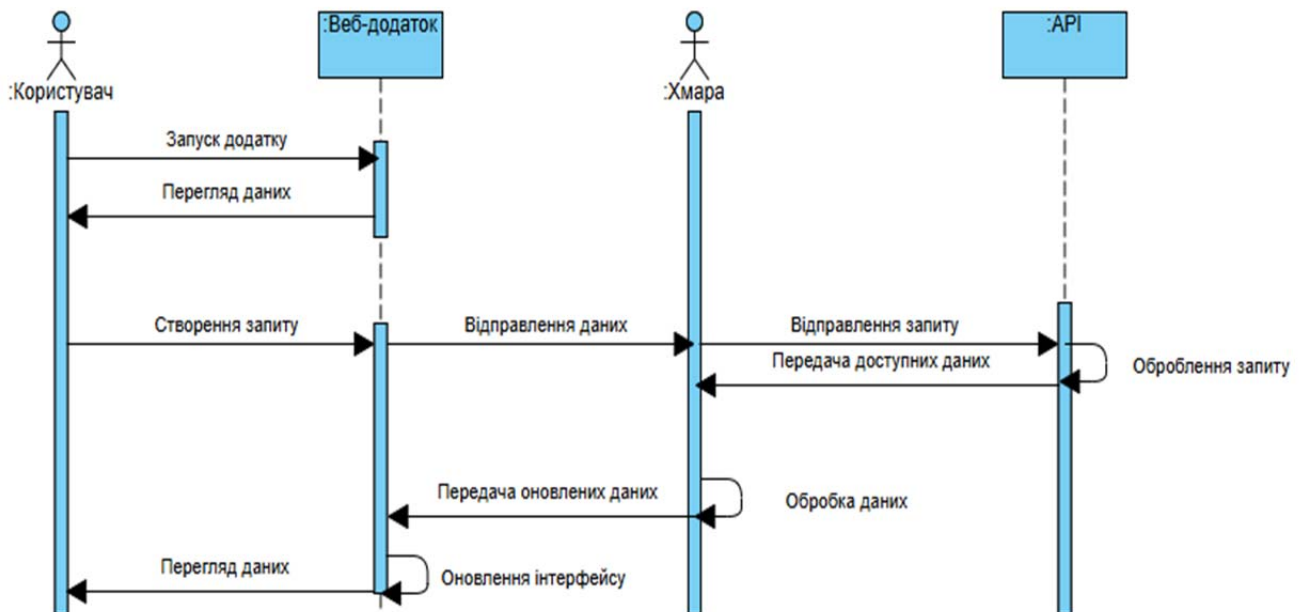


Рисунок 2.20 – Sequence-діаграма

Структурні схеми використовуються для зображення структури системи. Більш конкретно, вони застосовуються при розробці програмного забезпечення для представлення архітектури системи та того, як різні компоненти взаємопов'язані. Діаграма послідовностей зображена на рисунку 2.20.

2.5 Висновки до розділу

У цьому розділі спроектовано інтерфейс електричних з'єднань для основної плати мікроконтролера, OLED-дисплея, модуля температури та вологості DHT11 тощо. Проектування апаратної схеми є невід'ємною частиною всього проекту.

Розроблено блок-схему програмного потоку програмного забезпечення, а також блок-схеми проектування кожного компонента відповідно до принципу їх роботи, щоб відобразити структуру та функції системи. Завдяки детальному проектуванню кожного компонента було визначено шлях реалізації кожного модуля, що дозволило зменшити невизначеність і ризики переробок під час написання програмного коду у майбутньому. Цей процес відіграє ключову роль у всьому проєкті, забезпечуючи відповідність загальної функціональності очікуванням.

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		37

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Налаштування апаратного та програмного забезпечення

У цьому проєкті пропонується використовувати монтажні плати, і кожен модуль раціонально розміщено відповідно до принципової електричної схеми, і може здійснюється пайка елементів. Після виготовлення плати спочатку за допомогою мультиметра потрібно перевірити відсутність короткого замикання на конденсаторах у схемі. Далі, керуючись схемою, вимірюють вихідну напругу в ключових точках і перевіряють, чи відповідає вона очікуваним значенням. Якщо значення збігаються з проєктними, це означає, що налагодження апаратної частини завершено.

При написанні програми для мікроконтролера необхідно налагодити зовнішні модулі: зчитування температури й вологості через DHT11; передача даних через бездротовий модуль ESP8266; а також перевірити правильність поточного стану реєстрів.

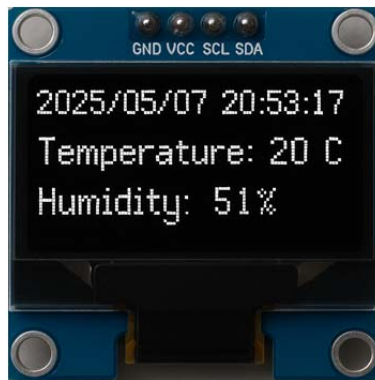


Рисунок 3.1 – Значення температури та вологості у приміщенні

Спочатку налагоджуються інтерфейси та функції кнопок. Текст буде виведено на екран. Після багаторазової перевірки переконуються, що вміст на дисплеї LCD правильний. На головному інтерфейсі дисплея відображаються значення температури та вологості, дата й годинник тощо. Головний інтерфейс OLED-дисплея показано на рисунку 3.1.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		38

Порівняно показники датчика температури та вологості DHT11 у приміщенні з вимірювальним приладом. Якщо значення, що відображається, близьке до значення на приладі, це означає, що пакет даних температури й вологості був успішно та правильно розпізнаний. Необхідно враховувати точність вимірювання сенсора. Нагрійте або зволожите область навколо датчика – значення повинні істотно змінитися. Випробування температури та вологості показано на рисунку 3.2.

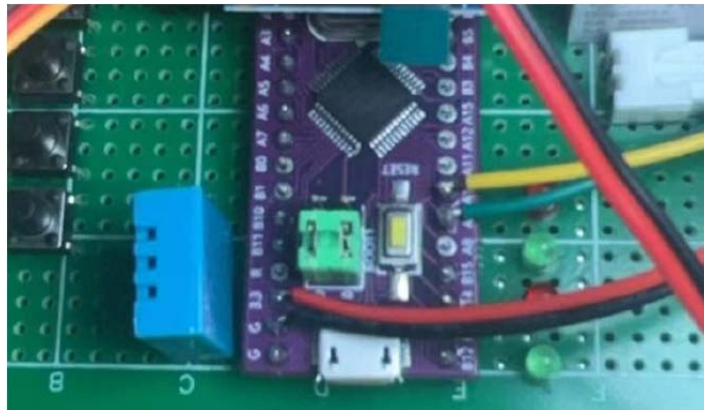


Рисунок 3.2 – Випробування сенсора температури та вологості

Щоб почати реалізацію скетчу, потрібно не лише використовувати стандартні бібліотеки, вбудовані в середовище, але й додати зовнішні – вони забезпечують зручніше управління контролером і підключенням через WI-FI, що інтегрований у сам модуль ESP8266. Бібліотека є окремим файлом із програмним кодом, призначеним для зберігання часто використовуваних методів, класів чи функцій. Існують бібліотеки двох основних типів – статичні та динамічні. Статичні бібліотеки можуть мати вигляд вихідного коду, що додається вручну програмістом під час написання програми, або представляти собою зкомпільований об'єктний файл, який включається до складу проєкту. У протилежність цьому, динамічна бібліотека – це набір окремих файлів, що містять функціонал, доступний для використання під час роботи програми, тобто завантажуються лише тоді, коли програма звертається до операційної системи із запитом на виконання певної функції з цієї бібліотеки. Якщо така бібліотека вже перебуває в оперативній пам'яті, то система повторно її не завантажує, а просто використовує наявну копію. Це дозволяє заощадити ресурси – як оперативну

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		39

пам'ять, так і час обробки, оскільки декілька додатків можуть користуватись одною вже завантаженою бібліотекою. Зазвичай динамічні бібліотеки мають встановлений формат файлів та розташовуються у визначених каталогах системи. Завдяки цій додатковій бібліотеці з'являється можливість налаштувати мікроконтролер ESP8266 для функціонування у двох доступних режимах. Перший – це точка доступу, коли мікроконтролер створює власну мережу, до якої можна приєднатися за допомогою WI-FI і, наприклад, отримати доступ до вебсервера. Інший режим – клієнтський, коли пристрій підключається до вже наявної бездротової мережі, і щоб взаємодіяти з ним, потрібно ввести його IP-адресу в браузері. У межах створеної системи мікроконтролер буде функціонувати саме в режимі клієнта. Нижче наведено частину програмного коду, де виконується підключення необхідних бібліотек:

```
#include <ESP8266WiFi.h>
```

Щоб забезпечити правильну взаємодію з бібліотекою, слід задати параметри з'єднання з точкою доступу. Йдеться про назву мережі (SSID) та відповідний пароль доступу. Це необхідно для коректного функціонування веб-сервера в межах визначеної домашньої Wi-Fi мережі. З цією метою ми оголосимо дві окремі змінні типу "char" та негайно присвоїмо їм відповідні значення. Дані бездротової мережі зберігаються саме в цих змінних. Надалі вони застосовуватимуться у процесі виклику функції підключення, що надається бібліотекою. Нижче наведено приклад коду, який відповідає за початкову ініціалізацію:

```
const char* ssid = "SSID*";  
const char* password = "SSID_password*";
```

Після підключення бібліотек потрібно оголосити змінні та здійснити перехід до основних функцій: `void setup()` і `void loop()`.

Функція `void setup()` призначена для виконання фрагменту коду тільки один раз після ввімкнення живлення плати мікроконтролера. Функція `void loop()` повторюється циклічно, тобто безперервно повторюється протягом усього часу роботи мікроконтролера.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		40

Далі здійснюємо підключення до мережі за допомогою бібліотечної функції, яка описана раніше:

```
WiFi.begin(ssid, password);
```

Процес підключення може тривати певний час тому використовується функція, яка очікує моменту встановлення підключення. Лістинг цієї функції:

```
while (WiFi.status() != WL_CONNECTED)
{
  delay(300);
  Serial.print(".");
}
```

Цикл WHILE буде працювати до тих пір поки, WiFi.status не набуде значення WL_Connected.

Після успішного підключення виводиться повідомлення:

```
Serial.println("");
Serial.println("WiFi connected");
```

Далі запускаємо веб-сервер:

```
server.begin();
Serial.println("Server started");
```

Інтерфейс користувача створюється за допомогою HTML і CSS – саме ці технології використовуються для його розробки. HTML є текстовим файлом, який підпорядковується певним синтаксичним правилам щодо назв, форматування та структурних елементів. Ці правила визначають, що файл потрібно сприймати саме як HTML-документ, а також інформують веб-сервер або комп'ютер про його призначення. Коли користувач застосовує відповідні правила до звичайного текстового документа у редакторі, він має можливість самостійно створити просту веб-сторінку.

Головним і базовим із HTML-конвенцій є розміщення декларації типу документа на самому початку коду. Її обов'язково потрібно писати першою – саме вона сигналізує системі, що перед нами HTML-файл. Зазвичай цей заголовок виглядає так: `<!DOCTYPE html>`. Запис обов'язковий без додаткового вмісту або

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		41

зміни структури. Усе, що знаходиться до цієї декларації, комп'ютер не буде інтерпретувати як частину HTML-документу.

Після цього відбувається запуск веб-сервера на мікроконтролері ESP8266. Головна функція такого сервера – отримувати запити через HTTP-протокол, обробляти їх та повертати відповідь у форматі HTML. Щоб отримати доступ до веб-сервера, користувач повинен під'єднатися до цієї самої Wi-Fi мережі, що й пристрій, а потім увести його IP-адресу в адресному рядку браузера.

Лістинг коду налаштування цілої системи подано в додатку А.

Використано модуль послідовної передачі даних для підключення модуля ESP8266 до комп'ютера. Згідно з технічним описом, введено AT-команди через інструмент для роботи з послідовним портом і перевірено, чи відповідає отримана відповідь очікуванню [27]. Потрібно відкрити мережевий відладчик, встановити протокол TCP-сервер, ввести IP-адресу та номер порту сервера, а потім натиснути «Створити сервер». Після цього ініціюється TCP-з'єднання для прийому та передавання даних. Дані спочатку надсилаються із сервера на модуль ESP8266, а потім передаються через послідовний порт у програму налагодження, де приймаються та відображаються. Інтерфейси мережевого відладчика та інструменту роботи з COM-портом показані на рисунку 3.4 та 3.3 відповідно.

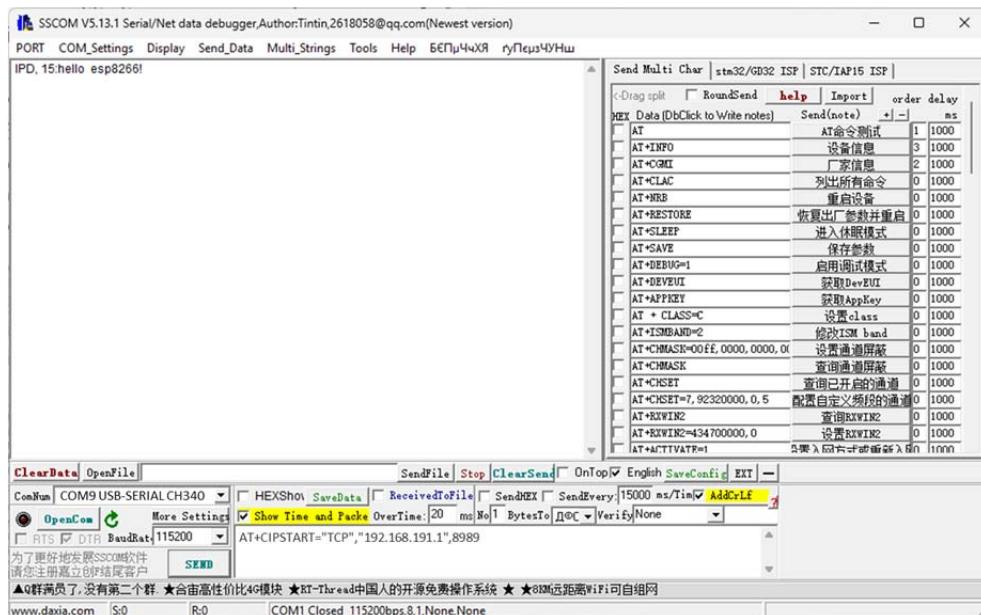


Рисунок 3.3 – Інтерфейс інструменту COM-порту

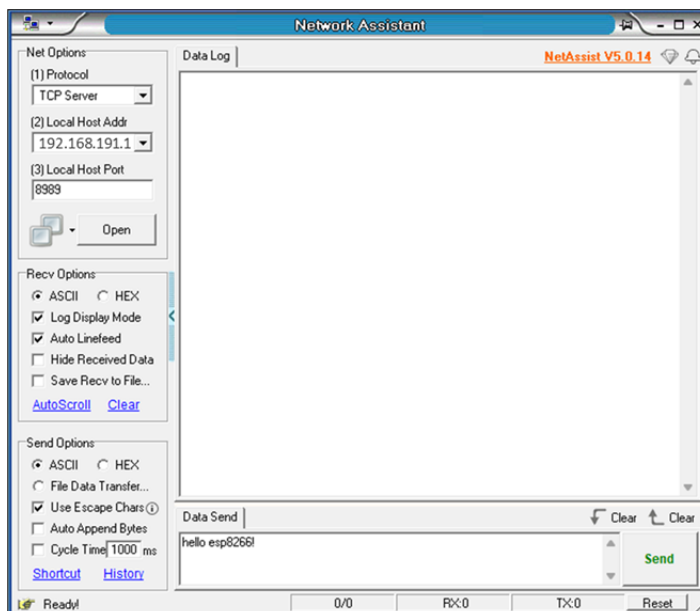


Рисунок 3.4 – Інструмент налагодження мережі

Хмарна платформа USR компанії з технологій Інтернету речей (Китай), яка спеціалізується на розробці, виробництві та продажу комунікаційної продукції й є провідним постачальником програмних та апаратних рішень для промислового Інтернету речей. Компанія має власне виробництво та здатна забезпечити повний цикл – від чипів до хмари. Вона надає клієнтам комунікаційні термінали, IoT-модулі, хмарні платформи та комплексні IoT-рішення.

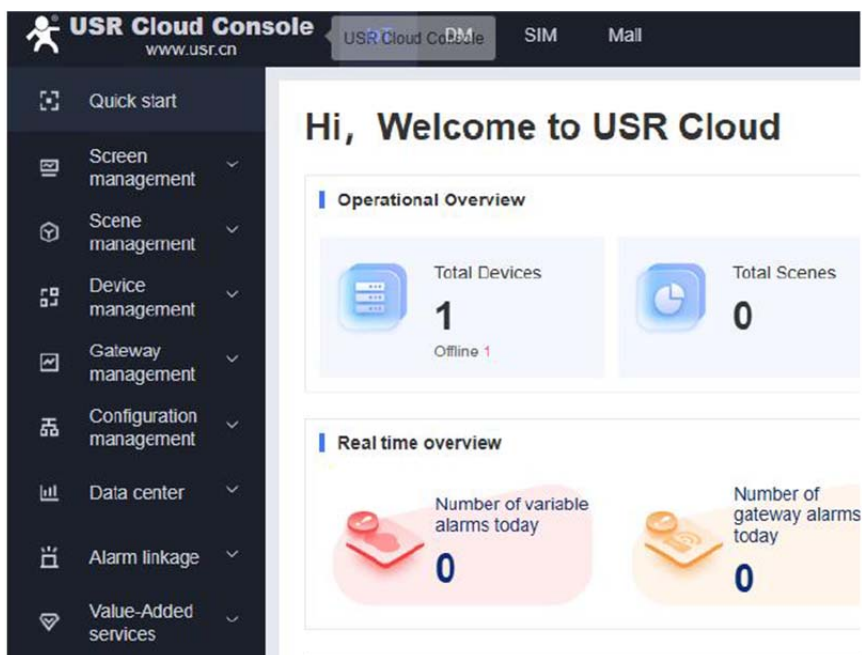


Рисунок 3.5 – USR cloud platform

Для окремих користувачів доступна хмарна платформа USR, яка дозволяє налаштовувати апаратне та програмне забезпечення, додати пристрої в хмару, а потім побудувати IoT-додатки. Користувачі можуть гнучко налаштовувати функції відповідно до власних цілей. Платформа підтримує керування кількома пристроями, а доступ до неї можливий через мініпрограми WeChat або вебсторінки для перегляду даних і керування пристроями. Інтерфейс хмарної платформи USR показано на рисунку 3.5.

Консоль USR Cloud має широкий функціонал: вона може в реальному часі відстежувати онлайн-статус кожного пристрою, формувати оглядову архітектуру, керувати сценаріями використання пристроїв і їхньою конфігурацією. Також підтримується функція тривожного зв'язування тощо. Консоль USR показано на рисунку 3.6.

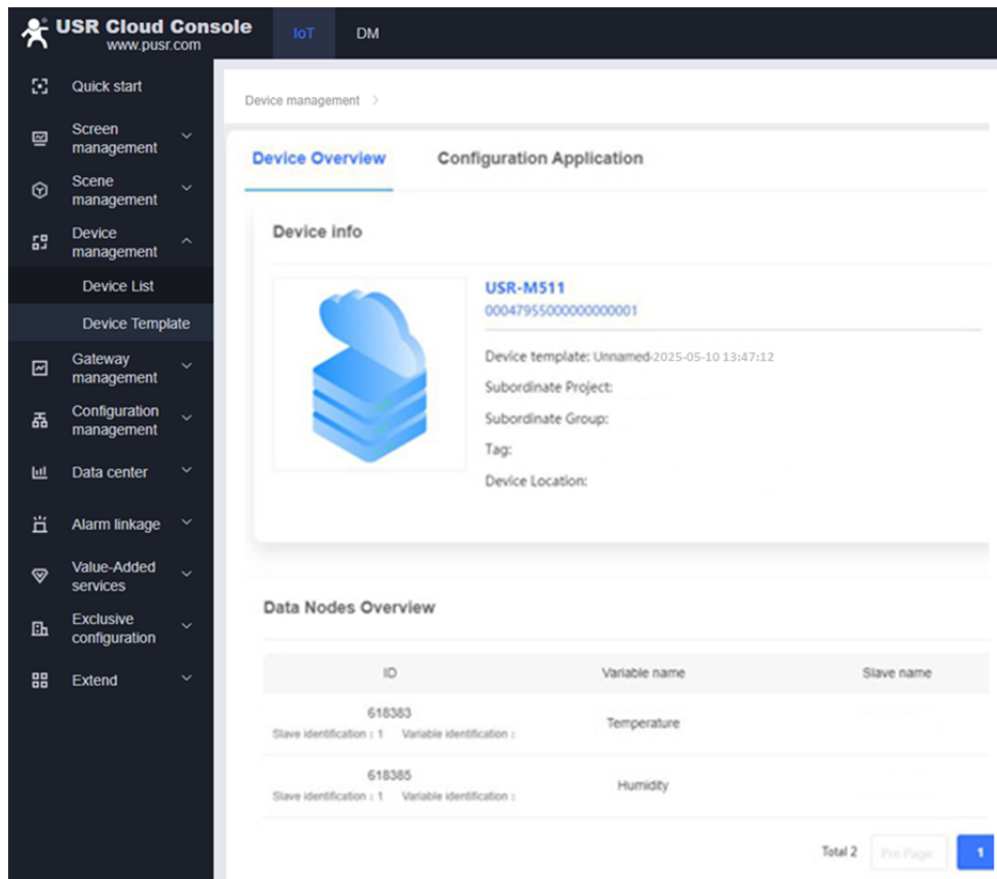


Рисунок 3.6 – USR Cloud Console

У цьому проєкті на платформі USR Cloud налаштовано різні змінні, і за допомогою функціоналу платформи безпосередньо здійснюється зчитування даних, таких як температура і вологість. Крім того, пристроєм можна керувати онлайн через консоль: можна окремо вмикати компоненти, а також змінювати порогові значення температури й вологості онлайн.

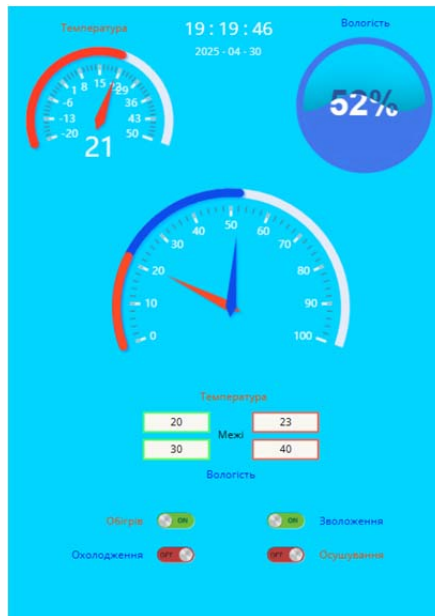


Рисунок 3.7 – USR cloud platform APP interface

Увійшовши на платформу розробки USR Cloud, можна безкоштовно створювати хмарні застосунки, налаштовувати інтерфейс для параметрів температури, вологості й інших змінних, вводити байтову інформацію змінних згідно з командами послідовного порту. Після завершення створення застосунку можна вибрати автоматично згенероване доменне ім'я або прив'язати власне. Інтерфейс застосунку USR Cloud показано на рисунку 3.7.

3.2 Висновки до розділу

У проєкті системи підтримки сталої температури та вологості в домі комплексно враховано кілька факторів, таких як продуктивність, енергоспоживання, доцільність, надійність тощо. Завдяки обчислювальним можливостям STM32 вдалося досягти вимог проєкту та реалізувати керування – від простих датчиків до інтеграції з IoT-пристроями.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		45

Проведено апаратне налагодження, програмне налагодження та функціональну перевірку.

Було впроваджено вирішення поставленої задачі, при цьому особливий акцент робився на використанні безкоштовного програмного забезпечення та додаткових інструментів, які надають можливість створювати недороге, просте за структурою програмне забезпечення. Таке рішення є повністю доступним для широкого кола користувачів і розробників, які мають змогу звернутись до відкритого програмного коду цієї системи.

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		46

ВИСНОВКИ

На основі переліку джерел посилань було виконано аналіз обраної предметної галузі проекту. В його ході було виділено основну мету створення запланованої системи, сформована та описана задача, функціональна складова системи та її структурний вигляд, були відповідно виділені вхідні та вихідні інформаційні дані системи та сформульовані вимоги до проєктованої системи.

В результаті роботи було:

- проведено аналіз систем-аналогів;
- визначено вимоги до функціоналу системи;
- створено автоматизовану систему;

Проєкт системи підтримання заданої температури та вологості у домі на основі мікроконтролера STM32 є інтегрованим та інтелектуальним рішенням. Він поєднує периферійні пристрої, такі як датчики температури, релейні модулі та дисплейні модулі, для створення системи домашньої автоматизації, яка може в реальному часі моніторити та регулювати температуру в приміщенні. Мікроконтролер може в реальному часі збирати дані з датчиків температури та вологості, виконувати попередню обробку та аналіз, щоб керувати клапанами подачі гарячої води, кондиціонерами та вимикачами зволожувача повітря. Що стосується віддаленого моніторингу та управління, то віддалене спостереження через мобільний додаток досягається завдяки модулю WIFI та хмарній платформі, що дозволяє здійснювати віддалений моніторинг та інтелектуальне управління внутрішнім середовищем.

Крім того, стабільність та безпека бездротової мережі є важливою частиною проєкту, яку не можна ігнорувати. Необхідно вжити заходів для забезпечення своєчасної передачі даних і безпечної роботи системи. Для вирішення цих проблем необхідно продовжити дослідження, оптимізувати конструктивну схему та покращити результати практичного застосування.

					<i>БР.КІ-12.00.00.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		47

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1 Shea, S. (2020). “Smart Home or Building (home automation or domotics).” URL : <https://internetofthingsagenda.techtarget.com> – Дата доступу: 15.04.2025.

2 Zhu Mingshuo. Development trend and application analysis of smart home system. Intelligent Building Electrical Technology, 2024, 18(01): 143-147.

3 Laurenson, C. (2021). “Smart Home Choices Wired vs. Wireless”. URL : <https://www.tltechsmart.com> (дата звернення: 25.02.2024).

4 Peng Yong, Tao Zengjie, Lin Zhen, et al. Design of smart home system based on STM32 and OneNET [J]. Internet of Things Technology, 2024, 14(02): 86-89.

5 Simonet, A. C.; Noyce, A. J. (2021). “Domotics, Smart Homes, and Parkinson’s Disease.” Journal of Parkinson’s Disease 11(s1): 1-9.

6 Guan Jiacheng, Li Xiaofeng, Huang Zhifang, Wu Peiwei. Internet of Things smart home based on ESP8266 and Gizwits Cloud. Internet of Things Technology, 2023, 13(03): 140-142.

7 Your helpful home starts here (Google Home app).” URL : <https://internetofthingsagenda.techtarget.com> – Дата доступу: 16.04.2025.

8 Liu Shibing, Liu Chengyin. Water quality online monitoring system based on human cloud. Internet of Things Technology, 2021, 11(10): 30-32.

9 CONTROL4 SMART HOMES. URL : <https://www.control4.com> – Дата доступу: 05.04.2025.

10 Chen Yacong, Feng Xingle, Chen Shupeng. Smart home modular expansion system based on MQTT protocol. Integrated Circuits and Embedded Systems, 2024, 24(02): 86-90.

11 This is Home Automation. URL : <https://hubitat.com> – Дата доступу: 27.05.2025.

12 Cheng Xuemin. Design of constant temperature intelligent control system based on STM32F103C8T6. Electronic Production, 2023, 31(19):26-29.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		48

13 S. Tsyurulnyk, Mobile applications and online WI-FI monitoring platforms of weather stations, Open educational e-environment of modern University, volume 9, 2020, pp. 181–192.

14 Hu Shuai, Jiang Yafeng, Du Mingquan, et al. Intelligent energy-saving constant temperature control device based on STM32. Mechanical Manufacturing and Automation, 2020, 49(04):184-187.

15 S. Tsyurulnyk, M. Tsyurulnyk, Temperature and humidity monitoring system on the IOT module, in: Proceedings of the II NPK. Computer Technologies of Data Processing (CTDP-2021), Vinnytsia, 2021, pp. 105-108.

16 Wu Yangbo, Yan Jinneng, Hu Haifeng. Design and implementation of smart home environment monitoring system based on ESP8266. Communications and Information Technology, 2024, (02):10-14.

17 European Smart Grids Technology Platform: vision and Strategy for Europe's Electricity Networks of the Future. – European Commission, 2006. – 44 p. URL : <https://op.europa.eu/en/publication-detail/-/publication/a2ea8d86-7216-444d-8ef5-2d789fa890fc/language-en> (дата звернення: 08.04.2025).

18 Wang Ziliang. Research on the design of temperature and humidity monitoring system based on single chip microcomputer. Modern Salt Chemical Industry, 2019, 46(06):72-73.

19 Що таке розумний будинок? Все що потрібно знати про систему Розумний Дім. URL : <https://bron.ua/article/schotake-rozumnij-budinok-vse-scho-potrбно-znati-prosistemu-rozumnij-dm/5/> (дата звернення : 08.03.2025).

20 Ma Qiang. Design and implementation of smart home system based on Internet of Things technology. Electronic Technology, 2024, 53(02): 252-253.

21 Все, що необхідно для розробки якісних додатків. URL : <https://visualstudio.microsoft.com/en/free-developerooffers/> (дата звернення: 19.02.2024).

22 Jie Xiao, Jing Tao Li. Design and Implementation of Intelligent Temperature and Humidity Monitoring System Based on ZigBee and WiFi[J]. Procedia Computer Science, 2020, 166.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		49

23 Ubiquiti Networks – Wi-Fi і мережеве обладнання. URL : <https://www.ui.com> / (дата звернення: 22.02.2025).

24 В. Orobchuk, V. Koval. Development and research of Wi-Fi network for receiving and tramitting telemechanical information in the training laboratory // Scientific Journal of TNTU (Tern.) // Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine, 2020. URL : <https://visnyk.tntu.edu.ua/pdf/99/570.pdf>

25 K. Küçük, C. Bayılmış, D. L. Msongaleli, Designing real-time IoT system course: prototyping with cloud platforms, laboratory experiments and term project, IJEEE, 58(3), 2021, pp. 743-772.

26 D. Karabekova, P. Kissabekova, V. Kucheruk, E. Mussenova, & S. Azatbek. Main characteristics of the heat flow meter, Eurasian Physical Technical Journal, 19.2(40), 2022, pp. 71-74.

27 Юемін Лі. Система постійної температури і вологості у домі : дипломний проект : 123 Комп'ютерна інженерія / Юемін Лі – Київ, 2024. – 88 с.

					БР.КІ-12.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		50

ДОДАТКИ

Програмний код системи

```
void handleKeyInput(void)
{
    // Обробка кнопки налаштування часу
    if (IsKeyDown3())
    {
        bsp_DelayMS(10);
        if (IsKeyDown3())
        {
            set_u_type = 0;
            set_time_type++;

            if (set_time_type == 1)
            {
                // Ініціалізація змінних з поточного часу
                set_year = g_tRTC.Year;
                set_mon = g_tRTC.Mon;
                set_day = g_tRTC.Day;
                set_hour = g_tRTC.Hour;
                set_min = g_tRTC.Min;
                set_sec = g_tRTC.Sec;

                OLED_ClrScr(0); // Очистити екран
            }
            else if (set_time_type == 7)
            {
                set_time_type = 0;
                RTC_WriteClock(set_year, set_mon, set_day,
set_hour, set_min, set_sec);
            }

            while (IsKeyDown3()); // Очікувати відпускання
кнопки
        }
    }

    // Обробка кнопки налаштування порогів
    if (IsKeyDown4())
    {
        bsp_DelayMS(10);
        if (IsKeyDown4())
        {
            set_time_type = 0;
            set_u_type++;

            if (set_u_type == 1)
            {
                OLED_ClrScr(0);
            }
            else if (set_u_type == 5)
            {
```

```
        set_u_type = 0;
    }

    while (IsKeyDown4());
}

// Обробка кнопки зменшення значення
if (IsKeyDown1())
{
    bsp_DelayMS(10);
    if (IsKeyDown1())
    {
        // Зменшення значень часу
        switch (set_time_type)
        {
            case 1: if (--set_year < 2024) set_year = 2024;
break;

            case 2: if (--set_mon < 1) set_mon = 12; break;
            case 3: if (--set_day < 1) set_day = 31; break;
            case 4: if (--set_hour < 0) set_hour = 23;
break;

            case 5: if (--set_min < 0) set_min = 59; break;
            case 6: if (--set_sec < 0) set_sec = 59; break;
        }

        // Зменшення значень порогів
        switch (set_u_type)
        {
            case 1: temp_ul--; break;
            case 2: temp_uh--; break;
            case 3: hum_ul--; break;
            case 4: hum_uh--; break;
        }

        while (IsKeyDown1());
    }
}

// Обробка кнопки збільшення значення
if (IsKeyDown2())
{
    bsp_DelayMS(10);
    if (IsKeyDown2())
    {
        // Збільшення значень часу
        switch (set_time_type)
        {
            case 1: if (++set_year > 2099) set_year = 2024;
break;

            case 2: if (++set_mon > 12) set_mon = 1; break;
            case 3: if (++set_day > 31) set_day = 1; break;

```

```
case 4: if (++set_hour > 23) set_hour = 0; break;
        case 5: if (++set_min > 59) set_min = 0; break;
        case 6: if (++set_sec > 59) set_sec = 0; break;
    }

    // Збільшення значень порогів
    switch (set_u_type)
    {
        case 1: temp_ul++; break;
        case 2: temp_uh++; break;
        case 3: hum_ul++; break;
        case 4: hum_uh++; break;
    }

    while (IsKeyDown2());
}
}

int main(void)
{
    char displayBuffer[16];
    uint16_t co2Value;

    // Налаштування шрифтів
    {
        tFont16.FontCode = FC_ST_16;
        tFont16.FrontColor = 1;
        tFont16.BackColor = 0;
        tFont16.Space = 0;

        tFont12.FontCode = FC_ST_12;
        tFont12.FrontColor = 1;
        tFont12.BackColor = 0;
        tFont12.Space = 0;
    }

    initialize_peripherals();
    OLED_DisplayString(0, 0, "Power ON", &tFont12);

    start_timer(0, 200); // Таймер 0: 200 мс
    start_timer(1, 1000); // Таймер 1: 1000 мс

    configure_external_interrupts();

    while (1)
    {
        process_wifi_reception();
        handle_automatic_tasks();
        check_buttons();

        if (is_timer_triggered(0))
```

```
{
    if (DHT11_GetData(&sensorData))
    {
        temperature = sensorData.Temp;
        humidity = sensorData.Hum;
    }

    OLED_BeginDraw();

    read_current_time();

    if (timeSettingMode == 0 && unitSettingMode == 0)
    {
        // Вивід реального часу
        sprintf(displayBuffer, "%4d/%02d/%02d
%02d:%02d:%02d",
                currentTime.Year, currentTime.Month,
currentTime.Day,
                currentTime.Hour, currentTime.Minute,
currentTime.Second);
        OLED_DisplayString(0, 0, displayBuffer,
&tFont12);

        sprintf(displayBuffer, "T:%3d°C", temperature);
        OLED_DisplayString(0, 16, displayBuffer,
&tFont16);

        sprintf(displayBuffer, "H:%3d%%", humidity);
        OLED_DisplayString(0, 32, displayBuffer,
&tFont16);

        sprintf(displayBuffer, "%4d", fanSpeed);
        OLED_DisplayString(0, 48, displayBuffer,
&tFont16);
    }
    else
    {
        // Вивід часу налаштування
        sprintf(displayBuffer, "%4d/%02d/%02d
%02d:%02d:%02d",
                setupYear, setupMonth, setupDay,
                setupHour, setupMinute, setupSecond);
        OLED_DisplayString(0, 0, displayBuffer,
&tFont12);

        sprintf(displayBuffer, "TL:%3d TH:%3d",
tempLowLimit, tempHighLimit);
        OLED_DisplayString(0, 16, displayBuffer,
&tFont16);

        sprintf(displayBuffer, "HL:%3d HH:%3d",
humLowLimit, humHighLimit);
```

```
        OLED_DisplayString(0, 32, displayBuffer,
&tFont16);
    }

    OLED_EndDraw();
}

if (is_timer_triggered(1))
{
    fanSpeed = pulseCount;
    pulseCount = 0;

    send_data_via_wifi();
}

return 0;
}

void ProcessCommand(uint8_t *data, uint8_t length)
{
    uint16_t value = (data[4] << 8) | data[5];

    switch (data[3]) {
        case 3: // Set lower temperature limit
            temp_ul = value;
            break;

        case 4: // Set upper temperature limit
            temp_uh = value;
            break;

        case 5: // Set lower humidity limit
            hum_ul = value;
            break;

        case 6: // Set upper humidity limit
            hum_uh = value;
            break;

        case 7: // Humidifier control
            if (data[5] == 1) {
                comSendBuf(COM1, (uint8_t *)ack_7_1, 8);
                bsp_jk_on(1);
                jk1_sw = 1;
            } else {
                comSendBuf(COM1, (uint8_t *)ack_7_0, 8);
                bsp_jk_off(1);
                jk1_sw = 0;
            }
            break;
    }
}
```

```
case 8: // Air conditioner control
    if (data[5] == 1) {
        comSendBuf(COM1, (uint8_t *)ack_8_1, 8);
        bsp_jk_on(2);
        jk2_sw = 1;
    } else {
        comSendBuf(COM1, (uint8_t *)ack_8_0, 8);
        bsp_jk_off(2);
        jk2_sw = 0;
    }
    break;

case 9: // Fan control
    if (data[5] == 1) {
        comSendBuf(COM1, (uint8_t *)ack_9_1, 8);
        bsp_fan_on();
        fan_sw = 1;
    } else {
        comSendBuf(COM1, (uint8_t *)ack_9_0, 8);
        bsp_fan_off();
        fan_sw = 0;
    }
    break;

case 10: // Water valve control
    if (data[5] == 1) {
        comSendBuf(COM1, (uint8_t *)ack_10_1, 8);
        bsp_open_water();
        step_motor_sw = 1;
    } else {
        comSendBuf(COM1, (uint8_t *)ack_10_0, 8);
        bsp_close_water();
        step_motor_sw = 0;
    }
    break;

default:
    // Unknown command; optionally handle or ignore
    break;
}
}

void manage_heating(void) {
    if (temp <= temp_ul) {
        step_motor_off_flag = 0;
        if (step_motor_on_flag == 0) {
            step_motor_on_flag = 1;
            step_motor_sw = 1;
            bsp_open_water();
        }
    } else {
        step_motor_on_flag = 0;
    }
}
```

```

        if (step_motor_off_flag == 0) {
            step_motor_off_flag = 1;
            step_motor_sw = 0;
            bsp_close_water();
        }
    }
}

void manage_air_conditioning(void) {
    if (temp >= temp_uh) {
        jk2_off_flag = 0;
        if (jk2_on_flag == 0) {
            jk2_on_flag = 1;
            jk2_sw = 1;
            bsp_jk_on(2);
        }
    } else {
        jk2_on_flag = 0;
        if (jk2_off_flag == 0) {
            jk2_off_flag = 1;
            jk2_sw = 0;
            bsp_jk_off(2);
        }
    }
}

void manage_humidifier(void) {
    if (hum <= hum_ul) {
        jk1_off_flag = 0;
        if (jk1_on_flag == 0) {
            jk1_on_flag = 1;
            jk1_sw = 1;
            bsp_jk_on(1);
        }
    } else {
        jk1_on_flag = 0;
        if (jk1_off_flag == 0) {
            jk1_off_flag = 1;
            jk1_sw = 0;
            bsp_jk_off(1);
        }
    }
}

void manage_fan(void) {
    if (hum >= hum_uh) {
        fan_off_flag = 0;
        if (fan_on_flag == 0) {
            fan_on_flag = 1;
            fan_sw = 1;
            bsp_fan_on();
        }
    }
}

```

```
    } else {
        fan_on_flag = 0;
        if (fan_off_flag == 0) {
            fan_off_flag = 1;
            fan_sw = 0;
            bsp_fan_off();
        }
    }
}

void update_leds(void) {
    if (step_motor_on_flag == 0 && jk2_on_flag == 0) {
        bsp_led_off(2);
        bsp_led_on(1);
    } else {
        bsp_led_off(1);
        bsp_led_on(2);
    }

    if (jk1_on_flag == 0 && fan_on_flag == 0) {
        bsp_led_off(4);
        bsp_led_on(3);
    } else {
        bsp_led_off(3);
        bsp_led_on(4);
    }
}

void task_auto(void) {
    manage_heating();
    manage_air_conditioning();
    manage_humidifier();
    manage_fan();
    update_leds();
}

void wifi_task_receive(void);
static void configure_EXTI(void);
void wifi_task_transmit(void)
{
    uint8_t data_buffer[32];
    uint16_t checksum;

    data_buffer[0] = 0x01; // Slave Address
    data_buffer[1] = 0x46; // Function code
    data_buffer[2] = 0x00; // Register Address
    data_buffer[3] = 0x00;
    data_buffer[4] = 0x00; // Number of registers
    data_buffer[5] = 11;
    data_buffer[6] = 22; // Number of data bytes
    data_buffer[7] = 0;
    data_buffer[8] = temperature;
    data_buffer[9] = 0;
}
```

```
data_buffer[10] = humidity;
data_buffer[11] = motor_speed >> 8;
data_buffer[12] = motor_speed;
data_buffer[13] = 0;
data_buffer[14] = temp_low;
data_buffer[15] = 0;
data_buffer[16] = temp_high;
data_buffer[17] = 0;
data_buffer[18] = humidity_low;
data_buffer[19] = 0;
data_buffer[20] = humidity_high;
data_buffer[21] = 0;
data_buffer[22] = switch1_status;
data_buffer[23] = 0;
data_buffer[24] = switch2_status;
data_buffer[25] = 0;
data_buffer[26] = fan_status;
data_buffer[27] = 0;
data_buffer[28] = motor_step_status;

checksum = calculate_CRC16(data_buffer, 29);
data_buffer[29] = checksum >> 8;
data_buffer[30] = checksum;

send_data(COM1, data_buffer, 31);
}

void open_water_system(void)
{
    set_pwm_timeout(GPIOA, GPIO_Pin_1, TIM2, 2, 50, 300);
}

void close_water_system(void)
{
    set_pwm_timeout(GPIOA, GPIO_Pin_1, TIM2, 2, 50, 1000);
}

void wifi_task_receive(void)
{
    uint8_t incoming_data;
    static uint8_t header_status = 0;
    static uint8_t buffer[64];
    static uint8_t buffer_position = 0;

    while (1)
    {
        if (receive_data(COM1, &incoming_data))
        {
            if (header_status == 0)
            {
                if (incoming_data == 0x01) // Data packet start
                {
```

```
        header_status = 1;
        buffer_position = 0;
        buffer[buffer_position++] = incoming_data;
    }
}
else if (header_status == 1)
{
    if (incoming_data == 0x06) // Data packet
continuation
    {
        header_status = 2;
        buffer[buffer_position++] = incoming_data;
    }
    else
    {
        if (buffer_position < sizeof(buffer))
        {
            buffer[buffer_position++] =
incoming_data;

            if (buffer_position == 8)
            {
                process_data(buffer,
buffer_position);

                header_status = 0;
            }
            else
            {
                header_status = 0;
            }
        }
        continue; // Continue analyzing the incoming
data
    }
    break;
}
}

static void configure_EXTI(void)
{
    GPIO_InitTypeDef gpio_config;
    EXTI_InitTypeDef exti_config;
    NVIC_InitTypeDef nvic_config;

    // Enable GPIO clock
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);

    // Configure PA0 as input floating mode
    gpio_config.GPIO_Pin = GPIO_Pin_0;
    gpio_config.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_Init(GPIOA, &gpio_config);
}
```

```
// Enable AFIO clock
RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);

// Connect EXTI Line to PA0
GPIO_EXTILineConfig(GPIO_PortSourceGPIOA, GPIO_PinSource0);

// Configure EXTI Line0
exti_config.EXTI_Line = EXTI_Line0;
exti_config.EXTI_Mode = EXTI_Mode_Interrupt;
exti_config.EXTI_Trigger = EXTI_Trigger_Rising; // Rising
edge trigger
exti_config.EXTI_LineCmd = ENABLE;
EXTI_Init(&exti_config);

// Enable EXTI0 Interrupt with lowest priority
nvic_config.NVIC_IRQChannel = EXTI0_IRQn;
nvic_config.NVIC_IRQChannelPreemptionPriority = 0x0F;
nvic_config.NVIC_IRQChannelSubPriority = 0x0F;
nvic_config.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&nvic_config);
}
```

Форм.	Зона	Поз.	Позначення	Найменування	Кіл.	Прим.
				<i>Акумулятор</i>		
			CR1	CR1220 3V	1	
				<i>Конденсатори</i>		
			C1	470-220-NCC-KXJ	3	
			C2, C8-C11	100nF 100V ±5% CL11 (2A104J)	5	
			C3	10uF 35V size-C 10% (TAJС106K035RNJ-AVX)	1	
			C4, C5	10pF 50V COG 5% 0603 AVX	2	
			C6, C7	6.8pF 50V NPO ±5%	2	
				<i>Діоди</i>		
			D1, D2	BAT54C UTC SOT-23	2	
				<i>Світлодіоди</i>		
			D3, D4	LTST-C193TGKT-5A (зелений)	2	
			D5, D6	LTSTC191KRKT (червоний)	2	
				<i>Роз'єми</i>		
			CON1	DC Power Plug 5.5×2.1mm Female	1	
			JP1	HDR1X2-M254	1	
			JP2, JP3	HDR1X3-M254	2	
				<i>Реле</i>		
			JK1, JK2	HF49FD/005-1H11 5A 250VAC/30VDC	2	
				<i>Дисплей</i>		
			OLED1	SH1107 OLED Display 12864, 128 x 64 0,96	1	
				<i>Транзистори</i>		
			Q1, Q2, Q3	A03400 SOT-23 FET N-ch 30V 5A	3	
				<i>Резистори</i>		
			R1, R4, R5,			
			R11, R13, R14,	0603-0,1-10 кОм±5%	8	
			R2, R3, R6,			
			R8, R10, R12	0603-0,1-1 кОм±5%	7	
				<i>Кнопки</i>		
			SW1-5	4 Pin SMD Push Button Switch Black6x6x6mm	5	
				<i>Мікросхеми</i>		
			U1	ME6211C33M5G-N SOT23-5 3.3V 0.5A	1	
			U2	DHT11	1	
			U3	ESP8266-12	1	
			U4	STM32F103C8T6	1	

БР.КІ-12.00.00.000 ПЕ

Змн. Арк. № док. Підп. Дата

Розроб. Мужик О. М.
 Перевір. Кропивницька В.Б.
 Реценз. Заячук Я. І.
 Н.контр. Лазорів А.М.
 Затверд. Мельничук С.

Розробка мікроконтролерної системи підтримання заданої температури та вологості у приміщенні на основі STM32F

Літ. Арк. Аркушів

62 1

ІФНТУНГ, КІ-21-2

Протокол аналізу звіту подібності науковим керівником

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

Автор: Мужик Олександр Мирославович

Співавтор:

Назва: Муїук

Науковий керівник: Кропивницька Віталія Богданівна

Підрозділ: Каф. КСМ

Коефіцієнт подібності 1: 1.2%

Мікропробіли: 1

Заміна букв: 2

Інтервали: 0

Білі знаки: 147

Дата створення звіту: 2025-05-07 17:37:57.0

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:

2025-05-07

Надія Ширмовська

Дата

експерт

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: "Розробка мікроконтролерної системи підтримання заданої температури та вологості у приміщенні на основі STM32F"

Обсяг пояснювальної записки 50 аркушів.

41 рисунки;

- таблиць;

1 додаток.

Дата завершення роботи: *10 червня 2025 р.*

Підпис студента-дипломника _____ *Мужик О. М.*