

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО
МОНІТОРИНГУ МІКРОКЛІМАТУ МЕДИЧНИХ ПРИМІЩЕНЬ.
SOFTWARE FOR REMOTE MONITORING OF MEDICAL PREMISES
MICROCLIMATE.

Виконав: студент 4 курсу, групи КНТ-218

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Третяк М.С.

(прізвище та ініціали)

Керівник Пархоменко А.В.

(прізвище та ініціали)

Рецензент Поляков М.О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет комп'ютерних наук і технологій
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Третяка Микити Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для віддаленого моніторингу мікроклімату медичних приміщень. Software for Remote Monitoring of Medical Premises Microclimate.

керівник проєкту (роботи) Пархоменко Анжеліка Володимирівна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 13 травня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Дослідження технологій організації бездротових сенсорних мереж. 3. Проєктування підсистеми віддаленого моніторингу показників мікроклімату. 4. Конструювання підсистеми віддаленого моніторингу показників мікроклімату 5. Керівництво програміста. 6. Керівництво користувача

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-6 Основна частина	Пархоменко А.В., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “20” лютого 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка архітектури підсистеми	4 тиждень	Розділ 2
4	Розробка програми	5-6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент

(підпис) Третяк М.С.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Пархоменко А.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
92 с., 32 рис., 15 табл., 2 дод., 54 джерела.

МЕДИЧНЕ ПРИМІЩЕННЯ, МІКРОКЛІМАТ, ДАТЧИК
ТЕМПЕРАТУРИ ТА ВОЛОГОСТІ, БЕЗДРОТОВА СЕНСОРНА МЕРЕЖА,
МОДУЛЬ ПЕРЕДАЧІ ДАНИХ, BLUETOOTH, ARDUINO IDE.

Об'єкт дослідження – процес віддаленого моніторингу показників мікроклімату медичних приміщень.

Предмет дослідження – програмні та апаратні засоби віддаленого моніторингу показників мікроклімату медичних приміщень.

Мета роботи – розробка програмного забезпечення для віддаленого моніторингу показників мікроклімату медичних приміщень на основі бездротової сенсорної мережі.

Матеріали, методи та технічні засоби: мова програмування C++, середовище розробки Arduino IDE, модуль LuaNode32 з мікроконтролером ESP-32, датчик температури та вологості DHT22, OLED дисплей.

Результати. Виконано розробку програмного забезпечення підсистеми віддаленого моніторингу температури та вологості повітря, що складається з передавальної та приймальної частин бездротової сенсорної мережі та виступає прототипом реальної підсистеми контролю показників мікроклімату приміщень.

Висновки. Виконана розробка має практичну цінність, оскільки може бути інтегрована в систему опалення, вентиляції та кондиціонування повітря медичних приміщень.

Галузь використання – системи опалення, вентиляції, кондиціонування повітря, системи домашньої та офісної автоматизації.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 92 pages, 32 figures, 15 tables, 2 appendixes, 54 sources.

MEDICAL PREMISES, MICROCLIMATE, TEMPERATURE AND HUMIDITY SENSOR, WIRELESS SENSOR NETWORK, DATA TRANSMISSION MODULE, BLUETOOTH, ARDUINO IDE.

Object of study is the process of remote monitoring of microclimate indicators of medical premises.

Subject of study are software and hardware for remote monitoring of microclimate indicators of medical premises.

The goal of the work is the development of software for remote monitoring of microclimate indicators of medical premises based on wireless sensor network.

Materials, methods and tools: C ++ programming language, Arduino IDE development environment, LuaNode32 module with ESP-32 microcontroller, DHT22 temperature and humidity sensor, OLED display.

Results. The software for the subsystem of the temperature and humidity remote monitoring, has been developed, which consists of transmitting and receiving parts of the wireless sensor network and acts as a prototype of the real subsystem of control of indoor microclimate indicators.

Conclusions. The completed development has practical value, as it can be integrated into the system of heating, ventilation and air conditioning of medical premises.

The field of application are heating, ventilation, air conditioning systems, home and office automation systems.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Показники мікроклімату медичних приміщень та їх вплив на здоров'я людини	10
1.2 Стандарти кліматичних параметрів у медичних закладах України	13
1.3 Аналіз роботи систем забезпечення мікроклімату приміщень.....	14
1.4 Аналіз структурних та функціональних особливостей системи контролю мікроклімату приміщень.....	17
1.5 Висновки до розділу 1.....	21
2 Дослідження технологій організації бездротових сенсорних мереж.....	22
2.1 Огляд особливостей бездротових сенсорних мереж.....	22
2.2 Аналіз технологій бездротової передачі даних	25
2.3 Огляд існуючих систем ОКВП на основі бездротових сенсорних мереж.....	31
2.4 Висновки до розділу 2.....	33
3 Проєктування підсистеми віддаленого моніторингу показників мікроклімату.....	34
3.1 Аналіз вимог.....	34
3.2 Архітектура підсистеми.....	35
3.3 Вибір апаратних засобів для реалізації прототипу підсистеми.....	36
3.3.1 Вибір модуля передачі даних.....	37
3.3.2 Вибір датчиків температури та вологості.....	39
3.3.3 Вибір дисплею.....	42
3.4 Вибір середовища для розробки програмного забезпечення підсистеми.....	45
3.5 Висновки до розділу 3.....	48

4 Конструювання підсистеми віддаленого моніторингу показників мікроклімату.....	49
4.1 Алгоритм функціонування підсистеми.....	49
4.2 Опис програмної реалізації підсистеми.....	52
4.2.1 Вузол-передавач підсистеми.....	52
4.2.2 Вузол-приймач підсистеми.....	54
4.3 Висновки до розділу 4.....	57
5 Керівництво програміста.....	58
5.1 Призначення й умови застосування підсистеми.....	58
5.2 Характеристики програми.....	58
5.3 Звернення до програми.....	53
5.4 Вхідні та вихідні дані.....	64
5.5 Повідомлення програмісту.....	64
6 Керівництво користувача.....	66
6.1 Призначення програми.....	66
6.2 Умови виконання програми.....	66
6.3 Робота з програмою.....	66
Висновки.....	70
Перелік джерел посилання.....	71
Додаток А Текст програми.....	77
Додаток Б Слайди презентації.....	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ADC	– Analog to Digital Converter;
AFH	– Adaptive Frequency Hopping;
BLE	– Bluetooth Low Energy;
DAC	– Digital to Analog Converter;
IDE	– Integrated Development Environment;
IoT	– Internet of Things;
I2C	– Inter Integrated Circuits;
GPIO	– General-Purpose Input/Output;
LCD	– Liquid-Crystal Display;
MSb	– Most Significant Bit;
OLED	– Organic Light-Emitting Diode;
PID	– Proportional Integral Derivative;
PWM	– Pulse-Width Modulation;
SPI	– Serial Peripheral Interface;
TFT	– Thin-Film Transistor;
UART	– Universal Asynchronous Receiver-Transmitter;
UUID	– Universal Unique Identifier;
Wi-Fi	– Wireless Fidelity;
БСМ	– бездротова сенсорна мережа;
МК	– мікроконтролер;
ОВКП	– опалення, вентиляція, кондиціонування повітря;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер;
СКМП	– система контролю мікроклімату приміщення.

ВСТУП

Медичні установи, поліклініки, стаціонари, палати та особливо операційні – це приміщення, до мікроклімату яких завжди висувались особливі вимоги. Наприклад, від рівня температури та вологості в операційній залежить не тільки стан здоров'я пацієнта, але й самопочуття лікарів, які виконують надзвичайно відповідальну роботу. Параметри мікроклімату визначають теплообмін організму людини та істотно впливають на функціональний стан різних систем організму, самопочуття, працездатність і здоров'я, відповідно, недотримання існуючих норм показників мікроклімату може призвести до небажаних наслідків [1].

Тому, мікроклімат медичних приміщень повинен жорстко підтримуватися в рамках встановлених норм, що дозволяє:

- забезпечити комфортні та здорові умови для лікарів, медичних працівників, обслуговуючого персоналу та пацієнтів;
- створити оптимальний мікроклімат для профілактики появи та розвитку бактерій, мікробів та вірусів;
- забезпечити збереження меблів, стінових покриттів та обладнання, що використовуються в медичних установах [1].

На сьогоднішній день існує досить велика кількість систем контролю показників мікроклімату приміщень. Проте всі ці системи мають досить складний принцип налаштування та потребують залучення фахівців для їх використання та обслуговування. Також варто відмітити, що більшість таких систем мають досить високу енергоємність.

Тому, розробка енергоефективних рішень для віддаленого моніторингу показників мікроклімату медичних приміщень є актуальною задачею.

Метою роботи є розробка програмного забезпечення (ПЗ) для віддаленого моніторингу показників мікроклімату медичних приміщень на основі бездротової сенсорної мережі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Показники мікроклімату медичних приміщень та їх вплив на здоров'я людини

На самопочуття пацієнтів та працездатність медичного персоналу у закритих медичних приміщеннях впливає сукупність факторів: тепловий, повітряний, світловий, колірний, акустичний (рис. 1.1) [2].

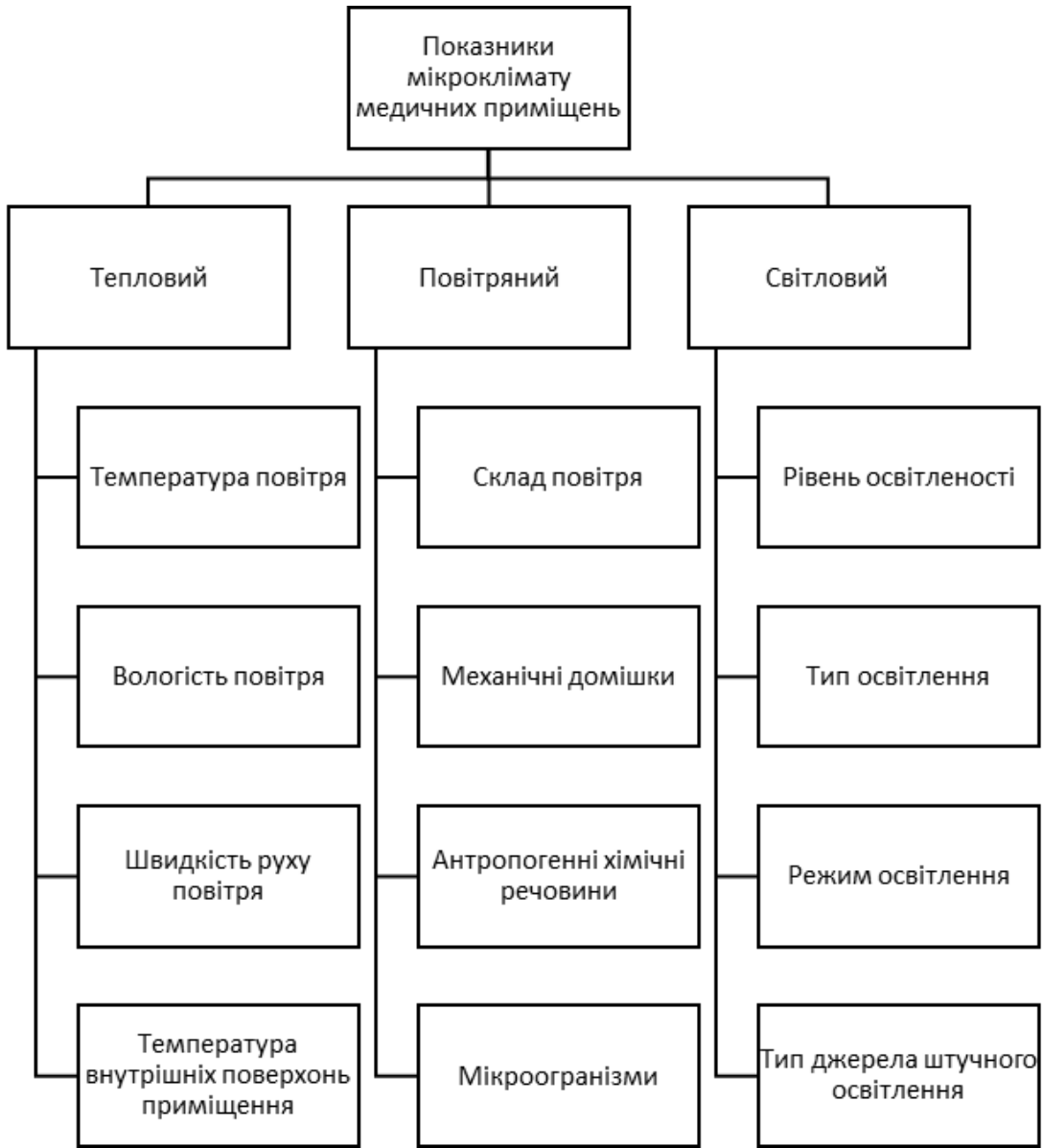


Рисунок 1.1 – Фактори, що впливають на самопочуття та працездатність людини [2]

Тепловий фактор – це сукупність чотирьох фізичних показників:

- температура повітря,
- вологість,
- швидкість руху повітря;
- температура внутрішніх поверхонь приміщення (стеля, стіни) [2].

Норми мікроклімату палат та інших приміщень лікарні повинні враховувати:

- вік хворого;
- особливості теплообміну хворих із різними захворюваннями;
- функціональне призначення приміщень;
- кліматичні особливості території [2].

Так, наприклад, температура більшості палат багатoproфільних лікарень складає 20°. Але, температура в палатах недоношених, новонароджених та немовлят має складати 25°, а у залах лікувальної фізкультури – 18° [2].

Оскільки доведено прямий вплив параметрів мікроклімату на людину, фахівці встановили норму кожного фізичного фактора, тобто оптимальні показники:

- відносна вологість повітря – 40-54%;
- швидкість руху повітряних потоків менше 0,2 м/с;
- температура повітря у теплий період – +18-22 °С;
- температура повітряних потоків у холодну пору року – +20-22 °С;
- допустиме насичення повітря токсичними речовинами – менше 0,8 [3].

Якщо параметри повітря всередині приміщення відповідають перерахованим показникам, знижується ризик розвитку захворювань суглобів, серцево-судинної системи, органів дихання. Відповідність кожного параметра мікроклімату існуючим нормам можна визначити тільки за допомогою спеціального обладнання [3].

Якщо людина знаходиться в умовах підвищеної або зниженої температури повітря, це шкодить організму, оскільки відбувається перегрів або переохолодження. У першому випадку виникають такі ознаки:

- головний біль, що часто повторюється;
- нудота, блювота;
- почервоніння обличчя;
- інтенсивне виділення поту;
- підвищення рівня тиску;
- слабкість;
- порушення координації рухів [3].

Перелічені ознаки свдчать про розвиток теплового удару, що може спричинити інсульт, інфаркт міокарда, зневоднення організму, гіповітаміноз, гіпоксію та інші порушення. Висока температура повітряних потоків усередині приміщення ускладнює достатнє насичення організму киснем. Його дефіцит призводить до порушення активності мозку, що виявляється у послабленні концентрації, пам'яті, промови, координації, мислення. Крім інших причин, перегрів тіла сприяє підвищенню густоти крові – вона збирається у тромби, тяжко циркулює усередині вен, капілярів, артерій [3].

Перебування в умовах низької температури повітря призводить до переохолодження. Цей стан небезпечний розвитком інфекційно-запальних хвороб, зниженням захисних властивостей організму. При регулярному переохолодженні, насамперед, страждають суглоби та нирки, що за відсутності лікування набуває хронічної форми, спричиняє настання інвалідності [3].

Рівень вологості повітря знижується за умови надмірно інтенсивного функціонування опалювальних приладів. Слизові оболонки носа пересихають, що стає причиною їхнього запалення. Інші наслідки недостатнього рівня вологості повітряних потоків:

- підвищення в'язкості крові, схильність до утворення небезпечних для життя тромбів;

- порушення функціонування нирок;
- уповільнення активності кишечника, проблеми з травленням їжі, набір ваги;
- збій у функціонуванні серця, схильність до розвитку ішемічної хвороби, інфаркту міокарда;
- погіршення загального самопочуття – підвищена стомлюваність, дратівливість, перепади настрою;
- зниження імунітету, схильність до розвитку різноманітних інфекційно-запальних захворювань [3].

Перелічені симптоми вказують на появу порушень у стані здоров'я і є підставою для негайного вживання заходів [3].

В свою чергу, високий рівень вологості повітряних потоків усередині приміщення сприяє розвитку захворювань дихальної та сечовидільної системи. Зростає ризик бронхіту та патологій нирок, які досить швидко приймають хронічний перебіг, шкодять здоров'ю, знижують якість життя [3].

Таким чином, основним методом забезпечення необхідних параметрів мікроклімату є застосування багатофункціональних систем опалювання, кондиціонування та вентиляції, разом з впровадженням систем контролю показників мікроклімату.

1.2 Стандарти кліматичних параметрів у медичних закладах України

Існують такі кліматичні стандарти у медичних закладах України [4]:

- системи опалення, вентиляції та кондиціонування повітря повинні забезпечувати оптимальні умови мікроклімату і повітряного середовища приміщень закладів охорони здоров'я, що надають первинну медичну (медико-санітарну) допомогу [4];

– приміщення закладів охорони здоров'я, що надають первинну медичну (медико-санітарну) допомогу, повинні бути обладнані системами припливно-витяжної вентиляції з механічним спонуканням і природною витяжкою без механічного спонукання. Робочі приміщення повинні мати умови для природної вентиляції, що здійснюється через кватирки, створи тощо, та мати автономний вентиляційний канал [4];

– розрахункову температуру та кратність повітрообміну приміщень наведено у таблиці 1.1 [4].

Таблиця 1.1 – Розрахункова температура та кратність повітрообміну приміщень [4]

Найменування приміщень	Розрахункова температура повітря, °C	Кратність повітрообміну за 1 годину		Кратність витяжки при природному повітрообміні
		приплив	витяжка	
Процедурний/ оглядовий кабінет / пункт щеплень	22	1,5	2	1,5
Реєстратура, довідкова кімната, вестибюль, гардеробна	18	-	1	1

1.3 Аналіз роботи систем забезпечення мікроклімату приміщень

Системи опалення, вентиляції, кондиціонування повітря (ОВКП) використовуються в різних типах будівель та забезпечують тепловий комфорт мешканців шляхом регулювання та зміни умов зовнішнього повітря до заданих значень. Залежно від зовнішніх умов, зовнішнє повітря втягується в будівлі і нагрівається або охолоджується, перш ніж розподілятися в зайняті приміщення, потім виводиться в навколишнє повітря або повторно використовується в системі [5].

Системи ОВКП можна класифікувати відповідно до необхідних основних процесів: нагрівання, охолодження та вентиляції, а також додаткових: зволоження та осушення. Для цього використовують відповідне обладнання ОВКП: системи опалення, кондиціонування, вентиляції та осушення повітря. Системи ОВКП потребують розподільної системи, щоб забезпечити необхідну кількість повітря з бажаними умовами навколишнього середовища. Система розподілу в основному залежить від типу холодоагенту та способу подачі, наприклад, обладнання для обробки повітря, фанкойли, повітропроводи та водопровідні труби [6].

Вибір системи залежить від трьох основних факторів: конфігурацію будівлі, кліматичні умови та бажання власника. Крім цього, можна розглянути додаткові критерії, такі як зміна клімату (наприклад, температура, вологість і тиск у приміщенні), потужність будівлі, просторові вимоги, вартість (наприклад капітальні витрати, експлуатаційні витрати та вартість обслуговування), аналіз життєвого циклу, а також надійність і гнучкість [6].

Також треба врахувати певні обмеження: доступну потужність відповідно до стандартів, конфігурацію будівлі, доступну площу, бюджет будівництва, доступне джерело комунальних послуг, навантаження на опалення та охолодження будівлі.

Основні компоненти або обладнання системи ОВКП [7]:

- змішана вентиляція і контроль зовнішнього повітря;
- повітряний фільтр;
- припливний вентилятор;
- витяжні або рельєфні вентилятори та випускний отвір;
- зовнішній забір повітря;
- протоки;
- кінцеві пристрої;
- система зворотного повітря;
- змійовики опалення та охолодження;

- автономний блок опалення або охолодження;
- бойлер;
- охолоджувач;
- контролер;
- охолоджувач води;
- обладнання для зволоження та осушення [7].

Основними різновидами систем ОВКП є центральна система та децентралізована або локальна система. Тип системи залежить від адресації основного розташування обладнання, яке має бути централізоване для кондиціонування всієї будівлі, як цілісний блок, або децентралізоване, як окреме кондиціонування певної зони, як частини будівлі. У таблиці 1.2 наведено порівняння централізованих і децентралізованих систем за критеріями відбору [7] - [8].

Основою для створення будь-яких систем ОВКП є чотири складові: основне обладнання, потреба в просторі, розподіл повітря та трубопроводи [8].

Таблиця 1.2 – Порівняння централізованих і децентралізованих систем ОВКП [7] - [9]

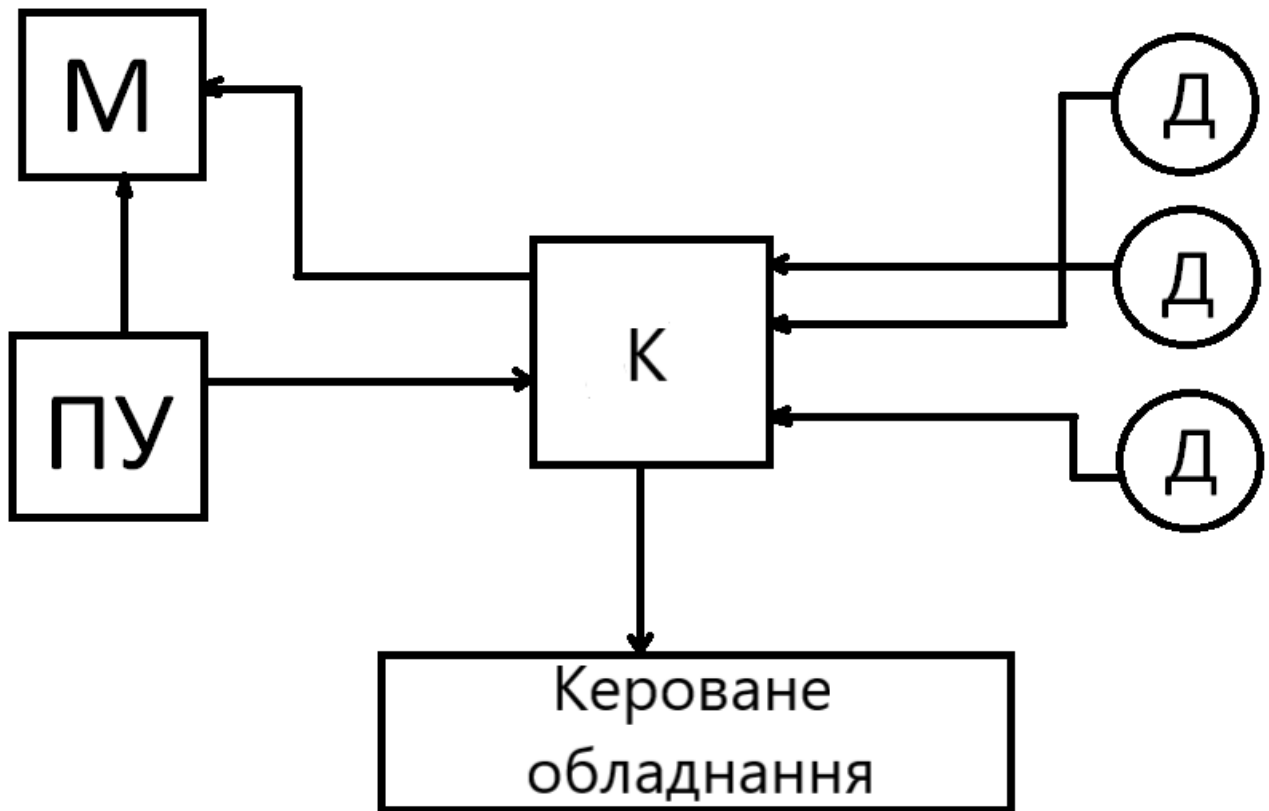
Критерій	Централізована система	Децентралізована система
1	2	3
Вимоги до потужності	Врахування факторів різноманітності ОВКП для зменшення потужності встановленого обладнання. Значні початкові та експлуатаційні витрати.	Для кожного обладнання необхідна максимальна потужність. Різноманітність розмірів обладнання обмежена
Надмірність	Резервне обладнання призначене для усунення несправностей та обслуговування	Немає резервного обладнання

Продовження таблиці 1.2

1	2	3
Спеціальні вимоги	Приміщення для обладнання розташоване за межами кондиціонованої зони або поруч із будівлею чи віддалено від неї. Установка додаткового обладнання для розподілу повітря та води, що вимагає додаткових витрат.	Додаткове приміщення для обладнання не потрібно. Обладнання може розташовуватися на даху та землі, що прилягає до будівлі.
Вартість обслуговування	Доступ до приміщення обладнання для обслуговування та збереження обладнання у відмінному стані, що дозволяє заощадити витрати на технічне обслуговування.	Доступ до обладнання, яке розташовується в підвалі або житловому приміщенні. Однак розташування на даху складнено через негоду
Надійність	Тривалий термін служби	Надійне обладнання, хоча передбачуваний термін служби обладнання може бути меншим
Гнучкість	Вибір резервного обладнання для забезпечення альтернативного джерела опалення, вентиляції та кондиціонування повітря або резервного	Розміщення в багатьох місцях, щоб бути більш гнучким.

1.4 Аналіз структурних та функціональних особливостей системи контролю мікроклімату приміщень

Система контролю мікроклімату приміщення (СКМП) – це система, що застосовується для підтримки заданих кліматичних характеристик, за допомогою управління системами вентиляції, опалення та кондиціонування повітря в автоматичному режимі. Сучасні системи клімат-контролю приміщення можуть працювати як самостійно, так і інтегруватися в інші системи. Загальну схему СКМП представлено на рисунку 1.2 [10].



М – монітор, К – контролер; Д – датчик; ПУ – пристрій управління.

Рисунок 1.2 – Структурна схема СКМП [10]

Головним пристроєм, що здійснює контроль за кліматичними показниками та характеристиками повітря, а також координує роботу всіх приладів опалення, кондиціювання, очищення та зволоження, подачі свіжого та витяжки відпрацьованого повітря є контролер, в якості якого може виступати мінікомп'ютер, тобто невеликий електронний блок з дисплеєм, запрограмований на створення оптимального мікроклімату [10].

Принцип дії клімат-контролю заснований на зборі спеціальними датчиками інформації про температурно-вологісні характеристики повітря по кожному приміщенню, аналізі даних та управлінні кліматичною технікою:

- управління опаленням здійснюється за допомогою сервоприводів, встановлених на опалювальних приладах та контурах;
- управління вентиляцією здійснюється за допомогою повітряних клапанів, встановлених на припливному та витяжному повітропроводі;

- управління витратою повітря на кінцевих вітках припливних повітропроводів здійснюється регуляторами витрати повітря;
- управління «теплою підлогою» здійснюється за допомогою термостатів та кліматичних контролерів [10].

Параметри і системи, які можуть самостійно контролюватися системою контролю клімату приміщення :

- температура повітря;
- рівень вологості приміщення;
- кондиціонування та переміщення повітряних мас;
- вентиляція [11].

Для здійснення управління мікрокліматом необхідно встановити певні датчики, які фіксуватимуть і передаватимуть сигнал в базовий модуль. Мінімальний набір для злагодженої роботи складається з датчиків руху, температури і вологості. Крім того будуть потрібні перемикачі та сенсорні панелі [11].

Перед запуском «розумної» кліматичної системи необхідно ретельно продумати проєкт і встановити в налаштуваннях необхідні параметри та показники. Можливість регулювання різної температури та вологості в приміщеннях має на увазі установку датчиків для кожної кліматичної зони. Інтелектуальна система клімат-контролю використовує закладені в неї алгоритми, подає команди на сервоприводи приладів або повітряні клапани системи вентиляції з системою охолодження повітря [11].

Термодатчик для системи кліматичного управління відстежує температуру всередині приміщення й зовні, регулюючи сприятливі умови для проживання в будь-який час року. Такі детектори можуть являти собою стандартний кабель або ж компактний пристрій з додатковими функціями. Датчики-кабелі зазвичай використовуються в дротовому вигляді системи управління кліматом [11].

Бездротові терморегулятори, на відміну від кабельних, часто здатні не тільки відслідковувати температуру в будинку, але і контролювати вологість повітря. Такий прилад активується за допомогою кнопки, яка зазвичай розташована на корпусі. Повітря з приміщення надходить через спеціальні отвори, після чого датчик аналізує показники. При покупці і наступному налаштуванні датчика слід звернути увагу на можливу похибку, яка виявляється при вимірюванні температури. Допустимі значення відхилення не більше - 0.5°C , вологості - не більше 3% [11].

У разі, якщо термостат не володіє можливістю моніторингу вологості повітря, доведеться витратитися на придбання датчиків вологості повітря, Модуль такого девайса працює на основі сенсора, а щохвилинне сканування даних відбувається за допомогою спеціального протоколу. Після збору даних, зняті показники відправляються в генеральний комп'ютер [11].

Для максимально ефективної роботи автоматизованої системи клімат-контролю знадобляться різні фільтри (брізери), зволожувачі повітря, клапани та заслінки, сучасні масляні або електричні радіатори, а також припливна система вентиляції з рекуператоро [11].

Таким чином, система кліматичного контролю дозволяє користувачам управляти всіма приладами за допомогою планшета, смартфона або пульта дистанційного керування. Сюди входить контроль роботи і управління різними приладами опалення, вентиляції та кондиціонування [11].

Клімат-контроль може бути як загальним, так і локальним. Мікроклімат для кожного приміщення може контролюватися окремо. Автоматичне підтримання температури та вологості може істотно відрізнятися. Ефективна робота кліматичного управління неможлива без якісної вентиляції повітря. Автоматика щохвилини ретельно перевіряє склад повітря. Багато систем автоматизованого клімат-контролю мають режим провітрювання за допомогою автоматичного відкривання вікон. Відсутність даної функції можна компенсувати системою припливної вентиляції з встановленими брізерами. Так

в приміщення буде регулярно надходити повітря, що пройшло повну очистку від пилу, бруду і алергенів [11].

Також для підтримки оптимального складу повітря, системи клімат-контролю використовують такі опції, як озонування та іонізацію повітря. Саме вони забезпечують насичення киснем в приміщенні та позбавляють від шкідливих речовин в складі повітря. Найчастіше ці функції включені в прилади-зволожувачі [11].

1.5 Висновки до розділу 1

Як показали проведені дослідження, системи автоматизації клімат-контролю надають наступні можливості: контроль та підтримання необхідної температури та вологості повітря у приміщеннях; очищення повітря від алергенів, пилу та шкідливих речовин; можливість підтримки індивідуального мікроклімату кожної локації.

Створення СКМП є складним проектом, що потребує значних коштів та часу. Тому, актуальною задачею є створення прототипу такої системи для удосконалення алгоритмів контролю, а також отримання ефективних +проектних рішень.

Метою роботи є розробка програмного забезпечення для віддаленого моніторингу показників мікроклімату медичних приміщень на основі бездротової сенсорної мережі.

Задачі роботи:

- дослідити існуючі технології організації бездротових сенсорних мереж;
- обрати апаратні та програмні засоби розробки;
- виконати програмну реалізацію та тестування підсистеми віддаленого моніторингу показників мікроклімату;
- розробити програмну документацію.

2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ОРГАНІЗАЦІЇ БЕЗДРОВОТИХ СЕНСОРНИХ МЕРЕЖ

2.1 Огляд особливостей бездротових сенсорних мереж

Бездротові сенсорні мережі (БСМ) перебувають у авангарді інноваційних проєктів і нових галузевих стандартів і, як наслідок, стають все більш поширеними у різних сферах. Наприклад, БСМ є провідною частиною систем Інтернету Речей (Internet of Things, IoT) і ключем до застосунків сфери охорони здоров'я, зокрема, таких як проєкти BT's Telecare та SAPHE (Smart & Aware Pervasive Healthcare Environment) [12].

Величезна кількість корисних даних може бути отримана, оброблена та зафіксована контролерами, наприклад відстань, вібрація, світло, вологість і температура за допомогою масиву мережевих вузлів датчиків, для подальшого збереження та аналізу. Як показано на рисунку 2.1, БСМ вже є поширені у більшості сфер через їх універсальність та відносно дешеве розгортання [12].

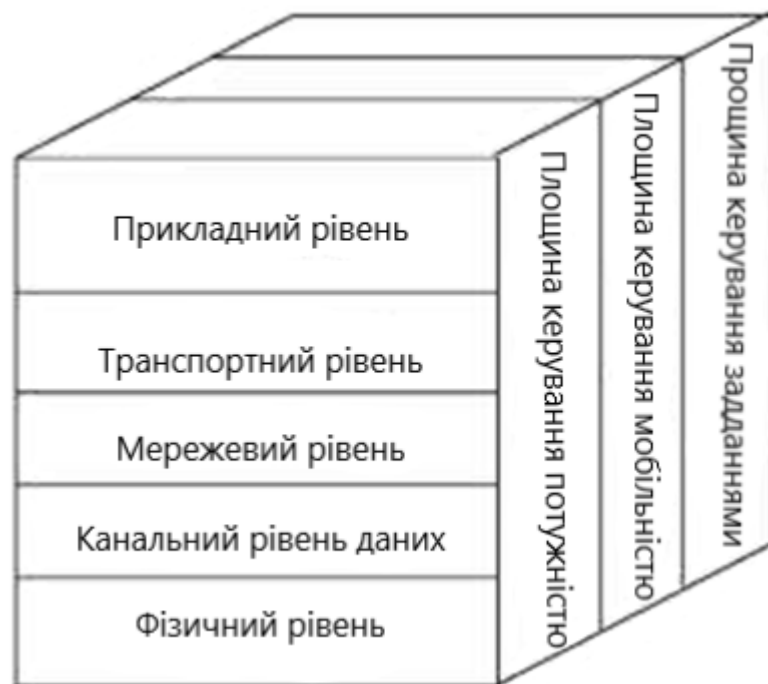


Рисунок 2.1 – Стек протоколів обробки даних датчиками [12]

Вузли в БСМ можуть бути оснащені різними типами датчиків, включаючи відеодатчик, аудіодатчик, інфрачервоний датчик, сейсмічний датчик, радарний датчик і магнітний датчик з низькою частотою дискретизації. На основі відповідних датчиків, БСМ здатна контролювати велику кількість параметрів навколишнього середовища, що включає в себе наступне:

- температура і вологість повітря;
- температура і вологість ґрунту;
- тиск повітря;
- стан освітлення;
- наявність, відсутність і рух конкретних об'єктів;
- параметр об'єктів, наприклад швидкість, напрямок і розмір.

Завдяки розробці цих нових бездротових датчиків, які можна використовувати в датчиках даних, обробці даних, зберіганні даних і бездротових мережевих з'єднаннях, бездротова сенсорна мережа може бути розгорнута в багатьох нових областях застосування, включаючи застосування для охорони здоров'я [12].

Бездротові сенсорні вузли, повинні функціонувати автономно, тому датчики мають бути оснащені ефективними модулями живлення, такими як сонячні батареї. Крім того, вузли бездротового датчика повинні ігнорувати перешкоди, які блокують пряму видимість бездротового датчика, і спілкуватися один з одним на певній відстані [12].

Бездротова сенсорна мережа може використовуватися в «розумних» середовищах та системах домашньої та офісної автоматизації. Ці самоорганізуючі бездротові сенсорні вузли повинні бути інтегровані з наявними вбудованими пристроями, забезпечувати зв'язок між користувачами для обміну інформацією та послугами, які вони пропонують, наприклад, друк і факс. В системах домашньої автоматизації розумні сенсорні вузли можуть бути вбудовані в побутову техніку та пристрої, наприклад, холодильники та мікрохвильові печі [12].

Застосунки БСМ, пов'язані зі здоров'ям, включають відстеження та моніторинг лікарів і пацієнтів у лікарні, прийом лікарських засобів та контроль у лікарнях, телевізійний моніторинг фізичних даних людини тощо. Встановлену бездротову сенсорну мережу можна використовувати для моніторингу та виявлення поведінки літніх людей та інвалідів, а зібрані протягом тривалого періоду часу фізичні та біологічні дані можуть використовуватися для подальшого аналізу лікарями [12].

Прикладом застосування БСМ в галузі охорони здоров'я є «розумний» дім Health, розгорнутий на медичному факультеті в Греноблі у Франції для перевірки такої системи. Через особливі характеристики БСМ, включаючи швидке розгортання, самоорганізацію мережі, високу відмовостійкість, високу чутливість тощо, БСМ мають широкий спектр застосування [12].

В комунікаційній архітектурі БСМ, кожен вузол здатен збирати дані та передавати їх за допомогою кількох кроків до приймачів та терміналів. Вузол-приймач може спілкуватися з кінцевими користувачами на персональних комп'ютерах (ПК) або мобільних пристроях через Інтернет. Архітектура зв'язку БСМ показана на рисунку 2.2 [12].

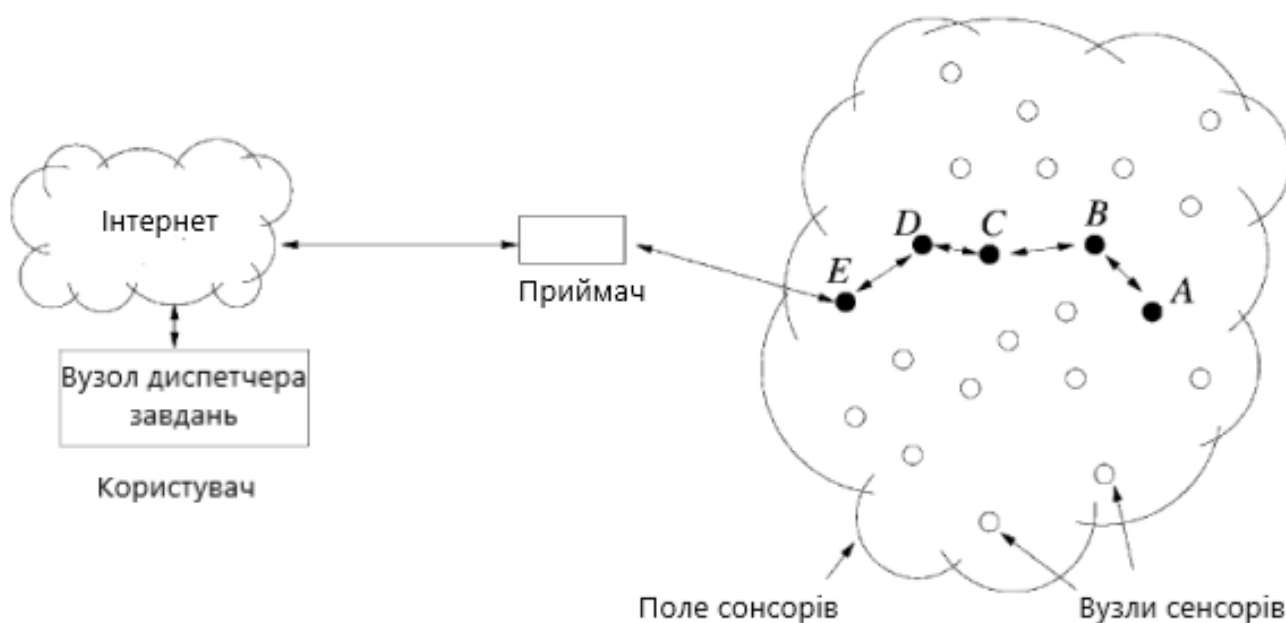


Рисунок 2.2 – Комунікаційна архітектура БСМ [12]

2.2 Аналіз технологій бездротової передачі даних

Технології бездротової передачі інформації застосовуються для передачі даних на певну відстань між декількома об'єктами. При цьому можуть використовуватися радіохвилі, лазерний, оптичний або інфрачервоний типи випромінювання. Існує багато протоколів, кожен з яких має певні особливості, які визначають область його застосування [13].

В таблиці 2.1 наведено результати порівняльного аналізу декількох найпопулярніших технологій Wi-Fi, Bluetooth, Zigbee [14].

Таблиця 2.1 – Порівняльна характеристика технологій бездротової передачі даних [14]

Параметр	Bluetooth	Zigbee	Wi-Fi
Діапазон частот	2,4 ГГц	868/915 МГц, 2,4 ГГц	2,4 ГГц, 5 ГГц
Максимальна швидкість сигналу	1 Мб/сек	250 Кб/сек	54 Мб/сек
Номінальна відстань	10 м	10-100 м	100 м
Пропускна здатність каналу	1 МГц	0,3-0,6 МГц, 2 МГц	22 МГц
Кількість радіочастотних каналів	79	1/10, 16	14 (2,4 ГГц)
Максимальна кількість вузлів	8	>65000	2007

Wi-Fi — це технологія, заснована на наборі стандартів IEEE 802.11, яка використовує радіочастоти (RF) для розширення дротових локальних мереж (LAN) на основі Ethernet на пристрої з підтримкою Wi-Fi, дозволяючи пристроям отримувати та надсилати інформацію з Інтернет. Wi-Fi використовує

Інтернет-протокол (IP) для зв'язку між кінцевими пристроями та локальною мережею. З'єднання Wi-Fi встановлюється за допомогою бездротового маршрутизатора, який під'єднано до мережі та дозволяє пристроям отримувати доступ до Інтернету [15].

Одним з недоліків Wi-Fi є те, що він може бути схильним до перешкод залежно від радіочастотного середовища, в якому він працює. Перешкодами можуть бути інші сигнали Wi-Fi, радіохвилі, які випромінюють мікрохвильові печі, цементні стіни. Але, Wi-Fi може транслювати на двох частотах (4 ГГц і 5 ГГц), допомагаючи своєму сигналу подолати весь шум і передати швидкий і сильний сигнал від бездротового маршрутизатора до пристрою [15].

Протокол Wi-Fi рекомендований для наступних застосувань: відео, електронна пошта та вебпрограми локальної мережі, які вимагають підключення до мережі з більш високою швидкістю передачі даних (1 Мбіт/с-1 Гбіт/с) [15].

У таблиці 2.2 наведено основні технічні характеристики стандартів IEEE 802.11 a, b, g і n. [13].

Таблиця 2.2 – Порівняльна характеристика стандартів IEEE 802.11 a, b, g, n [13]

Стандарт	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n
1	2	3	4	5
Частотний діапазон	5.15-5.25 5.67-5.58	2.4-2.483	2.4-2.483	2.4-2.483 5.15-5.25 5.67-5.58
Доступ до радіоканалу	CSMA-CA	CSMA-CA	CSMA-CA	CSMA-CA
Кількість абонентів на один канал	64	64	64	64

Продовження таблиці 2.2

1	2	3	4	5
Максимальна швидкість обміну даними	54 Мбіт/с	11 Мбіт/с	54 Мбіт/с	480 Мбіт/с
Метод модуляції	OFDM	BPSK,CCK	OFDM	BPSK,QPSK 16-QAM 64QAM
Дальність дії в приміщенні	10-20	20-100	20-50	10-20

Bluetooth – це стандарт для бездротового з'єднання мобільних телефонів, комп'ютерів та інших електронних пристроїв на малому радіусі дії, тобто глобальна персональна мережа 2,4 ГГц для бездротового зв'язку на невеликій відстані. Bluetooth надсилає та приймає радіохвилі в діапазоні 79 різних частот (каналів) з центром на частоті 2,45 ГГц, окремо від радіо, телебачення та мобільних телефонів, зарезервованих для використання промисловими, науковими та медичними гаджетами. Передавачі Bluetooth малої дії мають дуже низьке енергоспоживання і є більш безпечними, ніж бездротові мережі, які працюють на більших діапазонах, наприклад Wi-Fi [15].

Bluetooth найкраще використовувати для передачі даних з пристрою на пристрій, облікових даних мобільного телефону, крім того, бездротові динаміки та бездротові гарнітури часто підключаються за допомогою Bluetooth [15].

Ця технологія передбачає можливість використання з'єднання типу «точка - точка» або «точка – декілька точок». На основі декількох пристроїв, один з яких працює як основний (M - Master), а інші як залежні (S - Slave), створюється пікомережа (piconet). Декілька взаємодіючих пікомереж є основою для побудови «розгалуженої мережі» (scatternet). Типи пікомереж показано на рисунку 2.3 [13].

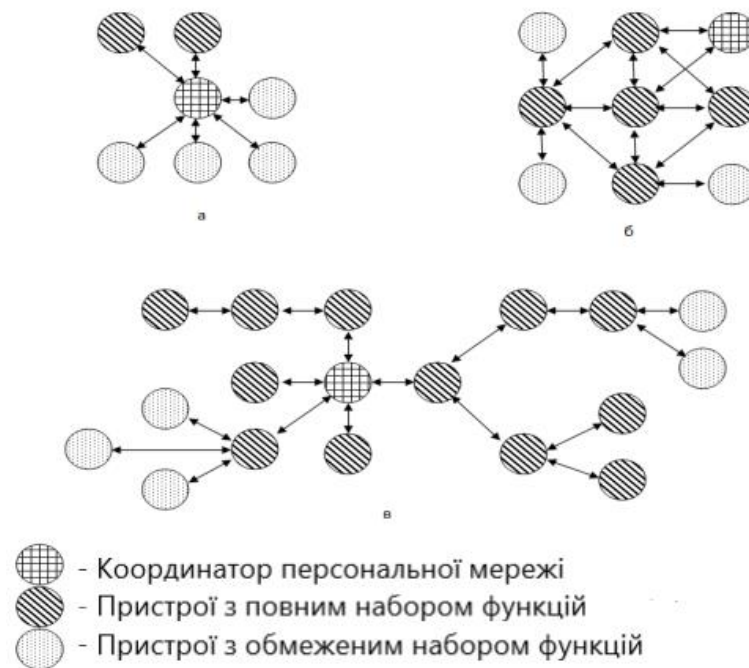
ZigBee – це 2,4 ГГц протокол локальної мережі (LAN). Він був розроблений як специфікація на основі IEEE 802.15.4 для набору протоколів зв'язку високого рівня, що використовуються для створення персональних мереж з невеликими цифровими радіоприймачами малої потужності. Пристрої ZigBee передають дані на великі відстані, передаючи їх через Mesh мережу проміжних пристроїв для досягнення більш віддалених. Мережі ZigBee мають визначену швидкість 250 кбіт/с і захищені 128-бітними симетричними ключами шифрування. ZigBee зазвичай використовується в програмах з низькою швидкістю передачі даних, які вимагають високої масштабованості, тривалого терміну служби акумулятора та безпечної мережі. Ця технологія простіше і дешевше, ніж Bluetooth або Wi-Fi, і зазвичай використовується для дому, будівель та промислової автоматизації, таких як контрольоване освітлення та термостати, монітори домашньої енергії, інтелектуальні вимірювання, збір даних медичних пристроїв, системи керування трафіком та інші потреби малої потужності та низької пропускної здатності [15].

Технічні характеристики ZigBee наведено в таблиці 2.3. На рисунку 2.5 наведено структуру ZigBee [13].

Таблиця 2.3 – Технічні характеристики ZigBee [13]

Діапазон частот	Кількість каналів	Швидкість передачі даних	Тип модуляції
868-870 МГц	1	20 Кбіт/с	BPSK
902-928 МГц	10	40 Кбіт/с	BPSK
2.4-2.4835 МГц	16	250 Кбіт/с	0-QSK

Як показав аналіз технологій бездротової передачі даних, задача їх вибору є досить складною навіть для досвідчених розробників.



а – деревоподібна топологія мережі; б,в – топологія Mesh

Рисунок 2.5 – Топологія мережі ZigBee [13]

Порівняння технологій Bluetooth, Wi-Fi і ZigBee на основі табл. 2.4 дозволило прийняти рішення про вибір технології Bluetooth, оскільки вона дозволяє ефективно передавати дані з пристрою на пристрій, створювати розгалуджену мережу, завдяки технології AFH уникати колізій та забезпечувати надійну роботи пристроїв мережі, а завдяки технології Bluetooth Low Energy (BLE) економити електроенергію [13] - [15].

Таблиця 2.4 – Основні критерії вибору типу технології [15]

Технологія	Застосування	Переваги	Швидкість передачі даних	Відстань
Wi-Fi	LAN, Internet	Швидкість, гнучкість	1-7 Гб/сек	100 м
Bluetooth	PAN, Мобільні дані	Вартість, зручність	48 Мб/сек	<100 м
ZigBee	Сенсорні мережі	Надійність, потужність, масштабованість, вартість	250 Мб/сек	70-300 м

2.3 Огляд існуючих систем ОВКП на основі бездротових сенсорних мереж

Традиційні системи ОВКП досягають бажаного рівня контролю за показниками мікроклімату за допомогою простого керування типу вкл/викл, що часто призводить до суттєвих витрат енергії. Потенційним рішенням цієї проблеми є інтелектуальні саморегуючі контролери, які базують свої дії/рішення на даних датчиків. Така підсистема складається з різних бездротових сенсорних вузлів і вузла керування (сервера). Сенсорні вузли надають потрібні дані, а вузол керування виконує алгоритм, який вирішує, активувати охолодження чи опалення на основі даних датчика. Така підсистема може бути інтегрована в системи ОВКП в медичних закладах [16].

Існує кілька комерційних систем, розроблених такими компаніями, як Honeywell, Toshiba, Hitachi та Mitsubishi, які пропонують різноманітні датчики та контролери [17]–[19]. Ці системи, як правило, дорогі та часто використовують власні бездротові протоколи, а не популярні бездротові стандарти, такі як наприклад ZigBee (IEEE 802.15.4) [20]. Крім того, більшість цих контролерів мають лише локальний дисплей, який надає людино-машинний інтерфейс, але не забезпечує можливість віддаленого доступу/налаштування контролера (наприклад, через вебінтерфейс). Передові системи [21]–[25] використовують PID (Proportional Integral Derivative) контролери, які спрямовані на мінімізацію помилок у циклі нагрівання/охолодження, але при цьому жертвують простотою встановлення та використання. Для компенсації змінних умов навантаження можна реалізувати самонастроювані PID-регулятори [26]. Цей підхід допомагає підвищити продуктивність, але все ще представляє проблему, оскільки важко отримати відповідні моделі для систем ОВКП. Таким чином, інтелектуальний контролер, який може самооптимізуватися за допомогою кількох датчиків, потенційно є життєздатним рішенням проблеми неефективності ОВКП.

Бездротова сенсорна система може бути інтегрована в систему ОВКП будівлі, як показано на рисунку 2.6 [27].

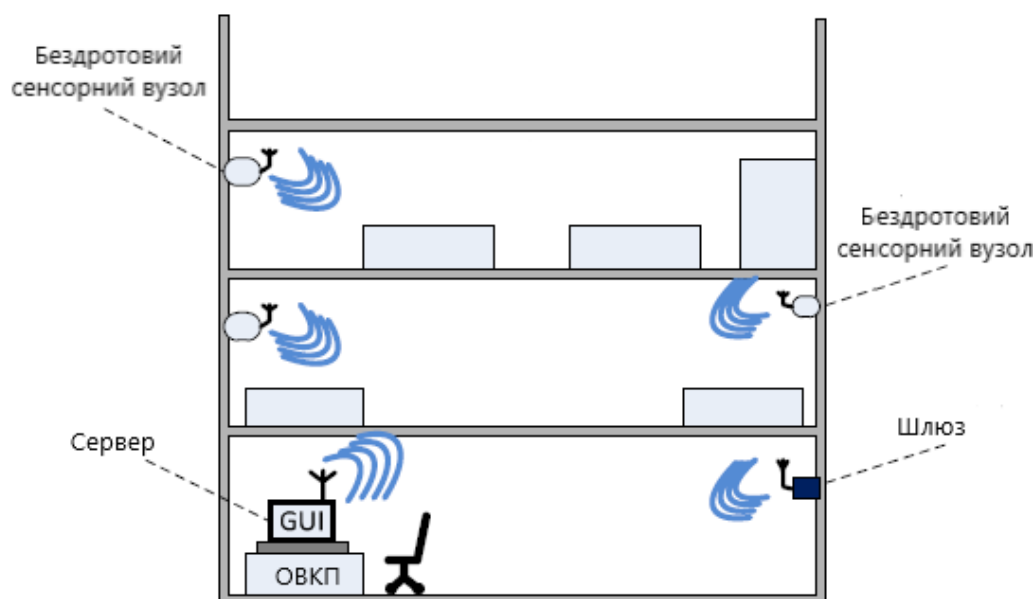


Рисунок 2.6 – Бездротова сенсорна мережа для системи ОВКП [27]

Ця система враховує дані датчика (тобто температура, вологість, освітленість, присутність та ін.) від різних датчиків, розміщених у стратегічних місцях у приміщенні, і вирішує, чи активувати охолодження чи опалення на основі даних датчика. Датчики повідомляють свої дані в режимі реального часу на керуючий вузол, відповідальний за виконання алгоритму керування для виконання певної дії.

Подібні системи, які також спираються на бездротові сенсорні мережі, описано в [28]–[30]. Система комфорту на основі стандарту IEEE 1451.2 представлена в [31]. Вона використовує електрохімічні та напівпровідникові датчики для визначення температури, вологості, окису вуглецю та вуглекислого газу. Однак система не надає інтерфейсу для мобільного користувача і не враховує рух, світло чи шум, які є параметрами, які можна використовувати для підвищення надійності виявлення присутності. Системи на основі стандарту IEEE 1451.2 також представлені в роботах [32] - [33]. У [18] була змодельована, але не реалізована система теплового комфорту, спрямована на застосування

систем ОВКП. У роботі [34] було розглянуто три параметри для контролю комфорту. Бездротова система моніторингу летючих органічних сполук (ЛОС) на основі ZigBee представлена в [35]. ЛОС виділяються у вигляді газів для певних рідин і твердих речовин (наприклад, засоби для зняття фарби, пестициди тощо), і тому дуже важливо контролювати ці гази в приміщенні для комфорту та здоров'я. Система керування, що представлена в [16], намагається досягти більшої енергоефективності, ніж звичайні системи ОВКП, використовуючи бездротові датчики. Ці сенсорні вузли контролюють навколишнє середовище та надають дані в реальному часі через бездротове з'єднання до вузла керування, який потім виконує відповідні дії, швидко реагуючи на зміни в навколишньому середовищі. Що стосується бездротового підключення системи, важливо також забезпечити наявність достатньо високоякісних каналів для забезпечення гарного зв'язку [36]. Система [16] була розроблена для зниження споживання електроенергії та втрат за рахунок підвищення ефективності системи ОВКП. Це досягається завдяки використанню мережі бездротових датчиків, які використовуються для вимірювання температури (та інших відповідних параметрів) у різних точках і визначення зайнятості приміщення. Використовуючи ці дані, можна підвищити ефективність стандартних систем ОВКП, застосовуючи кращі керуючі сигнали замість простого керування типу включення/виключення.

2.4 Висновки до розділу 2

Таким чином, основна увага розробників може бути зосереджена не на реалізації повної системи ОВКП, а підсистеми віддаленого моніторингу на основі БСМ. Ця підсистема має взаємодіяти з системою ОВКП та сервером, як контролер цієї системи. Інтеграція такої підсистеми в систему ОВКП дозволить підвищити ефективність ОВКП та зменшити рахунки за електроенергію.

3 ПРОЄКТУВАННЯ ПІДСИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ПОКАЗНИКІВ МІКРОКЛІМАТУ

3.1 Аналіз вимог

Підсистема віддаленого моніторингу показників мікроклімату медичних приміщень призначена для збору даних з вузлів БСМ та їх передачі на сервер, де реалізовано алгоритми управління системою ОВКП.

Розроблювана підсистема має забезпечувати наступні функції:

- коректне зчитування даних щодо температури та вологості з датчиків та візуалізація інформації;
- можливість підключення додаткових вузлів до БСМ та додаткових датчиків до кожного вузла, наприклад датчиків присутності;
- перевірка підключення бездротових сенсорних вузлів до БСМ та сповіщення користувача про помилку підключення конкретного вузла;
- здатність працювати в автономному режимі в разі відсутності зв'язку з сервером.

Нефункціональні вимоги:

- просте підключення та налаштування підсистеми;
- легка заміна компонентів для підтримки підсистеми в робочому стані;
- можливість інтеграції в систему ОВКП;
- достатня компактність підсистеми для встановлення в існуючу інфраструктуру системи ОВКП;
- універсальність, можливість застосування у інших системах, наприклад, домашньої та офісної автоматизації;
- мінімальна вартість підсистеми;
- низьке енергоспоживання.

3.2 Архітектура підсистеми

Архітектура підсистеми об'єднує бездротові сенсорні вузли (БСВ 1, БСВ 2, ..., БСВ N) з апаратними та програмними рішеннями для віддаленого моніторингу показників мікроклімату медичних приміщень, сервер для зберігання даних і клієнта, який може змінювати налаштування вузлів, а також переглядати дані, які повідомляє кожен вузол (рис. 3.1).

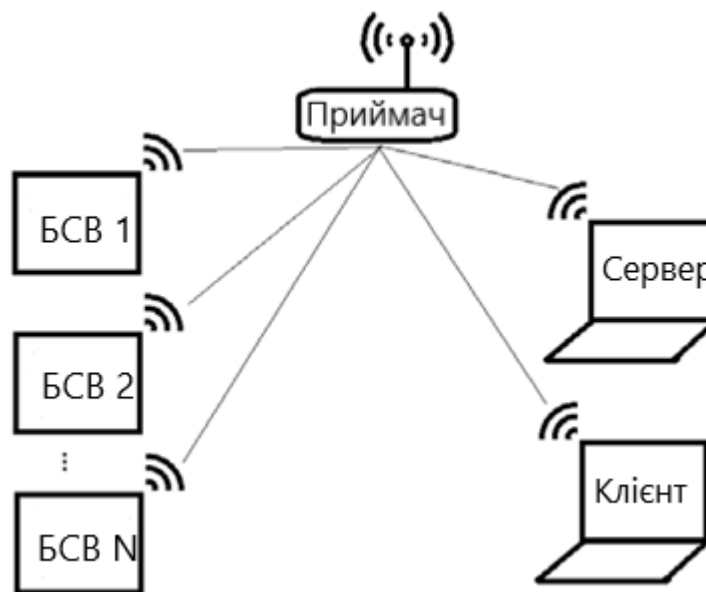


Рисунок 3.1 – Узагальнена архітектура підсистеми віддаленого моніторингу

Дані про температуру та вологість, а також додаткові відомості про присутність людей у приміщенні збираються з датчиків, приєднаних до вузлів-передавачів та відправляються на вузол-приймач, який отримує та відображає дані в текстовому та графічному вигляді.

Кожен вузол складається з модуля передачі даних, датчика температури та вологості, а також може містити додатково датчики присутності (рис. 3.2). Модуль передачі даних забезпечує зв'язок з вузлом-приймачем. Кожен БСВ має один модуль передачі даних.

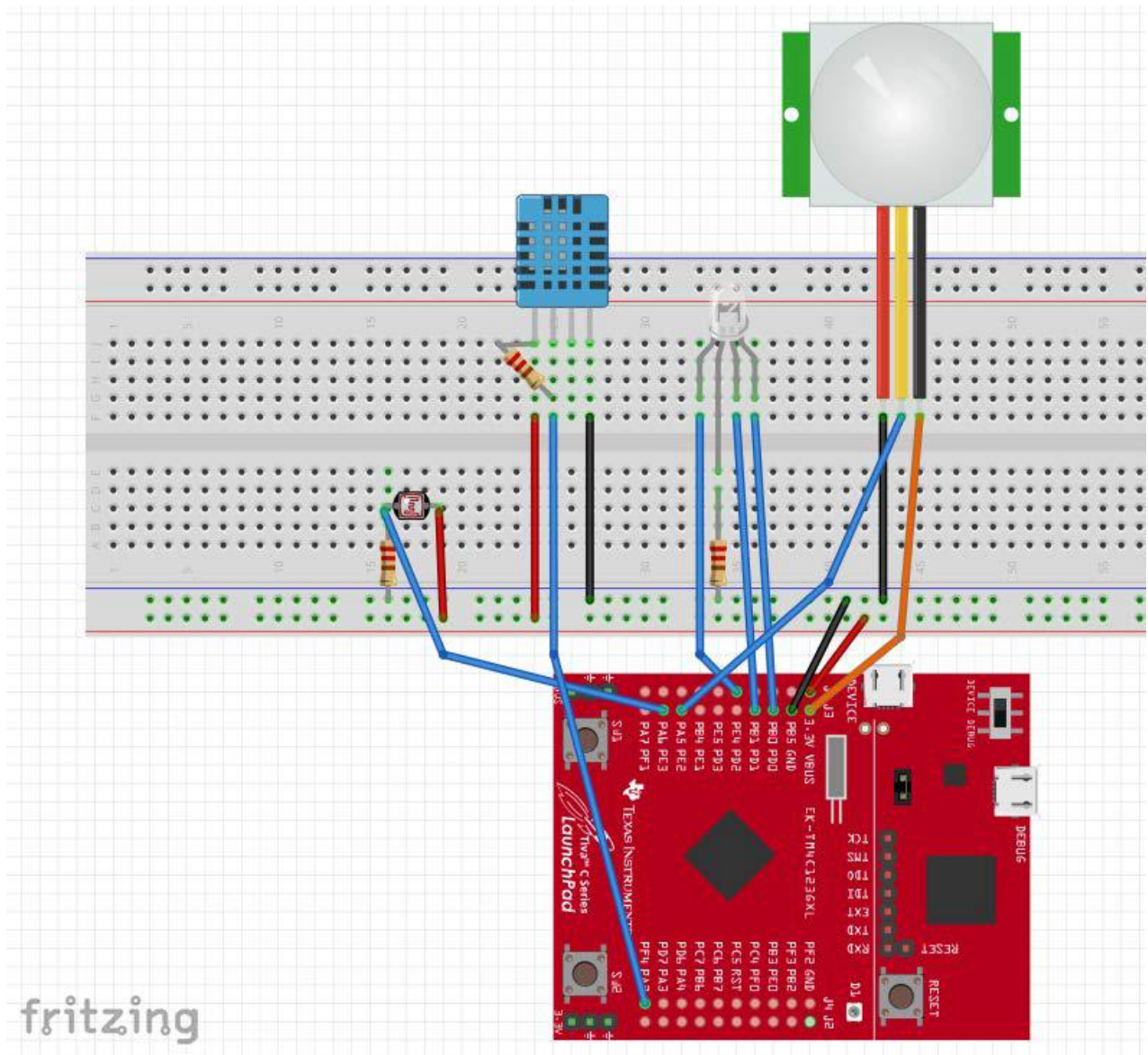


Рисунок 3.2 – Схема підключення групи датчиків

3.3 Вибір апаратних засобів для реалізації прототипу підсистеми

В якості апаратних засобів підсистеми виступають модулі бездротової передачі даних, датчики температури та вологості, а також дисплей для відображення даних моніторингу показників мікроклімату медичних приміщень.

3.3.1 Вибір модуля передачі даних

Існує багато модулів передачі даних, але більшість із них мають певні недоліки, пов'язані з розміром, продуктивністю та ціною. Тому, вибір модуля передачі даних здійснювався між доступними модулями ESP32 та ESP8266, а результати їх порівняння наведено в таблиці 3.1 [37].

Аналіз даних таблиці показує, що ESP32 - це потужна система з можливостями передачі даних по Wi-Fi і Bluetooth і високоінтегрованою структурою, що працює від двоядерного мікропроцесора Tensilica Xtensa LX6.

Таблиця 3.1 – Порівняльний аналіз модулів ESP32 та ESP8266 [37]

Характеристика	ESP32	ESP8266
Процесор	Tensilica Xtensa LX6 32 bit Dual-Core at 160/240 MHz	Tensilica LX106 32 bit at 80 MHz (up to 160 MHz)
Статична ОП	512 Кб	160 Кб
Флеш-пам'ять	2Мб (max 64Мб)	4Мб (max 16Мб)
Напруга	2.2 В - 3.6 В	3.0 В - 3.6 В
Робочий струм	80 мА	80 мА
Програмованість	C, C++, Lua та ін.	C, C++, Lua та ін.
Відкритість	+	+
Wi-Fi	802.11 b/g/n	802.11 b/g/n
Bluetooth®	4.2 BR/EDR + BLE	-
UART	3	2
GPIO	32	17
SPI	4	2
I ² C	2	1
ІІМ	8	-
АЦП	18 (12-bit)	1 (10-bit)
ЦАП	2 (8-bit)	-
Розмір	25.5 x 18.0 x 2.8 мм	24.0 x 16.0 x 3.0 мм
Вартість, грн	214	160

Завдяки невеликому розміру та опублікованій у відкритому доступі схемі модуля ESP32, його легко інтегрувати в проєктовану підсистему та створити компактний пристрій. Операційна система мікроконтролера (МК) FreeRTOS - це програмне забезпечення з відкритим вихідним кодом, яке забезпечує чудову підтримку додатків у реальному часі [37].

Таким чином, ESP32 є відмінним варіантом завдяки покращеним характеристикам, збільшеному функціоналу і великому об'єму пам'яті. Важливою перевагою ESP32 є більш швидкий Wi-Fi та Bluetooth. У ESP32 встановлений більш потужний процесор, що дозволяє реалізовувати складні проєкти. Цей модуль підходить для застосунків, у яких потрібен Інтернет або нові інтерфейси. Об'єм пам'яті у ESP32 збільшений – 512 Кб проти 160 Кб ESP8266. Також ESP32 відрізняється великою кількістю виводів GPIO. До декількох контактів на ESP32 прикріплені ємнісні сенсорні датчики та датчики температури. ESP32 має 18 12-бітних каналів АЦП, а у його попередника є всього один 10-бітний вивід АЦП. Потужність процесора значно впливає на швидкість роботи. Модуль ESP32 демонструє рекордну продуктивність порівняно з попередником ESP8266. Завантаження сторінок з довгим скетчем і безліччю графіків займає секунди.

На рисунку 3.3 представлено модуль LuaNode32 на основі ESP32 [38].



Рисунок 3.3 – Wi-Fi модуль LuaNode32 з ESP-32 [38]

Цей модуль є еволюцією плати Node Lua ESP8266, заснованої на ESP-IDF/ESP8266_RTOS_SDK, для ESP32 і сумісний з ESP8266. У нього вже є вбудована антена, підсилювач потужності, малопотужні підсилювачі, фільтри, модуль керування живленням і USB-Serial перетворювач. Все рішення займає мінімальну площу друкованої плати. Ця плата використовується з двоядерними чіпами Wi-Fi і Bluetooth з частотою 2,4 ГГц за технологією TSMC 40 нм з низьким енергоспоживанням, найкращими властивостями за потужністю та радіочастотами, що є безпечним, надійним і масштабованим для різноманітних застосувань [39]. Основні переваги:

- невеликий обсяг для легкого вбудовування в інші продукти;
- потужна функція з підтримкою протоколу LWIP, Freertos;
- Wi-Fi підтримка трьох режимів: AP, STA та AP+STA;
- підтримка програмування на Lua, простота розробки [39].

3.3.2 Вибір датчиків температури та вологості

При виборі датчика температури та вологості треба врахувати функціональність, точність, вартість. В роботі розглядалися декілька датчиків, порівняльну характеристику яких наведено в таблиці 3.2, зокрема датчики DHT11 / DHT22. Дослідження показало, що популярні в сфері IoT датчики серії DHT11 мають нижчу точність/діапазон, але й найнижчу вартість [40] – [43].

Таблиця 3.2 – Порівняння датчиків температури та вологості [40] – [43]

Характеристика	DHT11	DHT22	AM2320	AM2301
1	2	34	4	5
Вигляд				

Продовження таблиці 3.2

1	2	3	4	5
Діапазон визначення вологості	20-90% ± 5% RH	0-100% ± 2% RH	0-100% ± 2% RH	0-100% ± 2% RH
Діапазон визначення температури	0-50 °C ± 2%	- 40 ~ 80 ± 0.5%	- 40 ~ 80 ± 0.5%	- 40 ~ 80 ± 0.5%
Фізичний розмір (мм)	15.5 x 12 x 5.5	14 x 18 x 5.5	12 x 15 x 5.5	59 x 27 x 14
Кількість виводів	4	4	4	3
Живлення (В)	3.5-5.5	3.6-6	3.6-6	3.3-5
Наявність обв'язки	часткова	часткова	часткова	повна
Особливості	-	-	-	наявність захисного корпусу
Вартість (грн)	29	105	64	110

DHT11 / DHT22 – популярні датчики для визначення відносної вологості і температури, що складаються з ємнісного датчика вологості та термістора. Датчик містить в собі АЦП для перетворення аналогових значень вологості і температури [44].

Ці цифрові датчики засновані на протоколі, який для зв'язку використовує один дріт/шину з відкритим колектором, тому обов'язкове використання підтягуючих резисторів 5-10 КОм. Основна відмінність між DHT11 і DHT22 полягає в тому, що DHT22 має більш високу роздільну здатність і є більш точним в порівнянні з DHT11. Позначення пінів DHT22 і DHT11 зображено на рисунку 3.4 [44]. В таблицях 3.3, 3.4 наведено деякі важливі характеристики [44].

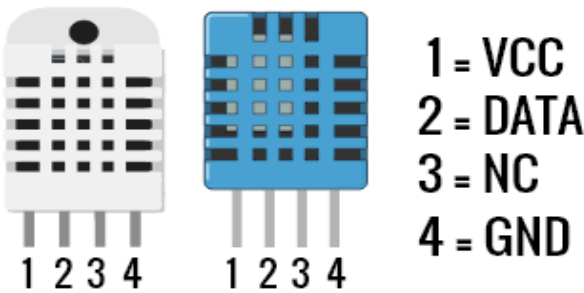


Рисунок 3.7 – Позначення пінів DHT22 і DHT11 [44]

Таблиця 3.3 – Характеристика пінів DHT22 і DHT11 [44]

Назва пінів DHT22 і DHT11	Характеристика
1 VCC	3-5 В живлення
2 DATA	Вивід даних
3 NC	Не використовується
4 GND	Загальний

Таблиця 3.4 – Технічні характеристики DHT22 і DHT11 [44]

Параметри		DHT22	DHT11
Живлення, В	мін	3	3
	макс	5.5	5.5
Споживаний струм, мА	мін	0.1	0.05
	макс	2.5	2.5
Вимірювання вологості	мін	20%	0%
	макс	80-95%	100%
	точність	5%	2-5%
Вимірювання температури	мін	0 С	-40 С
	макс	+50 С	+80 С
	точність	± 2С	± 0.5 С
Частота вимірювань		1 Гц	0,5 Гц
Роздільна здатність		8 біт	16 біт

3.3.3 Вибір дисплею

Як показали проведені дослідження, основними різновидами дисплеїв для проєктів на основі апаратно-програмних платформ та модулів є наступні сегментні та рідкокристалічні [45].

LCD (liquid-crystal display) - це звичайний монохромний дисплей, який дозволяє виводити символи певного обсягу та розміру. Прикладом може служити lcd-дисплей arduino 1602. Екран може відображати два рядки по 16 символів кожен. Зазвичай цього достатньо для відображення інформації в нескладному проєкті на Arduino. Є шилд LCD 1602 keypad shield, що дозволяє отримати екран з кнопками керування [45].

OLED (organic light-emitting diode) - це той самий TFT-екран лише замість рідких кристалів використовуються світлодіоди, що дає великий кут огляду, гарну картинку та менше споживання енергії. Однак їх ціна на достатньо висока [45].

TFT (thin-film transistor) - це графічний дисплей, який дозволяє відображати не тільки символи, а й зображення. Якість зображення залежить від роздільної здатності екрана. TFT екрани коштують дорожче, особливо сенсорні, але результат набагато краще. Дисплеї Arduino LCD TFT відмінно підходять для проєктів зі побудовою графіки. Є варіанти шилдів з можливістю підключення SD - картки. Сенсорні варіанти надають можливості для реалізації кнопок на одному шилді з екраном, що в свою чергу дозволяє економити на цифрових виходах [45].

OLED - це монохромний дисплей, що забезпечує зручність читання завдяки високій контрастності OLED дисплея. Кожен OLED екран виготовлено з окремих 128x64 або 128x32 білих органічних світлодіодів. Кожен світлодіод включається / виключається за допомогою чіпа управління. Оскільки сам дисплей є джерелом світла, додаткове підсвічування не потрібно. Це забезпечує

високу контрастність дисплея та значно знижує енерговитрати, необхідні для його живлення [46].

Приклад OLED дисплею зображено на рисунку 3.8 [47].



Рисунок 3.8 – OLED дисплей 0.96" I2C 128x64 [47]

Як драйвер в дисплеї використовується чіп SSD1306, який дозволяє передавати дані по протоколам I²C, SPI і 8-бітному паралельному протоколу. Зазвичай, такий широкий вибір інтерфейсів надає дисплей з роздільною здатністю 128x64 пікселів. У OLED дисплеях з роздільною здатністю 128x32, доступний тільки протокол SPI, що є універсальним і вимагає малої кількості контактів [46].

OLED дисплею і драйверу необхідне живлення 3.3 В і 3.3 В для живлення логіки та передачі даних. Живлення дисплеїв залежить від того, як саме завантажений екран, але в основному він споживає близько 20 мА і вимагає напруги живлення 3.3 В. Вбудований драйвер представляє собою простий перемикач, який перетворює 3.3 В - 5 В в керуючий сигнал для OLED дисплея. Дисплей можна запускати від одного джерела живлення 3.3 В або використовувати 3.3 В для живлення чіпа і до 4.5 В для самого екрану. Або

можно підключити 3.3 В до чіпу і подати 7-9 В безпосередньо на контакт живлення OLED дисплея [46].

В якості іншого варіанту було розглянуто використання кольорового дисплею для Arduino Mega 2560 - 3.5" 320x480 TFT LCD (ASC201), що зображений на рисунку 3.9 [48].

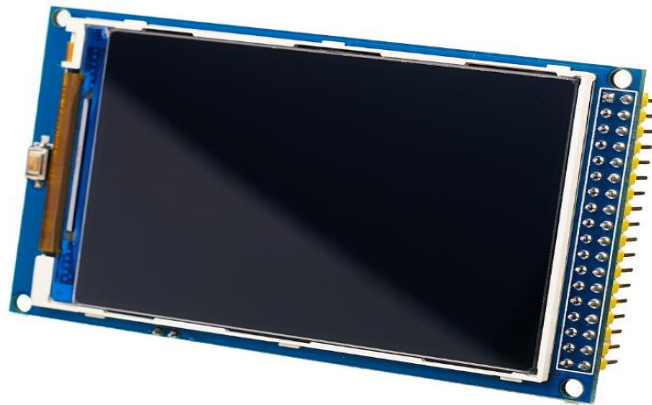


Рисунок 3.9 – Дисплей 3.5" 320x480 TFT LCD(ASC201) кольоровий [48]

Це дисплей з HVGA-роздільною здатністю 480×320, IPS-матрицею. На звороті модуля є слот для SD-картки. На флешці можна зберігати зображення у форматі RAW та використовувати їх для побудови графіки. Дисплей працює через паралельний інтерфейс, який з одного боку займає безліч пінів, але з іншого дозволяє досягти високої швидкості передачі даних. Піни виведені у вигляді штирькових контактів. На платі дисплея встановлені перетворювачі рівня напруги, тому він однаково добре і відразу працює як на 5 В платах, так і на 3,3 В. Для виведення графіки з Arduino можна скористатися готовою бібліотекою UTFT. Швидкість повної перезаливки екрана становить частки секунди. Можливостей дисплея цілком вистачить для виведення нескладної анімації [48].

Таким чином, до вибору дисплея потрібно підходити з вимог до проєкту. Оскільки, можливо знадобиться виводити графік залежності температури від часу, то краще використовувати TFT-дисплей, хоча OLED дисплей дозволяє знизити споживання електроенергії, що є вимогою до підсистеми моніторингу.

3.4 Вибір середовища для розробки програмного забезпечення підсистеми

Для програмування плат та модулів на ESP32 є досить великий набір мов програмування: Arduino C, Lua, MicroPython та Espruino.

В якості середовищ розробки можуть бути розглянуті Arduino Integrated Development Environment (IDE), PlatformIO, Espressif IDF, Lua RTOS, Espruino, Rymakr IDE та ін. [49] - [51].

Arduino IDE має відносно низький поріг входження, його можна вільно завантажити з офіційного сайту. Незважаючи на очевидні обмеження порівняно з професійним середовищем розробки, Arduino IDE покриває 90% того, що потрібно досягти для нескладних проєктів. У мережі також є достатня кількість статей на тему встановлення та налаштування Arduino IDE для програмування модулів ESP32. Програмуючи ESP32 у середовищі Arduino необхідно враховувати призначення пінів. Основна перевага розробки в середовищі Arduino IDE полягає у швидкому входженні та легкості створення проєктів, за допомогою тих самих принципів та багатьох бібліотек, що й для Arduino. Ще однією корисною особливістю є можливість поєднувати бібліотеки та принципи розробки Arduino з оригінальним ESP-IDF фреймворком [49].

PlatformIO – це кросплатформене IDE, екосистема для розробки вбудованих пристроїв, що підтримує безліч платформ, включаючи Arduino та ESP32. Як IDE використовується Visual Studio Code або Atom. Встановлення та налаштування досить просте – після встановлення редактора коду вибираємо PlatformIO зі списку плагінів та встановлюємо. У мережі є багато матеріалів на цю тему, починаючи від офіційного джерела, і продовжуючи статтями з докладними ілюстраціями. PlatformIO в порівнянні з Arduino IDE має всі якості сучасного середовища розробки: організація проєктів, підтримка плагінів, автодоповнення коду та багато іншого. Особливістю розробки на PlatformIO є уніфікована структура проєкту для всіх платформ [49].

Фреймворк IoT Development Framework, відомий як ESP-IDF, для ESP32 розробила компанія Espressif. Проект містить дуже гарну документацію та приклади, які можна взяти за основу. Установка та налаштування середовища добре документовані, є кілька варіантів установки і роботи з фреймворком [49].

ESP-IDF Eclipse плагін добре інтегрований в середовище розробки і автоматизує левову частку функціоналу. Але, в Eclipse версіях пізніше 2019-09 в ESP-IDF проектах у Windows досі є баг з індексуванням вихідних файлів [49].

ESP-IDF Visual Studio Code Extension – це офіційний плагін для Visual Studio Code. Як і PlatformIO, він легко встановлюється із розділу розширень. Встановлення та налаштування ESP-IDF фреймворку в цьому розширенні представлено як меню onboarding. Завантаження та встановлення всіх компонентів відбувається автоматично у процесі проходження етапів меню. Можна привести всі скріни процесу, але вони інтуїтивно зрозумілі і практично не вимагають пояснення. На користь PlatformIO можна відзначити більш зручний інструментарій білда, завантаження та моніторингу проекту. На відміну від цього ESP-IDF плагін керується за допомогою меню команд, яке можна викликати за допомогою клавіші F1 або комбінації клавіш, описаних у мануалі. Перевага використання плагіна полягає в тому, що дотримується класична структура проекту, немає необхідності якимось корегувати налаштування (в PlatformIO така необхідність виникає). Якщо ми хочемо відкрити раніше створений проект у Visual studio code з ESP-IDF плагіном, то нам лише необхідно скопіювати в корінь з проектом каталог .vscode, який можна отримати, згенерувавши хоча б один раз шаблонний проект за допомогою ESP- IDF плагіна [49].

Lua — це надійна, легка та вбудовувана мова сценаріїв, яка підтримує кілька методів програмування, включаючи процедурне, об'єктно-орієнтоване, функціональне та кероване даними програмування. Оскільки основна увага на Lua приділяється написанню сценаріїв, вона рідко використовується як окрема мова програмування. Натомість - як мова сценаріїв, яку можна інтегрувати

(вбудовувати) в інші програми, написані переважно на C і C++. Вона також підтримує інші мови програмування через плагіни сторонніх розробників (NLua/KeraLua для .NET/C#) [51].

Lua активно використовують:

- як популярний компонент у розробці відеоігор та ігрових движків;
- як мова програмування в багатьох мережевих програмах, таких як CISCO Systems, Nmap і ModSecurity;
- як мова програмування в промислових програмах, таких як Adobe Lightroom і MySQL Workbench;
- як бібліотека, яку розробники можуть інтегрувати зі своїми програмами, щоб задіяти функціональність сценаріїв [51].

Недоліки Lua:

- обмежена підтримка обробки помилок може призвести до збільшення часу налагодження для виявлення помилок у сценарії Lua;
- усі змінні створюються як глобальні змінні (глобальна область), що може призвести до помилок у присвоєнні змінних;
- обмежена підтримка відповідності шаблону [51].

В таблиці 3.5 наведено результати порівняльного аналізу двох найбільш популярних середовищ розробки для плат та модулів на ESP32. Очевидною перевагою є наявність вебверсії середовища Arduino IDE [49] - [52].

Arduino Web Editor – це онлайн інструмент, який дозволяє розробляти скетчі за допомогою веббраузера (Chrome, Firefox, Safari та Edge). Розробники платформи Arduino рекомендують використовувати браузер Google Chrome [52].

Ця IDE є частиною проекту Arduino Create, онлайн платформи, яка дозволяє розробникам створювати програми, конфігурувати плати, надає можливість доступу до навчальних матеріалів та ділитися своїми проєктами з іншими учасниками спільноти Arduino.

Таблиця 3.5 – Порівняння середовищ розробки для ESP32 [49] - [51]

Характеристика	Arduino IDE	PlatformIO
Простота	+	+
Підтримка	+	+
Платформа	Java	Python
Операційна система	Windows, OS X, Linux	Windows, macOS, Linux, FreeBSD
Менеджер бібліотек	-	+
Підтримка плагінів	-	+
Наявність вебверсії	+	-
Доступність	Open-source	Open-source

Arduino Create забезпечує взаємозв'язок між усіма процесами створення готового виробу, починаючи з його задуму і закінчуючи його здійсненням. Тобто, за допомогою цього сервісу можливо керувати всіма аспектами створення проєкту [52].

3.5 Висновки до розділу 3

В результаті проведених робіт з проєктування підсистеми, виконано аналіз вимог, розроблено архітектуру підсистеми, обрано апаратні засоби розробки (Wi-Fi модуль LuaNode32 з ESP-32, датчик температури та вологості DHT22, OLED дисплей) а також онлайн середовище розробки Arduino Web Editor.

4 КОНСТРУЮВАННЯ ПІДСИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ПОКАЗНИКІВ МІКРОКЛІМАТУ

4.1 Алгоритм функціонування підсистеми

Схему алгоритму функціонування підсистеми наведено на рисунку 4.1. Підсистема дозволяє контролеру активувати опалення або охолодження на основі невідповідності показників температури та/або вологості, які отримані від бездротових датчиків, а також з урахуванням присутності людей у приміщенні.

Етап 1. Початок.

Етап 2. Встановлення порогових значень показників мікроклімату.

Етап 3. Зчитування даних з датчиків

Етап 4. Перевірка присутності людей у приміщенні.

Етап 5. Перевірка відповідності отриманих в результаті вимірювання показників заданим пороговим значенням.

Етап 6. У разі невідповідності показників заданим значенням температури та вологості активується система ОВКП.

Етап 7. У разі відповідності показників заданим значенням температури та вологості система ОВКП вимикається.

Мікроконтролер ESP32 виступає в ролі ведучого пристрою шини та відповідає за ініціювання зв'язку (тобто зчитування даних). Датчик вологості і температури DHT22 завжди залишається в якості підлеглого пристрою та надає дані, коли МК запитує їх. Процес комунікації МК та DHT22 відбувається в декілька етапів [44]:

- МК подає стартовий сигнал - притискає шину до землі на 18-20 мс;
- після цього МК відпускає лінію і стежити за рівнем на ній, приблизно 20-40 мкс;

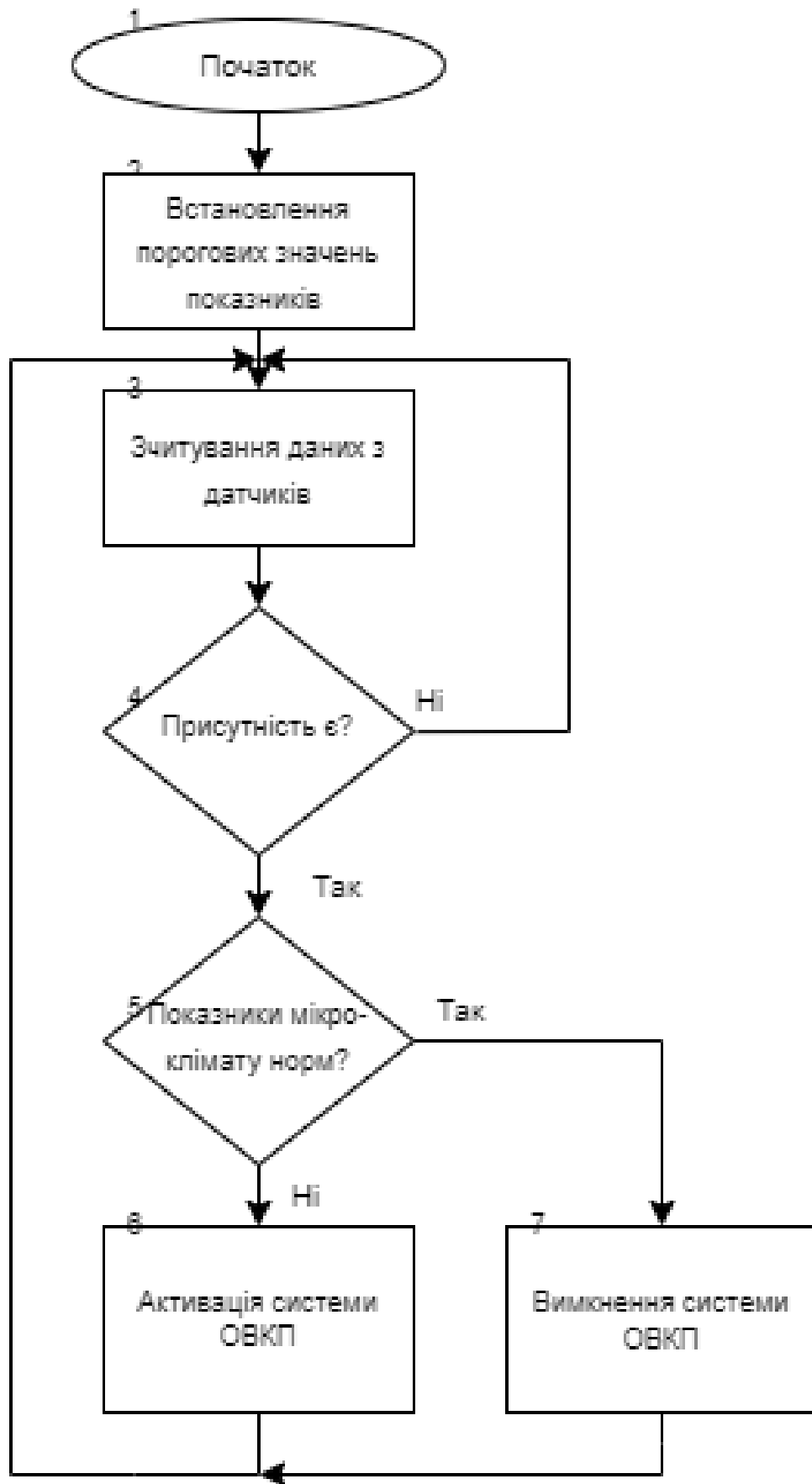


Рисунок 4.1 – Схема алгоритму функціонування підсистеми

- датчик, виявивши сигнал і почекавши поки рівень знову стане високим, сам притискає шину до землі на 80 мкс, за цей час робляться вимірювання та перетворення результатів;
- датчик відпускає лінію на 80 мкс, що вказує на те, що він готовий відправити дані;
- датчик відправляє 40 біт даних. Перед відправкою кожного біта датчик притискає шину до землі на 50 мкс, далі йде 26-28 мкс для «0» або 70 мкс для «1»;
- після завершення зв'язку лінія витягується підтягуючим резистором і переходить в стан очікування [44].

Коли датчик вологості і температури відправляє дані, він спочатку відправляє MSb (Most Significant Bit) - старший значущий біт. Дані від датчика передаються у вигляді відправлення, що складається з 40 біт даних - це 5 байт; з них перших два - вологість, наступні 2 - температура та байт парності. Байт парності дорівнює сумі попередніх байт. 1 і 2 байт містять відповідно цілу і дробову частину інформації про вологість, 3 і 4 байт містять цілу і дробову частину інформації про температуру. Для датчика DHT11 2-й і 4-й байти завжди нуль [44].

Значення цих байтів полягає в наступному

- 1-й байт: відносна вологість - ціла частина в %;
- 2-й байт: десята частина відносної вологості в % (нуль для DHT11);
- 3-й байт: ціла частина температури в ° C;
- 4-й байт: десята частина температури в ° C (нуль для DHT11);
- 5-й байт: контрольна сума (останні 8 біт {1-й байт + 2-й байт + 3-й байт + 4-й байт}) [44].

4.2 Опис програмної реалізації підсистеми

4.2.1 Вузол-передавач підсистеми

Вузол-передавач підсистеми складається з модуля LuaNode32 з ESP-32 та датчика температури та вологості DHT22. Додатково може бути підключений датчик контролю присутності. В таблиці 4.1 наведено опис використаних бібліотек, а в таблиці 4.2 – опис функцій програми. Текст програми наведено в Додатку А.1.

Коду програми розбито на сегменти функціями, що дозволило створити частини коду, які виконують певні завдання. Після виконання відбувається повернення в місце, звідки була викликана функція. Причиною створення функції є необхідність виконувати однакову дію кілька разів [53].

Поділу коду на функції має ряд переваг:

- функції дозволяють організувати програму (скласти її концепцію);
- функції кодують одну дію в одному місці програми;
- функції скорочують шанси на появу помилки при необхідності зміни коду;
- функції скорочують текст скетчів і роблять його компактним, тому що деякі секції використовуються багато разів;
- функції полегшують використання коду в інших програмах роблячи його модульним. У цьому випадку функції мають ще одним невеликим перевагою, роблячи код програми легким для читання [53].

На передавальній та на приймальній платі UUID (Universal Unique Identifier) мають бути однаковими.

В void setup() запускаємо датчик dht.begin(), також послідовну комунікацію Serial.begin(115200).

Таблиця 4.1 – Використані бібліотеки

Бібліотека	Призначення
DHT.h	Для організації програмної взаємодії датчика вологості і температури
ESP32 BLE Arduino	Передача даних між модулями вузла-передавача та вузла-приймача за допомогою Bluetooth з'єднання

Таблиця 4.2 – Функції програми

Функція	Призначення
#include	Підключення бібліотек
#define temperatureCelsius	Відправка температури в градусах цельсія
#define bleServerName "dhtESP32"	Привласнення імені серверу
#define DHTTYPE DHT22	Призначення використовуваного датчика температури та вологості
#define SERVICE_UUID "91bad492-b950-4226-aa2b-4ede9fa42f59"	Створення UUID
const int DHTPin = 14	Призначення контакту, до якого буде підключений датчик
DHT dht(DHTPin, DHTTYPE)	За допомогою зазначеного контакту створює об'єкт типу DHT під назвою DHT
#ifdef temperatureCelsius	Задання температурної характеристики
onConnect() и onDisconnect().	Функції зворотнього виклику
readHumidity()	Зчитує дані про зміну вологості з датчика типу DHT
begin ()	задає швидкість передачі даних по послідовному інтерфейсу

Створюємо BLE пристрій BLEDevice ::init(bleServerName), далі BLE сервер BLEServer *pServer = BLEDevice::createServer();

Наступним кроком створюємо BLE-сервіс BLEService *dhtService = pServer->createService(SERVICE_UUID). За допомогою умови #ifdef створюємо BLE-характеристики и BLE-дескриптор. Далі потрібно запустити сервіс за допомогою dhtService->start. За допомогою pServer->getAdvertising()->start() запускаємо розсилку повідомлень. В void loop запускаємо зчитування температури з датчика за допомогою float t = dht.readTemperature з заданою умовою що if (deviceConnected). Далі перевіряємо чи вдалося прочитати дані з датчика if (isnan(h) || isnan(t) || isnan(f)) якщо ні то виходимо з void loop та починаємо заново. Далі треба задати значення цельсій для температурної характеристики, та відправляємо значення підключеному клієнту dhtTemperatureCelsiusCharacteristics.setValue(temperatureCTemp). Наступним кроком задаємо характеристики вологості, та відправляємо її підключеному клієнту: dhtHumidityCharacteristics.setValue(humidityTemp). За допомогою функції Serial.print виводимо потрібну нам інформацію.

4.2.2 Вузол-приймач підсистеми

Вузол-приймач підсистеми складається з модуля LuaNode32 з ESP-32 та OLED дисплея. В таблиці 4.3 наведено опис використаних бібліотек, а в таблиці 4.4 – опис функцій програми. Текст програми наведено в Додатку А.2.

Задаємо змінні для виявлення потреби підключати пристрій чи підключення завершено static boolean doConnect = false, static boolean connected = false. Наступним кроком задаємо адресу периферійного пристрою static BLEAddress *pServerAddress , воно повинна бути знайдена під час сканування.

Далі задаємо характеристики, які треба прочитати static BLERemoteCharacteristic* temperatureCharacteristic, також задаємо для вологості static BLERemoteCharacteristic* humidityCharacteristic.

Таблиця 4.3 – Використані бібліотеки

Бібліотека	Призначення
Adafruit_SSD1306.h	Бібліотека, що відповідає за роботу дисплея
Adafruit_GFX.h	Бібліотека для шрифтів
ESP32 BLE Arduino	Передача даних між модулями вузла-передавача та вузла-приймача за допомогою Bluetooth з'єднання
Wire.h.	Бібліотека для послідовної передачі даних

Таблиця 4.4 – Функції програми

Функція	Призначення
#define bleServerName "dhtESP32"	Задання назви сервера
dhtServiceUUID("91bad492-b950-4226-aa2b-4ede9fa42f59")	Введення UUID для сервіса static BLEUUID
#ifdef temperatureCelsius	Задання температурної характеристики в градусах цельсія
humidityCharacteristicUUID("ca73b3ba-39f6-4ab3-91ae-186dc9577d99")	Задання характеристики для вологості static BLEUUID
callback()	Визначення функції зворотнього виклику повідомлення
readTemperature()	Зчитування даних про зміну температури з датчика типу DHT

Встановлюємо включення виключення повідомлень `const uint8_t notificationO [] = {0x1, 0x0}`. Функцією `#define SCREEN_WIDTH 128` задаємо ширину дисплея. Створюємо дисплей `SSD1306, Adafruit_SSD1306 display` вказуємо контакти зброду `#define OLED_RESET 4`.

Далі треба підключитися до к BLE-серверу, у якого є назва сервіса і характеристики `bool connectToServer(BLEAddress pAddress)`. В цій же частині коду підключаємося до віддаленого BLE-серверу `pClient->connect(pAddress)`,

потім зчитуємо UUID сервіса який ми шукаємо `BLERemoteService* pRemoteService = pClient->getService(dhtServiceUUID)`. Задаємо умову коли наша характеристика не знайдеться `else if (temperatureCharacteristic == nullptr || humidityCharacteristic == nullptr)`. Присвоюємо характеристикам функції зворотнього виклику `temperatureCharacteristic->registerForNotify(temperatureNotifyCallback)`.

Наступним кроком треба задати функції зворотнього виклику, коли ми отримали повідомлення від іншого пристрою про температуру `class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {void onResult(BLEAdvertisedDevice advertisedDevice)`. Виведемо її `display.print(" *C")`.

Також треба задати функцію зворотнього виклику, коли від передавача отримаємо разом з повідомлення дані про вологість `static void humidityNotifyCallback(BLERemoteCharacteristic* pBLERemoteCharacteristic, uint8_t* pData, size_t length, bool isNotify)` виведемо їх `Serial.print(" Humidity:")`.

В `void setup()` задаємо налаштування дисплею, задаємо напругу, а за допомогою функції `display` налаштовуємо дисплей, наприклад, колір `display.setTextColor`. Запускаємо послідовну комунікацію `Serial.begin(115200)`.

Ініціалізуємо пристрій `BLEDevice::init("")`. Наступним кроком створюємо екземпляр класу «BLEScan» для сканування і задаємо для цього об'єкту функцію зворотнього виклику, яка буде інформувати про те, чи знайдено новий пристрій. Додатково вказуємо, що нам потрібне активне сканування, а потім запускаємо 30-секундне сканування `BLEScan* pBLEScan = BLEDevice::getScan()`.

Наступним кроком в `void loop()` задаємо умову, якщо у змінній «doConnect» значення «true», то це означає, що сканування завершено, і ми знайшли потрібний BLE-сервер, до якого хочемо підключитися. Після підключення записуємо в «connected» значення «true».

Активуємо властивість «notify» для кожної характеристики `temperatureCharacteristic->getDescriptor(BLEUUID((uint16_t)0x2902))->writeValue((uint8_t*)notificationOn, 2, true)` . Так само і для вологості.

В іншому випадку, за допомогою `Serial.println` виводимо інформацію про те що, треба перезапустити сервер.

4.3 Висновки до розділу 4

В даному розділі виконано розробку керуючого алгоритму, що показує взаємодію розробленої підсистеми моніторингу показників мікроклімату медичних приміщень з системою ОВКП на основі даних з датчиків температури, вологості та присутності та формування відповідних керуючих впливів на обладнання системи ОВКП.

Розроблена програма складається з двох модулів, що забезпечують функціонування вузла-передавача та вузла-приймача бездротової сенсорної мережі на основі технології Bluetooth. Представлено опис використаних бібліотек та функцій програми.

5 КЕРІВНИЦТВО ПРОГРАМІСТА

5.1 Призначення й умови застосування підсистеми

Підсистема віддаленого моніторингу показників мікроклімату медичних приміщень призначена для збору даних з вузлів БСМ та їх передачі на сервер, де реалізовано алгоритми управління системою ОВКП.

Програма функціонує на модулях LuaNode32 з ESP-32 з використанням датчика температури та вологості DHT22 та OLED дисплея.

5.2 Характеристики програми

Розроблювана підсистема забезпечує наступні можливості:

- коректне зчитування даних щодо температури та вологості з датчиків та візуалізація інформації;
- можливість підключення додаткових вузлів до БСМ та додаткових датчиків до кожного вузла, наприклад датчиків присутності;
- перевірка підключення бездротових сенсорних вузлів до БСМ та сповіщення користувача про помилку підключення конкретного вузла;
- здатність працювати в автономному режимі в разі відсутності зв'язку з сервером.

5.3 Звернення до програми

Існує надбудова для Arduino IDE, яка дозволяє програмувати ESP32 за допомогою Arduino IDE та її мови програмування. Щоб встановити модуль ESP32 в Arduino IDE для будь якої ОС (Windows, Mac OS X або Linux) треба виконати наступне [54].

Перш ніж почати процедуру встановлення, переконайтеся, що на вашому комп'ютері встановлено останню версію Arduino IDE.

У середовищі Arduino IDE перейдіть до File> Preferences (рис.5.1).

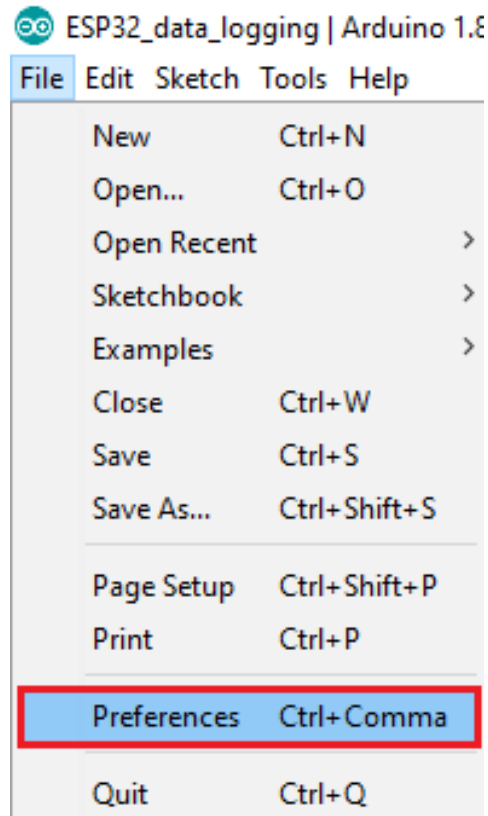


Рисунок 5.1 – Початок налаштувань

Введіть наступну інформацію в поле “Additional Board Manager URLs” (рис. 5.2):

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Натисніть Ок. Якщо у вас уже є URL-адреса плат ESP8266, ви можете відокремити URL-адреси комою наступним чином:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

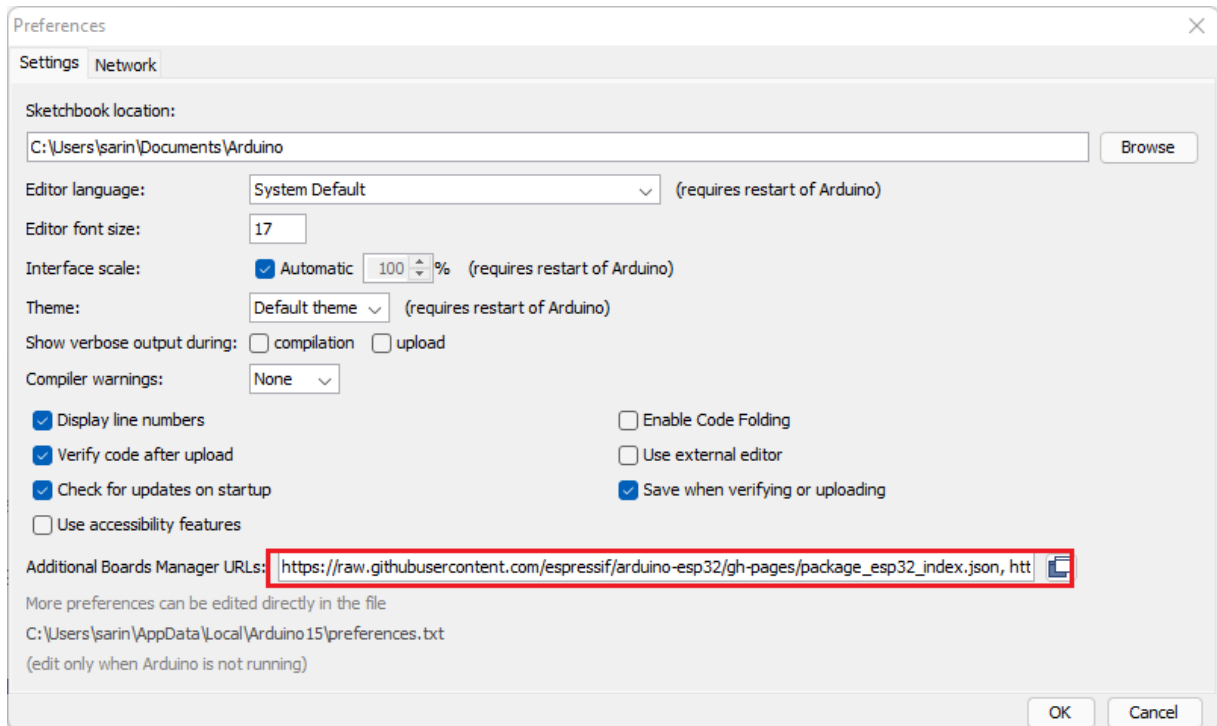


Рисунок 5.2 – Введення URL

Відкрийте Менеджер плат Tools > Board > Boards Manager (рис. 5.3).

Знайдіть ESP32 і натисніть кнопку встановлення для «ESP32 by Espressif Systems» (рис. 5.4). Після цього модуль ESP32 в Arduino IDE буде встановлено.

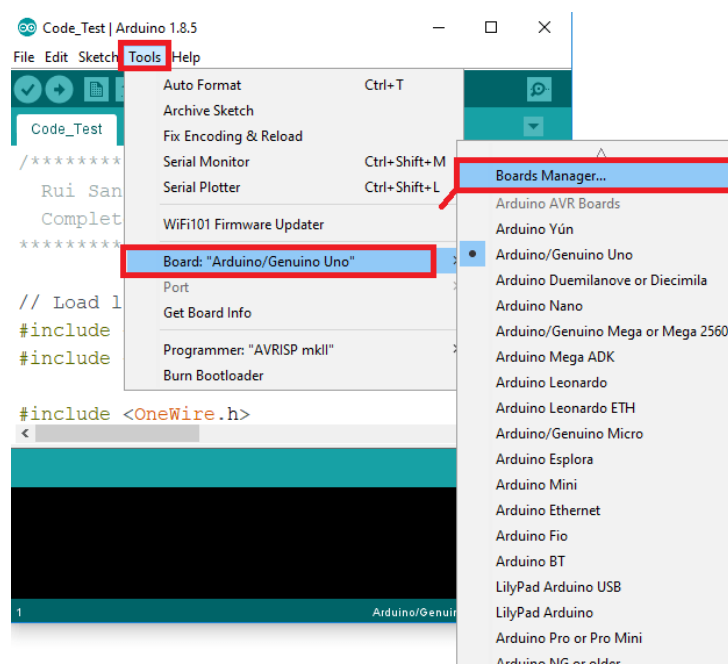


Рисунок 5.3 – Менеджер плат

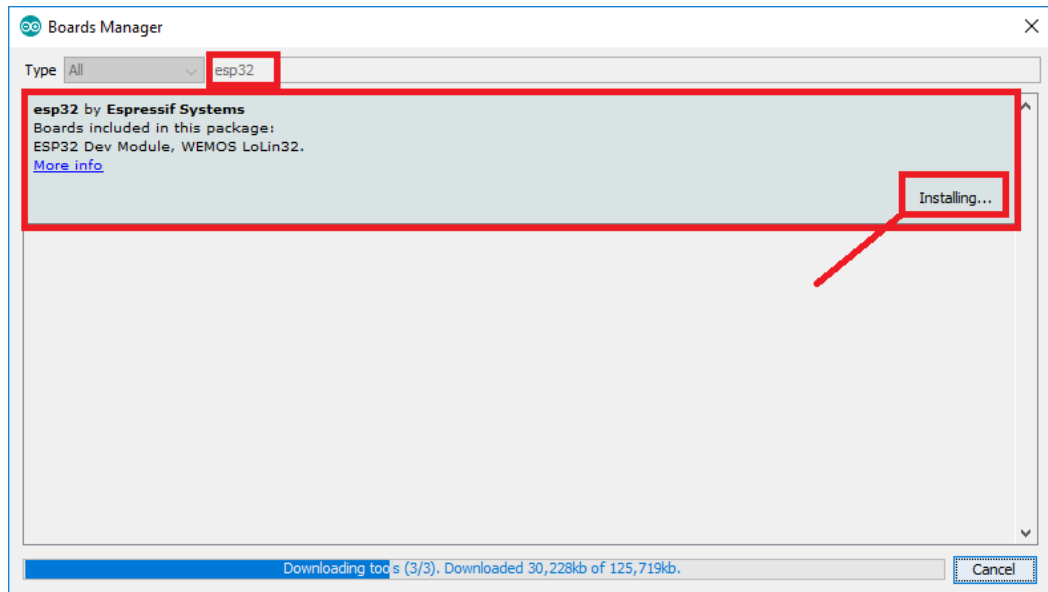


Рисунок 5.4 – Вибір ESP32

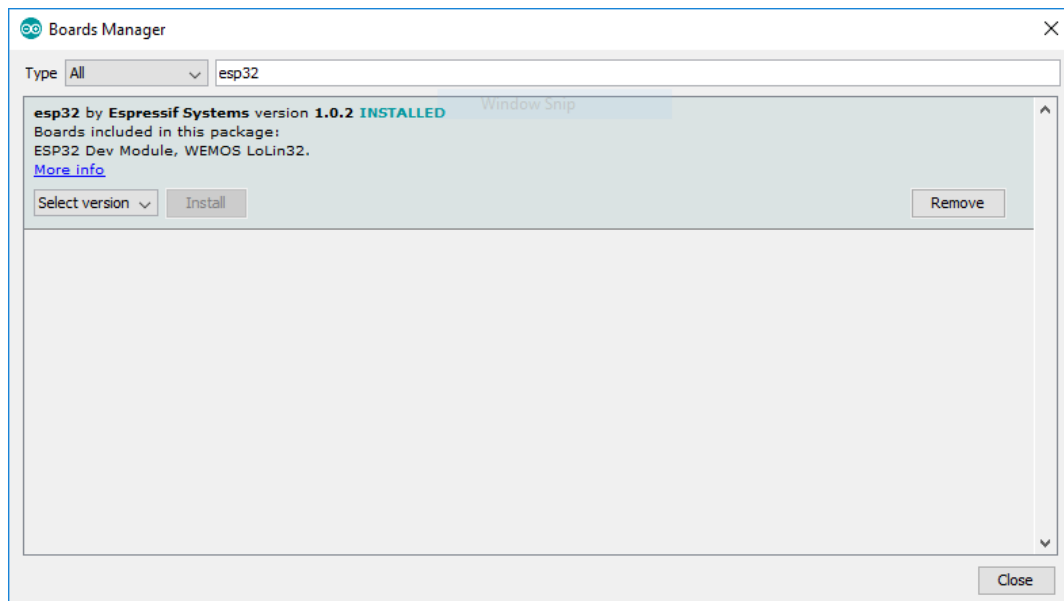


Рисунок 5.5 - Модуль ESP32 в Arduino IDE

Підключіть модуль ESP32 до комп'ютера. Відкривши Arduino IDE, виконайте такі дії. Виберіть модуль в меню Tools > Board menu.

Виберіть порт (якщо ви не бачите порт COM у своєму Arduino IDE, вам потрібно встановити драйвери VCP мосту CP210x USB to UART) (рис. 5.6).

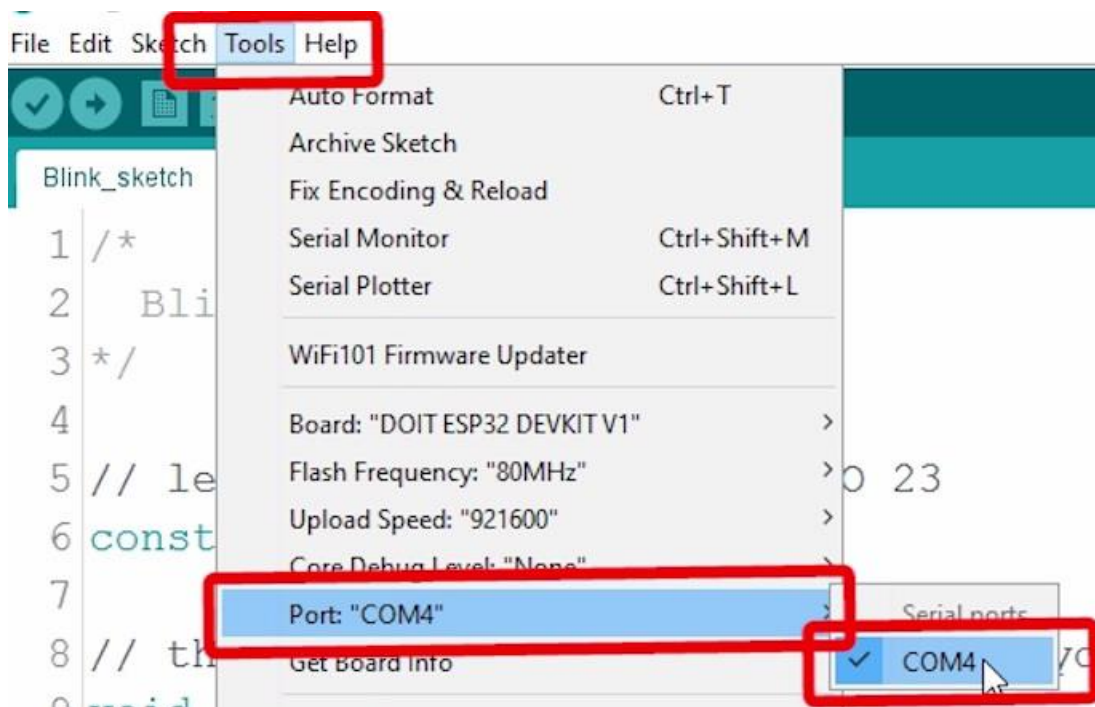


Рисунок 5.6 – Вибір COM порту

Відкрийте наступний приклад File > Examples > WiFi (ESP32) > WiFiScan (рис.5.7) та у вашому Arduino IDE відкриється новий скетч (рис.5.8).

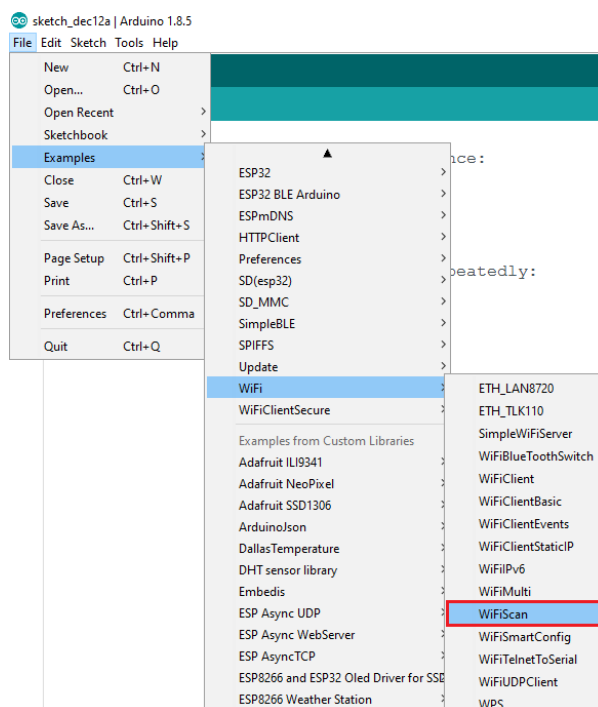


Рисунок 5.7 – Вибір прикладу

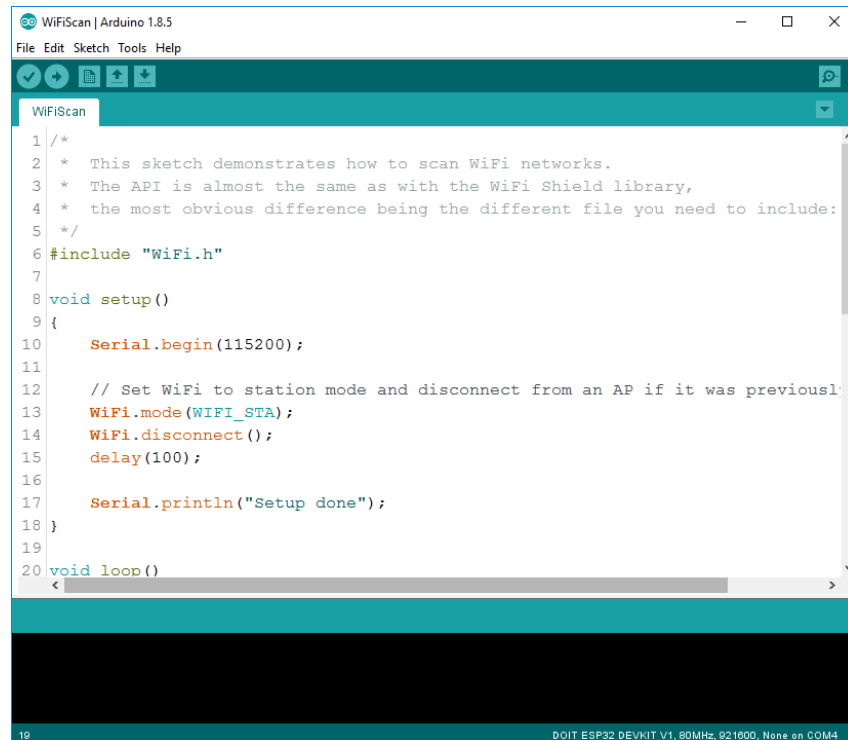


Рисунок 5.8 – Відкриття скетчу

Натисніть кнопку Upload в Arduino IDE. Зачекайте кілька секунд, поки код компілюється та завантажиться на ваш модуль. Якщо все пройшло, як очікувалося, ви побачите повідомлення «Done uploading» (рис. 5.9).

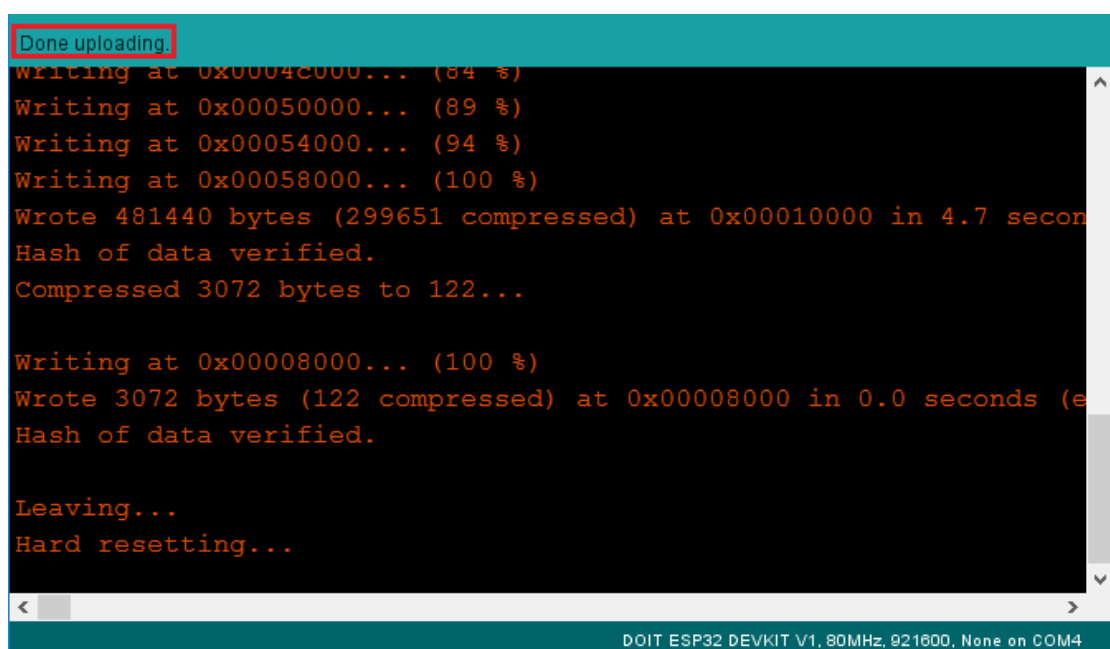


Рисунок 5.9 – Повідомлення про завершення завантаження

Відкрийте Arduino IDE Serial Monitor зі швидкістю передачі даних 115200. Натисніть вбудовану кнопку ввімкнення ESP32, і ви побачите мережі, доступні поблизу вашого модуля ESP32 (рис.5.10).

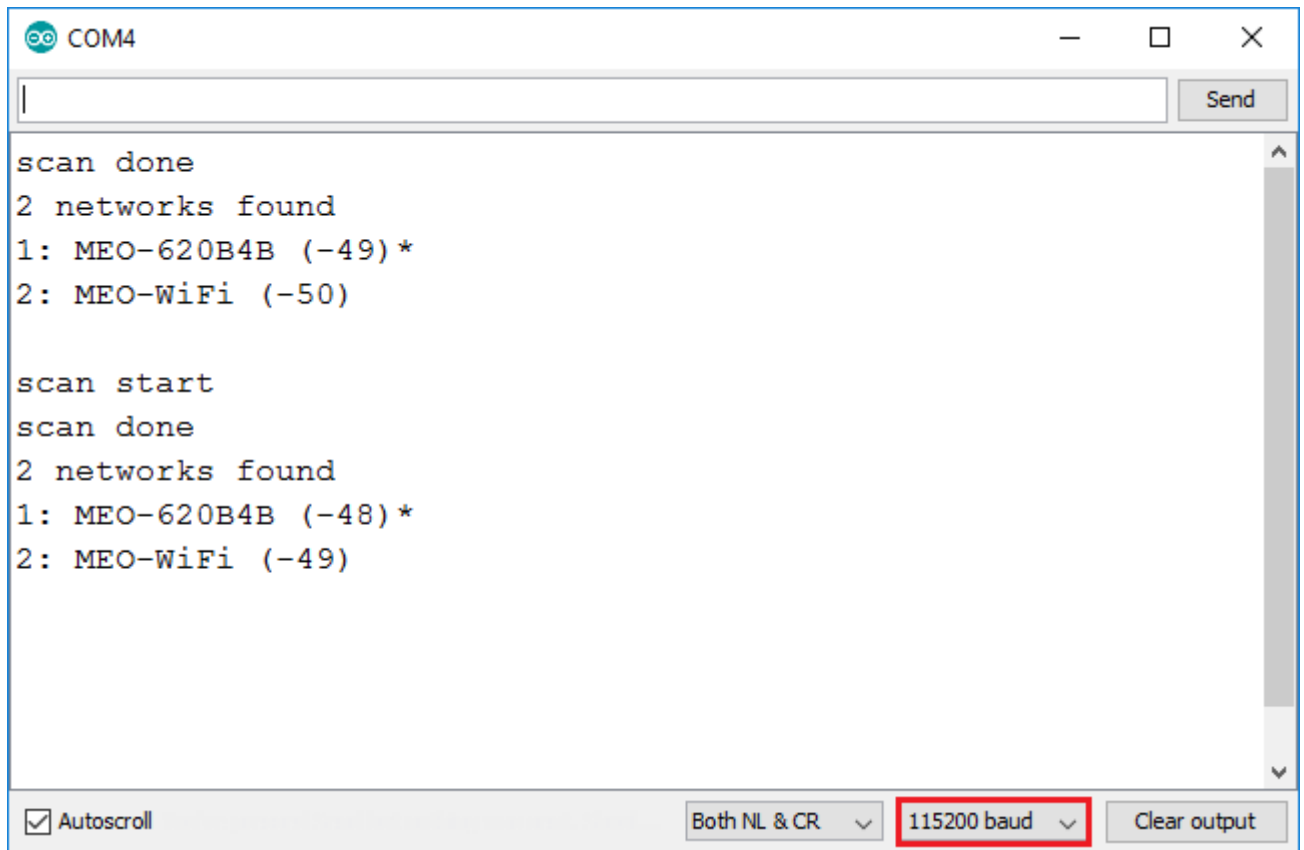


Рисунок 5.10 – Пошук мережі

5.4 Вхідні та вихідні дані

Вхідними даними до програми є інформація з датчика температури та вологості. Вихідними даними є сигнали керування для системи ОВКП.

5.5 Повідомлення програмісту

Якщо ви спробуєте завантажити новий ескіз до ESP32, і ви отримаєте таке повідомлення про помилку «Сталася фатальна помилка: не вдалося

підключитися до ESP32: минув час очікування... Підключення...». Це означає, що ваш ESP32 не перебуває в режимі мигання/завантаження.

Вибравши правильне ім'я плати та COM-пор, виконайте такі дії:

- утримуйте кнопку «BOOT» на платі ESP32;
- натисніть кнопку «Завантажити» в Arduino IDE, щоб завантажити свій ескіз;
- після того, як ви побачите «Підключення...». повідомлення у вашому Arduino IDE, відпустіть палець від кнопки «BOOT».

Після цього ви повинні побачити повідомлення «Завантаження завершено». На вашому ESP32 повинен працювати новий скетч. Натисніть кнопку «ENABLE», щоб перезапустити ESP32 і запустити новий завантажений скетч.

6 КЕРІВНИЦТВО КОРИСТУВАЧА

6.1 Призначення програми

Підсистема віддаленого моніторингу показників мікроклімату медичних приміщень призначена для збору даних з вузлів БСМ та їх передачі на сервер, де реалізовано алгоритми управління системою ОВКП.

6.2 Умови виконання програми

Програма функціонує на модулях LuaNode32 з ESP-32 з використанням датчика температури та вологості DHT22 та OLED дисплея.

6.3 Робота з програмою

Вузол-передавач підсистеми моніторингу мікроклімату складається з модуля LuaNode32 з ESP-32 та датчика температури та вологості DHT22 (рис. 6.1). Додатково може бути підключений датчик контролю присутності.

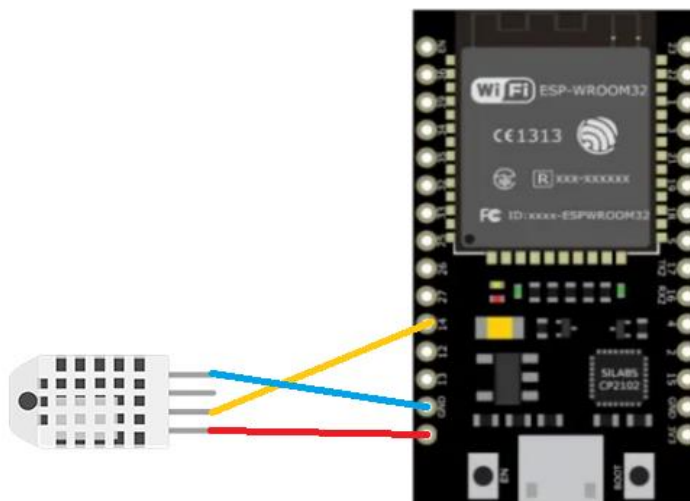


Рисунок 6.1 – Схема підключення датчика DHT22 до модуля LuaNode32

На рисунку 6.2 зображено результат підключення датчика DHT22 до модуля LuaNode32 з ESP-32.

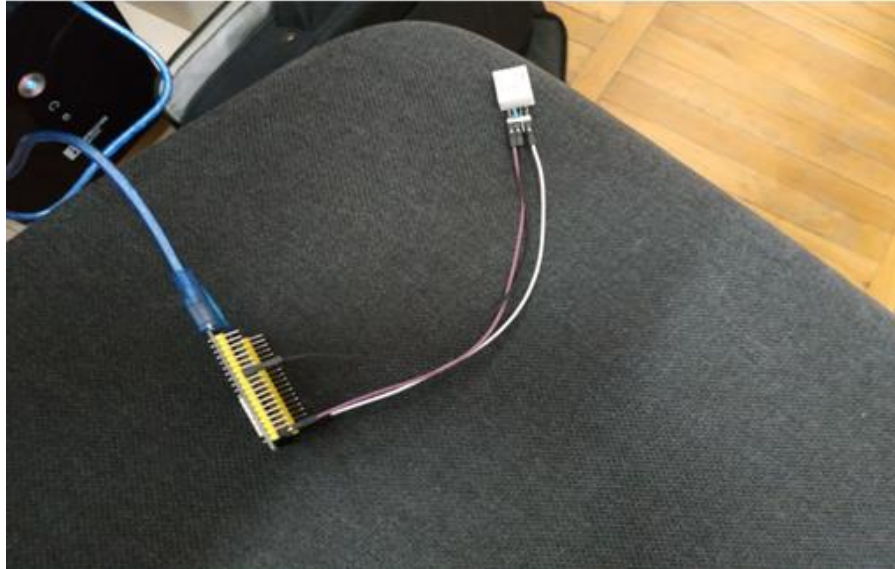


Рисунок 6.3 – Зовнішній вигляд вузла-передавача

Вузол-приймач складається з модуля LuaNode32 з ESP-32 та OLED дисплея (рис. 6.3).

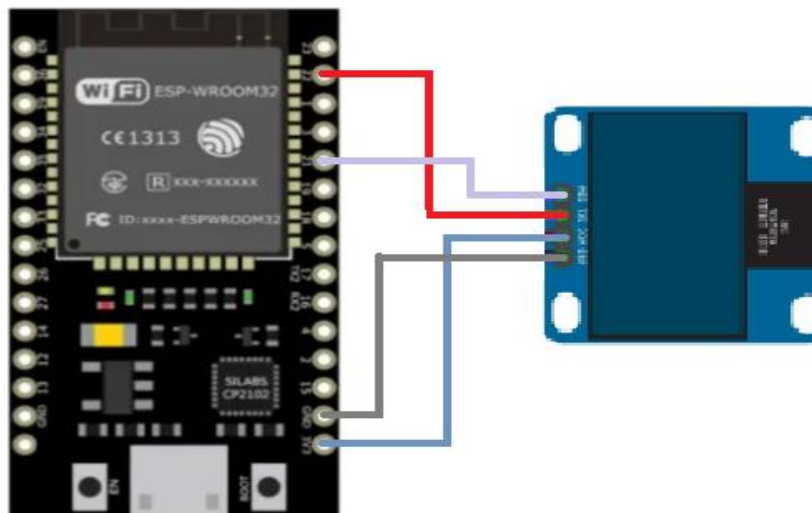


Рисунок 6.3 – Схема підключення дисплею до модуля LuaNode32

На рисунку 6.4 показано результат з'єднання модуля LuaNode32 з ESP-32 та OLED дисплея. Для правильного підключення необхідно ознайомитися з виводами, які є на дисплеї та перевірити правильність підключення контактів та якість їх закріплення.

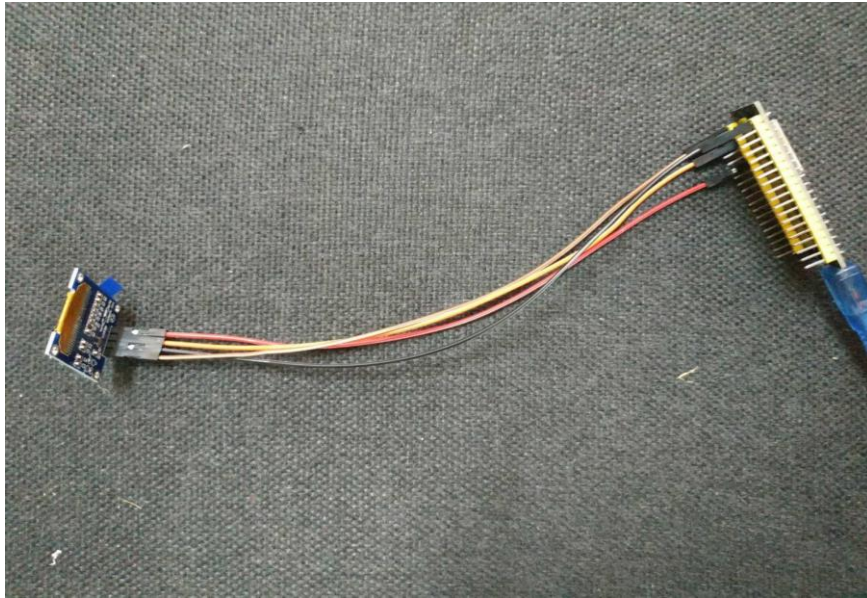


Рисунок 6.4 – Зовнішній вигляд вузла-приймача

На рисунку 6.5 показано загальний вигляд прототипу підсистеми моніторингу показників мікроклімату.

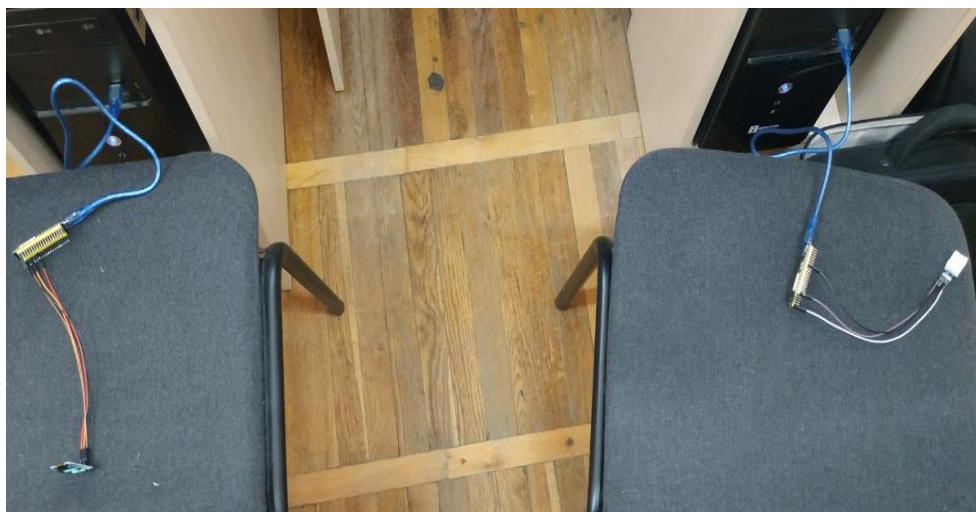


Рисунок 6.5 – Загальний вигляд прототипу підсистеми

На рисунку 6.6 показано результат роботи підсистеми. На дисплеї відображається температура в градусах цельсія, також може відображатися вологість. Передача інформації між вузлами відбувається за допомогою Bluetooth технології.



Рисунок 6.6 – Виведення на дисплей даних щодо температури у приміщенні

На рисунку 6.7 наведено результати візуалізації графіків залежності температури та вологості від часу.

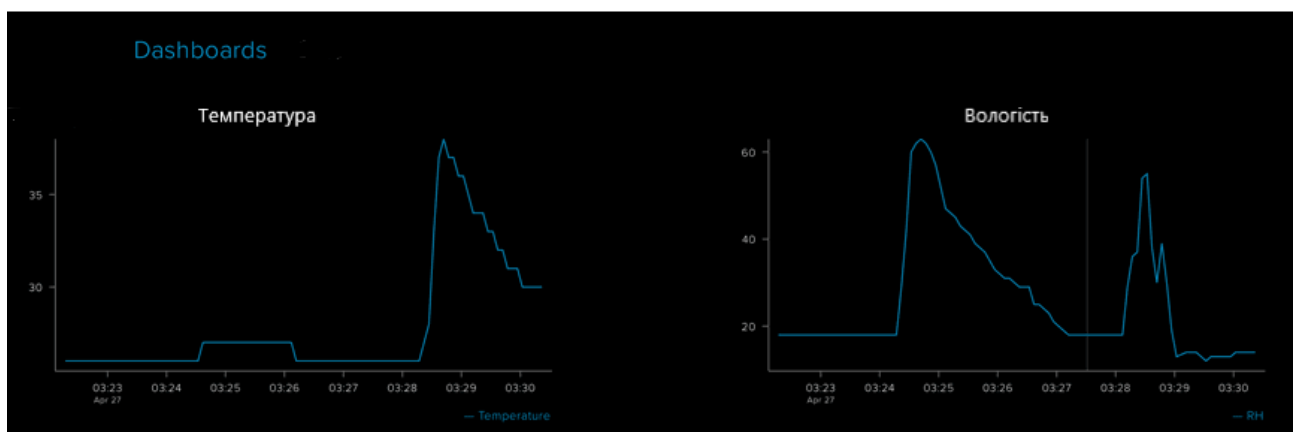


Рисунок 6.7 – Результат виведення графіка зміни температури та вологості у приміщенні

ВИСНОВКИ

В ході виконання дипломної роботи було розглянуто особливості побудови та функціонування існуючих систем опалення, вентиляції та кондиціонування медичних приміщень. Показано, що сучасна індустрія надає достатньо широкий спектр контролерів та датчиків, які можливо використовувати для реалізації підсистеми моніторингу показників мікроклімату медичних приміщень, забезпечення яких є надзвичайно важливим для здоров'я пацієнтів та працездатності медичного персоналу.

Як показали проведені дослідження, бездротова сенсорна мережа дозволяє застосувати сучасні технології передачі даних та побудувати ефективну підсистему моніторингу.

В роботі було проведено проектування та конструювання підсистеми моніторингу показників мікроклімату медичних приміщень. Основними компонентами передавальної та приймальної частин підсистеми є модулі LuaNode32 з ESP-32, OLED дисплей та датчик вологості та температури DHT22. Додатково передбачено використання датчика присутності для виявлення знаходження людей у приміщенні та більш раціонального використання енергоносіїв. Програмне забезпечення розроблено на мові програмування C++ в середовищі розробки Arduino IDE.

Перевагою розробленої підсистеми є те, є те що вона може виступати прототипом для подальшого розвитку та удосконалення подібних рішень. Вона легко збирається, має низьке енергоспоживання, що дозволяє їй працювати значний час в автономному режимі. Також ця система є мобільною завдяки використанню бездротових технологій.

Розроблена підсистема може бути інтегрована в систему опалення, вентиляції та кондиціонування повітря медичних приміщень, або в системи домашньої та офісної автоматизації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Операционная и больница – как поддерживать влаготемпературный режим [Электрон. ресурс]. – Режим доступа: <https://pobut.lviv.ua/ru/articles/osushitel-dlya-operacionnoj-bolnicy>.
2. Микроклимат больничных помещений. Физиологические свойства воздуха и их значение для организма. Комплексная оценка микроклимата больничных помещений [Электрон. ресурс]. – Режим доступа: <https://studfile.net/preview/5300703/page:5/>.
3. Влияние микроклимата на здоровье человека [Электрон. ресурс]. – Режим доступа: <https://atmeex.com/ru/vliyanie-mikroklimate-na-zdorove-cheloveka/>.
4. Про затвердження Державних санітарних норм і правил «Санітарно-протиепідемічні вимоги до закладів охорони здоров'я, що надають первинну медичну (медико-санітарну) допомогу» [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0570-13>.
5. 1996 ASHRAE Handbook. HVAC systems and equipment: Inch-Pound / [American society of Heating, refrigerating, and air conditioning engineers]. – Atlanta, GA: ASHRAE, 1996. - 667 p.
6. 2000 ASHRAE Handbook. HVAC systems and equipment: Inch-Pound / [American society of Heating, refrigerating, and air conditioning engineers]. – Atlanta, GA: ASHRAE, 2000. – 780 p.
7. Sugarman, S.C. HVAC Fundamentals / S.C. Sugarman // 2nd ed., CRC Press, The Fairmont Press, Inc., 2005. - 300 p.
8. ASHRAE Handbook. Fundamentals (SI Edition) / [American society of Heating, refrigerating, and air conditioning engineers]. – Atlanta, GA: ASHRAE, 2009. - 599 p.
9. Types of HVAC systems [Electronic resource]. – Access mode: <https://www.intechopen.com/chapters/62059>.

10. Климат контроль для дома: проблемы и решения [Электрон. ресурс]. – Режим доступа: <http://ventilationpro.ru/konditsionirovanie/klimat-kontrol-dlya-doma-problemy-i-resheniya.html>.
11. Климатический контроль в умном доме [Электрон. ресурс]. – Режим доступа: <https://gadgetpage.ru/smart-house/3007-klimaticheskij-kontrol-v-umnom-dome.html>.
12. Bhandari, P. IoT based design implementation of mushroom farm monitoring using Arduino microcontrollers & sensors / P. Bhandari, M. Kimothi // International journal of engineering sciences & research technology. – 2018. – Vol. 7, № 5. – P. 550-560.
13. Макаренко, А.Ю. Бездротові технології передачі даних / А. Ю. Макаренко, А. О. Парфенова, С. Б. Могильний // Вісник Національного технічного університету України "КПІ". Серія – Радіотехніка. Радіоапаратобудування. - 2010. - №41. – С.171-181.
14. Patil, K. S. A survey on IoT based Smart Home / K. S. Patil, A. R. Kokate, S. R. Wagh // International journal of engineering sciences & research technology. – 2017. – Vol. 6, № 3. - P. 176-178.
15. Comparing wireless communication protocols [Electronic resource]. – Access mode: https://www.anixter.com/en_mx/resources/literature/techbriefs/comparing-wireless-communication-protocols.html.
16. A web-based office climate control system using wireless sensors / [T. W. Foster, D. V. Bhatt, G. P. Hancke, Bruno Silva] // IEEE Sensors Journal. – 2016. – Vol. 16, № 15. - P. 6104-6113.
17. The SmartME controller: its brilliance and simplicity [Electronic resource]. – Access mode: <http://www.mitsubishipro.com/newcontrollers>.
18. Honeywell building controls - sensors [Electronic resource]. – Access mode: <https://buildingcontrols.honeywell.com/subcategory/Sensors>.

19. Honeywell building controls - HVAC controllers [Electronic resource]. – Access mode: <https://buildingcontrols.honeywell.com/HVAC-Controllers>.
20. The generic design of a high-traffic advanced metering infrastructure using ZigBee/ [H. Y. Tungm, K. F. Tsang, K. T. Chui et.al.] // IEEE Trans. Ind. Informat. – 2014. – Vol. 10, № 1. - P. 836–844.
21. Wang, Y. Sense, model and identify the load signatures of HVAC systems in metro stations / Y. Wang, H. Feng, X. Xi // IEEE PES General Meeting Conf. & Expo., National Harbor, Maryland, 27-31 July 2014: proceedings. – Los Alamitos: IEEE, 2014. – P. 1–5.
22. Model-free HVAC control using occupant feedback / [S. Purdon, B. Kusy, R. Jurdak et al.] // IEEE 38th Conf. Local Comput. Netw. Workshops (LCN Workshops), Sydney, Australia, 21-24 October 2013: proceedings. – Los Alamitos: IEEE, 2013. – P. 84–92.
23. Liang, J. Thermal comfort control based on neural network for HVAC application / J. Liang, R. Du // IEEE Conf. Control Appl., Toronto, ON, Canada, 28 - 31 August 2005: proceedings. – Los Alamitos: IEEE, 2005. – P. 819–824.
24. Rahmati, A. A hybrid fuzzy logic and PID controller for control of nonlinear HVAC systems / A. Rahmati, F. Rashidi, M. Rashidi // Systems, man and cybernetics. System security and assurance: IEEE International conference, Washington, DC, USA, 8 October 2003: proceedings. – Los Alamitos: IEEE, 2003. – P. 2249–2254.
25. Demonstration of an easy-to-apply, automated control tuning method for typical PID control loops in building energy systems / [J. Fütterer, P. J. E. Kraus, M. Schmidt et al.] // Industrial electronics and applications (ICIEA): IEEE 10th conference, Auckland, New Zealand, 15-17 June: proceedings. – Los Alamitos: IEEE, 2015. - P. 2069–2075.
26. Liu, J. Design and application of handheld auto-tuning PID instrument used in HVAC / J. Liu, W.-J. Cai, G.-Q. Zhang // Industrial electronics and

applications: 4th IEEE conference, Xian, China, 25-27 May 2009: proceedings. – Los Alamitos: IEEE, 2009. – P.1695–1698.

27. Your complete source for building automation sensors [Electronic resource]. – Access mode: <http://www.automatedbuildings.com/news/nov12/articles/enlighted/121026014101enlighted.html>.

28. Indoor air quality monitoring system using wireless sensor network (WSN) with Web interface / [S. M. Saad, A. R. M. Saad, A. M. Y. Kamarudin et al.] // Electrical, electronics and system engineering: International conference, Selangor, Malaysia, 4-5 December 2013: proceedings. – Los Alamitos: IEEE, 2013. – P. 60–64.

29. Smart monitoring for sustainable and energy-efficient buildings: A case study/ [D. Brunelli, I. Minakov, R. Passerone et al.] // Environmental, energy, and structural monitoring systems (EESMS): IEEE Workshop, Trento, Italy, 9-10 July 2015: proceedings. – Los Alamitos: IEEE, 2015. – P. 186–191.

30. Novel miniature airflow energy harvester for wireless sensing applications in buildings/ [D. Zhu, S. Beepy, M. Tudor et al.] // IEEE Sensors J. – 2013. – Vol. 13, № 2. - P. 691–700.

31. A smart thermal environment monitor based on IEEE 1451.2 standard for global networking / [Y. Wang, M. Nishikawa, R. Maeda et al.] // IEEE Trans. Instrum. Meas. – 2005. – Vol. 54, № 3. - P. 1321–1326.

32. Kumar, A. Energy efficient environment monitoring system based on the IEEE 802.15.4 standard for low cost requirements / A. Kumar, G. P. Hancke // IEEE Sensors J. – 2014. – Vol. 14, № 8. – P. 2557–2566.

33. Kumar, A. An energy-efficient smart comfort sensing system based on the IEEE 1451 standard for green buildings / A. Kumar, G. P. Hancke // IEEE Sensors J. – 2014. – Vol. 14, № 12. – P. 4245–4252.

34. Atthajariyakul, S. Real-time determination of optimal indoor-air condition for thermal comfort, air quality and efficient energy usage / S. Atthajariyakul, T. Leephakpreeda // Energy Buildings. – 2004. – Vol. 36, № 7. – P. 720–733.

35. Peng, C. Design and application of a VOC-monitoring system based on a ZigBee wireless sensor network / C. Peng, K. Qian, C. Wang, // IEEE Sensors J. - 2015. – Vol. 15, № 4. – P. 2255–2268.
36. Experimental link quality characterization of wireless sensor networks for underground monitoring / [B. Silva, R. M. Fisher, A. Kumar et al.] // IEEE Trans. Ind. Informat. – 2015. – Vol. 11, № 5. – P. 1099–1110.
37. Maier, A. Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things / A. Maier, A. Sharp, Y. Vagapov // Internet technologies and applications (ITA): Seventh international conference, Wrexham, North Wales, UK, 12-15 September 2017: proceedings. – 2017. - P.143-148.
38. Wi-Fi модуль LuaNode32 з ESP-32 [Электрон. ресурс]. – Режим доступа: <https://arduino.ua/prod2041-wi-fi-modyl-luanode32-s-esp-32>.
39. LuaNode32 ESP32 development board Wi-Fi and Bluetooth [Electronic resource]. – Access mode: <https://www.botnroll.com/en/ethernet-wi-fi/3554-esp32-development-board-wifi-and-bluetooth-esp32-wroom-32d-compatible.html>.
40. Aosong Digital-output relative humidity & temperature sensor/module – DHT11 [Electronic resource]. – Access mode: <https://arduino.ua/docs/DHT11.pdf/>.
41. Aosong Digital-output relative humidity & temperature sensor/module DHT22 [Electronic resource]. – Access mode: <https://arduino.ua/docs/DHT22.pdf/>.
42. Aosong Digital Temperature and Humidity Sensor [Electronic resource]. – Access mode: <https://datasheetspdf.com/pdf/952504/Aosong/AM2320/1/>.
43. Aosong temperature and humidity module AM2301 product manual [Electronic resource]. – Access mode: <https://arduino.ua/docs/ADC198/AM2301.pdf/>.
44. Датчики температуры и влажности DHT11 и DHT22 [Электрон. ресурс]. – Режим доступа: <http://surl.li/bxwgf>.
45. Какие бывают Arduino LCD дисплеи [Электрон. ресурс]. – Режим доступа: <https://arduinoomania.ru/kakie-by-vayut-arduino-lcd-displei/>.

46. OLED дисплеи и Arduino [Электрон. ресурс]. – Режим доступа: <https://arduino-diy.com/arduino-oled-display>.

47. OLED дисплей 0.96" I2C 128x64 (білий) [Электронный ресурс]. – Режим доступа: <https://arduino.ua/prod1263-oled-displei-modyl-belii>.

48. Цветной графический TFT-экран 480×320 [Электрон. ресурс]. – Режим доступа: <https://amperka.ru/product/tft-color-display-480x320>.

49. Программирование устройств на основе модуля ESP32 [Электрон. ресурс]. – Режим доступа: https://habr.com/ru/company/epam_systems/blog/522730/.

50. Программирование модуля ESP32 в двухъядерном режиме с помощью Arduino IDE [Электрон. ресурс]. – Режим доступа: <https://microkontroller.ru/arduino-projects/programmirovanie-modulya-esp32-v-dvuhyadernom-rezhime-s-pomoshhyu-arduino-ide/>.

51. The Lua Programming Language Beginner's Guide [Electronic resource]. – Access mode: <https://www.bmc.com/blogs/lua-programming-language/>.

52. Как начать работу с Arduino Web Editor – руководство для начинающих [Электрон. ресурс]. – Режим доступа: <https://microkontroller.ru/arduino-projects/kak-nachat-rabotu-s-arduino-web-editor-rukovodstvo-dlya-nachinayushhih/>.

53. Программирование ESP32 в среде Arduino IDE. Структура кода. [Электрон. ресурс]. – Режим доступа: <https://arduino-tex.ru/news/99/programmirovanie-esp32-v-srede-arduino-ide-struktura-koda.html>.

54. Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux) [Electronic resource]. – Access mode: <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>.

Додаток А
Текст програми

A.1 Фрагмент тексту програми для станції передавання даних

```

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
#include "usbdrv/usbdrv.h"
#define u8 unsigned char
void main_cycle(void);
void RunTasks(unsigned char tasks)

{
    if (CheckBit(tasks,0))    usbPoll();
    if (CheckBit(tasks,1))    wdt_reset();
    if (CheckBit(tasks,2))    CopyToTxBuffer();
}

void timerDelayMs(unsigned int ms)
{
    #define TIKS_1MS (F_CPU/64/1000)
    while(ms--){
        TCNT0 = 0;
        while(TCNT0 < TIKS_1MS) RunTasks(0xFF);
    }
}

struct status_t    DeviceStatus;
static unsigned char currentAddress;
static unsigned char bytesRemaining;
struct
{

```

```

unsigned char rx_buffer[50]; // receive buffer
unsigned char tx_buffer[50]; // transmit buffer
}

usb;

#define USB_MESSAGE_LENGTH sizeof(DeviceStatus)

void writedata(void);
void usb_writedata(void){
    unsigned char i;
    for(i=0; i<sizeof(DeviceStatus); i++)
        ((unsigned char*)&DeviceStatus)[i] = usb.rx_buffer[i];
    for(i=0; i<sizeof(logical_outputs_PIN); i++)
    {
        if (CheckBit(DeviceStatus.logical_outputs, i)) sbi(*logical_outputs_PORT[i],
logical_outputs_PIN[i]);
        else cbi(*logical_outputs_PORT[i], logical_outputs_PIN[i]);
    }

    ADC=DeviceStatus.ADC_DATA;
    OCR1A = DeviceStatus.PWM1;
    OCR1B = DeviceStatus.PWM2;

    unsigned char *buffer = usb.rx_buffer;
    for(unsigned char i=0;i < len;i++)

    return bytesRemaining == 0; // 0 means there is more data
    else if(rq->bRequest == USBRQ_HID_SET_REPORT){
        bytesRemaining = USB_MESSAGE_LENGTH;
    }

```

```

else{ // other requests are simply ignored
DDRB = 0b00111111; PORTB = 0b00000000;
DDRC = 0b00000000; PORTC = 0b00111110;
DDRD = 0b11100000; PORTD = 0b00000011;
TCCR0 = (1<<CS01) | (1<<CS00); // prescaler = 64
TCNT0 = 0;
TCCR1A  =(1<<COM1A1)|(1<<COM1B1)| // Clear OC1A/OC1B on
Compare Match, set OC1A/OC1B at TOP
(1<<WGM11)|(1<<WGM10); // Fast PWM
TCCR1B =(1<<CS10) | // no prescaling
(1<<WGM12); // Fast PWM (10bit)
MCUCR = 0b01010000;
ADUX = 0b00100000;
ADCSRA=0b10000111;
usbInit();
usbDeviceDisconnect();
timerDelayMs(300);
usbDeviceConnect();
sei();
wdt_enable(WDTO_1S);
while(1)
{
{

main_cycle();
void main_cycle(void)
{

```

```

RunTasks(0xFF);
sbi(ADCSRA, ADSC); // Start ADC conversion
asm("SLEEP");// Standby mode (the processor itself is sleeping, but the ADC
is working)

```

```

RunTasks(0xFF);
unsigned char i;
for(i=0; i<sizeof(logical_outputs_PIN); i++) {
if (CheckBit(*logical_outputs_PORT[i], logical_outputs_PIN[i]))
}

```

```

RunTasks(0xFF);
for(i=0; i<sizeof(logical_inputs_PIN); i++) {
if (CheckBit(*logical_inputs_PORT[i], logical_inputs_PIN[i]))
sbi(DeviceStatus.logical_inputs, i);
else cbi(DeviceStatus.logical_inputs, i);
}

```

```

DeviceStatus.ADC_DATA = ADCH;
DeviceStatus.PWM1 = OCR1A;
DeviceStatus.PWM2 = OCR1B;
}

```

A.2 Фрагмент тексту програми для станції прийому даних

```

#include <vcl.h>
#pragma hdrstop
#include "main.h"
#include "hidlibrary.h"
#include "usbconfig.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_Main *Form_Main;
char vendorName[] = {USB_CFG_VENDOR_NAME, 0};
char productName[] = {USB_CFG_DEVICE_NAME, 0};

#define sbi(reg,bit) reg |= (1<<bit) // set bit
HIDLibrary <status_t> hid; // create an instance of the class
    stringexampleDeviceName = "";
    {

hid.Connect(i);
if ( hid.GetConnectedDeviceName() == exampleDeviceName )
{
res = 1;
break;
}

return res;
__fastcall TForm_Main::TForm_Main(TComponent* Owner)
: TForm(Owner)
{
void __fastcall TForm_Main::Timer_USBTimer(TObject *Sender)

```

```

{
if (!connect()) return; // Exit if the device is not connected
hid.ReceiveData(&DeviceStatus); // Read data from device
for (char i = 0; i < 7; i++)
}
void __fastcall TForm_Main::DeviceSet(TObject *Sender)
}
void __fastcall TForm_Main::Timer1Timer(TObject *Sender)
{
if (StrToInt(Edit1->Text)>0){
Edit1->Text = IntToStr(StrToInt(Edit1->Text)-1);
}
if (Edit1->Text <= 0 && StrToInt(Edit2->Text)>0){
Edit2->Text = StrToInt(Edit2->Text)-1;
Edit1->Text = 59; // Field "seconds" - unchanged
}
if (Edit2->Text <= 0 && StrToInt(Edit3->Text)>0){
Edit3->Text = StrToInt(Edit3->Text)-1;
Edit2->Text = 60;
}
if (Edit3->Text == 0 && // Seconds field
Edit2->Text == 0 && // "Minutes" field
Edit1->Text == 0) // Field "hours"
{ // if the values in the timer are 0, then
Timer1->Enabled=false;
}
void __fastcall TForm_Main::UpDown1Click(TObject *Sender,
TUDBtnType Button)
{

```

```

if(Button == Comctrl::btNext)
{ Edit1->Text = IntToStr(StrToInt(Edit1->Text)+1); }
if(Button == Comctrls::btPrev)
{ Edit1->Text = IntToStr(StrToInt(Edit1->Text)-1); }
}

void __fastcall TForm_Main::UpDown2Click(TObject *Sender,
TUDBtnType Button)
{
if(Button == Comctrl::btNext)
{ Edit2->Text = IntToStr(StrToInt(Edit2->Text)+1); }
if(Button == Comctrls::btPrev)
{ Edit2->Text = IntToStr(StrToInt(Edit2->Text)-1); }
}

void __fastcall TForm_Main::UpDown3Click(TObject *Sender,
TUDBtnType Button)
{
if(Button == Comctrl::btNext)
{ Edit3->Text = IntToStr(StrToInt(Edit3->Text)+1); }
if(Button == Comctrls::btPrev)
{ Edit3->Text = IntToStr(StrToInt(Edit3->Text)-1); }
}

void __fastcall TForm_Main::Button1Click(TObject *Sender)
{
Timer1->Enabled = true; // Enable timer
sbi(DeviceStatus.logical_outputs, 6);
if ( connect() ) hid.SendData(&DeviceStatus);

}

void __fastcall TForm_Main::Button2Click(TObject *Sender)

```

```
{  
Timer1->Enabled = false; // Turn off the timer  
Edit1->Text = "0"; // Set the value in the "seconds" field to 0  
Edit2->Text = "0"; // Set the value in the "minutes" field to 0  
Edit3->Text = "0"; // Set the value in the "hours" field to 0  
}
```

Додаток Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

ДИПЛОМНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА НА ТЕМУ:

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО МОНІТОРИНГУ МІКРОКЛІМАТУ МЕДИЧНИХ ПРИМІЩЕНЬ

Виконав
ст. гр. КНТ-218

М. С. Третяк

Керівник
к.т.н., доц.

А. В. Пархоменко

Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2

Мета та задачі роботи

Метою роботи є розробка програмного забезпечення для віддаленого моніторингу показників мікроклімату медичних приміщень на основі бездротової сенсорної мережі.

Задачі роботи:

- дослідити існуючі технології організації бездротових сенсорних мереж;
- обрати апаратні та програмні засоби розробки;
- виконати програмну реалізацію та тестування підсистеми віддаленого моніторингу показників мікроклімату;
- розробити програмну документацію.

3

Рисунок Б.3 – Слайд 3

Дослідження особливостей організації бездротових сенсорних мереж (БСМ)



БСМ для системи контролю показників мікроклімату

4

Рисунок Б.4 – Слайд 4

Вибір технології та модуля бездротової передачі даних

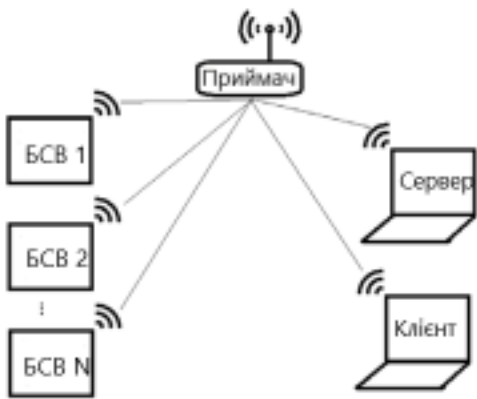
Параметр	Bluetooth	Zigbee	Wi-Fi
Діапазон частот	2.4 ГГц	868/915 МГц, 2.4 ГГц	2.4 ГГц, 5 ГГц
Максимальна швидкість сигналу	1 Мб/сек	250 Кб/сек	54 Мб/сек
Номинальна відстань	10 м	10-100 м	100 м
Пропускна здатність каналу	1 МГц	0.3-0.6 МГц, 2 МГц	22 МГц
Кількість радіочастотних каналів	79	1/10, 16	14 (2.4 ГГц)
Максимальна кількість вузлів	8	>65000	2007

Характеристика	ESP32	ESP8266
Процесор	Tensilica/DaveaLX6 32 bit Dual-Core at 160/240 MHz	TensilicaLX106 32 bit at 80 MHz (up to 160 MHz)
Статична ОП	512 Кб	160 Кб
Флеш-пам'ять	2Мб (max 64Мб)	4Мб (max 16Мб)
Напруга	2.2 В - 3.6 В	3.0 В - 3.6 В
Робочий струм	80 мА	80 мА
Програмованість	C, C++, Lua та ін.	C, C++, Lua та ін.
Відчутність	+	+
Wi-Fi	802.11 b/g/n	802.11 b/g/n
Bluetooth®	4.2 BR/EDR + BLE	-
UART	3	2
GPIO	32	17
SPi	4	2
PC	2	1
ШИМ	8	-
АЦП	18 (12-bit)	1 (10-bit)
ЦАП	2 (8-bit)	-
Розмір	25.5 x 18.0 x 2.8 мм	24.0 x 16.0 x 3.0 мм
Вартість, грн	214	160

5

Рисунок Б.5 – Слайд 5

Проектування підсистеми



Узагальнена архітектура підсистеми моніторингу на основі БСМ

Компоненти підсистеми

Вигляд	Назва
	Wi-Fi модуль LuaNode32 з ESP-32
	Датчик температури та вологості DHT22
	OLED дисплей 0.96" I2C 128x64

6

Рисунок Б.6 – Слайд 6

Вибір середовища розробки

Характеристика	Arduino IDE	PlatformIO
Простота	+	+
Підтримка	+	+
Платформа	Java	Python
Операційна система	Windows, OS X, Linux	Windows, macOS, Linux, FreeBSD
Менеджер бібліотек	-	+
Підтримка плагінів	-	+
Наявність вебверсії	+	-
Доступність	Open-source	Open-source

Відкриття скетчу для роботи з ESP32 в Arduino IDE

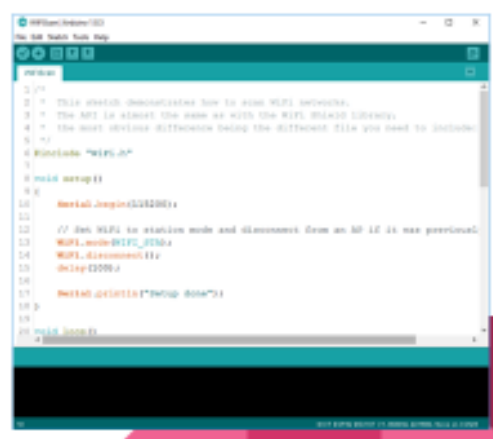


Рисунок Б.7 – Слайд 7

Конструювання підсистеми

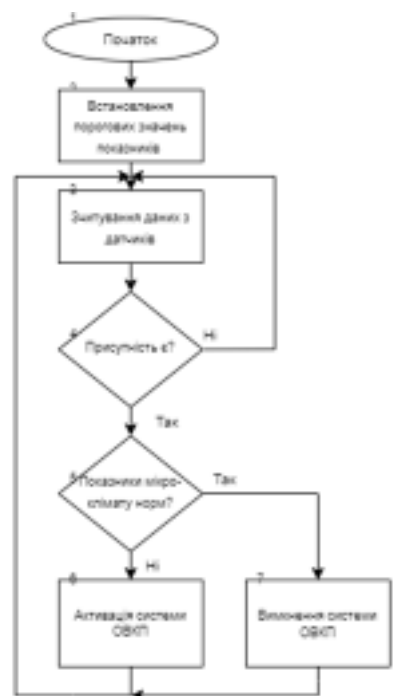


Схема алгоритму функціонування програми

Використані бібліотеки

Бібліотека	Призначення
DHT.h	Для організації програмної взаємодії датчика вологості і температури з модулем
ESP32 BLE Arduino	Передача даних між модулями вузла-передавача та вузла-приймача за допомогою Bluetooth з'єднання
Adafruit_SSD1306.h	Бібліотека, що відповідає за роботу дисплея
Adafruit_GFX.h	Бібліотека для шрифтів
Wire.h	Бібліотека для послідовної передачі даних

Рисунок Б.8 – Слайд 8

Зовнішній вигляд прототипу підсистеми



Рисунок Б.9 – Слайд 9

Результати тестування підсистеми

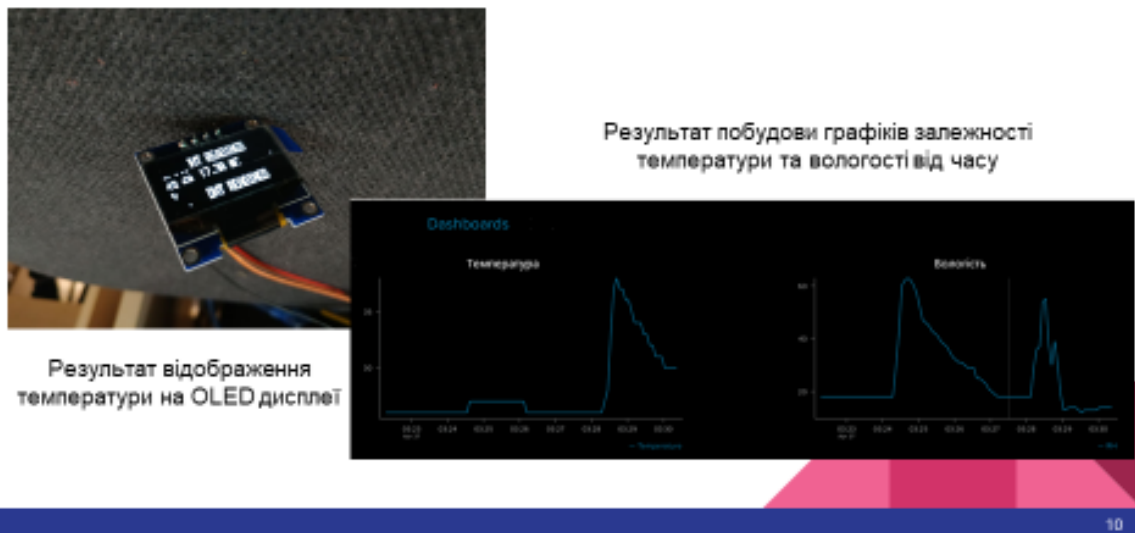


Рисунок Б.10 – Слайд 10

Висновки

- В роботі було проведено проєктування та конструювання підсистеми моніторингу показників мікроклімату медичних приміщень. Основними компонентами передавальної та приймальної частин підсистеми є модулі LuaNode32 з ESP-32, OLED дисплей та датчик вологості та температури DHT22. Додатково передбачено використання датчика присутності для виявлення знаходження людей у приміщенні та більш раціонального використання енергоносіїв. Програмне забезпечення розроблено на мові програмування C++ в середовищі розробки Arduino IDE.
- Перевагою розробленої підсистеми є те, є те що вона може виступати прототипом для подальшого розвитку та удосконалення подібних рішень. Вона легко збирається, має низьке енергоспоживання, що дозволяє їй працювати значний час в автономному режимі. Також ця система є мобільною завдяки використанню бездротових технологій.
- Розроблена підсистема може бути інтегрована в систему опалення, вентиляції та кондиціонування повітря медичних приміщень, або в системи домашньої та офісної автоматизації.

11

Рисунок Б.11 – Слайд 11

Дякую за увагу

12

Рисунок Б.12 – Слайд 12