

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумний

будинок»

Назва теми

КвРКІ. 190138.19.01.22 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва


Виконав: студент IV курсу, група KI2-19-1

  
Підпис

В. О. Письменюк

Ініціали, прізвище


Керівник

  
Підпис, дата

Д. М. Медзатий

Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

С.М. Лисенко

Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Т.О. Говорущенко

Ініціали, прізвище

«29» травня 2023 р.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорущенко

“ 11 ” 01 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Письменюку Володимиру Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумний будинок»

Керівник проекту (роботи) Медзатий Д. М., к.т.н., доцент.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 1.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Теоретичні основи досліджуваної проблеми

Проектування підсистеми керування мікрокліматом у кіберфізичній системі «Розумний будинок»

Програмно –апаратна реалізація підсистеми керування мікрокліматом у кіберфізичній системі «Розумний будинок»

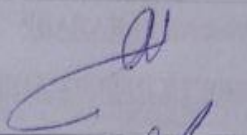



5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Апаратна складова підсистеми керування мікрокліматом

Алгоритм роботи підсистеми керування мікрокліматом

Інтерфейс проекту мобільного додатку при роботі підсистеми

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		


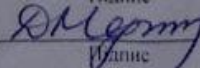
7. Дата видачі завдання « 01 » 03 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2023	виконано
3	Робота над розділом 1 – теоретичні основи досліджуваної проблеми	01.03.2023	виконано
4	Робота над розділом 2 – проектування підсистеми керування мікрокліматом житла у кіберфізичній системі «Розумний будинок»	01.04.2023	виконано
5	Робота над розділом 3 – програмно-апаратна реалізація підсистеми керування мікрокліматом житла у кіберфізичній системі «Розумний будинок»	15.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	22.05.2023	виконано
7	Попередній захист ВКР	26.06.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент

Керівник проекту (роботи)

  
Підпис  
  
Підпис

В. О. Письменюк

Ініціали, прізвище

Д. М. Медзатий

Ініціали, прізвище



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумний будинок».

Автор роботи: Письменюк Володимир Олександрович.

Керівник роботи: Медзатий Дмитро Миколайович

Пояснювальна записка: 65 с., 66 рис., 1 табл., 4 дод., 73 джерела.

Графічна частина: презентаційні слайди.

СИСТЕМА «РОЗУМНИЙ БУДИНОК», ARDUINO UNO, ДАТЧИКИ, ПРИЗНАЧЕННЯ, ЕЛЕМЕНТИ, ПІДКЛЮЧЕННЯ, РОЗРОБКА, ПРОГРАМА, СТВОРЕННЯ.

*Мета дослідження* – розроблення підсистеми керування мікрокліматом житла у кіберфізичній системі «Розумний будинок»

*Об'єкт дослідження* – кіберфізична система «Розумний будинок»

*Предмет дослідження* – підсистема керування мікрокліматом житла

*Практичне значення:* Розроблена підсистема керування мікрокліматом житла спрямована на полегшення життя людей, так як вона економить електроенергію, ефективно очищує повітря від вуглекислого газу, а також підтримує задані користувачем параметри мікролімата таких як: вологість та температура повітря. Використання енергонезалежної пам'яті плати Arduino UNO дозволяє підсистемі набути стійкості при аварійних ситуаціях, пов'язаних з електроенергією. Застосування розробленого мобільного проекту надає можливості для легкого та більш точного керування даною розробкою віддалено за допомогою телефона. Результати роботи можуть використовуватись для подальшого розвитку систем «Розумний будинок».



Підпис студента

29.05.23

Дата

## ЗМІСТ

ВСТУП.....	4
1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ .....	5
1.1 Аналіз предметної області і виявлення наявних проблем та завдань.....	5
1.2 Порівняльний аналіз переваг та недоліків існуючих рішень .....	10
1.3 Постановка задачі.....	18
2 ПРОЕКТУВАННЯ ПІДСИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТОМ ЖИТЛА У КІБЕРФІЗИЧНІЙ СИСТЕМІ «РОЗУМНИЙ БУДИНОК».....	20
2.1 Обґрунтування вибору мови програмування .....	20
2.2 Обґрунтування вибору середовища для програмування.....	25
2.3 Обґрунтування вибору середовища для програмування.....	27
2.4 Вибір датчика вологості та температури .....	34
2.5 Обґрунтування вибору середовища для програмування.....	35
2.6 Вибір сервомотору .....	37
2.7 Вибір каналного вентилятора.....	38
2.8 Вибір модуля двоканального реле.....	40
2.9 Вибір Bluetooth модуля та програми для телефону .....	41
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТОМ ЖИТЛА У КІБЕРФІЗИЧНІЙ СИСТЕМІ «РОЗУМНИЙ БУДИНОК» .....	43
3.1 Підключення модулів до плати Arduino UNO .....	43
3.2 Програмна реалізація підсистеми керування мікрокліматом житла .....	48
3.3 Створення проекту мобільного додатку та тестування підсистеми	
ВИСНОВОК.....	68
СПИСОК ПОСИЛАНЬ .....	69
ДОДАТОК А Лістинг програмної реалізації підсистеми керування мікрокліматом житла .....	76
ДОДАТОК Б Копія креслення «Апаратна складова підсистеми керування мікрокліматом».....	81

КвРКІ 190138.19.01.22 ПЗ								
Зм.	Арк.	Недокум.	Підпис	Дата	Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумний будинок»	Літера	Арквш	Аркувшів
Виконав		Письменок В.О.	<i>[Підпис]</i>	29.05		у		64
Перевір.		Медзатий Д.М.	<i>[Підпис]</i>	29.05				
Н.контр.		Лисенко С.М.	<i>[Підпис]</i>	29.05				
Затвер.		Говорушенко Т.О.	<i>[Підпис]</i>	29.05			ХНУ КІ2-19-1	

ДОДАТОК В КОПІЯ КРЕСЛЕННЯ «АЛГОРИТМ РОБОТИ ПІДСИСТЕМИ КЕРУВАННЯ МІКРОЛІМАТОМ».....	82
ДОДАТОК Г КОПІЯ КРЕСЛЕННЯ «ІНТЕРФЕЙС ПРОЕКТУ МОБІЛЬНОГО ДОДАТКУ ПРИ РОБОТІ ПІДСИСТЕМИ».....	83

					КВРКІ 190138.19.01.22 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

Протягом останніх років автоматизація інформаційних технологій впевнено посіла важливе місце в нашому житті. Вона поширюється майже на всі сфери людського життя, полегшуючи його. Це дає змогу забезпечувати часткову, а й інколи повну автоматизацію. Тому й не дивно, що такі технології з'явилися в будинках людей, місцях, де більшість людей проводять вільний час.

Можливість керувати електричними приладами за допомоги одного пульта або, навіть, голосових команд – надто привабливо для кожної людини. Також варто не забути про ще одну перевагу автоматизованої системи «Розумний будинок» - здатність відображати інформацію про поточний стан, що надає можливість людині відреагувати на надзвичайну ситуацію: пожежу, витік газу тощо.

Тому в останні часи тема автоматизованого будинку набуває все більшої популярності, а великі компанії розглядають можливості вийти на ринок зі своїм продуктом.

					КВРКІ 190138.19.01.22 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

## ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

### 1.1 Аналіз предметної області і виявлення наявних проблем та завдань

Часто, коли ми міркуємо про автоматизований дім, то здебільшого це про автоматичне вимикання світла, коли лягаємо спати, а також закриття вікон та дверей. В літературі, фантасти представляли собі дім, заповнений роботами, які доглядають за приміщенням, або дім, яким людина може керувати за допомоги мозку.

У 1898 році Нікола Тесла створив прилад, який міг на відстані керувати транспортними засобами та судами, що можна назвати першим пристроєм віддаленого контролю. Цей прилад виглядав як невелика за розмірами радіо передавальна коробка і був продемонстрований у Медісон-сквер-гарден, де винахідник керував моделлю корабля і його сигнальними вогнями без будь-якого підключення, що викликало великий інтерес та поставило питання щодо більших можливостей застосування дистанційного керування [1] – [6].

Але варто згадати, що ще на початку ХХ століття люди вручну прали одяг, що потребувало тяжких зусиль, багато часу та великої кількості води. Тому, побутові електричні пристрої, що з'явилися у 1915-1920 роках показали готовність суспільства замінити домашній персонал більш дешевими механічними пристроями, але через проблему з енергозбереженням, були доступні лише заможним людям.

Завдяки електричним пристроям, почав зростати рівень автоматизації в людських оселях.

У 1905 році британський виробник – Walter Griffiths, випустив «Покращений вакуумний апарат Гріффіта для видалення пилу з килимів» і був першим електричним пилососом та виглядав дуже схожим на сучасні аналоги [7] – [8].

У 1907 році компанія Hurley Electric Laundry Equipment розробила та представила електричну пральну машину «Thor-32», що вважається першим справжнім приладом автоматизації дому, зовнішній вигляд якої на рисунку 1.1.

					КвРКІ 190138.19.01.22 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 – Реклама електричної пральної машини «Thor-32» [9]

У 1929 році німецька компанія «Miele» представила першу посудомийну машину на електричному двигуні. Прилад був мобільний, адже мав колеса на яких його було легко перевести в потрібне місце, при необхідності, а завантаження посуду відбувалося через відкидну кришку. Нажаль, прилад не зустрів захоплення серед користувачів через погану якість миття посуду та високу вартість [10].

Зовнішній вигляд даного приладу можливо побачити на рисунку 1.2.

Далі у 1938 році було випущено першу електричну сушарку для білизни, а також в подальших роках відбувалось покращення електричних приладів та виробництво нових, наприклад, холодильників.

Ці пристрої досить сильно вплинули на життя людей, хоча й їх неможливо назвати «розумними», але вони дали потужний поштовх у напрямі автоматизації житла.

Зм.	Арк.	№ докум.	Підпис	Дата



Рисунок 1.2 – Реклама електричної посудомийної машини [10]

Перший дім, який можливо назвати «розумним», створив американський інженер-винахідник Еміль Матіас у 1950 році, що описує стаття у грудневому випуску цього ж року у «Popular Mechanics».

Будівля була розташована в Джексоні, штат Мічиган та отримала назву «Кнопковий дім». Для його створення було використано 2133 метри дроту, а також велика кількість реле, кнопок та двигунів, які створювали можливість дистанційного управління воротами гаражу, освітленням, побутовими пристроями, різними механізмами та шторами. Всі дроти та електричні кабелі були сховані в будівельній конструкції, тому було неможливо знайти будь-який дріт, який буде звисати. Також оселя мала систему сигналізації та безпеки.

Вона насправді відповідала своїй назві «Кнопковий дім», адже все управління контролюється за допомогою кнопок і більшість повсякденних завдань можливо вирішити лише одним натисканням. Наприклад, Еміль створив для своєї дружини туалетне дзеркало, яке при відкритті кофоду, світиться, а також штори, які закриваються та відкриваються автоматично [11] – [12].

					КВРКІ 190138.19.01.22 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

При написанні статті, винахідник розповів журналісту, що в нього є ще плани щодо створення механічного офіціанта для вечірок та автоматичного поливу газону, хоча й більшість винаходів були на той момент незавершеними.

На рисунку 1.3 зображено планування «Кнопкового дому».

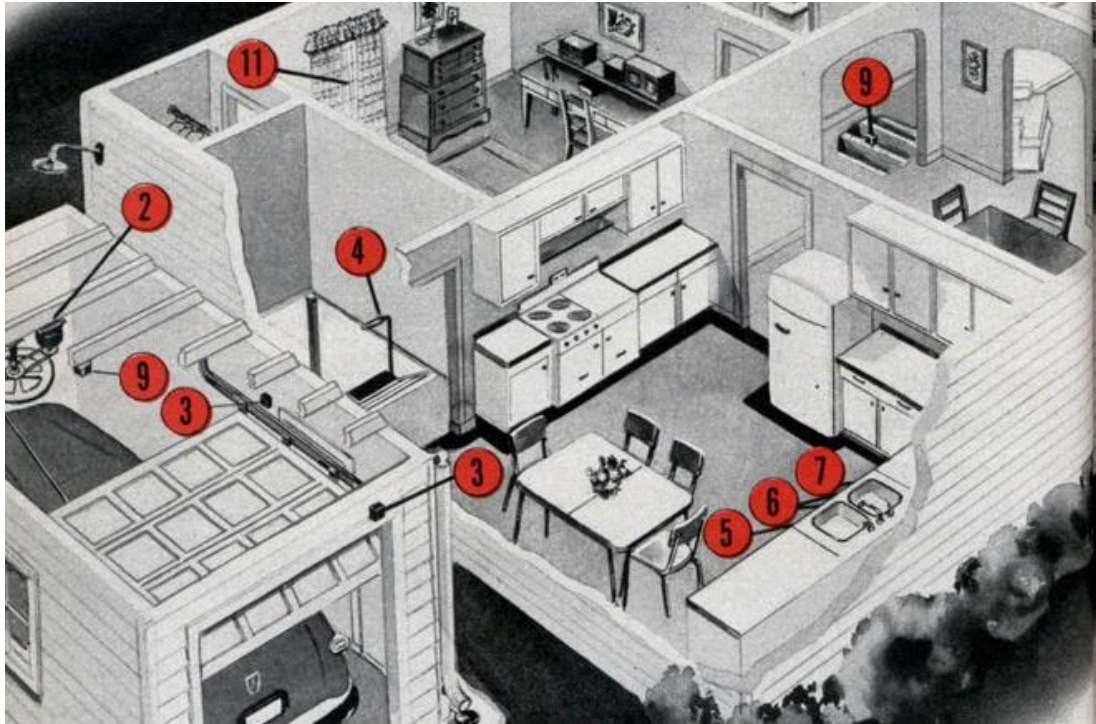


Рисунок 1.3 – Планування «Кнопкового дому» [12]

Будування та проектування першої інтегрованої системи «Розумний дім» Джеймсом Сазерлендом вважається найвизначнішою подією в історії такого типу систем. Спроектвана система мала назву «Echo IV», за суттю індивідуальним ПК, що дозволяв контролювати температуру в кімнатах будинку, отримувати доступ до прогноза погоди, вимикати та вмикати пристрої, а також створювати цифрові списки покупок.

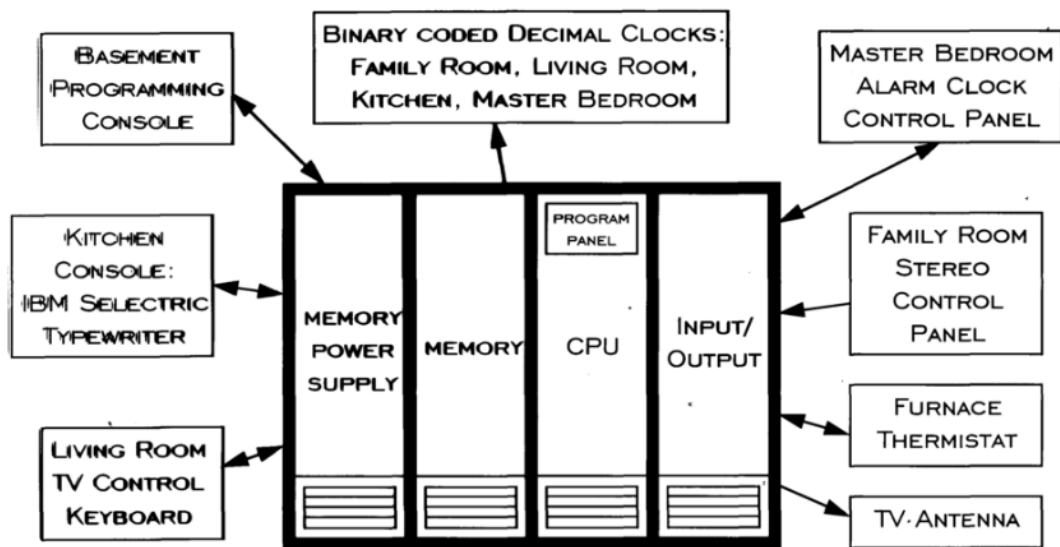
Система мала такі особливості, діаграму якої видно на рисунку 1.4:

- календар;
- управління кондиціонером;
- бухгалтерський облік;

Зм.	Арк.	№ докум.	Підпис	Дата

- метеорологічна програма, підключена до метеостанції для прогнозу погоди;
- господарський інвентар;
- управління телебаченням;
- управління через будинок цифровими годинниками.

## ECHO-IV SYSTEM DIAGRAM



MAIN BASEMENT CABINET: 6' HIGH, 7' WIDE, 1.5' DEEP

Рисунок 1.4 – Діаграма системи «Echo IV» [1]

Система Джеймса Сазерленда мала великий потенціал, а також показувала на що здатні технології «Розумний будинок», але випередила свій час і через свою дорожнечу не мала успіху в комерції.

У 1971 році був винайдений мікроконтролер, що дало зниження цін на електроніку, а електронні пристрої стали більш доступними. Це дало змогу переглянути переваги систем «Розумний дім» та рухатися в напрямку покращення та оптимізації таких будинків.

## 1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Техніка і датчики розміщені у будівлі та об'єднанні у єдину систему, яка підтримує керування та налаштування ними за допомогою сенсорної панелі, планшета, смартфона або комп'ютера - називається «Розумний будинок» (з англ. Smart house). Така технологія повинна позбавити людину від рутини, автоматизуючи повсякденні справи [14] – [16].

Кожна людина може підлаштувати систему під свої потреби, але виділяють такі основні функції «Розумного будинку» [17]:

- захист від непередбачуваних ситуацій. Система надає змогу мінімізувати шкоду або повністю захиститися від загорянь, затоплень, дій кradіїв та злодіїв, а також витоків газу;

- енергозбереження та економічність. «Розумний будинок» контролює споживання газу, електроенергії та води, коли вони використовуються без потреби, що знижує не тільки рахунки за комунальні послуги, а й ризики затоплення сусідів або інших аварійних ситуацій;

- забезпечення підвищеного комфорту. Автоматизація дому дозволяє користувачам витратити менше часу та зусиль в повсякденному житті. Наприклад, автоматичне ввімкнення світла за темної пори, а також контроль температури під час холодних та спекотних днів.

Так як система має досить великий спектр функцій, то потребує певну кількість пристроїв декількох типів [18]:

- сенсори та датчики. Цей тип компонентів надає інформацію системі про зовнішні умови, такі як: рівень освітленості, температури повітря, герметичного закриття дверей та вікон, а також наявності диму та руху;

- виконавчі пристрої. На дану групу приходиться найбільша кількість приладів. Вона здійснює керування певними домашніми приладами і несе відповідальність за виконання різних команд. Також таку групу пристроїв ще

					КВРКІ 190138.19.01.22 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

називають актуаторами. Сюди відносяться клімат-контролери, клапани для труб, вимикачі, розумні розетки та різні реле;

- центр управління, який ще називають шлюз, міст, хаб або контролер. Це пристрій, який надає змогу віддалено керувати роботою системи, а також об'єднує елементи системи «Розумний будинок» в єдине ціле [19].

Системи можуть підтримувати також деякі додаткові компоненти, але вони не дуже важливі, так як їх відсутність може компенсувати підтримка голосових команд.

Так як «Розумний будинок» має багато функцій, пристроїв, то в системі виділяють такі підсистеми [20] – [21]:

- освітлення. Ця підсистема відповідає за виключення та включення груп світильників або окремої лампи. Тут налаштовується інтенсивність освітлення, а також встановлюється необхідний користувачу режим;

- клімат контроль. Підсистема керує обігрівачем, зволожувачем повітря, вентиляцією та кондиціонером. За допомогою термостату, датчиків вологості та температури тощо, створює комфортні параметри для життя, а також забезпечує енергозбереження;

- безпека. Заради забезпечення контролю та реагування на загрози майну та інших злочинів у будинку та межах подвір'я, встановлюються камери спостереження, датчики руху, вібрацій та відкриття. Також можливо створити контроль доступу при налаштуванні домофону та електронних замків;

- мультимедіа. Сюди відноситься будь-які пристрої, які відтворюють аудіо-, відео-сигнали в розважальних цілях, наприклад, телевізор або звукова колонка. Тут можливе налаштування техніки на вимкнення та включення за часом, подією або датою.

Підсистеми, зазвичай, представляються як одне ціле системи «Розумний будинок», але, з часом, технології все більше розвиваються та стали більш гнучкі, тому заради зручності виробники надають можливості встановлення лише

					КВРКІ 190138.19.01.22 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		



потреб, тобто керувати підсистемами безпеки, клімату, мультимедіа. Системи вузького профілю часто потребують в професійному середовищі, адже вони створені для роботи в окремих галузях, наприклад, освітлення або забезпечення безпеки;

3. за протоколами. Розрізняють системи із закритими та відкритими протоколами. У першому типі, в систему «Розумний будинок» можна інтегрувати лише прилади, які випустив виробник цієї системи. У другому випадку, будь-який виробник може створити для цієї системи прилад, що надає більший вибір користувачу при покупці, а також підвищує універсальність;

4. за способом передачі даних виділяють бездротові та дротові системи. Бездротовий спосіб передачі реалізується за допомоги радіосигналів. В дротовому типі, основні складові системи взаємодіють через дрот.

Бездротові системи є високо популярними в побутових цілях, адже такі системи прості в оновленні, монтажі та модифікації, а сучасні розробки все більше підвищують стандарти стабільності, захищеності та економічності.

Вирізняють чотири бездротові стандарти зв'язку, які використовуються в системі «Розумний будинок»:

- Bluetooth;
- Wi-Fi;
- ZigBee;
- Z-Wave.

Протоколи ZigBee та Z-Wave використовуються через багато свої переваг, таких як: мінімальні затримки сигналу, низьке споживання електроенергії, недорогі модулі, а також надання прямого зв'язку між усіма вузлами системи, без використання хабу або мосту.

При компонуванні системи «Розумний будинок» пристроями від різних виробників, створюються проблема несумісності. Хоча й дані протоколи розроблялись для вирішення цього питання, на даний момент, вони як і Wi-Fi та Bluetooth потребують хаб. Якщо бажання щодо встановлення цього пристрою

немає, залишається лише варіант облаштування будинку пристроями одного стандарту.

Система базована на дротовому підключенні майже немає затримки сигналів, захищена від втрати даних, а також безпечнішою, ніж бездротове підключення, тому є надійнішою. Також існує великий недолік, який виражається в складнішій реалізації у вже готових приміщеннях, через потребу в прокладанні дроту.

На сьогоднішній день системи вузького профілю з закритими протоколами користуються попитом у сфері безпеки, а найпоширеніші системи «Розумний будинок» - широкий профіль з відкритими протоколами та централізованим типом керування з бездротовим підключенням.

Найпопулярніші системи «Розумний будинок» на ринку наочно демонструють описані раніше тенденції.

#### 1. Apple HomeKit

Одна з перших заявлених систем на ринку, від компанії Apple. В 2014 році, тобто при випуску на ринок, вона типу з закритим протоколом, що не давало додавати та розпізнавати сторонні пристрої, без проходження ними сертифікації від компанії Apple. На сьогодні всі сумісні прилади проходять сертифікацію, що дає гарантії якості, а також не обтяжує покупців складнощами, адже процедура була спрощена у 2016 році.

Після встановлення програми «Home» на телефон або інший пристрій, користувачу надається можливість керування системою «Розумний будинок». Дана програма вирізняється простим і зручним інтерфейсом, створеним на базі iOS, а також швидкістю інтегрування приладів у систему.

Покупцям також пропонують голосового сервіс-помічник «Siri», що характеризується підтримкою багатьох мов, великим вибором голосових команд та чіткістю їх сприйняття. У системі передбачена можливість віддаленого керування з будь-якої точки світу, але для цього потребується розумна колонка Apple HomePod, Apple TV або iPad.

					КВРКІ 190138.19.01.22 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток зображень на рисунку 1.6.

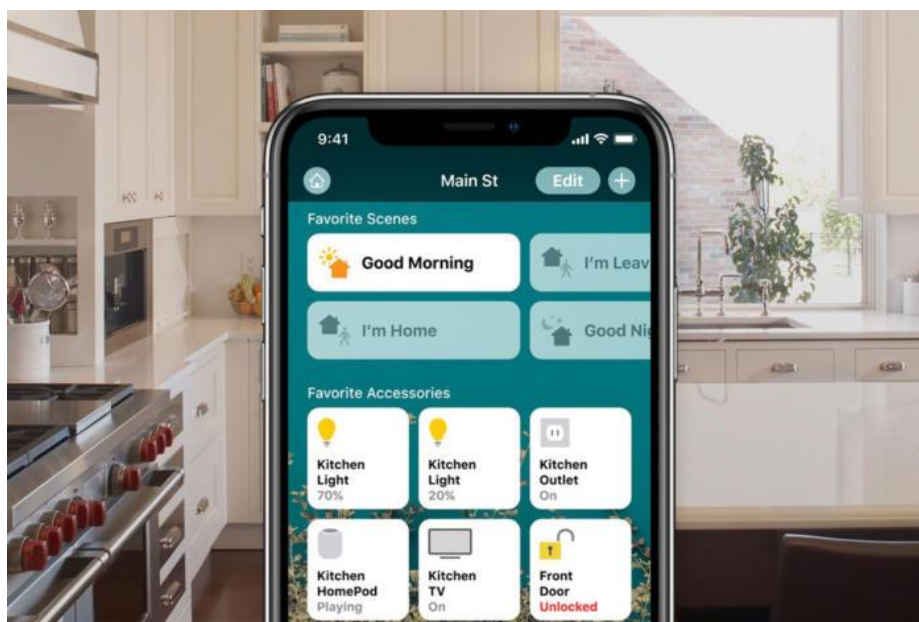


Рисунок 1.6 – використання додатку «Home» для керування системою Apple HomeKit [23]

З недоліків вирізняють високу ціну на сумісні пристрої та невелику їх кількість.

## 2. Amazon Alexa

Вважається, що презентація розумної колонки Amazon Echo дала поштовх для розвитку бездротових систем «Розумний будинок», адже даний портативний пристрій став першим, що оснащений голосовим помічником «Alexa». Даний помічник може керувати домашніми приладами, відповідати на прості питання, що збільшує можливості колонки, окрім прослуховування музики. На сьогоднішній день з Amazon Alexa сумісні розумні розетки Belkin, лампочки Tr-Link та Philips Hue, автомобілі BMW Mini та Ford та багато іншого, адже з самого початку систему представляли як відкриту.

Центральним керуючим пристроєм системи можуть виступати модифікації та дебютна версія колонки, а також її аналоги випущені сторонніми виробниками, що мають підтримку голосового помічника від компанії Amazon.

					КВРКІ 190138.19.01.22 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

Налаштування відбувається з мобільного додатку, який підтримують усі операційні системи, а зв'язок з приладами відбувається через Bluetooth або Wi-Fi.

Через відсутність локалізації, адже система підтримує лише англійську мову, а також таких можливостей як поповнення банківського рахунку, покупки квитків, виклику Uber, що використовуються користувачами цього «Розумного будинку» у США - у нашій країні ця система не є популярною.

Розумна колонка Amazon Echo представлена на рисунку 1.7.



Рисунок 1.7 – показ «розумної колонки» Amazon Echo, її модифікації Echo Dot та голосового помічника Amazon Alexa

### 3. Google Home

Так як компанія-виробник позиціонувала даний «Розумний будинок» як конкурент системі від компанії Amazon, був представлений продукт у вигляді розумної колонки Google Home, а також підтримкою голосового помічника від компанії Google – Google Assistant. Хоча й пристрій вийшов пізніше, ніж конкурент, окрім кращого голосового помічника, він мало чим відрізнявся від Amazon Alexa. У 2018 році Google надала користувачам оновлений додаток для

мобільних телефонів, у якому стало можливо керування різними пристроями, підключеними до Google Assistant, навіть без використання розумної колонки.

Також система підтримує досить великий вибір центральних керуючих, а також інших пристроїв від різних виробників.

Недоліки продукту виражаються у досить великій затримці, що становить одну-дві секунди, при застосуванні додатку для управління пристроями, а також у простоті можливих виконаних командах у порівнянні з іншими аналогами системи «Розумний будинок».

Розумна колонка від компанії Google зображена на рисунку 1.8.



Рисунок 1.8 – розумна колонка Google Home

На відміну від Amazon Alexa, у Google Home має у комплекті кращого помічника, що демонструється у більшій чіткості сприйняття команд, а також варіацій мов для роботи.

#### 4. Ajax

Система створена для забезпечення безпеки, українською командою розробників. Технологія являється з закритими протоколами та безпроводним

підключенням. Керування відбувається через центр управління Ajax Hub, до якого прив'язують різні датчики: виявлення затоплення, протипожежної безпеки, а також бездротові сенсори безпеки, наприклад, відкриття вікон та дверей, руху, розбиття скла.

Також компанія забезпечує користувачів мобільним додатком, який забезпечує не тільки сповіщення системи, а й можливість перегляду відео з камер спостереження.

Ajax пройшла сертифікацію, де довела спроможність протистояння вмілим зловмисникам зі спеціальним обладнанням, а у 2017 році отримала нагороду в номінації «Охорона система року».

Так як це система безпеки, компанія-виробник надає гарантії автономної роботи до 15 годин без енергоспоживання, захисту датчиків від глушіння зв'язку та сигналу, а також до 7 років праці без заміни акумуляторів або батарейок.

«Розумний будинок» має підтримку декількох облікових записів та одночасного підключення до півтори сотні пристроїв.

### 1.3 Постановка задачі

Завданнями роботи є:

- дослідити процедури функціонування системи «Розумний будинок»;
- провести теоретичний аналіз сфери;
- охарактеризувати структуру предметної області та базову модель;
- описати уже існуючі механізми реалізації, виділити наявні проблеми в галузі та шляхи їх вирішення;
- на основі проведених досліджень визначити основні функції системи, сформулювати низку функціональних та нефункціональних вимог, розробити модель функцій, які система повинна виконувати;
- підвести підсумки про необхідність розробки системи;

					КВРКІ 190138.19.01.22 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

- сформулювати об’єкт та мету для наступних досліджень;
- оцінити ступінь виконання поставлених завдань.

На основі цього розробити працездатну підсистему керування мікрокліматом у кіберфізичній системі «Розумний будинок» та зробити висновки на основі виконаної роботи.

					КВРКІ 190138.19.01.22 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

# ПРОЕКТУВАННЯ ПІДСИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТОМ ЖИТЛА У КІБЕРФІЗИЧНІЙ СИСТЕМІ «РОЗУМНИЙ БУДИНОК»

## 2.1 Обґрунтування вибору мови програмування

При створенні будь-якої системи, виникає потреба надати їй функціональність та можливості, тобто запрограмувати, використовуючи певні мови програмування. Застосування цих мов залежить від наявності їх підтримки в апаратній частині проекту, а також доцільності та складності використання.

Розрізняють такі мови програмування, які використовуються при роботі з мікрокомп'ютерами [24] – [25]:

- Assembly – також ще називають «Асемблер». Вона належить до мов програмування низького рівня, тобто є мнемонічним позначенням машинних команд процесора. Тому, ця мова є досить тяжкою для розуміння та вивчення людиною та громіздкою для підтримання [26] – [27].

На рисунку 2.1 представлено найпростішу програму на даній мові.

```
1 global _main
2 extern _printf
3
4 section .text
5 _main:
6 push message
7 call _printf
8 add esp,4
9 ret
10 message:
11 db 'Hello, World', 10,0
```

Рисунок 2.1 – приклад найпростішої програми на Assembly

До переваг відносять можливість чіткого налаштування підпрограм для додатків з точною синхронізацією, через надання доступу до центрального процесора. З більшим розповсюдженням подібних керуючих плат до Arduino, які використовують мови програмування високого рівня, почав застосовуватись все рідше, але не був витіснений повністю.

- MicroPython – мова програмування, яка реалізує Python на мікроконтролерах та інших платах.

Так як Python була і є одною з найпопулярніших мов, вона охоплювала майже усі сфери життя від настільного програмного забезпечення до статистичного аналізу, окрім сфери мікроконтролерів. Тому, у 2013 році завдяки ентузіасту Демієну Джорджу ця мова була адаптована до меншої версії себе – MicroPython, що заповнило прогалину в області програмування мікроконтролерів. З цього слідує, що при вивченні Python не виникає проблеми спеціально навчатися MicroPython. Також MicroPython завдяки інтерпретатору, який виконує дії в такій послідовності: зчитування введеної команди, оцінювання та виконання команди, при наявності вивід на консоль, повертається назад та повторює процес(Read-Eval-Print-Loop), забезпечує швидкий відклик та є достатньо потужний для використання в промисловості. На відміну, від Асемблера та інших конкурентів, цю мову є більш доступною для новачків, що гарантує не тільки менша складність, а й великою аудиторією розробників та спільнотою «Pythonistas» [28] – [29].

На рисунку 2.2 представлено найпростішу програму на MicroPython.

```
1 print("Hello, World!")
```

Рисунок 2.2 – приклад найпростішої програми на MicroPython

- Blockly – бібліотека програмування, яка підтримується компанією Google. На відміну, від звичайних мов програмування, які використовують певні слова та словосполучення, відповідно вимог документації мови програмування, для створення додатків та програм, ця бібліотека застосовує блоки, які з'єднуються

разом, так утворюючи програму. При використанні Blockly блоки не дозволяють з'єднатися з іншими несумісними блоками. Це забезпечено за рахунок специфічних з'єднань, що нагадують пазл, цим забезпечуючи майже повну відсутність помилок. Після цього дані блоки, які утворили програму, можливо перевести на декілька інших мов, наприклад, PHP, Dart, Python та JavaScript [31].

Приклад програми видно на рисунку 2.3.

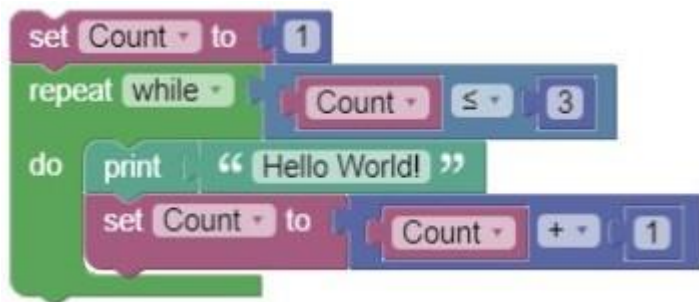


Рисунок 2.3 – приклад програми на Blockly

- C/C++ – мови програмування, які можуть застосовуватися для виконання майже всіх завдань через свою властивість контролю як в низькорівневих мовах програмування, а зручність написання, як у високорівневих. Вони чудово працюють при програмуванні мікроконтролерів та плат, адже мова програмування C надає можливість виконувати складні функції, наприклад, використання математичних виразів, операторів умовності та циклів. Окрім цього, варто зазначити про функцію, яка є дуже необхідною при роботі з мікроконтролерами - виділення пам'яті, адже ці пристрої не мають її у великій кількості. Хоча й мова наділена багатьма перевагами, вона є непростю в написанні через нетерпимість до помилок, а також складний синтаксис, що призводить до певних складнощів на початку вивчення.

На рисунках 2.4 та 2.5 буде зображено найпростіші програми «Hello World» на мовах програмування C та C++ відповідно, для розуміння складності синтаксису та розуміння способу роботи цих мов.

```
1 #include <stdio.h>
2 int main(void)
3 {
4     printf("hello, world\n")
5 }
```

Рисунок 2.4 – найпростіша програма «Hello World» на мові програмування C

```
1 #include <iostream>
2 int main()
3 {
4     std::cout<<"Hello, world!\n";
5     return 0;
6 }
```

Рисунок 2.5 – найпростіша програма «Hello World» на мові програмування C++

Також варто згадати про Arduino C++, який неможливо назвати самостійною мовою програмування, так як ця мова написана на простому C++ з певним додаванням методів, функцій та бібліотек, які специфічні для використання платою Arduino. Бібліотечні функції C++ не можуть працювати з обмеженою пам'яттю, як у мікроконтролерів, тому було вирішено створити підпрограми призначені для цих пристроїв. Arduino C++ дещо нагадує MicroPython, адже також є спрощеною версією мови програмування та розроблений спеціально для мікроконтролерів, хоча й тільки типу Arduino. Програмісти, які раніше займалися написанням коду на мові C++, також можуть без проблем перейти на використання Arduino C++. Тому, можливо дійти до висновку, що дана мова програмування характеризується високою швидкістю роботи, а й доступністю [32].

На рисунку 2.6 зображено програму «Hello World» на раніше Arduino C++.

```

1 void setup() {
2     // setup code here, that is runing once:
3     Serial.begin(9600);
4     Serial.println("Hello world!");
5 }
6
7 void loop() {
8     // main code, that is runing repeatedly:
9
10 }

```

Рисунок 2.6 – програма «Hello World» на мові програмування Arduino C++

Тому, на основі наявної інформації можна створити таблицю 1, яка допоможе в аналізі та виборі мови програмування для створення підсистеми керуванням мікрокліматом.

Таблиця 2.1 – Аналіз доцільності використання мови програмування

№	Мова програмування	Складність	Досвід роботи з мовою	Пристрої, які підтримують	Доцільність використання
1	Assembly	Складно	Відсутній	Raspberry PI	Недоцільно

## Продовження таблиці 2.1

№	Мова програмування	Складність	Досвід роботи з мовою	Пристрої, які підтримують	Доцільність використання
2	MicroPython	Нормально	Наявний, але менше ніж Arduino C++	Системи на базі ARM: ESP8266, ESP32, BBC micro :bit, Adafruit Circuit Playground Express, Raspberry PI та інші	Недоцільно
3	Blockly	Легко	Відсутній	BBC micro :bit, Adafruit Circuit Playground Express	Недоцільно
4	Arduino C++	Нормально	Наявний	Вся лінійка контролюючих плат Arduino	Доцільно

Так як в роботі більш доцільно вибрати мову програмування Arduino C++, тому в подальшому будуть розглядатись середовища програмування, які підтримують дану мову та контролюючі плати Arduino.

## 2.2 Обґрунтування вибору середовища для програмування

Для виконання роботи потрібно вибрати середовище в якому буде відбуватись написання коду, який потім буде завантажений на плату. Також виникає необхідність, щоб це середовище підтримувала можливість програмування на Arduino C++ та працювала з керуючими платами Arduino.

Для цього було обрано Arduino IDE. Це інтегроване середовище розробки (Integrated Development Environment) розроблене компанією Arduino.cc і представлено для всіх охочих на офіційному сайті. Середовище дозволяє програмувати плати Arduino, створюючи скетчі (файли з кодом), компілювати та завантажуючи їх на плату та її модулі. Дана IDE працює на програмній платформі Java, що надає влаштовані команди та функції, які дуже важливі при на всіх етапах програмування: редагування, налагодження та компіляції коду[33] – [35].

Arduino IDE є дуже поширеним серед багатьох людей, адже воно легке для використання не тільки досвідченими користувачами, а й новачками. Також домагає у поширенні продукту те, що він є з відкритим програмним кодом, надаючи можливості користувачам для створення нових бібліотек, модифікацій або інших функціональностей. Такий вид ліцензії також дозволяє виправляти помилки в програмі, роблячи її кращою та більш зручною для інших. Велику роль також грає те, що компанія Arduino.cc створила продукт, який доступний на більшості платформ: Mac OS, Windows та Linux та залишила його безкоштовним.

Так як це середовище пропонує розроблювати проекти користувачам будь-якого рівня, важливо мати зручний та простий для розуміння інтерфейс.

Інтерфейс Arduino IDE зображений на рисунку 2.7.

Середовище повністю задовольняє усі потреби та є оптимальним для використання у розробці проекту.

					КВРКІ 190138.19.01.22 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

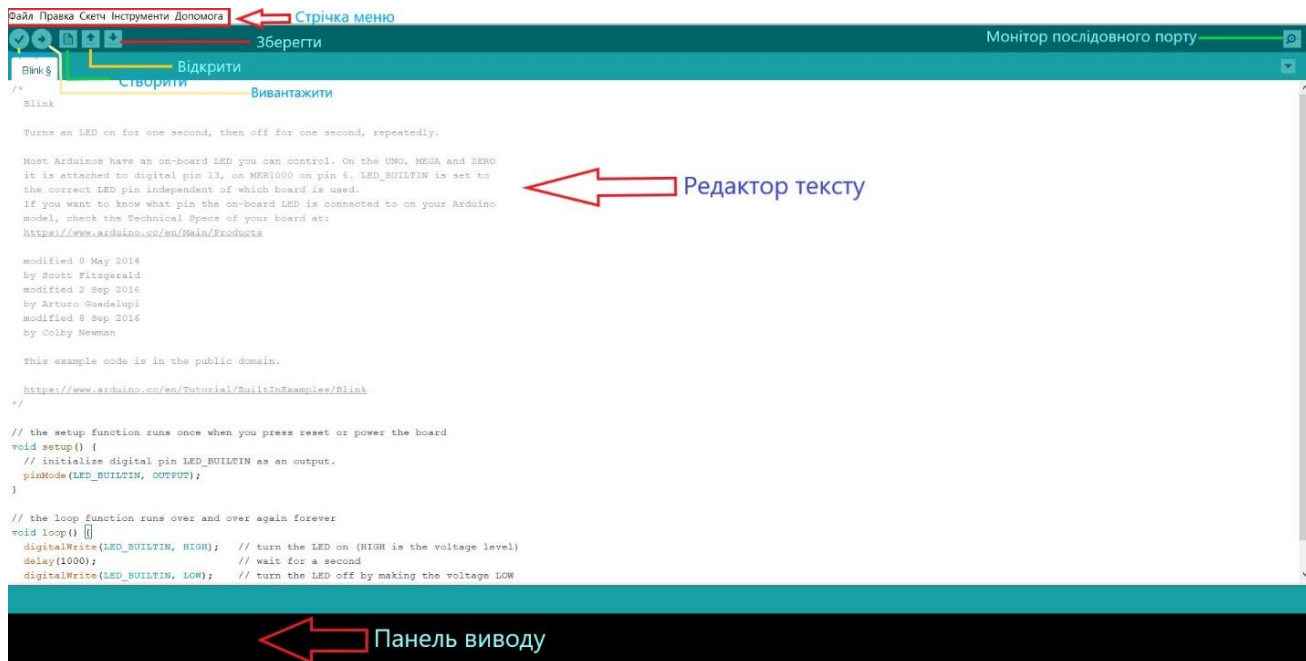


Рисунок 2.7 – Зовнішній вид інтерфейсу Arduino IDE та призначення кнопок

### 2.3 Обґрунтування вибору середовища для програмування

Arduino.cc надає не тільки середовище програмування, а й плати, які програмуються на Arduino C++. Ці плати можуть слугувати для багатьох завдань, наприклад, надаються можливість сконструювати систему управління мікрокліматом в будинку, або створити власний клімат контроль для автомобіля та інші речі, які не тільки корисні в побуті, а комерційних цілях.

Для створення власного пристрою або системи потрібно лише під'єднати потрібні компоненти: сенсори, мотори та кнопки, що будуть надавати можливість виконання певних дій таких, як ввімкнення мотору або запалювання світлодіоду та обертання оприскувачів, під час поливу газону.

Розрізняють такі додаткові компоненти:

- перемикачі та кнопки – дозволяють створити певні шаблони або сценарії поведінки для пристроїв та цілої системи, що надає можливість керування цими складовими проекту.

На рисунку 2.8 зображено приклад компонента «кнопка» [36].



Рисунок 2.8 – Зовнішній вигляд компонента «кнопка»

- датчики – відстежують певні параметри середовища та їх зміни, а потім передають цю інформацію на керуючу плату. Arduino надає можливість встановити сенсори для більшості потреб, які тільки можуть виникнути. В цей спектр входять вимірювачі рівня води, температури, вологості, світла та багато інших.

Варто зазначити, що існують складніші датчики, які підтримує Arduino, наприклад, датчик розбиття скла, або сенсор відстані.

На рисунку 2.9 зображено приклад компонента датчика освітленості [37].

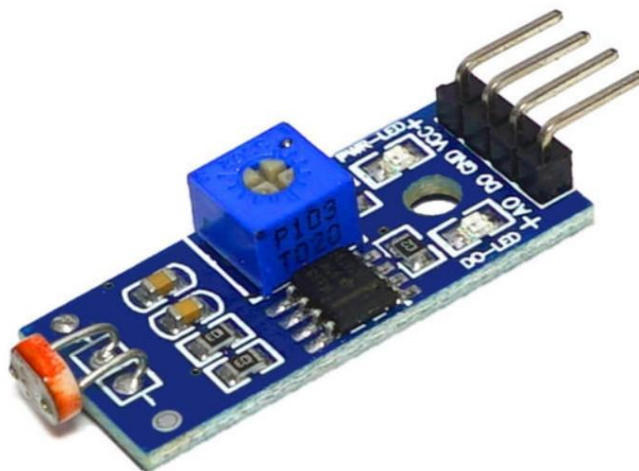


Рисунок 2.9 – Зовнішній вигляд датчика освітленості(світлорезистор)

Зм.	Арк.	№ докум.	Підпис	Дата

- резистори – наявні майже у всіх проектах, адже вони забезпечують електричний опір. Також інколи ці елементи використовуються для налаштування або регулювання чимось, виступаючи у ролі важелю управління.

На рисунку 2.10 зображено приклад компонента «резистор» [38].



Рисунок 2.10 – Зовнішній вигляд резистора

- макетна плата та проводи – використовуються для складання розроблювального проекту без застосування паяльника. Створення пристрою відбувається таким чином, що компоненти мають бути вставлені в отвори плати.

На рисунку 2.11 зображено зовнішній вигляд макетної плати та пояснюється призначення контактів.

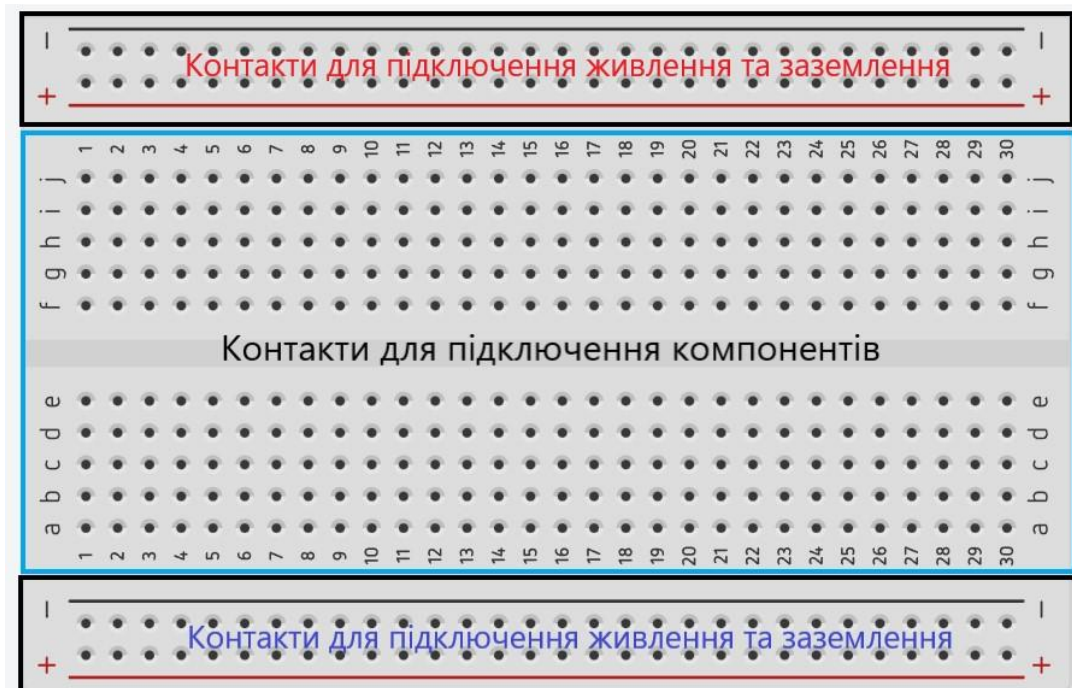


Рисунок 2.11 – Зовнішній вигляд макетної плати та призначення контактів

- LED – також кажуть «світлодіод». Починає випромінювати світло, коли отримує сигнал. Застосовується для перевірки роботи в проекті.

На рисунку 2.12 зображено приклад компонента «LED» [39].



Рисунок 2.12 – Зовнішній вигляд компонента «LED»

- дисплеї або екрани – використовуються для відображення інформації. Дисплеї бувають двох видів: рідкокристалічний (LCD), а також рідкокристалічний на тонкоплівкових транзисторах (TFT LCD).

На рисунку 2.13 зображено зовнішній вигляд рідкокристалічного дисплею.

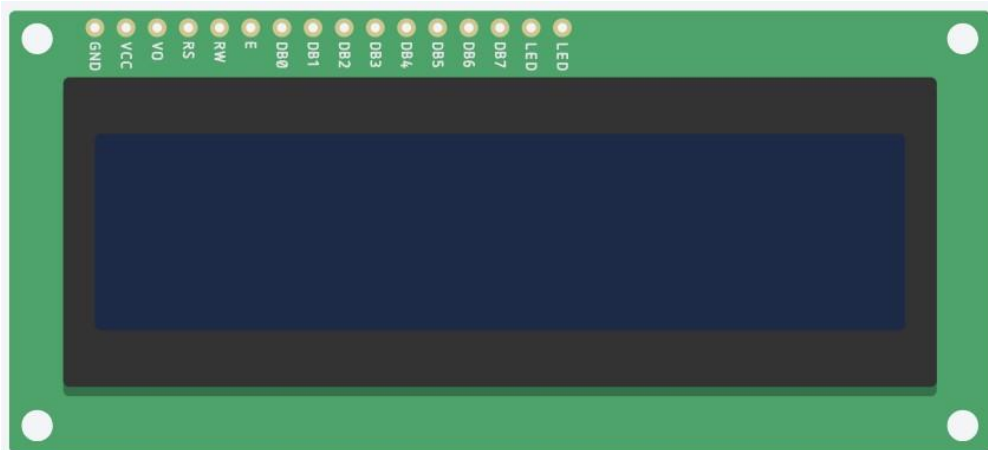


Рисунок 2.13 – Зовнішній вигляд рідкокристалічного дисплею

- сервомотори – ці компоненти необхідні для руху частин системи або конструкції, відповідно до заданих параметрів користувачем.

На рисунку 2.14 зображено приклад сервомотору [40].



Рисунок 2.14 – Зовнішній вигляд сервомотору

Після вибору та додавання необхідних компонентів до плати, необхідно написати програму для описання роботи модулів, і вивантажити даний «скетч» на плату.

Було вирішено для виконання проекту вибирати плату з лінійки Arduino.

Одна із плат, яка належить до лінійки Arduino – Arduino UNO. Вона є найпоширенішою контролюючою платою. Для цього є декілька причин. По-перше, ця плата є досить потужною платформою для використання у різних проектах. По-друге, це перша USB-плата, яка представила компанія Arduino.cc [41] – [43].

Через наявність USB-порту, у платі присутня можливість підключитися за допомоги USB-кабелю до комп'ютера, що дозволяє використовувати даний кабель, як спосіб живлення, а також послідовний порт. Також Arduino UNO має роз'єм, який запрограмований на основі інтегрованого середовища розробки і може працювати як на офлайн платформах, так і на онлайн. Це роз'єм з технологією програмування, яка дозволяє запрограмувати модуль або компонент, який вже встановлений в електронний пристрій і має назву ICSP(In Circuit Serial Programming). На контролюючій платі також присутні роз'єм для живлення, 14 цифрових входів/виходів та 6 аналогових.

					КВРКІ 190138.19.01.22 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

З рисунку 2.15, до елементів розміщені на платі Arduino UNO належать:

- світлодіоди TX та RX – використовуються для показу прийому/передачі даних між платою та комп'ютером через USB-кабель, тобто коли відбувається обмін інформацією дані світлодіоди починають миготіти;
- цифровий вхід/вихід – призначений для підключення до плати датчиків, сервомоторів та інших компонентів. Дані входи/виходи мають лише два значення: HIGH та LOW, які відповідають логічній одиниці або логічному нулю;
- USB – застосовується для програмування плати з комп'ютеру, підключаючись до останнього через USB-кабель;
- аналоговий вхід – використовується для зчитування інформації з аналогових датчиків, а також може замінити цифровий роз'єм при їх нестачі;
- кварцовий генератор – має підтримувати постійну частоту 16 MHz;
- GND – даний роз'єм призначений для заземлення, адже напруга ньому становить 0 V;
- роз'єм ICSP – дозволяє запрограмувати модуль або компонент, який вже встановлений в електронний пристрій, використовуючи влаштоване програмне забезпечення плати;
- Vin – порт, який дозволяє подавати напругу на зовнішні джерела живлення, а також прямо підключений до регулятора напруги плати. Має також функцію, яка допускає приймати живлення від зовнішніх джерел живлення, але це не рекомендується;
- регулятор напруги – перетворює вхідну напругу в 5 V, а зайва відводиться у вигляді тепла;
- кнопка перезавантаження – при натисканні, плата на короткий час очікує на завантаження програмного файлу, а далі відбувається виконання коду з самого початку. Можливо сказати, що ця кнопка створює ефект швидкого ввімкнення та увімкнення керуючої плати;
- мікроконтролер ATmega328 – містить у собі аналогово-цифровий перетворювач, таймер, реєстри, потри SPI та пам'ять [44];

					КВРКІ 190138.19.01.22 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

- світлодіодний індикатор живлення – використовується для інформування користувача про стан під'єднання живлення: світлодіод світиться при підключеному, а при відключеному - знаходиться у стані спокою;

- RESET – це контакт, який має аналогічний функціонал, як кнопка перезавантаження. Використовується при неможливості натиснення цієї кнопки;

- AREF – означає «Analog Reference». На цей контакт можливо надати напругу для аналогово-цифрового перетворювача на мікроконтролері.

Також варто зазначити, що при створенні критичних помилок або ситуацій, які можуть призвести до згоряння Arduino UNO, це ніяк не буде впливати на комп'ютер до якого ця плата була підключена, а через відносно малу ціну її можливо замінити.

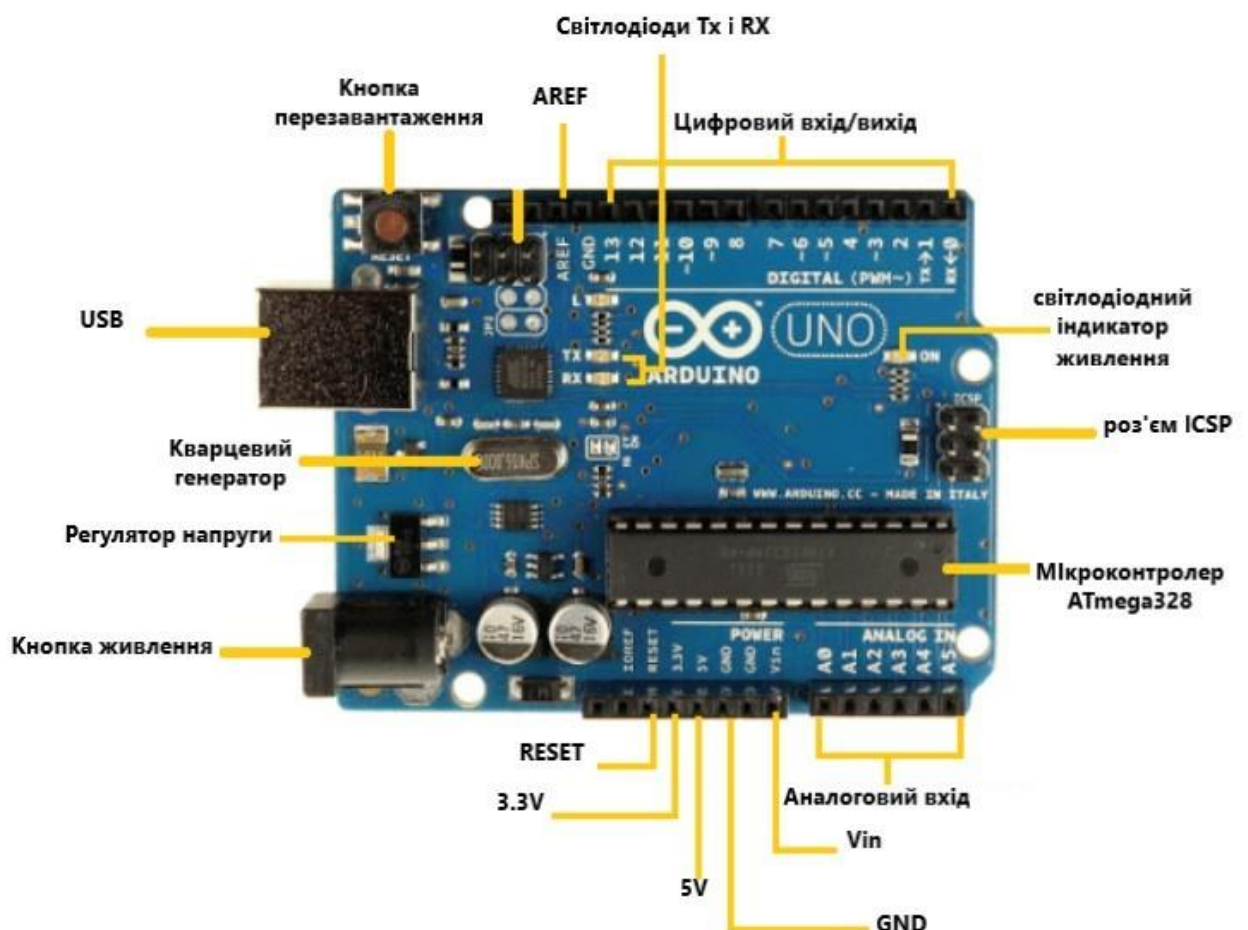


Рисунок 2.15 – Позначення елементів плати Arduino UNO

Беручи до уваги, багато переваг, якими наділена плата Arduino UNO, було вирішено застосувати цей пристрій, при роботі над системою керування мікрокліматом, яка буде реалізуватись за допомогою телефону, який буде слугувати дисплеєм для відслідковування показників, які будуть надсилати на сенсори, а також вентиляторів з сервомоторами, які будуть відсмоктувати вуглекислий газ, надлишкові вологість та тепло через отвори вентиляції.

В наступних кроках потрібно обрати додаткові компоненти, які допоможуть надати функціонал для реалізації даної ідеї.

#### 2.4 Вибір датчика вологості та температури

Для вимірювання температури та вологості буде використовуватись датчик DHT11.

Зовнішній вигляд цього сенсору зображено на рисунку 2.16.

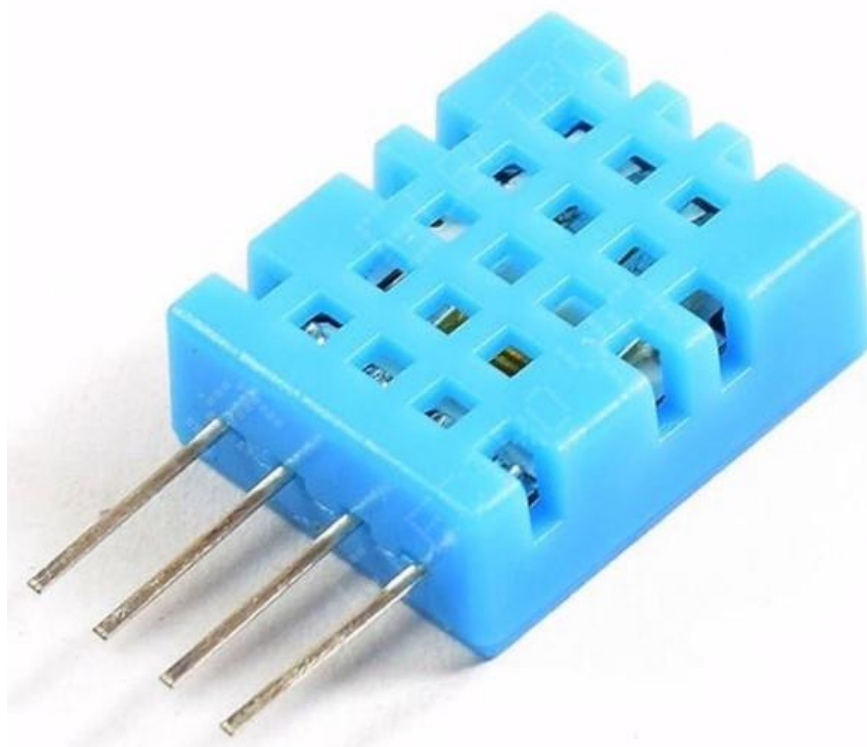


Рисунок 2.16 – Датчик DHT11

Моделі DHT є доволі простими та задовільняють потреби у вимірюванні даних. Так як сенсор зчитує навколишню температуру та вологість, що є двома параметрами мікроклімату, тому він складається з чіпа, який перетворює аналоговий сигнал, отриманий від термістора та датчика вологості, що розташовані у корпусі DHT11, у цифровий сигнал, котрий зчитується мікроконтролером [45] – [46].

Характеристики датчика DHT11:

- живлення відбувається у межах від 3 до 5 Вольт. Таку напругу може надати плата Arduino UNO, що підходить для використання у проєкті;
- робить вимірювання вологості з точністю до 5 відсотків з межами від 20% до 80% вологості, а температурні межі знаходяться від 0 С° до 50 С° з точністю до 2 С°, що робить оптимальним для застосування у розробці підсистеми, яка встановлюється всередині будинку;
- низьке енергоспоживання;
- можливість запиту один раз у секунду допомагає краще контролювати роботу підсистеми керування мікрокліматом, роблячи її більш точнішою;
- ціна становить 58 грн.

Існує також схожий сенсор - DHT22, але його інтервали між запитами даних більші, ніж у DHT11. Датчик DHT22 характеризується великим корпусом, більшими діапазонами для вимірювання, але температури більше 50 С° у приміщеннях не існує. Беручи до уваги, що ціна сенсору DHT22 становить 199 грн., що додатково вказує на недоцільність використання датчика цієї моделі.

## 2.5 Обґрунтування вибору середовища для програмування

Створимо можливість реагувати системі на наявність великої кількості вуглекислого газу, що буде впливати негативно на самопочуття людей, що знаходяться у приміщенні. Тому, варто встановити датчики вуглекислого газу.

					КВРКІ 190138.19.01.22 ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		

Для реалізації цього завдання, підходить датчик МН-Z19В. Його дозволяється використовувати при температурах від 0 С° до 50 С°, а вологості до 90%, що співпадає з раніше описаними умовами роботи сенсора DHT11.

МН-Z19В має певні особливості, наприклад, при зміні концентрації вуглекислого газу, датчик потребує близько однієї хвилини для обрахунку точних результатів. Додатково, підготовка до роботи при отримання живлення у датчика займає три хвилини і під час неї, результати вимірювань будуть некоректними [47] – [50].

Сенсор підключається до плати через PWM входи, що розташовані на платі. На Arduino UNO це - третій, п'ятий, шостий, десятий, одинадцятий контакти, тобто ті, які мають знак «-» перед порядковим номером.

Також, цей модуль характеризується невеликими розмірами, що можливо побачити на рисунку 2.17.

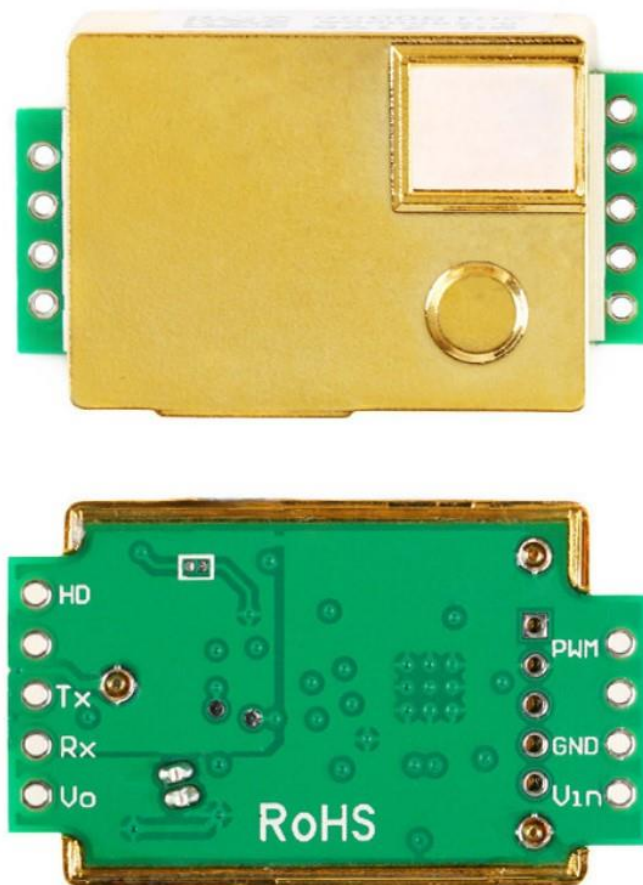


Рисунок 2.17 – Вигляд датчика МН-Z19В зверху та знизу

Отже, датчик МН-Z19В повністю сумісний з платою Arduino UNO, має досить багато переваг за оптимальну ціну, а також гарантію роботи п'ять років, що пояснює його вибір для використання у розробці.

## 2.6 Вибір сервомотору

Так як є датчики, які показують температуру, вологість та вуглекислий газ, потрібна певна реакція на незадовільні значення.

Для цього застосовуються сервомотор, який буде повертати решітку в вентиляційних отворах, щоб вентилятор міг краще виконувати свою роботу по видаленню надлишків.

Модель SG90 характерна своїми малими розмірами та невеликою вагою. Даний модуль може живитися від джерел, що й плата, наприклад, USB-порт, але рекомендують окремий блок живлення, спеціально для сервомоторів.

На рисунку 2.18 представлено зовнішній вигляд даної моделі модуля.

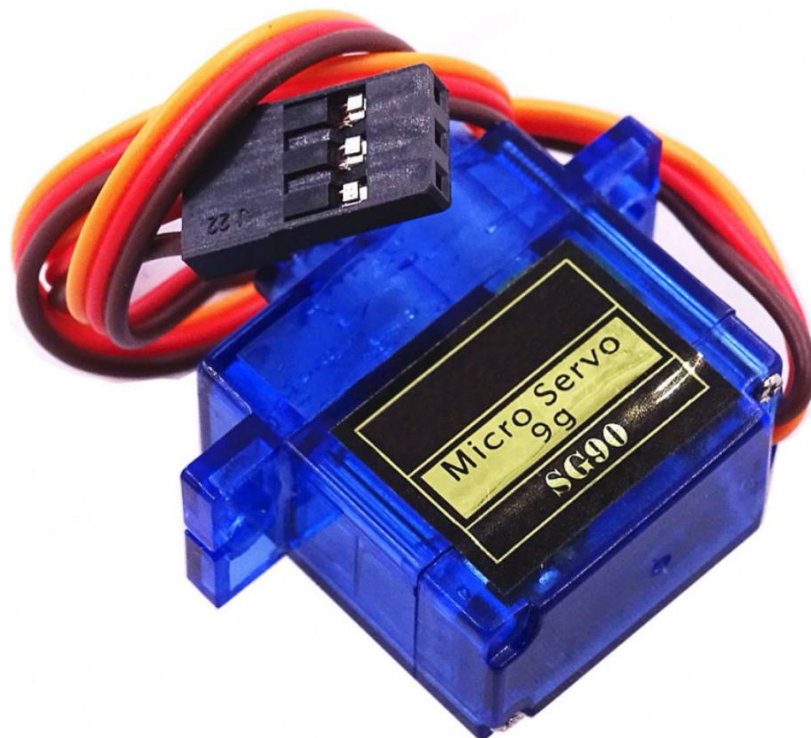


Рисунок 2.18 – Зовнішній вигляд сервомотору SG90

Кут повороту становить від  $0^\circ$  до  $180^\circ$ , а шаблон руху задається програмою, яка була розроблена користувачем. Також можливо налаштувати швидкість обертання [51] – [52].

Так як, сервомотор SG90 відповідає потребам, а ще й коштує лише 86 грн. , було вирішено обрати дану модель.

## 2.7 Вибір каналного вентилятора

Для регулювання кількості вуглекислого газу, вологості та температури потрібно встановити каналні вентилятори.

Дані пристрої розміщуються у вентиляції, де при потребі будуть відповідати за зменшення раніше описаних параметрів повітря.

Спосіб роботи каналного вентилятора та місце розташування зображено на рисунку 2.19.



Рисунок 2.19 – Спосіб роботи каналного вентилятора та місце розташування

При розробці проекту було прийнято рішення використовувати модель вентилятора AirRoxy aRc 100 S [53] – [55].

На рисунку 2.20 зображено зовнішній вигляд каналного вентилятора даної моделі.



Рисунок 2.20 – Канальний вентилятор AirRoxy aRc 100 S

Модель AirRoxy aRc 100 S має характеристики:

вологостійкість;

знижений рівень шуму – рівень шуму становить 45 дБ, що надає можливість для встановлення у побутових та комерційних приміщеннях;

два режими роботи – стабільний та періодичний;

продуктивність вентилявання повітря становить 103 м<sup>3</sup>/ годину;

ціна складає 690 грн.

Встановити даний вентилятор можливо самостійно, а також цей пристрій не вимагає технічного обслуговування, окрім очищення пилу.

Тому, зважаючи на багато переваг, а також оптимальність використання у проєкті, була обрана модель AirRoxy aRc 100 S.

## 2.8 Вибір модуля двоканального реле

Двоканальне твердотіле реле використовується для управління приладами, які ввімкнені у мережу 220 Вольт.

Так як в проєкті застосовані каналні вентилятори, які живляться від розетки, то виникає потреба у виборі даного типу модуля.

На відміну від механічного, твердотіле реле вирізняються беззвучною роботою, а також більшими строком служби та безпечністю. Більший термін роботи досягається за допомоги відсутності рухомих частин, які можуть зношуватись, а безпечність забезпечує відсутність фізичного контакту, який може утворити іскру [56] – [58].

Реле Omron G3MB-202P дозволяє плавно збільшувати струм навантаження, що збільшує строк служби підключених пристроїв.

Даний тип модуля продемонстрований на рисунку 2.21.

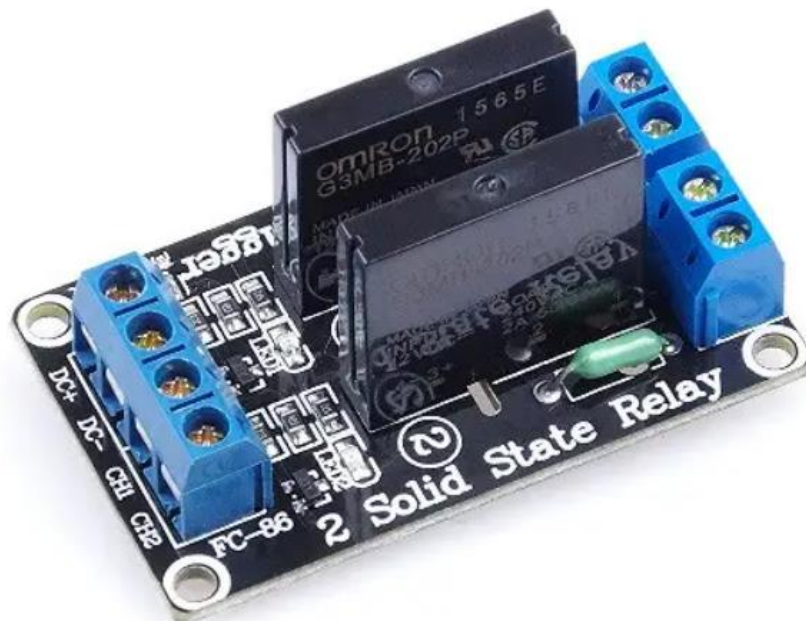


Рисунок 2.21 – Двоканальний модуль з твердотілим реле Omron G3MB-202P

					КВРКІ 190138.19.01.22 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

Так як твердотіле реле Omron G3MB-202P має багато переваг, а ціна становить 97 грн., було вирішено застосувати у проєкті саме цю модель.

## 2.9 Вибір Bluetooth модуля та програми для телефону

Для більшої зручності було вирішено використовувати в розробці проєкту Bluetooth модуль. Це дозволить отримувати сповіщення при певних умовах або просто відкривати програму, яка відображає показники підсистеми мікроклімату, навіть, без потреби вставати з ліжка.

Модель HC-06 виконує функцію підключення до плати Arduino UNO, або інших пристроїв, за допомоги Bluetooth. Модуль надає просте та надійне безпроводне підключення, при роботі у пасивному режимі, тобто від головного(управляючого) пристрою до приймача, який визначений. Можливо налаштувати швидкість передачі інформації, ім'я модуля, а також пароль для підключення. Налаштування відбувається за допомоги AT-команд і буде продемонстровано в подальшому.

На корпусі розташований світлодіод, який інформує чи модуль зв'язаний з будь-яким пристроєм, а ще показує стан підключення.

На рисунку 2.22 зображено зовнішній вигляд Bluetooth модуля HC-06 [59] – [61].



Рисунок 2.22 – Зовнішній вигляд Bluetooth модуля HC-06

					КВРКІ 190138.19.01.22 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

Так як було визначено Bluetooth модуль, який буде використовуватись при розробці, потрібно вирішити, якою програмою краще користуватися для прийняття даних від контролюючої плати.

Програма «Bluetooth Electronics» від британської компанії, дозволяє створювати електронні проекти, які надають можливості контролювати та спостерігати за змінами під час роботи розроблених систем, підключившись до Bluetooth модуля HC-06 або HC-05.

Хоча й розробники віддають перевагу платам лінійки Arduino, що можливо спостерігати по більшості прикладів, які створені для них, дана програма дозволяє працювати з Raspberry Pi, якщо наявний Bluetooth модуль.

Зручний та зрозумілий інтерфейс надає можливості легко розробляти власний проект на будь-який смак.

Інтерфейс програми зображений на рисунку 2.20.

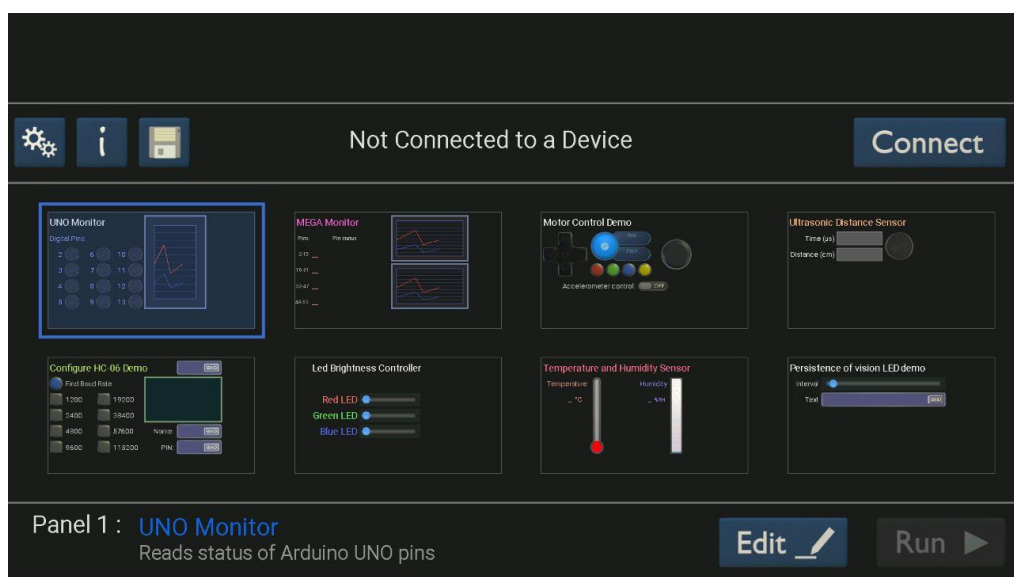


Рисунок 2.23 – Інтерфейс програми «Bluetooth Electronics»

У розділі було обрано мову програмування на якій буде виконуватись проект, на основі цього обрана плата Arduino UNO та додаткові компоненти, які будуть потрібні при проектуванні підсистеми керуванням мікрокліматом. Надалі потрібно підключити їх між собою, запрограмувати елементи на основі завдання, а також створити власний проект у додатку «Bluetooth Electronics» для спостереження та контролювання роботи підсистеми.

# ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТОМ ЖИТЛА У КІБЕРФІЗИЧНІЙ СИСТЕМІ «РОЗУМНИЙ БУДИНОК»

## 3.1 Підключення модулів до плати Arduino UNO

Після вибору додаткових модулів для створення підсистеми керування розумним будинком, потрібно підключити їх до керуючої плати Arduino UNO.

Для правильного під'єднання додаткових елементів використовується офіційна документація виробника, де буде вказано напругу та відповідне призначення входів та виходів модуля.

З рисунку 3.1 видно, що датчик вологості та температури DHT11 має чотири контакти [62] – [63]:

- Vcc – живлення з напругою 5 V;
- Data – використовується для передачі показників з датчика на плату.

Зазвичай, потребує підключення до цифрового входу;

- NC – цей контакт не підключений, так як він не застосовується та не вимагає офіційна документація сенсору;
- GND – роз'єм заземлення або ще його просто називають «земля».

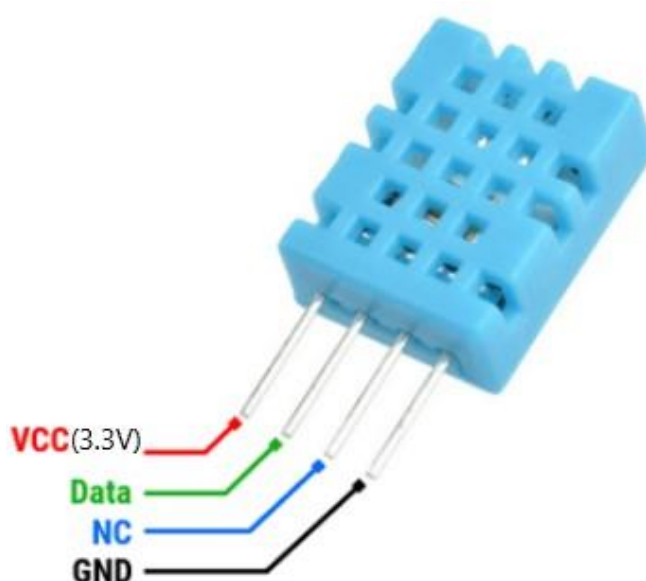


Рисунок 3.1 – Призначення контактів датчика DHT11

					КвРКІ 190138.19.01.22 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

Після підключення DHT11 потрібно під'єднати інший модуль – датчик вуглекислого газу MH-Z19B.

Даний сенсор та назви його контактів зображено на рисунку 3.2.

З рисунку видно, що сім контактів пристрою мають призначення [64]:

- $V_{in}$  – контакт вхідного живлення;
- $V_o$  – аналоговий вихід;
- PWM – контакт широко-імпульсної модуляції;
- GND – контакт заземлення;
- HD – застосовується для калібрування, а також встановлення нульової точки, яка встановлює відповідність концентрації вуглекислого газу до дійсного нульового значення (400 ppm);
- RX – UART(RXD)TTL Level data input;
- TX – UART(TXD)TTL Level data output;

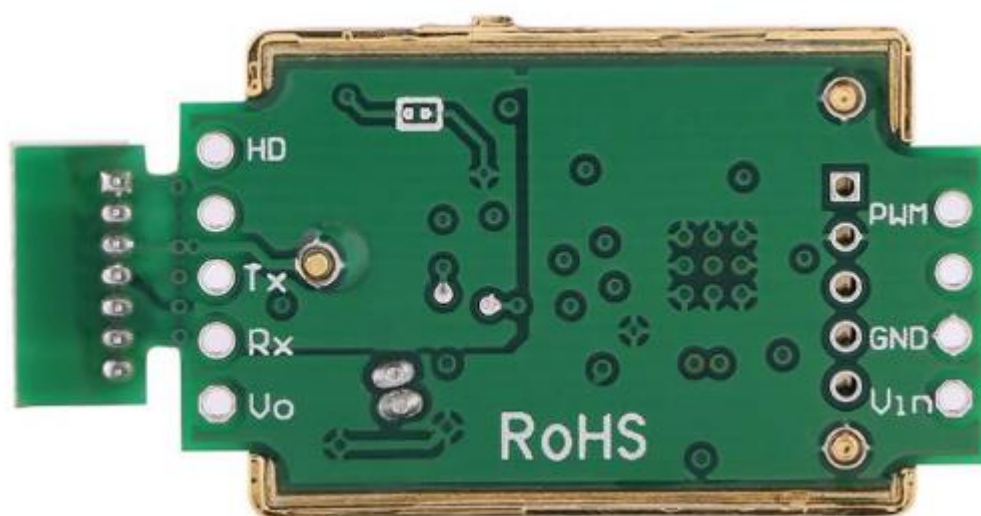


Рисунок 3.2 – Призначення контактів датчика MH-Z19B

Підключивши усі модулі MH-Z19B, можна перейти до під'єднання сервомоторів SG90 [65].

Даний модуль має три контакти, що видно з рисунку 3.3:

- $V_{cc}$  – живлення з напругою 5 V;
- GND – заземлення;



- DC- – контакт заземлення;
- Signal 1 та Signal 2 – контакти отримання сигналу.

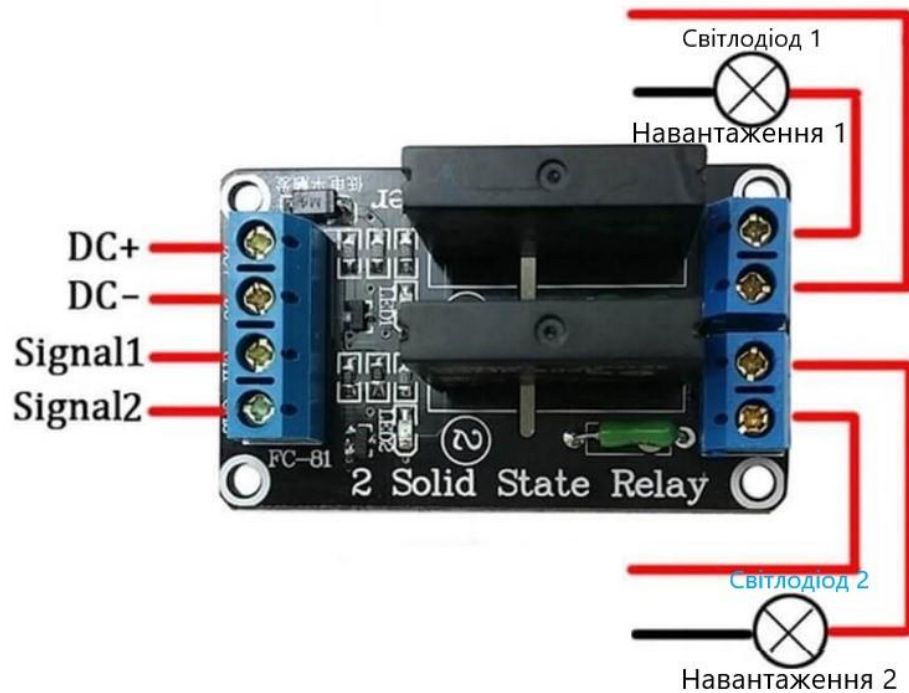


Рисунок 3.5 – Призначення контактів двоканального твердотілого реле Omron G3MB-202P

З іншого боку елемента є дві пари контактів. Кожна пара становить собою роз'єм для підключення змінного струму, а також контакт зі світлодіодом, який приймає і передає далі управляючий сигнал на пристрої, у випадку цього проекту – каналні вентилятори.

Схематичне зображення повної підсистеми з підключеними модулями можливо побачити на рисунку 3.6.

На рисунку 3.7 можливо побачити готову апаратну частину підсистеми керування мікрокліматом житла.

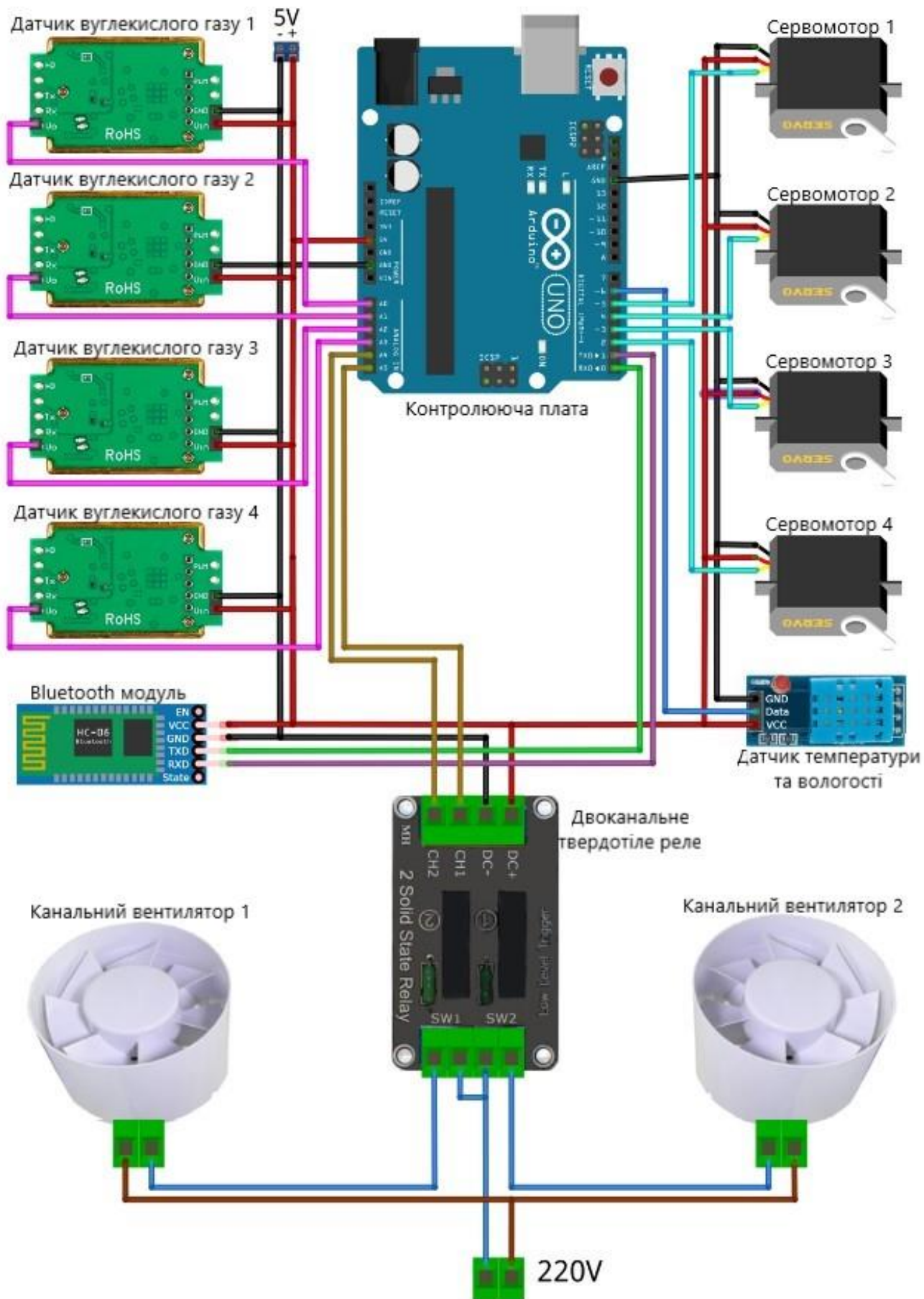


Рисунок 3.6 – Схематичне зображення підсистеми

Зм.	Арк.	№ докум.	Підпис	Дата



Після під'єднання світлодіодний індикатор живлення почне світитись зеленим кольором, а оперативна система Windows автоматично встановить драйвери плати Arduino UNO.

Далі потрібно відкрити Arduino IDE і на верхній панелі інтерфейсу обрати вкладку «Інструменти».

У відкритому меню, при наведенні курсору миші на пункт «Плата», з правого боку з'явиться меню «Менеджер плат», де потрібно обрати плату «Arduino UNO». Також потрібно обрати необхідний порт у меню «Інструменти», натиснувши пункт «Порт».

На рисунку 3.8 відображено виконання даних кроків.

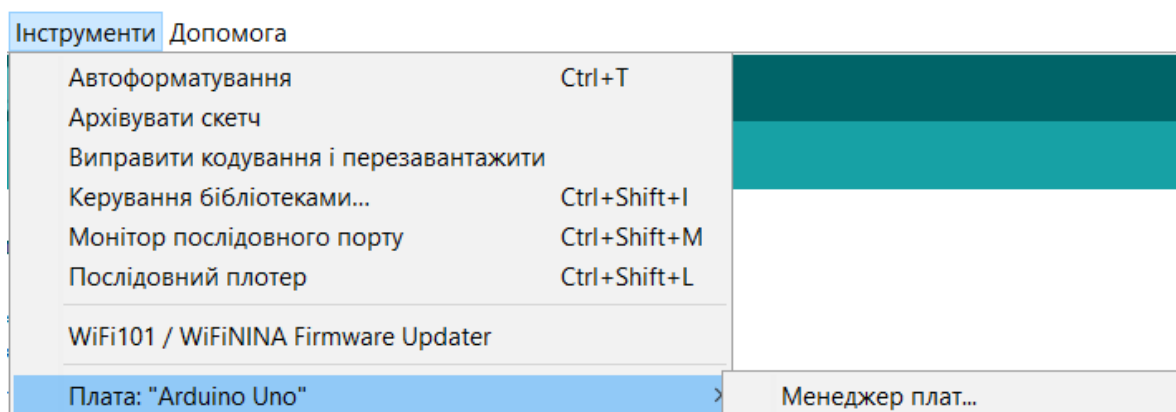


Рисунок 3.8 – Обрання необхідної плати в Arduino IDE

Після виконання описаних кроків можливо переходити до написання програмного коду.

Спочатку файлу потрібно додати необхідні бібліотеки [70] – [72], що представлено на рисунку 3.9:

- “EEPROM.h” – надає можливість роботи з енергонезалежною пам’яттю (EEPROM) плати Arduino UNO, що дозволяє записувати значення у комірки пам’яті за адресами. Це гарантує стабільну роботу додаткових елементів, наприклад, датчиків, якщо відбулась аварійна ситуація або зупинилась подача живлення;

- “Servo.h” – дозволяє керувати сервомоторами;

- “SimpleDHT” – потрібно застосувати, щоб мати змогу використовувати датчики DHT22 та DHT11, і спростити підключення до них.

```
sketch_system

#include "EEPROM.h" //бібліотека енергонезалежної пам'яті
#include "Servo.h"//бібліотека для управління сервомоторами
#include "SimpleDHT.h"//бібліотека для роботи з датчиками DHT11 та DHT22
```

Рисунок 3.9 – Додавання необхідних бібліотек

Бібліотеки “EEPROM.h” та “Servo.h” є стандартними для середовища Arduino IDE, тому не виникає потреби в їх встановленні, на відміну від “SimpleDHT”. Для застосуванні у програмі даної бібліотеки потрібно перейти у вкладку «Інструменти» та у відкритому меню вибрати «Керування бібліотеками» або скористатися поєднанням клавіш «Ctrl+Shift+I».

Ці кроки можливо побачити на рисунку 3.10.

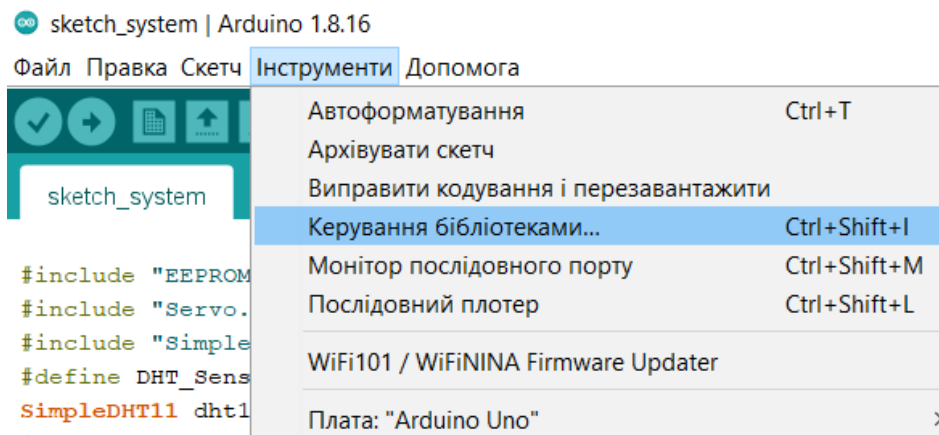


Рисунок 3.10 – Пункт «Керування бібліотеками»

Після натискання на даний пункт, відкриється вікно «Менеджер бібліотек». У вікно можливо сортувати бібліотеки що відображаються за темою і типом, а також ввести назву необхідного файлу в текстове поле пошуку.

Скориставшись пошуком за назвою програма представить збіги, якщо вони є. Також надається можливість обрання необхідної версії для встановлення.

На рисунку 3.11 зображено встановлення бібліотеки “SimpleDHT” для подальшого застосування при написанні програми проекту.

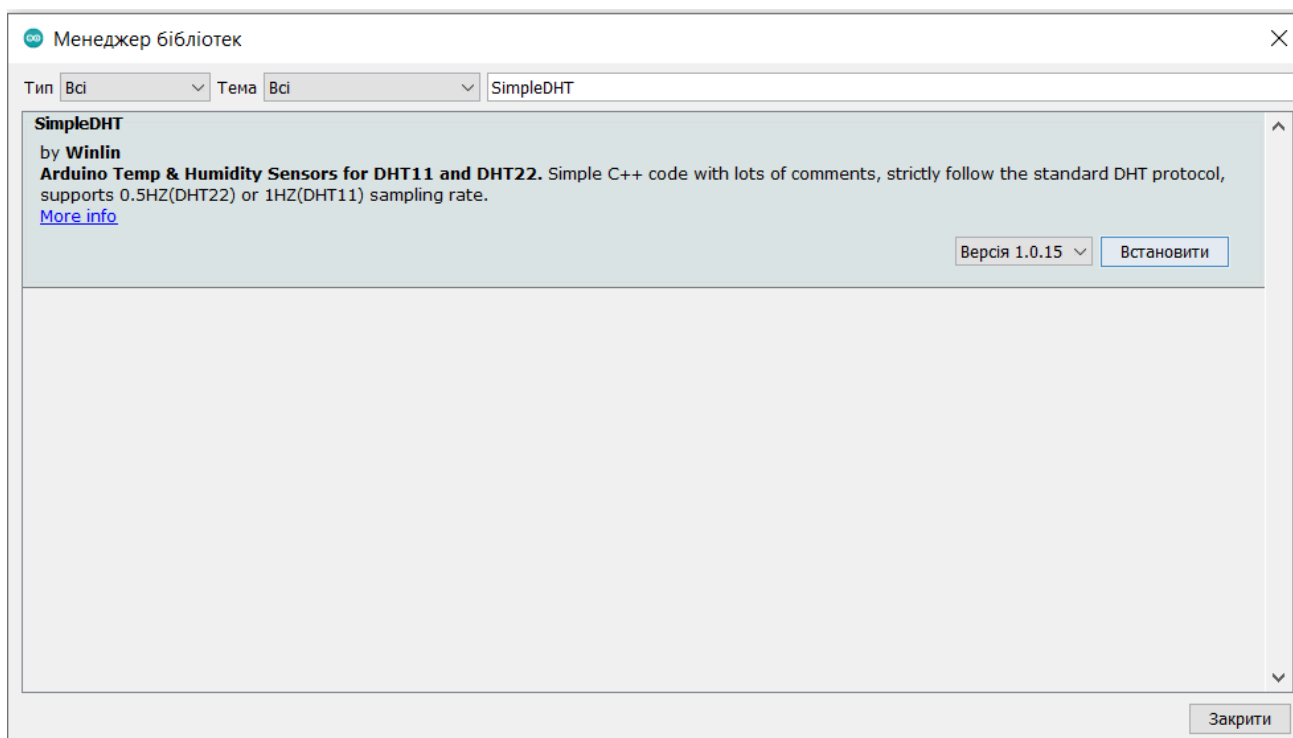


Рисунок 3.11 – Встановлення бібліотеки “SimpleDHT” через «Менеджер бібліотек»

Після цього відбувається робота з датчиком температури та вологості DHT11 [73].

Спочатку потрібно вказати до якого контакту підключається сенсор. Це відбувається за допомоги створення константи «DHT\_Sensor». Конструкція «#define» дозволяє економити на місці в пам’яті програми на чіпу, адже при компіляції будь-яке посилання на константу буде замінюватись її значенням.

Друга стрічка на рисунку 3.12 застосовує бібліотеку “SimpleDHT” та вказує яка модель датчика використовується, а в дужках описується номер контакту підключення даного сенсору.

Наступна стрічка коду встановлює початкові значення температури та вологості, тобто «0».

```
#define DHT_Sensor 6 //контакт для підключення датчика DHT11
SimpleDHT11 dht11(DHT_Sensor);
int temperature_=0, humidity_=0;
```

Рисунок 3.12 – Оголошення контакту підключення, моделі та початкових значень датчика температури та вологості

Далі слід створити об'єкти для управління сервомоторами, а так як у проекті даних модулів – чотири, тому об'єкти типу «Servo» буде відповідна кількість. Це дозволяє зробити бібліотека “Servo.h”.

На рисунку 3.13 продемонстровано даний процес.

```
Servo myservo_1;
Servo myservo_2;
Servo myservo_3;
Servo myservo_4;
```

Рисунок 3.13 – Створення об'єктів для управління сервомоторами

Після створення об'єктів для управління варто підключити сервомотори. Спосіб запису номеру контактів підключення сервомоторів аналогічний до під'єднання датчика температури та вологості..

Також потрібно оголосити змінні, які встановлюють значення кутів повороту вентиляційної решітки валами сервомоторів. Так як кількість кімнат становить відповідність вентиляційним решіткам, то програма буде містити змінні «vgrille\_open\_n» та «vgrille\_close\_n», де n – це номер решітки та кімнати.

У змінних з назвою «vgrille\_open» значення «90» відповідає куту оберту в градусах, при яких вентиляційна решітка відкрита, а в змінних з назвою «vgrille\_close» при значенні «0» - зачинена.

На рисунку 3.14 продемонстровано підключення сервомоторів та встановлення значень для решітки вентиляції.

```

//контакти підключення сервомоторів
#define servo_1 2
#define servo_2 3
#define servo_3 4
#define servo_4 5

//встановлення значень в градусах, коли решітка відкрита, а коли закрита
#define vgrille_open_1 90
#define vgrille_close_1 0

#define vgrille_open_2 90
#define vgrille_close_2 0

#define vgrille_open_3 90
#define vgrille_close_3 0

#define vgrille_open_4 90
#define vgrille_close_4 0

```

Рисунок 3.14 – Підключення сервомоторів та встановлення значень для решітки вентиляції

Далі оголошуються змінні «flag\_n» та «flag\_nforce» булевого типу, тобто вони будуть приймати лише логічний «0» або «1», а «n» означає номер кімнати. Змінні «flag\_nforce» використовуються для позначення того, що варто запустити систему у певній кімнаті, номер якої відповідає числу в назві, незважаючи на показники датчиків командою відправленою через телефон. «flag\_n» застосовується при перевірці умов для ввімкнення системи, але приймаючи до уваги показання сенсорів. Логічні змінні «flag\_1234» та «flag\_1234force» виконують аналогічні функції.

Оголошення цих змінних видно на рисунку 3.15.

```

boolean flag_1=0, flag_2=0, flag_3=0, flag_4=0, flag_1force=0, flag_2force=0, flag_3force=0, flag_4force=0;
boolean flags_1234=0, flags_1234force=0;

#define ventilationunit_general 18
#define ventilationunit_kitchen 19

int time_=5; //інтервал запиту сенсорів
int SensorPins[4][4]={
  {A0,A1,A2,A3}, // контакти підключення датчиків вуглекислого газу
  {0,0,0,0},
  {0,0,0,0},
  {0,0,0,0},
};

```

Рисунок 3.15 – Підключення вентиляторів та перевірка значень датчика

Наступний крок написаного програмного коду – оголошення контактів до яких підключаються каналні вентилятори.

Подальші стрічки коду оголошують цілочисельну змінну «time\_», яка встановлює інтервал запиту сенсорів у секундах і використовується при роботі з датчиками вуглекислого газу. Також оголошується багатомірний масив «SensorPins», де в першій стрічці вказуються аналогові контакти підключення датчиків МН-Z19В, а в наступних рядках – запис та перевірка значень, які передали ці сенсори.

Даний процес відбувається так: датчики виміряли кількість вуглекислого газу і записали у другий рядок, кожен у свій стовпчик. Наступні показання датчиків вказуються у третьому рядку, але запис в даний рядок відбувається лише тоді, коли перші та другі значення не відрізняються більше ніж на «8». Аналогічний процес перевірки відбувається між третім та четвертим рядком масиву. Якщо ж виміри відрізняються більше ніж на «8», процес запису даних та перевірки їх відбувається з початку.

Це допомагає отримувати оптимальні виміри вуглекислого газу, адже користувачу передаються значення записані в останньому рядку масиву.

На рисунку 3.16 можливо побачити, оголошення змінних.

```
String status1, status2, status3, status4, kitchen;
int limit_1, limit_2, limit_3, limit_4, limit_1s = 800, limit_2s = 800, limit_3s = 800, limit_4s = 800;
int address_0 = 23, address_1 = 25, address_2 = 28, address_3 = 30, address_4 = 32, flag_start = 4, flag_st;
int count = 0, scan_count = 1, ppm_BUF = 0, result = 0, while_count = 0;
long ppm, timing1 = 0;
```

Рисунок 3.16 – Оголошення змінних статусу роботи, граничних значень, адрес пам'яті EEPROM та перевірки значень вуглекислого газу

У першому рядку змінні мають тип «string», що означає прийняття стрічок. Вони будуть використовуватись для описання статусу роботи вентиляторів «ON» та «OFF».

					КВРКІ 190138.19.01.22 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

У другій стрічці оголошено значення вуглекислого газу, при яких мають працювати каналні вентилятори.

У третьому рядку оголошено адреси пам'яті EEPROM, а також змінні, які допоможуть в подальшій роботі з нею.

Решта програмного коду – змінні, які будуть використовуватись для роботи з датчиками МН-Z19В.

На рисунку 3.17 видно, що в програмі було використано функцію setup(), яка є обов'язковою для коректного виконання програми.

Дана функція спрацьовує один раз при натисканні кнопки «Reset» або підключення живлення до плати, а в тілі setup() встановлюють режими роботи портів.

```
void setup() {
  Serial.begin(9600);
  //читання з енергонезалежної пам'яті (EEPROM)
  EEPROM.get(address_0, flag_st);

  //одноразовий запис початкових налаштувань при першому ввімкненні після перепрошивання
  if (flag_st != flag_start){
    EEPROM.put(address_0, flag_start);
    EEPROM.put(address_1, limit_1s);
    EEPROM.put(address_2, limit_2s);
    EEPROM.put(address_3, limit_3s);
    EEPROM.put(address_4, limit_4s);
  }
  //читання з енергонезалежної пам'яті (EEPROM)
  EEPROM.get(address_1, limit_1);
  EEPROM.get(address_2, limit_2);
  EEPROM.get(address_3, limit_3);
  EEPROM.get(address_4, limit_4);
}
```

Рисунок 3.17 – Створення функції setup () та робота з EEPROM

Serial.begin() використовується для встановлення швидкості передачі даних Bluetooth модуля у біт/секунду.

Після цього відбувається робота з енергонезалежною пам'яттю застосовуючи команди, які надала бібліотека “EEPROM.h”. Функції get() дозволяє читати дані з пам'яті EEPROM.

Встановивши швидкість передачі даних, проводиться перевірка чи співпадає значення в комірці змінній «flag\_start».

При першому запуску плати після оновлення програм в комірках пам'яті знаходяться значення «0» або «255», тому відбувається запис гранично допустимих показань датчиків функцією put().

В подальшому вираз в умовній конструкції не буде виконуватись, а буде відбуватись просте зчитування гранично допустимих значень.

З використанням функції attach(), що належить до бібліотеки “Servo.h”, відбувається підключення сервомоторів до контактів. Друга функція цієї ж бібліотеки – write(), встановлює вали сервомоторів у закриті положення, тобто на кут нуль градусів.

pinMode() призначає контактам для підключення двоканального твердотілого реле тип «Вихідний», а за допомогою digitalWrite() цифровим роз'ємам встановлюється високий рівень напруги, так як на модуль реле включається низьким рівнем, що представлено на рисунку 3.18.

```
//підключення сервомоторів
myservo_1.attach(servo_1);
myservo_2.attach(servo_2);
myservo_3.attach(servo_3);
myservo_4.attach(servo_4);

//встановлення валів сервомоторів в стан закриття вентиляційних решіток
myservo_1.write(vgrille_close_1);
myservo_2.write(vgrille_close_2);
myservo_3.write(vgrille_close_3);
myservo_4.write(vgrille_close_4);

//встановлення вентиляторів у стан виключення
pinMode(ventilationunit_general, OUTPUT); digitalWrite(ventilationunit_general, HIGH);
pinMode(ventilationunit_kitchen, OUTPUT); digitalWrite(ventilationunit_kitchen, HIGH);
}
```

Рисунок 3.18 – Встановлення типу та напруги для модуля реле

Після завершення функції setup(), потрібно створити нову – loop(), яка на відміну від першої може виконуватись безліч кількості разів, тому в ній прописується логіка роботи датчиків та інших додаткових елементів.

З рисунку 3.19 видно, що на початку loop() описується перевірка концентрації вуглекислого газу, умовою, яка залежить від часу роботи сенсора, а також містить в собі логіку перевірки записів даних у масив «SensorPins», який зображений на рисунку 3.15.

```
void loop() {
  //----- Перевірка концентрації вуглекислого газу по 4 датчикам MH-Z19B -----

  if (millis() - timing1 > time_ * 1000){
    timing1 = millis();

    while(while_count < 20){
      int Val = analogRead(SensorPins[0][count]);
      ppm = map(Val, 18, 650, 650, 2000);

      if(scan_count > 1 && (ppm > ppm_BUF - 8 && ppm < ppm_BUF + 8)){
        ppm_BUF = ppm; SensorPins[scan_count][count] = ppm;
        scan_count++;
        if(scan_count > 3){

          scan_count = 1; count++; if(count > 3){ count = 0;}
        }
      }else{ scan_count = 1;}

      if(scan_count == 1){ SensorPins[scan_count][count] = ppm; ppm_BUF = ppm;
        scan_count++;
      }
      while_count++;
    }
    while_count = 0;
  }
}
```

Рисунок 3.19 – Перевірка концентрації вуглекислого газу

Також функція loop() містить програмну логіку роботи з сенсором температури та вологості DHT11.

Робота з датчиком вологості та температури представлена на рисунку 3.20.

```
//----- Перевірка температури та вологості по датчику DHT11 -----
byte temperature = 0;
byte humidity = 0;
dht11.read(&temperature, &humidity, NULL);
temperature_ = (int)temperature; humidity_ = (int)humidity;
if(humidity_ > 70){digitalWrite(ventilationunit_kitchen, LOW); kitchen = " ON";
}else{digitalWrite(ventilationunit_kitchen, HIGH); kitchen = " OFF";}
if(temperature_ > 23){digitalWrite(ventilationunit_kitchen, LOW); kitchen = " ON";
}else{digitalWrite(ventilationunit_kitchen, HIGH); kitchen = " OFF";}
```

Рисунок 3.20 – Перевірка температури та вологості

					КВРКІ 190138.19.01.22 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

З рисунку 3.20 видно, що після вимірів показників, відбувається їх перевірка на відповідність граничного значення, тому якщо значення більше умови – ввімкнеться каналний вентилятор, якщо ж ні – буде надалі перебувати у стані спокою.

Наступний крок – створення можливості встановлення нових гранично допустимих значень вуглекислого газу через телефон користувача, що видно на рисунку 3.21.

```
//--| Налаштування гранично допустимих значень для датчика MH-Z19B через телефон

if (Serial.available()) {
  char c = Serial.read();
  if (c == 'A') {
    limit_1 += 5; EEPROM.put(address_1, limit_1);
    Serial.print(F("GuestRoom +5    LIMIT = ")); Serial.println(limit_1);
  }
  if (c == 'a') {
    limit_1 -= 5; EEPROM.put(address_1, limit_1);
    Serial.print(F("GuestRoom -5    LIMIT = ")); Serial.println(limit_1);
  }
  if (c == 'B') {
    limit_2 += 5; EEPROM.put(address_2, limit_2);
    Serial.print(F("Someroom +5    LIMIT = ")); Serial.println(limit_2);
  }
  if (c == 'b') {
    limit_2 -= 5; EEPROM.put(address_2, limit_2);
    Serial.print(F("Someroom -5    LIMIT = ")); Serial.println(limit_2);
  }
  if (c == 'C') {
    limit_3 += 5; EEPROM.put(address_3, limit_3);
    Serial.print(F("BedRoom +5    LIMIT = ")); Serial.println(limit_3);
  }
  if (c == 'c') {
    limit_3 -= 5; EEPROM.put(address_3, limit_3);
    Serial.print(F("BedRoom -5    LIMIT = ")); Serial.println(limit_3);
  }
  if (c == 'D') {
    limit_4 += 5; EEPROM.put(address_4, limit_4);
    Serial.print(F("Cabinet +5    LIMIT = ")); Serial.println(limit_4);
  }
  if (c == 'd') {
    limit_4 -= 5; EEPROM.put(address_4, limit_4);
    Serial.print(F("Cabinet -5    LIMIT = ")); Serial.println(limit_4);
  }
}
```

Рисунок 3.21 – Налаштування користувачем нових гранично допустимих значень вуглекислого газу через телефон

Для цього відбувається умова перевірки наявності нових даних, які надійшли на послідовний порт, а потім зчитування символів, які означають команди.

В умовних конструкціях перевіряють чи отриманий символ відповідає умові, а потім виконують наступні дії, які описані в даній конструкції, або перевірка проводиться далі.

Дані символи отримуються від взаємодії користувача з елементами мобільного додатку, який відсилає команди через Bluetooth модуль.

При отриманні великого символу граничне значення зменшується на «5», а при символі малого регістра – збільшення на аналогічне значення.

Нове гранично допустиме значення записується у комірку енергонезалежної пам'яті, а інтерфейс мобільного додатку проінформує користувача про збільшення/зменшення значення у відповідній кімнаті та новий ліміт вуглекислого газу після обчислення.

На рисунку 3.22 описано логіку отримання інформації про підсистему.

```
//----- Запит щодо встановлених гранично допустимих значень та станів роботи сервомоторів
//та вентиляторів, а також показ значень датчиків MH-Z19B, DHT11 -----

if (c == 'E') {
  Serial.print(F("GuestRoom = ")); Serial.print(SensorPins[3][0]);
  Serial.print(F("  LIMIT = "));   Serial.print(limit_1);
  Serial.print(F("    "));         Serial.println(status1);

  Serial.print(F("Someroom = ")); Serial.print(SensorPins[3][1]);
  Serial.print(F("  LIMIT = "));   Serial.print(limit_2);
  Serial.print(F("    "));         Serial.println(status2);

  Serial.print(F("BedRoom = "));  Serial.print(SensorPins[3][2]);
  Serial.print(F("  LIMIT = "));   Serial.print(limit_3);
  Serial.print(F("    "));         Serial.println(status3);

  Serial.print(F("Cabinet = "));  Serial.print(SensorPins[3][3]);
  Serial.print(F("  LIMIT = "));   Serial.print(limit_4);
  Serial.print(F("    "));         Serial.println(status4);

  Serial.print(F("Temperature= ")); Serial.print(temperature_); Serial.println(F("*C"));
  Serial.print(F("Humidity = "));  Serial.print(humidity_);   Serial.print(F("%"));
  Serial.println(kitchen);
}
```

Рисунок 3.22 – Отримання користувачем інформації про стан всієї підсистеми

При прийнятті символу «Е» користувач отримує виміри зі всіх датчиків по всьому будинку, а також встановлені граничні значення та статус роботи сервомоторів та вентиляторів, що видно на рисунку 3.22.

Якщо користувач бажає ввімкнути вентилятори у певній кімнаті, незалежно від показників сенсорів, він може передати команду для примусової роботи, рисунок 3.23.

```
//----- Примусове ввімкнення вентилятора та сервомотора через команду -----  
  
if (c == 'F') {Serial.println(F("GuestRoom ON")); flag_1force = 1;}  
if (c == 'f') {Serial.println(F("GuestRoom OFF")); flag_1force = 0;}  
if (c == 'G') {Serial.println(F("Someroom ON")); flag_2force = 1;}  
if (c == 'g') {Serial.println(F("Someroom OFF")); flag_2force = 0;}  
if (c == 'H') {Serial.println(F("BedRoom ON")); flag_3force = 1;}  
if (c == 'h') {Serial.println(F("BedRoom OFF")); flag_3force = 0;}  
if (c == 'I') {Serial.println(F("Cabinet ON")); flag_4force = 1;}  
if (c == 'i') {Serial.println(F("Cabinet OFF")); flag_4force = 0;}  
  
}
```

Рисунок 3.23 – Отримання системою команди на примусове ввімкнення

Подальший код програми перевіряє потребу в роботі сервомоторів, а також у ввімкненні каналних вентиляторів через умовні конструкції, що зображено на рисунку 3.24.

```
//----- ввімкнення сервомоторів -----  
  
if(flag_1 == 1 || flag_1force == 1){myservo_1.write(vgrille_open_1); status1 = "ON";}  
else{myservo_1.write(vgrille_close_1); status1 = "OFF";}  
  
if(flag_2 == 1 || flag_2force == 1){myservo_2.write(vgrille_open_2); status2 = "ON";}  
else{myservo_2.write(vgrille_close_2); status2 = "OFF";}  
  
if(flag_3 == 1 || flag_3force == 1){myservo_3.write(vgrille_open_3); status3 = "ON";}  
else{myservo_3.write(vgrille_close_3); status3 = "OFF";}  
  
if(flag_4 == 1 || flag_4force == 1){myservo_4.write(vgrille_open_4); status4 = "ON";}  
else{myservo_4.write(vgrille_close_4); status4 = "OFF";}  
  
//----- ввімкнення головного вентилятора -----  
  
if (flag_1 == 1 || flag_2 == 1 || flag_3 == 1 || flag_4 == 1){  
flags_1234 = 1;}else{flags_1234 = 0;}  
  
if (flag_1force == 1 || flag_2force == 1 || flag_3force == 1 || flag_4force == 1){  
flags_1234force = 1;}else{flags_1234force = 0;}  
  
if (flags_1234 == 1 || flags_1234force == 1){  
digitalWrite(ventilationunit_general, LOW);  
}else{ digitalWrite(ventilationunit_general, HIGH);}  
  
}
```

Рисунок 3.24 – Умови для ввімкнення сервомоторів та вентиляторів

Після написання всієї програми варто перевірити її на помилки через відповідну кнопку інтерфейсу програми Arduino IDE.

На рисунку 3.25 представлено результати компілювання програми.

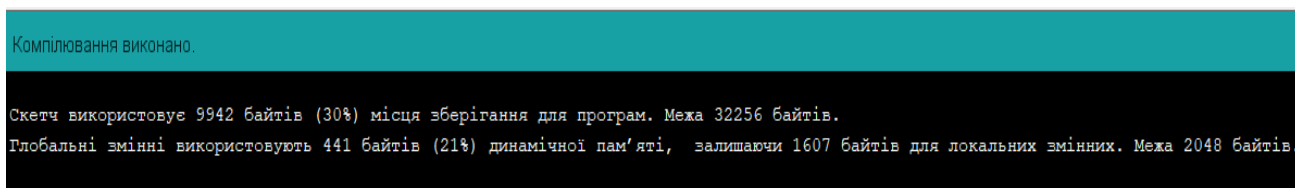


Рисунок 3.25 – Результати успішного компілювання програми

Отримавши задовільні результати, можна вивантажувати створений скетч на плату Arduino UNO.

### 3.3 Створення проекту мобільного додатку та тестування підсистеми

Перед початком створення мобільного додатку, варто підключити телефон користувача та модуль HC-06.

На телефоні, Bluetooth виявить доступний пристрій для під'єднання з назвою «hc-06», а пароль для підключення зазвичай становить комбінацію чисел «0000» або «1234».

Після цього потрібно через Arduino IDE відправити AT-команду на Bluetooth модуль, де у відповідь має бути прислано «ОК». Зазвичай, даними командами можливо встановити швидкість передачі даних, ім'я пристрою та пароль для підключення.

Зв'язавши телефон та додатковий елемент за допомоги Bluetooth, світлодіод на модулі HC-06 перестане миготіти, а буде стабільно випромінювати світло.

На рисунку 3.26 зображено реакція Bluetooth модуля на підключення іншого пристрою користувачем.



Рисунок 3.26 – Реакція Bluetooth модуля на підключення іншого пристрою користувачем

Для налаштування мобільного додатку потрібно відкрити програму «Bluetooth Electronics» та вибрати пустий проект для редагування.

На рисунку 3.27 можна побачити інтерфейс пустого проекту.

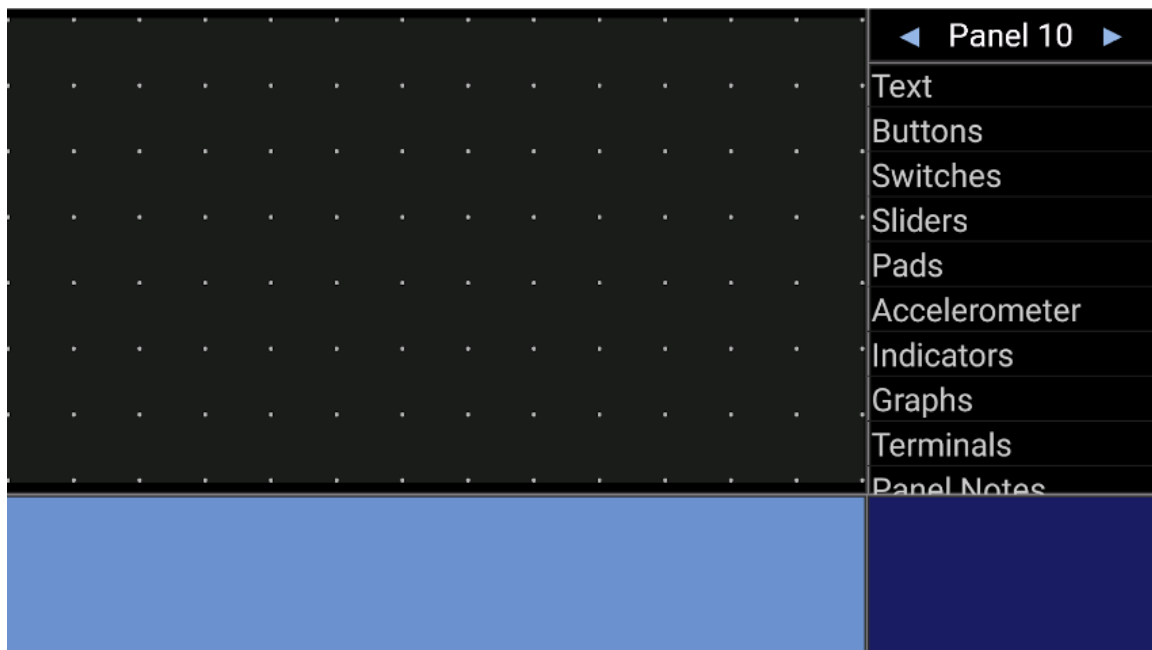


Рисунок 3.27 – Початок створення власного проекту в мобільному додатку «Bluetooth Electronics»

В проєкті інтерфейс поділений на декілька частин:

- Місце для розміщення елементів, яке містить сітку для полегшення розташування елементів;
- В правій частині розташовано меню для вибору елементів для переміщення на сітку та інформація про номер проєкту з можливістю переходу до інших;
- Блакитним кольором виділена частина, де надається можливість вибору вигляду елементів тип яких був обраний у меню правої частини інтерфейсу;
- Темно-синя частина інтерфейсу призначена для виводу та налаштування інформації про переміщений елемент на сітку, тобто його назву, та команди, які він відсилає при роботі.

На початку, варто встановити задовільні розміри сітки для розташування елементів.

Для цього потрібно вибрати у меню компонентів пункт «Grid Size» та налаштувати її розмір на шкалі, що з'явилася.

На рисунку 3.28 видно налаштування розмірів сітки.

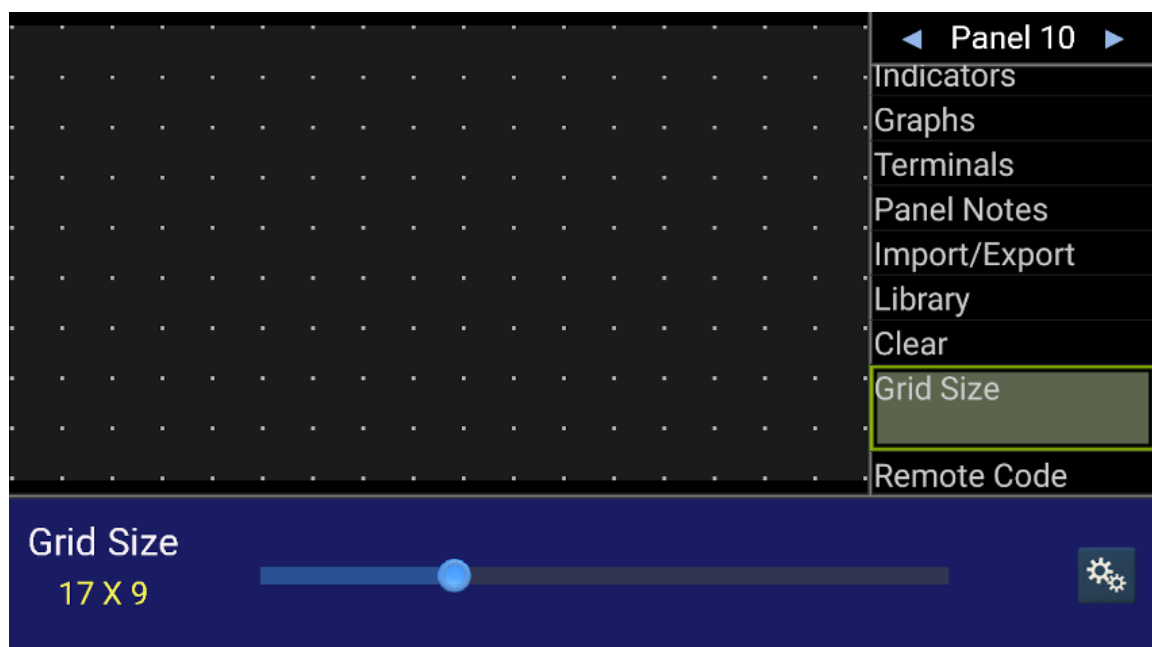


Рисунок 3.28 – Налаштування розмірів сітки проєкту

Виконавши цю дію, було обрано пункт «Terminals» та перенесення монітору розмірами «14×48» на шаблон.

Даний елемент буде використовуватись для відображення всієї інформації про підсистему.

Натиснувши пункт «Switches», який надав доступ застосування та переміщення на шаблон перемикачів.

При створенні проекту було застосовано чотири перемикачі, дев'ять кнопок, чотири текстових поля та монітор. Всі елементи потрібно обирати у відповідних пунктах меню компонентів.

На рисунку 3.29 видно усі переміщені елементи та команди, які вони відсилають.

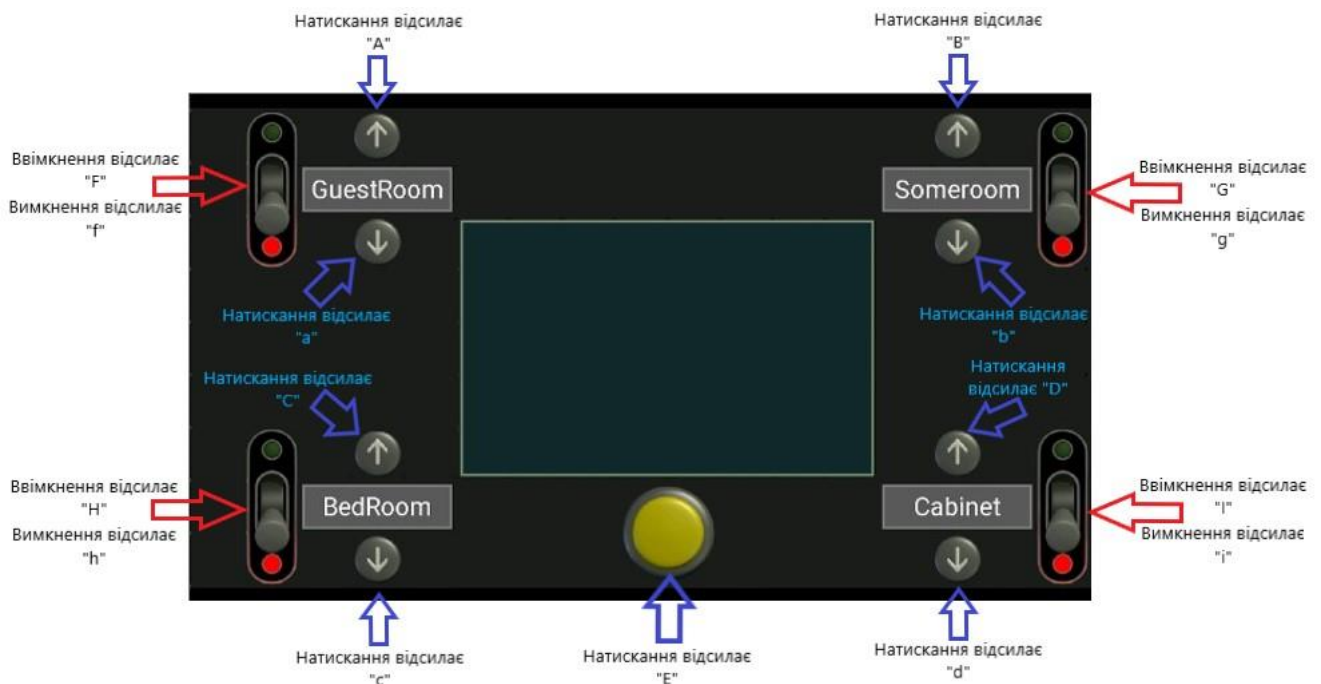


Рисунок 3.29 – Вигляд проекту після розміщення всіх компонентів та пояснення команд, які надсилаються

Команди, які надсилають компоненти описані в програмному кодї.

Тепер потрібно перевірити, як працює система та чи існує реакція на запити користувача.

Спочатку треба натиснути велику жовту кнопку для отримання інформації про підсистему, результати дії зображено на рисунку 3.30.

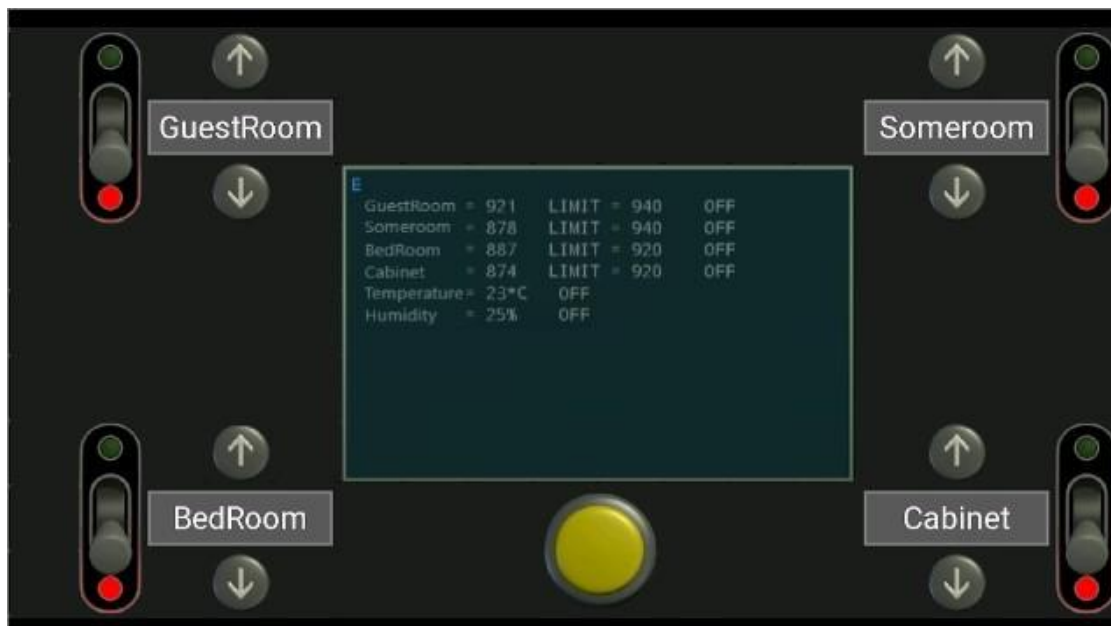


Рисунок 3.30 – Отримання інформації після натиснення жовтої кнопки

Тепер можливо спробувати збільшити та зменшити гранично допустимі значення. Для збільшення потрібно натиснути кнопку зі стрілкою вгору, а щоб зменшити – кнопку зі стрілкою вниз.

На рисунку 3.31 видно збільшення значень.



Рисунок 3.31 – Збільшення та зменшення граничних значень

Потрібно перевірити чи збереглись зміни. Для цього слід натиснути жовту кнопку, що видно на рисунку 3.32.

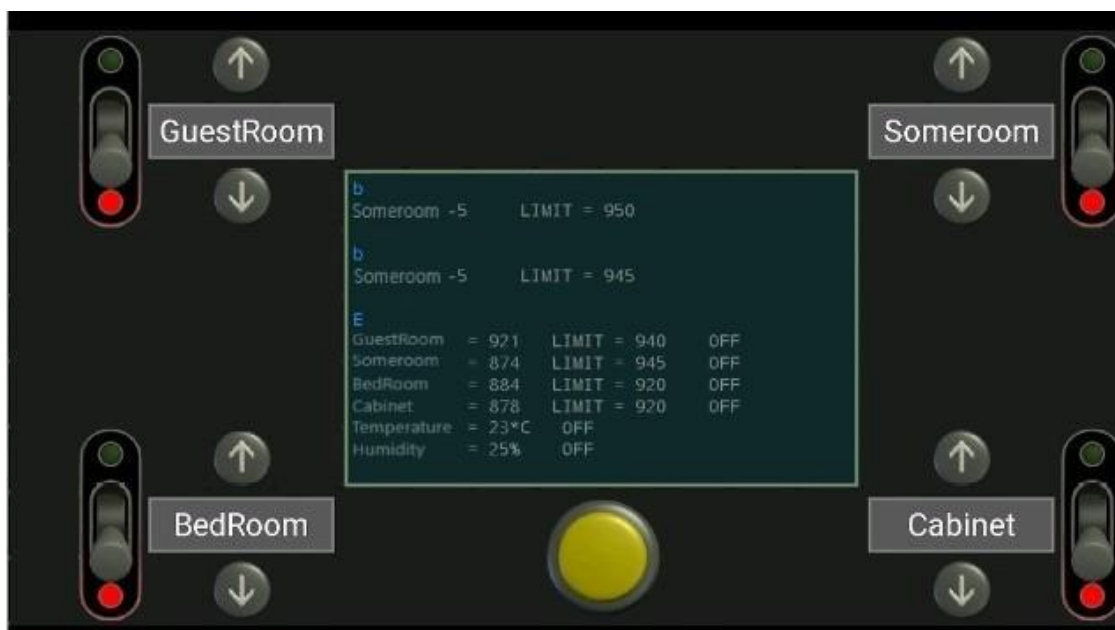


Рисунок 3.32 – Відображення змінених гранично допустимих значень

Задля перевірки примусового ввімкнення вентиляторів та сервомоторів, потрібно перемістити важіль перемикача.

Робота проекту після переміщення важелю представлена на рисунку 3.33.

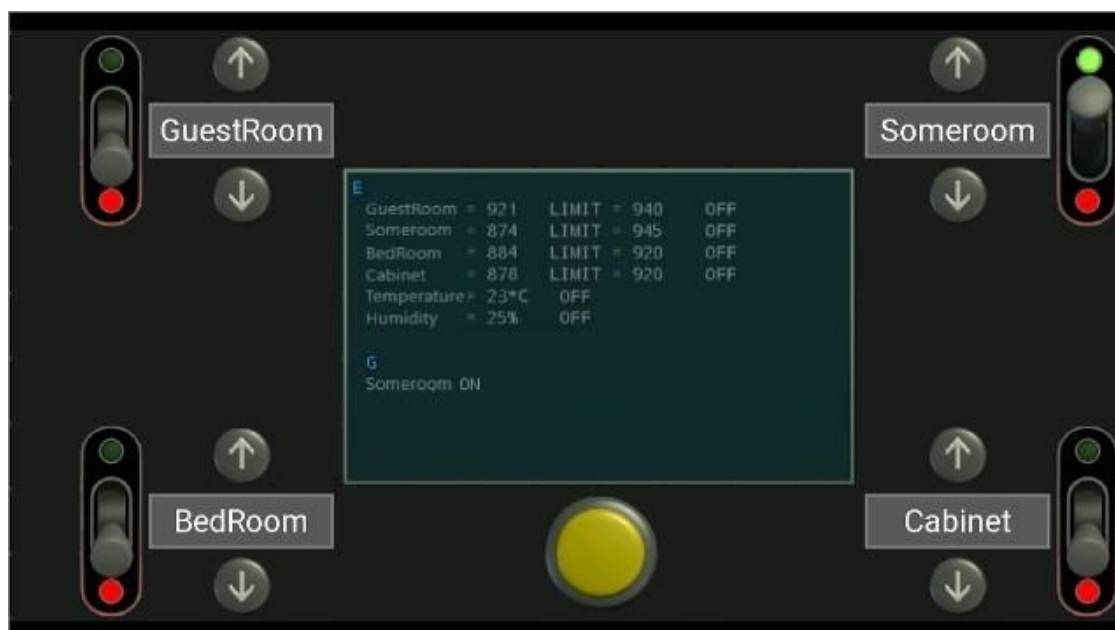


Рисунок 3.33 – Примусове ввімкнення вентиляторів та сервомоторів

Після цього починає працювати каналний вентилятор, повертається вал сервомотору та запалюється світлодіод реле, що видно на рисунку 3.34.

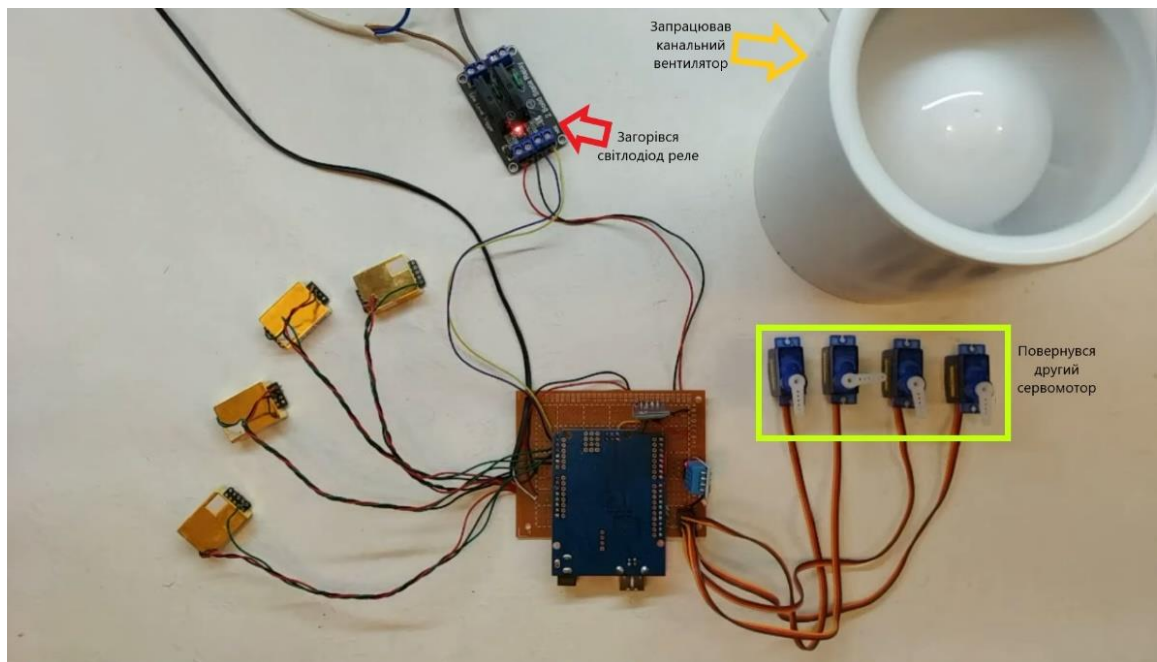


Рисунок 3.34 – Реакція апаратної частини на дії

Для перевірки датчиків можна подихати на них. МН-Z19В отримує необхідний вуглекислий газ, а DHT11 – вологість. Система також буде запускати вентилятори та сервомотори, якщо буде перевищено граничне значення, що представлено на рисунку 3.35.



Рисунок 3.35 – Перевірка роботи датчиків

## ВИСНОВОК

Під час виконання кваліфікаційної роботи було проведено аналіз предметної області та існуючих рішень, визначено постановлені задачі.

В результаті проектування підсистеми керуванням мікрокліматом житла у кіберфізичній системі «Розумний будинок» було обґрунтовано вибір мови програмування та середовища для створення програми, що дало можливість обрати плату Arduino UNO та додаткові модулі для функціонування підсистеми.

Під час програмно-апаратної реалізації, були підключені елементи та розроблена програма. Окрім цього, був створений проект в мобільному додатку, який застосовується для передачі та виведення інформації.

Цілі кваліфікаційної роботи досягнуті.

					КВРКІ 190138.19.01.22 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

## СПИСОК ПОСИЛАНЬ

1. The history of the Smart Home. URL: <https://smarthomeenergy.co.uk/the-history-of-the-smart-home/> (дата звернення 27.02.2023).
2. Історія створення «SMART HOUSE». URL: <https://dss-bi.com.ua/sitelab1/%D1%96%D1%81%D1%82%D0%BE%D1%80%D1%96%D1%8F%D1%81%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F-smart-house/> (дата звернення 01.03.2023).
3. Історія розумного будинку. URL: <https://www.smarthouse.ua/ua/istoriya-umnogo-doma.html> (дата звернення 26.02.2023).
4. The Evolution of the Smart Home: How it Started [Part 1]. URL: <https://ubuntu.com/blog/the-evolution-of-the-smart-home-how-it-started-part-1> (дата звернення 26.02.2023).
5. ІСТОРІЯ ВИНИКНЕННЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК». URL: <https://ionosfera.org/uk/istoriya-vozniknoveniya-sistemyi-umnyij-dom.html> (дата звернення 26.02.2023).
6. The history of the Smart Home technology. URL: <https://www.familyhandyman.com/article/the-history-of-smart-home-technology/> (дата звернення 27.02.2023).
7. History of the Vacuum Cleaner. URL: [https://www.morclean.co.uk/index.php?\\_route\\_=History%20of%20the%20Vacuum%20Cleaner](https://www.morclean.co.uk/index.php?_route_=History%20of%20the%20Vacuum%20Cleaner) (дата звернення 27.02.2023).
8. History of the Vacuum. URL: <https://whatvacuum.com/history-of-the-vacuum/> (дата звернення 27.02.2023).
9. Borates in Detergents: The Next Step in Laundry Technology. URL: <https://www.borax.com/news-events/august-2018/borates-in-detergents-the-next-step-in-laundry-tec> (дата звернення 27.02.2023).
10. Miele dishwashers: The milestones. URL: <https://www.miele.de/en/m/miele-dishwashers-the-milestones-4967.htm> (дата звернення 27.02.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

11. Push-Button Manor: The Original Smart Home. URL: <https://www.hotfootdesign.co.uk/white-space/push-button-manor-original-smart-home/> (дата звернення 28.02.2023).

12. Перший розумний будинок. URL: <http://d8.com.ua/first-smart-house/> (дата звернення 28.02.2023).

13. The History of Smart Homes. URL: <https://www.afcdud.com/fr/smart-city/422-how-the-history-of-smart-homes.html> (дата звернення 28.02.2023).

14. Що таке «розумний будинок» і навіщо він потрібен? URL: <https://stylus.ua/uk/articles/528.html> (дата звернення 28.02.2023).

15. Що таке Розумний Будинок. URL: [https://www.smarthouse.ua/ua/umnyj\\_dom.html](https://www.smarthouse.ua/ua/umnyj_dom.html) (дата звернення 28.02.2023).

16. Розумний Будинок. URL: <https://researchclub.com.ua/technology/rozumnyj-budynok/> (дата звернення 28.02.2023).

17. Smart Home: Definition, How They Work, Pros and Cons. URL: <https://www.investopedia.com/terms/s/smart-home.asp> (дата звернення 01.03.2023).

18. Що таке розумний будинок: функції, види, складові та екосистеми. URL: <https://ek.ua/ua/post/1990/618-cho-takoe-umnyy-dom-funkcii-vidy-sostavlyayuschie-i-ekosistemy/> (дата звернення 01.03.2023).

19. Центри управління розумним будинком. URL: <https://z-wave.com.ua/ua/g857098-tsentry-upravleniya-umnym> (дата звернення 01.03.2023).

20. Життя у три кліки: що таке розумний будинок. URL: [https://zaxid.net/zhittya\\_u\\_tri\\_kliki\\_shho\\_take\\_rozumnyy\\_budinok\\_n1511276](https://zaxid.net/zhittya_u_tri_kliki_shho_take_rozumnyy_budinok_n1511276) (дата звернення 01.03.2023).

21. Клімат. URL: [https://www.smarthouse.ua/ua/upravlenie\\_klimatom.html](https://www.smarthouse.ua/ua/upravlenie_klimatom.html) (дата звернення 01.03.2023).

22. Нова концепція житла - система «розумний дім». URL: <https://dominant-wood.com.ua/ua/news/217-nova-kontseptsiya-zhitla-sistema-rozumnyy-dim> (дата звернення 01.03.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

23. Home app. URL: <https://www.apple.com/ua/home-app/> (дата звернення 01.03.2023).
24. Resources for Prototyping: The Basics of New Languages for Programming Microcontrollers. URL: <https://www.allaboutcircuits.com/news/resources-for-prototyping-basics-new-languages-programming-microcontrollers/> (дата звернення 25.04.2023).
25. Top 10 Best Embedded Programming Languages to Learn in 2023. URL: <https://www.analyticsinsight.net/top-10-programming-languages-to-develop-embedded-systems-in-2023/> (дата звернення 25.04.2023).
26. Assembly - Introduction. URL: [https://www.tutorialspoint.com/assembly\\_programming/assembly\\_introduction.htm](https://www.tutorialspoint.com/assembly_programming/assembly_introduction.htm) (дата звернення 25.04.2023).
27. Assembly Language. URL: <https://www.investopedia.com/terms/a/assembly-language.asp> (дата звернення 25.04.2023).
28. Interacting With Python. URL: <https://realpython.com/interacting-with-python/> (дата звернення 25.04.2023).
29. What is the difference between Python and MicroPython? URL: <https://www.educative.io/answers/what-is-the-difference-between-python-and-micropython> (дата звернення 25.04.2023).
30. What are the Key Pros and Cons of the Arduino Programming Language? URL: <https://emeritus.org/blog/coding-arduino-programming-language/> (дата звернення 25.04.2023).
31. Blockly. URL: <https://developers.google.com/blockly> (дата звернення 29.04.2023).
32. Arduino - Hello World. URL: <https://arduinogetstarted.com/tutorials/arduino-hello-world> (дата звернення 29.04.2023).
33. Arduino Integrated Development Environment (IDE) v1. URL: <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics> (дата звернення 29.04.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

34. What is arduino software (IDE), and how use it ? URL: <https://andprof.com/tools/what-is-arduino-software-ide-and-how-use-it/> (дата звернення 29.04.2023).

35. Introduction to Arduino IDE. URL: <https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html> (дата звернення 29.04.2023).

36. Push Button Switch. URL: <https://electrosome.com/?s=Push+Button+Switch> (дата звернення 30.04.2023).

37. Модуль світлорезистор фоторезистор Arduino. URL: <https://bigl.ua/p398923589-modul-svetorezistor-fotorezistor> (дата звернення 30.04.2023).

38. Electrical resistor isolated on white background. URL: <https://www.shutterstock.com/image-photo/electrical-resistor-isolated-on-white-background-1114667282> (дата звернення 30.04.2023).

39. Led 5mm Verde. URL: <https://megatronica.cc/producto/led-5mm-verde/> (дата звернення 30.04.2023).

40. Control a Servo Motor with Arduino. URL: <https://gpiocc.github.io/learn/arduino/2020/04/10/martin-ku-control-a-servo-motor-with-arduino.html> (дата звернення 30.04.2023).

41. Arduino UNO. URL: <https://www.javatpoint.com/arduino-uno> (дата звернення 30.04.2023).

42. Arduino - Board Description. URL: [https://www.tutorialspoint.com/arduino/arduino\\_board\\_description.htm](https://www.tutorialspoint.com/arduino/arduino_board_description.htm) (дата звернення 30.04.2023).

43. What is Arduino Uno? URL: <https://www.flyrobo.in/blog/what-is-arduino-uno> (дата звернення 30.04.2023).

44. Introduction to ATmega328. URL: <https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html> (дата звернення 30.04.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

45. SENSOR DE TEMPERATURA Y HUMEDAD DHT11.  
[URL:https://www.electroya.com/producto/sensor-de-temperatura-y-humedad-dht11/](https://www.electroya.com/producto/sensor-de-temperatura-y-humedad-dht11/)  
(дата звернення 31.04.2023).

46. DHT11, DHT22 and AM2302 Sensors. URL:  
<https://learn.adafruit.com/dht/overview> (дата звернення 31.04.2023).

47. Датчик углекислого газа CO2 MH-Z19B модуль. URL: <https://www.mini-tech.com.ua/datchik-co2> (дата звернення 31.04.2023).

48. MH-Z19B ndir CO2 sensor for indoor air quality monitoring. URL:  
<https://www.winsen-sensor.com/sensors/co2-sensor/mh-z19b.html> (дата звернення 31.04.2023).

49. The MH-Z19B CO2 sensor bible. URL: <https://emariete.com/en/sensor-co2-mh-z19b/> (дата звернення 31.04.2023).

50. MH-Z19B. URL: <https://revspace.nl/MH-Z19B> (дата звернення 31.04.2023).

51. Серводвигун SG90 2кг. URL: <https://arduino.ua/prod416-servoprivod-sg90-2kg> (дата звернення 31.04.2023).

52. All about Servo Motor SG90. URL: <https://robocraze.com/blogs/post/all-about-servo-motor-sg90> (дата звернення 31.04.2023).

53. Канальный вентилятор VENTS 100 ВКО1 12. URL:  
<https://www.santechshara.ua/ua/kanal-nij-ventilyator-vents-100-vko1-12/> (дата звернення 31.05.2023).

54. Витяжной канальный вентилятор AirRoxy aRc 15W d100 S (Білий).  
URL:[https://elektrokonyk.com.ua/ventylator-arc-100-s/?gclid=CjwKCAjw04yjBhApEiwAJcvNoTPKPIR4wzxAZP3NBVrlfZwTkZXPT071gAa-OcTZbQhUhyP6waDbrxoCz1QQA\\_vD\\_BwE](https://elektrokonyk.com.ua/ventylator-arc-100-s/?gclid=CjwKCAjw04yjBhApEiwAJcvNoTPKPIR4wzxAZP3NBVrlfZwTkZXPT071gAa-OcTZbQhUhyP6waDbrxoCz1QQA_vD_BwE) (дата звернення 31.05.2023).

55. Канальный вентилятор AirRoxy aRc 100 S (01-049). URL:  
<https://kipr.kiev.ua/ru/kanalny-ventyliator-airroxy-arc-100-s-01-049/> (дата звернення 31.05.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 73
Зм.	Арк.	№ докум.	Підпис	Дата		

56. Grove - 2-Channel Solid State Relay. URL: [https://wiki.seeedstudio.com/Grove-2-Channel\\_Solid\\_State\\_Relay/](https://wiki.seeedstudio.com/Grove-2-Channel_Solid_State_Relay/) (дата звернення 31.05.2023).

57. Модуль твердотільного реле на 2 каналу OMRON G3MB-202P, 5V для Arduino [#D-5]. URL: <https://ardu.prom.ua/ua/p1008612669-modul-tverdotelnogo-rele.html> (дата звернення 31.05.2023).

58. 2-канальний реле модуль на твердотілих реле OMRON. URL: <https://www.mini-tech.com.ua/2-kanalniy-rele-modul-tverdotelniyij> (дата звернення 31.05.2023).

59. Arduino and Bluetooth module HC-06. URL: <https://www.aranacorp.com/en/arduino-and-bluetooth-module-hc-06/> (дата звернення 31.05.2023).

60. Módulo Bluetooth HC-06. URL: <https://tienda.bricogeek.com/modulos-bluetooth/1351-modulo-bluetooth-hc-06.html> (дата звернення 31.05.2023).

61. Bluetooth модуль HC-06. URL: <https://arduino.ua/prod241-bluetooth-modul-hc-06> (дата звернення 31.05.2023).

62. Connecting to a DHTxx Sensor. URL: <https://learn.adafruit.com/dht/connecting-to-a-dhtxx-sensor> (дата звернення 15.05.2023).

63. Arduino - Temperature Humidity Sensor. URL: <https://arduinogetstarted.com/tutorials/arduino-temperature-humidity-sensor> (дата звернення 15.05.2023).

64. Intelligent Infrared CO2 Module. – С. 5. – URL: [https://www.winsensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1\\_0.pdf](https://www.winsensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf) (дата звернення 15.05.2023).

65. Interface SG90 servo motor with Arduino. URL: <https://www.electrovigyan.com/arduino/sg90-servo-motor/> (дата звернення 15.05.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 74
Зм.	Арк.	№ докум.	Підпис	Дата		

66. HC-06 Pinout, specifications, datasheet and Arduino connection. URL: <https://www.etechnophiles.com/hc06-pinout-specifications-datasheet/> (дата звернення 15.05.2023).

67.2 Channels Solid State Relay Module(Low Trigger). URL: <https://www.cytron.io/p-2-channels-solid-state-relay-module-low-trigger> (дата звернення 15.05.2023).

68. Модуль твердотілого реле на 2 канала OMRON G3MB-202P, 5V для Arduino [#D-5]. URL: <https://bigl.ua/p1008612669-modul-tverdotel'nogo-rele> (дата звернення 15.05.2023).

69. EEPROM Library. URL: <https://docs.arduino.cc/learn/built-in-libraries/eprom> (дата звернення 20.05.2023).

70. Servo. URL: <https://www.arduino.cc/reference/en/libraries/servo/> (дата звернення 20.05.2023).

71. SimpleDHT. URL: <https://reference.arduino.cc/reference/en/libraries/simpleshdt/> (дата звернення 20.05.2023).

72. SimpleDHT. URL: <https://www.ardu-badge.com/SimpleDHT> (дата звернення 20.05.2023).

73. #define. URL: <https://www.arduino.cc/reference/en/language/structure/further-syntax/define/> (дата звернення 20.05.2023).

					КВРКІ 190138.19.01.22 ПЗ	Арк. 75
Зм.	Арк.	№ докум.	Підпис	Дата		

## Додаток А (обов'язковий)

### Лістинг програмної реалізації підсистеми керування мікрокліматом житла

Код програми:

```
#include "EEPROM.h" //бібліотека енергонезалежної пам'яті
#include "Servo.h"//бібліотека для управління сервомоторами
#include "SimpleDHT.h"//бібліотека для роботи з датчиками DHT11 та DHT22
#define DHT_Sensor 6 //контакт для підключення датчика DHT11
SimpleDHT11 dht11(DHT_Sensor);
int temperature_=0, humidity_=0;

Servo myservo_1;
Servo myservo_2;
Servo myservo_3;
Servo myservo_4;

//контакти підключення сервомоторів
#define servo_1 2
#define servo_2 3
#define servo_3 4
#define servo_4 5

//встановлення значень в градусах, коли решітка відкрита, а коли закрита
#define vgrille_open_1 90
#define vgrille_close_1 0

#define vgrille_open_2 90
#define vgrille_close_2 0

#define vgrille_open_3 90
#define vgrille_close_3 0

#define vgrille_open_4 90
#define vgrille_close_4 0

boolean flag_1=0, flag_2=0, flag_3=0, flag_4=0, flag_1force=0, flag_2force=0,
flag_3force=0, flag_4force=0;
boolean flags_1234=0, flags_1234force=0;

#define ventilationunit_general 18
#define ventilationunit_kitchen 19

int time_=5; //інтервал запиту сенсорів
int SensorPins [4][4] = {
    {A0, A1, A2,A3}, // контакти підключення датчиків вуглекислого газу
    {0,0,0,0},
    {0,0,0,0},
    {0,0,0,0},
};

String status1, status2, status3, status4,kitchen;
int limit_1, limit_2, limit_3, limit_4, limit_1s = 800, limit_2s = 800, limit_3s =
800, limit_4s = 800;
int address_0 = 23, address_1 = 25, address_2 = 28, address_3 = 30, address_4 =
32, flag_start = 4, flag_st;
int count = 0, scan_count = 1, ppm_BUF = 0, result = 0, while_count = 0;
```

```

long ppm, timing1 = 0;

void setup() {
  Serial.begin(9600);
  //читання з енергонезалежної пам'яті (EEPROM)
  EEPROM.get(address_0, flag_st);

  //одноразовий запис початкових налаштувань при першому ввімкненні після
  перепрошивання
  if (flag_st != flag_start){
    EEPROM.put(address_0, flag_start);
    EEPROM.put(address_1, limit_1s);
    EEPROM.put(address_2, limit_2s);
    EEPROM.put(address_3, limit_3s);
    EEPROM.put(address_4, limit_4s);
  }
  //читання з енергонезалежної пам'яті (EEPROM)
  EEPROM.get(address_1, limit_1);
  EEPROM.get(address_2, limit_2);
  EEPROM.get(address_3, limit_3);
  EEPROM.get(address_4, limit_4);

  //підключення сервомоторів
  myservo_1.attach(servo_1);
  myservo_2.attach(servo_2);
  myservo_3.attach(servo_3);
  myservo_4.attach(servo_4);

  //встановлення валів сервомоторів в стан закриття вентиляційних решіток
  myservo_1.write(vgrille_close_1);
  myservo_2.write(vgrille_close_2);
  myservo_3.write(vgrille_close_3);
  myservo_4.write(vgrille_close_4);

  //встановлення вентиляторів у стан виключення
  pinMode(ventilationunit_general, OUTPUT); digitalWrite(ventilationunit_general,
HIGH);
  pinMode(ventilationunit_kitchen, OUTPUT); digitalWrite(ventilationunit_kitchen,
HIGH);
}

void loop() {
  //----- Перевірка концентрації вуглекислого газу по 4 датчикам MH-Z19B -----
  --

  if (millis() - timing1 > time_ * 1000){
    timing1 = millis();

    while(while_count < 20){
      int Val = analogRead(SensorPins[0][count]);
      ppm = map(Val, 18, 650, 650, 2000);

      if(scan_count > 1 && (ppm > ppm_BUF - 8 && ppm < ppm_BUF + 8))
      {
        ppm_BUF = ppm; SensorPins[scan_count][count] = ppm;
        scan_count++;
        if(scan_count > 3){
          scan_count = 1; count++; if(count > 3){ count = 0;}
        }
      }
    }
  }
  else
  {

```

```

scan_count = 1;
}

    if(scan_count == 1)
{ SensorPins[scan_count][count] = ppm; ppm_BUF = ppm;
  scan_count++;
  }
  while_count++;
}
while_count = 0;

//----- Превірка температури та вологості по датчику DHT11 -----
---

byte temperature = 0;
byte humidity = 0;
dht11.read(&temperature, &humidity, NULL);
temperature_ = (int)temperature; humidity_ = (int)humidity;
if(humidity_ > 70)
{
digitalWrite(ventilationunit_kitchen, LOW); kitchen = "    ON";
}
else
{digitalWrite(ventilationunit_kitchen, HIGH);          kitchen = "    OFF";
}
    if(temperature_ > 23)
{
digitalWrite(ventilationunit_kitchen, LOW); kitchen = "    ON";
}
else
{
digitalWrite(ventilationunit_kitchen, HIGH);          kitchen = "    OFF";
}
}

//----- Налаштування гранично допустимих значень для датчика MH-Z19B
через телефон -----

if (Serial.available())
{
char c = Serial.read();
if (c == 'A') {
limit_1 += 5; EEPROM.put(address_1, limit_1);
Serial.print(F("GuestRoom +5    LIMIT = ")); Serial.println(limit_1);
}
if (c == 'a') {
limit_1 -= 5; EEPROM.put(address_1, limit_1);
Serial.print(F("GuestRoom -5    LIMIT = ")); Serial.println(limit_1);
}
if (c == 'B') {
limit_2 += 5; EEPROM.put(address_2, limit_2);
Serial.print(F("Someroom +5    LIMIT = ")); Serial.println(limit_2);
}
if (c == 'b') {
limit_2 -= 5; EEPROM.put(address_2, limit_2);
Serial.print(F("Someroom -5    LIMIT = ")); Serial.println(limit_2);}
if (c == 'C') {
limit_3 += 5; EEPROM.put(address_3, limit_3);
Serial.print(F("BedRoom +5    LIMIT = ")); Serial.println(limit_3);
}
if (c == 'c') {
limit_3 -= 5; EEPROM.put(address_3, limit_3);
Serial.print(F("BedRoom -5    LIMIT = ")); Serial.println(limit_3);
}
}

```

```

}
if (c == 'D') {
    limit_4 += 5; EEPROM.put(address_4, limit_4);
    Serial.print(F("Cabinet +5      LIMIT = ")); Serial.println(limit_4);
}
if (c == 'd') {
    limit_4 -= 5; EEPROM.put(address_4, limit_4);
    Serial.print(F("Cabinet -5      LIMIT = ")); Serial.println(limit_4);
}

//----- Запит щодо встановлених гранично допустимих значень та станів роботи
сервомоторів
//та вентиляторів, а також показ значень датчиків MH-Z19B, DHT11 -----

if (c == 'E') {
    Serial.print(F("GuestRoom = ")); Serial.print(SensorPins[3][0]);
    Serial.print(F("    LIMIT = "));   Serial.print(limit_1);
    Serial.print(F("    "));           Serial.println(status1);

    Serial.print(F("Someroom = ")); Serial.print(SensorPins[3][1]);
    Serial.print(F("    LIMIT = "));   Serial.print(limit_2);
    Serial.print(F("    "));           Serial.println(status2);

    Serial.print(F("BedRoom = ")); Serial.print(SensorPins[3][2]);
    Serial.print(F("    LIMIT = "));   Serial.print(limit_3);
    Serial.print(F("    "));           Serial.println(status3);

    Serial.print(F("Cabinet = ")); Serial.print(SensorPins[3][3]);
    Serial.print(F("    LIMIT = "));   Serial.print(limit_4);
    Serial.print(F("    "));           Serial.println(status4);

    Serial.print(F("Temperature= ")); Serial.print(temperature_);
Serial.println(F("*C"));
    Serial.print(F("Humidity = ")); Serial.print(humidity_);
Serial.print(F("%"));
    Serial.println(kitchen);
}

//----- Примусове ввімкнення вентилятора та сервомотора через команду -----
-

if (c == 'F') {Serial.println(F("GuestRoom ON"));  flag_1force = 1;}
if (c == 'f') {Serial.println(F("GuestRoom OFF")); flag_1force = 0;}
if (c == 'G') {Serial.println(F("Someroom ON"));   flag_2force = 1;}
if (c == 'g') {Serial.println(F("Someroom OFF"));  flag_2force = 0;}
if (c == 'H') {Serial.println(F("BedRoom ON"));    flag_3force = 1;}
if (c == 'h') {Serial.println(F("BedRoom OFF"));   flag_3force = 0;}
if (c == 'I') {Serial.println(F("Cabinet ON"));    flag_4force = 1;}
if (c == 'i') {Serial.println(F("Cabinet OFF"));   flag_4force = 0;}

}

//----- перевірка умов для ввікнення вентиляювання по датчикам MH-Z19B -----

if(SensorPins[3][0] > limit_1 + 5){flag_1 = 1;}
if(SensorPins[3][0] < limit_1 - 5){flag_1 = 0;}
if(SensorPins[3][1] > limit_2 + 5){flag_2 = 1;}
if(SensorPins[3][1] < limit_2 - 5){flag_2 = 0;}
if(SensorPins[3][2] > limit_3 + 5){flag_3 = 1;}
if(SensorPins[3][2] < limit_3 - 5){flag_3 = 0;}
if(SensorPins[3][3] > limit_4 + 5){flag_4 = 1;}
if(SensorPins[3][3] < limit_4 - 5){flag_4 = 0;}

```

```

//----- ввімкнення сервомоторів -----
if(flag_1 == 1 || flag_1force == 1){myservo_1.write(vgrille_open_1); status1 =
"ON";}
else{myservo_1.write(vgrille_close_1); status1 = "OFF";}

if(flag_2 == 1 || flag_2force == 1){myservo_2.write(vgrille_open_2); status2 =
"ON";}
else{myservo_2.write(vgrille_close_2); status2 = "OFF";}

if(flag_3 == 1 || flag_3force == 1){myservo_3.write(vgrille_open_3); status3 =
"ON";}
else{myservo_3.write(vgrille_close_3); status3 = "OFF";}

if(flag_4 == 1 || flag_4force == 1){myservo_4.write(vgrille_open_4); status4 =
"ON";}
else{myservo_4.write(vgrille_close_4); status4 = "OFF";}

//----- ввімкнення головного вентилятора -----

if (flag_1 == 1 || flag_2 == 1 || flag_3 == 1 || flag_4 == 1)
{
    flags_1234 = 1;}else{flags_1234 = 0;
}

if (flag_1force == 1 || flag_2force == 1 || flag_3force == 1 || flag_4force ==
1)
{
    flags_1234force = 1;
}
else
{
    flags_1234force = 0;
}

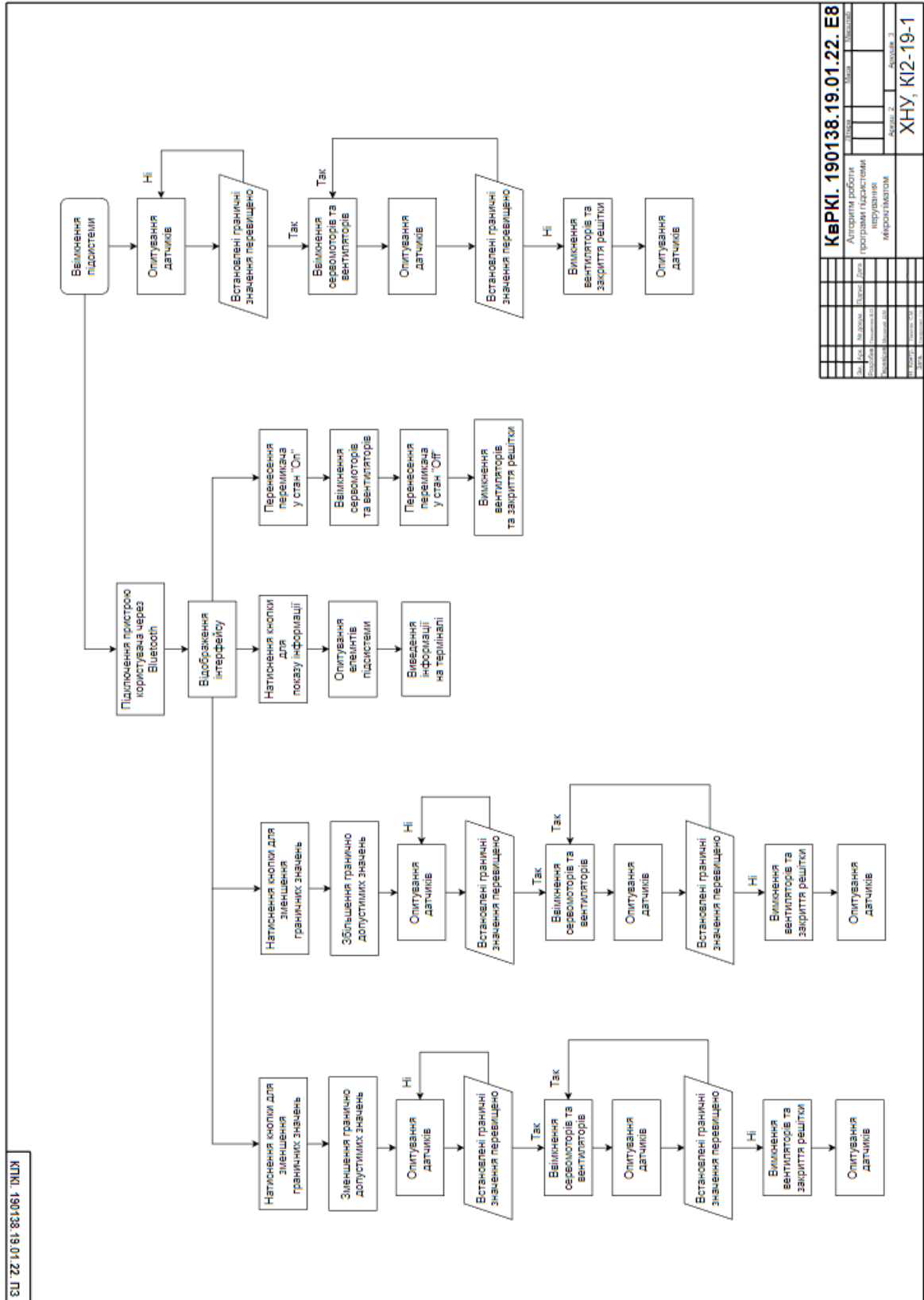
if (flags_1234 == 1 || flags_1234force == 1)
{
    digitalWrite(ventilationunit_general, LOW);
}
else
{
    digitalWrite(ventilationunit_general, HIGH);
}
}

```



## Додаток В (обов'язковий)

Копія креслення «Алгоритм роботи підсистеми керування мікрокліматом»

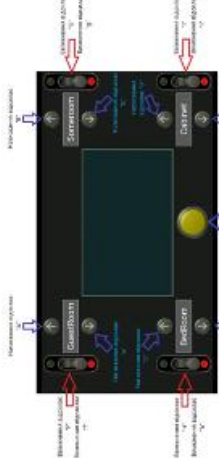


КвРКІ. 190138.19.01.22. Е8		Лист	1	з	1
Алгоритм роботи програми підсистеми керування мікрокліматом		Лист	1	з	1
ХНУ, КІ2-19-1		Лист	2	з	3


## Додаток Г (обов'язковий)

Копія креслення «Інтерфейс проекту мобільного додатку при роботі підсистеми»

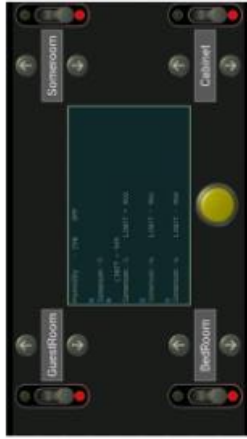
КПКІ. 190138.19.01.22.ПЗ



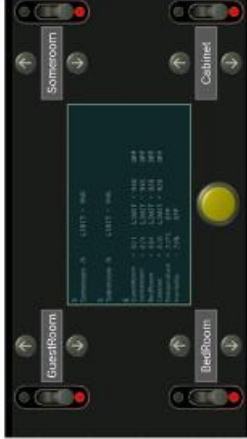
Призначення елементів мобільного проекту



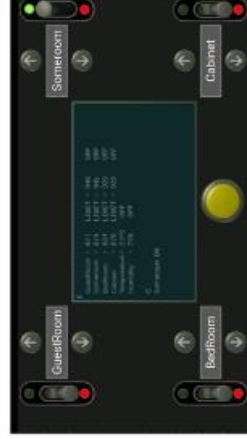
Отримання інформації про стан підсистеми



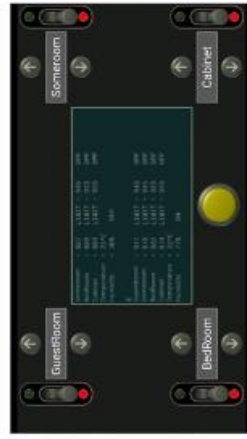
Збільшення та зменшення граничних значень через телефон



Перевірка підсистеми на збереження змін



Примусове ввімкнення вентиляторів та сервомоторів



Перевірка роботи датчиків

<b>КвРКІ. 190138.19.01.22. Е8</b>	
Інтерфейс проекту мобільного додатка при роботі підсистеми	У
Апрель 2019	Апрель 2019
ХНУ, КІ2-19-1	

Ім'я користувача:  
Кафедра КІ

Дата перевірки:  
23.05.2023 13:45:13 EEST

Дата звіту:  
24.05.2023 09:44:27 EEST

ID перевірки:  
1015205386

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005591

Назва документа: Письменюка\_Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумн...

Кількість сторінок: 81 Кількість слів: 10160 Кількість символів: 77385 Розмір файлу: 4.60 MB ID файлу: 1014883050

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

## 6.05% Схожість

Найбільша схожість: 0.79% з джерелом з Бібліотеки (ID файлу: 1008419808)

5.69% Джерела з Інтернету 304 ..... Сторінка 83

1.83% Джерела з Бібліотеки 115 ..... Сторінка 85

## 0.02% Цитат

Цитати 1 ..... Сторінка 86

Посилання 1 ..... Сторінка 86

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 6

Підозріле форматування 21  
сторінка

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 1.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилки в документах: 15%**

ID: 113925 Назва: БКР Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумний будинок» Додано в БД: 2023-05-24 Автора: В.О. Письменюк Керівники: Д.М. Медзатий Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	67461	606	1215 (2%)	18 (3%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Письменюк Володимир Олександрович

Тема: Підсистема керування мікрокліматом житла у кіберфізичній системі  
«Розумний будинок»

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   77  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення підсистеми керування мікрокліматом житла у кіберфізичній системі «Розумний будинок».

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи виконано дослідження предметної області. Крім цього, в першому розділі виконано постановку задачі подальшого дослідження. В другому розділі кваліфікаційної роботи проведено проектування підсистеми керування мікрокліматом житла у кіберфізичній системі «Розумний будинок». В третьому розділі кваліфікаційної роботи виконано програмно-апаратну реалізацію підсистеми керування мікрокліматом житла у кіберфізичній системі «Розумний будинок».

4. Позитивні сторони роботи: Висока практична цінність роботи.

5. Негативні сторони роботи:

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному інженерно-технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: відмінно (4.75/A)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Мергшиш В'ячеслав Володимирович,  
зав. кафедр. АІТгеР

“ ” \_\_\_\_\_ 2023 р.

 (підпис)

Завідувачу кафедри КІС  
д-р.техн.наук, проф. Говорущенко Т. О.

Письменюка Володимира Олександровича

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-19-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

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

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

22 травня 2023 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Підсистема керування мікрокліматом житла у кіберфізичній системі «Розумний будинок»

Автор: Письменюк Володимир Олександрович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Медзатий Д.М, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

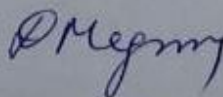
Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) найбільшу схожість встановлено з одним документом і становить вона 0.67% в частині загальноприйнятої термінології.

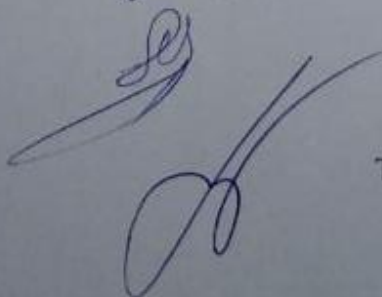
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 6.05% і адресується до 419 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



Д. М. Медзатий

Гарант ОПП



С. М. Лисенко

Завідувач кафедри КІС

Т. О. Говорущенко