

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ**

Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК  
рішенням кафедри радіотехніки та радіоелектронних систем  
від \_\_\_\_\_ 2025 року, протокол № \_\_\_\_.  
Завідувач кафедри доктор фіз.-мат. наук, професор  
\_\_\_\_\_ Ігор АНІСІМОВ

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему:

**«СИСТЕМА РОЗУМНОГО БУДИНКУ НА МІКРОКОНТРОЛЕРІ АРДУІНО»**

**Виконала:**

студент 4-го курсу  
денної форми навчання  
спеціальності 172 – Електронні комунікації та радіотехніка  
ОПП «Інформаційна безпека телекомунікаційних систем і мереж»  
Єременко Олександр Станіславович \_\_\_\_\_

**Науковий керівник:**

доцент кафедри радіотехніки  
та радіоелектронних систем  
Жиров Геннадій Борисович \_\_\_\_\_

**Рецензент:**

Старший науковий співробітник  
науково дослідного центру ВІКНУ  
Мирослав Олександрович Коваль \_\_\_\_\_

Засвідчую, що у цій бакалаврській роботі  
немає запозичень з праць інших авторів без  
відповідних посилань

Студент Єременко Олександр Станіславович

**Київ - 2025**

## РЕФЕРАТ

Дипломна робота: 44 с., 0 табл., 25 рис., 10 джерел.

СИСТЕМА РОЗУМНОГО БУДИНКУ, МІКРОКОНТРОЛЕР ARDUINO.

Об'єкт дослідження – система розумного будинку на мікроконтролері Arduino.

Мета роботи – побудова системи розумного будинку на мікроконтролері Arduino.

Розроблено систему системи розумного будинку на мікроконтролері Arduino UNO. Для цього було проаналізовано вже наявні приклади існуючих систем та створено загальну концепцію та архітектуру системи “розумного будинку”. У другій частині роботи було детально описано схема підключення компонентів у цій системі та опис функціонування усієї системи з перерахуванням усіх результатів у висновку даної роботи.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ I СТАН СУЧАСНИХ СИСТЕМ АВТОМАТИЗАЦІЇ ПОБУТУ .....</b>	<b>6</b>
1.1 Поняття та принципи роботи систем “розумного будинку” .....	6
1.2 Класифікація сучасних систем автоматизації.....	8
1.3 Огляд і порівняння існуючих систем.....	11
1.4 Огляд електронних компонентів, що використовуються .....	15
1.5 Висновки до розділу .....	18
<b>РОЗДІЛ II ОБҐРУНТУВАННЯ І ВИБІР МЕТОДІВ ДЛЯ СТВОРЕННЯ CRM-СИСТЕМИ ДЛЯ ОНЛАЙН-КУРСІВ .....</b>	<b>21</b>
2.1 Загальна концепція та архітектура системи .....	21
2.2 Опис функціональних можливостей системи .....	22
2.3 Схема підключення компонентів .....	23
2.4 Програмна реалізація логіки керування .....	25
2.5 Висновки до розділу .....	39
<b>ВИСНОВКИ.....</b>	<b>41</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....</b>	<b>43</b>
<b>ДОДАТОК .....</b>	<b>44</b>

## ВСТУП

У сучасному світі автоматизація побутових процесів стала важливим елементом повсякденного життя, і системи “розумного будинку” набувають все більшої популярності. Ці технології дозволяють інтегрувати різноманітні пристрої та системи у єдину мережу, що дає змогу ефективно управляти освітленням, опаленням, безпекою та іншими елементами житла, що значно підвищує комфорт і зручність користувачів. Враховуючи швидкий розвиток технологій, системи автоматизації побуту стали доступнішими для широкого кола користувачів, що стимулює їх впровадження в повсякденне життя.

Одним з найбільш популярних і доступних інструментів для створення систем “розумного будинку” є мікроконтролер Arduino. Завдяки своїй простоті використання, відкритому програмному забезпеченню та величезному вибору доступних електронних компонентів, Arduino став ідеальним рішенням для створення персоналізованих та економічних систем автоматизації побуту. Зокрема, використання мікроконтролера Arduino Uno дозволяє реалізувати безліч функцій для автоматизації будинку, таких як управління освітленням, клімат-контролем, охоронними системами тощо.

Актуальність цієї дипломної роботи обумовлена зростаючим попитом на системи автоматизації побутових процесів, а також на необхідності розробки доступних та ефективних рішень для “розумних будинків” на основі доступних технологій. Розробка такої системи дозволить не лише продемонструвати можливості мікроконтролера Arduino, але й створити практичне рішення для покращення якості життя через автоматизацію побутових функцій.

Метою даної роботи є розробка системи “розумного будинку” на базі мікроконтролера Arduino Uno, яка дозволить ефективно автоматизувати основні побутові процеси, такі як управління освітленням, опаленням та безпекою, а також реалізувати просту і зручну систему керування для кінцевого користувача.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Оцінити сучасний стан технологій автоматизації побутових процесів, зокрема в контексті використання Arduino.
2. Розробити загальну концепцію та архітектуру системи “розумного будинку”.
3. Описати функціональні можливості системи та розробити програмну реалізацію логіки керування.
4. Скласти схему підключення компонентів системи.

Об’єктом дослідження є процеси автоматизації побутових функцій за допомогою систем “розумного будинку”, а предметом дослідження є технології, інструменти та методи, що використовуються для створення таких систем. У даному випадку використовувались бюджетні апаратні засоби, за допомогою яких вдалося побудувати систему яка регулює температуру, опалення та безпеку.

Методи дослідження включають аналіз існуючих рішень у сфері автоматизації, проектування та реалізацію програмного забезпечення для керування пристроями, а також тестування розробленої системи в реальних умовах.

Практичне значення одержаних результатів полягає в можливості створення доступної та ефективної системи автоматизації побутових процесів, що дозволить значно покращити якість життя користувачів та підвищити енергоефективність житлових приміщень. Результати дослідження можуть бути використані для створення подібних рішень для побутової автоматизації, що використовують мікроконтролери, зокрема Arduino.

## РОЗДІЛ І

### СТАН СУЧАСНИХ СИСТЕМ АВТОМАТИЗАЦІЇ ПОБУТУ

#### 1.1 Поняття та принципи роботи систем “розумного будинку”

Система “розумного будинку” (Smart Home) є комплексною технологічною інфраструктурою, яка об’єднує різні електронні пристрої, сенсори, системи автоматизації та програмне забезпечення для забезпечення ефективного, зручного та безпечного управління побутовими процесами в житлових приміщеннях. Це може включати такі функції, як контроль освітлення, опалення, кондиціонування повітря, безпека, а також інтеграцію з іншими побутовими пристроями, такими як побутова техніка, медіа-центри, електронні пристрої для управління доступом (замки, двері), та навіть садові системи поливу. Метою таких систем є створення комфортного, енергоефективного та безпечного середовища для мешканців, з автоматизацією багатьох побутових функцій, що дозволяє зменшити потребу в фізичному втручанні [1].

Принципи роботи системи “розумного будинку” ґрунтуються на інтеграції та автоматизації процесів через використання сенсорних технологій, електронних компонентів та програмного забезпечення. Зазвичай у такій системі використовуються різні типи сенсорів, які можуть моніторити зміни навколишнього середовища, такі як температура, вологість, рівень освітленості, рух або навіть якість повітря. Сенсори передають дані на центральний контролер або сервер, який за допомогою спеціальних алгоритмів обробляє ці дані та приймає рішення про те, як реагувати на ті чи інші зміни. Наприклад, якщо датчик температури виявить, що температура в кімнаті занадто висока або низька, система може автоматично регулювати роботу кондиціонера або опалювальної системи.

Одним із важливих компонентів розумного будинку є центральний контролер, який часто виконується на базі мікроконтролера або спеціалізованої платформи, такої як Arduino, Raspberry Pi або більш складні серверні рішення. Цей контролер виступає як “мозок” системи, який збирає інформацію від сенсорів та

приймає відповідні рішення. Наприклад, система може автоматично вимкнути освітлення в кімнаті, коли сенсор руху не фіксує присутність людей, або ж налаштувати ідеальний температурний режим в залежності від часу доби чи погодних умов [2].

Основною перевагою системи “розумного будинку” є її здатність до автоматизації повсякденних завдань. Вона дозволяє значно знизити необхідність у ручному управлінні різними приладами, що забезпечує високу зручність для мешканців. Наприклад, через мобільний додаток або голосових помічників, таких як Amazon Alexa чи Google Assistant, користувачі можуть управляти освітленням, температурою, відеоспостереженням або іншими функціями будинку, навіть перебуваючи поза домом. Це створює додатковий рівень комфорту, адже мешканці можуть керувати своїм будинком з будь-якої точки світу.

Особливу роль у системі “розумного будинку” відіграють енергозберігаючі функції, які допомагають знижувати витрати на комунальні послуги. За допомогою спеціальних сенсорів та алгоритмів управління, система може оптимізувати енергоспоживання. Наприклад, розумний термостат може регулювати температуру в приміщеннях в залежності від часу доби, погодних умов або ж присутності людей у кімнаті. Це дозволяє знижувати витрати на опалення та кондиціонування, а також робить будинок більш екологічно чистим і енергоефективним.

Також системи “розумного будинку” можуть бути інтегровані з пристроями безпеки. Це можуть бути датчики руху, камери відеоспостереження, сигналізації, або навіть замки дверей, які забезпечують високий рівень захисту житла. Наприклад, камера може автоматично включатися при виявленні руху в коридорі, а датчики дверей можуть відправляти сповіщення на мобільний телефон, коли двері або вікна будуть відчинені. Деякі системи також дозволяють власникам домівок замикати чи відкривати двері дистанційно через мобільний додаток або голосові команди.

Інтеграція з різними побутовими пристроями є ще однією важливою складовою розумного будинку. Системи автоматизації можуть контролювати роботу побутової техніки, такої як пральні машини, пилосмоки, мультиварки,

холодильники та інші прилади, що дає змогу планувати та автоматизувати побутові завдання. Наприклад, холодильник може автоматично відправляти повідомлення про необхідність поповнення запасів продуктів, а пральна машина може запускати цикл прання в найзручніший час для власника, економлячи час та енергію.

Головною метою створення системи “розумного будинку” є забезпечення високого рівня комфорту, безпеки та енергоефективності. Однак варто зазначити, що впровадження таких технологій потребує певних початкових інвестицій у необхідне обладнання, налаштування та інтеграцію систем. Проте з огляду на переваги, які ці системи надають, такі витрати швидко окупаються, завдяки зниженню витрат на енергоносії та покращенню рівня життя мешканців [3].

## **1.2 Класифікація сучасних систем автоматизації**

Сучасні системи автоматизації можна класифікувати за кількома основними критеріями, включаючи рівень інтеграції, типи автоматизованих процесів, технології та підходи до управління, а також функціональні можливості. Розуміння цих класифікацій дозволяє точніше визначити можливості й обмеження кожної системи, їхню застосовність у контексті розумного будинку, що є важливим при виборі відповідної технології або розробці системи автоматизації.

За рівнем інтеграції системи автоматизації можна поділити на локальні, централізовані та розподілені. Локальні системи автоматизації є найбільш базовими і зазвичай автоматизують конкретні функції в межах однієї зони або приміщення. Наприклад, система, яка лише автоматично вмикає і вимикає освітлення або регулює температуру в кімнаті, не взаємодіючи з іншими частинами будинку. Такі системи прості в реалізації та зазвичай не потребують складних мережевих зв'язків, але вони мають обмежену функціональність [4].

Централізовані системи автоматизації управляються через єдиний контрольний центр, який об'єднує всі елементи системи в одну платформу. Всі пристрої та датчики, включаючи освітлення, клімат-контроль, охоронні системи, мультимедіа, підключені до центрального сервера або контролера, який забезпечує

їх синхронізацію і управління через єдиний інтерфейс. Центральна система автоматизації дозволяє більш ефективно керувати всіма аспектами житлового середовища з одного місця, надаючи користувачам можливість швидко налаштовувати і змінювати режими роботи будинку. Це також дозволяє здійснювати моніторинг стану системи та отримувати сповіщення про будь-які неполадки або події в будинку.

Розподілені системи автоматизації відрізняються від централізованих тим, що компоненти системи не залежать від єдиного центрального сервера або контролера. Кожен пристрій, як правило, має свій власний процесор та програмне забезпечення для виконання своїх функцій. Всі пристрої в такій системі взаємодіють між собою за допомогою мережі, і кожен з них може виконувати певні автономні завдання. Розподілені системи є більш гнучкими, оскільки дозволяють додавати нові пристрої без необхідності перебудови центральної інфраструктури. Вони також мають високу стійкість до збоїв, оскільки в разі відмови одного компонента система продовжить функціонувати, оскільки інші пристрої залишаються працездатними [5].

З точки зору автоматизованих процесів, сучасні системи можуть включати різноманітні функції для покращення комфортності, безпеки та енергоефективності. Однією з основних функцій є автоматизація освітлення, яка може включати в себе різноманітні аспекти: від простого вмикання/вимикання світла в залежності від часу доби або наявності людей у приміщенні до більш складних сценаріїв, таких як зміна яскравості або кольору освітлення залежно від обставин (наприклад, для створення відповідної атмосфери). Це дозволяє значно знизити витрати енергії і підвищити зручність використання системи.

Клімат-контроль є ще однією важливою складовою систем автоматизації. Він включає автоматичне регулювання температури, вологості та якості повітря, забезпечуючи комфортні умови для мешканців. Це може включати використання термостатів, кондиціонерів, обігрівачів, які регулюються на основі датчиків температури та вологості. Вони можуть працювати в синхронії з іншими компонентами будинку для досягнення оптимальних умов [6].

Безпекові системи також є важливою складовою автоматизованих будинків. Вони можуть включати системи відеоспостереження, датчики руху, розпізнавання осіб, охоронні сигналізації, автоматичні замки та інші компоненти, що забезпечують безпеку. Такі системи дозволяють не лише запобігти злочинам, але й контролювати доступ до будинку, перевіряти, чи всі двері та вікна закриті, і навіть автоматично зв'язуватись із поліцією чи охоронною компанією у разі спрацювання сигналізації.

Побутові пристрої, такі як пральні машини, холодильники, кавоварки, посудомийні машини та інші, також можуть бути інтегровані в систему автоматизації. Наприклад, пральна машина може почати роботу в певний час, коли власники будинку не вдома, або холодильник може автоматично оптимізувати споживання енергії в залежності від кількості продуктів всередині.

З точки зору технологій, сучасні системи автоматизації використовують різні комунікаційні стандарти для з'єднання пристроїв. Найпоширенішими є Wi-Fi, Bluetooth, ZigBee, Z-Wave та інші. Вибір конкретної технології залежить від вимог до дальності дії, енергоспоживання та пропускну здатності. Wi-Fi зазвичай використовують для з'єднання пристроїв із високою пропускну здатністю, наприклад, для відеоспостереження або потокового аудіо/відео. ZigBee та Z-Wave є більш енергоефективними технологіями, які часто використовуються для створення мережі для сенсорів і невеликих пристроїв [7].

Важливим аспектом є також інтеграція системи автоматизації з хмарними платформами. Це дозволяє користувачам управляти системою з будь-якого місця за допомогою мобільних додатків або веб-інтерфейсів, отримувати сповіщення про події та моніторити стан дому в реальному часі.

Крім того, існує тенденція до використання штучного інтелекту та машинного навчання для адаптації системи до звичок і потреб мешканців будинку. Це дозволяє автоматизованим системам ставати більш інтелектуальними, здатними передбачати бажання власників і адаптуватися до їхніх звичок, створюючи максимально комфортні умови без необхідності постійно вносити ручні налаштування.

### 1.3 Огляд і порівняння існуючих систем

Огляд і порівняння існуючих систем автоматизації будинку є важливим етапом у виборі оптимального рішення для забезпечення комфорту, безпеки та енергоефективності. Сучасний ринок пропонує безліч варіантів для створення “розумного будинку”, і для більш детального порівняння можна розглянути три популярні системи: Amazon Alexa, Google Nest та Apple HomeKit. Кожна з цих систем має свої особливості, переваги та недоліки, що дозволяє користувачам вибирати найбільш підходящий варіант залежно від їхніх потреб.

Amazon Alexa є однією з найпоширеніших і найвідоміших систем для автоматизації будинку. Це хмарна платформа, що працює через голосове керування за допомогою голосового помічника Alexa [8]. Вона підтримує широкий спектр пристроїв і є однією з найбільш універсальних систем у плані інтеграції з різними технологіями (рис. 1.1).

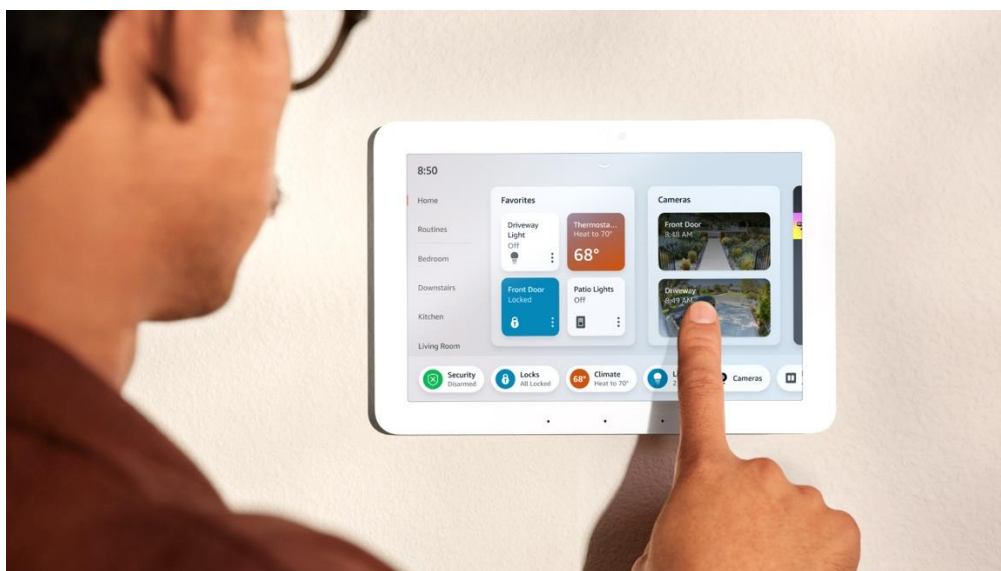


Рисунок 1.1 – Amazon Alexa

Переваги:

- Широка підтримка пристроїв: Alexa працює з великою кількістю пристроїв від різних виробників. Це дозволяє створювати

багатофункціональну систему автоматизації, що охоплює все від освітлення до систем безпеки і побутової техніки.

- **Голосове управління:** Однією з основних переваг є можливість голосового управління пристроями, що забезпечує зручність і простоту використання.
- **Широкий вибір аксесуарів:** Оскільки Alexa підтримує багато пристроїв, можна легко додавати нові елементи до системи автоматизації.
- **Потужна хмарна платформа:** Система працює через Amazon Cloud, що дозволяє отримувати доступ до налаштувань і управління будинком з будь-якої точки світу через мобільний додаток або веб-інтерфейс.

Недоліки:

- **Залежність від Інтернету:** Оскільки Alexa працює через хмару, будь-яка втрата підключення до Інтернету може порушити роботу системи.
- **Обмежена безпека даних:** Оскільки система є хмарною, існують певні ризики з безпекою особистих даних, що зберігаються на серверах Amazon.
- **Складність у налаштуванні інтеграції з деякими пристроями:** Хоча система підтримує велику кількість пристроїв, іноді інтеграція може бути не такою безпроблемною, як у конкурентів.

Google Nest — це система автоматизації, яка поєднує в собі продукти для управління кліматом, безпекою та іншими аспектами розумного будинку. Вона інтегрована з Google Assistant, що дає можливість керувати пристроями за допомогою голосу або через мобільний додаток (рис. 1.2).



Рисунок 1.2 – Google Nest

#### Переваги:

- Інтеграція з Google Assistant: Зручний доступ до голосового управління через Google Assistant, що дозволяє інтегрувати автоматизацію з іншими продуктами Google, такими як смартфони, смарт-годинники та інші пристрої.
- Інтуїтивно зрозумілий інтерфейс: Додаток Google Home має простий і зрозумілий інтерфейс, що дозволяє легко налаштовувати і управляти пристроями [9].
- Інтелектуальне управління кліматом: Google Nest відомий своїми термостатами, які автоматично налаштовують температуру в будинку залежно від звичок мешканців, що дозволяє зменшити споживання енергії.
- Безпечність і конфіденційність: Google активно працює над захистом даних, забезпечуючи високий рівень безпеки і конфіденційності користувачів.

#### Недоліки:

- Обмежена кількість підтримуваних пристроїв: Хоча система інтегрується з багатьма продуктами, кількість сумісних пристроїв може бути меншою порівняно з Alexa.

- Залежність від Інтернету: Подібно до інших хмарних систем, функціонування Google Nest залежить від підключення до Інтернету, що може бути проблемою під час збоїв у мережі.
- Складності з інтеграцією деяких сторонніх пристроїв: Для інтеграції деяких пристроїв може знадобитися спеціальне налаштування, і не всі пристрої будуть працювати належним чином.

Apple HomeKit — це система автоматизації для пристроїв Apple, яка дозволяє управляти розумним будинком за допомогою iPhone, iPad або Siri. Вона забезпечує високу рівень інтеграції з продуктами Apple, що особливо привабливо для користувачів, які вже використовують техніку Apple (рис. 1.3).

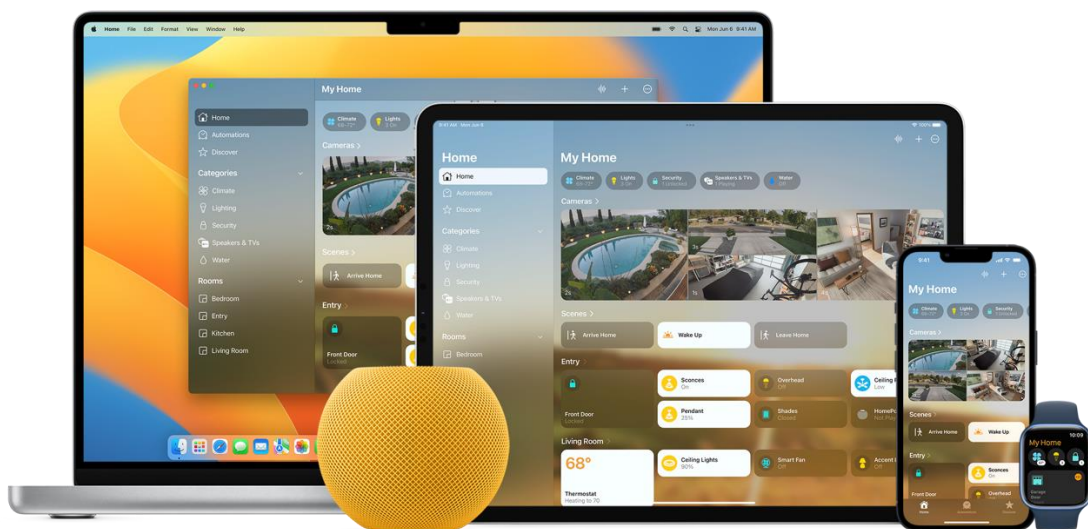


Рисунок 1.3 – Apple HomeKit

#### Переваги:

- Безпека і конфіденційність: Apple славиться своєю увагою до безпеки та конфіденційності даних користувачів. Дані, що зберігаються і передаються через HomeKit, зашифровані і захищені.
- Гарна інтеграція з іншими продуктами Apple: HomeKit ідеально інтегрується з іншими продуктами Apple, такими як iPhone, iPad, Apple Watch, а також забезпечує голосове управління через Siri [10].

- Широка підтримка пристроїв: HomeKit підтримує безліч пристроїв від різних виробників, що дозволяє створювати комплексні системи автоматизації будинку.
- Автономність і локальна робота: HomeKit може працювати без Інтернету завдяки підтримці локальної мережі, що дає можливість автоматизації навіть при відсутності доступу до Інтернету.

Недоліки:

- Залежність від екосистеми Apple: Для повної функціональності система вимагає пристроїв Apple, таких як iPhone або iPad. Це обмежує доступність для користувачів, які не є частиною екосистеми Apple.
- Вища ціна пристроїв: Продукти Apple зазвичай дорожчі за аналогічні продукти від інших виробників, що може стати бар'єром для деяких користувачів.
- Обмежена інтеграція з іншими платформами: Хоча HomeKit підтримує багато пристроїв, він може мати меншу сумісність із пристроями інших платформ порівняно з Alexa.

Кожна з цих систем має свої сильні та слабкі сторони. Amazon Alexa найбільше підходить для тих, хто хоче створити багатофункціональну систему з широкою підтримкою пристроїв і не проти працювати з хмарними платформами. Google Nest приваблює інтеграцією з Google Assistant та інтелектуальним управлінням кліматом, в той час як Apple HomeKit є ідеальним вибором для користувачів екосистеми Apple, що шукають високий рівень безпеки та локальну роботу системи.

#### **1.4 Огляд електронних компонентів, що використовуються**

Електронні компоненти, що використовуються в системах автоматизації будинку, складають основу для роботи «розумних будинків» та забезпечують їх функціонування. Ці компоненти можна класифікувати за кількома напрямками:

сенсори, виконавчі механізми, контролери, джерела живлення та інтерфейси зв'язку.

Сенсори в системах автоматизації будинку виконують ключову роль у зборі інформації про навколишнє середовище. Вони можуть вимірювати різні фізичні величини та перетворювати їх на електричні сигнали для подальшої обробки. Наприклад, температурні сенсори (як-от термістори або цифрові датчики типу DS18B20) забезпечують точне вимірювання температури в приміщенні, що дозволяє автоматично регулювати систему опалення чи кондиціонування. Сенсори вологості, як-от DHT11 або DHT22, вимірюють рівень вологості в повітрі і застосовуються для регулювання системи вентиляції. Датчики руху на основі інфрачервоних (PIR) або ультразвукових технологій використовуються для виявлення руху в кімнатах, що дозволяє автоматично включати освітлення або запускати інші системи, наприклад, для охорони. Датчики освітленості вимірюють рівень освітленості, що дає змогу автоматично регулювати рівень світла в приміщеннях залежно від зовнішнього освітлення. Датчики диму та газу, зокрема для виявлення небезпечних концентрацій СО чи метану, використовуються для забезпечення безпеки в будинку, запобігаючи пожежам або отруєнням. Датчики відкриття дверей чи вікон часто використовуються в системах безпеки для виявлення несанкціонованого доступу.

Виконавчі механізми виконують фізичні дії на основі команд, отриманих від контролера. Одним з найпоширеніших компонентів є реле, яке дає змогу контролювати потужні навантаження, наприклад, включати чи вимикати освітлення або побутову техніку. Соленоїди можуть використовуватися для автоматичного відкриття дверей або вікон, перетворюючи електричний сигнал на механічний рух. Сервоприводи дозволяють точно керувати рухом частин системи, таких як жалюзі або штори, регулюючи кут їхнього повороту чи переміщення. Моторизовані вентилятори й клапани можуть автоматично регулювати вентиляцію та температуру, забезпечуючи комфортні умови в приміщеннях.

Контролери або мікроконтролери, такі як Arduino, ESP32 або Raspberry Pi, виконують роль «мозку» системи. Вони отримують сигнали від сенсорів,

обробляють інформацію та надсилають відповідні команди до виконавчих механізмів. Ці пристрої забезпечують централізоване управління всіма компонентами автоматизованої системи. Програмовані логічні контролери (PLC) використовуються в більш складних системах і дозволяють автоматизувати різноманітні процеси, від освітлення до вентиляції та опалення.

Джерела живлення необхідні для забезпечення роботи всіх компонентів системи. Блоки живлення забезпечують стабільне постачання енергії для сенсорів, контролерів і виконавчих механізмів. Вони повинні бути здатні забезпечувати правильну напругу для кожного елемента системи. В деяких випадках, особливо для портативних пристроїв або віддалених датчиків, використовуються батареї або акумулятори, що дає можливість безперервної роботи навіть при відключенні основного джерела живлення.

Інтерфейси зв'язку дозволяють компонуванню системи «розумного будинку» ефективно взаємодіяти між собою. Безпроводні технології, такі як Wi-Fi, Bluetooth, ZigBee та Z-Wave, забезпечують з'єднання між пристроями, що дозволяє створювати розподілену мережу для управління будинком. Wi-Fi зазвичай використовується для підключення компонентів до Інтернету, що дає можливість віддаленого керування через смартфони чи комп'ютери. Bluetooth підходить для короткодіючих з'єднань між пристроями, ZigBee та Z-Wave забезпечують низьке споживання енергії, що робить їх ідеальними для інтеграції в бездротові сенсорні мережі.

Всі ці компоненти працюють разом, створюючи систему автоматизації, яка дозволяє управляти різними аспектами «розумного будинку» — від освітлення та клімат-контролю до безпеки і моніторингу стану будинку. Важливою характеристикою таких компонентів є їхня здатність працювати в реальному часі та інтегруватися в єдину систему, що забезпечує зручність, енергоефективність і безпеку для користувачів.

## 1.5 Висновки до розділу

У цьому розділі було проведено детальний огляд основних аспектів сучасних систем автоматизації побуту, зокрема розглянуто принципи роботи систем «розумного будинку», класифікацію таких систем, а також детальне вивчення електронних компонентів, що використовуються для їх реалізації. Системи автоматизації побуту відіграють важливу роль у розвитку інтелектуальних середовищ, оскільки дозволяють значно підвищити комфорт, безпеку, енергоефективність та зручність користування різними побутовими пристроями та системами. Усі ці системи об'єднуються в єдину автоматизовану екосистему, яка забезпечує взаємодію пристроїв між собою, а також дозволяє користувачам зручно управляти ними.

Аналіз принципів роботи систем «розумного будинку» показав, що їх основною особливістю є інтеграція різних технологій для створення автоматизованих сценаріїв, таких як автоматичне включення освітлення в залежності від часу доби або рівня освітленості, контроль клімату в приміщенні через сенсори температури та вологості, а також управління безпекою за допомогою датчиків руху або камер спостереження. Основні принципи роботи таких систем полягають у зборі даних з навколишнього середовища та їх обробці для прийняття рішень і подальшого управління пристроями через контролери або мобільні додатки.

Вивчення класифікації сучасних систем автоматизації побуту показало, що на ринку представлені різні моделі, які класифікуються за рівнем централізації управління, типом технології зв'язку та ступенем інтеграції з іншими системами. Системи з централізованим управлінням, де є головний контролер або сервер, управляють всіма пристроями з одного місця. Це може бути як програмно, так і через спеціальні апаратні пристрої. Дистрибуційні системи, навпаки, не мають єдиного центрального управління, а кожен пристрій або група пристроїв має власний контролер, що дозволяє створювати більш гнучкі, але складніші у налаштуванні і підтримці системи. Щодо технологій зв'язку, важливими є

протоколи Wi-Fi, ZigBee, Z-Wave та інші, що забезпечують необхідний рівень швидкості та надійності передачі даних.

Порівняння існуючих систем «розумного будинку», таких як Amazon Alexa, Google Nest і Apple HomeKit, дозволило виявити їхні переваги та обмеження. Amazon Alexa пропонує зручність використання завдяки широким можливостям інтеграції з численними пристроями та системами, а також підтримує голосове управління, що робить взаємодію з пристроями інтуїтивно зрозумілою. Однак її налаштування може бути складним для користувачів, які не мають технічного досвіду.

Google Nest відзначається високою енергоефективністю та чудовою інтеграцією з іншими сервісами Google, такими як Google Assistant, що забезпечує зручність голосового управління та доступ до даних через інтерфейс Google Home. Проте система більше орієнтована на користувачів, які вже використовують інші продукти Google, що може обмежити її універсальність. Apple HomeKit, з іншого боку, пропонує високу безпеку та зручність інтеграції з іншими продуктами Apple, але обмежена сумісність з пристроями сторонніх виробників може стати бар'єром для деяких користувачів. Вибір між цими системами залежить від технічних вимог, бюджету та ступеня інтеграції з іншими технологіями, які використовує користувач.

Огляд електронних компонентів, що використовуються в системах автоматизації будинку, підкреслив важливість сенсорів, виконавчих механізмів, контролерів і джерел живлення. Сенсори, які вимірюють різноманітні фізичні параметри, такі як температура, вологість, рівень освітленості, рух, гази тощо, є важливими для точного збору даних та автоматичного регулювання параметрів середовища. Виконавчі механізми, такі як реле, мотори, серво-мотори, використовуються для управління різними пристроями, такими як освітлення, штори, системи вентиляції та опалення. Контролери, зокрема мікроконтролери на базі Arduino, Raspberry Pi або спеціалізовані контролери, відповідають за обробку даних з сенсорів та управління виконавчими механізмами, що дозволяє реалізувати сценарії автоматизації.

Загалом, дослідження показало, що системи «розумного будинку» є потужним інструментом для автоматизації побутових процесів, однак для створення ефективної та надійної системи необхідно ретельно вибирати компоненти та технології, враховуючи не тільки технічні характеристики, але й вимоги до зручності використання, безпеки та енергоефективності. Важливими аспектами є також забезпечення безперебійної роботи всіх компонентів системи, що дозволяє домогтися максимальної ефективності автоматизації та зручності для кінцевих користувачів.

## РОЗДІЛ II

### ОБҐРУНТУВАННЯ І ВИБІР МЕТОДІВ ДЛЯ СТВОРЕННЯ CRM-СИСТЕМИ ДЛЯ ОНЛАЙН-КУРСІВ

#### 2.1 Загальна концепція та архітектура системи

Система “Розумний дім”, яка буде розроблена на базі мікроконтролера Arduino Uno, передбачатиме реалізацію автоматизованого управління низкою побутових функцій, включаючи освітлення, опалення, відкриття та закриття дверей, вікон і воріт, а також функціонуватиме як охоронна система із датчиком руху. Основна ідея полягає у створенні доступного, гнучкого та надійного рішення для автоматизації житлового простору з можливістю подальшого розширення та модернізації.

Архітектура системи буде побудована за централізованим принципом, де плата Arduino Uno виконуватиме роль головного контролера. Вона буде забезпечувати прийом даних з сенсорів, обробку інформації та керування виконавчими елементами. Вся взаємодія користувача з системою здійснюватиметься через інфрачервоний пульт дистанційного керування. ІЧ-приймач, підключений до цифрового входу мікроконтролера, фіксуватиме сигнали від пульта й передаватиме відповідні команди для подальшої обробки.

Для автоматизації фізичних дій, таких як відкривання дверей, вікон і воріт, будуть використані серводвигуни, які підключатимуться до цифрових пінів 3, 5 та 6. Це дозволить системі реагувати на команди користувача та забезпечити керування доступом до приміщення. Температурний режим у приміщенні буде контролюватися за допомогою цифрового датчика DHT22, підключеного до піна 7. Отримані дані про температуру та вологість виводитимуться на LCD-дисплей 16x2, що буде підключений через I2C-інтерфейс (A4 — SDA, A5 — SCL), що дозволить користувачу оперативно відстежувати параметри мікроклімату.

Для реалізації охоронної функції системи буде застосовано PIR-сенсор руху, який виявлятиме присутність у зоні дії. У разі фіксації руху система активуватиме звукову сигналізацію через буюер, підключений до аналогового піна А1, а також світлову індикацію червоним світлодіодом, підключеним до піна А0. Додатково використовуватимуться жовтий та синій світлодіоди, які сигналізуватимуть про інші стани системи, зокрема активне освітлення або опалення.

Усі компоненти будуть змонтовані на макетній платі із застосуванням з'єднувальних дротів, резисторів та відповідного джерела живлення. Такий підхід дозволить протестувати систему в різних режимах, забезпечить простоту обслуговування і подальшого розширення. У перспективі система буде доопрацьована з можливістю додавання бездротових модулів (Wi-Fi, Bluetooth), голосового керування, віддаленого моніторингу через інтернет, а також підключення додаткових сенсорів — наприклад, датчиків диму, газу чи витоків води.

Хотів би додати, що під час вибору для розроблення даної системи я обрав керування через інфрачервоний пульт, оскільки це більш безпечний та локально захищений метод, який не потребує підключення до Інтернету. Також це більш доступний та бюджетніший варіант, ніж система з управлінням через сервер. Але я все ж таки в пункті роботи 1.3 розглянув системи керування за допомогою Інтернету. При цьому мій варіант є хорошою базою для удосконалення та покращення системи з можливістю працювати через мережу Інтернет.

## **2.2 Опис функціональних можливостей системи**

Інтелектуальна система автоматизованого керування постійно відстежує температуру та вологість у приміщенні за допомогою цифрового датчика. Дані зчитуються у реальному часі й відображаються на LCD-дисплеї, що дозволяє користувачеві бачити поточний стан мікроклімату. У разі зниження температури нижче 18°C система автоматично вмикає опалення, а при підвищенні вище 24°C —

вимикає його. Одночасно із цим активуються візуальні індикатори: при нагріванні загоряється червоний світлодіод, а при охолодженні — синій.

Система освітлення реагує на рівень освітленості в кімнаті. Фотоелемент безперервно визначає інтенсивність освітлення. Коли рівень світла падає нижче заданого порогу (наприклад, 300 люксів), автоматично вмикається освітлення, і загоряється відповідний індикатор. Якщо освітлення досягає достатнього рівня, система вимикає лампу, забезпечуючи енергоефективність.

Крім того, система оснащена датчиком руху, який спрацьовує при виявленні присутності людини. У разі фіксації руху система активує світлову або звукову сигналізацію, а при відсутності руху протягом певного часу автоматично вимикає непотрібні пристрої. Завдяки цьому досягається економія електроенергії та підвищується безпека.

Уся система працює на базі мікроконтролера Arduino, який координує взаємодію всіх сенсорів і виконавчих пристроїв. Програма на Arduino обробляє дані, приймає рішення та керує пристроями відповідно до заданих умов. Весь процес відбувається автономно, без необхідності постійного втручання користувача, що робить систему зручною для повсякденного використання та адаптованою до змін навколишнього середовища.

### **2.3 Схеми підключення компонентів**

У процесі побудови системи автоматизованого керування важливою складовою є правильне підключення всіх електронних компонентів до мікроконтролера. Для забезпечення коректної роботи кожного елемента система використовує стандартизовану схему з'єднань, яка враховує логіку обробки сигналів та особливості живлення окремих модулів.

Мікроконтролер Arduino Uno виступає центральним вузлом, до якого підключаються всі сенсори та виконавчі пристрої. Датчик температури та вологості DHT11 підключається до цифрового піну, який підтримує зчитування даних з одного проводу, зазвичай через pin 2 або 3. Світлодіоди, що сигналізують про

роботу опалення або охолодження, під'єднуються до цифрових виходів із відповідними резисторами, що обмежують струм.

Датчик освітленості на основі фоторезистора підключається через дільник напруги до аналогового входу, наприклад А0, що дозволяє вимірювати рівень освітленості в кімнаті. Датчик руху PIR приєднується до цифрового входу, зазвичай pin 7, і передає логічний сигнал у разі фіксації руху. Реле, яке керує живленням освітлення або інших пристроїв, підключається до окремого цифрового виходу, що здатний керувати модулями з високим струмом через транзистор чи модуль реле.

LCD-дисплей з інтерфейсом I2C значно спрощує підключення, оскільки використовує лише два дроти — SDA і SCL, які з'єднуються з відповідними пінами мікроконтролера (А4 та А5 для Arduino Uno).

Загальна схема підключення дозволяє наочно уявити структуру всієї системи, взаємозв'язки між її компонентами та спосіб обміну даними. Вона є основою для подальшої реалізації програмного забезпечення та ефективного функціонування пристрою.

Схема підключення компонентів до Arduino наведена на рисунку 2.1

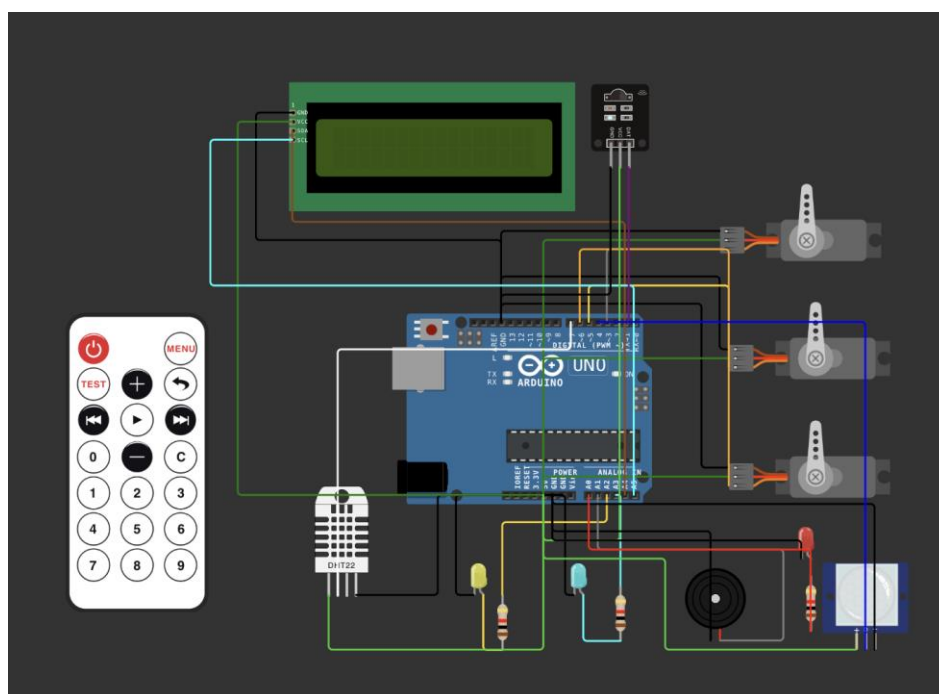


Рисунок 2.1 – Схема підключення компонентів

## 2.4 Програмна реалізація логіки керування

Після запуску системи, мікроконтролер Arduino ініціалізує всі підключені пристрої та компоненти. Серед них: датчик температури та вологості DHT22, LCD-дисплей для виводу інформації, інфрачервоний приймач для прийому команд з пульта, сервомотори для керування дверима, вікном та воротами, а також світлодіоди й буюер для візуальної та звукової індикації. Всі пристрої на початковому етапі знаходяться у неактивному стані: двері, вікна та ворота залишаються зачиненими, освітлення та опалення вимкнені, сигналізація також перебуває у неактивному режимі.

```
void setup() {  
    Serial.begin(9600);  
    IrReceiver.begin(IR_RECEIVE_PIN);  
    dht.begin();  
    lcd.init();  
    lcd.backlight();  
    ...  
    doorServo.attach(3);  
    windowServo.attach(5);  
    gateServo.attach(6);  
    doorServo.write(0);  
    windowServo.write(0);  
    gateServo.write(0);  
    ...  
}
```

На початку роботи на LCD-дисплеї коротко з'являється напис "Smart Home", що вказує на готовність системи до роботи (рис. 2.2).

```
lcd.setCursor(0, 0);  
lcd.print(" Smart Home ");  
delay(3000);  
lcd.clear();
```

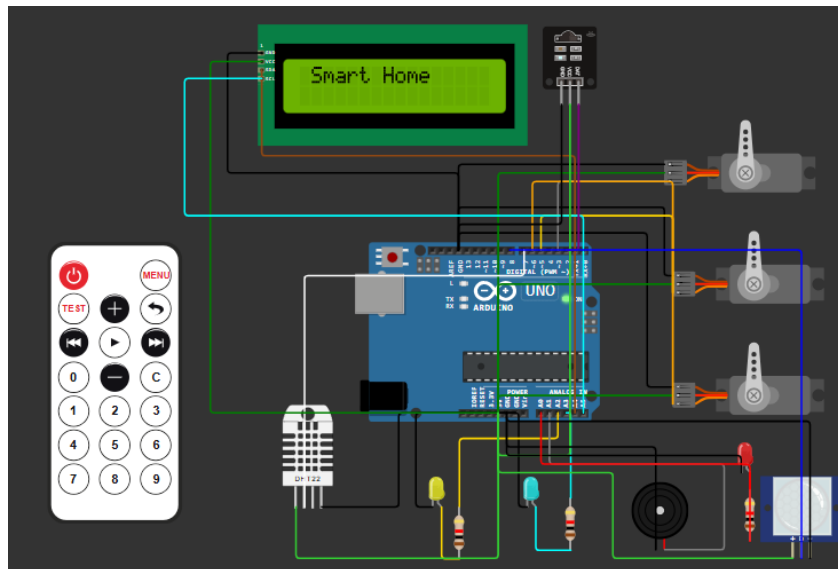


Рисунок 2.2 – На початку роботи на LCD-дисплеї коротко з'являється напис "Smart Home"

Після цього дисплей починає відображати актуальні значення температури та вологості повітря, які оновлюються кожні дві секунди за допомогою датчика DHT22. Якщо користувач не перейшов у ручний режим встановлення цих параметрів, значення температури і вологості постійно оновлюються в реальному часі (рис. 2.3).

```
void updateTemperatureDisplay() {
  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(temperature, 1);
  lcd.print((char)223);
  ...
}
```

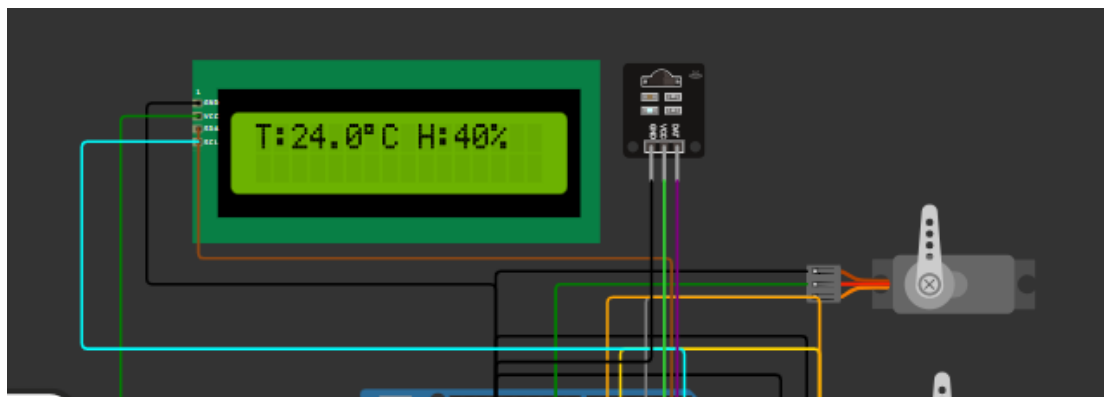


Рисунок 2.3 – Актуальні значення температури та вологості повітря

Система дозволяє користувачеві керувати різними пристроями за допомогою інфрачервоного пульта. Ось як працює кожна з функцій:

### 1. Керування дверима:

- Натискання кнопки "1" (код 0x30) відкриває двері, повертаючи відповідний сервомотор на 90 градусів. На екрані відображається повідомлення "Door: OPEN" (рис. 2.4).

```
case 0x30: doorServo.write(90); lcd.print("Door: OPEN      "); break;
```

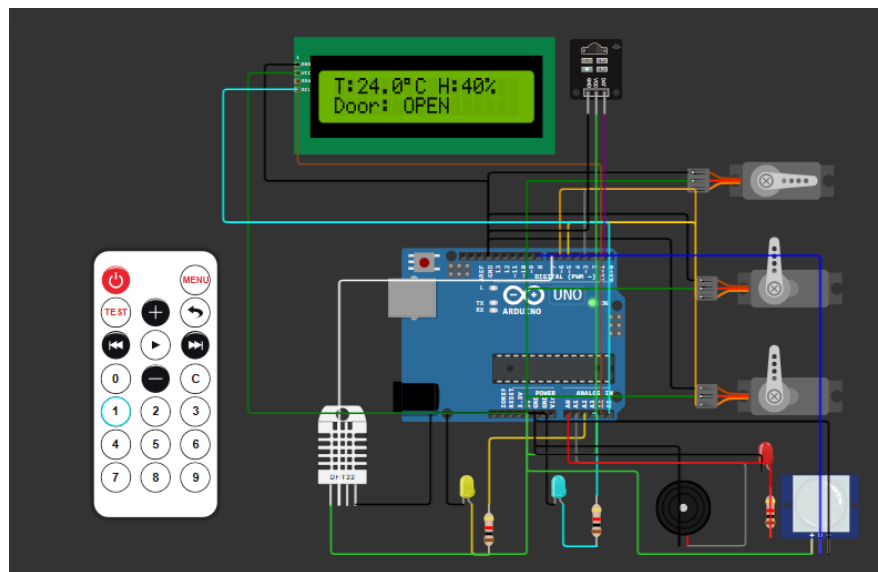


Рисунок 2.4 – Door: OPEN

- Натискання кнопки "2" (код 0x18) зачиняє двері, повертаючи сервомотор у вихідне положення (0 градусів). На екрані відображається "Door: CLOSED" (рис. 2.5).

```
case 0x18: doorServo.write(0); lcd.print("Door: CLOSED      "); break;
```

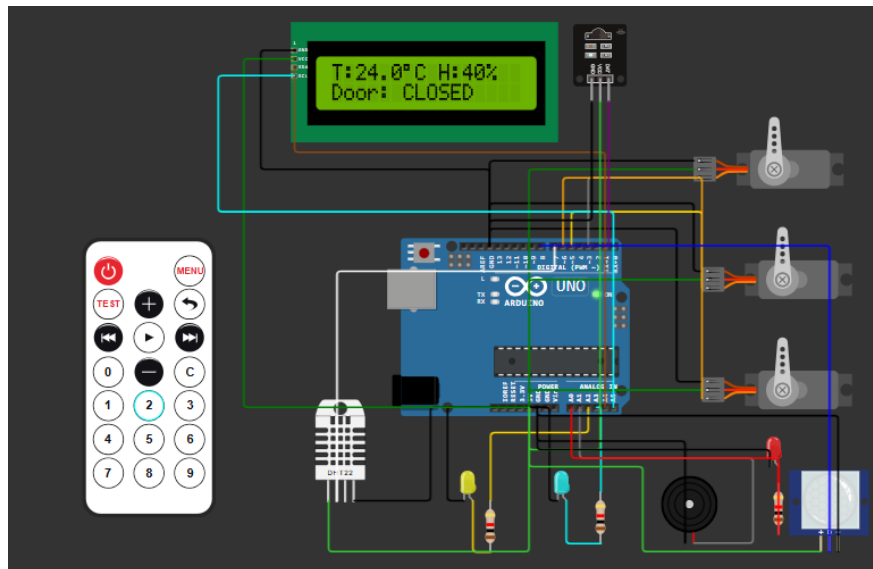


Рисунок 2.5 – Door: CLOSED

## 2. Керування вікнами:

- Натискання кнопки "3" (код 0x7A) відкриває вікна, повертаючи сервомотор на 90 градусів. На дисплеї з'являється "Window: OPEN" (рис. 2.6).

```
case 0x7A: windowServo.write(90); lcd.print("Window: OPEN"); break;
```

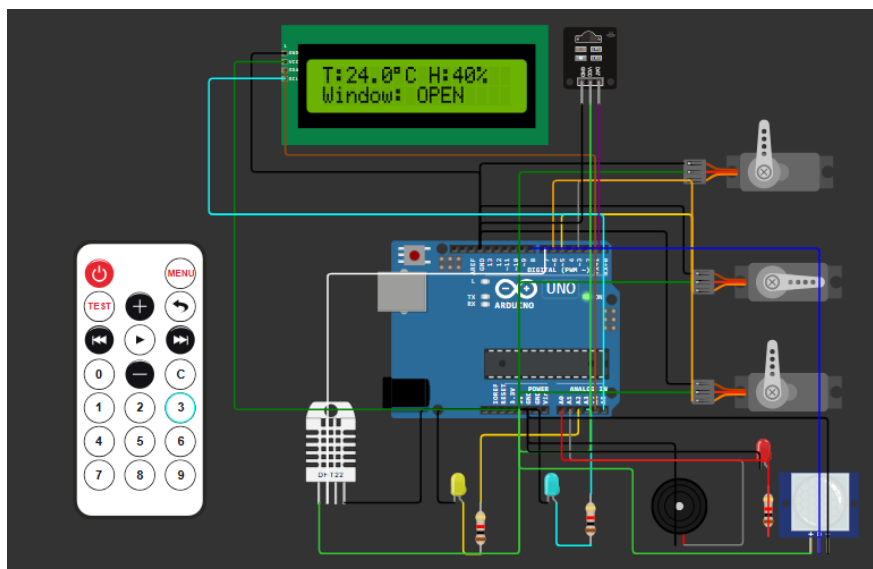


Рисунок 2.6 – Window: OPEN

- Натискання кнопки "4" (код 0x10) закриває вікна, повертаючи сервомотор у початкове положення (0 градусів). На дисплеї відображається "Window: CLOSED" (рис. 2.7).

```
case 0x10: windowServo.write(0); lcd.print("Window: CLOSED "); break;
```

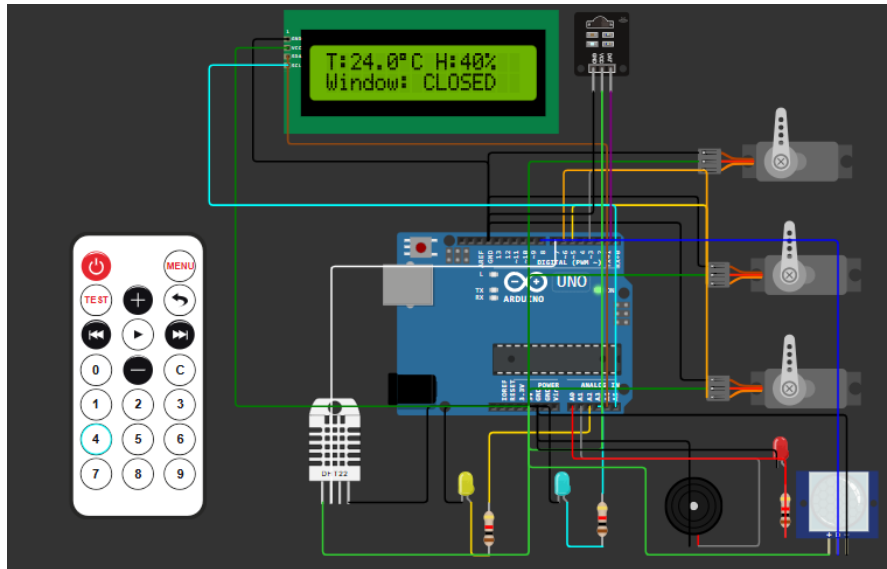


Рисунок 2.7 – Window: CLOSED

### 3. Керування воротами:

- Кнопка "5" (код 0x38) відкриває ворота, що супроводжується написом "Gate: OPEN" на дисплеї (рис. 2.8).

```
case 0x38: gateServo.write(90); lcd.print("Gate: OPEN "); break;
```

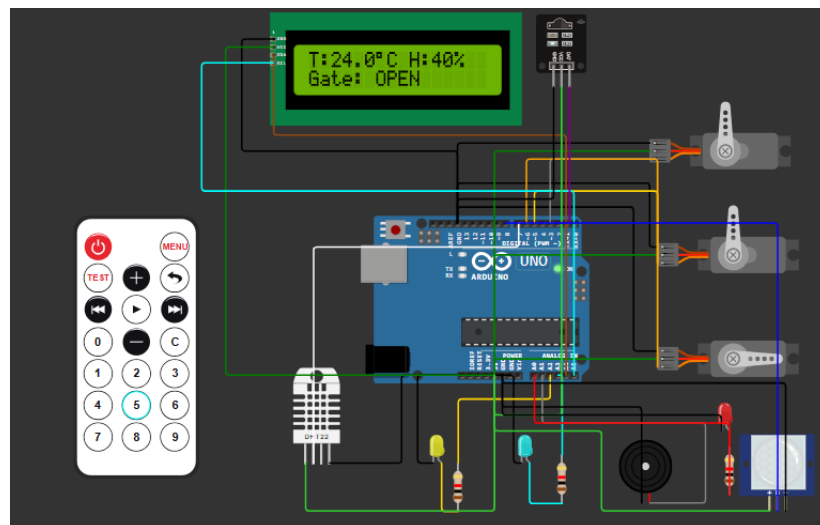


Рисунок 2.8 – Gate: OPEN

- Кнопка "6" (код 0x5A) зачиняє ворота, після чого на дисплеї з'являється "Gate: CLOSED" (рис. 2.9).

```
case 0x5A: gateServo.write(0); lcd.print("Gate: CLOSED"); break;
```

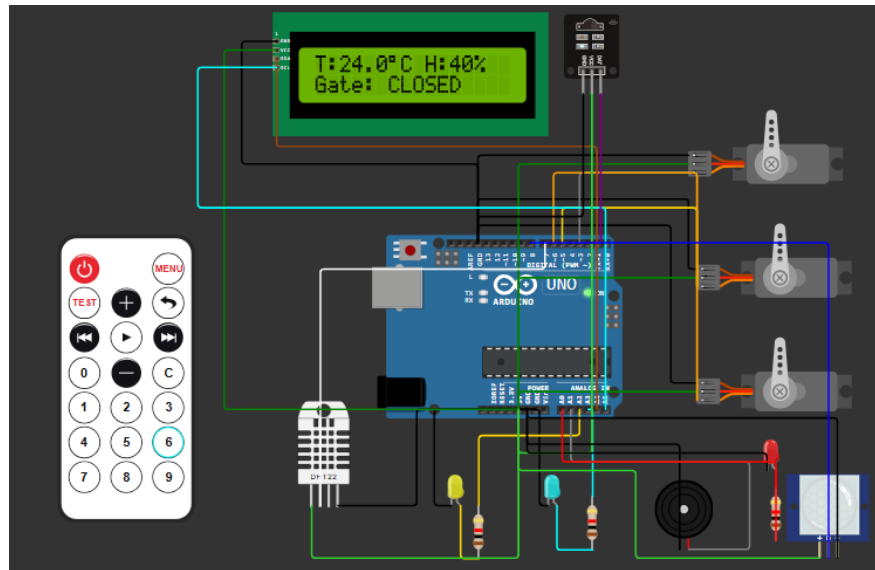


Рисунок 2.9 – Gate: CLOSED

#### 4. Керування освітленням:

- Натискання кнопки "7" (код 0x42) вмикає освітлення, що сигналізується загорянням жовтого світлодіода. На дисплеї з'являється "Light: ON" (рис. 2.10).

```
case 0x42: lightOn = true; digitalWrite(YELLOW_LED, HIGH);  
lcd.print("Light: ON"); break;
```

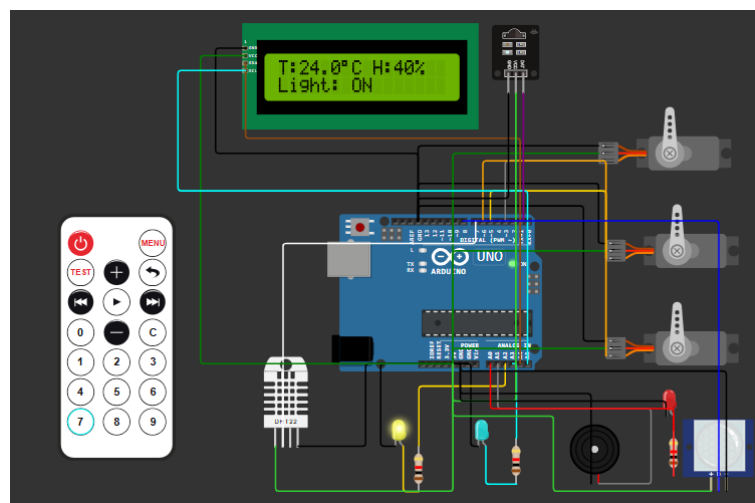


Рисунок 2.10 – Light: ON

- Натискання кнопки "8" (код 0x4A) вимикає освітлення, гаснучи жовтий світлодіод. На дисплеї відображається "Light: OFF" (рис. 2.11).

```
case 0x4A: lightOn = false; digitalWrite(YELLOW_LED, LOW);
lcd.print("Light: OFF      "); break;
```

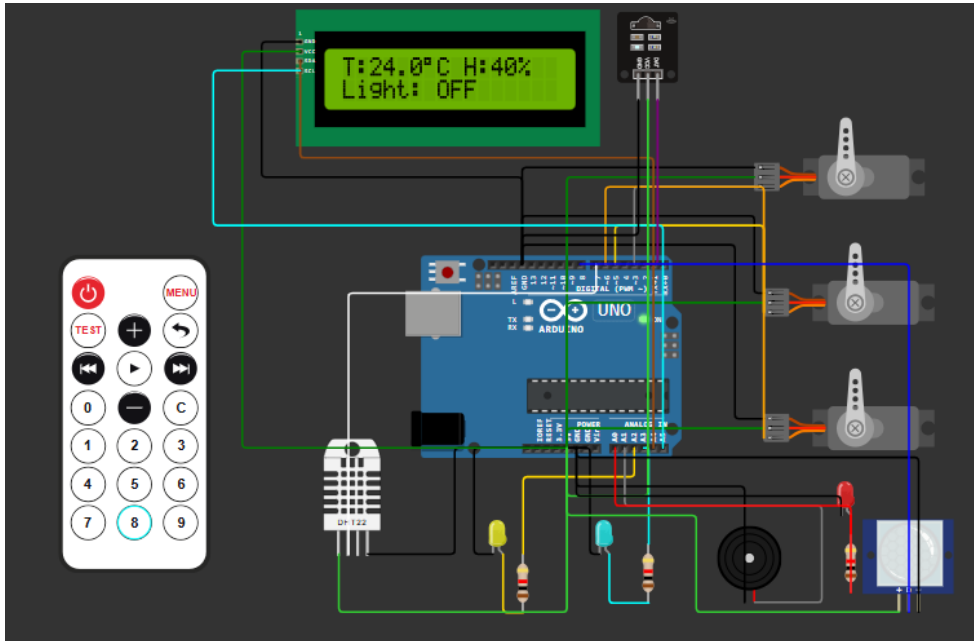


Рисунок 2.11 – Light: OFF

## 5. Керування опаленням:

- Кнопка "9" (код 0x52) активує систему опалення, що супроводжується загорянням синього світлодіода та написом "Heating: ON" на дисплеї (рис. 2.12).

```
case 0x52: heatingOn = true; updateHeatingStatus(); break;
```

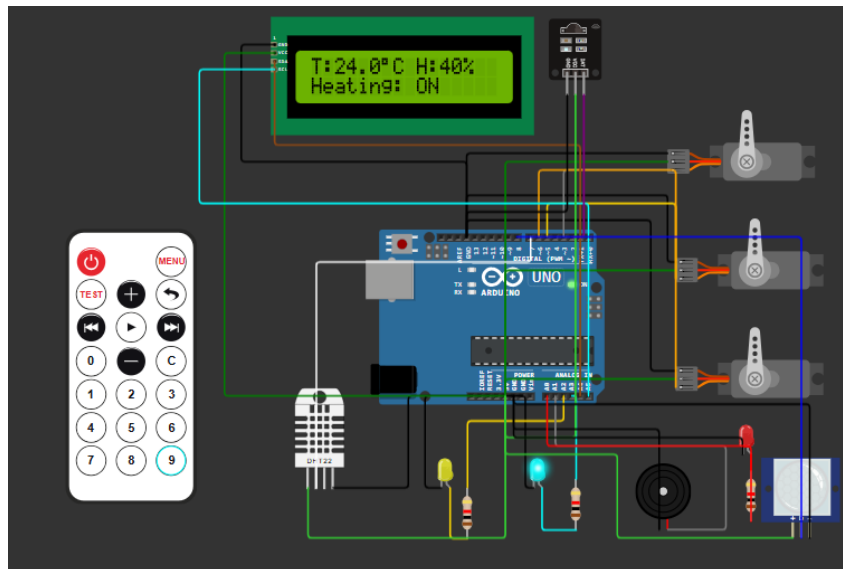


Рисунок 2.12 – Heating: ON

- Кнопка "0" (код 0x68) вимикає опалення, синій світлодіод гасне, а на дисплеї з'являється "Heating: OFF" (рис. 2.13).

```
case 0x68: heatingOn = false; updateHeatingStatus(); break;
```

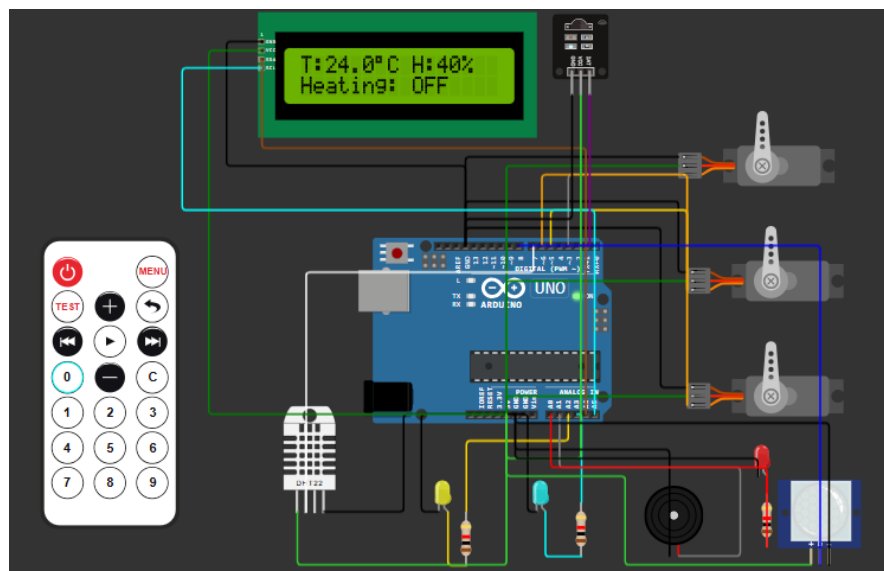


Рисунок 2.13 – Heating: OFF

## 6. Ручне регулювання температури:

- Натискання кнопки "+" (код 0x02) підвищує температуру на 1°C. Це відображається на дисплеї у вигляді "Temp +1°C". При цьому система

переходить у ручний режим, припиняючи автоматичне оновлення температури з датчика (рис. 2.14).

```

case 0x02:

    temperature += 1; manualTempSet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Temp increased  ");
    break;

```

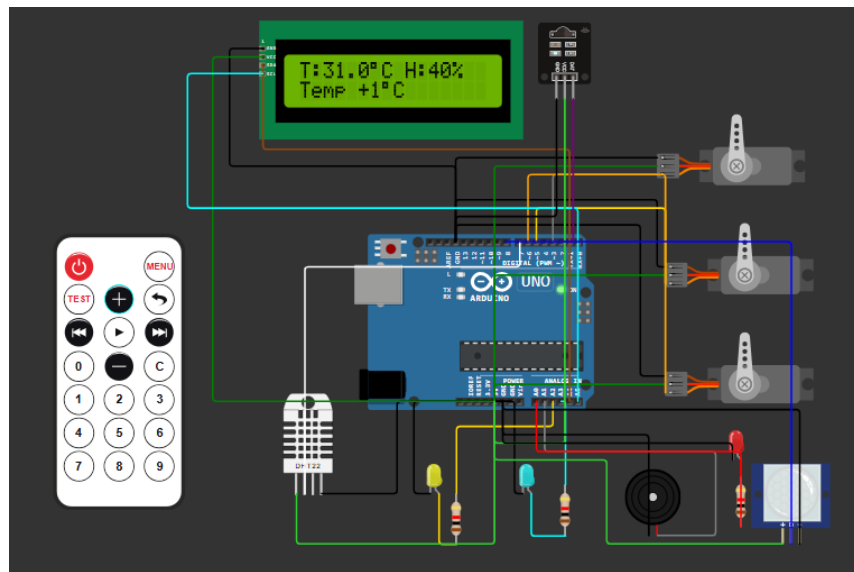


Рисунок 2.14 – Підвищення температури на 1°C

- Натискання кнопки "-" (код 0x98) зменшує температуру на 1°C, виводячи повідомлення "Temp -1°C" (рис. 2.15).

```

case 0x98:

    temperature -= 1; manualTempSet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Temp decreased  ");
    break;

```

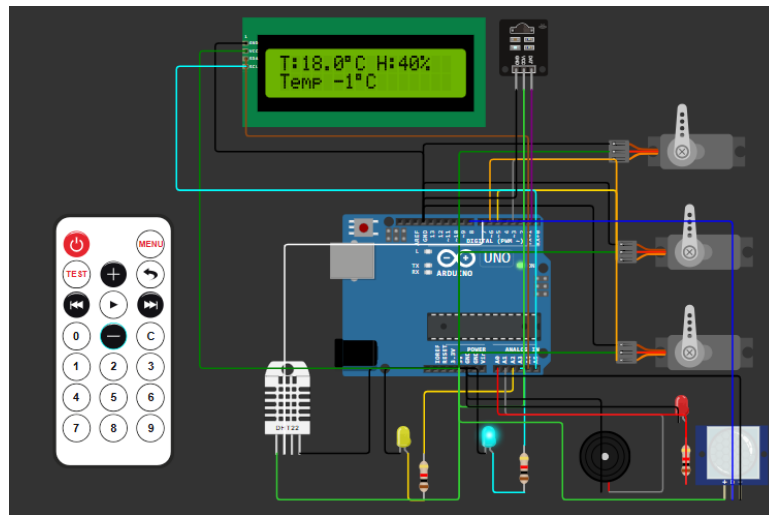


Рисунок 2.15 – Зменшення температури на 1°C

## 7. Ручне регулювання вологості:

- Кнопка ">>" (код 0x90) збільшує рівень вологості на 1%, якщо вона не перевищує 100%. Виводиться повідомлення "Humidity +1%" (рис. 2.16).

```

case 0x90:
    if (humidity < 100) humidity += 1;
    manualHumiditySet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Humidity up      ");
    break;

```

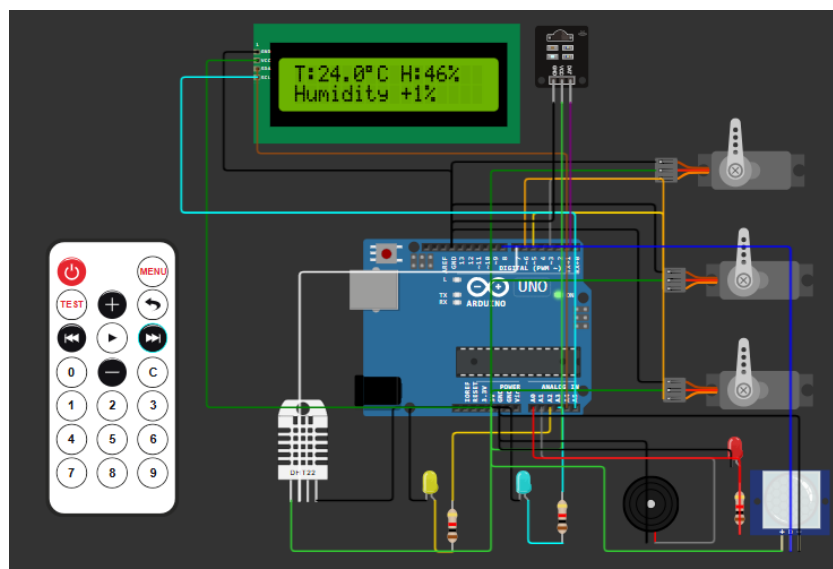


Рисунок 2.16 – Збільшення рівня вологості на 1%

- Кнопка "<<" (код 0xE0) зменшує рівень вологості на 1%, якщо вона більша за 0%. На дисплеї з'являється "Humidity -1%" (рис. 2.17).

```

case 0xE0:
    if (humidity > 0) humidity -= 1;
    manualHumiditySet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Humidity down  ");
    break;

```

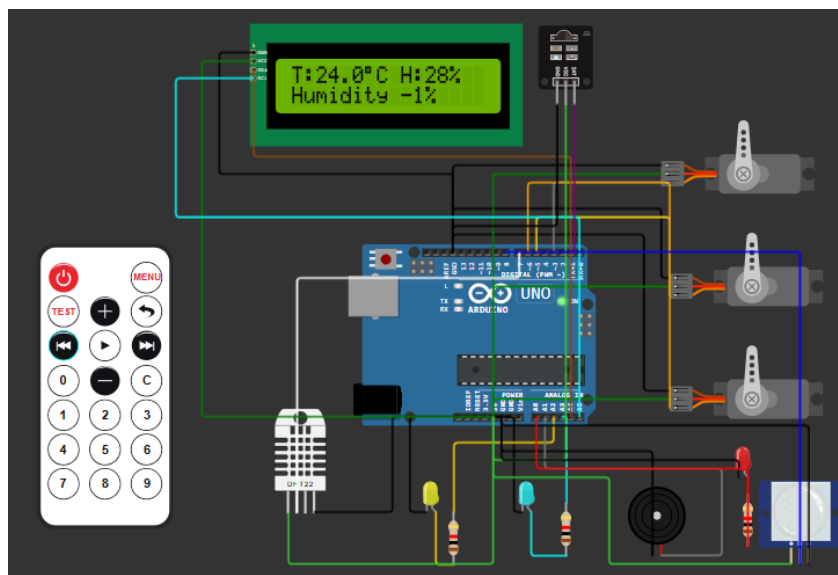


Рисунок 2.17 – Зменшення рівня вологості на 1%

## 8. Скидання ручного керування:

- Кнопка "↶" (код 0xC2) повертає систему до автоматичного зчитування температури та вологості з датчика, скидаючи ручні налаштування. Виводиться повідомлення "Manual: RESET" (рис. 2.18).

```

case 0xC2:

    temperature = 24.0;
    humidity = 40.0;
    manualTempSet = true;
    manualHumiditySet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Temp: 24C Hum:40%");
    break;

```

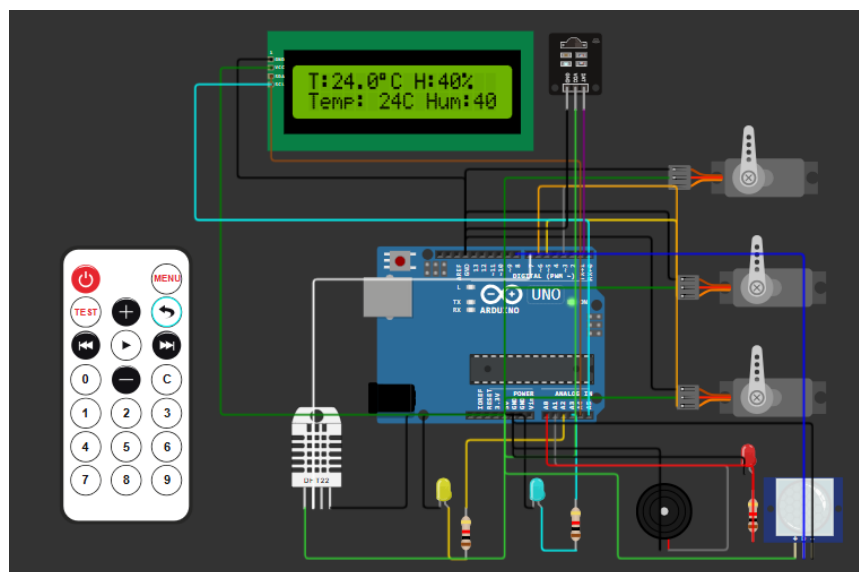


Рисунок 2.18 – Manual: RESET

## 9. Повне вимкнення системи:

- Натискання кнопки "POWER" (код 0xA2) вимикає освітлення, опалення та закриває всі двері, вікна та ворота. На дисплеї з'являється "All OFF & CLOSED" (рис. 2.19).

```

case 0xA2:
    lightOn = false;
    heatingOn = false;
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(BLUE_LED, LOW);
    doorServo.write(0);
    windowServo.write(0);
    gateServo.write(0);
    lcd.print("All OFF & CLOSED ");
    break;

```

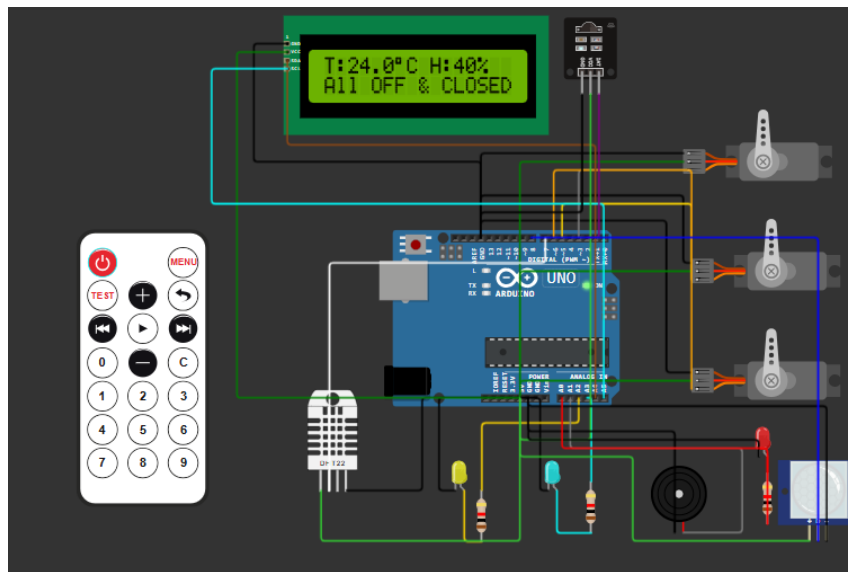


Рисунок 2.19 – All OFF & CLOSED

## 10. Охоронна система:

- Натискання кнопки "MENU" (код 0xE2) активує охоронний режим. На дисплеї відображається "Alarm: ARMED" (рис. 2.20).

```

case 0xE2:
    Serial.println("Manual alarm activated!");
    triggerAlarm();
    break;

```

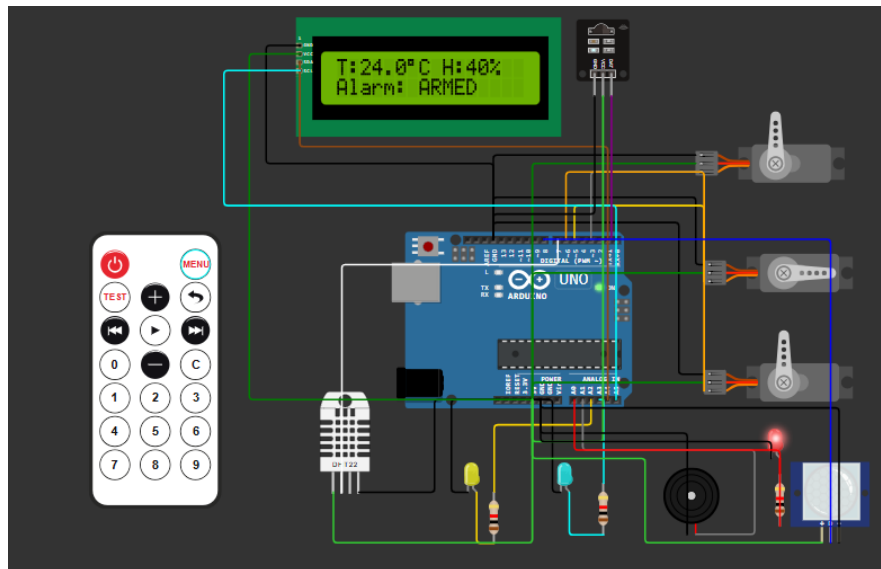


Рисунок 2.20 – Alarm: ARMED

- Якщо у цьому режимі датчик руху PIR фіксує активність, система переходить у режим тривоги:
  - Всі двері, вікна та ворота зачиняються автоматично.
  - Вмикається звукова сигналізація (бuzzer).
  - Червоний світлодіод починає миготіти кожні 300 мс.
  - На дисплеї з'являється "ALARM TRIGGERED" (рис. 2.21).

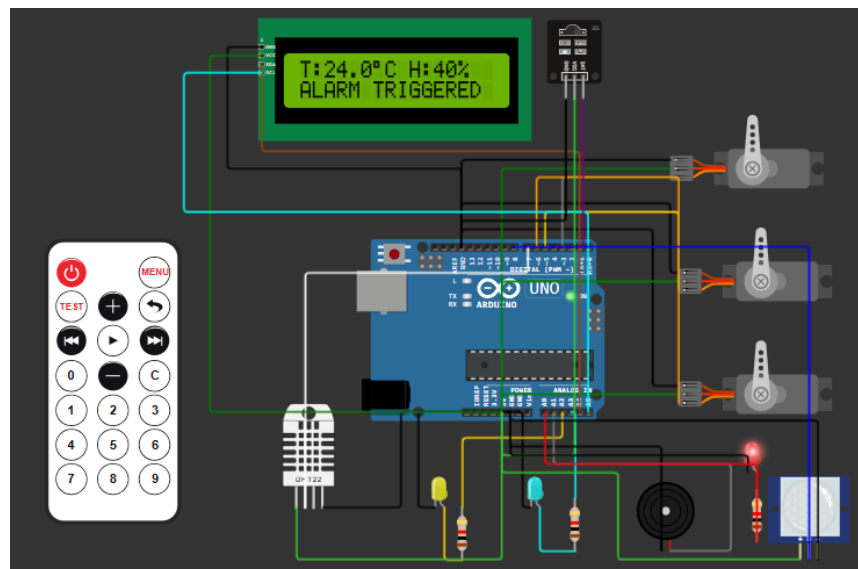


Рисунок 2.21 – Alarm: TRIGGERED

- Кнопка "TEST" (код 0x22) вимикає режим охорони, сигналізація та світлодіоди вимикаються, а на дисплеї з'являється "Alarm: OFF" (рис. 2.22).

```

case 0x22:
    alarmArmed = false;
    alarmTriggered = false;
    digitalWrite(BUZZER, LOW);
    digitalWrite(RED_LED, LOW);
    lcd.print("Alarm: OFF      ");
    break;

```

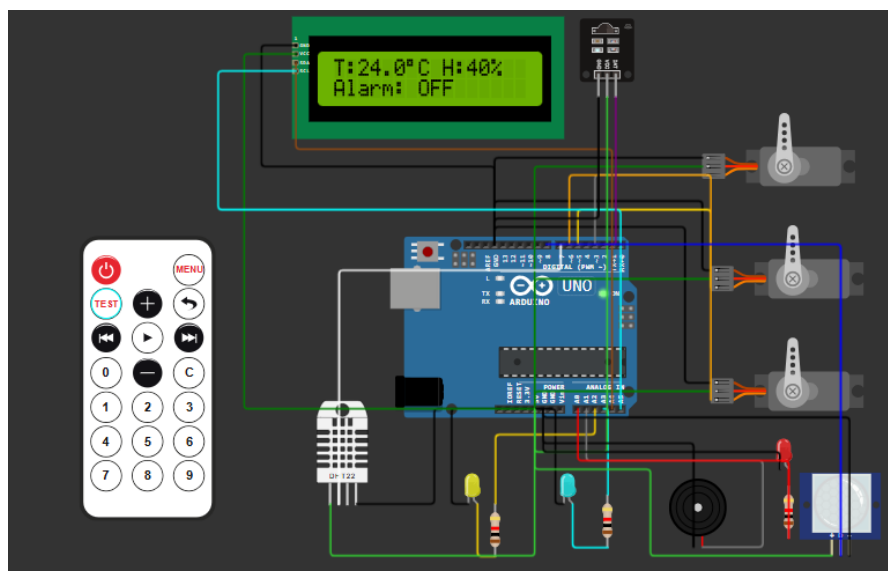


Рисунок 2.22 – Alarm: OFF

## 2.5 Висновки до розділу

У другому розділі дипломної роботи було детально розглянуто процес розробки системи “Розумний дім” на основі мікроконтролера Arduino Uno. На основі сформованих вимог до функціональності системи було визначено її загальну концепцію та архітектуру. Система складається з декількох основних підсистем: освітлення, контролю температури й вологості, безпеки (датчики руху та відкриття

дверей), а також дистанційного керування за допомогою мобільного додатку або веб-інтерфейсу.

Були описані функціональні можливості, які реалізуються завдяки взаємодії між апаратними модулями — такими як датчики температури та вологості DHT11/DHT22, PIR-датчики руху, реле, світлодіоди, модулі Bluetooth або Wi-Fi (наприклад, ESP8266), тощо. Кожен з компонентів відіграє важливу роль у забезпеченні автономності, інтелектуальності та адаптивності системи до умов навколишнього середовища.

У підрозділі, присвяченому схемі підключення компонентів, розглянуто логічну структуру зв'язків між мікроконтролером і периферійними пристроями. Це дозволило забезпечити надійну передачу даних, живлення та ефективну реакцію системи на зовнішні події.

Особливу увагу було приділено програмній реалізації системи. Для написання логіки керування використовувалась мова програмування C++ у середовищі Arduino IDE. У коді реалізовані функції зчитування показників сенсорів, опрацювання умов керування виконавчими пристроями та обмін даними з користувачем. Програма структурована таким чином, щоб забезпечити гнучкість, масштабованість та зручність для майбутньої модернізації.

У підсумку можна зробити висновок, що створена система відповідає поставленим вимогам, демонструє стабільну роботу в різних умовах та може бути розширена для реалізації додаткових функцій. Розроблена архітектура і логіка системи підтверджують доцільність використання мікроконтролера Arduino Uno для побудови бюджетного та функціонального “розумного будинку”.

## ВИСНОВКИ

У процесі написання та реалізації дипломного проєкту було виконано повний цикл розробки прототипу системи автоматизації побуту — “Розумного будинку” — з використанням мікроконтролера Arduino Uno. Результати роботи доводять, що на базі відносно недорогих апаратних засобів та відкритих програмних інструментів можливо створити ефективну, масштабовану та зручну в користуванні систему керування домашнім середовищем.

На етапі теоретичного дослідження було визначено ключові поняття, що лежать в основі концепції “розумного будинку”, серед яких: автоматизація побутових процесів, дистанційне керування пристроями, оптимізація витрат енергії, підвищення рівня безпеки та зручності. Було здійснено класифікацію існуючих систем, що дало змогу краще зрозуміти їхні архітектурні особливості, принципи роботи та переваги/недоліки. Також було проведено огляд найпоширеніших електронних компонентів, які використовуються в таких системах: датчиків температури, вологості, руху, освітленості, а також засобів зв'язку (модулі Wi-Fi, Bluetooth).

У практичному розділі реалізовано власний варіант системи “розумного будинку” із модульною архітектурою, що дозволяє додавати або вилучати окремі блоки без повного перепроєктування системи. У якості обчислювального ядра використано мікроконтролер Arduino Uno, до якого підключено датчики температури та вологості (DHT11), руху (PIR), освітленості (фоторезистор), реле для керування освітленням та іншими приладами. Всі компоненти підключено відповідно до розробленої схеми, яка забезпечує їх сумісну й стабільну роботу.

Програмна реалізація забезпечує зчитування даних із сенсорів, обробку отриманої інформації та прийняття відповідних рішень згідно з заданою логікою — наприклад, автоматичне вмикання світла при виявленні руху або регулювання мікроклімату залежно від температури.

Система має низку переваг, серед яких: простота встановлення, масштабованість, економічність, гнучкість у налаштуванні та можливість

подальшого розширення. До недоліків можна віднести обмежену обчислювальну потужність Arduino Uno, що частково компенсується оптимізацією програмного коду та можливістю використання потужніших контролерів у майбутньому. Система є хорошою базою для продовження з вдосконаленням з інтеграцією керування через Інтернет.

Таким чином, у рамках дипломної роботи було не лише проаналізовано сучасні тенденції в галузі автоматизації побуту, але й спроектовано й реалізовано повноцінну систему “розумного дому”, що демонструє високий потенціал впровадження подібних технологій у повсякденне життя. Отримані результати підтверджують ефективність і доцільність використання мікроконтролерів у системах домашньої автоматизації.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. «Що таке Розумний будинок і для чого він потрібен». URL: <https://alterair.ua/stati/chto-takoe-umnyy-dom/> (дата звернення: 25.04.2025).
2. Розумний дім. URL: [https://uk.wikipedia.org/wiki/Розумний\\_дім](https://uk.wikipedia.org/wiki/Розумний_дім) (дата звернення: 25.04.2025).
3. Системи розумного будинку — для безпечного та комфортного простору. URL: <https://www.iqdim.ua/post/systemu-rozumnogo-budunku> (дата звернення: 25.04.2025).
4. Класифікація пристроїв та систем автоматички. URL: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/Електрона%20Книга%20Автоматизація%20технологічних%20процесів%20та%20виробництв/lek1/lek2.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/Електрона%20Книга%20Автоматизація%20технологічних%20процесів%20та%20виробництв/lek1/lek2.html) (дата звернення: 25.04.2025).
5. Розподілена система керування. URL: [https://uk.wikipedia.org/wiki/Розподілена\\_система\\_керування](https://uk.wikipedia.org/wiki/Розподілена_система_керування) (дата звернення: 25.04.2025).
6. “Розумний клімат контроль.” ConnectHome. URL: <https://c-home.ua/uk/%D0%B1%D0%B5%D0%B7-%D1%80%D1%83%D0%B1%D1%80%D0%B8%D0%BA%D0%B8/umnyj-klimat-kontrol/> (дата звернення: 25.04.2025).
7. “Бездротові стандарти. Що таке ZigBee?”. URL: <https://www.sea.com.ua/ua/besprovodnye-komponenty/news/bezdrotovi-standarti-so-take-zigbee/> (дата звернення: 25.04.2025).
8. “Alexa Smart Home – Learn about Home Automation.” Amazon.com. URL: <https://www.amazon.com/alexa-smart-home/b?ie=UTF8&node=21442899011> (дата звернення: 25.04.2025).
9. Control smart home devices with Google Assistant. URL: <https://support.google.com/assistant/answer/7314909?co=GENIE.Platform%3DAndroid&hl=en> (дата звернення: 25.04.2025).
10. Home app. URL: <https://www.apple.com/home-app/> (дата звернення: 25.04.2025).

## ДОДАТОК

```

#include <Servo.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#include <IRremote.h>

#define IR_RECEIVE_PIN 2
#define RED_LED A0
#define YELLOW_LED A2
#define BLUE_LED A3
#define BUZZER A1
#define DHTPIN 7
#define DHTTYPE DHT22
#define PIR_SENSOR_PIN 4

Servo doorServo, windowServo, gateServo;
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

float temperature = 22.0;
float humidity = 50.0;
bool manualTempSet = false;
bool manualHumiditySet = false;
bool heatingOn = false;
bool lightOn = false;
bool alarmArmed = false;
bool alarmTriggered = false;
bool pirState = LOW;

unsigned long lastTempUpdate = 0;
const unsigned long tempInterval = 2000;
unsigned long lastBlinkTime = 0;
const unsigned long blinkInterval = 300;
bool ledBlinkState = false;

void setup() {
  Serial.begin(9600);
  IrReceiver.begin(IR_RECEIVE_PIN);
  dht.begin();
  lcd.init();
  lcd.backlight();

  doorServo.attach(3);
  windowServo.attach(5);
  gateServo.attach(6);
  doorServo.write(0);
  windowServo.write(0);
  gateServo.write(0);

  pinMode(RED_LED, OUTPUT);

```

```

pinMode(YELLOW_LED, OUTPUT);
pinMode(BLUE_LED, OUTPUT);
pinMode(BUZZER, OUTPUT);
pinMode(PIR_SENSOR_PIN, INPUT);

digitalWrite(REDA_LED, LOW);
digitalWrite(YELLOW_LED, LOW);
digitalWrite(BLUE_LED, LOW);
digitalWrite(BUZZER, LOW);

lcd.setCursor(0, 0);
lcd.print(" Smart Home ");
delay(3000);
lcd.clear();
}

void updateTemperatureDisplay() {
  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(temperature, 1);
  lcd.print((char)223);
  lcd.print("C H:");
  lcd.print(humidity, 0);
  lcd.print("% ");
}

void updateHeatingStatus() {
  if (heatingOn) {
    lcd.setCursor(0, 1);
    lcd.print("Heating: ON ");
    digitalWrite(BLUE_LED, HIGH);
  } else {
    lcd.setCursor(0, 1);
    lcd.print("Heating: OFF ");
    digitalWrite(BLUE_LED, LOW);
  }
}

void automaticHeatingControl() {
  if (temperature < 18 && !heatingOn) {
    heatingOn = true;
    updateHeatingStatus();
  }
  if (temperature > 24 && heatingOn) {
    heatingOn = false;
    updateHeatingStatus();
  }
}

void loop() {
  unsigned long currentMillis = millis();

```

```

if (currentMillis - lastTempUpdate >= tempInterval) {
  float realTemp = dht.readTemperature();
  float realHumidity = dht.readHumidity();

  if (!isnan(realTemp) && !manualTempSet) temperature = realTemp;
  if (!isnan(realHumidity) && !manualHumiditySet) humidity = realHumidity;

  updateTemperatureDisplay();
  lastTempUpdate = currentMillis;
}

automaticHeatingControl();

if (IrReceiver.decode()) {
  int key = IrReceiver.decodedIRData.command;
  Serial.print("IR code: ");
  Serial.println(key, HEX);
  handleIR(key);
  IrReceiver.resume();
}

int pirValue = digitalRead(PIR_SENSOR_PIN);

if (pirValue == HIGH && pirState == LOW) {
  pirState = HIGH;
  Serial.println("Motion detected!");
  triggerAlarm();
}

if (pirValue == LOW && pirState == HIGH) {
  pirState = LOW;
  Serial.println("Motion ended!");
}

if (alarmTriggered && currentMillis - lastBlinkTime >= blinkInterval) {
  ledBlinkState = !ledBlinkState;
  digitalWrite(RED_LED, ledBlinkState);
  lastBlinkTime = currentMillis;
}

if (!alarmArmed && !alarmTriggered) {
  digitalWrite(RED_LED, LOW);
}
}

void handleIR(int key) {
  lcd.setCursor(0, 1);
  switch (key) {
    case 0x30: doorServo.write(90); lcd.print("Door: OPEN      "); break;
    case 0x18: doorServo.write(0); lcd.print("Door: CLOSED    "); break;
  }
}

```

```

case 0x7A: windowServo.write(90); lcd.print("Window: OPEN    "); break;
case 0x10: windowServo.write(0); lcd.print("Window: CLOSED  "); break;
case 0x38: gateServo.write(90); lcd.print("Gate: OPEN      "); break;
case 0x5A: gateServo.write(0); lcd.print("Gate: CLOSED   "); break;
case 0x42: lightOn = true; digitalWrite(YELLOW_LED, HIGH); lcd.print("Light:
ON      "); break;
case 0x4A: lightOn = false; digitalWrite(YELLOW_LED, LOW); lcd.print("Light:
OFF     "); break;
case 0x52: heatingOn = true; updateHeatingStatus(); break;
case 0x68: heatingOn = false; updateHeatingStatus(); break;
case 0x02:
    temperature += 1; manualTempSet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Temp increased  ");
    break;
case 0x98:
    temperature -= 1; manualTempSet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Temp decreased  ");
    break;
case 0x90:
    if (humidity < 100) humidity += 1;
    manualHumiditySet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Humidity up    ");
    break;
case 0xE0:
    if (humidity > 0) humidity -= 1;
    manualHumiditySet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Humidity down  ");
    break;
case 0xA2:
    lightOn = false;
    heatingOn = false;
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(BLUE_LED, LOW);
    doorServo.write(0);
    windowServo.write(0);
    gateServo.write(0);
    lcd.print("All OFF & CLOSED ");
    break;
case 0xE2:
    Serial.println("Manual alarm activated!");
    triggerAlarm();
    break;
    break;

```

```

case 0x22:
    alarmArmed = false;
    alarmTriggered = false;
    digitalWrite(BUZZER, LOW);
    digitalWrite(RED_LED, LOW);
    lcd.print("Alarm: OFF      ");
    break;
case 0xC2:
    temperature = 24.0;
    humidity = 40.0;
    manualTempSet = true;
    manualHumiditySet = true;
    updateTemperatureDisplay();
    lcd.setCursor(0, 1);
    lcd.print("Temp: 24C Hum:40%");
    break;
default:
    lcd.print("Unknown command ");
    break;
}
}

void triggerAlarm() {
    alarmTriggered = true;
    digitalWrite(BUZZER, HIGH);
    doorServo.write(0);
    windowServo.write(0);
    gateServo.write(0);
    lcd.setCursor(0, 1);
    lcd.print("!!! ALARM !!!  ");
}

```