

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ
ДИСТАНЦІЙНОГО УПРАВЛІННЯ
КОНДИЦІОНЕРОМ»

Виконав(ла): студент(ка) 2 курсу, групи 8.1219

спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

Д.В. Кучерова

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент к.т.н., Мухін В.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної
математики, доцент, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Математичний

Кафедра програмної інженерії

Рівень вищої освіти Магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

« »

2020 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ(СТУДЕНТЦІ)

Кучеровій Діані Віталіївні

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка мобільного додатку

дистанційного управління кондиціонером

керівник роботи (проекту) Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затвержені наказом ЗНУ від « 20 » 05 2020 року № 576-с

2. Строк подання студентом роботи 30.11.2020

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі.

2. Огляд технологій.

3. Проектування додатку.

4. Реалізація додатку.

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

Презентація

6. Консультанти розділів роботи

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

7. Дата видачі завдання 20.05.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	28.05.2020	Виконано
2.	Збір вихідних даних.	15.06.2020	Виконано
3.	Обробка методичних та теоретичних джерел.	15.08.2020	Виконано
4.	Розробка першого і другого розділу.	16.10.2020	Виконано
5.	Розробка третього розділу.	07.11.2020	Виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи.	30.11.2020	Виконано
7.	Захист кваліфікаційної роботи.	16.12.2020	Виконано

Студент _____
(підпис)

Д.В. Кучерова
(ініціали та прізвище)

Керівник роботи _____
(підпис)

В.В. Мухін
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.В. Кудін
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка мобільного додатку дистанційного управління кондиціонером»: 46 с., 12 рис., 29 джерел, 3 додатки.

АНДРОЇД, АРДУІНО, АВТОМАТИЗОВАНІ СИСТЕМИ, РОЗУМНИЙ ДІМ, ІНТЕРНЕТ РЕЧЕЙ, МОБІЛЬНИЙ ДОДАТОК, WI-FI.

Об'єкт дослідження – процес розробки мобільного додатку; взаємодія мобільного додатку та побутового приладу (кондиціонеру).

Мета роботи: дослідження технологій віддаленого керування та взаємодії мобільного додатку з побутовими приладами для створення системи віддаленого керування.

Методи дослідження – об'єктно орієнтовний аналіз, аналітичний, описовий.

У роботі проведено аналіз засобів віддаленого керування, існуючих на ринку систем розумного дому, що мають функціонал, що подібний до розроблюваного додатку та середовищ розробки подібних додатків. Представлено реалізацію мобільного застосунку.

Результатом роботи є робочий додаток, що знаходиться в експериментальній експлуатації, що розроблений для домашнього використання.

SUMMARY

Master's Qualification Thesis "Development of a Mobile Application for Air Conditioning Remote Controlling": 46 pages, 12 figures, 29 sources, 3 applications.

ANDROID, ARDUINO, AUTOMATED SYSTEMS, SMART HOME, INTERNET OF THINGS, MOBILE APP, WI-FI.

The object of the study is the process of mobile application development; interaction of mobile application and household appliance (air conditioner).

The aim of the study is research of remote control technologies and interaction of mobile application with household appliances to create a remote control system.

The methods of research are object-oriented analysis, analytical, descriptive. The paper analyzes the remote control tools available on the market of smart home systems that have functionality similar to the developed application and development environments for such applications. The implementation of the mobile application is presented.

The purpose of the work is to create a mobile application and implement its interaction with a household appliance on the example of an air conditioner, which will provide remote control and monitoring.

The system provides timely information on the use of household appliances, the possibility of remote temperature control. Technical solution: to interact with the mobile application, the air conditioner is equipped with a NodeMcu board based on the ESP8266 module for transmitting a signal to a local network or the Internet via Wi-Fi. Board programming in Arduino IDE. Create a remote control application for Android smartphones using APP Inventor.

The result is a working application that is in experimental operation

ЗМІСТ

Завдання	2
Реферат	4
Summary	5
Вступ.....	7
1 Огляд технологій.....	9
1.1 Системи «розумний дім»	9
1.2 Існуючі готові рішення	11
1.3 Опис обладнання.....	13
1.4 Середовища розробки мобільних додатків.....	18
1.5 Середовище розробки прошивок для плат	21
1.6 Висновки до першого розділу	22
2 Проектування системи.....	23
2.1 Вступ до проектування	23
2.2 Вимоги до системи	23
2.3 Висновки другого розділу	24
3 Реалізація	25
3.1. Програмування плати NodeMCU	25
3.2 Створення мобільного додатку.....	33
3.3. Підключення плати	35
4 Висновки.....	36
Перелік посилань.....	38
Додаток А.....	41
Додаток Б	42
Додаток В.....	46

ВСТУП

Ще з давніх часів людство намагалось покращити ефективність своєї роботи та побуту. Крізь століття змінювались знаряддя та способи організації праці, поки індустріальна революція не подарувала людям усі переваги використання машин у виробництві. Сучасні ж роботизовані фабрики потребують лише контролю з боку персоналу. Не дивно, що у динамічному сьогоденні людині хочеться спростити і свій побут також, а автоматизовані системи чудово для цього підходять.

Ми живемо світі коли те, що здавалось фантастикою лише сотню років тому, стає все більш звичним. Життя сучасної людини тепер не обмежується лише домом, а інколи навіть містом і континентом. Постають проблеми безпеки, наприклад, охорони будинку під час відсутності господарів, своєчасної перевірки комунікацій, проблеми економії електроенергії, води та газу, підтримання кліматичних умов тощо.

Автоматизовані системи вирішують ці різноманітні задачі і мають чимало переваг. Вони полегшують ведення господарства, допомагають персонам з особливими потребами, контролюють інженерні системи, охороняють дім та багато іншого, що раніше робили лише спеціально найняті працівники. Використання автоматизованих систем економить також багато часу, що є найціннішим ресурсом. Ринок надає чимало готових рішень, проте попри всі переваги вони мають суттєві недоліки: системи є досить дорогими, є проблема інтегрування існуючої техніки в систему, не усі рішення можуть бути персоналізованими, а технології досить швидко застарівають.

Ціль роботи – створення мобільного додатку та реалізація його взаємодії з побутовим приладом на прикладі кондиціонера, що забезпечить віддалене керування та контроль.

Для досягнення мети роботи сформульовано наступні завдання:

- 1) аналіз технологій – опис існуючих підходів та готових рішень.

- 2) проектування додатку – аналіз вимог та створення діаграм майбутнього продукту;
- 3) розробка компоненту;
- 4) перелік рекомендацій з використання додатку.

Система забезпечує своєчасне отримання інформації про використання побутового приладу, можливість дистанційного керування температурою.

Технічне рішення: для взаємодії з мобільним додатком кондиціонер оснащено платою NodeMcu на основі модуля ESP8266 для передачі сигналу в локальну мережу або інтернет через Wi-Fi. Програмування плати в Arduino IDE. Створення програми віддаленого керування для смартфонів на базі Android за допомогою APP Inventor.

Система знаходиться в експериментальній експлуатації.

1 ОГЛЯД ТЕХНОЛОГІЙ

1.1 Системи «розумний дім»

Інтернет речей (з англ. Internet of things, IoT) – це концепція обчислювальної мережі фізичних предметів («речей»), що оснащені вбудованими технологіями для взаємодії один з одним або з зовнішнім середовищем [1], яка розглядає організацію таких мереж як явище, здатне перебудувати економічні та суспільні процеси, що виключає з частини дій і операцій необхідність участі людини [2].

Концепція була сформульована в 1999 році як осмислення перспектив широкого застосування засобів радіочастотної ідентифікації для взаємодії фізичних предметів між собою та з зовнішнім оточенням. Починаючи з минулого десятиріччя наповнення концепції різноманітним технологічним змістом і впровадження практичних рішень стало стійкою тенденцією в інформаційних технологіях [3]. Галузь швидко розвивається завдяки розповсюдженню бездротових мереж, появи хмарних обчислень, розвитку технологій межмашинної взаємодії, початку активного переходу на IPv6 і освоєння програмно-визначених мереж.

Домашня автоматизація розглядається як окремий випадок інтернету речей. Вона охоплює доступні через інтернет домашні пристрої, в той час як інтернет речей включає будь-які пов'язані через інтернет пристрої взагалі.

Домашня автоматизація (англ. Home automation), або розумний будинок (англ. Smart home) - система домашніх пристроїв, що здатні виконувати дії і вирішувати певні повсякденні завдання без участі людини. Домашня автоматизація сьогодні - надзвичайно гнучка сучасна система, яку користувач конструює і налаштовує самостійно відповідно до власних потреб. Тобто власник розумного будинку самостійно визначає, які пристрої куди встановити і які завдання вони будуть виконувати [4, 6].

Система розумного будинку включає такі типи пристроїв:

- 1) Контролер (хаб) — керуючий пристрій, який з'єднує всі елементи системи між собою і зв'язує систему з зовнішнім світом.
- 2) Датчики (сенсори) — пристрої для отримання інформації про зовнішні умови.
- 3) Актуатори — пристрої, що безпосередньо виконують команди. В цю групу входять розумні (автоматичні) вимикачі, розумні (автоматичні) розетки, розумні (автоматичні) клапани для труб, сирени, клімат-контролери і так далі.

Автоматизувати можна майже будь-яку побутову функцію. Найбільш поширені приклади автоматичних дій в «розумному будинку»: автоматичне включення і вимкнення світла, автоматична корекція роботи опалювальної системи або кондиціонера, автоматичні повідомлення про вторгнення, спалах або витік води, помилки в роботі приладів.

В даний час використовуються дві системи для дистанційного керування розумним домом:

- 1) Через інтернет-шлюз. Наявність Wi-Fi роутера і мережі Інтернет в цьому випадку обов'язкова.
- 2) З використанням GSM модуля управління. Потрібен спеціальний GSM модуль з сім-картою оператора стільникового зв'язку. GSM може також використовуватись також як резервний канал. GSM-модуль - це пристрій, що працює в якості мобільного телефону або модему я використовується для зв'язку будь-якого пристрою через мережу. GSM-модуля для роботи потрібна SIM-карта і він працює в діапазоні мережі, яку підтримує він сам і мобільний оператор. Залежно від виробника і призначення, передача команд управління на GSM-модуль може виконуватися такими способами:
 - телефонний виклик (дозвон) Приймавши дзвінок і отримавши номер абонента, система порівнює його з номерами з внутрішнього списку. При збігу з одним з номерів вона передає сигнал на блок управління. Дзвінок відповідає лише одній конкретній команді;

- SMS технологія прийому та передачі коротких текстових повідомлень. Можна передати безліч команд;
- мобільні додатки, для зручного та багатофункціонального керування.

GSM-модуль складається з радіоблоків, процесора, пам'яті і інтерфейсів для інтеграції з периферійним обладнанням. Залежно від моделі, GSM-модулі, крім голосового зв'язку і SMS, можуть підтримувати GPRS-передачу даних [8].

1.2 Існуючі готові рішення

Станом на 2020 рік галузь стрімко розвивається, тому існує чимало готових рішень для розумного будинку в цілому та віддаленого керування конкретними побутовими приладами. Їх спільними недоліками є ціна обладнання, неможливість або складність налаштування під індивідуальні потреби, етичні питання (наприклад, якщо користувач не підтримує виробництво Росії). Далі надано основну інформацію про існуючі готові рішення.

1) Qivicon - це базова система компанії Telekom для розумного будинку. Для підключення в домашню мережу потрібен пристрій Home Base вартістю близько \$170. Він підключається до роутера і забезпечує зв'язок між компонентами розумного будинку. Управління та налагодження проводиться через веб-інтерфейс (браузер на ПК) або через мобільний додаток. Додатки для управління відповідними пристроями входять в стартовий пакет. До пакета EnBW надається програма EnBW, при придбанні набору Telekom надається додаток Smart Home. Можливість скачати утиліти не передбачена, для активації додатків потрібні коди, які надаються тільки разом з пакетами. Тобто, концепція відкритого програмного забезпечення на ці програми не поширюється. EnBW пропонує безкоштовне користування додатком протягом

року, після цього підписка € 50 на рік. Telekom надає два роки безкоштовної підписки, далі коштує € 30 на рік. У підсумку всі заощаджені на енергії гроші будуть йти на оплату ліцензій програм [9].

2) Homee: управління розумним будинком без хмари. Homee, як і Qivicon, комплексною системою для управління будинком. Система може контролювати окремі опалювальні прилади, стежить за дверима і вікнами, рухати жалюзі і стежити за приміщеннями за допомогою датчиків руху, через датчики рівня вологості стежити за підвалом і гаражем. Посередником між технологіями є модуль Brain Cube, що через WLAN надає підключення до смартфона або до роутера. Комплект з Brain Cube з кубиком для розширення буде коштувати \$200 [10].

3) HomeMatic. Найдешевша і об'ємна система з власним (і самостійно розробленим) протоколом бездротового зв'язку. Програмування та включення в мережу компонентів дещо складні для сьогоднішніх умов. Цікавою особливістю HomeMatic є можливість комбінування бездротового і дротового (RS485) зв'язку всіх компонентів. Це дає можливість автоматизувати об'єкти щоскладаються з декількох віддалених будинків і приміщень або багатоповерхові будівлі, де віддалені будівлі або поверхи з'єднуються між собою за допомогою провідної лінії. У середині ж кожного об'єкта в радіусі 100-300 метрів комунікація всіх компонентів «розумного будинку» здійснюється за бездротовою технологією [11].

4) RWE Smart Home. В цій системі робиться ставка на простоту установки і управління за порівняно невеликими цінами [12].

5) Netatmo – французька компанія, що надає безліч рішень для розумного дому [13].

Оскільки тема роботи передбачує створення програми для взаємодії з кондиціонером, далі буде розглянуто готові рішення для віддаленого керування саме кондиціонерами.

1) Xiaomi Aqara/Mi Home, Broadlink RM Pro/Mini
Універсальні ІК-передавачі, можуть застосовуватись майже з будь-

якими кондиціонерами. Цінник доступний (\$ 20-35 за кімнату, в залежності від можливостей). В деяких випадках додатки китайською.[14]

- 2) Sensibo. Система управління кондиціонером складається з двох частин: модуля Sensibo і комутатора, який з'єднує девайс з Інтернетом (через Wi-Fi роутер) та забезпечує зв'язок відразу з декількома кондиціонерами в приміщенні за умови що для кожного кондиціонера куплений свій модуль (є різні комплекти з різною кількістю керуючих модулів). Вартість пристрою складе від \$99 доларів. Кіт для девелопера (1 модуль, 1 комутатор і API) коштує \$199. [15]
- 3) Tado Smart AC Control V3+, за ціною 90 фунтів: замінює існуючий пульт дистанційного керування та підключається до Wi-Fi мережі - без додаткової дротів. Для керування декількома пристроями за допомогою tado потрібен tado Smart Control Conditioning Control для кожного з них. [16]
- 4) СРК-М і СРК-М-01 може працювати з кондиціонерами будь-яких виробників. Різниця тільки в способі підключення. [18]
- 5) CoolPlug - це апаратний пристрій, який який разом з CoolLinkHub забезпечує інтеграцію систем спліт, мульти-спліт або міні-спліт HVAC із системами домашньої автоматизації, системами управління будинками та хмарними програмами. CoolPlug і CoolLinkHub спільно працюють над створенням універсального каналу зв'язку, що дозволяє розділеним системам вентиляції та кондиціонування працювати з широким спектром систем автоматизації.[20]

1.3 Опис обладнання

NodeMCU (Рисунок 1.1) - це плата на основі модуля ESP8266. Плата призначена для зручного управління різними схемами на відстані. При цьому

підключення до мережі здійснюється до домашньої / офісної локальної мережі з виходом в інтернет через роутер. Власник пристрою може контролювати його за допомогою планшета або комп'ютера через свою локальну мережу або віддалено, через Інтернет. Найбільше місця на платі займає чип ESP8266, на якому заставлений мікропроцесор з тактовою частотою 80 МГц (можна розігнати до 160 МГц).[15]

Використовується для створення облаштувань Інтернету речей, систем розумного будинку або роботів, керованих на відстані.

Технічні параметри

- напруга живлення : 5 В (від USB);
- максимальний струм: 220 мА ;
- модуль ESP8266: ESP - 12e ;
- підтримка Wi - Fi протоколів : 802.11 b / g / n;
- підтримуються режими: WiFi Direct (P2P), soft - AP (точка доступу);
- антена: PCB- антена 3dbi;
- вбудований стек: TCP / IP;
- габарити: 49 мм x 24.5 мм x 13 мм;
- ціна: 120 грн.

D0 (GPIO16), D1 (GPIO5), D2 (GPIO4), D3 (GPIO0), D4 (GPIO2), D5 (GPIO14), D6 (GPIO15), D7 (GPIO13), D8 (GPIO15) – виводи для підключення зовнішніх пристроїв.

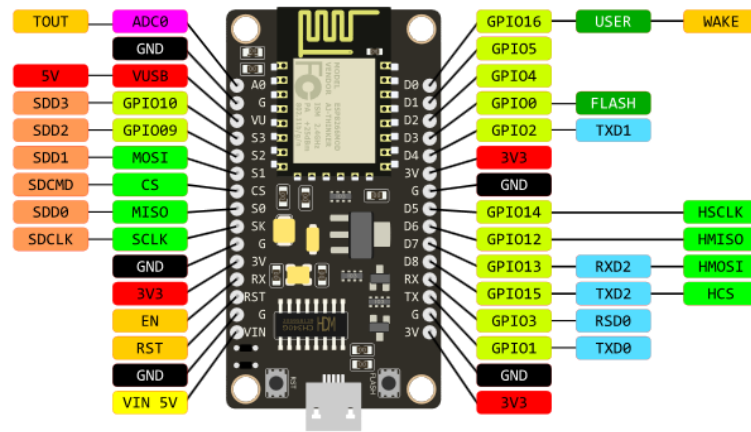


Рисунок 1.1 – Плата NodeMCU з розпіновкою

DHT22 (Рисунок 1.2) – датчик температури і вологості, що може бути використаний в багатьох пристроях, в тому числі на Arduino. Передача даних здійснюється по одному дроту з використанням власного протоколу. Для роботи з Arduino існують готові бібліотеки. Перевагою даних датчиків є невеликі розміри, низьке енергоспоживання, висока дальність передачі, з невеликою затримкою в передачі показань в 2 секунди [29].

Характеристики:

- напруга живлення: 3.5 ... 5.5 В;
- діапазон температур: -40 ... + 80 ° С;
- похибка показань температури: ± 0.5 ° С;
- діапазон вологості: 0 - 100%;
- похибка показань вологості: $\pm 2\%$;
- ціна: 99грн.

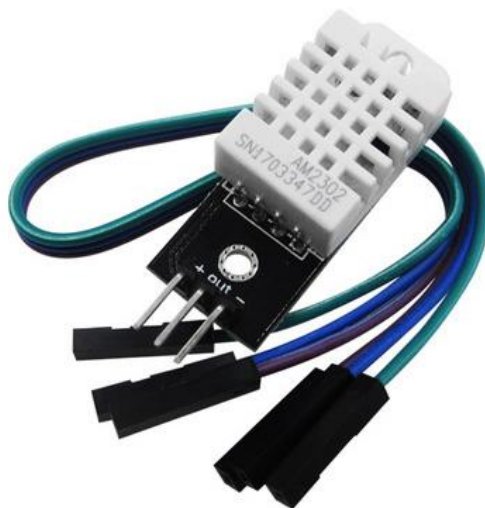


Рисунок 1.2 Датчик DHT22

Електромагнітне реле - це електричний пристрій, що механічним шляхом замикає або розмикає ланцюг навантаження за допомогою магніта, складається з електромагніту, рухливого якоря і перемикача. Електромагніт - це дріт, який намотаний на котушку з феромагнетика. В ролі якоря виступає пластина з магнітного матеріалу. У деякі моделі пристрою можуть бути вбудовані додаткові електронні компоненти: резистор для точнішого спрацьовування реле, конденсатор для зменшення перешкод, діод для усунення перенапружень.

Модуль 4-х каналного твердотільного реле побудований на G3MB-202P і призначений для комутації змінної напруги. Особливістю твердотільних реле є відсутність рухомих контактів і контактів взагалі. Цим обумовлена їх надійність і безшумність роботи. Для управління реле досить подати на нього цифровий сигнал з мікроконтролера – струмообмежувальний резистор знаходиться всередині самого реле.

Опис контактів: DC + позитивне живлення, DC-: негативне живлення, CN: вхід управління.

Характеристики:

- твердотільний модуль: G3MB-202P;
- кількість каналів: 4;
- напруга, що комутується: від 100 до 240 В;

- комутований струм: макс. 2А;
- напруга управління: від 0 до 5В;
- рівень сигналу управління: високий (high);
- максимальний струм управління: 12.5 мА;
- робочий струм управління: 2 мА Розміри: 57 * 55 * 25 мм;
- ціна: 170 грн.[28]

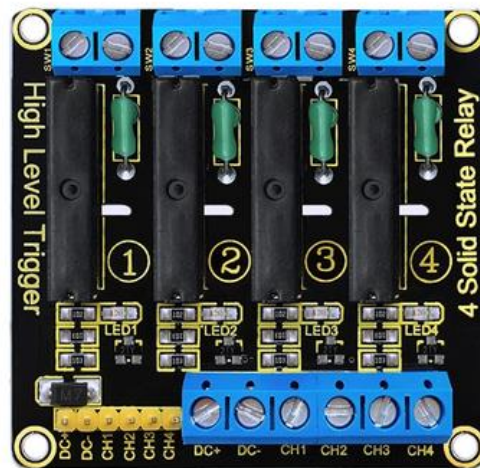


Рисунок 1.3 Чотириканальне твердотільне реле

Датчик наявності змінної напруги використовується якщо потрібно контролювати наявність змінної напруги в побутовій мережі 220 В. Для використання датчика потрібно спочатку підключити до клем затискачів, позначених на платі VCC і GND, напруги 5 В постійного струму. Далі до клем затискачів, позначених на платі L і N, потрібно підключити провідники, які будуть підключатися в побутову мережу 220 В. До клемі затиску, яка позначена на платі OUT, потрібно підключити цифровий вхід мікроконтролера, потім підключити в побутову мережу 220 В. Після підключення на платі датчика почне горіти зелений світлодіод, позначений LED. При наявності напруги 220 В світлодіод LED горить, якщо така напруга відсутня - світлодіод LED горіти не буде.

Характеристики:

- вхідна напруга: 220 В змінного струму;
- напруга живлення: 5 В постійного струму;
- вихідний сигнал: 0 - 0,5 В постійного струму;
- розміри: 72 x 15 x 16 мм;
- ціна: 115грн [27].



Рисунок 1.4 Датчик змінної напруги

DuPont – сполучні дроти для підключення датчиків до плати. Ціна: 50грн/120 шт., для розробки вистачить 20 шт.



Рисунок 1.5 Сполучні дроти

Загальна ціна обладнання за цінами і курсом валют на момент написання роботи складає 515грн (\$18,5), що близько до рішень Xiaomi, але значно дешевше за рішення інших виробників.

1.4 Середовища розробки мобільних додатків

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework. [21]

Eclipse (з англ. - «затемнення») - вільна інтегрована среда розробки модульних кроссплатформених додатків. Розвивається та підтримується Eclipse Foundation. Використовує багато системних ресурсів і процесорів, повільно запускається. Хоча базові речі використовувати легко, пошук деяких просунутих функцій спершу викликає утруднення, якщо ви не звикли до того, як працює Eclipse. Часто речі ховаються за прихованими настройками або всередині відключеною панелі. Зустрічаються труднощі з зворотною сумісністю. [22]

Android Studio — інтегроване середовище розробки для платформи Android з відкритим кодом. [23]. Недоліком є необхідність наявності 3 GB RAM мінімум, 8 GB RAM рекомендовано. Необхідно також 500 MB простору для встановлення Android Studio, щонайменше 1.5 GB для SDK, зображень емульованих систем, та кешів. Робота з Android Studio може значно сповільнювати роботу комп'ютера.

App Inventor — середовище візуальної розробки android-застосунків, що вимагає від користувача мінімальних знань програмування. Спочатку розроблене в Google Labs, після закриття цієї лабораторії було передане Массачусетському технологічному інституту. App Inventor - це хмарний інструмент, що означає, що ви можете створювати програми прямо у своєму веб-браузері. Можна тестувати додатки відразу на своєму мобільному пристрої, при чому існує кілька шляхів передачі APK – через USB, за

допомогою сканування QR-коду чи числового коду в додатку на мобільному пристрої. [24]

Оскільки взаємодія з користувачем не потребує складної логіки та user-інтерфейсу, для розробки Android-додатку був обраний саме App Inventor.

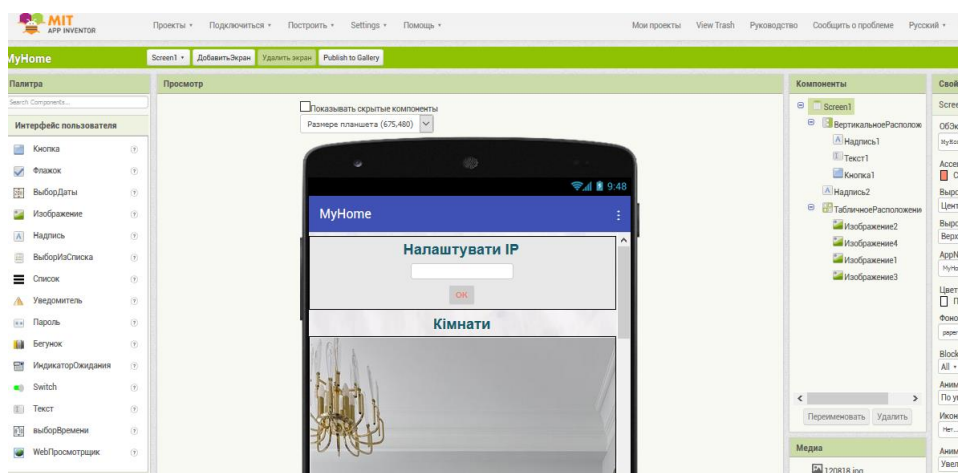


Рисунок 1.6 APP Inventor

Одна з опцій передачі APK-файлу на мобільний пристрій – за допомогою генерації QR-коду. Для встановлення додатку на смартфон потрібно спочатку встановити з PlayMarket додаток MIT AI2 Companion, і відсканувати згенерований код.

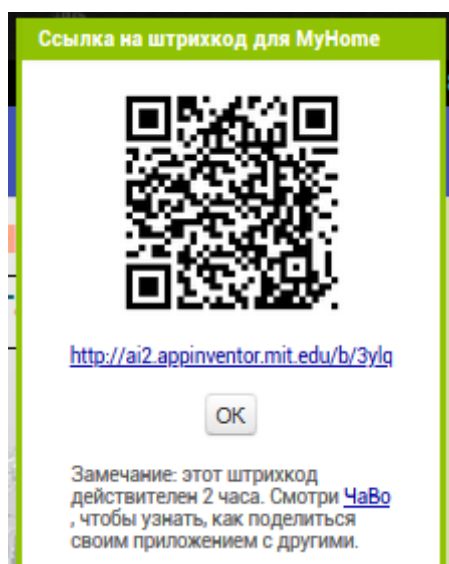


Рисунок 1.7 QR-код з посиланням на завантаження APK

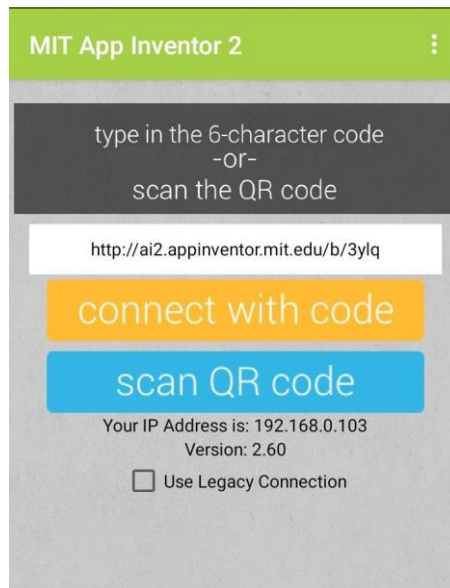


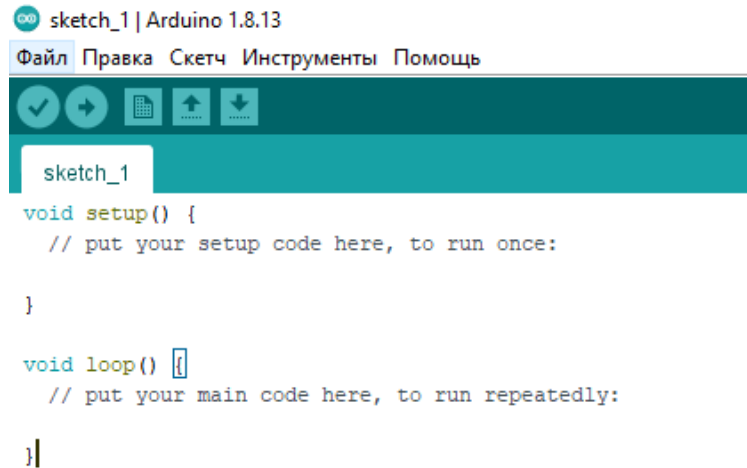
Рисунок 1.8 Способи завантаження APK з додатку

1.5 Середовище розробки прошивок для плат

Для роботи з датчиками та мобільним додатком плати NodeMCU потрібно запрограмувати. Для створення власних проектів на базі Arduino, необхідно писати прошивки і завантажувати їх в мікроконтролер. Для цього підходить середовище Arduino IDE. Є й інші способи написання прошивок і завантаження скетчів на плату, але найпростіший - це використати Arduino IDE.

Arduino IDE - інтегроване середовище розробки для Windows, MacOS і Linux, розроблена на C і C ++, призначена для створення і завантаження програм на Arduino-сумісні плати, а також на плати інших виробників. Використовується як для створення автономних об'єктів, так і підключення до програмного забезпечення через дротові і бездротові інтерфейси. Мова програмування Arduino називається «Arduino C» і являє собою мову C ++ з фреймворком Wiring. Є набір бібліотек, що включає в себе функції і об'єкти. При компіляції програми IDE створює тимчасовий файл з розширенням *.cpp. Програми, написані програмістом Arduino, називаються скетчами (від англ.

Sketch) і зберігаються в файлах з розширенням * .ino. Ці файли перед компіляцією обробляються препроцесором Arduino. Також існує можливість створювати і підключати до проєкту стандартні файли C ++.[25]



```
sketch_1 | Arduino 1.8.13
Файл Правка Скетч Инструменты Помощь
sketch_1
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Рисунок 1.9 Arduino IDE

1.6 Висновки до першого розділу

У першому розділі надано короткий опис існуючих готових рішень як для систем розумного дому в цілому, так і керування кондиціонером зокрема. В даний час існує багато технічних рішень, які мають як переваги, так і недоліки. Основним недоліком є як ціна модулів, так і ціна програмного забезпечення для них. Надано основні характеристики обладнання, за допомогою якого буде створюватись власне рішення: плати NodeMCU, реле, датчику температури і вологості, датчику змінної напруги. Для розробки мобільного додатку для Android буде використовуватись APP Inventor, візуальне середовище розробки. Програмування платформи NodeMCU буде здійснюватись у Arduino IDE.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Вступ до проектування

Як зрозуміло з попереднього розділу, апаратна частина системи віддаленого керування кондиціонером складається з плати NodeMCU, реле, датчика напруги та датчика температури-вологості. Програмна частина складається з, по-перше, з мобільного додатку для віддаленого керування, по-друге, з коду, що ним запрограмовано плату. Для реалізації потрібен також кондиціонер, а саме доступ до його блоку керування, та існування у помешканні мережі Wi-Fi.

Тобто, необхідно реалізувати:

- 1) Програмування плати.
- 2) Код мобільного додатку
- 3) Інтерфейс користувача
- 4) Під'єднання плати до кондиціонера.
- 5) Механізм передачі даних – через веб сервер.

2.2 Вимоги до системи

Мінімальним функціоналом для системи віддаленого керування кондиціонером є можливість за допомогою мобільного додатку дізнатись, чи ввімкнений кондиціонер в мережу, яка температура в кімнаті, яка вологість в кімнаті. Додатковою є можливість вмикати і вимикати прилад як з пульта, так і з мобільного додатку.

Визначимо поведінку кнопки вимкнення кондиціонера. Якщо кондиціонер включено:

1. Реле знаходиться в стані включено (логічний 0), детектор напруги повертає логічний 0 (є змінна напруга на кондиціонері), то для того щоб вимкнути кондиціонер, програмою необхідно відправити get запит виключити реле /16off. В результаті реле вимкнено (логічна 1), детектор напруги повертає теж логічну 1.
2. Реле знаходиться в стані вимкнено (логічна 1), а детектор напруги повертає логічний 0 (включено). (Кондиціонер було включено за допомогою пульта дистанційного керування). Для того, щоб вимкнути кондиціонер за допомогою мобільного додатку, треба надіслати get запит включити реле /16on. Статус реле набуде стан включено (логічний 0), детектор напруги поверне логічну 1 (вимкнено).

Передача даних відбувається через ESP8266WebServer: дані з плати передаються на ESP8266WebServer, звідки зчитуються мобільним додатком.

2.3 Висновки другого розділу

Стандартні бібліотеки NodeMCU дозволяють зчитувати значення показників на датчиках, та створити веб-сервер для передачі даних за допомогою запитів.

3 РЕАЛІЗАЦІЯ

3.1. Програмування плати NodeMCU

Для програмування плати NodeMCU треба підключити її до комп'ютера та налаштувати Arduino IDE. Для виконання попереднього налаштування потрібно, щоб комп'ютер був підключений до мережі Інтернет. Першим кроком буде установка драйвера CH340 і відкриття Arduino IDE, наступним – встановлення додаткового посилання для Менеджера плат (http://arduino.esp8266.com/stable/package_esp8266com_index.json), потім вибір пакета «esp8266 by ESP8266 Community» у вікні Менеджера плат. Після завантаження файлів треба обрати необхідну плату та порт.

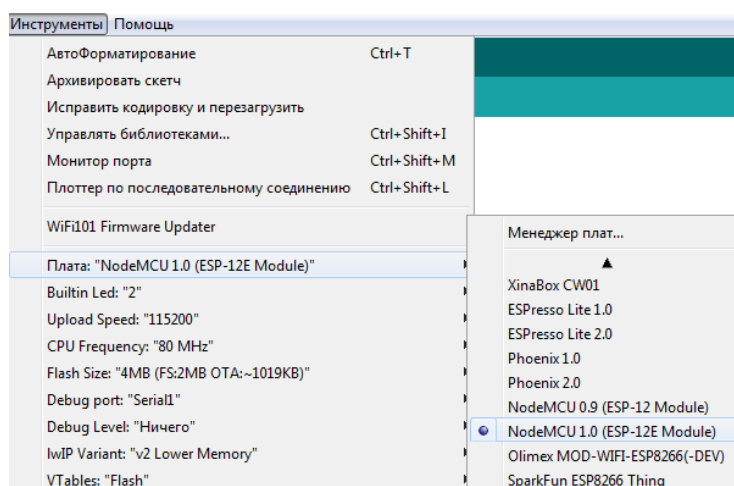


Рисунок 3.1 Вибір необхідної плати

Плату треба запрограмувати на передачу даних за допомогою Wi-Fi. Серед бібліотек, які будуть використовуватись є ESP8266WiFi.h. Саме вона містить функції, за допомогою яких ESP8266 здійснює підключення до Wi-Fi. Підключення до Wi-Fi ініціалізується рядком: `WiFi.begin ("network-name", "pass-to-network")`. Якщо з'єднання було встановлено, а потім з якоїсь причини втрачено, то ESP8266 автоматично перепідключиться до останньої робочої

точки доступу, коли вона знову буде доступна. Це буде виконано автоматично самою WiFi-бібліотекою, без втручання користувача.

В основному циклі відбувається отримання основних команд по мережі, зчитування значень з заданих входів детекторів напруги і з заданих виходів реле, отримання температури і вологості з датчика, виконувані функції температури і вологості, перемикання реле. Остання функція data – створення строки показів з датчиків і стану реле розділених комами для відправки в Android додаток.

```

1. #include <ESP8266WiFi.h> //стандартна бібліотека
2. #include <WiFiClient.h> //стандартна бібліотека
3. #include <ESP8266WebServer.h> //стандартна бібліотека
4. #include <DHT.h> //взаємодія з датчиком
5. #define DHTTYPE DHT22
6. #define DHTPIN 14 // D5

7. const char* ssid = "*****"; //задання мережі
8. const char* password = "*****";

9. ESP8266WebServer server(80); //запуск серверу

10.DHT dht(DHTPIN, DHTTYPE, 22); // робота датчика з esp8266
11.float humidity, temp_c; // дані з датчика
12.String webString=""; // текст для відправки
13.// Реле
14.int Rele1Pin=16; //D0
15.int Rele1State = HIGH;
16.int Rele2Pin=5; //D1
17.int Rele2State = HIGH;
18.int Rele3Pin=4; //D2

```

```
19.int Rele3State = HIGH;
20.int Rele4Pin=0;    //D3
21.int Rele4State = HIGH;
22.int Rele5Pin=2;    //D4
23.int Rele5State = HIGH;
24.// Детектор напруги
25.int Voltage_Sensor_1=LOW;
26.int Voltage_Sensor_Pin1=12; //D6
27.int Voltage_Sensor_2=LOW;
28.int Voltage_Sensor_Pin2=13; //D7
29.int Voltage_Sensor_3=LOW;
30.int Voltage_Sensor_Pin3=15; //D8
31.int Voltage_Sensor_4=LOW;
32.int Voltage_Sensor_Pin4=3; //Rx
33.int Voltage_Sensor_5=LOW;
34.int Voltage_Sensor_Pin5=1; //Tx
35.unsigned long previousMillis = 0; // час останнього читання
36.const long interval = 2000;      // інтервал читання з датчика
37.void setup() {
38.Serial.begin(9600);
39.dht.begin(); // датчик температури
40.pinMode(Voltage_Sensor_Pin1,INPUT); // готуємо GPI для реле
41.digitalWrite(Voltage_Sensor_Pin1,Voltage_Sensor_1);
42.pinMode(Voltage_Sensor_Pin2,INPUT);
43.digitalWrite(Voltage_Sensor_Pin2,Voltage_Sensor_2);
44.pinMode(Voltage_Sensor_Pin3,INPUT);
45.digitalWrite(Voltage_Sensor_Pin3,Voltage_Sensor_3);
46.pinMode(Voltage_Sensor_Pin4,INPUT);
47.digitalWrite(Voltage_Sensor_Pin4,Voltage_Sensor_4);
48.pinMode(Voltage_Sensor_Pin5,INPUT);
```

```
49.digitalWrite(Voltage_Sensor_Pin5,Voltage_Sensor_5);

50.pinMode(Rele1Pin, OUTPUT);
51.digitalWrite(Rele1Pin,Rele1State);
52.pinMode(Rele2Pin, OUTPUT);
53.digitalWrite(Rele2Pin,Rele2State);
54.pinMode(Rele3Pin, OUTPUT);
55.digitalWrite(Rele3Pin,Rele3State);
56.pinMode(Rele4Pin, OUTPUT);
57.digitalWrite(Rele4Pin,Rele4State);
58.pinMode(Rele5Pin, OUTPUT);
59.digitalWrite(Rele5Pin,Rele5State);
60.Serial.println();
61.Serial.println();
62.Serial.print("Connecting to ");
63.Serial.println(ssid);
64.WiFi.begin(ssid, password);
65.while (WiFi.status() != WL_CONNECTED) {
66.delay(500);
67.Serial.print(".");
68.}
69.Serial.println("");
70.Serial.println("WiFi connected");

71.server.begin();// стартує сервер
72.Serial.println("Server started");
73.Serial.print("IP address: ");// вивід отриманого IP
74.Serial.println(WiFi.localIP());
75.server.on("/16on", rel1_on);
76.server.on("/16off", rel1_off);
```

```
77.server.on("/5on", rel2_on);
```

```
78.server.on("/5off", rel2_off);
```

```
79.server.on("/4on", rel3_on);
```

```
80.server.on("/4off", rel3_off);
```

```
81.server.on("/0on", rel4_on);
```

```
82.server.on("/0off", rel4_off);
```

```
83.server.on("/2on", rel5_on);
```

```
84.server.on("/2off", rel5_off);
```

```
85.server.on("/t", data);
```

```
86. }
```

```
87.void loop() {
```

```
88.server.handleClient();
```

```
89.Rele1State= digitalRead(Rele1Pin);
```

```
90.digitalWrite(Rele1Pin,Rele1State);
```

```
91.Rele2State= digitalRead(Rele2Pin);
```

```
92.digitalWrite(Rele2Pin,Rele2State);
```

```
93.Rele3State= digitalRead(Rele3Pin);
```

```
94.digitalWrite(Rele3Pin,Rele3State);
```

```
95.Rele4State= digitalRead(Rele4Pin);
```

```
96.digitalWrite(Rele4Pin,Rele4State);
```

```
97.Rele5State= digitalRead(Rele5Pin);
```

```
98.digitalWrite(Rele5Pin,Rele5State);
```

```
99.Voltage_Sensor_1 = digitalRead(Voltage_Sensor_Pin1);
```

```
100.    digitalWrite(Voltage_Sensor_Pin1,Voltage_Sensor_1);
```

```
101.    Voltage_Sensor_2 = digitalRead(Voltage_Sensor_Pin2);
```

```
102.    digitalWrite(Voltage_Sensor_Pin2,Voltage_Sensor_2);
```

```
103. Voltage_Sensor_3 = digitalRead(Voltage_Sensor_Pin3);
104. digitalWrite(Voltage_Sensor_Pin3,Voltage_Sensor_3);

105. Voltage_Sensor_4 = digitalRead(Voltage_Sensor_Pin4);
106. digitalWrite(Voltage_Sensor_Pin4,Voltage_Sensor_4);

107. Voltage_Sensor_5 = digitalRead(Voltage_Sensor_Pin5);
108. digitalWrite(Voltage_Sensor_Pin5,Voltage_Sensor_5);
109. gettemperature();
110. }

111. // отримання температури
112. void gettemperature() {
113. unsigned long currentMillis = millis();

114. if(currentMillis - previousMillis >= interval) {
115. previousMillis = currentMillis;
116. humidity = dht.readHumidity(); // читання вологості

117. // temp_f = dht.readTemperature(true); // температура в
    фаренгейтах
118. temp_c = dht.readTemperature(false); // температура в цельсіях
119. if (isnan(humidity) || isnan(temp_c)) {
120. Serial.println("Failed to read from DHT sensor!");
121. return;
122. }
123. }
124. }
```

```
125. void rel1_on() {
126.   digitalWrite(Rele1Pin, LOW); // управління реле
127.   server.send(200, "text/html", "gpio16 on");
128. }
129. void rel1_off() {
130.   digitalWrite(Rele1Pin, HIGH);
131.   server.send(200, "text/html", "gpio16 off");
132. }

133. void rel2_on() {
134.   digitalWrite(Rele2Pin, LOW);
135.   server.send(200, "text/html", "gpio5 on");
136. }
137. void rel2_off() {
138.   digitalWrite(Rele2Pin, HIGH);
139.   server.send(200, "text/html", "gpio5 off");
140. }

141. void rel3_on() {
142.   digitalWrite(Rele3Pin, LOW);
143.   server.send(200, "text/html", "gpio4 on");
144. }
145. void rel3_off() {
146.   digitalWrite(Rele3Pin, HIGH);
147.   server.send(200, "text/html", "gpio4 off");
148. }

149. void rel4_on() {
150.   digitalWrite(Rele4Pin, LOW);
151.   server.send(200, "text/html", "gpio0 on");
```

```

152.     }
153.     void rel4_off() {
154.         digitalWrite(Rele4Pin, HIGH);
155.         server.send(200, "text/html", "gpio0 off");
156.     }

157.     void rel5_on() {
158.         digitalWrite(Rele5Pin, LOW);
159.         server.send(200, "text/html", "gpio2 on");
160.     }
161.     void rel5_off() {
162.         digitalWrite(Rele5Pin, HIGH);
163.         server.send(200, "text/html", "gpio2 off");
164.     }

165.     // відправка температури і вологості, статусу реле та детектора
        напруги
166.     void data() {
167.         webString = String((int)temp_c)+","+String((int)humidity)+","+
            String((int)Rele1State) +","+ String((int) Voltage_Sensor_1)+","+
            String((int)Rele2State) +","+ String((int) Voltage_Sensor_2)+","+
            String((int)Rele3State) +","+ String((int) Voltage_Sensor_3)+","+
            String((int)Rele4State) +","+ String((int) Voltage_Sensor_4)+","+
            String((int)Rele5State) +","+ String((int) Voltage_Sensor_5);
168.         Serial.print
            (String((int)temp_c)+","+String((int)humidity)+","+String((int) Rele1State)
            +","+String((int) Voltage_Sensor_1)+","+String((int) Rele2State)
            +","+String((int) Voltage_Sensor_2)+","+String((int) Rele3State)
            +","+String((int) Voltage_Sensor_3)+","+String((int) Rele4State)

```



```

+"," +String((int) Voltage_Sensor_4)+"," +String((int) Relе5State)
+"," +String((int) Voltage_Sensor_5));

```

```

169.     server.send(201, "text/plain", webString);

```

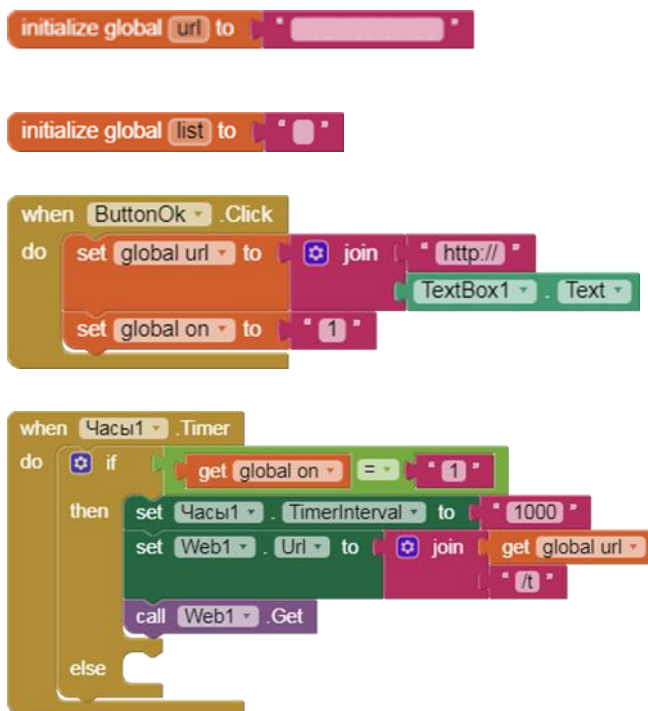
```

170.     }

```

3.2 Створення мобільного додатку

Мобільний додаток створюється у APP Inventor. Додаток має три екрани: стартовий, для задання IP адреси та вибору кімнати, екран вибору приладу (з кількох підключених розумних приладів у кімнаті), та екран перегляду параметрів на приладі. Наразі демонструється для функціоналу кондиціонеру.



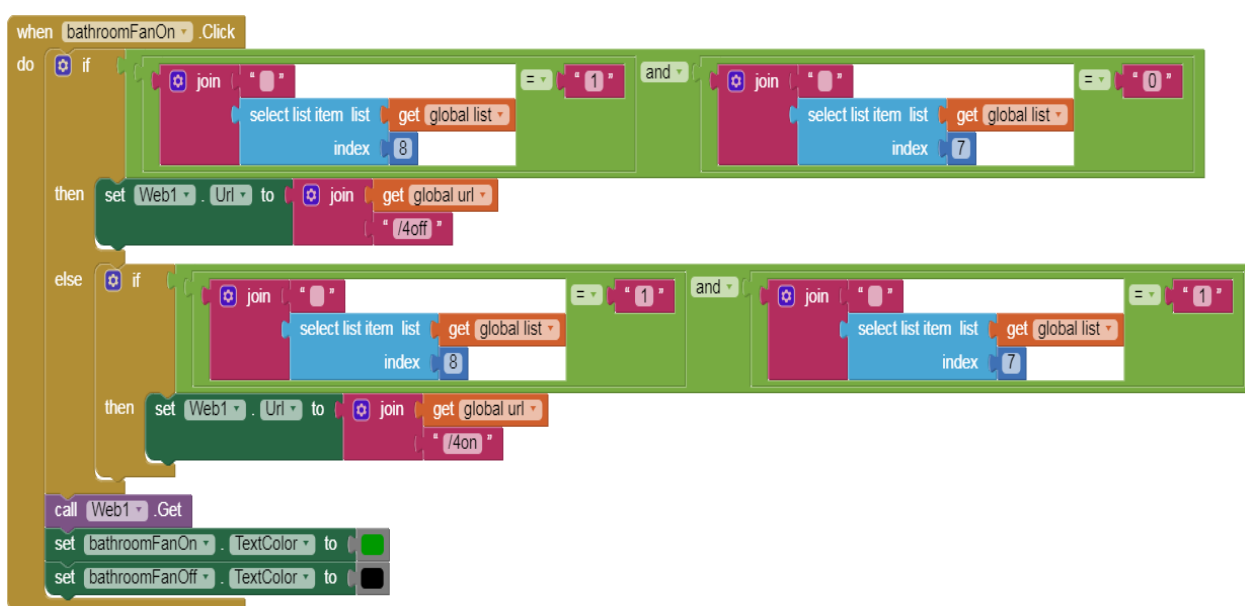
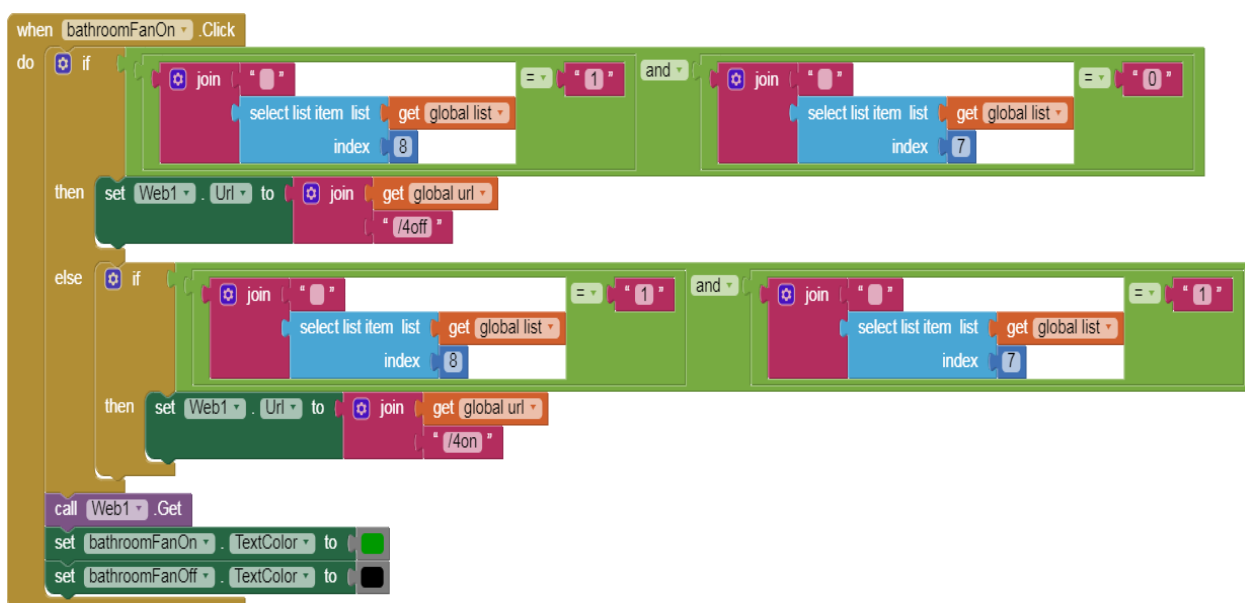
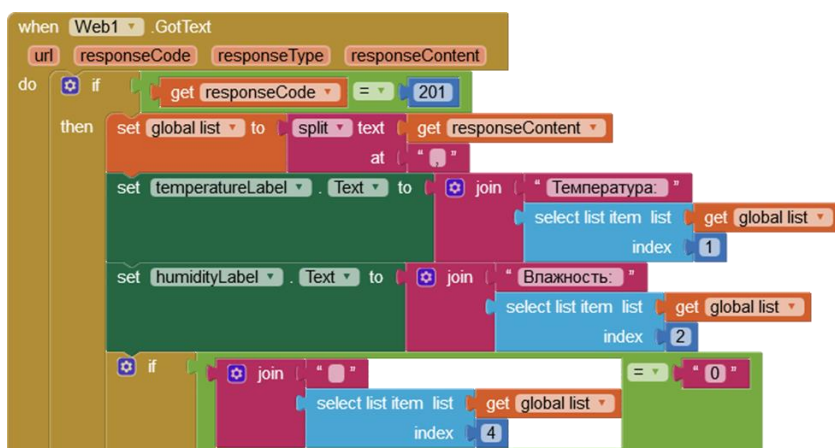


Рисунок 2.2 Скетч додатку

3.3. Підключення плати

Для роботи з кондиціонером треба приєднати зібрану конструкцію до плати керування кондиціонером. Треба, знявши корпус, дістати плату управління. На моделях, що підтримують модулі WiFi, роз'єм CN105 вже встановлений, якщо його немає, то необхідно припаяти, і зібрати необхідні елементи.

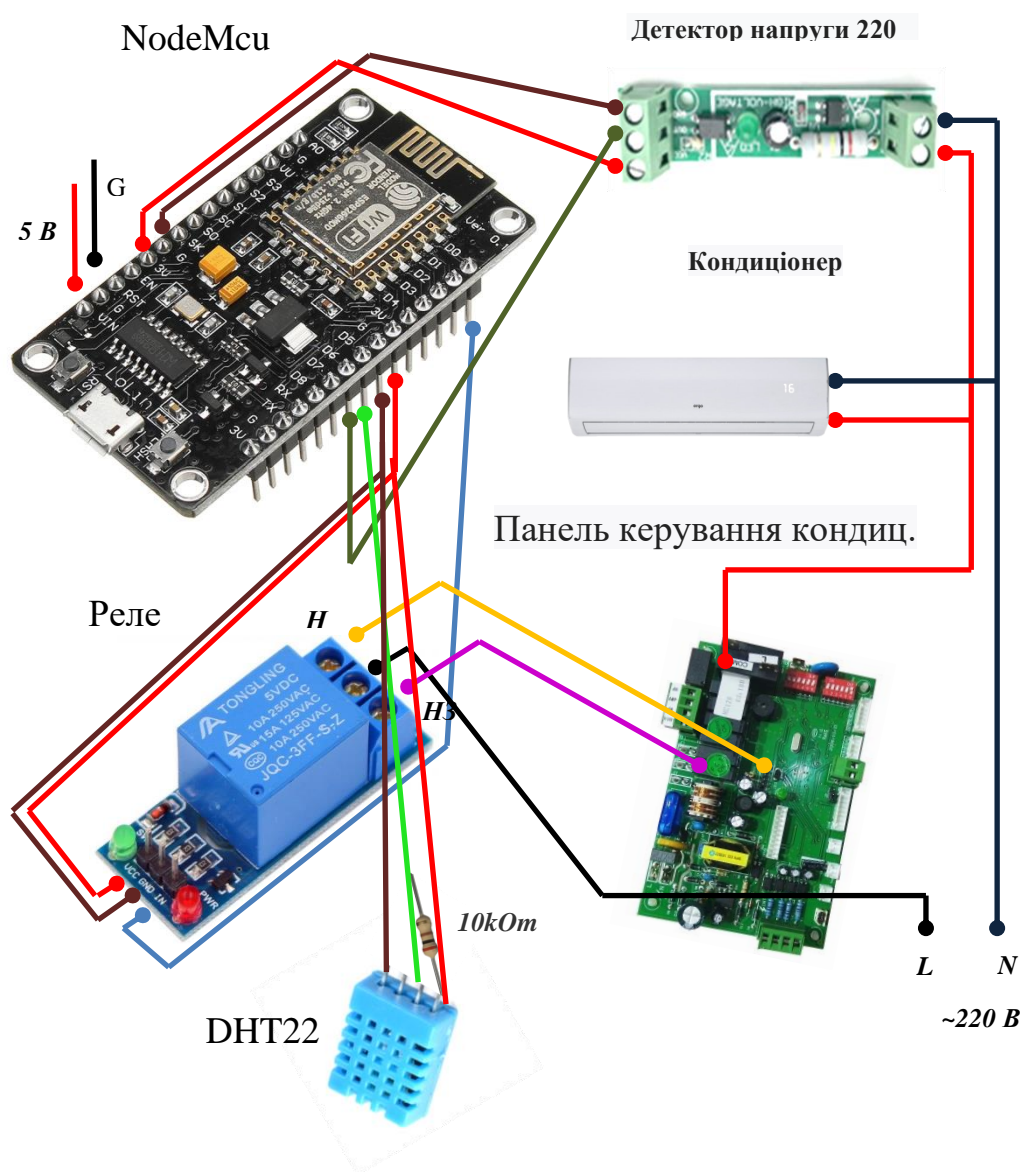


Рисунок 3.3 Схема підключення

4 ВИСНОВКИ

Основною тенденцією в останні роки є бурхливий ріст кількості пристроїв, підключених і керованих Інтернетом. Широкий спектр застосувань технологій автоматизації означає, що особливості можуть сильно відрізнятися від пристрою до пристрою, але більшість із них мають подібні характеристики. Інтернет речей створює можливості для інтеграції фізичного світу в комп'ютерні системи, що призводить до підвищення ефективності, економічних вигод та зменшення людських зусиль. Домашня автоматизація, як окремий випадок інтернету речей, охоплює доступні через інтернет домашні пристрої. Домашня автоматизація вирішує різноманітні задачі і має чимало переваг: полегшує ведення господарства, допомагає персонам з особливими потребами, контролює інженерні системи, та багато іншого, з мінімальною участю людини. Дистанційне керування домашніми пристроями є складовою домашньої автоматизації. Засоби керування можуть бути вбудованими чи встановленими додатково, виробники з багатьох країн надають як апаратні, так і програмні засоби для цього. Засоби для дистанційного керування кондиціонером можна придбати або зробити самостійно, як описано у роботі.

Метою роботи було створення мобільного додатку та реалізація його взаємодії з побутовим приладом на прикладі кондиціонера, щоб забезпечити віддалене керування та контроль.

Був виконаний аналіз технологій та описані існуючі підходи та готові рішення, розроблений додаток та перелік рекомендацій з його використання.

Система забезпечує своєчасне отримання інформації про використання побутового приладу, дає можливість дистанційного керування температурою.

Технічне рішення: для взаємодії з мобільним додатком кондиціонер оснащено платою NodeMcu на основі модуля ESP8266 для передачі сигналу в локальну мережу або інтернет через Wi-Fi. Плата запрограмована в Arduino

IDE. Користувацький інтерфейс додатку віддаленого керування для смартфонів на базі Android створено за допомогою APP Inventor.

ПЕРЕЛІК ПОСИЛАНЬ

1. TechTarget. Internet of things (IoT). Margaret Rouse Blog. URL : <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> (дата звернення: 20.08.2020).
2. Linux.com. 21 Open Source Projects for IoT. Linux.com News. URL : <https://www.linux.com/news/21-open-source-projects-iot/> (дата звернення: 21.08.2020).
3. ITU. Internet of Things Global Standards Initiative. Global Standards URL : Initiatives. <https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx> (дата звернення: 21.08.2020).
4. London Datastore. The Trouble with the Internet of Things. London Datastore Blog. URL : <https://data.london.gov.uk/blog/the-trouble-with-the-internet-of-things/> (дата звернення: 23.08.2020).
5. Gerhart J. Home Automation and Wiring / New York : McGraw-Hill Professional, 1999. 322 p. ISBN 0-07-024674-2.
6. Springer Open. An enhanced security framework for home appliances in smart home. Springer Open Articles. URL : <https://hcis-journal.springeropen.com/articles/10.1186/s13673-017-0087-4> (дата звернення: 02.09.2020).
7. Mann W. Smart Technology for Aging, Disability, and Independence: The State of the Science / Mississauga : John Wiley & Sons Canada, Ltd., 2005. 379 p. ISBN 0-471-69694-3.
8. Habr. GSM модули для умного дома. Habr Blog. URL : <https://habr.com/ru/company/intems/blog/513798/> (дата звернення: 12.09.2020).
9. Qivicon. The platform for Smart Home. Qivicon Articles. URL : <https://www.qivicon.com/en/> (дата звернення: 23.09.2020).

10. Homee. Smart home. Homee Articles. URL : <https://hom.ee/en/> (дата звернення: 23.09.2020).
11. Megasensor. Умный дом на платформе Homematic. Megasensor Products. URL : <https://megasensor.com/products/umnyj-dom-na-platfome-homematic/> (дата звернення: 25.09.2020).
12. RWE Group. Energy for a sustainable future. RWE Articles. URL : <https://www.group.rwe/> (дата звернення: 26.09.2020).
13. Netatmo shop. Netatmo Products. Netatmo Main Page. URL : <https://shop.netatmo.com/en-us/> (дата звернення: 23.09.2020).
14. Broadlink. Умный дом Aqara. Broadlink Articles. URL : <https://broadlink.ru/aqara-smart-home-obzor> (дата звернення: 26.09.2020).
15. Sensibo. Make your air conditioner smart. Sensibo Main Page. URL : <https://sensibo.com/> (дата звернення: 03.10.2020).
16. Tado. Tado Products. Tado Overview. URL : <https://www.tado.com/gb-en/tado-products-overview> (дата звернення: 03.10.2020).
17. Espressif. ESP8266EX Datasheet. Espressif Documentation. URL : https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf (дата звернення: 09.10.2020).
18. СРК-М. Цены на СРК-М2, СРК-М, СРК-М-01 и программное обеспечение MonSRK. Продукты СРК-М. URL : <http://www.cpk-m.ru/ceny/ceny-na-srk-m2-srk-m-srk-m-01-i-programnoe-obespechenie-monsrk.html> (дата звернення: 20.11.2020).
19. Coolautomation. CoolPlug Coolinkhub. Coolautomation Products. URL : <https://coolautomation.com/products/coolplug-coolinkhub/> (дата звернення: 03.11.2020).
20. AliExpress. Модуль ESP8266 NodeMcu v3. AliExpress Products. URL : https://cutt.ly/aliexpress_esp8266 (дата звернення: 20.11.2020).

21. Wikipedia. Microsoft Visual Studio. Wikipedia Articles.
URL : https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення: 13.10.2020).
22. Eclipse. Eclipse Foundation. Eclipse Main Page.
URL : <https://www.eclipse.org/> (дата звернення: 13.10.2020).
23. Android Developer. Android Studio. Android Developer Articles.
URL : <https://developer.android.com/studio> (дата звернення: 13.10.2020).
24. MIT. MIT App Inventor. MIT App Inventor Main Page.
URL : <https://appinventor.mit.edu/> (дата звернення: 23.09.2020).
25. Arduino. Arduino Software Release Notes. Arduino Articles.
URL : <https://www.arduino.cc/en/Main/ReleaseNotes> (дата звернення: 13.10.2020).
26. Arduino.ua. Arduino marketplace. Arduino.ua Main Page.
URL : <https://arduino.ua/> (дата звернення: 20.10.2020).
27. Rozetka.com.ua. Датчик TZT контролю наявності мережевого напруження 220В з опторозв'язкою.
URL : <https://rozetka.com.ua/161334055/p161334055/> (дата звернення: 29.11.2020).
28. GSM Комплект. Модуль твердотельного реле 4-х каналний, 5В, 2А. URL : <https://gsm-komplekt.ua/product/58176/> (дата звернення: 29.11.2020).
29. Arduino.ua. Arduino marketplace. Датчик вологості і температури DHT22. URL : <https://arduino.ua/prod301-datchik-vlajnosti-i-temperatyri-dht22> (дата звернення: 29.11.2020).

ДОДАТОК А

Обладнання

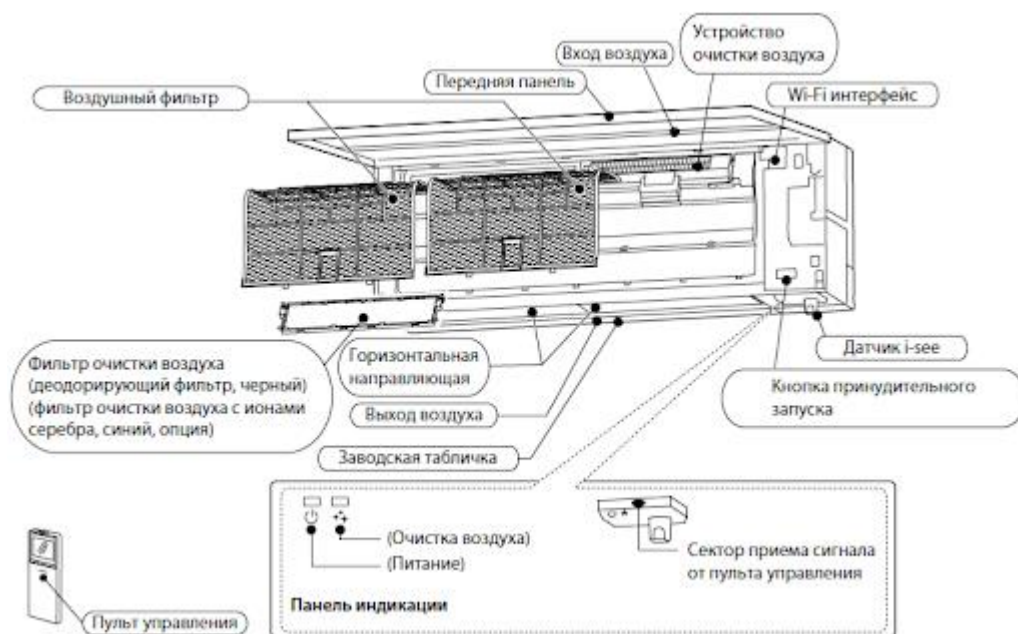


Рисунок А.1 Кондиціонер

ДОДАТОК Б

Скріншоти додатку



Рисунок Б.1 Головна сторінка

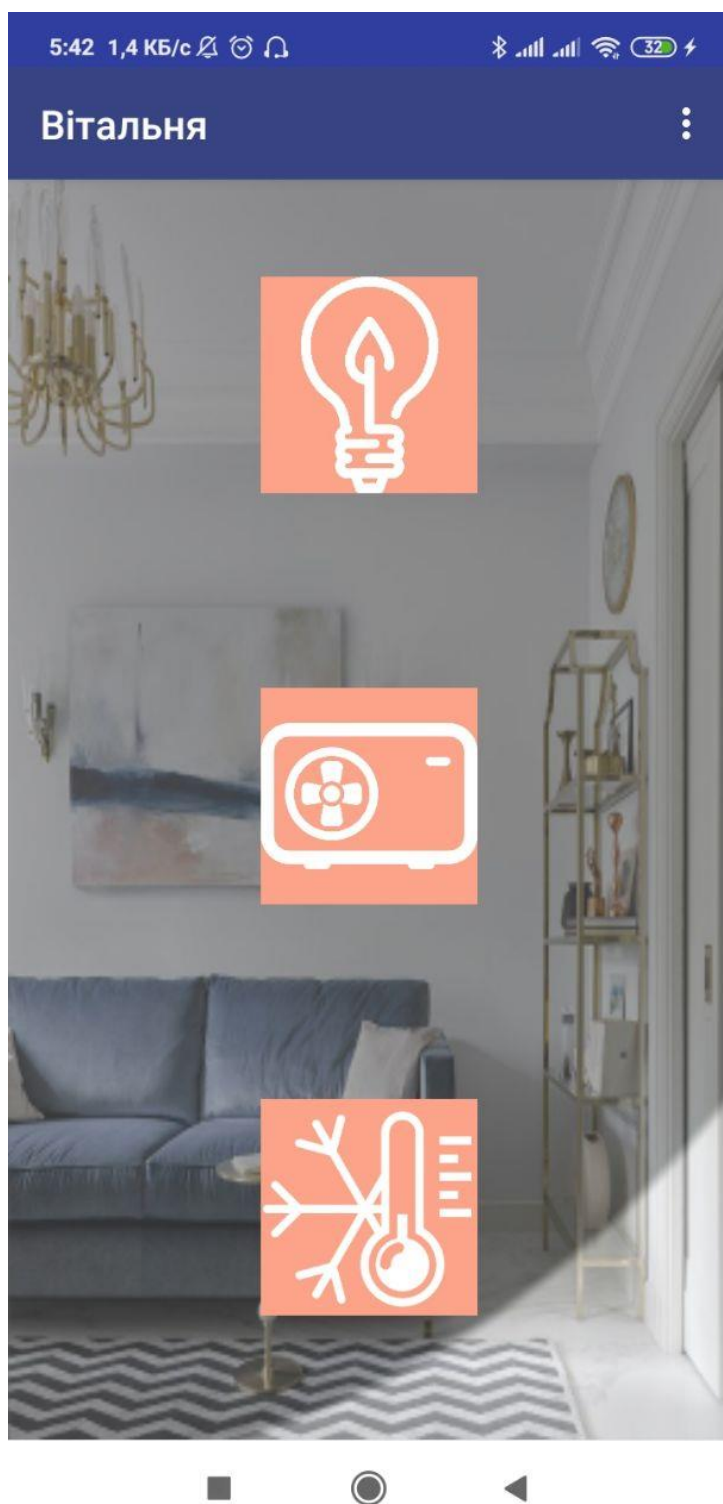


Рисунок Б.2 Доступні прилади для кімнати

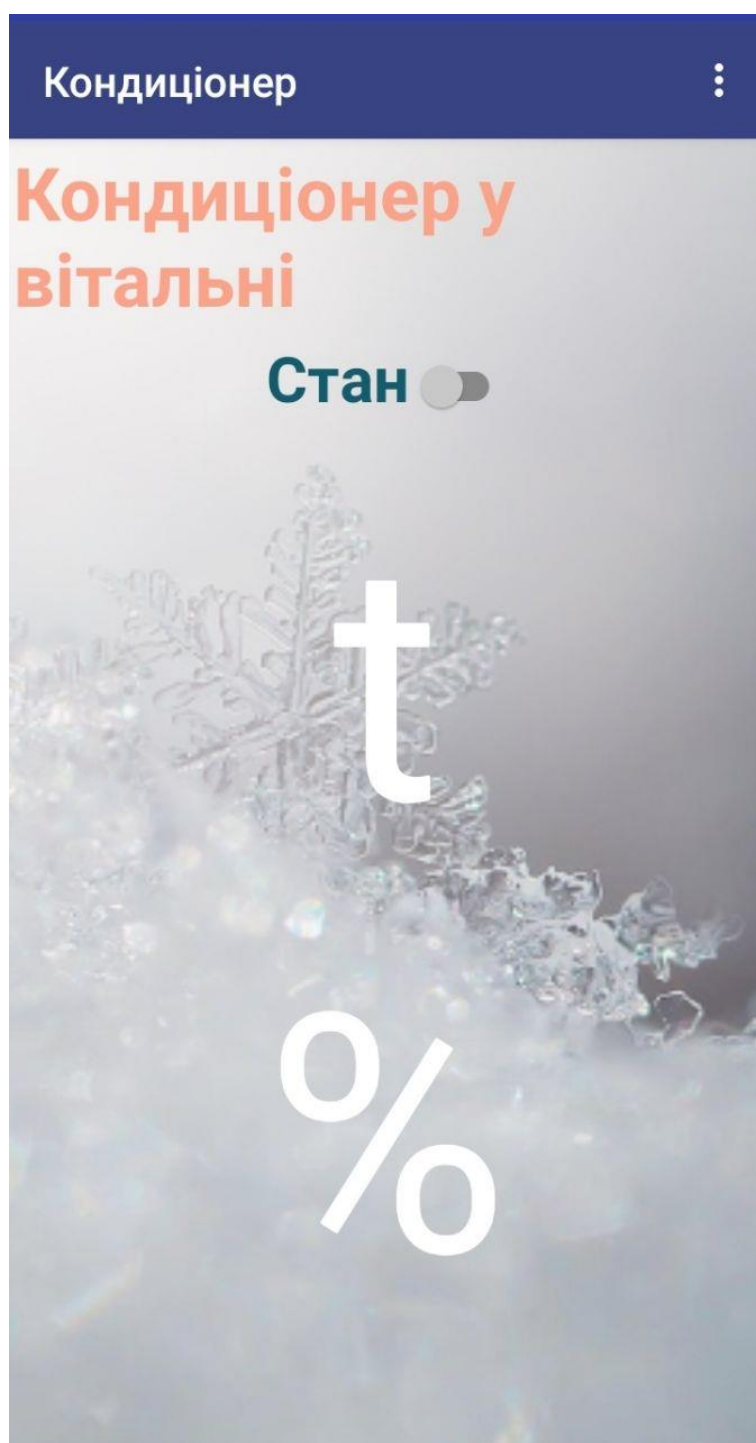


Рисунок Б.3 Керування кондиціонером



Рисунок Б.4 Отримання показників

ДОДАТОК В

Інструкція

1. Перед початком роботи необхідно вказати коректний IP.
2. Після збереження IP користувачу пропонується у блоці «Кімнати» перейти до зони квартири, показниками в якій потрібно керувати. На вибір надаються зони. У версії для однокімнатної квартири надаються зони «Вітальня», «Кухня», «Коридор», «Ванна кімната».
3. Для обраної кімнати відображаються піктограми функцій, що можуть бути налаштовані: наприклад, «Освітлення», «Кондиціонування», «Вологість приміщення».
4. При кліку на відповідну піктограму відбувається перехід на екран показника. Відображаються стани і дії, що їх може зробити користувач. Користувач може побачити, чи ввімкнений пристрій, ввімкнути чи вимкнути, побачити поточний стан, налаштувати параметри.