

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач кафедри

\_\_\_\_\_ Литвиненко О.Є.

«\_\_\_» \_\_\_\_\_ 2021 р.

**ДИПЛОМНИЙ ПРОЕКТ**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ  
"БАКАЛАВР"**

Тема: \_\_\_\_\_ Система дистанційного керування роботою побутових приладів

Виконавець: \_\_\_\_\_ Кіслов О.І.

Керівник: \_\_\_\_\_ Апенько Н.В.

Нормоконтролер: \_\_\_\_\_ Тупота Є.В.

**Київ 2021**

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет \_\_\_\_\_ кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління \_\_\_\_\_

Освітнього ступеня \_\_\_\_\_ бакалавр

Напрямок (спеціальність) 123 "Комп'ютерна інженерія"  
(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Литвиненко О. Є.

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

## ЗАВДАННЯ на виконання дипломної роботи (проекту)

Кіслова Олександра Ігоровича

(прізвище, ім'я, по батькові)

**1. Тема роботи:** \_\_\_\_\_ Система дистанційного керування роботою побутових приладів \_\_\_\_\_

затверджена наказом ректора від "04" \_\_\_\_\_ лютого 2021 року № 135 /ст.

**2. Термін виконання роботи:** з 17.05.2021 до 20.06.2021

**3. Вихідні дані до роботи:** 1) середовище розробки *Android Studio*;

2) мова програмування *Java*; 3) середовище розробки *Arduino*, 4) мова програмування *Arduino*, 5) функціональні та нефункціональні вимоги

**4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):**

1) аналіз методів дистанційного керування роботою побутових приладів та постановка завдання проектування;

2) принципи роботи системи дистанційного керування побутовими приладами;

3) описання прийнятих проектних рішень.

**5. Перелік обов'язкового графічного матеріалу:**

1) діаграма прецедентів;

2) діаграма послідовності для керування пристроєм через додаток;

3) діаграма послідовності для керування пристроєм через *SMS*;

4) діаграма взаємодії;

5) робочі вікна програмного модуля;

6) схема алгоритму роботи системи;

7) схема алгоритму процедури *Device work process*.

## 6. Календарний план

№ п/п	Етапи виконання дипломного проекту	Термін виконання етапів	Примітка
1	Провести аналіз літератури за темою дипломного проекту та аналіз існуючих систем	17.05.21-18.05.21	
2	Провести аналіз принципів побудови систем дистанційних керування побутовими приладами	19.05.21-22.05.21	
3	Розробити модель системи дистанційного керування побутовими приладами	23.05.21-27.05.21	
4	Розробити апаратні засоби системи дистанційного керування побутовими приладами	28.05.21-29.05.21	
5	Розробити програмні модулі системи дистанційного керування побутовими приладами	30.05.21-02.06.21	
6	Написати пояснювальну записку	03.06.21-10.06.21	
7	Підготувати презентацію та графічні матеріали	11.06.21-14.06.21	

## 7. Дата видачі завдання « 17 » травня 2021 р.

Керівник дипломного проекту \_\_\_\_\_ Апенько Н.В.  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_ Кіслов О.І.  
(підпис студента)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту “Система дистанційного керування роботою побутових приладів”: 56 с., 25 рис., 18 літературних джерел.

*BLUETOOTH, GSM, ANDROID, ARDUINO, СКЕТЧ.*

Об’єкт – дистанційний спосіб керування роботою побутових приладів.

Предмет – система дистанційного керування роботою побутових приладів.

Мета – розробити програмний модуль та апаратний модуль системи дистанційного керування роботою побутових приладів.

Методи проектування – визначення основних правил обміну даними між модулями системи та параметрів системи, які при цьому змінюються, розробка програмного забезпечення з застосуванням мов програмування високого рівня.

Прогнозні припущення щодо розвитку об’єкта – створення робочого зразка системи та використання його для дистанційного керування роботою побутового приладу.

Результати дипломного проектування рекомендується використовувати при розробці нових програмних засобів для дистанційного керування роботою побутових приладів.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ДИСТАНЦІЙНОГО КЕРУВАННЯ РОБОТОЮ ПОБУТОВИХ ПРИЛАДІВ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ.....	11
1.1. Аналіз сучасних технологій та методів реалізації систем дистанційного керування побутовими приладами .....	11
1.2. Аналіз аналогів та готових програмних рішень в області дистанційного керування побутовими приладами .....	19
1.3. Методи та технології дистанційного керування побутовими приладами.....	21
1.4. Постановка завдання проектування .....	25
1.5. Висновки до розділу .....	26
РОЗДІЛ 2 ПРИНЦИПИ РОБОТИ СИСТЕМИ ДИСТАНЦІЙНОГО КЕРУВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ.....	27
2.1. Функціональні вимоги до системи дистанційного керування побутовими приладами .....	27
2.2. Структура системи дистанційного керування побутовими приладами.....	28
2.3. Функціональне забезпечення системи дистанційного керування побутовими приладами .....	32
2.4. Висновки до розділу .....	33
РОЗДІЛ 3 ОПИСАННЯ ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ .....	34
3.1. Алгоритм роботи системи дистанційного керування побутовими приладами .....	34
3.2. Розробка апаратного модуля.....	36
3.3. Розробка програмного модуля.....	43

3.4. Керівництво користувача .....	48
3.5. Висновки до розділу .....	51
ВИСНОВКИ .....	53
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТОК А.....	57

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

*UML – Unified Modeling Language*

*GSM – Global System for Mobile Communications*

*EIB – European Installation Bus*

*AES – Advanced Encryption Standard*

*WPA – Wi-Fi Protected Access*

*WEP – Wired Equivalent Privacy*

*WPAN – Wireless personal area network*

## ВСТУП

Концепція інтернету речей – обчислювальної мережі фізичних предметів, які з'єднуються між собою за допомогою вбудованих технологій, – була створена в 1999 році. До цього спроби дистанційного керування побутовими приладами та домашніми системами (що отримало термін «розумний будинок»), звичайно, теж були, але тільки в останні 10 років можна говорити про єдину мережу.

Останнім часом «розумний дім» ставав все розумнішими і розумнішими, а з часом став ще і доступним для всіх (простий і недорогий). Тепер є дійсністю можливість, коли можна прийти додому в нагріту або, навпаки, охолоджену квартиру, до готової чашки кави і підігрітого обіду.

Останнім часом «інтернет речей» став доступний буквально з розетки - якщо, звичайно, в неї встроєний спеціальний керуючий центр.

Сучасне житло обладнане безліччю електроприладів, деякі з яких можна ввімкнути або вимкнути за допомогою найпростіших приладів дистанційно керування. Наприклад для того щоб ввімкнути або вимкнути освітлення, доведеться встати і підійти до вимикача. За допомогою системи дистанційного керування без зусиль можна управляти побутовими приладами.

В даний час з'являється все більше пристроїв, що працюють дистанційно – так званих «розумних» пристроїв. Їх головними функціями є дистанційне керування (в найпростішому випадку ввімкнення і вимкнення) будь-якого обладнання.

Безумовними перевагами розумного будинку є те, що керування побутовими приладами та пристроями за допомогою засобів автоматики дозволяє:

1. Забезпечити високу точність і оперативність;
2. Уникнути помилок, які визначаються людським фактором;
3. Організувати дистанційний (віддалений) контроль і доступ до обладнання.



Зростання популярності автоматизованих систем, таких як розумний будинок, обумовлено прагненням людини до комфорту і зручності. Додатковою перевагою є безпека, будь то протипожежна система або сигналізація з дистанційним оповіщенням.

Об'єкт – дистанційний спосіб керування роботою побутових приладів.

Предмет – система дистанційного керування роботою побутових приладів.

Мета – розробити програмний модуль та апаратний модуль системи дистанційного керування роботою побутових приладів.

Методи проектування – визначення основних правил обміну даними між модулями системи та параметрів системи, які при цьому змінюються, розробка програмного забезпечення з застосуванням мов програмування високого рівня.

Завдання проектування:

1. Проаналізувати сучасні програмні рішення та методи реалізації систем дистанційного керування;

2. Розробити структуру системи дистанційного керування роботою побутових приладів;

3. Реалізувати алгоритм системи дистанційного керування роботою побутових приладів;

4. Реалізувати програмний модуль та апаратний модуль системи дистанційного керування роботою побутових приладів.

Для вирішення поставленої задачі використані наступні методи:

1. Теоретичні: аналіз, систематизація, зіставлення, класифікація джерел, що дало змогу отримати та узагальнити інформацію про різні функції систем дистанційного керування роботою побутових приладів;

2. Практичні: в середовищі розробки *Android Studio*, за допомогою об'єктно-орієнтованого методу програмування та мови програмування *Java*, створено додаток для керування апаратною частиною. Апаратна частина створена на базі платформи *Arduino NANO* мовою програмування *Arduino*. Також для створення апаратної частини використані: *GSM*-модуль *SIM800L*, *Bluetooth*

HC-05, датчик температури *Dallas DS18B20* в захищеному корпусі, 1-канальний реле модуль та блок живлення 12V.

Практичне значення дипломного проекту полягає у наданні можливості дистанційного керування роботою побутового приладу за допомогою *SMS* повідомлень та *Android* додатку для управління по *Bluetooth*, а також результати дипломного проектування можна використовувати при розробці нових програмних засобів для дистанційного керування роботою побутових приладів.

У першому розділі описано сучасні технології та методи реалізації систем дистанційного керування побутовими приладами. Для аналізу обрано такі відомі рішення як: *1-Wire*, *C-Bus*, *EIB/KNX*, *LanDrive*, *LonWorks*, *X10* та *Z-Wave*. Проаналізовано аналоги та готові програмні рішення в області дистанційного керування побутовими приладами. Для аналізу обрано такі відомі рішення як: *REDMOND SkyKettle M173S-E*, *MIE Smart Kettle*, *Xiaomi Mijia Multifunctional Electric Cooker*, *E.Zicom Aromatea*, *Element El'kettle WF11MB / MW*, *Polaris PWK 1725CGLD WIFI IQ Home*. Проаналізовано технології створення апаратних модулів дистанційного керування побутовими приладами. Для реалізації даного проекту обрано технологію *Bluetooth*, оскільки вона підтримує досить високу швидкість передачі даних і досить дешева. Поставлено завдання проектування.

У другому розділі описано базові функціональні вимоги до системи дистанційного керування побутовими приладами та показано структуру системи. Розроблено діаграми прецедентів, послідовності та взаємодії, а також функціональна схема роботи програмного модуля. Описано функціональне забезпечення системи дистанційного керування побутовими приладами.

У третьому розділі описано алгоритм роботи системи дистанційного керування роботою побутових приладів, який відображає основні процеси, що відбуваються у системі. Наведено схему підключення всіх складових апаратної частини, принципову схему апаратної частини та покрокову розробку апаратної та програмної частин системи дистанційного керування роботою побутових приладів. Описано керівництво користувача, у якому показано основні функції системи дистанційного керування роботою побутових приладів.

РОЗДІЛ 1  
АНАЛІЗ МЕТОДІВ ДИСТАНЦІЙНОГО КЕРУВАННЯ РОБОТОЮ  
ПОБУТОВИХ ПРИЛАДІВ ТА ПОСТАНОВКА ЗАВДАННЯ  
ПРОЕКТУВАННЯ

**1.1. Аналіз сучасних технологій та методів реалізації систем  
дистанційного керування побутовими приладами**

Сучасні технології надають можливість створювати систему «Розумний будинок» на основі модульної структури, тобто вибирати тільки ті модулі, які відповідають за певні і необхідні функції розумного будинку. При цьому вартість таких систем буде невисокою, що дозволить істотно заощадити фінансові та матеріальні витрати.

Перелік застосовуваних технологій побудови розумного будинку в наш час: *1-Wire, C-Bus, EIB/KNX, Helvar, LanDrive, LonWorks, ONE-NET, X10, Z-Wave, PLC, BACnet* і ін. Залежно від регіону світу ті чи інші технології створення розумного будинку набули найбільшого поширення:

- на американському континенті – *LonWorks*;
- в Європі – *EIB*;
- в Великобританії, Австралії та на більшій території Азії – *C-Bus*.

1.1.1. Принципи дистанційного керування побутовими приладами за технологією *1-Wire*

*1-Wire* – технологія, що дозволяє об'єднати різні датчики і прилади в одну загальну мережу (рис. 1.1). Управління в мережі здійснюється за допомогою об'єктно-орієнтованого методу персонального комп'ютера.

Кафедра КСУ				НАУ 21 06 34 000 ПЗ			
<i>Виконав</i>	<i>Кіслов О.І.</i>			Аналіз методів дистанційного керування роботою побутових приладів та постановка завдання проектування	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	<i>11</i>	<i>56</i>
<i>Консульт.</i>					123 СП 437		
<i>Норм. контр.</i>	<i>Тупота Є.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

Основа побудови мережі на базі 1-Wire - це топологія загальної шини, коли кожне з технічних пристроїв підключається безпосередньо до єдиної магістралі, без каскадних з'єднань або розгалужень.

У 1-Wire-мережах для передачі даних застосовується тільки один дріт.

Особливості системи на базі 1-Wire – просте і нескладне налаштування, легкість налагодження та обслуговування мережі будь-якої конфігурації.

Системи на базі 1-Wire включають в себе:

1. Персональний комп'ютер, адаптери для *COM* або *USB*-порту;
2. Датчики (температурні, цифрові, харчування, руху, протікання води, освітленості і вологості);
3. Відеокомутатори;
4. Розетки з інтерфейсом 1-Wire;
5. Модулі аналогових сигналів, модулі управління навантаженням 220В.

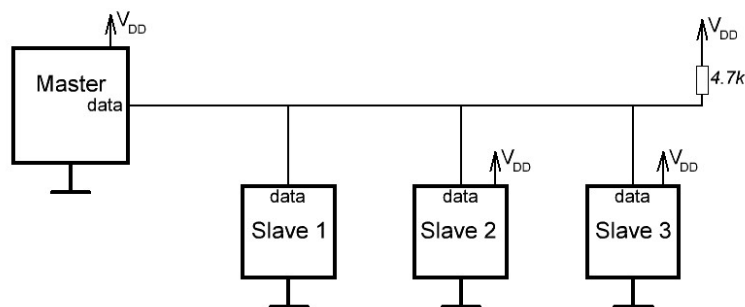


Рис. 1.1. Приклад реалізації технології 1-Wire

### 1.1.2. Принципи дистанційного керування побутовими приладами за технологією *C-bus*

*C-bus* – технологія, розроблена компанією *Clipsal (Clipsal Integrated Systems)* (рис. 1.2). Протокол *C-bus* є відритим стандартом для передачі даних, і тому є можливість доповнювати і поєднувати в проекті технічні пристрої різних виробників. Системи на базі *C-Bus* є децентралізованими.

Програмне забезпечення, що використовується в системі *C-bus*, сумісне з усіма версіями *Windows* і забезпечує управління системами будь-якого рівня.

У всіх блоках (датчиках, вимикачах і т.д.) системи встановлені контролери, причому кожен блок системи оснащений пам'яттю, яка не пошкоджується в разі

збою в подачі електроенергії. Цим пояснюється висока надійність системи *C-bus*, внаслідок чого на базі *C-bus* створені найскладніші і найбільш безпечні системи.

В системі *C-bus* використовується кабель даних *Cat5* (неекранована вита пара). Дані передаються від панелей управління або перемикачів через шину *C-Bus* до відповідних активним блокам-реле.

Системою «Розумний будинок» на базі *C-Bus* можна управляти на відстані, не маючи сервера – і це є однією з головних переваг системи. В цьому випадку команди управління надсилаються з віддаленого комп'ютера або телефону на *IP*-інтерфейс системи.

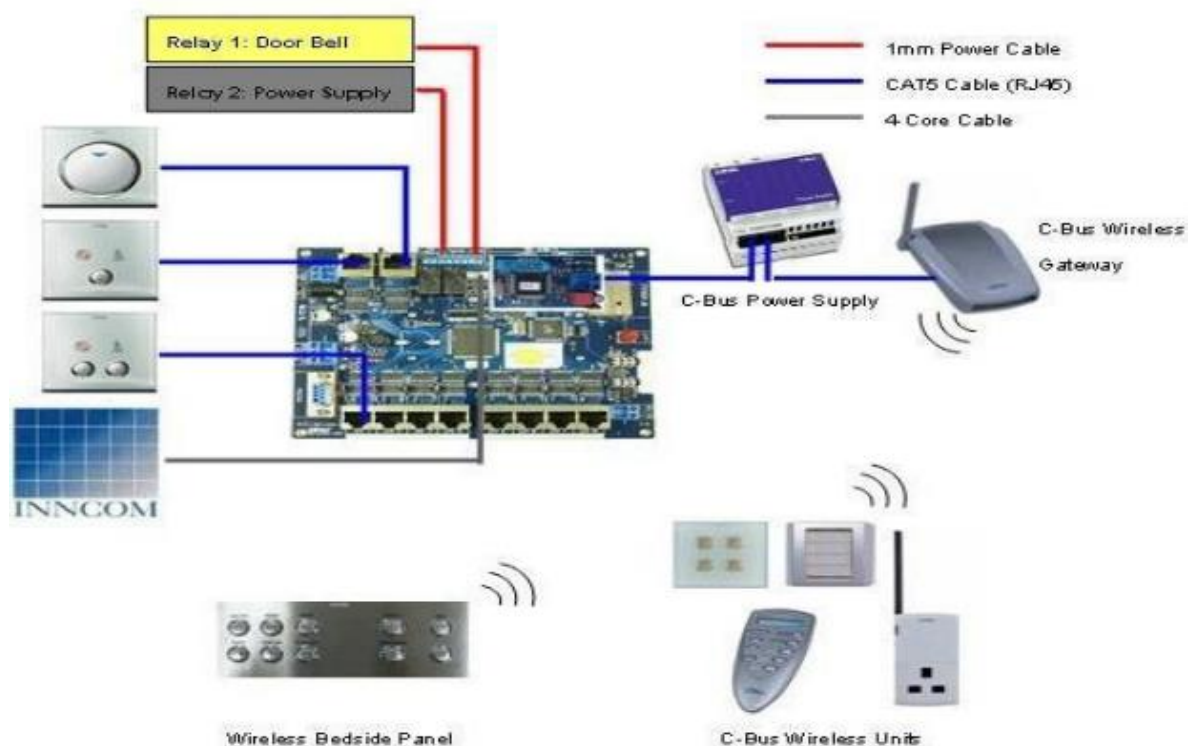


Рис. 1.2. Приклад реалізації технології *C-Bus*

### 1.1.3. Принципи дистанційного керування побутовими приладами за технологією *EIB*

*EIB* (*European Installation Bus*) – технологія, найбільш часто використовувана в Європі (рис. 1.3). У системах *EIB* поряд з силовим кабелем паралельно проводять виту пару, яка є керуючою шиною. До цієї шини підключаються всі технічні пристрої, а через неї підтримується зв'язок між усіма пристроями системи. Всі пристрої з'єднуються один з одним без контролюючих приладів, при цьому будь-яка ієрархія в побудові системи не застосовується.

Передавачі передають потрібну інформацію через шину на всі приймачі відразу. Але реагують на цю інформацію тільки ті приймачі, для яких вона адресована.

На основі *EIB* бувають децентралізовані і централізовані системи. Технологія *EIB* є розподіленою, але незалежно від цього, на її базі можна також створити систему з централізованим управлінням. У цьому випадку керуючий процесор підключається до шини в будь-якому місці. Оскільки системи на базі *EIB* добре захищені від помилок в передачі даних, то вони вважаються досить надійними.

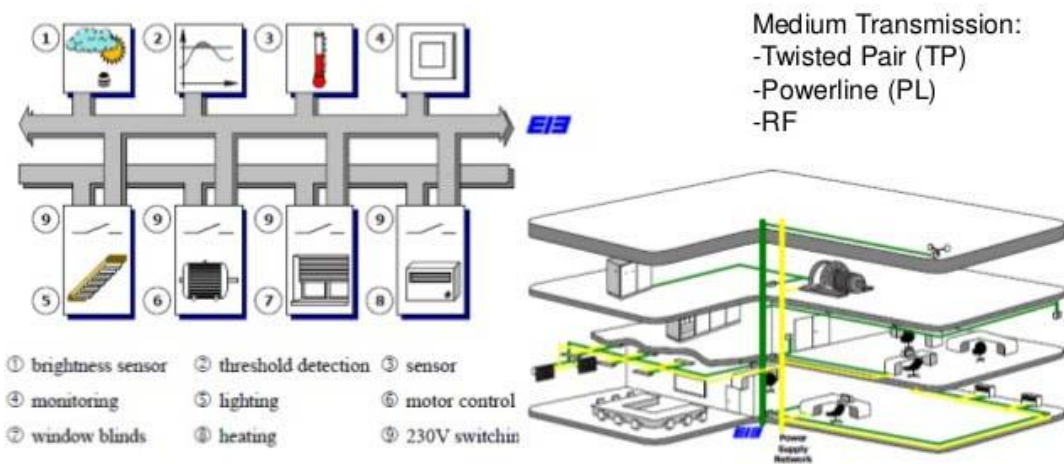


Рис. 1.3. Принцип реалізації технології *EIB*

#### 1.1.4. Принципи дистанційного керування побутовими приладами за технологією *LanDrive*

*LanDrive* – платформа для побудови шинних розподілених систем управління різними електроприладами, зовнішнім (вуличним) і внутрішнім освітленням, аудіо- та відеотехнікою, воротами, ролетами, шторами, домашніми кінотеатрами і т.д. Система «Розумний будинок» на *LanDrive* дозволяє керувати технологічними процесами, інженерними системами, системою охоронної сигналізації, контролю і управління доступом і т.д.

Останнім часом *LanDrive* застосовується в основному в системах обліку і заощадження енергоресурсів, контролю та управління доступом, охоронно-пожежної сигналізації.

Система «Розумний будинок» на *LanDrive* складається з:

1. Центрального контролера;
2. Загальномережевого блоку живлення;
3. Виконавчих модулів – до них підключається кероване обладнання.

#### 1.1.5. Принципи дистанційного керування побутовими приладами за технологією *LonWorks*

*LonWorks* – технологія, розроблена американською компанією *Echelon* і має схожість за своєю мережевою структурою з *EIB*. Технологія *LonWorks* дозволяє створювати більш складні проекти систем автоматизації будівель. Система на базі *LonWorks* будується за допомогою керуючої мережі *LON (Local Operating Network)*, яка має мінімальну кількість рівнів ієрархії. У такій системі відсутній центральний керуючий пристрій. Основою мережі *LonWorks* є мікропроцесор *Neuron*, до складу якого включені три 8-бітових мікропроцесора, вбудована пам'ять, а також 11 контактів введення-виведення. Технологія *LonWorks* дозволяє з'єднати до 127 *LON*-вузлів в кожній підмережі і до 255 підмереж в одному домені.

Як канали передачі інформації можуть служити: кабель типу «вита пара», коаксіальний кабель, радіоефір, електропроводка або інші провідники. В системі на базі *LonWorks* для обміну інформацією використовують контролери, *IP*-сервери, адаптери (*LON, EtherNet* і *PCI*), маршрутизатори і різні датчики.

В даний час технологія *LonWorks* отримала найбільше поширення в США і застосовується в основному для побудови розподілених систем, для яких характерна наявність значної кількості вузлів, віддалених один від одного на певній відстані.

Основні області застосування технології *LonWorks*:

- розподілені мережі управління на транспорті;
- системи автоматизації будівель;
- системи автоматизації технологічних процесів промислових підприємств.

### 1.1.6. Принципи дистанційного керування побутовими приладами за технологією X10

X10 – технологія, найбільш поширена в наш час для побудови «Розумного будинку». Стандарт X10 був розроблений в 1975 році компанією *Pico Electronics* для керування побутовими електроприладами. На базі X10 була вперше створена проста система автоматизації будинку. Ця система функціонує за допомогою звичайної електропроводки (220 В, 50 Гц), по якій передаються інформаційні сигнали. Тому технологія X10 і отримала широке поширення – не потрібна прокладка додаткових кабелів, для побудови системи годиться наявна в будинку електропроводка (рис. 1.4).

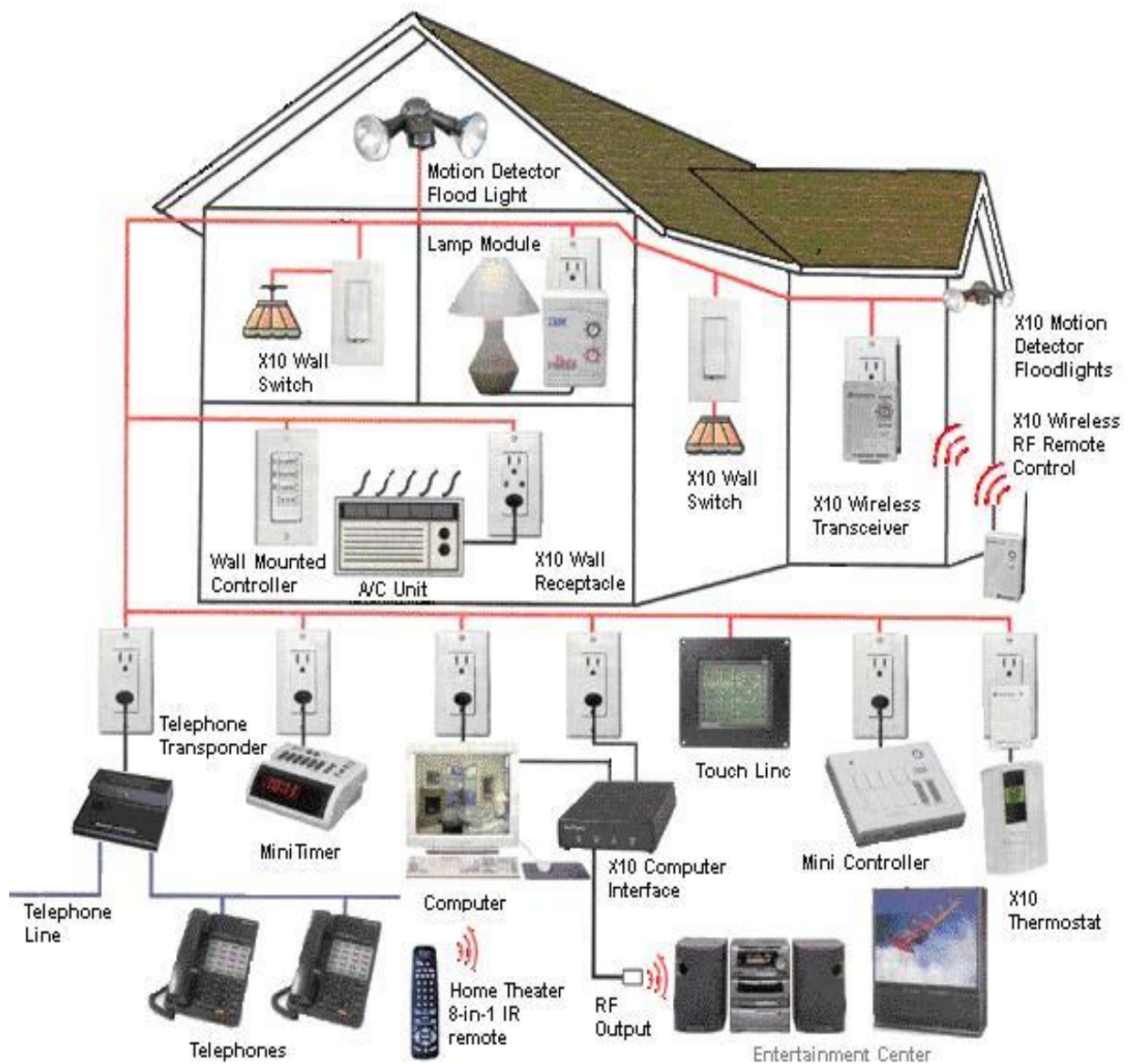


Рис. 1.4. Приклад реалізації технології X-10



X10 може працювати як з центральним процесором, так і без нього.

Системи на базі X10 включають в себе:

– виконавчі пристрої, які приймають команди X10 по електромережі і виконують їх – диммерні модулі, модулі управління приводами, релейні модулі, звукові модулі і т.п.;

– модулі, що передають команди X10 в мережу – сценарні модулі, таймери, командні модулі, контролери типу *Ocelot*, *Leopard II*, *Omni proII*, *Alfa SE* і інші;

– трансивери – модулі, які беруть команди з інфрачервоних пультів дистанційного керування або команди в вигляді радіосигналів з ПДУ. Після надходження команд до трансивера, вони трансформуються і передаються у вигляді X10-команд в мережу.

Систему «Розумний будинок» на X-10 можна будувати поступово за допомогою модулів, технологія X-10 дозволяє нарощувати і змінювати модулі практично постійно. Для початку, досить встановити пульт і один модуль, потім, по мірі необхідності додати ще один і т.д. В даний час системи, побудовані на цій технології, найдешевші.

Оскільки в електромережі можливі перебої, то систему на X-10 не можна вважати достатньо надійною. Тому, дуже складну систему бажано створювати на основі інших технологій.

#### 1.1.7. Принципи дистанційного керування побутовими приладами за технологією *Z-Wave*

*Z-Wave* – повністю бездротова технологія, розроблена компанією *Zensys*. В основі даної технології лежить *mesh*-мережа (чарункова мережа). Кожен технічний пристрій в мережі *Z-Wave* виконує функції приймача і передавача, внаслідок чого підвищується надійність мережі. Якщо, наприклад, вийде з ладу якийсь один пристрій, то сигнал пройде через сусідній. Додаючи нові пристрої, можна легко розширити зону покриття *Z-Wave* мережею. В системі *Z-Wave* не потрібно використовувати додаткові репітери і підсилювачі сигналу. Необхідно тільки, щоб будь-який пристрій в цій системі знаходилося в радіусі дії сусіднього пристрою.

В технології *Z-Wave* використовується чіп (*Single Chip*), який можна інтегрувати в будь-який пристрій. Перевагами чіпа є мале споживання енергії, низька ціна і невеликі розміри. Система на базі *Z-Wave* являє собою *mesh*-мережу з використанням дуплексного бездротового радіозв'язку, і застосовується в житлових і офісних приміщеннях.

В даний час в світі налічується близько 200 виробників, які підтримують технологію *Z-Wave*. До них відносяться такі компанії як *Navicom Technologies*, *Panasonic*, *Lantic Systems*, *Logitech*, *ViewSonic*, *Intel*, *Intermatic*, *Leviton*, *Monster Cable* та ін.

Системи на базі *Z-Wave* можуть включати в себе наступні технічні пристрої:

1. Настінні вимикачі та регулятори яскравості;
2. Патрони для ламп;
3. Розеткові модулі (комутатори і регулятори потужності);
4. Пристрої керування електромеханічними навантаженнями;
5. Термостати і датчики температури;
6. Датчики руху, відкриття дверей або вікон, протікання води, вологості, диму, газу, якості повітря і т.д .;
7. Дверні замки з *Z-Wave* інтерфейсом;
8. Пристрої енергозбереження;
9. Настінні і портативні сенсорні панелі управління;
10. Пульти дистанційного керування;
11. ПК адаптери.

При придбанні пристроїв для системи *Z-Wave* треба враховувати те, що робоча частота у них повинна бути однією і тією ж. Треба також мати на увазі ту обставину, що в різних країнах *Z-Wave* пристрої працюють на різних частотах (рис. 1.5). Наприклад, робоча частота для *Z-Wave* пристроїв, застосовуваних на території України, становить 868,42 МГц (це європейська частота).

На відміну від систем «Розумний будинок» на базі інших технологій, для установки системи *Z-Wave* не потрібно спеціальних кабельних проводок і професійних знань – будь-який бажаючий зможе встановити і налаштувати її.



Рис. 1.5. Приклад реалізації технології *Z-Wave*

З причини своєї доступності і широкого асортименту технологія *Z-Wave* є безперечним лідером для побудови систем домашньої автоматизації [1].

## **1.2. Аналіз аналогів та готових програмних рішень в області дистанційного керування побутовими приладами**

На даний момент у світі існує декілька готових рішень. Проаналізуємо кожне з них.

### **1.2.1. Аналіз системи *REDMOND SkyKettle M173S-E***

Максимальна потужність 2200 *Вт*. Дискретний терморегулятор, що дозволяє плавно налаштовувати температуру нагрівання води в діапазоні від 40 до 100 градусів Цельсія. Функція дистанційного керування через смартфон в реальному часі. Матеріал корпусу – метал, щоб не знижувати планку часу підтримки заданої температури (12 годин).

### **1.2.2. Аналіз системи *MIE Smart Kettle***

Розумний чайник *MIE Smart Kettle* з 13-позиційним терморегулятором від 40 до 100 градусів Цельсія і функцією заварювання чаю від 1 до 20 хвилин. Обладнання: дисплей, таймер, режим підтримки тепла, індикатор включення разом зі знімною кришкою. Відсутня функція дистанційного керування. Як матеріал корпусу тут виступає поєднання металу і скла, а нагрівальним

елементом служить закрита спіраль з нержавіючої сталі. Максимальна потужність *MIE Smart Kettle* становить 1200 Вт, яких впевнено вистачає для нагрівання 1,7 літрів води.

### 1.2.3. Аналіз системи *Xiaomi Mijia Multifunctional Electric Cooker*

Потужність даного розумного чайника складає 1800 Вт, а місткість – 1,5 літра. 24 запрограмованих режими приготування чаю. У повній відповідності із заявленими функціями *Xiaomi Mijia Multifunctional Electric Cooker* представляє можливість регулювання потужності, які впливають на швидкість і температуру нагрівання, а також функція дистанційного керування через смартфон. В наявності блокування включення без води, всі необхідні індикатори, дисплей та знімна кришка.

### 1.2.4. Аналіз системи *E.Zicom Aromatea*

Незважаючи на низьке енергоспоживання (потужність моделі складає всього 1000 Вт) він досить швидко нагріває до 1,7 літрів води, даючи можливість підтримувати температуру протягом 30-40 хвилин. Це обумовлено застосуванням комбінації скла і металу при виготовленні корпусу. Завдяки ступінчастому терморегулятору *E.Zicom Aromatea* забезпечує нагрів води від 37 до 100 градусів Цельсія, з можливістю плавного налаштування необхідної температури за допомогою дисплея. З недоліків відсутність функцій безпеки.

### 1.2.5. Аналіз системи *Element El'kettle WF11MB / MW*

Потужність даного розумного чайника складає 2200 Вт. Модель оснащена ступінчастим терморегулятором з можливістю підігріву води до температури від 60 до 100 градусів Цельсія. Корпус виконаний з цілісного металу, а всі елементи фільтрації і нагріву – з нержавіючої сталі. Перевагою даного розумного чайника над конкурентами стала наявність звукової індикації всіх виконуваних дій, це додає незаперечний "плюс" безпеки. Обладнання: індикатори рівня води, наявність підсвічування при включенні чайника в роботу, відображення поточної температури і функцію збереження тепла до 30 хвилин [2].

### 1.2.6. Аналіз системи *Polaris PWK 1725CGLD WIFI IQ Home*

Розумний чайник з *WI-FI* управлінням. Легко підключається і керується через додаток *Polaris IQ Home* з будь-якої точки світу, де є інтернет. Безкоштовний додаток можна скачати в *AppStore* і *GooglePlay*. На ручці чайника можна вибрати один з п'яти попередньо встановлених режимів розігріву води: 50, 70, 80, 90 і 100 градусів Цельсія. В додатку міститься 11 попередньо встановлених режимів для різного виду чаю, а також можливість вибрати будь-яку температуру нагрівання від 30 до 100 градусів, рецепти приготування чаю з фотографіями, розігрів з утриманням і налаштування роботи за розкладом. Присутня функція розумного кип'ятіння – ступінчастий нагрів води до закипання, виключаючи бурхливе кипіння [3].

Після аналізу аналогів та готових програмних рішень можна побачити, що готових рішень з дистанційним керуванням не так багато, як очікувалось, і всі вони мають певні недоліки.

## **1.3. Методи та технології дистанційного керування побутовими приладами**

### 1.3.1. Особливості використання технології *Bluetooth* для дистанційної передачі даних

*Bluetooth* – специфікація бездротових персональних мереж (*Wireless personal area network, WPAN*). *Bluetooth* забезпечує обмін інформацією між такими пристроями, як персональні комп'ютери (настільні, кишенькові, ноутбуки), мобільні телефони, інтернет-планшети, принтери, цифрові фотоапарати, миші, клавіатури, джойстики, навушники, гарнітури і акустичні системи на надійній, безкоштовній, повсюдно доступній радіочастоті для ближнього зв'язку. *Bluetooth* дозволяє цим пристроям зв'язуватися, коли вони знаходяться один від одного в радіусі близько ста метрів старих версій протоколу і до тисячі п'ятсот метрів починаючи з версії *Bluetooth 5*. Дальність сильно залежить від перешкод, навіть в одному приміщенні [4].

Роботи зі створення *Bluetooth* почав виробник телекомунікаційного устаткування *Ericsson* в 1994 році як бездротову альтернативу кабелям *RS-232*. Спочатку ця технологія була пристосована під потреби системи *FLYWAY* в функціональному інтерфейсі між мандрівниками і системою.

Специфікація *Bluetooth* була розроблена групою *Bluetooth Special Interest Group (Bluetooth SIG)*, яка була заснована в 1998 році. До неї увійшли компанії *Ericsson, IBM, Intel, Toshiba* і *Nokia*. Згодом *Bluetooth SIG* і *IEEE* досягли угоди, на основі якої специфікація *Bluetooth* стала частиною стандарту *IEEE 802.15.1* (дата опублікування – 14 червня 2002 року).

Принцип дії заснований на використанні радіохвиль. Радіозв'язок *Bluetooth* здійснюється в *ISM*-діапазоні (*Industry, Science and Medicine*), який використовується в різних побутових приладах і бездротових мережах. Частоти *Bluetooth*: 2.402-2.48 ГГц [5]. У *Bluetooth* застосовується метод розширення спектра зі стрибкоподібною перебудовою частоти (*Frequency Hopping Spread Spectrum, FHSS*). Метод *FHSS* простий в реалізації, забезпечує стійкість до широкосмугових перешкод, а обладнання недороге.

Відповідно до алгоритму *FHSS*, в *Bluetooth* несуча частота сигналу стрибкоподібно змінюється 1600 разів в секунду. Послідовність перемикання між частотами для кожного з'єднання є псевдовипадковою і відома тільки передавачу і приймачу, які кожні 625 мікросекунд синхронно перебудовуються з однієї несучої частоти на іншу. Таким чином, якщо поруч працюють кілька пар приймач-передавач, то вони не заважають один одному. Цей алгоритм є також складовою частиною системи захисту конфіденційності інформації, що передається: перехід відбувається по псевдовипадковому алгоритму і визначається окремо для кожного з'єднання. При передачі цифрових даних і аудіосигналу використовуються різні схеми кодування: аудіосигнал не повторюється, а цифрові дані в разі втрати пакета інформації будуть передані повторно.

Протокол *Bluetooth* підтримує не тільки з'єднання «*point-to-point*», а й «*point-to-multipoint*» [6].

### 1.3.2. Особливості використання технології *Wi-Fi* дистанційної передачі даних

*Wi-Fi* (*Wireless Fidelity*) – стандарт *IEEE* 802.11 передачі цифрових потоків даних по радіоканалах. Обладнання, що відповідає стандарту *IEEE* 802.11, може бути протестовано *Wi-Fi Alliance*, щоб отримати відповідний сертифікат і право нанесення логотипу *Wi-Fi*. Поширеним на сьогодні є протокол *IEEE* 802.11n.

Встановлення *Wireless LAN* доцільне для побудови мереж, де розгортання кабельної системи є неможливим або економічно недоцільним. Поточні реалізації *Wi-Fi* дозволяють отримати швидкість передачі даних понад 100 *Mbit/s*, при цьому користувачі можуть переміщуватися між точками доступу на території покриття мережі *Wi-Fi*, використовуючи мобільні пристрої, оснащені клієнтськими приймально-передавальними пристроями *Wi-Fi* та отримувати доступ в Інтернет.

Дальність передавання інформації залежить від потужності передавача (яка в окремих моделях обладнання регулюється програмно), наявності та характеристики перешкод, типу антени.

Окрім 802.11b, ще є бездротовий стандарт 802.11a, який використовує частоту 5 *ГГц* та забезпечує максимальну швидкість 54 *Mbit/s*, а також 802.11g, що працює на частоті 2,4 *ГГц* і також забезпечує 54 *Mbit/s*.

Стандарт 802.11n, забезпечує до 320 *Mbit/s*. На сьогоднішній день даний стандарт широко застосовується.

Ядром бездротової мережі *Wi-Fi* є так звана точка доступу (*Access Point*), яка підключається до якоїсь наземної мережевої інфраструктури (каналів Інтернет-провайдера) та забезпечує передачу радіосигналу. Зазвичай точка доступу складається із приймача, передавача, інтерфейсу для підключення до дротової мережі та програмного забезпечення для налаштування. Навколо точки доступу формується просторова область радіусом 50-100 метрів (її називають хот-спотом або зоною *Wi-Fi*), у межах якої можна користуватися бездротовою мережею.

Зазвичай, схема мережі *Wi-Fi* містить не менш однієї точки доступу та може легко масштабуватись. Також можливо підключення двох клієнтів в

режимі точка-точка, коли точка доступу не використовується, а клієнти з'єднуються за участю мережевих адаптерів «напрямую». Точка доступу передає свій ідентифікатор мережі (*SSID*) з допомогою спеціальних сигнальних пакетів на швидкості 0,1 Мбіт/с кожні 100 мс. Тому 0,1 Мбіт/с – найменша швидкість передачі даних для *Wi-Fi*. Знаючи *SSID* мережі, клієнт може з'ясувати, чи можливо підключення до даної точки доступу. При потраплянні в зону дії двох точок доступу з ідентичними *SSID* приймач може вибирати між ними на основі даних про рівень сигналу. Стандарт *Wi-Fi* дає клієнту повну свободу при виборі критеріїв для з'єднання.

Однак стандарт не описує всі аспекти побудови безпроводних локальних мереж *Wi-Fi*. Тому кожен виробник устаткування вирішує цю задачу по-своєму, застосовуючи ті підходи, які він вважає за якнайкращі з тієї або іншої точки зору. Тому виникає необхідність класифікації способів побудови безпроводних локальних мереж.

За способом об'єднання точок доступу в єдину систему можна виділити:

1. Автономні точки доступу;
2. Точки доступу, що працюють під управлінням контролера;
3. Безконтролерні, але не автономні (керовані без контролера).

За способом організації і управління радіоканалами можна виділити безпроводні локальні мережі:

1. Із статичними налаштуваннями радіоканалів;
2. З динамічними (адаптивними) налаштуваннями радіоканалів;
3. З «шаруватою» або багат шаровою структурою радіоканалів.

До основних переваг технології *Wi-Fi* можна віднести:

– дозволяє розвернути мережу без прокладки кабелю, що може зменшити вартість розгортання або розширення мережі. Місця, де не можна прокласти кабель, наприклад, поза приміщеннями і в будівлях, що мають історичну цінність, можуть обслуговуватися безпроводними мережами;

– дозволяє мати доступ до мережі мобільним пристроям;

– *Wi-Fi*-пристрої широко поширені на ринку. Гарантується сумісність устаткування завдяки обов'язковій сертифікації устаткування з логотипом *Wi-Fi*;



– випромінювання від *Wi-Fi*-пристроїв у момент передачі даних на два порядки (у 100 разів) менше, ніж біля стільникового телефону;

– *Wi-Fi* – це набір глобальних стандартів. На відміну від стільникових телефонів, *Wi-Fi*-устаткування може працювати в різних країнах по всьому світу.

До основних недоліків технології *Wi-Fi* можна віднести:

– невелика ширина використовуваного спектра частот, відсутність можливостей роумінгу й авторизації не дозволяють *Wi-Fi*-пристроєм потіснити на ринку мобільний зв'язок. Проте компанії *ZyXEL*, *SocketIP* і *Symbol Technologies* пропонують рішення з організації *Wi-Fi*-телефонії;

– частотний діапазон і експлуатаційні обмеження в різних країнах неоднакові. У багатьох європейських країнах дозволено два додаткові канали, які заборонені в США; у Японії є ще один канал у верхній частці діапазону, а інші країни, наприклад Іспанія, забороняють використання низькочастотних каналів. Більш того, деякі країни, наприклад Росія, Білорусь і Італія, вимагають реєстрації всіх мереж *Wi-Fi* приміщень, що працюють зовні, або вимагають реєстрації *Wi-Fi*-оператора;

– найпопулярніший стандарт шифрування *WEP* може бути відносно легко зламаний навіть при правильній конфігурації (через слабку стійкість алгоритму). Не зважаючи на те, що нові пристрої підтримують досконаліший протокол шифрування даних *WPA* і *WPA2*, багато старих точок доступу не підтримують його і вимагають заміни. Ухвалення стандарту *IEEE 802.11i (WPA2)* в червні 2004 року зробило доступною безпечнішу схему, яка доступна в новому устаткуванні. Обидві схеми вимагають стійкіший пароль, ніж ті, які зазвичай призначаються користувачами. Багато організацій використовують додаткове шифрування (наприклад *VPN*) для захисту від вторгнення [7].

#### **1.4. Постановка завдання проектування**

Актуальність дипломного проекту обумовлена тим, що система дистанційного керування роботою побутових приладів є ефективним та зручним

засобом підвищення рівня комфорту і життя, так як частина процесів відбувається автоматично, а рештою можна керувати віддалено.

Мета – розробити програмний модуль та апаратний модуль системи дистанційного керування роботою побутових приладів.

Відповідно до мети проекту поставлено наступні завдання:

- проаналізувати сучасні програмні рішення та методи реалізації систем дистанційного керування;

- розробити структуру системи дистанційного керування роботою побутових приладів;

- реалізувати алгоритм системи дистанційного керування роботою побутових приладів;

- реалізувати програмний модуль та апаратний модуль системи дистанційного керування роботою побутових приладів.

## **1.5. Висновки до розділу**

В межах даного розділу було проаналізовано:

- сучасні технології та методи реалізації систем дистанційного керування побутовими приладами. Для аналізу обрано такі відомі рішення як: *1-Wire, C-Bus, EIB/KNX, LanDrive, LonWorks, X10 та Z-Wave*;

- аналоги та готові програмні рішення в області дистанційного керування побутовими приладами. Для аналізу обрано такі відомі рішення як: *REDMOND SkyKettle M173S-E, MIE Smart Kettle, Xiaomi Mijia Multifunctional Electric Cooker, E.Zicom Aromatea, Element El'kettle WF11MB / MW, Polaris PWK 1725CGLD WIFI IQ Home*;

- технології створення апаратних модулів дистанційного керування побутовими приладами. Для реалізації даного проекту обрано технологію *Bluetooth*, оскільки вона підтримує досить високу швидкість передачі даних і досить дешева.

РОЗДІЛ 2  
ПРИНЦИПИ РОБОТИ СИСТЕМИ ДИСТАНЦІЙНОГО КЕРУВАННЯ  
ПОБУТОВИМИ ПРИЛАДАМИ

**2.1. Функціональні вимоги до системи дистанційного керування побутовими приладами**

Функціональні вимоги (*Functional Requirements*) – це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система. На відміну від нефункціональних вимог, які визначають якою система повинна бути, функціональні вимоги визначають, що система повинна робити. Функціональні вимоги до програмного забезпечення визначаються на першій стадії процесу розробки ПЗ – на етапі аналізу вимог.

Базові функціональні можливості системи дистанційного керування:

- забезпечити коректну роботу пристрою та повноцінне відображення інтерфейсу програмного модуля для зручності використання користувачем та виконання належного функціоналу;
- два режими керування: за допомогою *SMS* повідомлень та *Android* додатку для управління по *Bluetooth*;
- повноцінне функціонування всіх складових апаратного модуля;
- своєчасне сповіщення користувача в разі виконання заданих команд;
- вимкнення пристрою в разі нагріву води;
- програмний модуль на ОС *Android*.

Кафедра КСУ				НАУ 21 06 34 000 ПЗ				
<i>Виконав</i>	<i>Кіслов О.І.</i>			Принципи роботи системи дистанційного керування побутовими приладами	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>		27	56
<i>Консульт.</i>					123 СП 437			
<i>Норм. контр.</i>	<i>Тупота Є.В.</i>							
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>							

## 2.2. Структура системи дистанційного керування побутовими приладами

Система дистанційного керування складається з апаратної та програмної частин.

Апаратна частина створена на базі платформи *Arduino NANO* мовою програмування *Arduino*. Також для створення апаратної частини необхідні: *GSM*-модуль *SIM800L*, *Bluetooth HC-05*, датчик температури *Dallas DS18B20* в захищеному корпусі, 1-канальний реле модуль та блок живлення *12V*.

Програмна частина створена у середовищі розробки *Android Studio*, за допомогою об'єктно-орієнтованого методу програмування та мови програмування *Java*.

Структурна схема – схема, на якій показано основні функційні частини виробу, їх взаємозв'язки та призначеність для отримання загальної уяви про виріб. Під функціональною частиною розуміють складову частину схеми: елемент, пристрій, функціональну групу, функціональну ланку [8].

Структурна схема призначена для відображення загальної структури пристрою, тобто його основних блоків, вузлів, частин та головних зв'язків між ними. Із структурної схеми повинно бути зрозуміло, навіщо потрібний даний пристрій і як він працює в основних режимах роботи, як взаємодіють його частини (рис. 2.1).

При проектуванні комплексу розроблено діаграму прецедентів (рис. 2.2). Діаграма відображає відношення між прецедентами та акторами, а саме: користувачем – *User*, мобільним додатком – *Android app* та пристроєм – *Device*.

Клієнт може надсилати *SMS* команди (*Sending SMS commands*) або обрати пристрій (*Select device*) за допомогою андроїд додатку (*Android app*) та керувати пристроєм за допомогою *Bluetooth* команд.

Пристрій здійснює ввімкнення (*Turning on the device*), контроль за температурою (*Temperature level control*), вимкнення (*Turning off the device*) та відправку *SMS* повідомлення (*Sending SMS notifications*) користувачу.

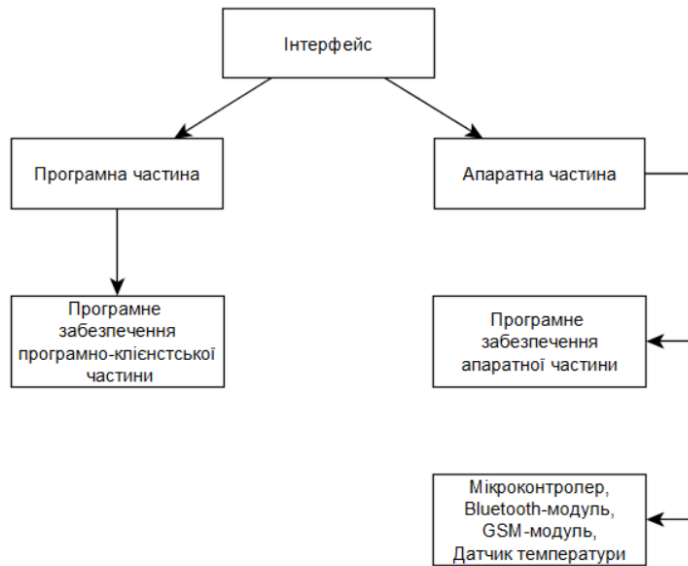


Рис. 2.1. Принцип побудови системи дистанційного керування роботою чайника

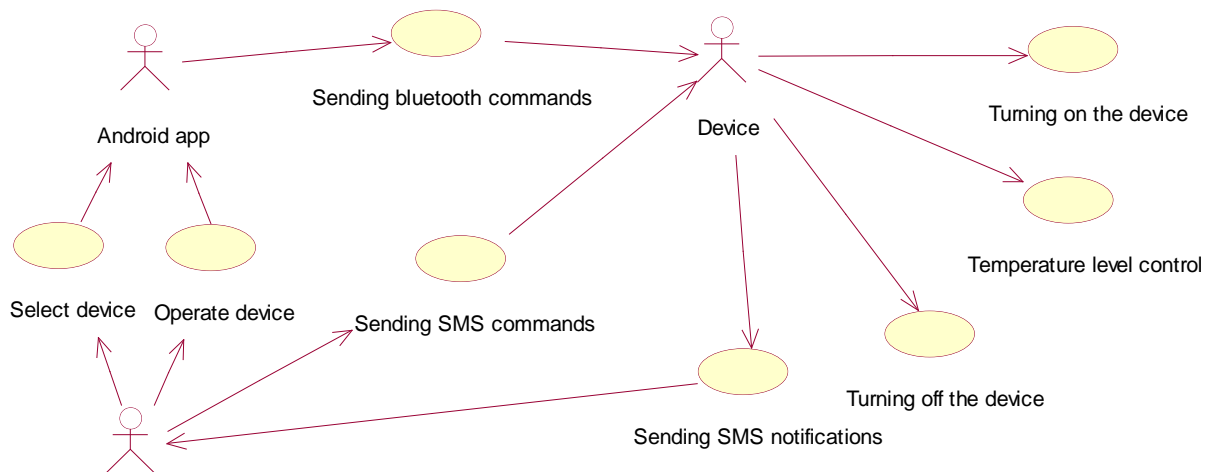


Рис. 2.2. Діаграма прецедентів

При проектуванні модуля розроблено діаграми послідовності у яких показано керування пристроєм за допомогою андроїд додатку (рис. 2.3) та за допомогою *SMS* команд (рис. 2.4).

При керуванні пристроєм за допомогою андроїд додатку користувач обирає пристрій до якого підключається по *Bluetooth* (*Select device*), пристрій успішно з'єднується з додатком (*Successful connecting to device*). Користувач вмикає пристрій за допомогою кнопки у додатку (*Turn on the device*). Андроїд додаток відправляє пристрою команду ввімкнення ("*Turn on*" *command*). Пристрій вмикається (*Turn on the device*), слідкує за рівнем температури (*Temperature level control*) та вимикається (*Turn off the device*), після чого надсилає користувачу *SMS* повідомлення (*SMS notification "Your water boiled!"*).

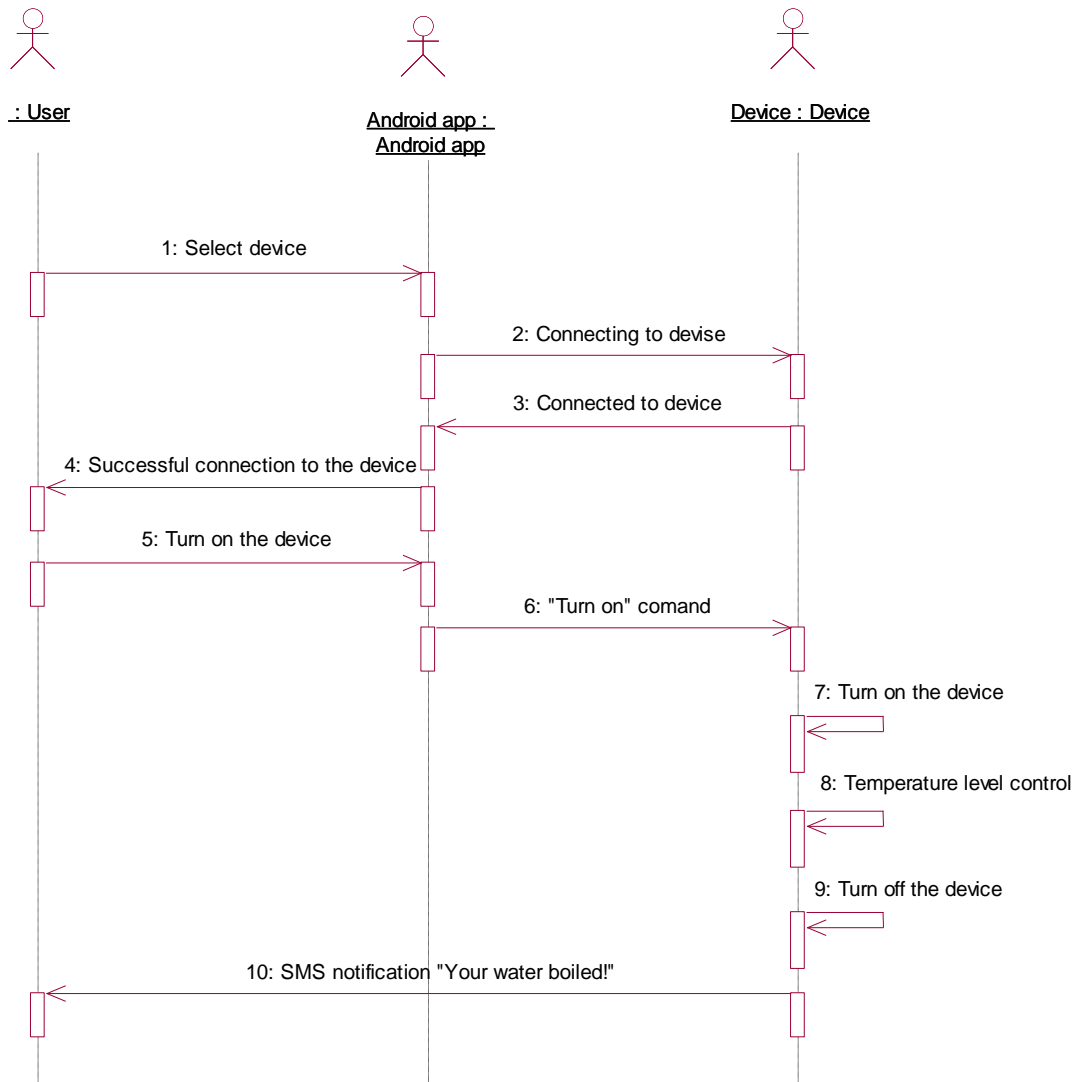


Рис. 2.3. Діаграма послідовності для керування пристроєм через додаток

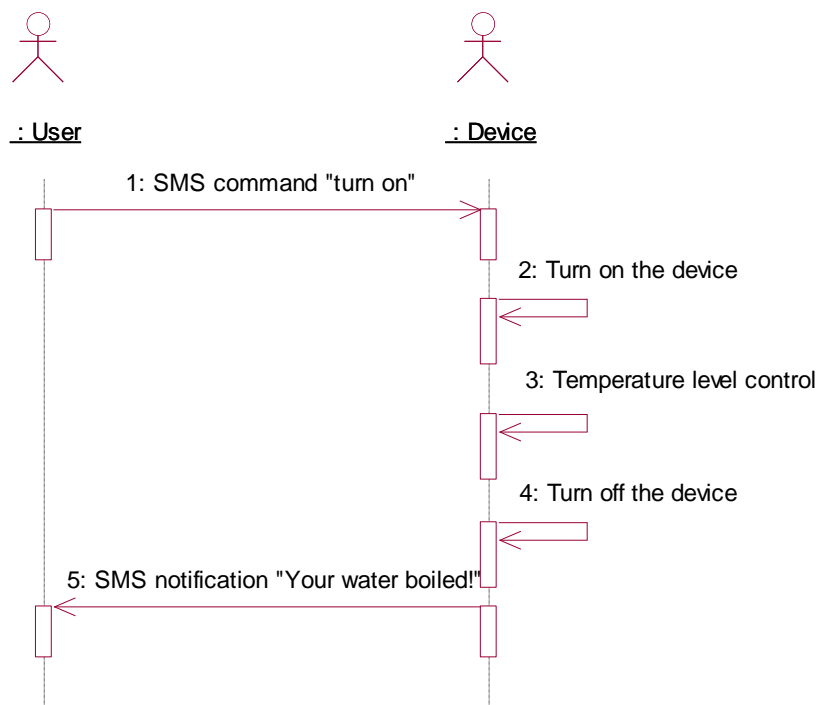


Рис. 2.4. Діаграма послідовності для керування пристроєм через SMS

При керуванні пристроєм за допомогою *SMS* команд користувач надсилає пристрою *SMS* з командою ввімкнення (*SMS command “turn on”*). Пристрій вмикається (*Turn on the device*), слідкує за рівнем температури (*Temperature level control*) та вимикається (*Turn off the device*), після чого надсилає користувачу *SMS* повідомлення (*SMS notification “Your water boiled!”*).

При проектуванні модуля розроблено діаграму взаємодії (рис. 2.5) у якій показано співпрацю користувача – *User*, мобільного додатку – *Android app* та пристрою – *Device*.

Користувач обирає пристрій до якого підключається по *Bluetooth* (*Select device*), додаток з’єднується з пристроєм (*Connect to device*), а пристрій у свою чергу з’єднується з додатком (*Connect to android app*). Користувач вмикає пристрій (*Turn on the device*) за допомогою кнопки у додатку, який відправляє пристрою команду ввімкнення (*Give “Turn on” comand to the device*). Пристрій вмикається (*Turn on the device*), слідкує за рівнем температури (*Temperature level control*) та вимикається (*Turn off the device*), після чого надсилає користувачу *SMS* повідомлення (*Send SMS notification*). Користувач отримує *SMS* повідомлення (*Receive SMS notificaton*).

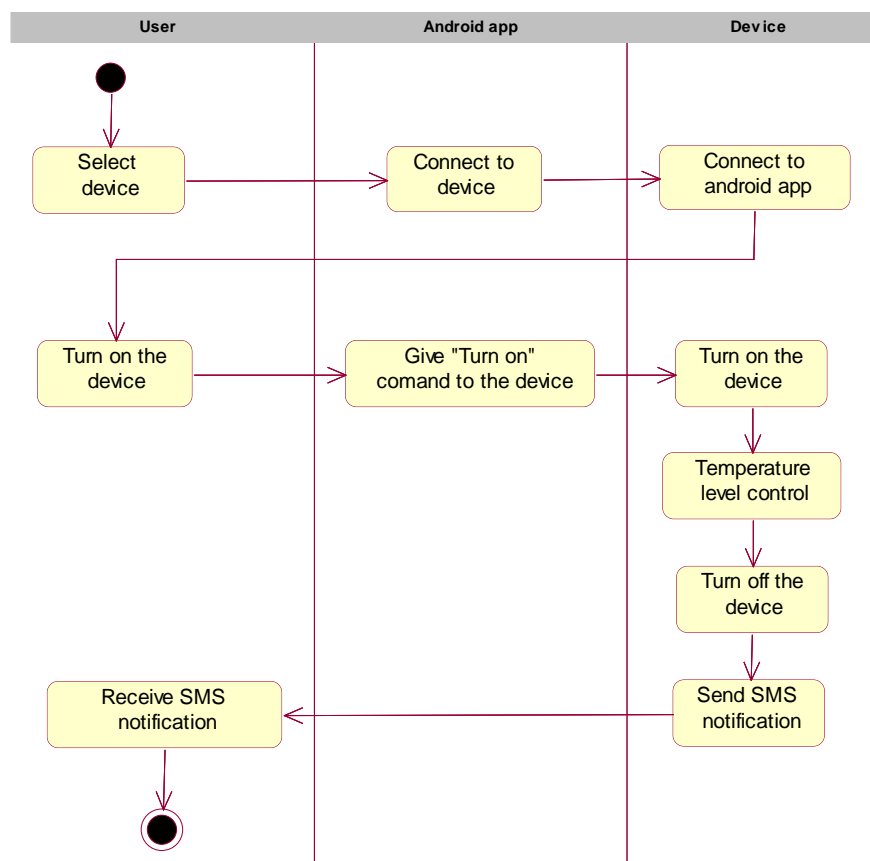


Рис. 2.5. Діаграма взаємодії

При проектуванні модуля розроблено функціональну схему програмного модуля (рис. 2.6).

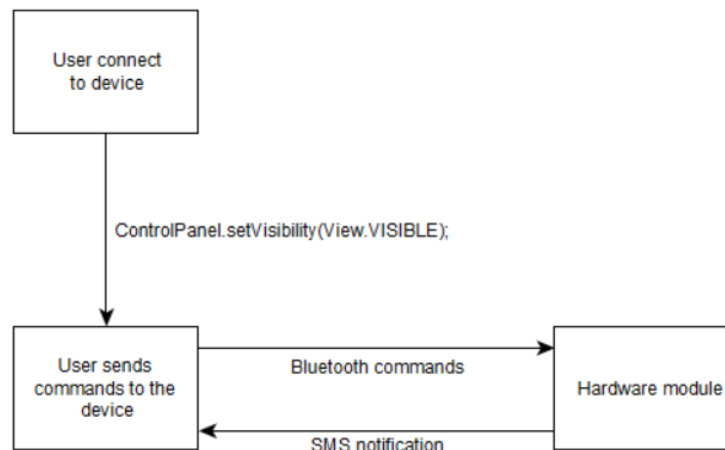


Рис. 2.6. Функціональна схема системи

### 2.3. Функціональне забезпечення системи дистанційного керування побутовими приладами

Апаратний модуль має таке функціональне забезпечення:

1. Ввімкнення та вимкнення пристрою;
2. Контроль рівню температури;
3. *SMS* повідомлення про закінчення роботи.

Програмний модуль має таке функціональне забезпечення:

1. Підключення до апаратного модулю;
2. Дистанційне керування пристроєм.

Зв'язок між модулями реалізовано за допомогою технології *Bluetooth* та стандарту *GSM*.

Мобільний додаток працює на пристроях під управлінням системи *Android*. Мінімальна версія системи – 4.1. Остання версія, на якій проводилося тестування додатку – *Android 11*. Відповідно до інформації від розробника *Android* компанії *Google*, це покриває приблизно 99,8% активних пристроїв під управлінням даної операційної системи, або іншими словами, пристрої випущені, починаючи з 2012-2013 років.



## 2.4. Висновки до розділу

В даному розділі було описано можливості систем дистанційного керування та Розроблено структуру системи дистанційного керування побутовими приладами, яка складається з апаратної та програмної частин.

Розроблена структура передбачає, що апаратна частина містить мікроконтролер, *Bluetooth*-модуль, *gsm*-модуль, датчик температури та програмне забезпечення для мікроконтролера, програмна частина – інтерфейс та функціонал мобільного додатку.

Також в розділі наведено діаграми прецедентів, послідовності та взаємодії роботи основних частин системи дистанційного керування побутовими приладами.

Розроблена функціональна схема роботи програмного модуля, відображає роботу основних компонентів ПЗ системи (підключення до апаратного модуля та надсилання *Bluetooth* команд).

РОЗДІЛ 3  
ОПИСАННЯ ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

**3.1. Алгоритм роботи системи дистанційного керування побутовими приладами**

Під час проектування системи розроблено схему алгоритму роботи системи дистанційного керування роботою побутових приладів (рис. 3.1). На якій показано основні процеси, яку відбуваються у системі, а саме: вибір типу керування та робота апаратної частини.

До основних команд, які побутові пристрої повинні отримувати та передавати на мобільний додаток, можна віднести:

- включення / вимикання зі смартфона;
- установка та редагування налаштувань;
- попередження і захист від включення при недопустимих параметрах;
- установка часу автовключення;
- повідомлення про готовність.

Так для роботи з чайником основний функціонал можна описати наступним чином:

- включення / вимикання зі смартфона;
- установка будь-якої температури нагріву води;
- можливість дізнатися поточну температуру;
- відстеження поточного обсягу води в чайнику;
- попередження і захист від включення при малому обсязі води;
- установка часу автовимикання;
- повідомлення про готовність.

Кафедра КСУ				НАУ 21 06 34 000 ПЗ			
<i>Виконав</i>	<i>Кіслов О.І.</i>			Описання прийнятих проектних рішень	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	<i>34</i>	<i>56</i>
<i>Консульт.</i>					123 СП 437		
<i>Норм. контр.</i>	<i>Тупота Є.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

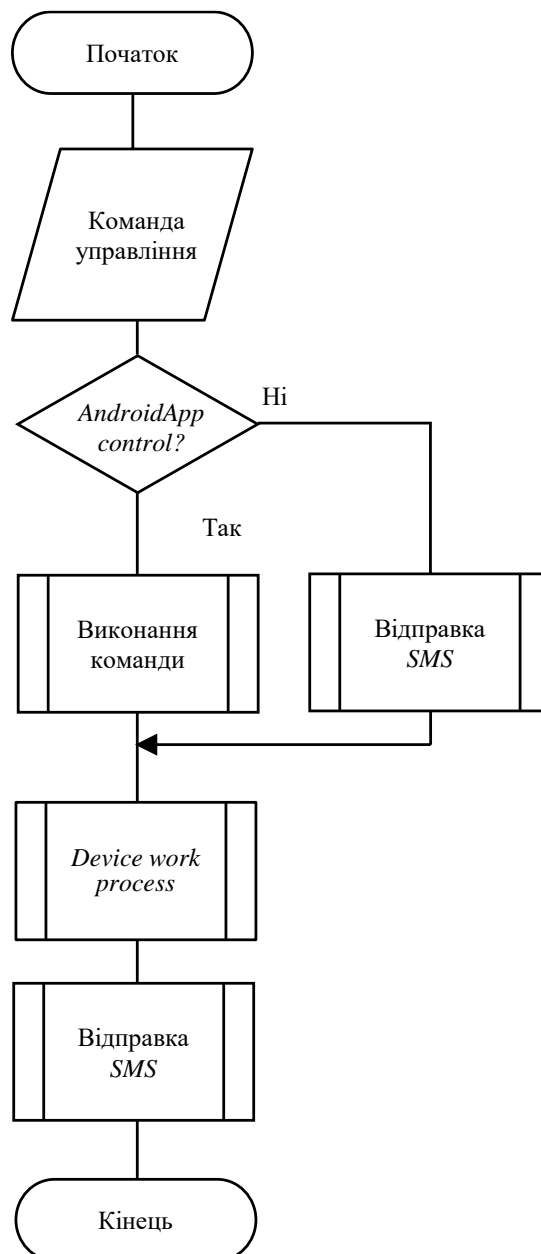


Рис. 3.1. Схема алгоритму роботи системи

У процесі «*Device work process*» відбувається ввімкнення пристрою, контроль рівню температури, вимкнення пристрою та відправлення користувачу *SMS* повідомлення. Схема алгоритму роботи «*Device work process*» представлено на рис. 3.2.

Для реалізації автоматичного вимкнення було використано водонепроникний термодатчик (*NTC* термістор), який витримує температуру більше 100 градусів, а також застосовано годинник реального часу для системного опитування датчику. Для установки часу і отримання інформації про цей датчик було використано бібліотеку *DS1307*.

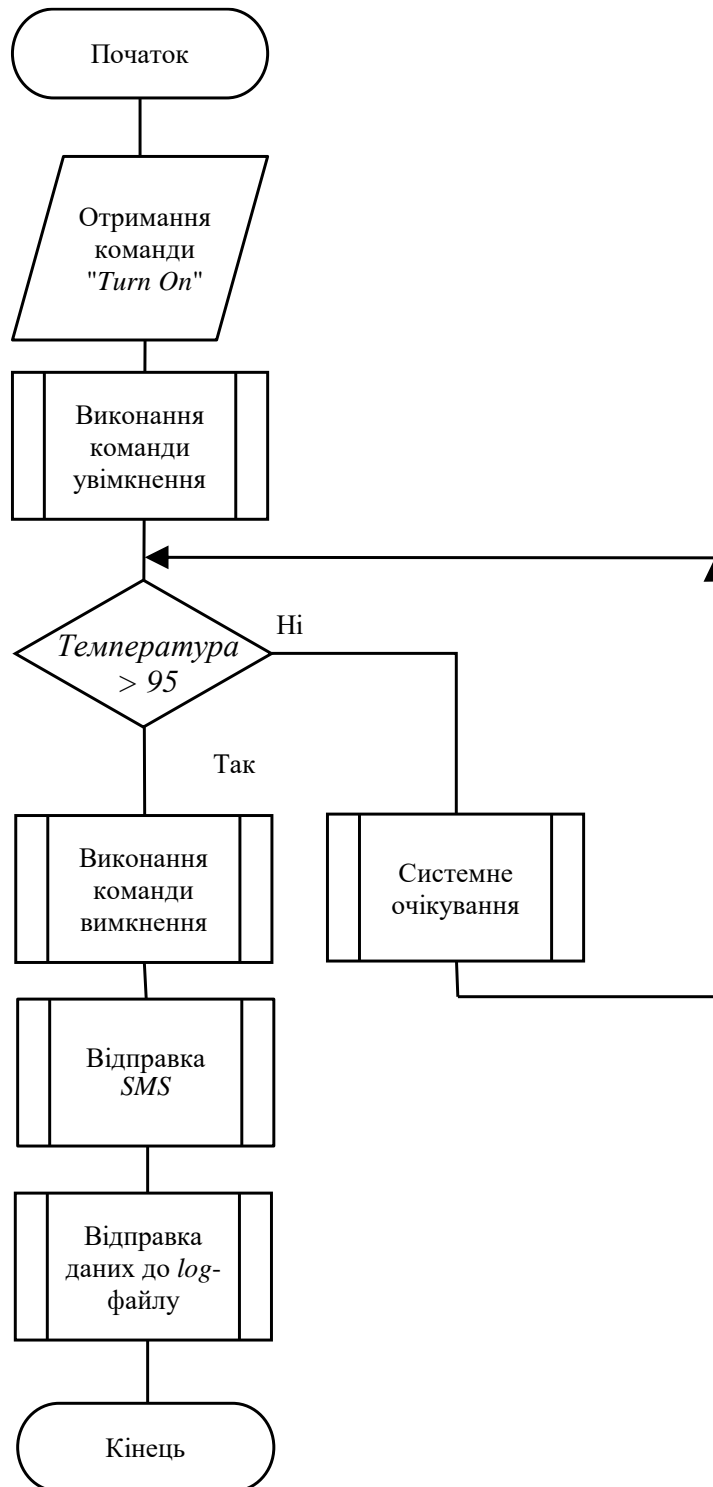


Рис. 3.2. Схема алгоритму процедури *Device work process*

### 3.2. Розробка апаратного модуля

На першому етапі проектування проведено збір усіх деталей, необхідних для виконання поставленого завдання (рис. 3.3).

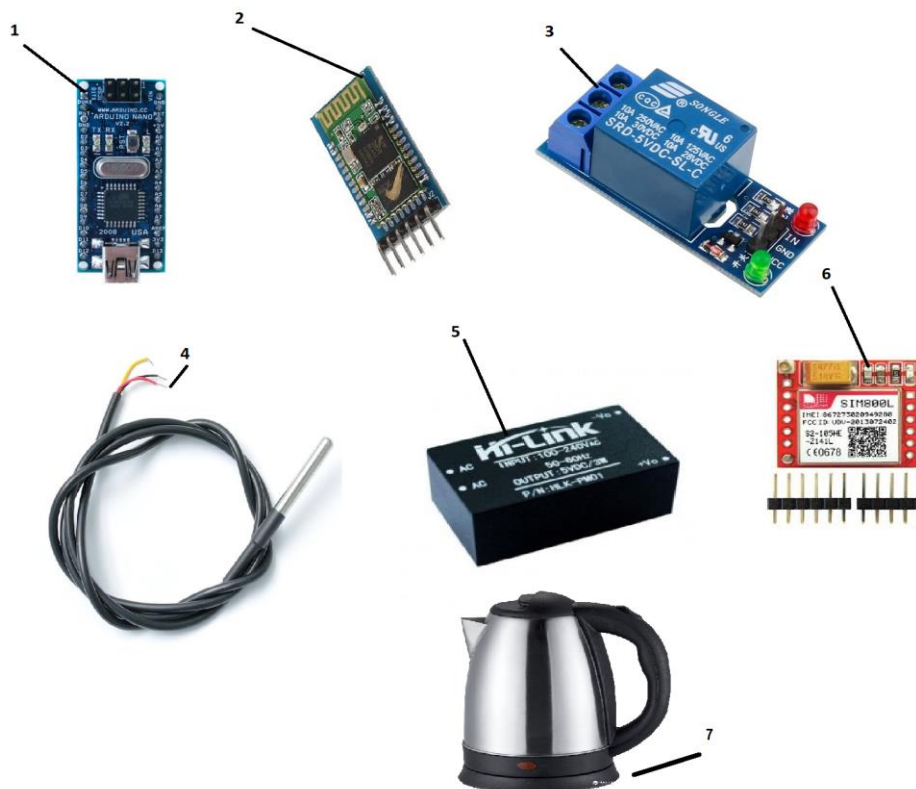


Рис 3.3. Необхідні деталі

Перелік необхідних деталей:

- плата *Arduino NANO*;
- *Bluetooth HC-05*;
- реле;
- датчик температури *Dallas DS18B20 / 18B20* в захищеному корпусі;
- блок живлення 5В;
- *GSM*-модуль *SIM800L*;
- найпростіший електрочайник.

Наступним етапом є проектування схеми підключення всіх складових пристрою (рис 3.4).

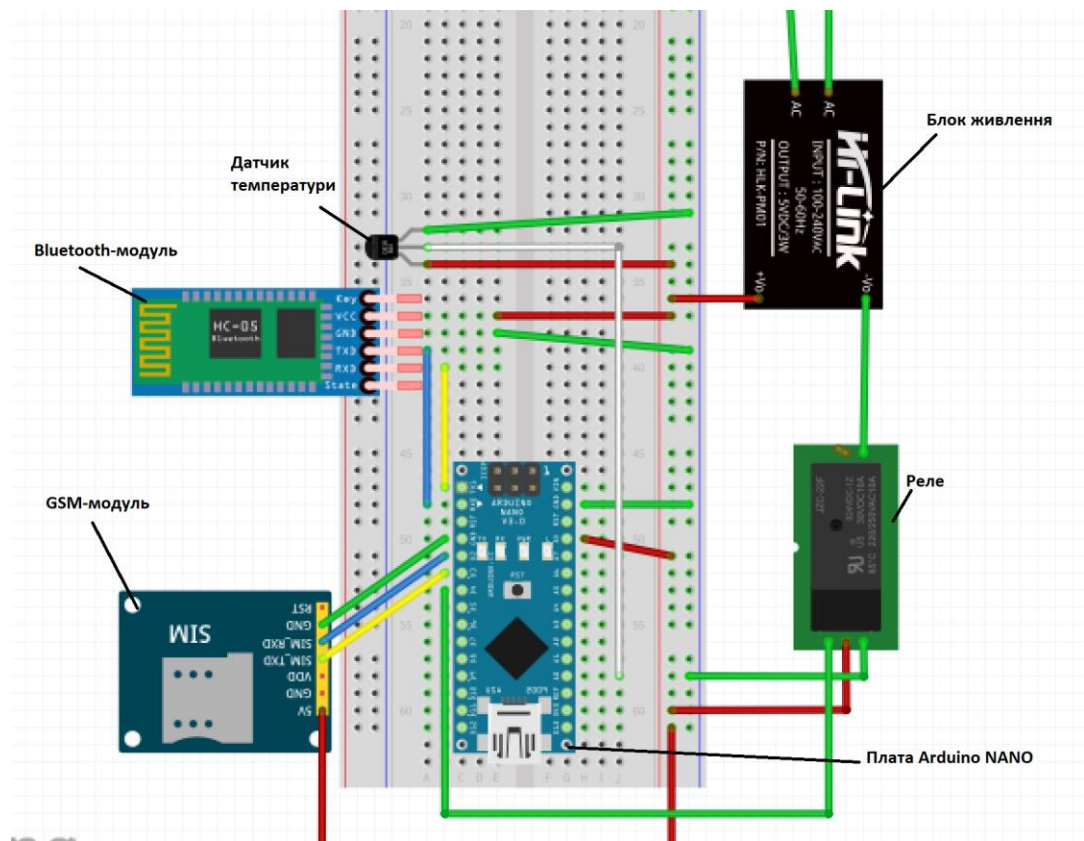


Рис. 3.4. Схема пристрою на макетній платі

Розроблено принципову схему (рис. 3.5).

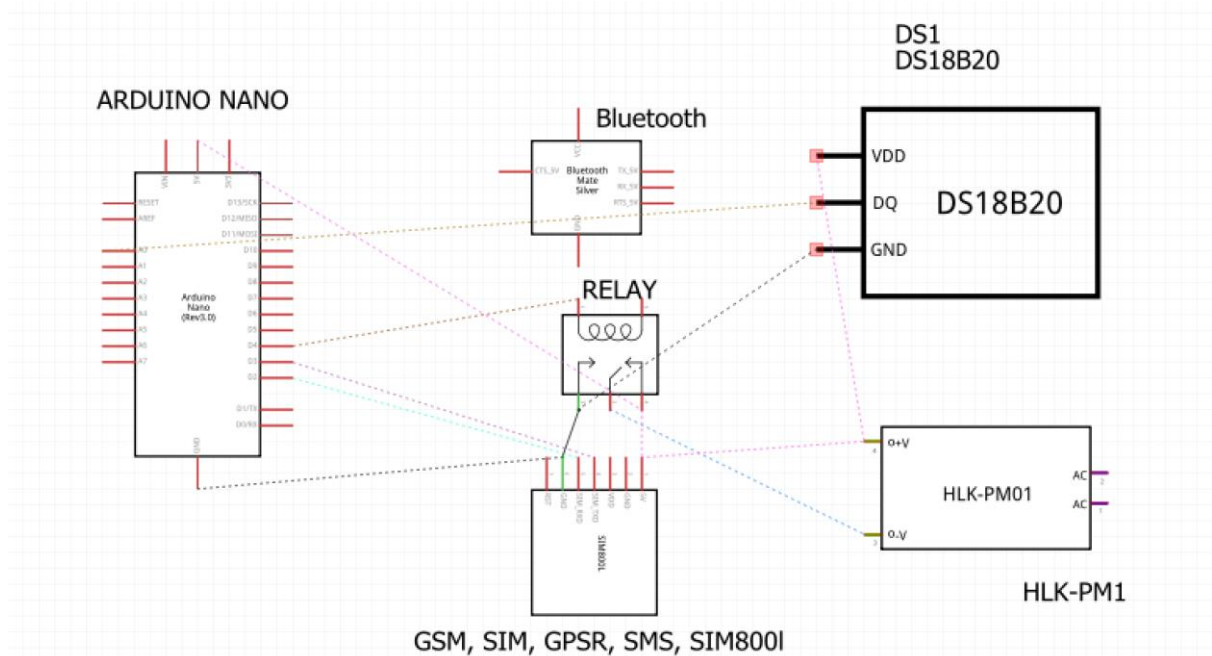


Рис 3.5. Принципова схема апаратної частини

В процесі програмування апаратної частини використано середовище розробки *Arduino IDE*.

Процес розробки розпочинається зі створення нового проекту в *Arduino IDE* (рис. 3.6).

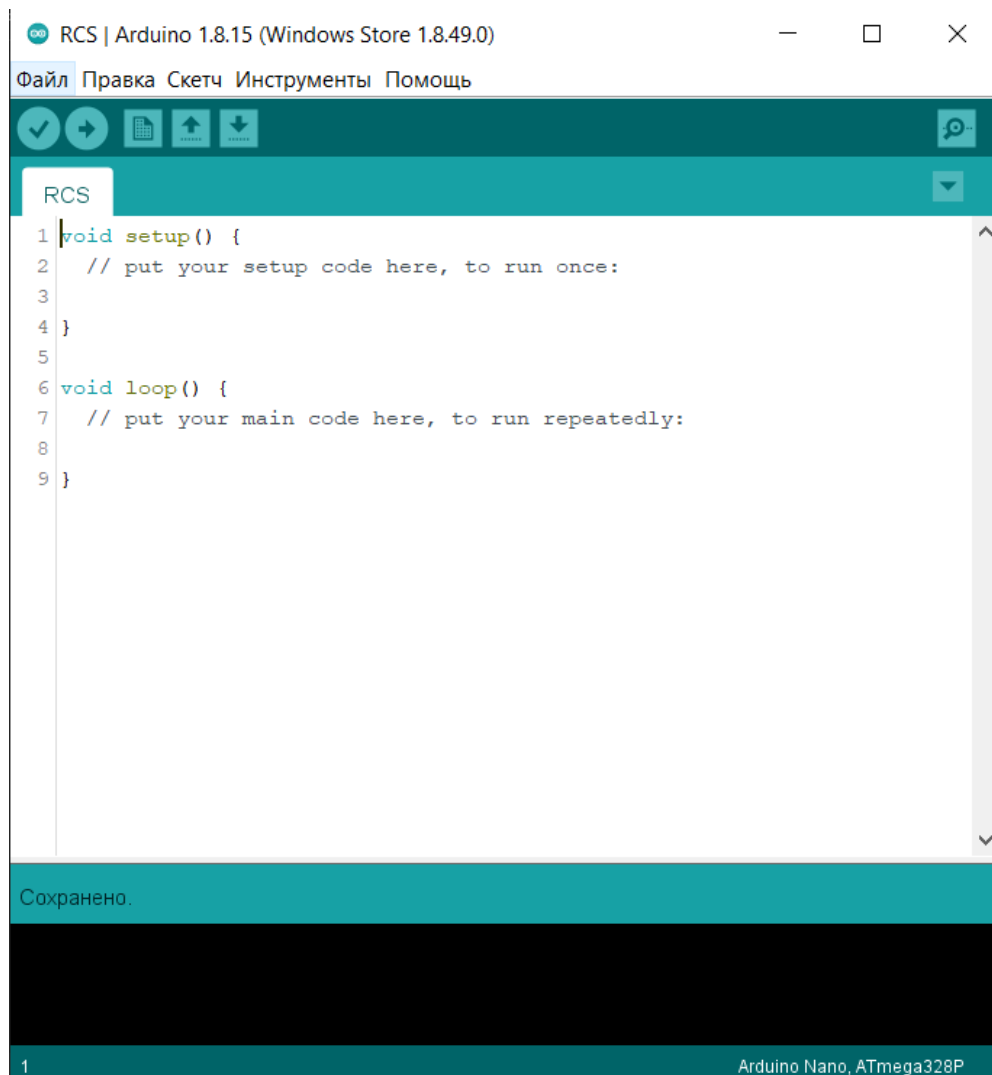


Рис. 3.6. Новий проект

Для використання *Bluetooth* та *GSM* модулів підключено бібліотеку *SoftwareSerial.h*. Для цього у рядку меню вибрано пункт “Скетч”, у випадаючому списку вибрано “Підключити бібліотеку” та бібліотека *SoftwareSerial.h* (рис. 3.7).

Після підключення бібліотеки в кодї програми автоматично генерується рядок:

```
#include <SoftwareSerial.h>
```

Створено змінну для *GSM* модуля. На другий цифровий пін – *D2* підключено пін *TX SIM800L*, а на *D3* – *RX SIM800L*:

```
SoftwareSerial Sim800(2,3);
```

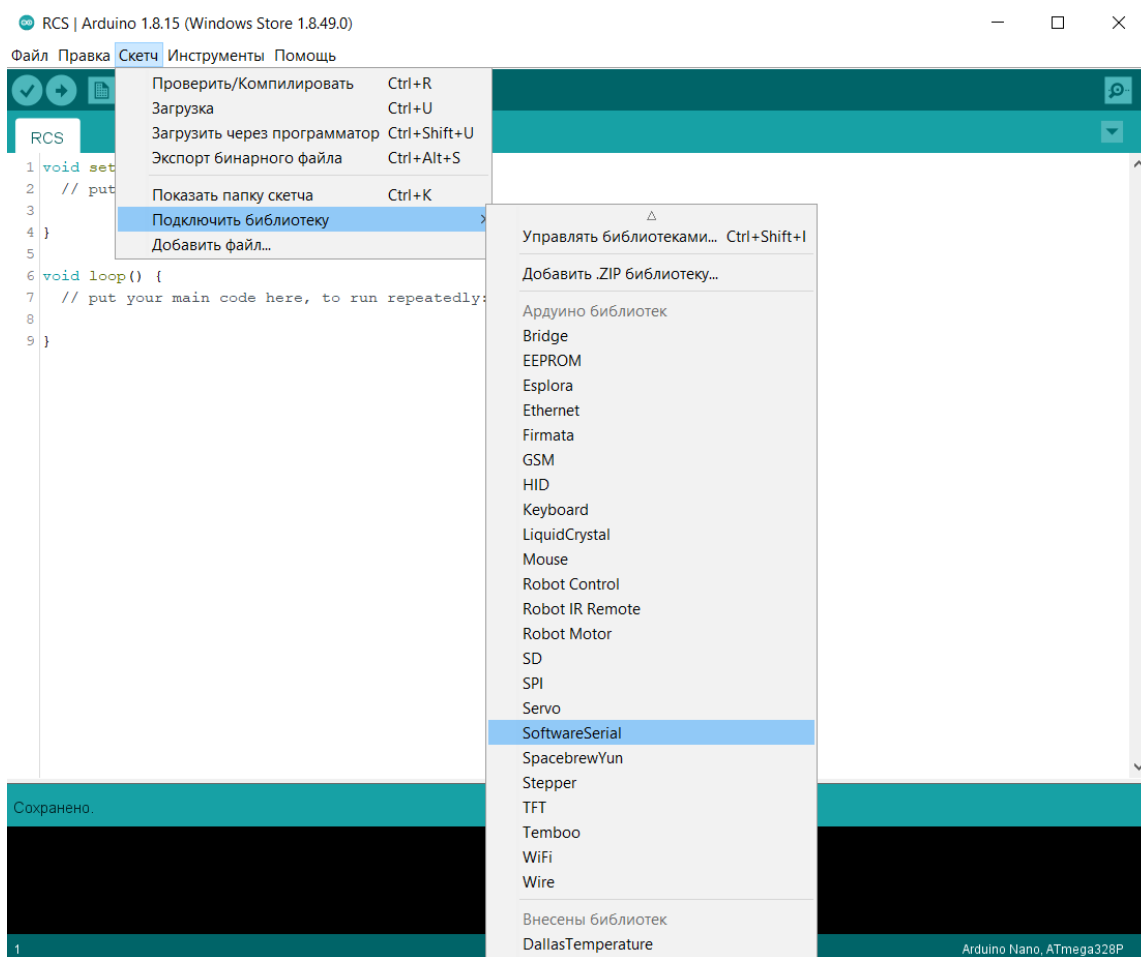


Рис. 3.7. Підключення бібліотеки *SoftwareSerial.h*

Наступним кроком є підключення *Serial.monitor*, для налаштування *GSM* та *Bluetooth* модулів:

```
Serial.begin(9600);
```

```
Sim800.begin(9600);
```

9600 – швидкість обміну даними з комп'ютером.

Для налаштування модулів використано AT-команди та послідовний монітор порт.

Функція *Serial.available()* отримує кількість байт (символів) доступних для читання з послідовного інтерфейсу зв'язку. Це ті байти які вже надійшли і записані в буфер послідовного порту.

Створено змінну для зберігання отриманих байтів та перевірку на наявність отриманих байтів:

```
if (Serial.available() > 0) {
    int incomingByte = Serial.read();
```



У разі отримання даних відбувається перевірка отриманого байту. В залежності від значення байту встановлюється відповідна температура та команда мікроконтролера:

```
if (incomingByte == '0')
{
    digitalWrite(4, LOW);
}
if (incomingByte == '1')
{
    t = 59;
    digitalWrite(4, HIGH);
}
if (incomingByte == '2')
{
    t = 79;
    digitalWrite(4, HIGH);
}
if (incomingByte == '3')
{
    t = 99;
    digitalWrite(4, HIGH);
}
```

Каступним кроком описано отримання команд безпосередньо через модуль *GSM*:

```
boolean isStringMessage = false;
```

Перевірка на отримання даних від модуля *SIM800L*:

```
if (!Sim800.available()){
return;
```

Створено змінну *Symb* для зберігання отриманих символів:

```
char Symb = Sim800.read();
```

Перевірка на завершення отримання повідомлення:

```
if ('\r' == Symb)
```

```
{
```

```
if (isStringMessage) {
```

Перевірка на відповідність отриманих символів командам ввімкнення та вимкнення:

```
if (strcmp(Symb, "turn on") != 0)
```

```
{
```

```
t = 99;
```

```
digitalWrite(4, HIGH);
```

```
}
```

```
if (strcmp(Symb, "turn off") != 0)
```

```
digitalWrite(4, LOW);
```

Під час роботи з датчиком температури використано послідовний периферійний інтерфейс. Послідовний периферійний інтерфейс (*SPI*) – фактичний послідовний синхронний повнодуплексний стандарт передачі даних, розроблений фірмою *Motorola* для забезпечення простого сполучення мікроконтролерів та периферії.

Для роботи інтерфейсу підключено бібліотеку *OneWire.h*, а для роботи з датчиком температури – *DallasTemperature.h*.

Функція, яка відправляє запит на зчитування показань з сенсора:

```
dallasSensors.requestTemperatures();
```

Створено змінну для зберігання значення температури:

```
int temperature = dallasSensors.getTempCByIndex(0);
```

Перевірка температури води. Якщо температура перевищить задану при ввімкненні, пристрій вимикається та надсилає *SMS* повідомлення.

```
if (temperature > t)
```

```
{
```

```
digitalWrite(4, LOW);
```

```
Serial.print("Your water boiled! The device is turned off");
```

```
delay(5000);
```

```
SendSMS();
```

```

    }
Код надсилення SMS повідомлення:
void SendSMS() {
    Serial.println("Sending SMS...");
    Sim800.print("AT+CMGF=1\r");
    delay(100);
    Sim800.print("AT+CMGS=\"+38096*****\"\r");
    delay(500);
    Sim800.print("Your water boiled!");
    delay(500);
    Sim800.print((char)26);
    delay(500);
    Sim800.println();
    Serial.println("Text Sent.");
    delay(500);
}

```

### 3.3. Розробка програмного модуля

В процесі розробки додатку використано середовище розробки *Android Studio*, об'єктно-орієнтований метод програмування та мову програмування *Java*.

Процес розробки розпочинається зі створення нового проекту в *Android Studio*. У якості шаблону обираємо *Empty Activity* (рис. 3.5).

Проекту дано назву та обрано мінімальні системні вимоги (рис. 3.6).

У файлі *AndroidManifest.xml* додано дозволи на використання додатком технології *Bluetooth* (рис. 3.7).

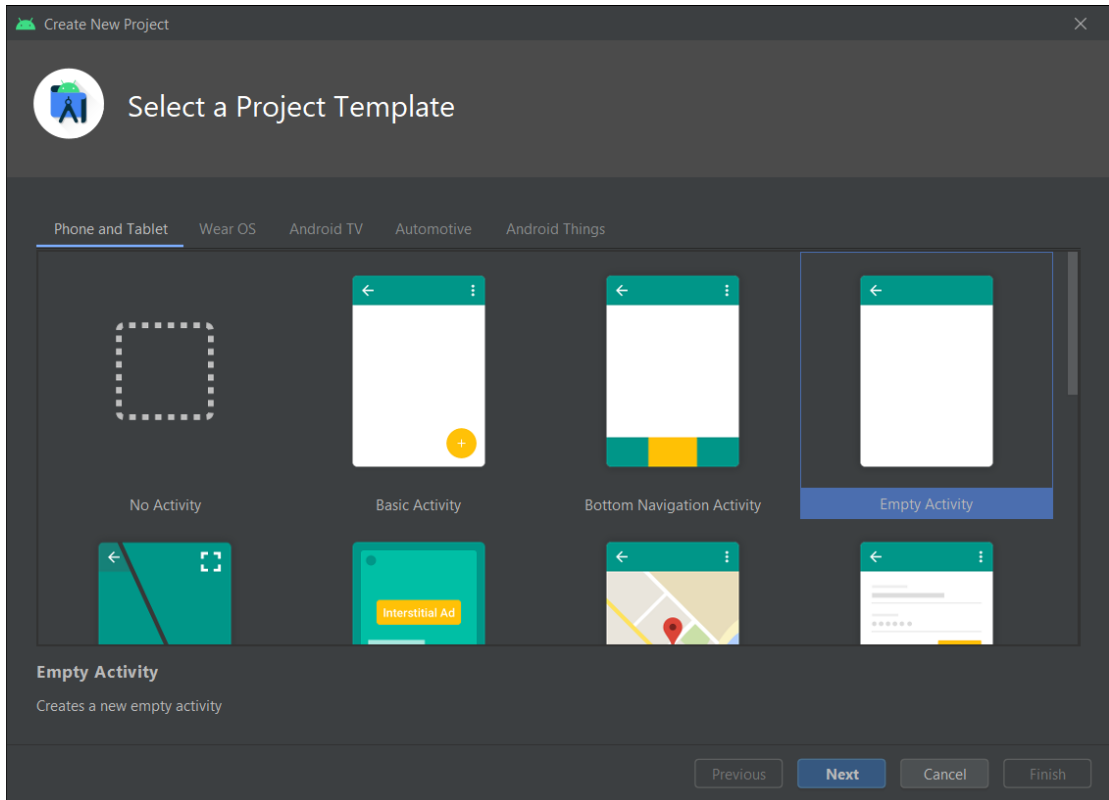


Рис. 3.5. Вибір шаблону *Empty Activity*

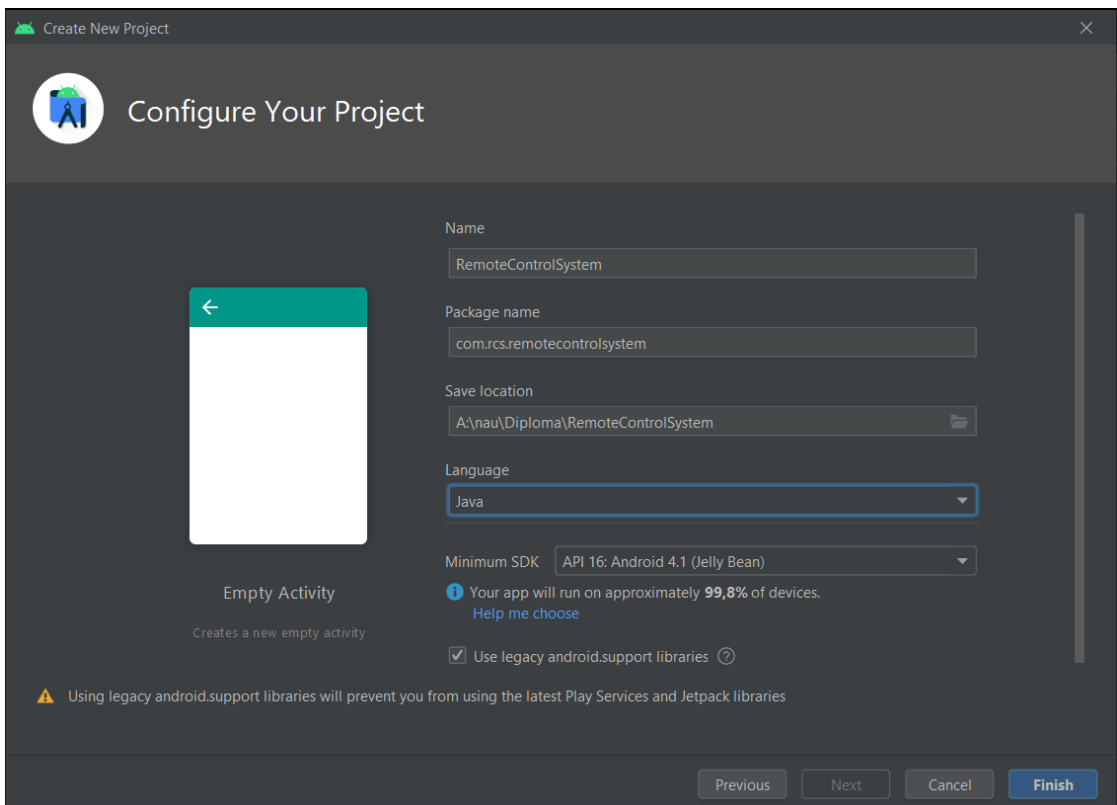


Рис. 3.6. Створення класу у проекті *Android Studio*

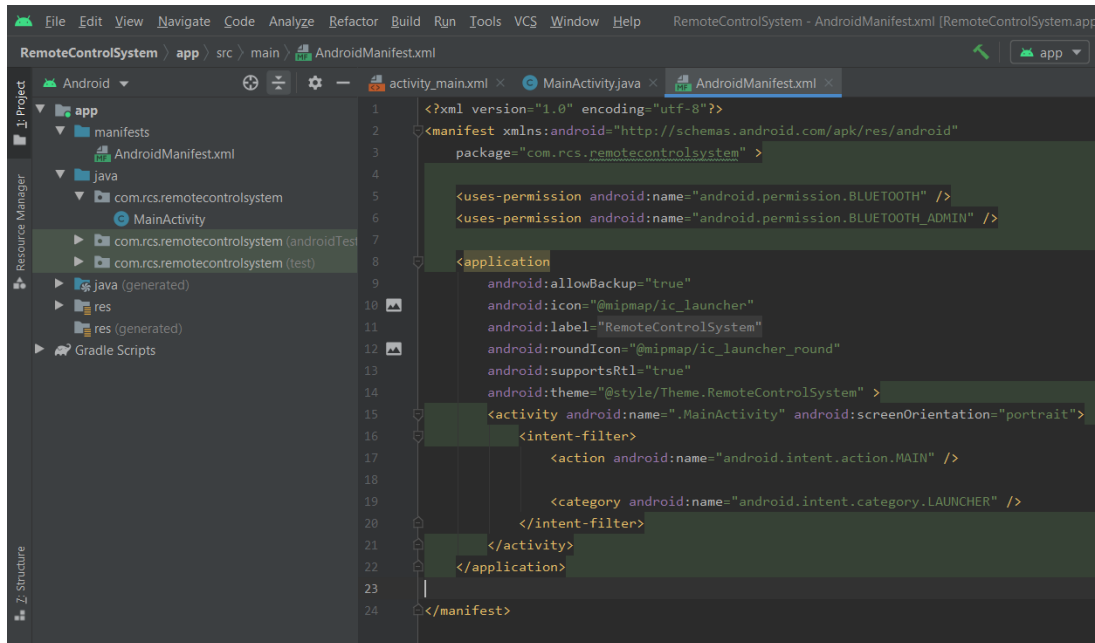


Рис. 3.7. Файл *AndroidManifest.xml*

У файлі *activity\_main.xml* створено елементи *RelativeLayout*, *TextView*, *ListView*, *FrameLayout*, *RadioGroup*, *RadioButton* та *ToggleButton* (рис. 3.8).

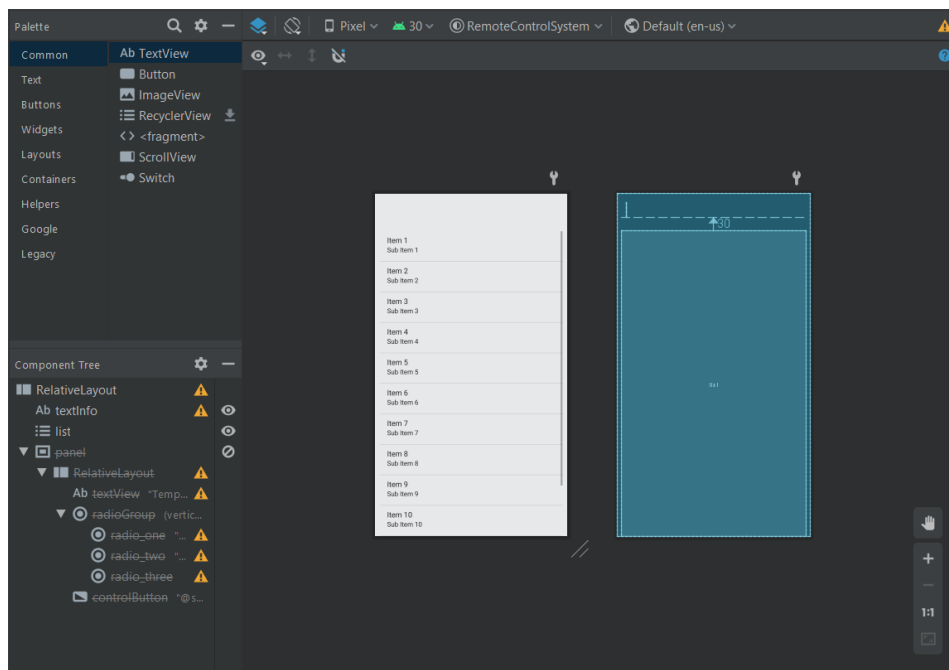


Рис. 3.8. Файл *activity\_main.xml*

Наступним кроком є програмування класу *MainActivity*. Основними методами класу *MainActivity* є *run()* та *onCheckedChanged()*. Метод *run()* відповідає за підключення до апаратної частини за допомогою *Bluetooth* модуля. Якщо підключення успішне, на екрані додатку з'являється елемент *FrameLayout*,

який містить у собі елементи *RadioGroup*, *RadioButton* та *ToggleButton* для керування апаратною частиною.

*@Override*

```
public void run() {  
    boolean success = false;  
    try {  
        bluetoothSocket.connect();  
        success = true;  
    }  
    catch (IOException e) {  
        e.printStackTrace();  
        runOnUiThread(new Runnable() {  
            @Override  
            public void run() {  
                Toast.makeText(MainActivity.this, "No connection, check the  
Bluetooth device you want to connect to!", Toast.LENGTH_LONG).show();  
                listView.setVisibility(View.VISIBLE);  
            }  
        });  
        try {  
            bluetoothSocket.close();  
        }  
        catch (IOException e1) {  
            e1.printStackTrace();  
        }  
    }  
    if(success) {  
        runOnUiThread(new Runnable() {  
            @Override  
            public void run() {  
                ControlPanel.setVisibility(View.VISIBLE);  
            }  
        });  
    }  
}
```

```

        }
    });
    myThreadConnected = new ThreadConnected(bluetoothSocket);
    myThreadConnected.start();
}
}

```

Метод `onCheckedChanged()` відповідає за кнопку `ToggleButton` для керування апаратною частиною. При натисканні на кнопку програма буде надсилати апаратній частині байт з символами "0", "1", "2" або "3" в залежності від обраної температури.

```

@Override
public void onCheckedChanged(CompoundButton compoundButton, boolean
isChecked) {
    byte[] bytesToSend;
    if (compoundButton.getId() == R.id.controlButton) {
        if (isChecked) {
            if (myThreadConnected != null) {
                switch (radioGroup.getCheckedRadioButtonId()) {
                    case R.id.radio_one:
                        bytesToSend = "3".getBytes();
                        myThreadConnected.write(bytesToSend);
                        break;
                    case R.id.radio_two:
                        bytesToSend = "2".getBytes();
                        myThreadConnected.write(bytesToSend);
                        break;
                    case R.id.radio_three:
                        bytesToSend = "1".getBytes();
                        myThreadConnected.write(bytesToSend);
                        break;

```

```

        }
    }
    Toast.makeText(MainActivity.this, "TURNED ON",
Toast.LENGTH_SHORT).show();
    } else {
        if (myThreadConnected != null) {
            bytesToSend = "0".getBytes();
            myThreadConnected.write(bytesToSend);
        }
        Toast.makeText(MainActivity.this, "TURNED OFF",
Toast.LENGTH_SHORT).show();
    }
}
}
}

```

### 3.4. Керівництво користувача

Мобільний додаток працює на пристроях під управлінням *Android*. Мінімальна версія системи – 4.1. Остання версія, на якій проводилося тестування додатку – *Android* 11. Відповідно до інформації від розробника *Android* компанії *Google*, це покриває більше ніж 99,8% активних пристроїв під управлінням даної операційної системи, або іншими словами, пристрої випущені, починаючи з 2012-2013 років.

Для роботи з додатком потрібно встановити *.apk* файл на смартфон.

Розпочати роботу з додатком можна натиснувши на його іконку у себе на смартфоні (рис. 3.9).

Розпочавши роботу, користувач повинен надати дозвіл на використання *Bluetooth* (рис. 3.10).





Рис. 3.9. Іконка додатку на робочому столі смартфона

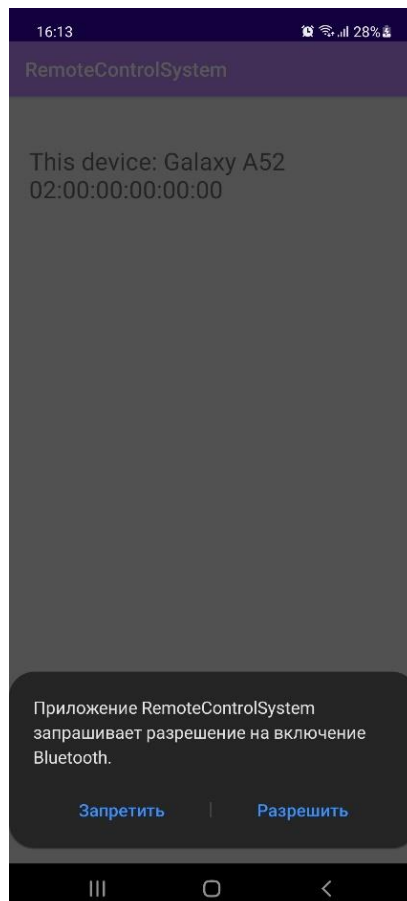


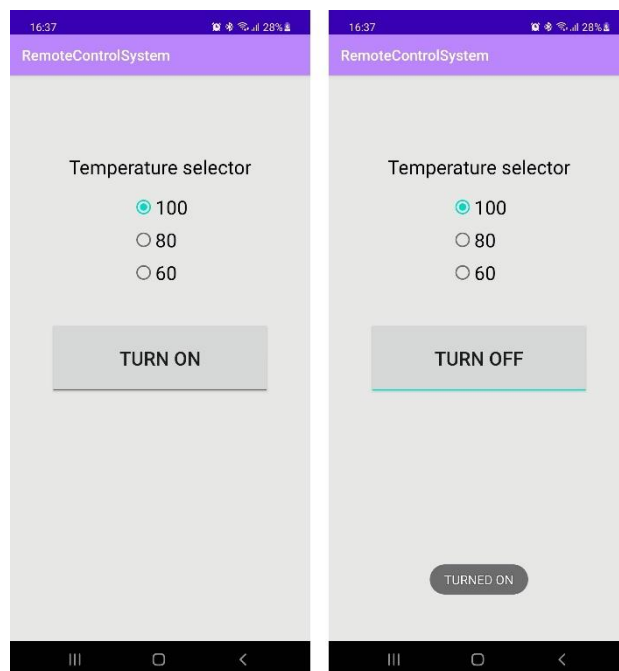
Рис. 3.10. Запит на ввімкнення *Bluetooth*

Користувач має обрати пристрій для підключення, а саме *HC-05* (рис. 3.11).



Рис. 3.11. Список пристроїв для підключення

Після підключення користувач потрапляє у меню керування апаратною частиною у якому може обрати температуру та ввімкнути або вимкнути пристрій (рис. 3.12).



а)

б)

Рис. 3.12. Меню керування апаратною частиною:

а) при стані вимкнено, б) при стані увімкнено

Для керування апаратною частиною з допомогою *GSM* користувачу необхідно відправити смс з текстом “*turn on*”, якщо користувач захоче завчасно вимкнути нагрівання він може надіслати смс з текстом “*turn off*” (рис. 3.13).

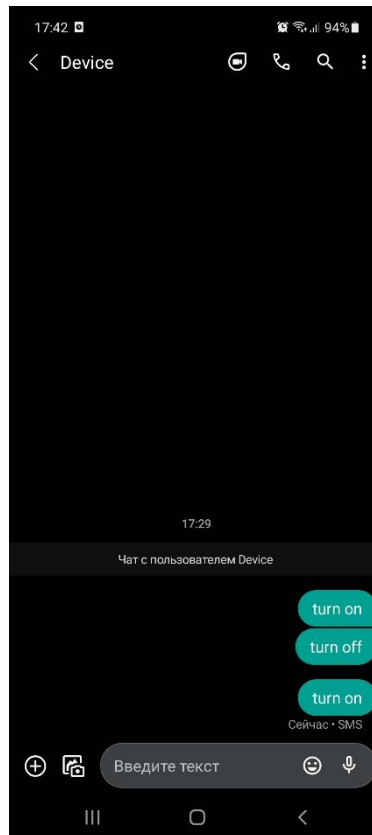


Рис. 3.13. Керування апаратною частиною з допомогою *GSM*

Після нагрівання і вимкнення пристрою користувач отримує *SMS* повідомлення (рис. 3.14).

### 3.5. Висновки до розділу

В даному розділі реалізовано алгоритм роботи системи дистанційного керування роботою побутових приладів, який відображає основні процеси, що відбуваються у системі, а саме: вибір типу керування та робота апаратної частини.

Побудовано схему підключення всіх складових апаратної частини та принципову схему апаратної частини.

Реалізовано апаратну частину системи дистанційного керування роботою побутових приладів.

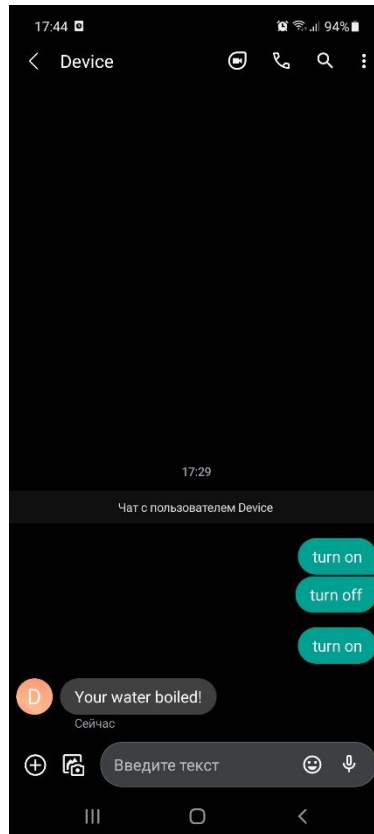


Рис. 3.14. SMS повідомлення про завершення роботи

Реалізовано програмну частину системи дистанційного керування роботою побутових приладів.

Описано керівництво користувача, у якому показано основні функції системи дистанційного керування роботою побутових приладів. Покроково описано цикл дистанційного керування роботою побутового приладу користувачем.

## ВИСНОВКИ

Дипломний проект присвячено актуальній темі дистанційного керування побутовими приладами. На основі розробки системи керування електрочайником, в якій було спроектовано та розроблено апаратну та програмну частини, було досліджено основні підходи до розробки подібних систем.

В межах першого розділу було проаналізовано:

– сучасні технології та методи реалізації систем дистанційного керування побутовими приладами. Для аналізу обрано такі відомі рішення як: *1-Wire, C-Bus, EIB/KNX, LanDrive, LonWorks, X10* та *Z-Wave*;

– аналоги та готові програмні рішення в області дистанційного керування побутовими приладами. Для аналізу обрано такі відомі рішення як: *REDMOND SkyKettle M173S-E, MIE Smart Kettle, Xiaomi Mijia Multifunctional Electric Cooker, E.Zicom Aromatea, Element El'kettle WF11MB / MW, Polaris PWK 1725CGLD WIFI IQ Home*;

– технології створення апаратних модулів дистанційного керування побутовими приладами. Для реалізації даного проекту обрано технологію *Bluetooth*, оскільки вона підтримує досить високу швидкість передачі даних і досить дешева.

В другому розділі було описано можливості систем дистанційного керування та розроблено структуру системи дистанційного керування побутовими приладами, яка складається з апаратної та програмної частин.

Розроблена структура передбачає, що апаратна частина містить мікроконтролер, *Bluetooth*-модуль, *gsm*-модуль, датчик температури та програмне забезпечення для мікроконтролера, програмна частина – інтерфейс та функціонал мобільного додатку.

Також в розділі наведено діаграми прецедентів, послідовності та взаємодії роботи основних частин системи дистанційного керування побутовими приладами.

Розроблена функціональна схема роботи програмного модуля, відображає роботу основних компонентів ПЗ системи (підключення до апаратного модуля та надсилання *Bluetooth* команд).

В третьому розділі реалізовано алгоритм роботи системи дистанційного керування роботою побутових приладів, який відображає основні процеси, що відбуваються у системі, а саме: вибір типу керування та робота апаратної частини.

Побудовано схему підключення всіх складових апаратної частини та принципову схему апаратної частини.

Реалізовано апаратну частину системи дистанційного керування роботою побутових приладів.

Реалізовано програмну частину системи дистанційного керування роботою побутових приладів.

Описано керівництво користувача, у якому показано основні функції системи дистанційного керування роботою побутових приладів. Покроково описано цикл дистанційного керування роботою побутового приладу користувачем.

Подальші розробки будуть спрямовані на універсалізацію програмної та апаратної частин, що дозволить підключатись до різноманітних пристроїв та дистанційно ними керувати.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технологии построения системы "Умный дом" [Электронный ресурс] // [www.e-field.biz](http://www.e-field.biz) - Режим доступа: <http://www.e-field.biz/smart-house/technologies/2>
2. 7 лучших умных чайников [Электронный ресурс] // [expertology.ru](http://expertology.ru) - Режим доступа: <https://expertology.ru/7-luchshikh-umnykh-chaynikov/>
3. *Polaris PWK 1725CGLD* [Электронный ресурс] // [tekhnik.top](http://tekhnik.top) - Режим доступа: <https://tekhnik.top/rejting-umnyh-chajnikov-2019/#Polaris-PWK-1725CGLD-WIFI-IQ-Home>
4. *How to Achieve Ranges of over 1 Km using Bluetooth Low Energy* [Электронный ресурс] // [www.novelbits.io](http://www.novelbits.io) - Режим доступа: <https://www.novelbits.io/long-range-bluetooth-coded-phy/>
5. *Soltanian A., Van Dyck R.E. Performance of the Bluetooth system in fading dispersive channels and interference // IEEE Global Telecommunications Conference, 2001 (GLOBECOM '01).* - с. 3499-3503.
6. Вишневикий и др. Широкополосные беспроводные сети передачи данных. - М.: Техносфера, 2005. - 592 с. - ISBN 5-94836-049-0.
7. *Wi-Fi* - Вікіпедія [Электронный ресурс] // [uk.wikipedia.org](http://uk.wikipedia.org) - Режим доступа: <https://uk.wikipedia.org/wiki/Wi-Fi>.
8. Сташин В.В. Проектирование цифровых устройств на однокристалльных микроконтроллерах / Сташин В.В. - М.: Энергоатомиздат, 2010. – 224 с.
9. Щелкунов Н.Н. Микропроцессорные средства и системы / Щелкунов Н.Н., Дианов А.Н. - М.: Радио и связь, 2009. – 360 с.
10. Бодрова А.А., Логвин В.И. Позиционирование и взаимодействие в беспроводных сенсорных сетях / А.А. Бодрова, В.И. Логвин // Молодой ученый. – 2015. – №6. – С. 129-132.
11. Высокопроизводительные сети. Энциклопедия пользователя./ Под ред. Марка А. Спортака: Пер. с англ. – К.,: ДиаСофт, 2017. – 836 с.

12. Информационные компьютерные сети: Учебно-методическое пособие для практических занятий.–М.: ИМПЭ им.А.С. Грибоедова, 2012. – 63 с.
13. Марк Э.С. Практические советы и решения по созданию "Умного дома", НТ Пресс, 2017. – 256 с.
14. Педжман Р., Джонатан Л. Основы построения беспроводных локальных сетей стандарта 802.11 / Р. Педжман, Л. Джонатан – М.:, 2006. – 315 с.
15. Тесля Е.В. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире, 2018. – 212 с.
16. Харке В.А., Умный дом. Объединение в сеть бытовой техники и системы коммуникаций в жилищном строительстве // Техносфера, 2016. – 212 с.
17. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.
18. Бойченко С.В., Иванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.





```
import java.io.IOException;

import java.io.OutputStream;

import java.util.ArrayList;

import java.util.Set;

import java.util.UUID;

import static android.R.layout.simple_list_item_1;

public class MainActivity extends AppCompatActivity implements
CompoundButton.OnCheckedChangeListener{

    ToggleButton cB;

    private static final int REQUEST_ENABLE_BT = 1;

    public TextView textInfo;

    BluetoothAdapter bluetoothAdapter;

    ArrayList<String> pairedDeviceArrayList;

    ListView listView;

    FrameLayout ControlPanel;

    ArrayAdapter<String> pairedDeviceAdapter;

    ThreadConnectBTdevice myThreadConnectBTdevice;

    ThreadConnected myThreadConnected;

    private UUID myUUID;

    RadioGroup radioGroup;

    RadioButton radioButton;
```

```

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    cB = (ToggleButton) findViewById(R.id.controlButton);

    cB.setOnCheckedChangeListener(this);

    final String UUID_STRING_WELL_KNOWN_SPP = "00001101-0000-
1000-8000-00805F9B34FB";

    textInfo = findViewById(R.id.textInfo);

    listView = findViewById(R.id.list);

    ControlPanel = findViewById(R.id.panel);

    if
(!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOT
H)){

        Toast.makeText(this, "BLUETOOTH NOT support",
Toast.LENGTH_LONG).show();

        finish();

        return;

    }

    myUUID = UUID.fromString(UUID_STRING_WELL_KNOWN_SPP);

    bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    if (bluetoothAdapter == null) {

        Toast.makeText(this, "Bluetooth is not supported on this hardware
platform", Toast.LENGTH_LONG).show();
    }

```

```

        finish();

        return;
    }

    String stInfo = bluetoothAdapter.getName() + " " +
bluetoothAdapter.getAddress();

    textInfo.setText(String.format("\n" + "This device: %s", stInfo));

    radioGroup = findViewById(R.id.radioGroup);
}

@Override

protected void onStart() {

    super.onStart();

    if (!bluetoothAdapter.isEnabled()) {

        Intent enableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

        startActivityForResult(enableIntent, REQUEST_ENABLE_BT);

    }

    setup();
}

private void setup() {

    Set<BluetoothDevice> pairedDevices =
bluetoothAdapter.getBondedDevices();

    if (pairedDevices.size() > 0) {

        pairedDeviceArrayList = new ArrayList<>();

```

```

        for (BluetoothDevice device : pairedDevices) {

            pairedDeviceArrayList.add(device.getName() + "\n" +
device.getAddress());

        }

        pairedDeviceAdapter = new ArrayAdapter<>(this, simple_list_item_1,
pairedDeviceArrayList);

        listView.setAdapter(pairedDeviceAdapter);

        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

            @Override

            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

                listView.setVisibility(View.GONE);

                String itemValue = (String) listView.getItemAtPosition(position);

                String MAC = itemValue.substring(itemValue.length() - 17);

                BluetoothDevice device2 =
bluetoothAdapter.getRemoteDevice(MAC);

                myThreadConnectBTdevice = new
ThreadConnectBTdevice(device2);

                myThreadConnectBTdevice.start();

            }

        });

    }

}

@Override

protected void onDestroy() {

```

```

        super.onDestroy();

        if(myThreadConnectBTdevice!=null) myThreadConnectBTdevice.cancel();
    }

    @Override

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == REQUEST_ENABLE_BT) {

            if(resultCode == Activity.RESULT_OK) {

                setup();

            } else {

                Toast.makeText(this, "BlueTooth turned off",
Toast.LENGTH_SHORT).show();

                finish();

            }

        }

    }

    private class ThreadConnectBTdevice extends Thread {

        private BluetoothSocket bluetoothSocket = null;

        private ThreadConnectBTdevice(BluetoothDevice device) {

            try {

                bluetoothSocket =
device.createRfcommSocketToServiceRecord(myUUID);

            }

        }

    }

```

```
        catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
@Override
```

```
public void run() {
```

```
    boolean success = false;
```

```
    try {
```

```
        bluetoothSocket.connect();
```

```
        success = true;
```

```
    }
```

```
    catch (IOException e) {
```

```
        e.printStackTrace();
```

```
        runOnUiThread(new Runnable() {
```

```
            @Override
```

```
            public void run() {
```

```
                Toast.makeText(MainActivity.this, "No connection, check the  
Bluetooth device you want to connect to!", Toast.LENGTH_LONG).show();
```

```
                listView.setVisibility(View.VISIBLE);
```

```
            }
```

```
        });
```

```
    try {
```

```
        bluetoothSocket.close();
```

```
    }
```

```

        catch (IOException e1) {
            e1.printStackTrace();
        }
    }

    if(success) {
        runOnUiThread(new Runnable() {

            @Override
            public void run() {
                ControlPanel.setVisibility(View.VISIBLE);
            }
        });

        myThreadConnected = new ThreadConnected(bluetoothSocket);
        myThreadConnected.start();
    }
}

public void cancel() {
    Toast.makeText(getApplicationContext(), "Close - BluetoothSocket",
Toast.LENGTH_LONG).show();

    try {
        bluetoothSocket.close();
    }

    catch (IOException e) {
        e.printStackTrace();
    }
}

```



```

    }
}

}

private static class ThreadConnected extends Thread {
    private final OutputStream connectedOutputStream;

    public ThreadConnected(BluetoothSocket socket) {
        OutputStream out = null;
        try {
            out = socket.getOutputStream();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
        connectedOutputStream = out;
    }

    public void write(byte[] buffer) {
        try {
            connectedOutputStream.write(buffer);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

```
    }  
  }  
  
}
```

*@Override*

```
public void onCheckedChanged(CompoundButton compoundButton, boolean  
isChecked) {
```

```
    byte[] bytesToSend;
```

```
    if (compoundButton.getId() == R.id.controlButton) {
```

```
        if (isChecked) {
```

```
            if (myThreadConnected != null) {
```

```
                switch (radioGroup.getCheckedRadioButtonId()) {
```

```
                    case R.id.radio_one:
```

```
                        bytesToSend = "3".getBytes();
```

```
                        myThreadConnected.write(bytesToSend);
```

```
                        break;
```

```
                    case R.id.radio_two:
```

```
                        bytesToSend = "2".getBytes();
```

```
                        myThreadConnected.write(bytesToSend);
```

```
                        break;
```

```
                    case R.id.radio_three:
```

```
                        bytesToSend = "1".getBytes();
```

```
                        myThreadConnected.write(bytesToSend);
```

```

        break;
    }
}

    Toast.makeText(MainActivity.this, "TURNED ON",
Toast.LENGTH_SHORT).show();

    } else {

        if (myThreadConnected != null) {

            bytesToSend = "0".getBytes();

            myThreadConnected.write(bytesToSend);

        }

        Toast.makeText(MainActivity.this, "TURNED OFF",
Toast.LENGTH_SHORT).show();

    }

}

}

public void checkButton(View v){

    int radioId = radioGroup.getCheckedRadioButtonId();

    radioButton = findViewById(radioId);

    Toast.makeText(this, "Selected temperature is " + radioButton.getText(),
Toast.LENGTH_SHORT).show();

}

}

```