

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ ТА ПРОГРАМНОЇ
ІНЖЕНЕРІЇ**

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Литвиненко О.Є.

«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"МАГІСТР"**

Тема: “Програмна система дистанційного керування побутовими приладами”

Виконавець: _____ Олександр КІСЛОВ

Керівник: _____ Євген АРТАМОНОВ

Нормоконтролер: _____ Євген АРТАМОНОВ

Київ 2022

6. Календарний план

| № п/п | Етапи виконання Кваліфікаційної роботи | Термін виконання етапів | Примітка |
|-------|---|-------------------------|----------|
| 1 | Провести аналіз літератури за темою кваліфікаційної роботи та аналіз існуючих систем | 05.09.2022 – 06.09.2022 | |
| 2 | Провести аналіз принципів побудови систем дистанційних керування побутовими приладами | 07.09.2022 – 17.09.2022 | |
| 3 | Розробити модель системи дистанційного керування побутовими приладами | 18.09.2022 – 20.09.2022 | |
| 4 | Розробити апаратні засоби системи дистанційного керування побутовими приладами | 21.09.2022 – 23.10.2022 | |
| 5 | Розробити програмні модулі системи дистанційного керування побутовими приладами | 24.10.2022 – 10.11.2022 | |
| 6 | Написати пояснювальну записку | 11.11.2022 – 15.12.2022 | |
| 7 | Підготувати презентацію та графічні матеріали | 12.12.2022 – 22.12.2022 | |

7. Дата видачі завдання _____ 05.09.2022 _____

Керівник _____ Євген АРТАМОНОВ _____
(підпис)

Завдання прийняв до виконання _____ Олександр КІСЛОВ _____
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи “Програмна система дистанційного керування побутовими приладами”: 92 с., 22 рис., 30 літературних джерел, 1 додаток.

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 модулі 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 | 12 |
| 1.1. Аналіз сучасних технологій та методів реалізації систем дистанційного керування побутовими приладами..... | 12 |
| 1.2. Аналіз технології X10 | 23 |
| 1.3. Постановка завдання проектування | 27 |
| 1.4. Висновки до розділу..... | 28 |
| РОЗДІЛ 2 ПРИНЦИПИ РОБОТИ СИСТЕМИ ДИСТАНЦІЙНОГО КЕРУВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ | 30 |
| 2.1. Функціональні вимоги до системи дистанційного керування побутовими приладами..... | 30 |
| 2.2. Огляд систем автоматизації тестування апаратно-програмних систем | 30 |
| 2.3. Структура системи дистанційного керування побутовими приладами..... | 41 |
| 2.4. Функціональне забезпечення системи дистанційного керування побутовими приладами..... | 45 |
| 2.3.1. Нефункціональна вимога | 46 |
| 2.3.2. Випадки використання | 48 |
| 2.3.3. Нефункціональні вимоги..... | 52 |
| 2.5. Висновки до розділу..... | 62 |
| РОЗДІЛ 3 ОПИСАННЯ ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ | 64 |
| 3.1. Алгоритм роботи системи дистанційного керування побутовими приладами..... | 64 |
| 3.2. Розробка апаратного модуля | 70 |

| | |
|--|----|
| 3.3. Розробка програмного модуля..... | 82 |
| 3.4. Керівництво користувача..... | 86 |
| 3.5. Висновки до розділу..... | 89 |
| ВИСНОВКИ | 90 |
| СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 93 |
| Додаток А | 96 |

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

Наплив пристроїв Інтернету речей (*IoT*) створив величезний попит на підключення всього до всього. Крім того, цей пульт дистанційного керування для побутової техніки дозволяє підключати ваші звичайні повсякденні прилади для керування ними за допомогою пульта. Все, що вам потрібно зробити, це підключити цю схему до будь-якої вашої побутової техніки (лампи, вентилятора, радіо тощо), і все готово.

Система керування побутовою технікою (*HACS*) — це система, яка надає різноманітні послуги для дистанційного керування побутовою технікою, наприклад мікрохвильовою піччю, телевізором, дверима гаража тощо, за допомогою віддалених пристроїв, таких як мобільний телефон, настільний комп'ютер і кишеньковий пристрій.

Система управління побутовою технікою управляється або мобільним телефоном, або долонею, або ПК. Вони з'єднані через протокол бездротової мережі (*WAP*), Інтернет або інтрамережу. Система *HACS* отримує команди від віддалених пристроїв, якими керує користувач. Система, у свою чергу, надсилає команди відповідним пристроям, які виконуватимуть дії. *HACS* відповідає за відстеження стану пристроїв. Якщо щось піде не так, він повідомить користувача, надіславши повідомлення на віддалені пристрої, а також у відділ екстреної допомоги, якщо це необхідно.

Сьогодні все більше людей намагаються зробити свій будинок екологічно чистим. Розумний дім дозволяє користувачеві управляти споживаною енергією та збільшувати економію шляхом контролю освітлення, віконних покриттів, зрошення та моніторингу використання. Портативність і технології смартфона збільшили інтерес користувачів до керування своїми приладами з смартфона. Автоматизоване керування приладом дозволяє користувачам виконувати завдання до приходу додому. Система управління розумним будинком надає рішення для допоміжних технологій, особливо для людей з обмеженими можливостями та людей похилого віку, які використовують мобільні програми

дистанційного керування. Згідно з повідомленням в [1], приблизно 72% респондентів сказали, що саморегульований термостат, а 71% сказали, що двері, які можуть бути заблокованими з віддаленого місця, були найважливішими функціями, коли справа доходить до найбажаніших пристроїв розумного дому.

Дослідження «розумного дому» стали популярними, однак ця система не є зручною для деяких груп людей, таких як інваліди та люди похилого віку, через її складність і вартість. Використання зв'язку GSM призводить до додаткової оплати за кожне повідомлення, надіслане через мережу. Крім того, запропонована система повинна мати простий у використанні графічний інтерфейс для керування та моніторингу. Використання веб-сервера є найкращим вибором для подолання цієї проблеми, оскільки один веб-сайт може охоплювати користувачів на різних типах мобільних пристроїв, тоді як рідні програми потребують розробки окремої версії для кожного типу пристроїв.

Це дослідження представляє систему управління розумним будинком, яка покращила б ручне управління домом. Потрібна менша кількість персоналу, одночасно підвищуючи загальну безпеку за рахунок інтеграції автоматичних побутових приладів на основі зчитування датчиків і кнопки інструкції користувача в розроблений інтерфейс веб-сайту. Автоматична функція, заснована на інформації датчика, дозволила системі керування працювати ефективно та ефективно. Концепція IP-мережових додатків і пристроїв у домі дозволяє керувати побутовою технікою з будь-якого місця з ноутбука, мобільного телефону, планшета чи смарт-телевізора, за умови, що ці пристрої мають доступ до Інтернету. Веб-сайт забезпечує зручність для розробників і користувачів, особливо для людей з обмеженими можливостями та літніх людей. Це дослідження надає захищений паролем веб-сайт для вирішення проблеми безпеки. Крім того, проблему батареї можна подолати, додавши сонячний зарядний пристрій для підзарядки слабкої батареї. Цей контролер сонячної зарядки також можна використовувати як резервний ресурс під час відключення електроенергії.

Систему управління розумним будинком (SHCS) можна інтегрувати в існуючу побутову техніку, щоб зменшити потребу в людському втручанні, підвищити безпеку та енергоефективність. Однак це все ще відкрита проблема через такі труднощі, як відстань до мережі, перешкоди сигналу, незручність для користувача, підвищена вартість і енергоспоживання. У цьому документі розглядаються різні теми технологій розумного дому, включаючи систему керування, розумну домашню мережу, розумну побутову техніку та сенсорні технології для розумного дому.

Розумний дім – це будинок, обладнаний спеціальними засобами, які дозволяють мешканцям контролювати або програмувати низку автоматизованих домашніх електронних пристроїв. Наприклад, домовласник у відпустці може поставити на охорону домашню систему безпеки, контролювати датчики температури, вмикати та вимикати електроприлади, контролювати освітлення, програмувати домашній кінотеатр або розважальну систему та виконувати багато інших завдань. Розумний дім став розумнішим, якщо керувати ним можна з будь-якої віддаленої точки. Наша основна увага - це управління побутовою технікою з віддаленого місця.

Мотивація дистанційного керування побутовою технікою проста. Не завжди можливо фізично перебувати поблизу дому, але іноді дуже важливо контролювати прилади для багатьох цілей. Таким чином, дистанційне керування передає контроль над домом за межі дому та в руки людей. Якщо простий мобільний телефон бере на себе додаткову відповідальність за керування розумним будинком, то керування доступним майже звідусіль, де люди подорожують і живуть на землі. Передбачається, що така високоякісна технологія полегшить використання різноманітних утиліт, що полегшують життя, у нову епоху та наближає речі до долоні.

Для дистанційного зв'язку існує кілька доступних засобів. Інтернет є хорошим прикладом такого типу віддаленого спілкування. Інтернет практично не обмежує географічне розміщення, тому за нашим визначенням вважається «досить» віддаленим. Інтернет - це місце, переповнене різними типами трафіку, часто ворожими один одному. Вразливість безпеки є найяскравішим джерелом

попередження в Інтернеті. Кожного разу, коли веб-додаток починає працювати, потрібно докласти багато зусиль, перш ніж можна буде сказати, що він захищений, якщо взагалі. Коли ми говоримо про дистанційне керування, ми хочемо переконатися, що жодна зловмисна сторона ніколи не отримає контроль і не скасує все. Також для використання Інтернету потрібні такі ресурси, як бездоганне підключення до Інтернету та хостинг-сервери, що не завжди відповідає концепції дистанційного керування.

Іншим варіантом вирішення цієї проблеми віддаленого спілкування є використання мобільного телефону. Мобільна телефонія пропонує широкий спектр комунікаційних послуг, таких як передача голосу та даних через SMS та інші розширені протоколи передачі даних, такі як GPRS, EDGE, за відносно низькою ціною та в різноманітних місцях на землі. З іншого боку, безпека краще досягається за допомогою суворого контролю руху. Ми зупинилися на цьому способі дистанційного керування побутовою технікою через його неперевершену доступність і скромну безпеку за доступною ціною.

Об'єкт – дистанційний спосіб керування роботою побутових приладів.

Предмет – система дистанційного керування роботою побутових приладів.

Мета – розробити програмний модуль та апаратний модулі 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.

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.

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. Відеокмутатори;
4. Розетки з інтерфейсом 1-Wire;
5. Модулі аналогових сигналів, модулі управління навантаженням 220В;

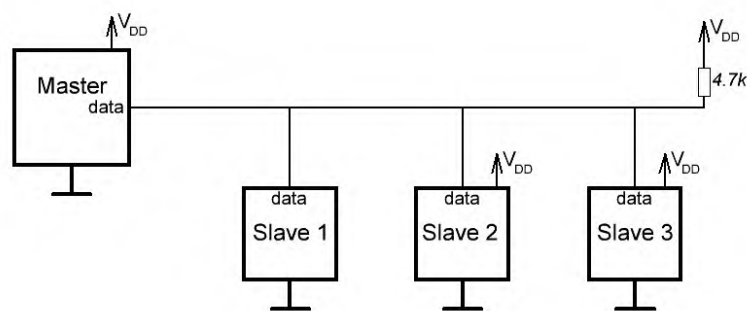


Рис. 1.1. Приклад реалізації технології 1-Wire

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.

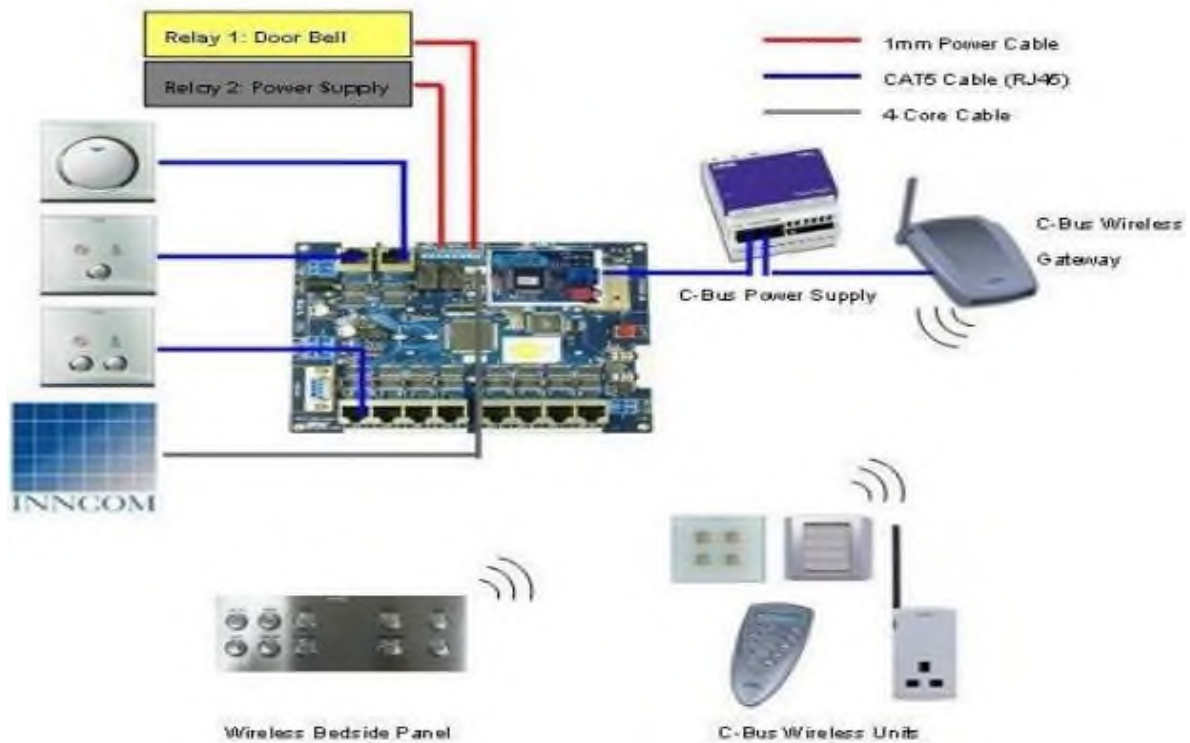


Рис. 1.2. Приклад реалізації технології *C-Bus*

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.

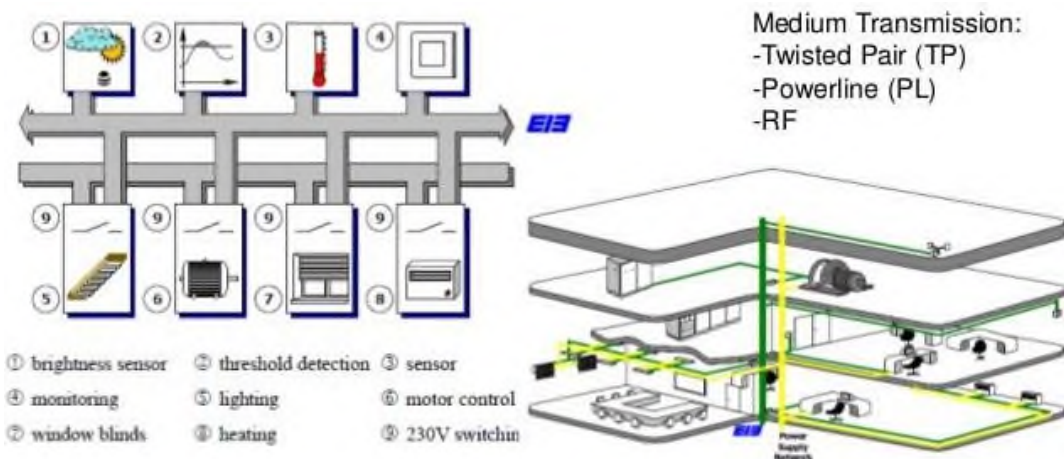


Рис. 1.3. Принцип реалізації технології *EIB*

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).

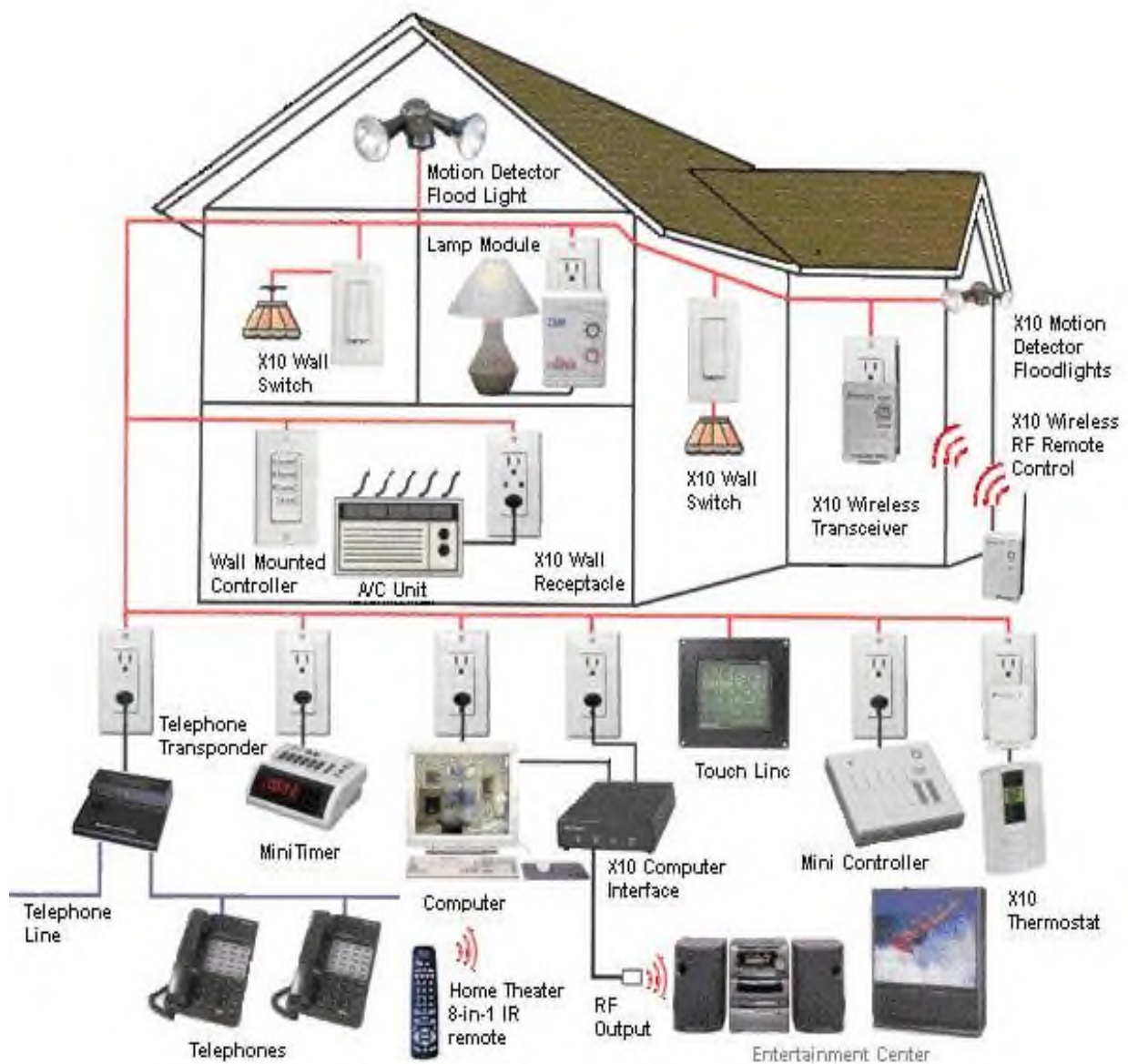


Рис. 1.4. Приклад реалізації технології X-10

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.

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].

1.1.8. Аналіз принципів використання мобільної телефонії

У сфері телекомунікацій телефонія охоплює загальне використання обладнання для забезпечення голосового зв'язку на відстані, зокрема шляхом підключення телефонів один до одного. Термін мобільна телефонія походить від оригінальної телефонії для позначення зв'язку, який сприяє мобільності за допомогою бездротових технологій. Мобільна телефонія [1] пропонує такі послуги, як передача голосу та даних. Передача даних здійснюється за допомогою SMS та деяких інших служб підвищеної швидкості передачі даних, таких як GPRS і EDGE. Останні два забезпечують доступ до Інтернету для мобільних телефонів. Служба коротких повідомлень (SMS) — це телекомунікаційний протокол, який дозволяє надсилати короткі (160 символів або менше) текстові повідомлення. Він доступний на більшості цифрових мобільних телефонів і деяких персональних цифрових помічниках із вбудованим бездротовим зв'язком. Такі як Bluetooth і AT-команди також іноді можуть надсилати SMS-повідомлення за допомогою мобільних телефонів. Зручність використання цього типу SMS може бути використана в контролері, який ми хочемо розробити.

X10 Active Home Controller Pro: контролер X10 [2] поставляється як пакет, який містить один модуль контролера та інші модулі приладів, класифіковані за класами, як-от освітлення, вентилятори тощо. Контролер підключається до комп'ютера за допомогою стандартних інтерфейсів. Його можна налаштувати за допомогою наданого програмного забезпечення з комп'ютера, а також з пульта дистанційного керування, який постачається в комплекті. Між контролером і приладами вставляється спеціальний модуль або загальний модуль. Контролер безпосередньо впливає на модулі, а не на прилади, підключені до модулів. Таким чином, пристрої абсолютно не знають про наявність системи управління будинком X10, і тому X10 не обмежує свою діяльність певними постачальниками. Контролер X10 використовує лінію живлення для надсилання та отримання команд модулям. Цей сигнал передається з використанням смуги пропускання, яка не заважає існуючим з'єднанням живлення. Щоб керувати конкретним пристроєм із багатьох підключених таким чином, X10 використовує механізм адресації для виявлення потрібного. Ця адресація встановлюється в модулях приладу перед його підключенням і може бути змінена в будь-який час. Кожного разу, коли контролер X10 повинен надіслати деякі команди, він транслює команду на лінію живлення. Команда містить адресу пристрою, який призначений для керування. Отже, модуль, адреса якого збігається з адресою в команді, відповідає негайно. Таким чином, він обробляє конкретний запит, виданий контролером для керування пристроєм. Деталі технології X10 будуть надані пізніше. X10 використовує механізм адресації для виявлення потрібного. Ця адресація встановлюється в модулях приладу перед його підключенням і може бути змінена в будь-який час. Кожного разу, коли контролер X10 повинен надіслати деякі команди, він транслює команду на лінію живлення. Команда містить адресу пристрою, який призначений для керування. Отже, модуль, адреса якого збігається з адресою в команді, відповідає негайно. Таким чином, він обробляє конкретний запит, виданий контролером для керування пристроєм. Деталі технології X10 будуть надані пізніше. X10 використовує механізм адресації для виявлення потрібного. Ця адресація встановлюється в модулях приладу перед його підключенням і може бути змінена в будь-який час. Кожного

разу, коли контролер X10 повинен надіслати деякі команди, він транслює команду на лінію живлення. Команда містить адресу пристрою, який призначений для керування. Отже, модуль, адреса якого збігається з адресою в команді, відповідає негайно. Таким чином, він обробляє конкретний запит, виданий контролером для керування пристроєм. Деталі технології X10 будуть надані пізніше. модуль, адреса якого збігається з адресою в команді, відповідає негайно. Таким чином, він обробляє конкретний запит, виданий контролером для керування пристроєм. Деталі технології X10 будуть надані пізніше. модуль, адреса якого збігається з адресою в команді, відповідає негайно. Таким чином, він обробляє конкретний запит, виданий контролером для керування пристроєм. Деталі технології X10 будуть надані пізніше.

Інстеон: Технологія Insteon [16] — це дводіапазонна мережева топологія, яка використовує лінії живлення та радіочастотний (РЧ) протокол для зв'язку та автоматизації домашніх електронних пристроїв і приладів, які зазвичай працюють незалежно. Це мережева технологія домашньої автоматизації, винайдена компанією Smart Labs, Inc. Insteon було розроблено на основі моделі X10 для додатків керування та вимірювання вдома. Insteon призначений для об'єднання простих пристроїв у мережу за допомогою лінії електроживлення та/або радіочастоти (РЧ). Усі пристрої Insteon є одноранговими, тобто кожен пристрій може передавати, отримувати та повторювати будь-які повідомлення протоколу Insteon, не потребуючи головного контролера чи складного програмного забезпечення маршрутизації.

Z-хвиля: Z-Wave [17] — це бездротова екосистема нового покоління, яка дозволяє всій домашній електроніці спілкуватися один з одним за допомогою дистанційного керування. Він використовує прості, надійні радіохвилі малої потужності, які легко проходять через стіни, підлогу та шафи. Функціональність Z-Wave можна додати практично до будь-яких електронних пристроїв. Керувати домом Z-Wave можна дистанційно з ПК та Інтернету з будь-якої точки світу. Z-Wave об'єднує всю домашню електроніку в інтегровану бездротову мережу без складного програмування та нових кабелів. Будь-який пристрій із підтримкою Z-Wave можна легко додати до цієї мережі, а багато пристроїв без Z-Wave можна

зробити сумісними, просто підключивши їх до додаткового модуля Z-Wave. За лічені секунди пристрій підключається до мережі та може бездротово спілкуватися з іншими модулями та контролерами Z-Wave. Оскільки Z-Wave працює на власній унікальній частоті, він не створюватиме перешкод іншому бездротовому обладнанню вдома, наприклад бездротовим телефонам і маршрутизаторам Wi-Fi. Технологія Z-Wave надзвичайно доступна, забезпечуючи потужне керування будинком за невелику частку вартості звичайних технологій розумного будинку.

ZigBee: ZigBee [15] побудовано на глобальному стандарті 802.15.4 Інституту інженерів з електротехніки та електроніки (IEEE), подібному до стандартів, які регулюють Bluetooth і Wi-Fi. Відкриті стандарти заохочують інновації та конкуренцію, що знижує витрати. Але на відміну від мереж Bluetooth і Wi-Fi, які потребують центральних концентраторів, які розподіляють інформацію на розосереджені пристрої, ZigBee дозволяє пристроям утворювати сітчасті мережі, тобто всі пристрої в мережі оснащені радіостанціями та можуть спілкуватися один з одним, а не лише з контролер (пристрій, який виконує функцію центрального керування мережею). Домовласники вперше спробують ZigBee у вигляді адаптерів, до яких можна підключати лампи, стереосистеми та інші прилади. Адаптери активуються настінними бездротовими перемикачами або портативними пристроями, що означає, що весь будинок можна керувати одним дистанційним керуванням. Очікується, що ZigBee стане ключем до розумного будинку в майбутньому.

1.2. Аналіз технології X10

X10 дозволяє керувати приладами та освітленням у всьому будинку. Сигнали надсилаються через існуючу проводку в будинку між модулями для ввімкнення пристроїв. Модулі приладів — це прості адаптери, які підключаються до розетки, а потім можна підключити пристрій, яким потрібно керувати, до розетки на модулі. Потім контролер X10 вмикає модуль, і він дозволяє живленню повністю пройти до підключеного пристрою.

Користувач може керувати модулями кількома способами. Блок контролера X10 підключається до електричної розетки в будинку, і він передає повідомлення модулям. Альтернативним способом надсилання повідомлень до контролера є використання комп'ютера або мікроконтролера для передачі повідомлень до пристрою.

Контролеру X10 наказано виконати дію, і він надсилає керуюче повідомлення через електропроводку в будинку, а модуль із правильною адресою приймає це повідомлення. Кожен модуль має адресу, яку встановлює користувач. Адреса складається з літери між A і P, відомої як код будинку, і номера пристрою від 1 до 16, відомого як код пристрою. Це дає 256 можливих адрес, що дозволяє контролювати 256 різних модулів, а отже, до 256 різних пристроїв можна контролювати в одному будинку. Керуюче повідомлення може бути одного з кількох типів залежно від дії, яку потрібно виконати. Основні повідомлення: «Увімк.», «Вимк.», «Темніший» або «Ясніший». Також можна надсилати інші повідомлення, наприклад «Всі пристрої вимкнено», на які відповідатимуть усі модулі. Сигнали керування надсилаються у вигляді пакетів тривалістю 1 мілісекунду на частоті 120 кГц. Кожен пакет надсилається тричі, щоб переконатися, що він не буде скасований частотою 50 Гц змінного струму в лініях електропередачі. Модуль приладу, який використовується для керування приладами, містить електромеханічний релейний перемикач. Коли він увімкнено, це дозволяє струму проходити до підключеного пристрою. Модуль лампи, який використовується для керування освітленням, містить перемикач TRIAC, який контролює величину струму, що протікає через перемикач, дозволяючи приєднаному світлу затемнюватись. Через електроніку, необхідну для регулювання яскравості світла, лампові модулі можна використовувати з лампами потужністю до 300 Вт. Якщо прилад підключено до модуля лампи, через збільшення струму може виникнути пошкодження як приладу, так і модуля лампи. Тому модуль приладу використовується для керування іншими приладами. Однак лампу можна приєднати до модуля приладу [5].

Коди будинків і коди пристроїв: Як ми згадували раніше, коди будинків і коди пристроїв варіюються від A до P і від 1 до 16 відповідно, хоча вони не

дотримуються двійкової послідовності. Формат кодування цих кодів можна знайти в [10]. Наприклад, щоб вибрати пристрій (з кодом 1) будинку (з кодом А), нам потрібно вибрати двійкове значення 0110, а щоб увімкнути його, нам потрібно вибрати двійкове значення 0010.

Стандартна трансмісія:Передача X10 від ПК до інтерфейсу контролера зазвичай відноситься до передачі коду будинку та комбінації коду пристрою або передачі коду функції. Цей формат є типовим для всіх передач між ПК та інтерфейсом контролера з різницею в першій передачі з ПК.

Код пристрою використовується, коли надсилається адреса, а функція використовується для надсилання команди певної дії. Зауважте, що функція працює лише для пристроїв, адресованих однаковим кодом будинку.

Контрольна сума інтерфейсу та підтвердження ПК:Коли інтерфейс отримує передачу від ПК, він підсумовує всі байти, а потім повертає контрольну суму байтів. Якщо контрольна сума правильна, ПК має повернути значення 0x00, щоб вказати, що передача має відбутися. Однак якщо контрольна сума неправильна, ПК має знову спробувати передати ту саму комбінацію заголовка: коду та дочекатися нової контрольної суми.

Інтерфейс готовий до прийому: Після завершення передачі X10 інтерфейс надішле ПК 0x55, щоб вказати, що він знаходиться в стані «готовності».

X10 Прийоми:Щоразу, коли інтерфейс починає отримувати дані від лінії живлення, він автоматично встановлює сигнал послідовного дзвінка (RI), щоб ініціювати процедуру пробудження ПК. Після завершення прийому даних інтерфейс почне опитувати ПК, щоб завантажити його буфер даних (максимум 10 байт). Якщо ПК не відповідає, буфер даних інтерфейсу буде переповнено, і додаткові дані не зберігатимуться в буфері.

Сигнал опитування інтерфейсу:Для опитування ПК інтерфейс постійно надсилатиме 0x5A.

Цей сигнал повторюватиметься раз на секунду, доки ПК не відповість.

Реакція ПК на сигнал опитування:Щоб припинити опитування інтерфейсу та розпочати передачу даних, ПК має надіслати підтвердження 0XcC до сигналу опитування інтерфейсу.

Тьмянний або яскравий: Після тьмяного або яскравого коду ПК очікуватиме, що наступний байт буде зміною рівня яскравості. Модуль X10 має 201 дискретний рівень яскравості. Тому цей байт буде еквівалентним зміні яскравості $(n/210)*100\%$.

Пристрої X10: Типова система автоматизації X10 складається з модуля приймача та кількох модулів приймача. Трансивер, який може передавати команди та отримувати статус приймача, в основному взаємодіє з ПК або мікроконтролером через USB або послідовний порт і спілкується з модулями приймача через домашню лінію живлення. Модулі приймача — це наші побутові пристрої електронної пошти, такі як світло, вентилятор, обігрівач, датчик води, датчики безпеки тощо. Вони отримують команди від трансиверів X10 і діють відповідно до команди.

Передавачі X10 — це пристрої керування, що складаються з простих кнопкових консолей, настінних вимикачів, програмованих таймерів і контролерів або навіть домашнього ПК чи мікроконтролера. Ці передавачі надсилають коди сигналу через систему електропроводки будинку одним натисканням кнопки або автоматично до приймачів X10, які очікують почути свої індивідуальні коди. Існує кілька різних трансиверів X10, які можна використовувати для взаємодії з комп'ютером або мікроконтролером і керування іншими модулями приймача. CM15A і CM11A є одними з найбільш використовуваних трансиверів.

Кожен модуль приймача X10 має код будинку та код пристрою, які унікально ідентифікують його серед 256 різних можливих комбінацій. Коли модуль приймача чує свій код, який встановлює користувач, він відповідає, виконуючи відповідну команду, як-от увімкнути, вимкнути, затемнити або підвищити яскравість світла, увімкнути або вимкнути вентилятор, кавоварку, датчики безпеки та багато іншого. Вони також слухають сигнали All On або All Off, які дозволяють користувачеві вмикати або вимикати всі модулі одним натисканням кнопки. Приймачем може бути плагінний модуль, дротова електрична розетка або настінний вимикач верхнього світла. X10 пропонує різні типи приймачів: модулі ламп, модулі приладів, датчики безпеки тощо [6]. Нижче наведені їх короткі описи.

Модулі ламп X10:Модулі X10 Lamp дають нам можливість вмикати або вимикати лампи, зменшувати або посилювати рівень освітленості лампи з будь-якої точки будинку без додаткової проводки. Лампові модулі можуть підтримувати лише пристрої малої потужності, такі як лампи, і їх не слід використовувати для керування потужними пристроями, такими як обігрівачі, холодильники тощо. Найбільш поширеними ламповими модулями є LM 465, LM15 і LM14A.

Модулі пристроїв X10:Модулі приладів X10 використовуються для керування побутовою технікою високої потужності, як-от водонагрівачі, кондиціонери, осушувачі повітря, стереосистеми, телевізори, деякі кухонні плити та багато іншого. Доступні два різні модулі приладів на 15 ампер і 20 ампер відповідно. Загальними модулями приладів є AM486 і AM466. Щоб керувати приладом, нам просто потрібно підключити 2- або 3-контактний пристрій до модуля приладу, а потім підключити модуль приладу до будь-якої звичайної настінної розетки.

Датчики безпеки X10:Продукти X10 дозволяють користувачам забезпечити безпеку свого дому за допомогою різноманітних датчиків безпеки, таких як датчик руху EagleEye, датчик руху HawkEye, бездротові датчики руху тощо. Вони не лише забезпечують безпеку, але й можуть використовуватися для багатьох інших цілей. Наприклад, можна налаштувати вмикання світла, коли хтось входить у кімнату за допомогою датчиків руху. Бездротові датчики руху мають радіус дії майже 100 футів.

1.3. Постановка завдання проектування

Актуальність роботи 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 модулі 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 модулі системи дистанційного керування роботою побутових приладів.

1.4. Висновки до розділу

Автоматизація розумних будинків більше не існує як концепція дизайну для майбутнього покоління. У зв'язку зі швидким зростанням і зростанням тенденцій у технологіях інтелектуальних систем розвиток розумних будинків показав інноваційні тенденції в дизайні. Люди похилого віку, хворі на хворобу Альцгеймера та аносію та люди з обмеженими можливостями стикаються з багатьма труднощами, займаючись своїми справами вдома. Проблеми, з якими вони стикаються, особливо коли вони вдома самі, становлять загрозу їхній безпеці.

Це дослідження представляє систему управління розумним будинком, яка покращила б ручне управління домом. Потрібна менша кількість персоналу, одночасно підвищуючи загальну безпеку за рахунок інтеграції автоматичних побутових приладів на основі зчитування датчиків і кнопки інструкції користувача в розроблений інтерфейс веб-сайту. Автоматична функція, заснована на інформації датчика, дозволила системі керування працювати ефективно та ефективно. Концепція IP-мережевих додатків і пристроїв у домі дозволяє керувати побутовою технікою з будь-якого місця з ноутбука, мобільного телефону, планшета чи смарт-телевізора, за умови, що ці пристрої мають доступ до Інтернету. Веб-сайт забезпечує зручність для розробників і користувачів, особливо для людей з обмеженими можливостями

та літніх людей. Це дослідження надає захищений паролем веб-сайт для вирішення проблеми безпеки. Крім того, проблему батареї можна подолати, додавши сонячний зарядний пристрій для підзарядки слабкої батареї. Цей контролер сонячної зарядки також можна використовувати як резервний ресурс під час відключення електроенергії.

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*.

2.2. Огляд систем автоматизації тестування апаратно-програмних систем

Кожен проект унікальний і кожна команда має свої запити. Однак усіх нас поєднує бажання працювати з якісними інструментами, які економлять час і

дозволяють QA-фахівцям тестувати якісніше та швидше, в ідеалі, щоб *TMS* могла поєднувати ручне та автоматизоване тестування.

Було проаналізовано перевірені часом та нові системи управління тестуванням, які зараз популярні на ринку. Вибрали функції, які мають бути в *Test Management System* нашої мрії, порівняли можливості продуктів та вивчили відгуки користувачів. Ділимося списком інструментів, один з яких точно підійде вашій команді.

Тут немає рейтингу, кожен інструмент має свої переваги і недоліки. В основному інструменти тест-менеджменту платні, однак у всіх є безкоштовна пробна версія.

Що ми хочемо від зручного *Test Management System*?

Користувач *TMS* очікує побачити таке:

- Зручна установка та підтримка.
- Зручний та зрозумілий інтерфейс.
- Створення та управління проектами.
- Створення користувачів та проектних ролей для користувачів.
- Зручна інтеграція із автоматичними тестами.
- Робота із тестовими артефактами: тест-план, тест-кейс, чек-лист, загальні кроки.
- Версіонування тест-кейсу/чек-листа.
- Створення атрибутів/конфігурацій.
- Зрозуміла система звітності.
- Вбудована система баг-трекінгу.
- Можливість оповіщення колег усередині та поза системою.
- Можливість інтеграції з іншими інструментами.

Для чого потрібна *TMS*?

Вирішити завдання створення єдиної *TMS* для роботи з усією документацією проекту можна кількома способами:

1. Найдешевший спосіб - не морочитися і вибрати *Google Docs* для матриці трасування, а дефекти вести в *open-source* баг-трекері.

2. Інший спосіб - використовувати одну з популярних *TMS*-ок, інтегровану з баг-трекером компанії.

3. *Next-level* спосіб - вибрати *Test Management System*, виходячи зі специфіки проектів, обсягів завдань, типів документації та видів тестування, що використовуються.

Дуже важливо підійти до питання вибору *TMS* відповідально, адже для компанії ціна помилки може бути високою.

Популярні системи управління тестуванням на 2020 р

– *ALM Octane*

– *Test IT*

– *TestRail*

– *Zephyr*

– *Allure EE*

– *TM4J*

– *Qase*

– *PractiTest*

– *Testuff*

– *Azure*

– *MTM TFS*

– *Kualitee*

Розглянемо вибрані інструменти докладніше:

1. *ALM Octane*

ALM Octane - довгожитель серед систем управління та життєвим циклом продукту, та його тестуванні. Інструмент дозволяє здійснювати планування, створення, тестування та контроль на всіх етапах розробки. Складний у початковому освоєнні, але незамінний для компанії великої руки, де особлива увага приділяється деталям виробництва. Підтримано функціональність спільних кроків. Робота з автоматизованими тестами. Фактичний час проходження для кожного тестового сценарію. Реалізована функціональність вебхуків. Присутній внутрішній баг-трекер. Зручна система конструювання звітів.

Саме тому, що продукт вже обкатаний, в інтернеті є безліч мануалів і відеогайдів з налаштування та використання.

Можливості:

- Загальний доступ до бібліотек вимог та ресурсів
- Детальні відомості про код, тестування, управління ризиками та їх оцінку, а також про відповідність вимогам
- Швидкий доступ до показників, наприклад до даних про неусунені дефекти
- Швидке налаштування лабораторного середовища для усунення помилок конфігурації в середовищах *Agile*
- Робота з автоматизованими тестами
- Вебхукі
- Зручна система звітів
- Створення вимог та відстеження їх виконання на всіх етапах життєвого циклу додатку
- Аналітика на будь-який смак і колір: звіти, що гнучко настроюються.
- Інтеграція з 50+ інструментами

Безкоштовна пробна версія: 30 днів

Посилання на скачування

2. *Test IT*

Test IT - спритна російська *TMS*, яку виробляють тестувальники для тестувальників. Основна фішка - поєднання ручних та автотестів в одному інтерфейсі, що дуже сприяє об'єднанню *QA*-команди. Аналіз автотестів тепер можливий в одній системі із тестовою моделлю! Розробники програми приділяють велику увагу автоматизованому тестуванню, кожен тестовий випадок у бібліотеці тестів можна лінкувати з автотестами *API*. Правильно налаштована інтеграція з автотестами дозволяє стежити за прогонами та їх результатами безпосередньо з *TMS* у режимі реального часу. Ви зможете бачити, які автоматичні тести в процесі виконання, аналізувати їх результати та переглядати вихідний код безпосередньо з *Test IT*. При необхідності можна створити тест-ран

поза системою та заповнити його своїми автотестами без лінка з тестовими сценаріями.

Цей інструмент вирізняє продуманий і красивий інтерфейс. У середині системи можна створювати проекти і вести для кожного структуровану бібліотеку тестових випадків і чек-листів, операції, що часто повторюються, виділяються в загальні кроки. Інструмент гнучкий — у кожному проекті створюються додаткові атрибути користувача/конфігурації, розподіляються проектні ролі та права, що спрощує налаштування *TMS* під процеси компанії. *Test IT* допомагає менеджерам рівномірно розподіляти навантаження між тестувальниками і контролювати виконання робіт за допомогою запитів користувача та звітів. Також у рамках самої системи є можливість спілкуватися з колегами, не використовуючи сторонніх месенджерів. Є елемент гейміфікації та можливість збирати ачивки.

Test IT активно розвивається, бере фічареквести від своїх користувачів та забиває ними 40% беклогу, є грамотна техпідтримка.

Можливості:

- Управління тест-планами, тест-кейсами та чек-листами
 - Загальні кроки для повторюваних дій
 - Створення атрибутів/конфігурацій.
 - Автоматичне розподілення тестів на *QA* інженерів
 - Інтеграція автоматичних тестів за допомогою *API*
 - Створення тест-ранів поза системою з управлінням стану
 - Аналітика як з автоматичних, так і з ручних тестів
 - Внутрішній чат та вебхукі у зовнішні системи
 - Рольова модель та персоналізація
 - Гейміфікація та ачивки
 - Двостороння інтеграція з *JIRA*
 - Розширена функціональність *Public API*
 - Імпорт із інших систем управління тестуванням
- Безкоштовна пробна версія: Відкрита демо-версія на сайті

Посилання на скачування

3. *TestRail*

Це програмне забезпечення зручне як команд *QA*, так розробки. План тестування можна побудувати як за сценарієм гнучкої методології, так більш традиційного підходу. Інструмент дозволяє отримати уявлення про перебіг тестування у реальному часі. Ви можете створювати конфігуровані звіти за необхідними метриками. У нових версіях з'явилася інтеграція із *Assembla*. Також була підтримана функціональність внутрішнього чату та оповіщень у зовнішню систему.

Крім можливості розкладання своїх тестових сценаріїв за тест-ранами з їх подальшим розміщенням у тест-планах, ви можете використовувати таку сутність як *Milestone*, яка дозволить зручно налаштувати процес проходження тестування.

Можна налаштувати типізацію проекту для ведення тестової документації.

Можливості:

- Відстеження стану та результатів окремого тесту
- Порівняння результатів кількох тестів, конфігурацій та етапів
- Типізація проектів для ведення тестової документації
- Внутрішній чат та оповіщення у зовнішню систему
- Відстеження робочого навантаження команди для коригування завдань та ресурсів

– Розгорнуті звіти та метрики

– Широкі можливості налаштування, хмарні або локальні варіанти встановлення

– Інтеграція з *JIRA*, *Redmine*, *YouTrack*, *GitHub*, *Jenkins*, *Selenium* та *Visual Studio*

– Зручний *REST API*

Безкоштовна пробна версія: 30 днів

Посилання на скачування

4. *Zephyr*

Zephyr - це плагін для всієї відомої *JIRA*, який інтегрує тестування в проектний цикл, дозволяючи вам відстежувати якість програмного забезпечення та приймати рішення в стилі *go/no-go*. У нових версіях було підтримано роботу з автоматизованими тестами.

Тест кейси можуть створюватись, виконуватись та відстежуватись так само, як і будь-які інші завдання у *JIRA*. Для оптимальної фіксації процесу тестування є інтеграція з інструментами управління якістю, автоматизації, безперервної інтеграції та аналітики. Є можливість створення атрибутів користувача. Крім того, у продукту швидко відповідає техпідтримка.

Можливості:

- Посилання на *user-stories*, завдання, вимоги, дефекти
- Конфігурації деплою: у хмарі, на сервері, у дата-центрі
- Розширена інформація на дашбордах аналітики та *DevOps*
- Інтеграція з *JIRA*, *Confluence*, *Selenium*, *Jenkins* та *Bamboo*
- Автоматизовані тести
- Створення атрибутів користувача
- Відома система звітів.

Безкоштовна пробна версія: 30 днів

Посилання на скачування

5. *Allure EE*

Allure – це фреймворк для створення простих та зрозумілих звітів автотестів (для будь-якої мови), представляє собою інструмент, що дозволяє внести прозорість у процес створення та виконання функціональних тестів. Звіти *Allure* допомагають команді вирішити багато проблем і почати розмовляти однією мовою. *Allure Enterprise* став логічним продовженням підходу *automation first* та підтримує оновлення тестової документації з коду автотестів. Інструмент має безліч інтеграцій з фреймворками тестування та різними мовами програмування.

Можливості

- Інтеграція з *automation tools*
- Компіляція коду в кейси

- Детальні звіти з автотестів
- Поєднання роботи з мануальними та автоматизованими кейсами
- Багатопоточність автотестів
- Атрибути користувача
- Тайм-менеджмент (*TimeLine*)

Безкоштовна пробна версія: 30 днів

Посилання на скачування

6. *TM4J*

TM4J від *Adaptavist* – це додаток для управління тестуванням в *JIRA*, дозволяє вести тестову документацію в *JIRA*. Лінкування тестових сценаріїв і *issue* безпосередньо в *JIRA*. Підтримано роботу з автоматизованими тестами. Можливість об'єднання повторюваних кроків у загальний крок.

Можливості:

- Лінкування тестових сценаріїв та *issue* не виходячи з *JIRA*
- Робота з автоматизованими тестами
- Внутрішній багтрекер
- Зрозуміла система звітів
- Використання загального кроку
- Фактичний час проходження тесту
- Експорт даних в *Excel*

Безкоштовна пробна версія: 30 днів

Посилання на скачування

7. *Qase*

Qase це хмарна *TMS*, яка допоможе вашій команді підвищити продуктивність та організувати зручний флоу тестування програмного забезпечення. Підтримана функціональність об'єднання дій, що постійно повторюються, в загальний крок, імпорт даних з інших *TMS*. Управління ролями та дозволами для користувачів системи. У новій версії було доопрацьовано систему звітів.

Можливості:

- Тестовий репозиторій: вибудовування тестів у логічні групи
- Складання кроків для кейсів, встановлення пріоритету та серйозності
- Запуск тестових прогонів з трекінгом часу по кожному тесту
- Зберігання документації за проектом
- Автоматичний заклад дефектів в інтегровані трекери
- Інтеграція з *JIRA*, *Redmine*, *YouTrack* та *Slack*
- Об'єднання результатів автотестів із *REST API*

Безкоштовно для невеликих компаній

Посилання на скачування

8. *PractiTest*

PractiTest – це комплексний засіб управління тестами. Воно дає повну картину процесу тестування та глибше розуміння результатів тестування. Цей інструмент допоможе організувати тест-сьюти відповідно до ваших циклів та спринтів. Тестові набори можна формувати за різними критеріями, такими як компоненти, версії або типи. Тул заточений на *Agile*-тестування, регресійне тестування, тестування мікросервісів та *DevOps*.

У нових версіях було доопрацьовано функціональність роботи з автоматизованими тестами.

У техпідтримці працюють навчені *QA*-співробітники, які можуть швидко зрозуміти вашу проблему.

Можливості:

- Легке додавання тестів нових фіч до регресійного тестування
- Угруповання тестів на основі мікросервісів, які вони охоплюють, навіть крос-сервісні
- Різне відображення інформації для різних груп користувачів
- Дашборди в реальному часі показують стан тестів, прогонів на етапах розробки і при деплої
- Інтеграція з *JIRA*, *Redmine*, *Jenkins*, *GitLab* та *Slack*

Безкоштовна пробна версія: 14 днів

Посилання на скачування

9. *Testuff*

Команда *Testuff* робить дійсно зручний інструмент, дана *TMS* намагається об'єднати в собі всі методи тестування, починаючи від *waterfall model* і закінчуючи *black box testing*. Розробники *Testuff* окремо виділили свій продукт як єдину *TMS*, яку можна використовувати на будь-якому девайсі: смартфони, планшети тощо. д

У даному рішенні є все необхідне для *TMS*: класний інтерфейс, зрозумілий і зручний спосіб ведення тестової документації, конфігурації та атрибути користувача, два способи інтеграції з будь-яким існуючим баг-трекером, а також додали відеорекодер.

Можливості:

- Планування тестів
- Інтуїтивний *drag-n-drop* інтерфейс
- Наочні звіти з детальними графіками
- Два способи інтеграції зі сторонніми інструментами баг-трекінгу
- Можливість тестувати з будь-якого девайсу

Безкоштовна пробна версія: 30 днів

Посилання на скачування

10. *Azure DevOps Server*

Це потужний інструмент роботи з тестами та автотестами, за рахунок своєї комплексності ви можете налаштувати свій робочий простір, як вам потрібно. Працюйте безпосередньо зі своїми *CI/CD* сервісами, інтегруйте свої репозиторії прямо в *Azure*, ведіть тестову документацію зі спринтів, які розкладатимете по бордах, робіть максимально детальні звіти з вашої тестової документації та результатів її проходження.

Окремо варто згадати можливість інтеграції з *IDE* від компанії *Microsoft*, ви можете редагувати та налаштовувати свій код прямицінько через *Azure* та інтегруватися з усілякими системами від компанії *Microsoft*.

Можливості:

- Інтеграція з будь-яким продуктом *Microsoft*
- Нативний інтерфейс
- Інтеграція з будь-яким *CI/CD*

– Ведення зручних *Dashboards*

– Робота з автотестами

– Атрибути користувача

Безкоштовна пробна версія: 30 днів

Посилання на скачування

11. *MTM TFS*

Team Foundation Server (TFS) - комплексне рішення від *Microsoft*, яке включає систему управління версіями, збір даних, побудова звітів, відстеження статусів і змін по проекту.

Microsoft Test Manager є частиною цього продукту і вимагає інсталяції *Visual Studio*. Таке поєднання дає можливість пов'язати завдання, поставлені перед тестувальником, із заведеними дефектами та звітами про витрачений на роботу час.

Плани та результати тестування зберігаються на сервері *Team Foundation Server*. *MTM* включає тест-план, тест-кейс і конфігурації.

Сам *TFS* є пропрієтарним ПЗ, ліцензія - комерційна. Працює на трьох рівнях: клієнтський рівень, прикладний рівень та рівень даних, залежно від чого можлива робота або через *web* або через десктоп-додаток. *MTM* працює тільки на прикладному рівні, тому потрібна установка на сервер (якщо віддалений сервер, робота проводиться через *VPN*).

Можливості:

– Дослідницьке тестування

– Планування та виконання ручних тестів

– Кросплатформні конфігурації тестів (різні версії одного тесту для різних платформ/релізів)

– Діагностика проходження тесту (логи, відео тощо)

– Імпорт-експорт тестів

– Міжпроектний імпорт-експорт тестів

– Запис та відтворення ручних тестів (рекордер)

– Автоматизація тестів

Безкоштовна пробна версія: 30 днів

Посилання на скачування

12. *Kualitee*

Kualitee - продукт *Kualitatem*, компанії з тестування програмного забезпечення та інформаційної безпеки, що спеціалізується на забезпеченні безпомилкової роботи програм всюди.

Kualitee пропонує функції керування проектами, керування тестуванням, керування дефектами з інтеграцією з баг-трекерами. Гнучка система атрибутів користувача дозволяє дуже точно налаштувати необхідний робочий простір. Додатково є можливість глибокого налаштування профілів користувача та його прав доступу.

Можливості:

- Управління проектами
- Управління дефектами
- Управління тестовою документацією
- Персональна панель інструментів
- Продумане налаштування користувачів

2.3. Структура системи дистанційного керування побутовими приладами

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*.

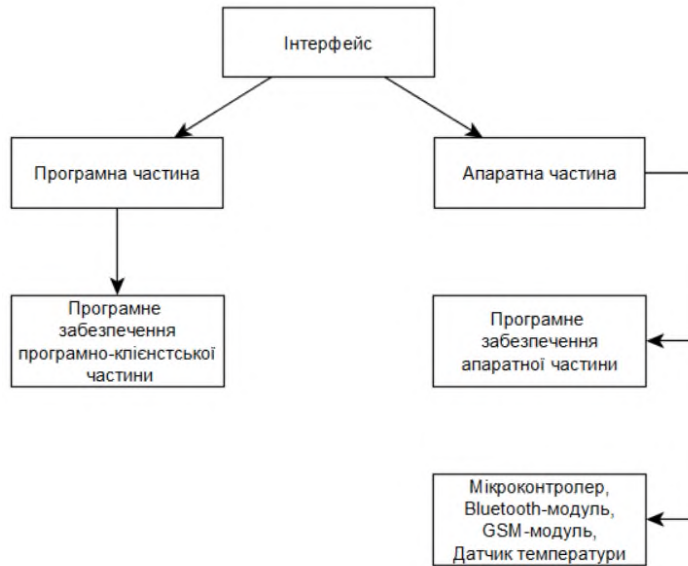


Рис. 2.1. Принцип побудови системи дистанційного керування роботою чайника

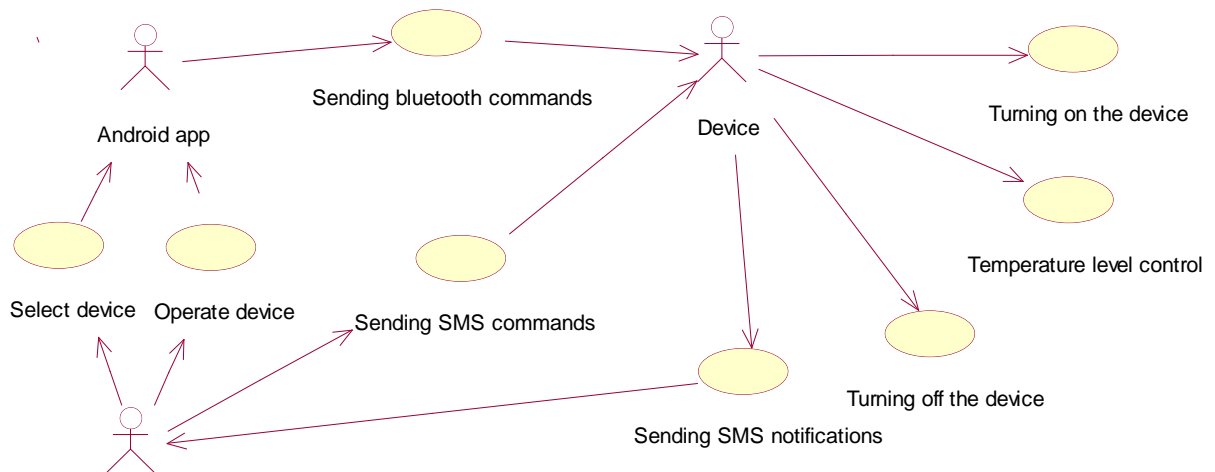


Рис. 2.2. Діаграма прецедентів

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).

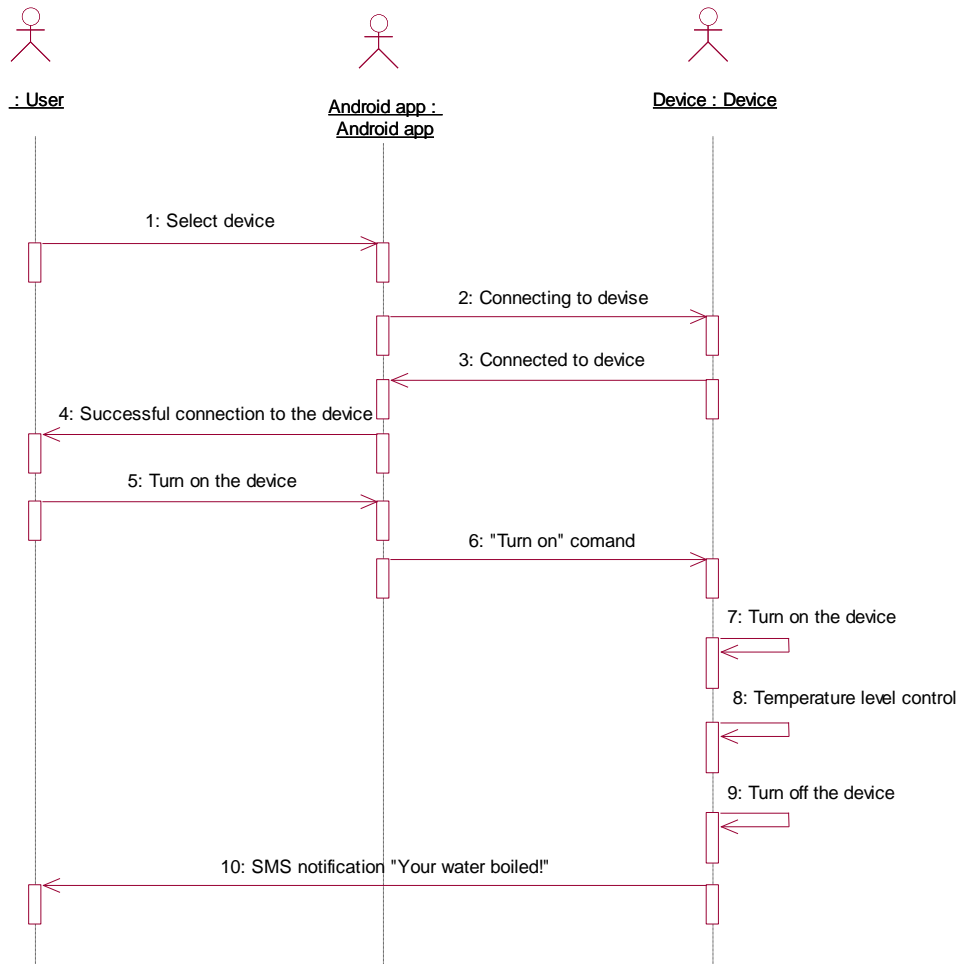


Рис. 2.3. Діаграма послідовності для керування пристроєм через додаток

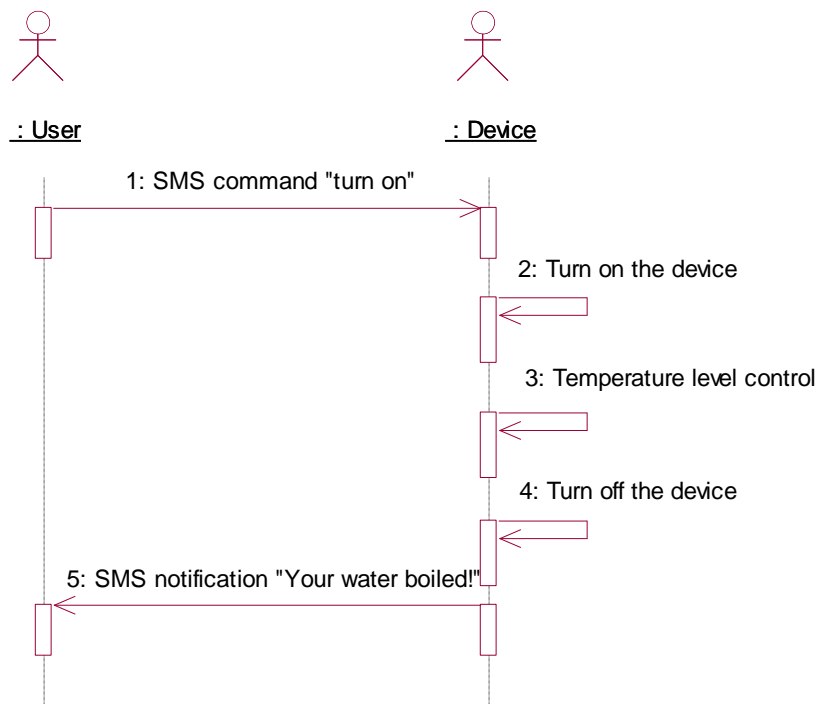


Рис. 2.4. Діаграма послідовності для керування пристроєм через SMS

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

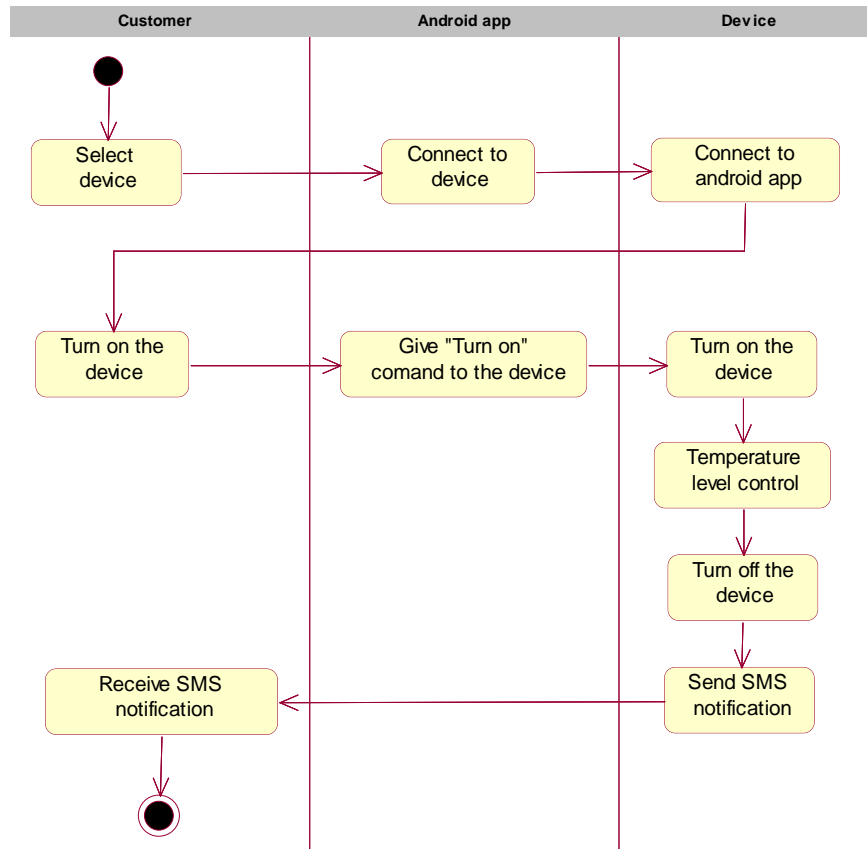


Рис. 2.5. Діаграма взаємодії

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

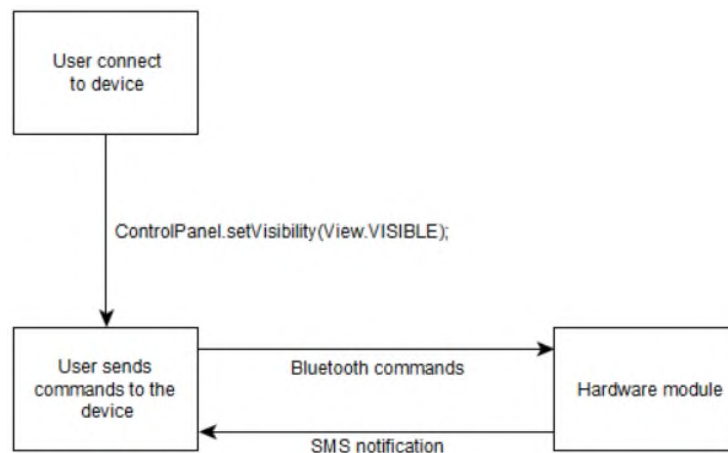


Рис. 2.6. Функціональна схема системи

2.4. 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*.

Система керування побутовою технікою (*HACS*) — це система, яка керується дистанційною системою, як-от мобільний телефон або кишеньковий пристрій, і в той же час контролює, контролює та координує побутову техніку, таку як кондиціонер, мікрохвильова піч, гараж. двері, телевізор, відеомагнітофон, аудіо контролер, внутрішнє/зовнішнє освітлення, спринклер, система безпеки будинку, контролер ванни тощо.

Щоб активувати побутову техніку та дозволити різні способи приготування їжі, *HACS* потребує механізмів для зв'язку між різними пристроями в системі та для координації між різними процесами, що виконуються на таких пристроях.

HACS також потребує механізмів адаптації до різних потреб користувача. Наприклад, коли користувач дуже голодний, мікрохвильовій печі може знадобитися відповісти на запит користувача, щоб вона працювала максимально, щоб приготувати їжу якнайшвидше. Для іншого прикладу, якщо користувач голодний, втомлений і може повернутися додому пізно, тоді системі може бути запропоновано повністю приготувати їжу до очікуваного часу прибуття та періодично розігріватися кожні 10 хвилин після цього.

2.3.1. Нефункціональна вимога

HACS має бути адаптованим. Будь-яка зміна в середовищі повинна бути виявленою, і *HACS* має бути трансформованим шляхом розпізнавання необхідної зміни в *HACS*, і розпізнана зміна має бути введена в дію в системі. Система може адаптуватися автоматично або вручну. Оскільки швидкість має першочергове значення для будь-якої системи реального часу, їй слід надавати пріоритет, який можна порівняти з вимогою адаптивності. Імовірно, це відповідно до потреб клієнта. Виявлення змін у навколишньому середовищі зазвичай займає дуже багато часу, а отже, це шкодить швидкості. Так само виявлення подій може спричинити значне зниження продуктивності.

HACS також має бути безпечним. Наприклад, мікрохвильова піч тепер має здутися або стати занадто гарячою, щоб доторкнутися до неї.

Діаграма варіантів використання

Діаграми варіантів використання призначені для моделювання функціональних вимог системи. Він показує набір варіантів використання та акторів та їхні відносини. Його завжди вибирають як початкову точку розробки програмного забезпечення. На наступному малюнку показано діаграму варіантів використання системи *HACS* з точки зору кінцевого користувача.

В даному розгортанні системи передбачені наступні актори:

– користувач, який означає особу, яка має обліковий запис і пароль, може увійти в *HACS* і дистанційно керувати побутовою технікою за допомогою *HACS*.

– адміністратор — це різновид користувача, який має спеціальні права, наприклад Додати/Видалити пристрій і Додати/Видалити віддалену систему, крім основних операцій.

– пристрій – відноситься до комп'ютерної вбудованої побутової техніки, яку можна додати до *HACS*, налаштувати та керувати дистанційно.

– телевізор – це спеціальний побутовий прилад, який має кілька операцій, як-от увімкнення, вимкнення, збільшення гучності, зменшення гучності та перемикання каналів, якими можна керувати дистанційно.

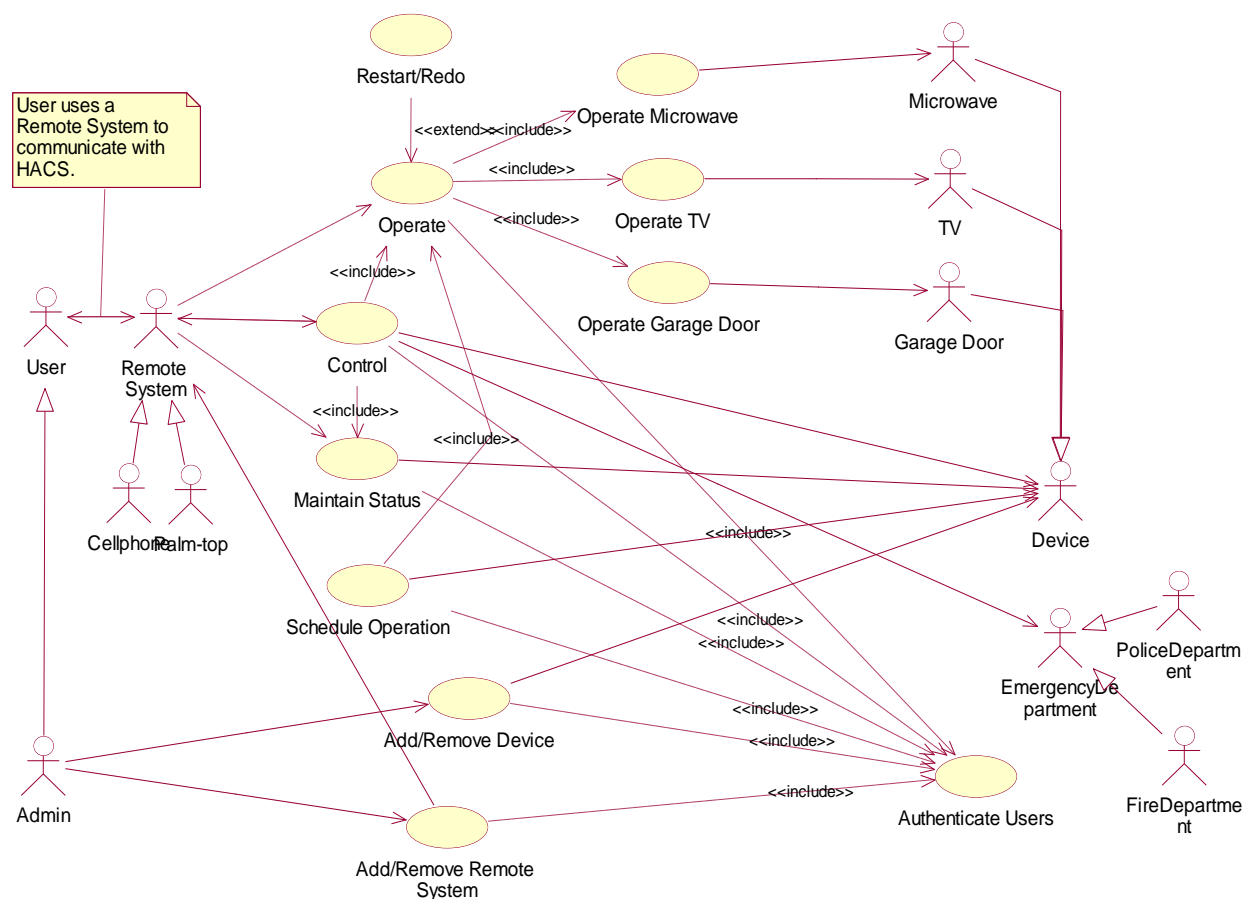


Рис. 2.7. Діаграма використання

– гаражні ворота - це спеціальний побутовий прилад, який має дистанційне керування операціями відкриття та закриття.

– мікрохвильова піч — це спеціальний побутовий прилад, який має кілька операцій, як-от команда налаштування, встановлення таймера, запуск тощо, якими можна керувати дистанційно.

– віддалена система відноситься до системи, яка має бездротове підключення. Її можна додати до системи *HACS*, налаштувати за допомогою *HACS* і обмінюватися даними з *HACS*.

– долоня – *Palm-top* – спеціальна дистанційна система.

– мобільний телефон - це спеціальна дистанційна система.

– відділення невідкладної допомоги

– департамент швидкої допомоги - це державний відділ, який може допомогти, якщо в будинку трапиться надзвичайна ситуація.

– департамент поліції/ Поліцейський відділ - це спеціальний відділ екстреної допомоги, який буде помічений, коли вламуються через двері гаража.

– пожежна частина - це спеціальна служба екстреної допомоги, яку помітять, коли в будинку виникне пожежа.

2.3.2. Випадки використання

Варіант використання *Operate* — це послуга, яка надається користувачам для роботи на пристроях, підключених до *HACS* через віддалені пристрої. Наприклад, користувач може надіслати повідомлення про ввімкнення телевізора, запуск мікрохвильової печі для приготування протягом п'яти хвилин і так далі до *HACS*, а потім *HACS* виконає дію.

Перезапустити/повторити

Варіант використання *Restart/Redo* є розширенням варіанту використання *Operate*. Якщо *HACS* не може виконати команду, надіслану користувачем, він повинен виявити та виправити помилку або перезапустити пристрій, щоб він працював нормально. І операція буде проведена ще раз. Крім того, *HACS* інформує користувача про цей виняток.

Увімкніть мікрохвильову піч

Варіант використання *Operate Microwave* — це послуга, яка надається користувачам для роботи з мікрохвильовою піччю, підключеною до *HACS* через віддалені системи. Користувачі можуть надсилати повідомлення про приготування, розморожування, зупинку тощо.

Керуйте дверима гаража

Варіант використання *Operate Garage Door* – це послуга, яка надається користувачам для роботи з дверима гаража, які підключені до *HACS* через віддалені системи. Користувачі можуть надсилати повідомлення про його відкриття та закриття.

Керувати телевізором

Варіант використання *Operate TV* – це послуга, яка надається користувачам для керування телевізором, підключеним до *HACS* через віддалені системи.

Користувачі можуть надсилати повідомлення про ввімкнення, вимкнення, збільшення гучності, зменшення гучності та зміну каналів.

Перевірити/оновити статус

Варіант використання *Check/Update Status* означає, що *HACS* зберігає записи про стан усіх підключених пристроїв. Він час від часу оновлює їхні стани. Коли користувачам потрібна інформація про стан якогось пристрою через віддалені системи, *HACS* надсилає останній стан цього пристрою до віддаленої системи, яка запитувала.

Контроль

Іноді пристрій може працювати несправно. Наприклад, мікрохвильова піч може готувати протягом години, що не потрібно користувачам. Варіант використання *Control* означає, що він може виявити таку несправність, оновивши стан пристроїв, зупинити його (за допомогою оператора сценарію використання) та повідомити користувача про те, що сталося. В іншому випадку може статися якась надзвичайна ситуація, наприклад, проникнення через двері гаража або початок пожежі, *HACS* має виявити це та повідомити поліцейську та пожежну службу відповідно.

Додати/Видалити пристрій

Варіант використання *Add/Remove Device* — це послуга, за допомогою якої користувачі можуть додавати пристрій, наприклад, мікрохвильову піч, телевізор і двері гаража, до *HACS* для дистанційного й автоматичного керування, а також видаляти пристрій із *HACS*, який більше не підлягає контролю. Коли пристрій підключено до *HACS*, і *HACS*, і пристрій налаштовані для взаємодії один з одним.

Додати/Видалити віддалену систему

Варіант використання «*Add/Remove Remote System*» — це послуга, за допомогою якої користувачі можуть додавати або видаляти систему дистанційного керування, наприклад, стільниковий телефон або кишеньковий пристрій, у *HACS*, щоб віддалену систему можна було використовувати для зв'язку з *HACS*. і керувати пристроями.

Діаграма класів

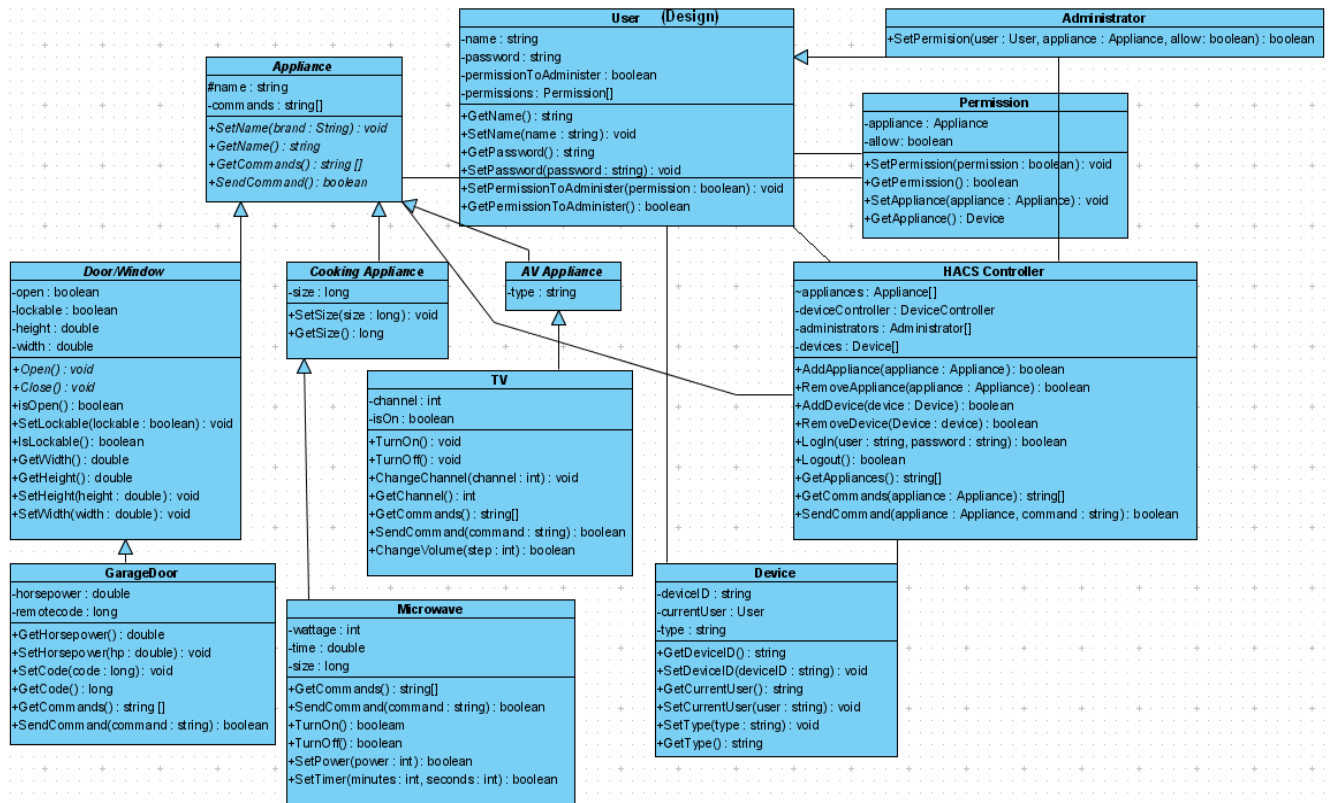


Рис. 2.8. Діаграма та опис класів *UML*

Клас *Appliance* повинен діяти як загальний батьківський клас для всіх різних типів пристроїв. Використання спільного батьківського класу дозволяє забезпечити певний інтерфейс між контролером *HACS* і пристроями, які необхідно застосовувати. Застосування спільного інтерфейсу гарантує, що для додавання пристрою в систему не потрібно перекомпілювати всю систему, а навпаки, пристрої можна додавати або видаляти з системи динамічно, просто додаючи скомпільований файл класу пристрою до системного каталогу, а потім вказати системі додати пристрій до системи.

Клас дверей/вікон, кухонних приладів і аудіо/відеоприладів

Класи *Door/Window*, *Cooking Appliance* та *AV Appliance* служать прикладом шару типів приладів класів. Ці класи безпосередньо успадковують клас *Appliance*, але самі по собі не представлятимуть фізичні пристрої. Швидше, цей рівень служить для забезпечення спільного інтерфейсу та загальних властивостей для певного типу пристроїв.

Гаражні двері, мікрохвильова піч та класи пристроїв

Класи *GarageDoor*, *Microwave* і *Device* є прикладами класів, які представляють реальні фізичні пристрої. Ці класи успадковують від відповідного класу типу пристрою, а отже, від класу *Appliance*. Ці класи будуть містити фактичну реалізацію функції пристрою. Оскільки всі ці класи успадковуються від одного класу *Appliance*, усі вони можуть бути проіндексовані системою *HACS* і можуть надсилатися команди таким же чином.

Клас контролера *HACS* є серцем системи. Це клас, який керує приладами, пристроями та користувачами. Контролер *HACS* містить масив пристроїв, який є списком усіх пристроїв у системі. Команду можна надіслати до пристрою, просто викликавши метод *SendCommand(string)* цього пристрою. Кожен метод *Appliance SendCommand(string)* може інтерпретувати команду, вбудовану в рядковий параметр, і викликати відповідні внутрішні методи. А метод *Appliance GetCommands()* повертає список усіх дійсних команд для цього *Appliance*, який потім можна передати користувачеві. Клас *HACS Controller* також керує взаємодією між системою та віддаленими пристроями, а також керує входом і виходом користувачів із системи. Крім того, він керує додаванням і видаленням користувачів і пристроїв до/з системи.

Клас пристрою – представляє фізичний пристрій, який використовується для віддаленого адміністрування системи. Він містить інформацію про сам пристрій і користувача, який зараз використовує пристрій.

Клас користувача – клас *User* представляє користувача. Він містить ім'я та пароль користувача, незалежно від того, має він дозвіл на адміністрування системи, а також масив дозволів, що вказує, якими пристроями користувач може чи ні.

Клас адміністратора – екземпляр класу дозволів містить одне посилання між користувачем і пристроєм і вказує, чи дозволено цьому користувачеві використовувати цей пристрій.

Клас дозволу – екземпляр класу дозволів містить одне посилання між користувачем і пристроєм і вказує, чи дозволено цьому користувачеві використовувати цей пристрій.

2.3.3. Нефункціональні вимоги

HACS має бути адаптованим. Будь-яка зміна в середовищі повинна бути виявленою, і *HACS* має бути трансформованим шляхом розпізнавання необхідної зміни в *HACS*, і розпізнана зміна має бути введена в дію в системі. Система може адаптуватися автоматично або вручну. Оскільки швидкість має першочергове значення для будь-якої системи реального часу, їй слід надавати пріоритет, який можна порівняти з вимогою адаптивності. Імовірно, це відповідно до потреб клієнта. Виявлення змін у навколишньому середовищі зазвичай займає дуже багато часу, а отже, це шкодить швидкості. Так само виявлення подій може спричинити значне зниження продуктивності.

HACS також має бути безпечним. Наприклад, мікрохвильова піч тепер має здутися або стати занадто гарячою, щоб доторкнутися до неї.

Діаграми варіантів використання призначені для моделювання функціональних вимог системи. Він показує набір варіантів використання та акторів та їхні відносини. Його завжди вибирають як початкову точку розробки програмного забезпечення. На наступному малюнку показано діаграму варіантів використання системи *HACS* з точки зору кінцевого користувача.

Актори

– Користувач означає особу, яка має обліковий запис і пароль, може увійти в *HACS* і дистанційно керувати побутовою технікою за допомогою *HACS*.

– Адміністратор — це різновид користувача, який має спеціальні права, наприклад Додати/Видалити пристрій і Додати/Видалити віддалену систему, крім основних операцій.

– Пристрій відноситься до комп'ютерної вбудованої побутової техніки, яку можна додати до *HACS*, налаштувати та керувати дистанційно.

– Телевізор — це спеціальний побутовий прилад, який має кілька операцій, як-от увімкнення, вимкнення, збільшення гучності, зменшення гучності та перемикання каналів, якими можна керувати дистанційно.

– Гаражні ворота - це спеціальний побутовий прилад, який має дистанційне керування операціями відкривання та закривання.

Мікрохвильова піч — це спеціальний побутовий прилад, який має кілька операцій, як-от команда налаштування, встановлення таймера, запуск тощо, якими можна керувати дистанційно.

Віддалена система відноситься до системи, яка має бездротове підключення. Її можна додати до системи *HACS*, налаштувати за допомогою *HACS* і обмінюватися даними з *HACS*.

Palm-top – спеціальна дистанційна система.

Мобільний телефон - це спеціальна дистанційна система.

Департамент швидкої допомоги - це державний відділ, який може допомогти, якщо в будинку трапиться надзвичайна ситуація.

Поліцейський відділ - це спеціальний відділ екстреної допомоги, який буде помічений, коли вламуються через двері гаража.

Пожежна частина - це спеціальна служба екстреної допомоги, яку помітять, коли в будинку виникне пожежа.

Варіант використання *Operate* — це послуга, яка надається користувачам для роботи на пристроях, підключених до *HACS* через віддалені пристрої. Наприклад, користувач може надіслати повідомлення про ввімкнення телевізора, запуск мікрохвильової печі для приготування протягом п'яти хвилин і так далі до *HACS*, а потім *HACS* виконає дію.

Варіант використання *Restart/Redo* є розширенням варіанту використання *Operate*. Якщо *HACS* не може виконати команду, надіслану користувачем, він повинен виявити та виправити помилку або перезапустити пристрій, щоб він працював нормально. І операція буде проведена ще раз. Крім того, *HACS* інформує користувача про цей виняток.

Варіант використання *Operate Microwave* — це послуга, яка надається користувачам для роботи з мікрохвильовою піччю, підключеною до *HACS* через віддалені системи. Користувачі можуть надсилати повідомлення про приготування, розморожування, зупинку тощо.

Варіант використання *Operate Garage Door* – це послуга, яка надається користувачам для роботи з дверима гаража, які підключені до *HACS* через

віддалені системи. Користувачі можуть надсилати повідомлення про його відкриття та закриття.

Варіант використання *Operate TV* – це послуга, яка надається користувачам для керування телевізором, підключеним до *HACS* через віддалені системи. Користувачі можуть надсилати повідомлення про ввімкнення, вимкнення, збільшення гучності, зменшення гучності та зміну каналів.

Варіант використання *Check/Update Status* означає, що *HACS* зберігає записи про стан усіх підключених пристроїв. Він час від часу оновлює їхні стани. Коли користувачам потрібна інформація про стан якогось пристрою через віддалені системи, *HACS* надсилає останній стан цього пристрою до віддаленої системи, яка запитувала.

Іноді пристрій може працювати несправно. Наприклад, мікрохвильова піч може готувати протягом години, що не потрібно користувачам. Варіант використання *Control* означає, що він може виявити таку несправність, оновивши стан пристроїв, зупинити його (за допомогою оператора сценарію використання) та повідомити користувача про те, що сталося. В іншому випадку може статися якась надзвичайна ситуація, наприклад, проникнення через двері гаража або початок пожежі, *HACS* має виявити це та повідомити поліцейську та пожежну службу відповідно.

Варіант використання *Add/Remove Device* — це послуга, за допомогою якої користувачі можуть додавати пристрій, наприклад, мікрохвильову піч, телевізор і двері гаража, до *HACS* для дистанційного й автоматичного керування, а також видаляти пристрій із *HACS*, який більше не підлягає контролю. Коли пристрій підключено до *HACS*, і *HACS*, і пристрій налаштовані для взаємодії один з одним.

Варіант використання «*Add/Remove Remote System*» — це послуга, за допомогою якої користувачі можуть додавати або видаляти систему дистанційного керування, наприклад, стільниковий телефон або кишеньковий пристрій, у *HACS*, щоб віддалену систему можна було використовувати для зв'язку з *HACS*. і керувати пристроями.

Клас *Appliance* повинен діяти як загальний батьківський клас для всіх різних типів пристроїв. Використання спільного батьківського класу дозволяє

забезпечити певний інтерфейс між контролером *HACS* і пристроями, які необхідно застосовувати. Застосування спільного інтерфейсу гарантує, що для додавання пристрою в систему не потрібно перекомпілювати всю систему, а навпаки, пристрої можна додавати або видаляти з системи динамічно, просто додаючи скомпільований файл класу пристрою до системного каталогу, а потім вказати системі додати пристрій до системи.

Класи *Door/Window*, *Cooking Appliance* та *AV Appliance* служать прикладом шару типів приладів класів. Ці класи безпосередньо успадковують клас *Appliance*, але самі по собі не представлятимуть фізичні пристрої. Швидше, цей рівень служить для забезпечення спільного інтерфейсу та загальних властивостей для певного типу пристроїв.

Класи *GarageDoor*, *Microwave* і *Device* є прикладами класів, які представляють реальні фізичні пристрої. Ці класи успадковують від відповідного класу типу пристрою, а отже, від класу *Appliance*. Ці класи будуть містити фактичну реалізацію функції пристрою. Оскільки всі ці класи успадковуються від одного класу *Appliance*, усі вони можуть бути проіндексовані системою *HACS* і можуть надсилатися команди таким же чином.

Клас контролера *HACS* є серцем системи. Це клас, який керує приладами, пристроями та користувачами. Контролер *HACS* містить масив пристроїв, який є списком усіх пристроїв у системі. Команду можна надіслати до пристрою, просто викликавши метод *SendCommand(string)* цього пристрою. Кожен метод *Appliance SendCommand(string)* може інтерпретувати команду, вбудовану в рядковий параметр, і викликати відповідні внутрішні методи. А метод *Appliance GetCommands()* повертає список усіх дійсних команд для цього *Appliance*, який потім можна передати користувачеві. Клас *HACS Controller* також керує взаємодією між системою та віддаленими пристроями, а також керує входом і виходом користувачів із системи. Крім того, він керує додаванням і видаленням користувачів і пристроїв до/з системи.

Клас пристрою представляє фізичний пристрій, який використовується для віддаленого адміністрування системи. Він містить інформацію про сам пристрій і користувача, який зараз використовує пристрій.

Клас *User* представляє користувача. Він містить ім'я та пароль користувача, незалежно від того, має він дозвіл на адміністрування системи, а також масив дозволів, що вказує, якими пристроями користувач може чи ні.

Екземпляр класу дозволів містить одне посилання між користувачем і пристроєм і вказує, чи дозволено цьому користувачеві використовувати цей пристрій.

Екземпляр класу дозволів містить одне посилання між користувачем і пристроєм і вказує, чи дозволено цьому користувачеві використовувати цей пристрій.

Щоб керувати системою *HACS*, користувач повинен керувати кожним побутовим приладом або за допомогою прямого контакту, або за допомогою системи дистанційного керування. Діаграма послідовності пояснює роботу системи дистанційного керування. Для підключення до віддаленої системи користувачеві потрібні вбудовані в комп'ютер пристрої зв'язку, такі як мобільний телефон, КПК або портативний комп'ютер.

Хід подій:

1. Після підключення користувача до віддаленої системи йому необхідно увійти в систему, використовуючи дійсне ім'я користувача та пароль.

- Спроба входу обмежена 3 разами з міркувань безпеки.

2. Після всього використання системи користувач повинен вийти з системи, і зв'язок між користувачем і віддаленою системою буде розірвано.

Керуйте дверима гаража

Хід подій:

1. Після входу користувача в систему віддалена система запитує у контролера *HACS* список пристроїв.

2. Користувач просто обирає гаражні ворота зі списку приладів.

3. Дистанційна система запитує список команд гаражних воріт.

4. Контролер *HACS* запитує двері гаража список команд.

5. Двері гаража повертають список команд до контролера *HACS*, потім контролер *HACS* повертає список до віддаленої системи, а потім віддалена система відображає список команд

6. Користувач вибирає одну зі списку команд, і контролер *HACS* виконує операцію.

Керувати телевізором

Хід подій:

1. Після входу користувача в систему віддалена система запитує у контролера *HACS* список пристроїв.

2. Користувач просто вибирає телевізор зі списку пристроїв.

3. Дистанційна система запитує список команд телевізора.

4. Контролер *HACS* запитує у телевізора список команд.

5. Двері гаража повертають список команд до контролера *HACS*, потім контролер *HACS* повертає список до віддаленої системи, а потім віддалена система відображає список команд

6. Користувач вибирає одну зі списку команд, і контролер *HACS* виконує операцію.

Увімкніть мікрохвильову піч

Хід подій:

1. Після входу користувача в систему віддалена система запитує у контролера *HACS* список пристроїв.

2. Користувач просто обирає мікрохвильову піч зі списку приладів.

3. Дистанційна система запитує список команд мікрохвильової печі.

4. Контролер *HACS* запитує мікрохвильову піч список команд.

5. Двері гаража повертають список команд до контролера *HACS*, потім контролер *HACS* повертає список до віддаленої системи, а потім віддалена система відображає список команд

6. Користувач вибирає одну зі списку команд, і контролер *HACS* виконує операцію.

Діаграма послідовності

Щоб керувати системою *HAC*, користувач повинен керувати кожним побутовим приладом або за допомогою прямого контакту, або за допомогою системи дистанційного керування. Діаграма послідовності пояснює роботу системи дистанційного керування. Для підключення до віддаленої системи

користувачеві потрібні вбудовані в комп'ютер пристрої зв'язку, такі як мобільний телефон, КПК або портативний комп'ютер.

Аутентифікація

Хід подій:

Після підключення користувача до віддаленої системи йому необхідно увійти в систему, використовуючи дійсне ім'я користувача та пароль.

Спроба входу обмежена 3 разами з міркувань безпеки.

Після всього використання системи користувач повинен вийти з системи, і зв'язок між користувачем і віддаленою системою буде розірвано.

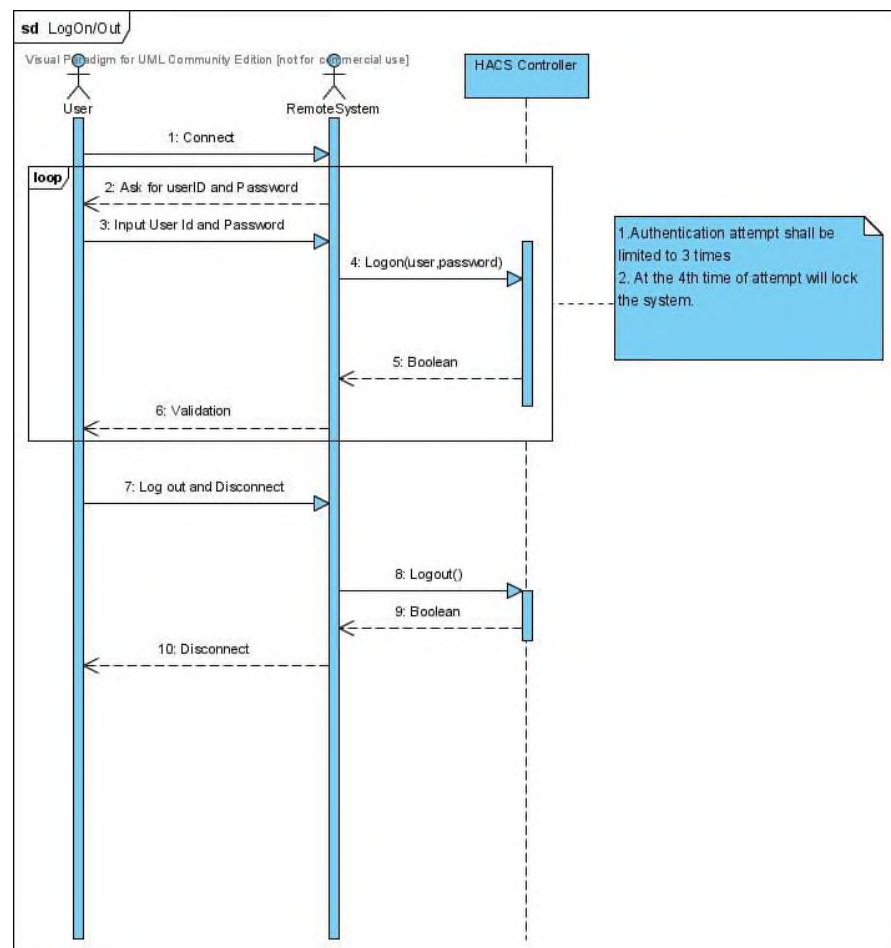


Рис. 2.9. Аутентифікація

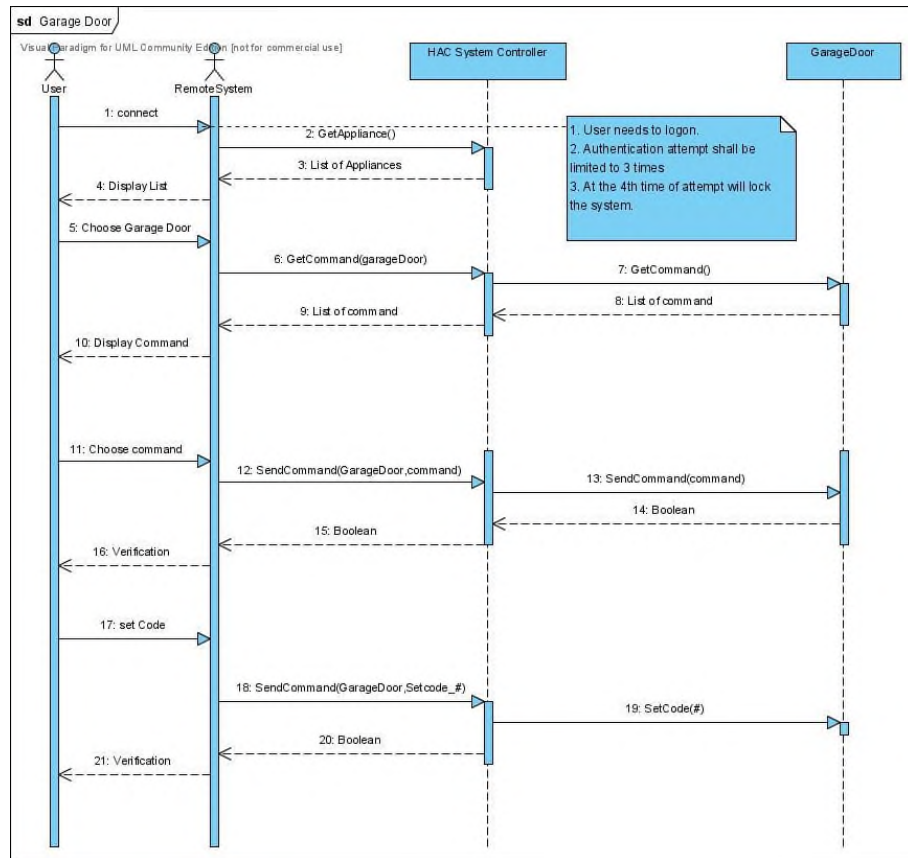


Рис. 2.10. Керування дверима гаража

Хід подій:

Після входу користувача в систему віддалена система запитує у контролера *HACS* список пристроїв.

Користувач просто обирає гаражні ворота зі списку приладів.

Дистанційна система запитує список команд гаражних воріт.

Контролер *HACS* запитує двері гаража список команд.

Двері гаража повертають список команд до контролера *HACS*, потім контролер *HACS* повертає список до віддаленої системи, а потім віддалена система відображає список команд

Користувач вибирає одну зі списку команд, і контролер *HACS* виконує операцію.

Хід подій:

Після входу користувача в систему віддалена система запитує у контролера *HACS* список пристроїв.

Користувач просто вибирає телевізор зі списку пристроїв.

Дистанційна система запитує список команд телевізора.

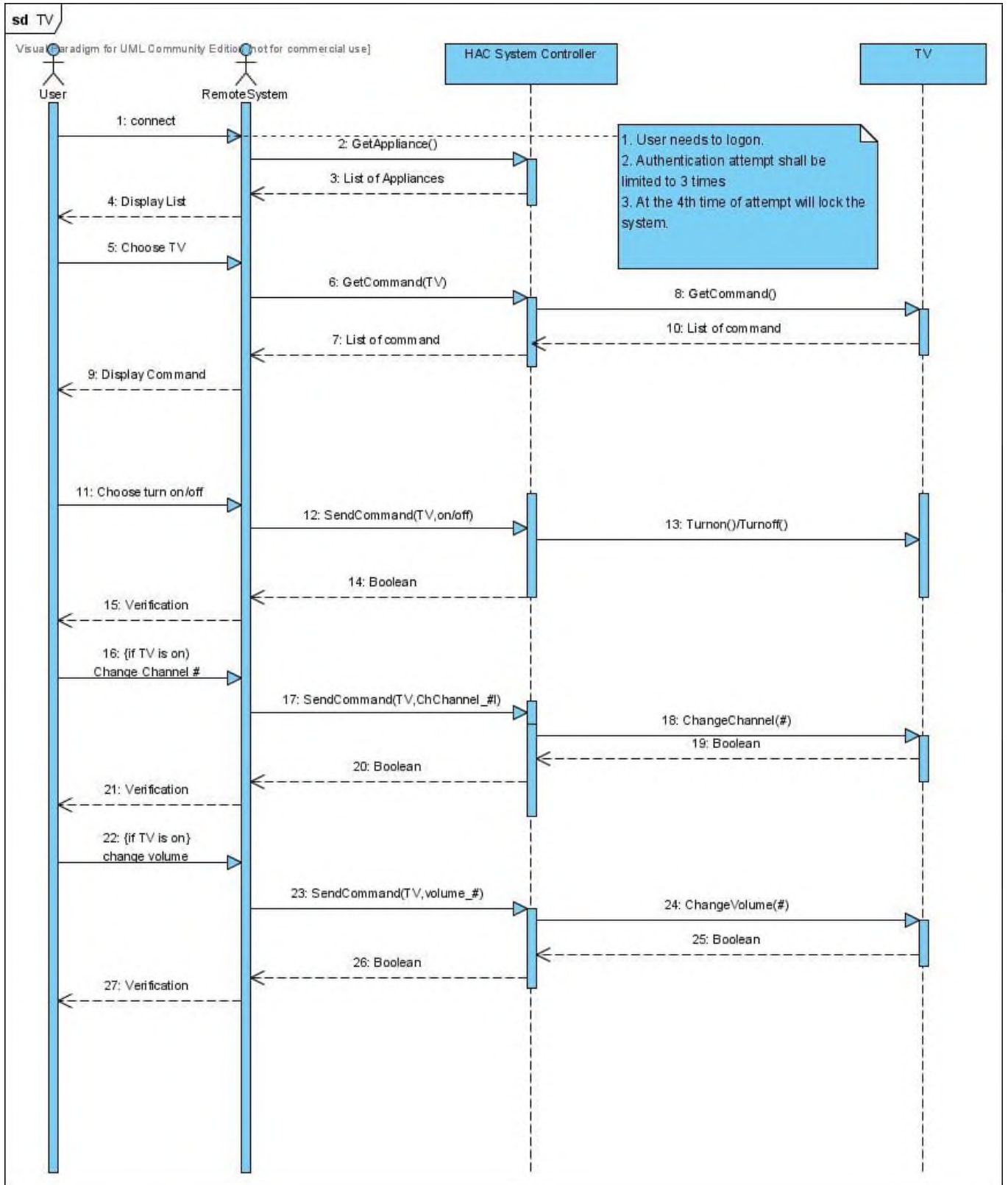


Рис. 2.11. Керування телевізором

Контролер *HACS* запитує у телевізора список команд.

Двері гаража повертають список команд до контролера *HACS*, потім контролер *HACS* повертає список до віддаленої системи, а потім віддалена система відображає список команд

Користувач вибирає одну зі списку команд, і контролер *HACS* виконує операцію.

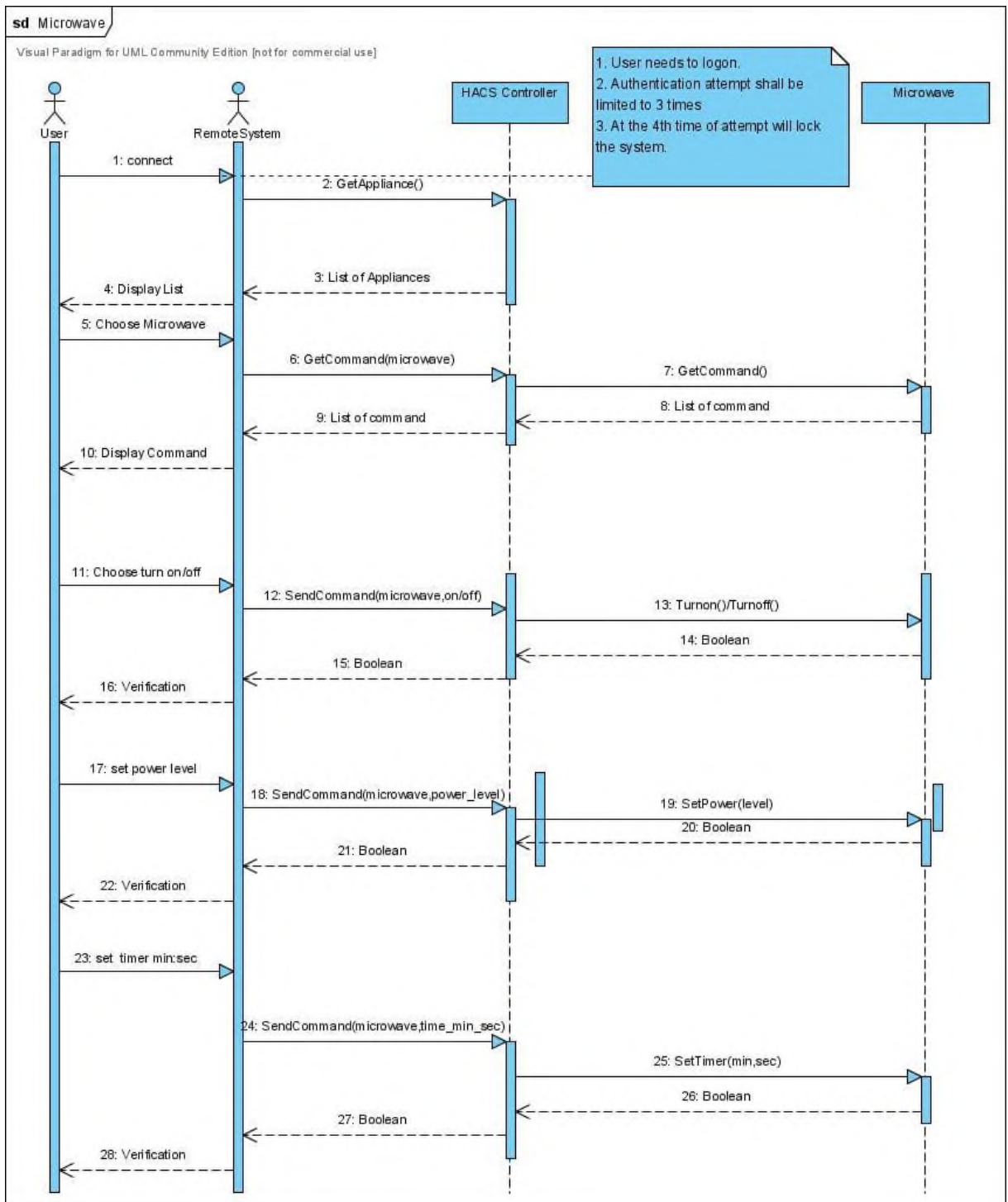


Рис. 2.12. Увімкніть мікрохвильову піч

Хід подій:

– Після входу користувача в систему віддалена система запитує у контролера *HACS* список пристроїв.

– Користувач просто обирає мікрохвильову піч зі списку приладів.

– Дистанційна система запитує список команд мікрохвильової печі.

– Контролер *HACS* запитує мікрохвильову піч список команд.

– Двері гаража повертають список команд до контролера *HACS*, потім контролер *HACS* повертає список до віддаленої системи, а потім віддалена система відображає список команд

– Користувач вибирає одну зі списку команд, і контролер *HACS* виконує операцію.

2.5. 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.

Система «Розумний дім» - це контроль і управління інтегрованою кількістю невеликих систем вдома. Невелика система може бути вимикачем лампи, моніторингом температури, виявлення руху, домашнім спостереженням та іншими датчиками. Датчиками в цих системах будуть керувати користувачі за допомогою пристроїв інтерфейсу, таких як дистанційне керування, комп'ютер і смартфон. Завдяки збільшенню типу датчика, яким потрібно керувати, основна система повинна бути більш спеціалізованою, щоб інтегрувати підсистеми, щоб стати системою «Розумний дім». Мережа системи може бути дротовою або бездротовою залежно від програми.

Несучий режим домашньої автоматизації передає сигнал для передачі електроенергії на побутові прилади. Є чотири типи режиму несучої; системи ліній електропередач, бездротові системи, апаратні системи та системи Інтернет-протоколу.

Зв'язок по лінії електропередач - це передача або передача даних через провідник. На систему проводки подається модульований несучий сигнал. Різні типи зв'язку по лініях електропередачі мають різні діапазони частот залежно від характеристик передачі сигналу використовуваної електропроводки. Оскільки система електропроводки спочатку була призначена для передачі енергії змінного струму (AC), схеми силових проводів були обмежені для передачі на вищих частотах.

Основними стандартами для дротового режиму передачі є INSTEON, KNX і X10. Стандарт INSTEON використовує сітчасту топологію та розроблений на основі моделі X10, що дозволяє об'єднувати прості пристрої в мережу за допомогою лінії електропередач або радіочастот. Пристрої ліній електропередач INSTEON можуть передавати дані на частоті 131,65 кГц з відповідними бездротовими пристроями на частоті 904 МГц. Пристрої INSTEON є одноранговими, тому головний контролер або програмне забезпечення для маршрутизації не потрібні. KNX — це стандарт мережевого протоколу зв'язку на основі OSI для розумних будівель.

РОЗДІЛ 3

ОПИСАННЯ ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

3.1. Алгоритм роботи системи дистанційного керування побутовими приладами

Було реалізовано прототип розглянутого дистанційного контролера побутової техніки. Наша реалізація контролює побутову техніку розумного будинку за допомогою мобільних телефонів з будь-якого місця. Ми також розробили та реалізували різні способи зв'язку мобільного телефону з комп'ютером або мікроконтролером. X10 є однією з широко використовуваних технологій у світі Smart Home. Ми використали X10 Active Home Pro, щоб подбати про реальний механізм керування приладом, коли йому було достатньо вхідних даних. Використовуючи цей набір пристроїв, ми зосередилися на комунікаційному аспекті та придумали різні рішення.

Коротке порівняння різних альтернативних рішень: Більшість комерційно доступних рішень для керування побутовою технікою є веб-основою. Тому ми почали думати про веб-рішення, щоб з'ясувати переваги та недоліки доступних рішень придумати кращі рішення. Веб-рішення вимагає розміщення веб-сервера та підключення до Інтернету як на мобільному телефоні, так і на домашньому комп'ютері. За умови виконання цих вимог домашній комп'ютер відповідає за отримання команд із мобільного телефону та передачу команд у відповідному форматі на контролер X10. Головним недоліком веб-рішення є його постійна залежність від Інтернету. Тому це рішення не можна назвати справді віддаленим. Рішення на основі FBUS [7] викликає додаткову складність кодування, а також незграбне підключення до домашнього мобільного телефону. Налаштування вимагає, щоб домашній мобільний телефон знаходився вдома та підключався до комп'ютера. Управління здійснюється за допомогою віддаленого мобільного пристрою. Недоступність деталей протоколу FBUS і сильна залежність від мобільних телефонів Nokia, стає все важче створювати надійні програми.

Рішення на базі Bluetooth є дуже сильним кандидатом, щоб бути визнаним найкращим серед усіх. Bluetooth забезпечує повний двонаправлений зв'язок між комп'ютером і мобільним телефоном. Крім того, стандартна клієнт-серверна модель програмування полегшує завдання. Управління можна здійснювати як з домашнього, так і з віддаленого мобільного телефону, а також домашній мобільний є певною мірою мобільним у тому сенсі, що Bluetooth забезпечує невеликий діапазон мобільності. Також Bluetooth працює для будь-якого мобільного телефону з Bluetooth API незалежно від постачальника. Єдиним обмежувальним фактором для цього рішення є потреба втручання людини для спеціально створених програм у мобільному телефоні для надсилання й отримання даних із зовнішніх ресурсів. За умови, що цього кроку можна уникнути, здається дуже логічним скористатися цим рішенням. Ми запропонували рішення на основі команд АТ через його простоту та легкість використання. Це не вимагає жодних додаткових витрат і доступне з усіма мобільними телефонами. З'єднання не має обмежень для домашнього мобільного телефону, оскільки його можна під'єднати будь-яким із доступних способів, як-от Bluetooth, інфрачервоний зв'язок або кабель передачі даних. Це також зменшує необхідність будь-якого втручання людини. Програмування стосується лише комп'ютера і нічого не стосується домашнього мобільного телефону, що спрощує завдання.

Опис прототипу: Реалізація прототипу передбачає два мобільні телефони, один комп'ютер і систему X10 Active Home Pro як апаратні компоненти. Програмне забезпечення, що полегшує весь зв'язок, розроблено з використанням мов програмування Java Standard Edition (J2SE) і Micro Edition (J2ME), а також програма C використовується з Java за допомогою JNI (Java Native Interface) [4]. Компоненти описані нижче:

Віддалений мобільний: віддалений мобільний телефон може працювати будь-яким із двох способів. Кращим є для мобільних телефонів із підтримкою Java. Ми розробили додаток J2ME для віддаленого мобільного. Програма дозволяє призначати зрозумілий для людини набір імен замість імен за

замовчуванням для модулів X10. Користувач може вільно використовувати або призначені імена, або спеціальні імена X10, використовуючи простий графічний інтерфейс користувача зі свого мобільного телефону. Для решти мобільних пристроїв, які не підтримують Java, користувача просять надіслати повідомлення Home Control Message для керування побутовою технікою. Віддалений мобільний пристрій повністю не залежить від постачальника, і єдиною необхідною вимогою для нього є можливість надсилання та отримання SMS.

Домашній мобільний: Домашній мобільний телефон підключається до комп'ютера за допомогою стандартного кабелю передачі даних. Комп'ютер зв'язується з мобільним за допомогою команди AT.

Домашній комп'ютер: комп'ютер у нашому прототипі використовує протокол команд AT, щоб визначити, чи отримано нове повідомлення Home Control на домашній мобільний телефон. Прототип програмного забезпечення розроблено на платформі Windows, але його можна легко перенести на інші ОС, такі як Unix/Linux тощо. Після розпізнавання та автентифікації повідомлення управління домом комп'ютер шукає відповідність між зрозумілими людині іменами та спеціальними кодами X10. Після формування ключа машини, що відповідає пристрою X10, повідомлення надсилається на спеціальний порт USB, який підключається до контролера X10.

Пакет домашнього контролера X10 : X10 — це протокол зв'язку, подібний до мережевих протоколів, таких як TCP/IP. Однак X10 працює через домашні лінії електропередач. Подібно до мережі трансляції, кожна команда надсилається через кожен дріт у будинку; кожен окремий пристрій вирішує, чи потрібно йому відповідати на певну команду. За допомогою X10 різноманітні пристрої можуть спілкуватися один з одним, а також з комп'ютером. CM15A, двосторонній інтерфейс ПК, знаходиться на іншому кінці стрибка зв'язку від комп'ютера. На основі надісланого повідомлення Home Control Message ми надсилаємо відповідний код X10 на CM 15A. Також ми надсилаємо адресу цільового пристрою. Потім CM15A транслює це повідомлення через лінію електропередачі. Усі підключені пристрої X10 отримують повідомлення, але відповідає лише цільовий пристрій, указаний у повідомленні. Тут застосоване обмеження:

Повідомлення X10 дійсні, лише якщо між контролером і приймальним модулем немає трансформатора. Модуль лампи LM465 і модуль приладу AM486 використовуються в нашому прототипі для керування освітленням та іншими загальними приладами, як-от вимикачі тощо. Але це не є обмежувальною функцією, оскільки додавання інших модулів є лише питанням підключення до домашньої лінії електроживлення. Програмний модуль контролера завершує нижній рівень зв'язку комп'ютера з CM15A через USB. Програмний модуль представляє інтерфейс, який дозволяє клієнту цього модуля передавати необроблені дані, не турбуючись про деталі реалізації протоколу низького рівня? Після надсилання цього повідомлення підключений CM15A отримує дані та передає їх через лінію електропередачі на визначеній частоті. Модуль лампи (також інші подібні модулі) попередньо налаштований на дію після отримання цих керуючих сигналів. CM15A отримує ці сигнали підтвердження від лінії живлення та пересилає його на ПК через USB. Нарешті, програма-обгортка підтверджує успішне виконання виданої команди. Переваги нашого рішення описані нижче

Наше рішення не потребує Інтернету, який не є безпечним і надійним медіа.

Це перше рішення такого роду, яке використовує мобільні пристрої, якими зараз користуються всі.

Використання мобільних пристроїв забезпечує мобільність рішення, і побутовою технікою можна керувати майже з будь-якої точки світу.

Наше рішення використовує стандартний сторонній контролер для керування побутовою технікою, що робить наше рішення надійним. Наше рішення керує контролером стороннього виробника, а контролер стороннього виробника бере на себе відповідальність за керування побутовою технікою. Ми запропонували низку механізмів зв'язку, за допомогою яких можна підтримувати велику кількість контролерів.

Розроблена система не тісно пов'язана з певним постачальником сторонніх контролерів. Отже, ми можемо мати будь-який сторонній контролер і все одно використовувати мобільність і доступність.

Незважаючи на те, що мова йде про контроль над домом, основне ядро технології можна використовувати для керування будь-якою системою за допомогою мобільного телефону.

Це порівняно недороге, просте в обслуговуванні і найбільш доступне рішення.

Скрізь, де АТ-команди надають комп'ютеру найбільш гнучкий спосіб контролювати та досліджувати служби та ресурси мобільного пристрою. Команди АТ дозволяють надсилати та отримувати SMS із комп'ютера, а також дозволяють комп'ютеру переглядати ресурси мобільного телефону, такі як пам'ять, телефонна книга тощо. «АТ» означає команду «Увага». Мобільні телефони стандарту GSM оснащені вбудованими модемами GSM, які відповідають на команди, що надходять у вигляді SMS від підключеного комп'ютера. АТ-команди створюють логічний двонаправлений зв'язок між комп'ютером і мобільним телефоном.

Формат АТ-команд: команди подібні до будь-яких інших команд, які ми використовуємо в різних терміналах. Команди відповідають чітко визначеному синтаксису. Почнемо з того, що він не чутливий до регістру. Загальний префікс АТ для всіх команд. Кілька команд можна вказати в одному рядку, починаючи лише з одного 'АТ' як загального префікса. Символи < > і [] позначають значення обов'язкових і додаткових параметрів відповідно. Коли комп'ютер видає команду, ефект зберігає своє значення, доки його не буде явно змінено. Перевірка з'єднання виконується шляхом надсилання команди «АТ», і якщо мобільний телефон під'єднано правильно, комп'ютер сповіщає про це через повідомлення «ОК». В іншому випадку надсилається повідомлення про помилку.

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.

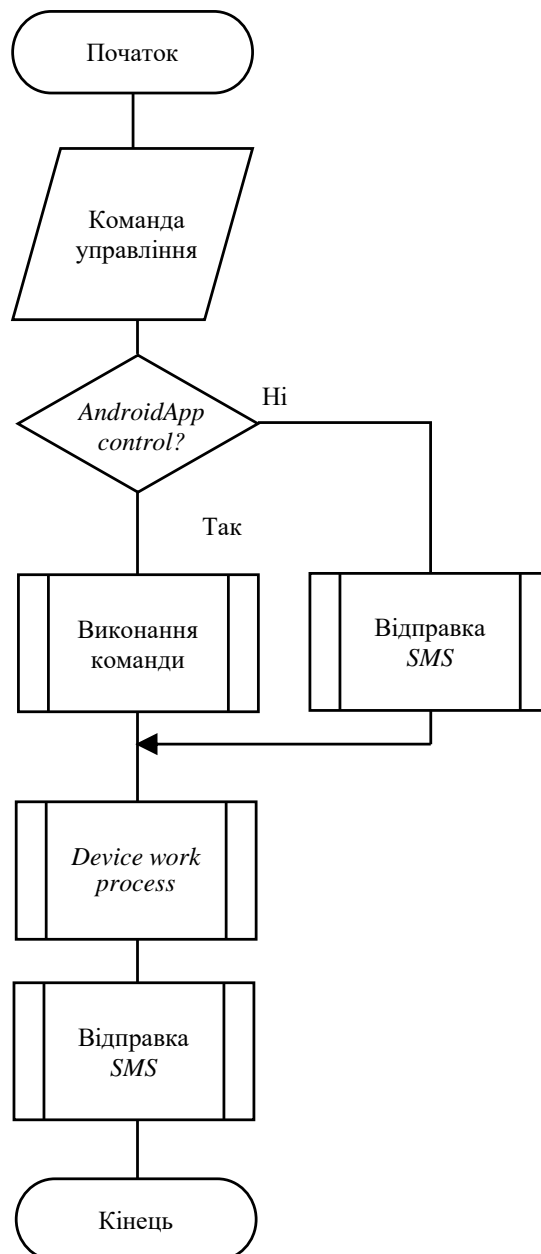


Рис. 3.1. Схема алгоритму роботи програмного модуля

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*.

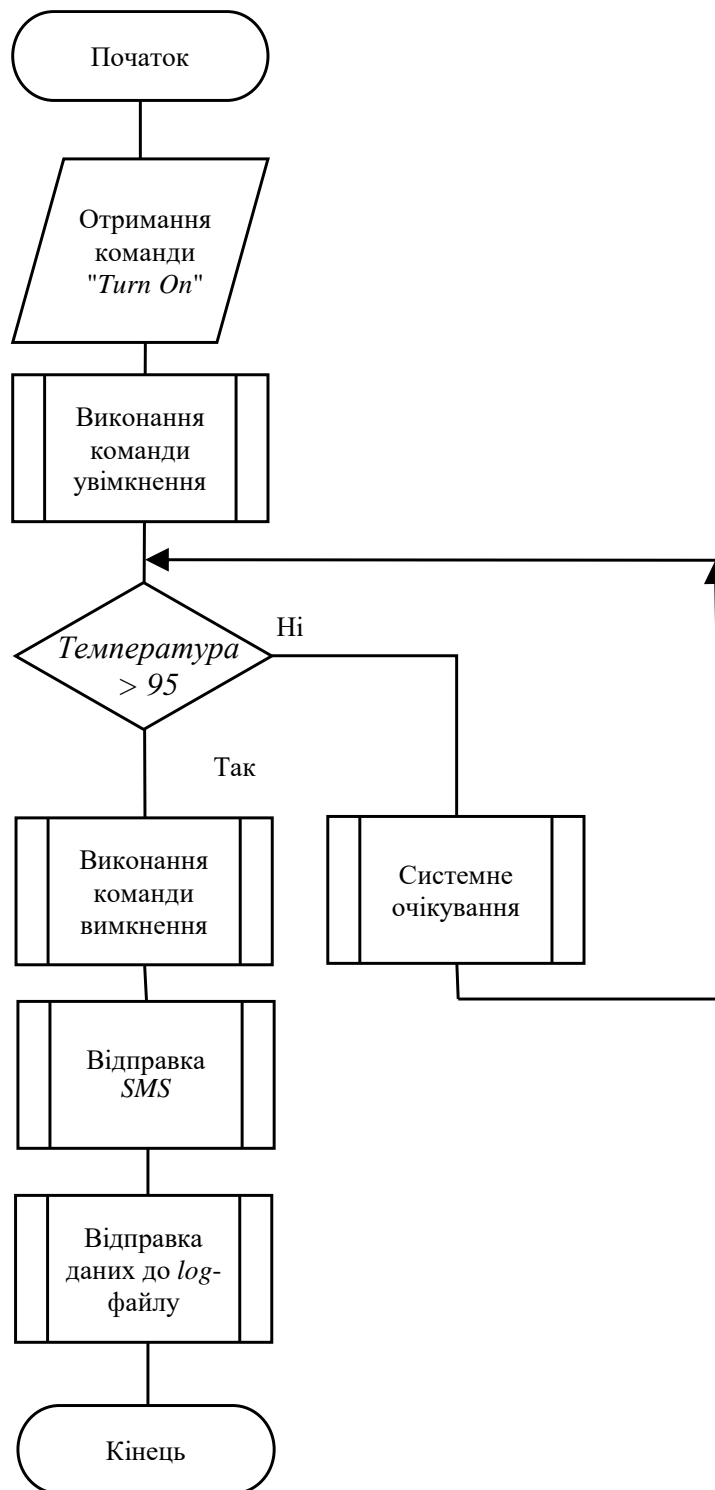


Рис. 3.2. Схема алгоритму процедури *Device work process*

3.2. Розробка апаратного модуля

Дизайн системи дає уявлення про архітектуру системи, яка є недорогою, ефективною, з меншим енергоспоживанням і можливостями віддаленого моніторингу, а також логіку роботи, що стоїть за різними функціями розумного

дому. У нашому проекті представлено недорогу допоміжну систему розумного будинку для дистанційного керування та моніторингу середовища розумного будинку. Архітектуру можна налаштувати різними способами, щоб відповідати різним сценаріям застосування в розумному будинку з мінімальним перекодуванням і дизайном, і вона розділена на три рівні: домашнє середовище, домашній шлюз і віддалене середовище (рис. 1). Функції, які пропонує дизайн, включають:

Контроль систем управління енергією, таких як освітлення, розетки та системи опалення, вентиляції та кондиціонування повітря;

Безпека, як виявлення пожежі та виявлення вторгнення з сигналізацією та автоматичними сповіщеннями; і

Дистанційне керування середовищем розумного будинку, наприклад, підтримання певної температури в приміщенні та вмикання та вимикання пристроїв з віддаленого місця.

Домашнє середовище – це частина системи, з якою люди з обмеженими можливостями та похилого віку перебувають у прямому контакті, і складається з датчиків, які сприймають різні сигнали та виконують певну активну реакцію, щоб полегшити життя літніх та людей з обмеженими можливостями. Розгорнуті датчики включають:

Фотосенсори, які здійснюють автоматизацію освітлення;

Детектор газу, який вимикає газ при виявленні витоку газу;

Датчик руху, який визначить присутність зловмисника і попередить мешканців будинку; і

Датчик температури, який ініціює відключення і включення системи охолодження в приміщенні.

У домашніх умовах розгорнуті датчики вибирають сигнали, надсилають їх на плату Arduino, яка виконує інтерпретацію, і надсилає команди для виконання певних дій. Усі датчики в цьому середовищі взаємодіють безпосередньо з платою Arduino. Датчики та виконавчі механізми підключені безпосередньо до головного контролера Arduino.

Домашній шлюз для автоматизації розділений на дві частини; а саме серверне прикладне програмне забезпечення, яке є бібліотечною реалізацією мікро-веб-сервера, що працює на Arduino Uno з використанням екрану Ethernet, змонтованого до мікропрограми Arduino та мікроконтролера. Для успішного обміну даними між віддаленим користувачем і домашнім шлюзом на Arduino реалізовано рівні конфігурації, сенсора та рівня керування приводом. Домашнє середовище можна контролювати та контролювати з віддаленого місця за допомогою програми розумного дому або веб-браузера, який спілкується з мікровеб-сервером через Інтернет. Кінцеві користувачі з правильними обліковими даними можуть отримати доступ.

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

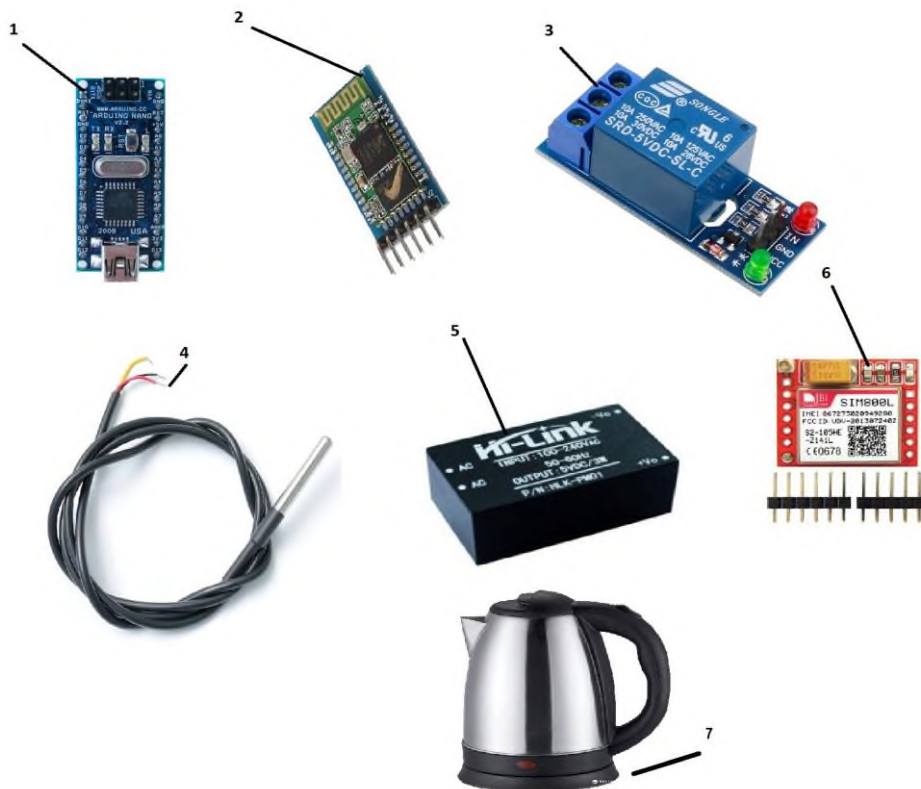


Рис 3.3. Необхідні деталі

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).

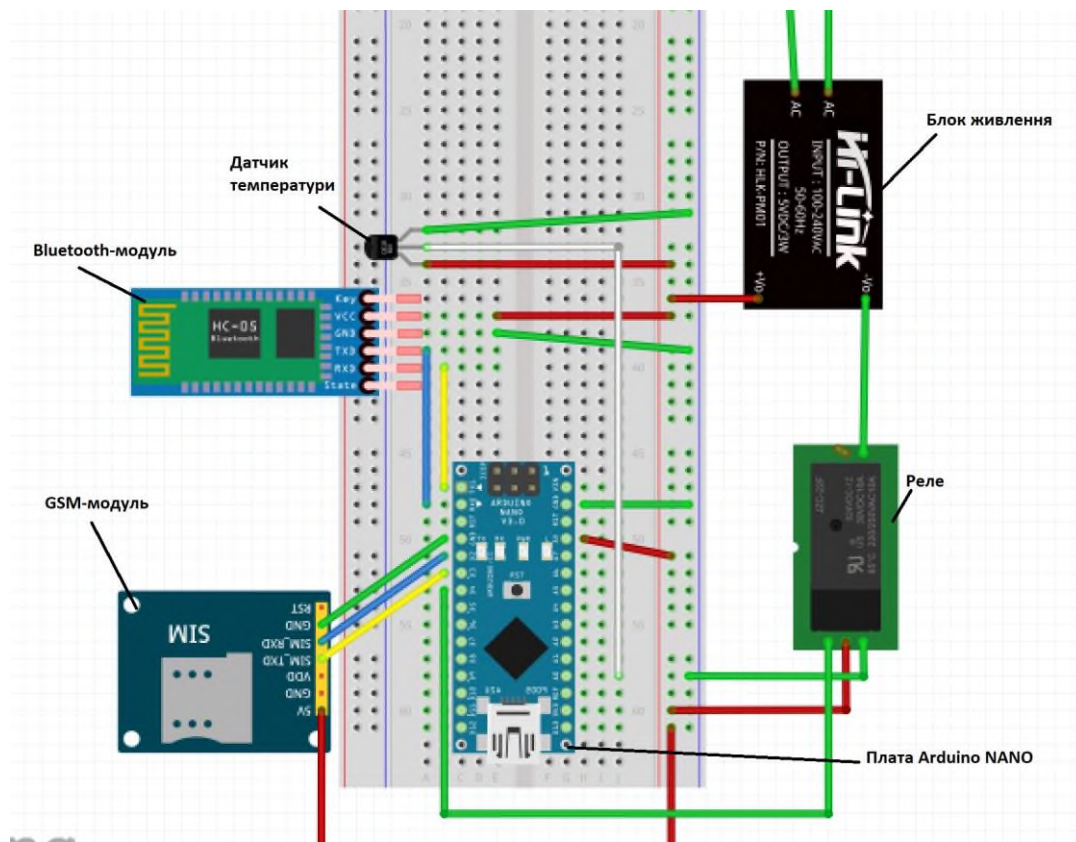


Рис. 3.4. Схема пристрою на макетній платі

Розроблено принципову схему (рис. 3.5).

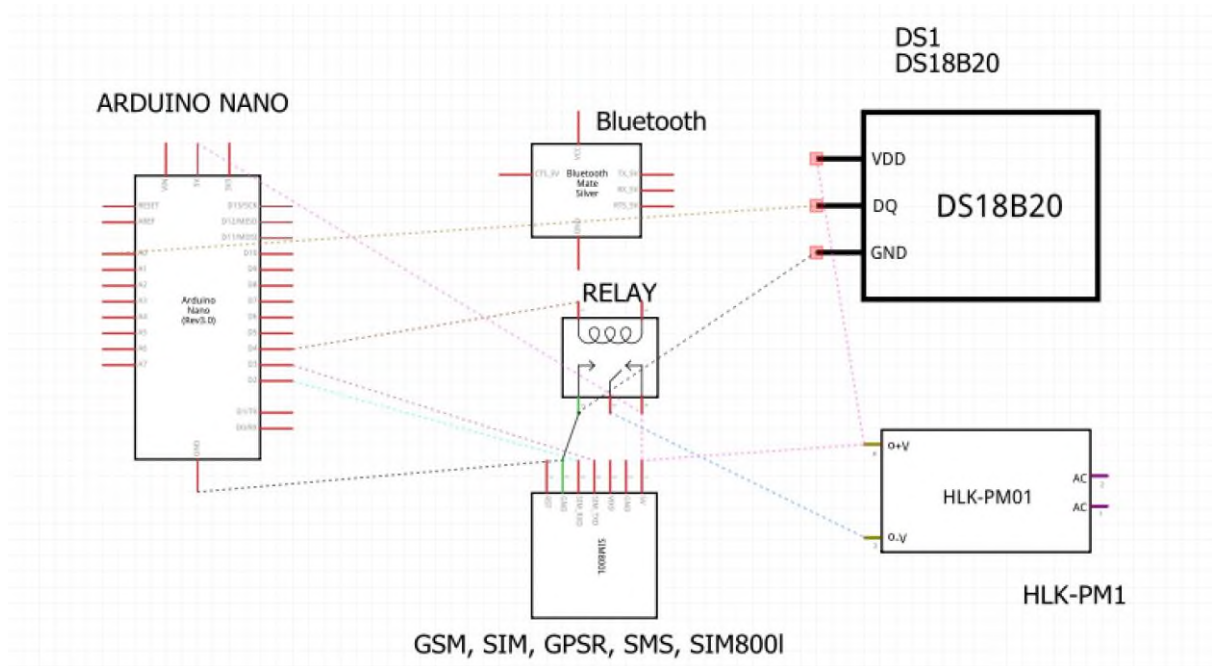


Рис 3.5. Принципова схема апаратної частини

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

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

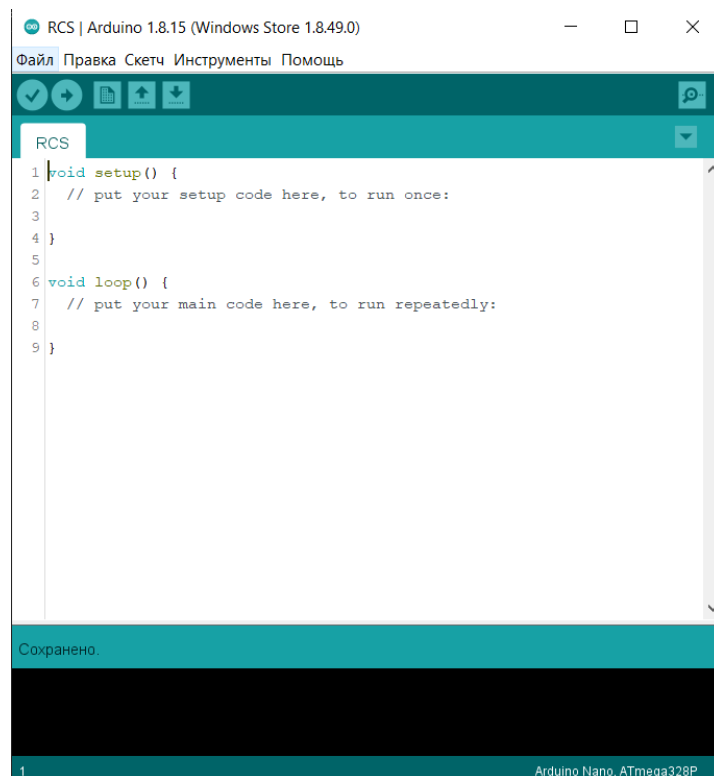


Рис 3.6. Новий проект

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).

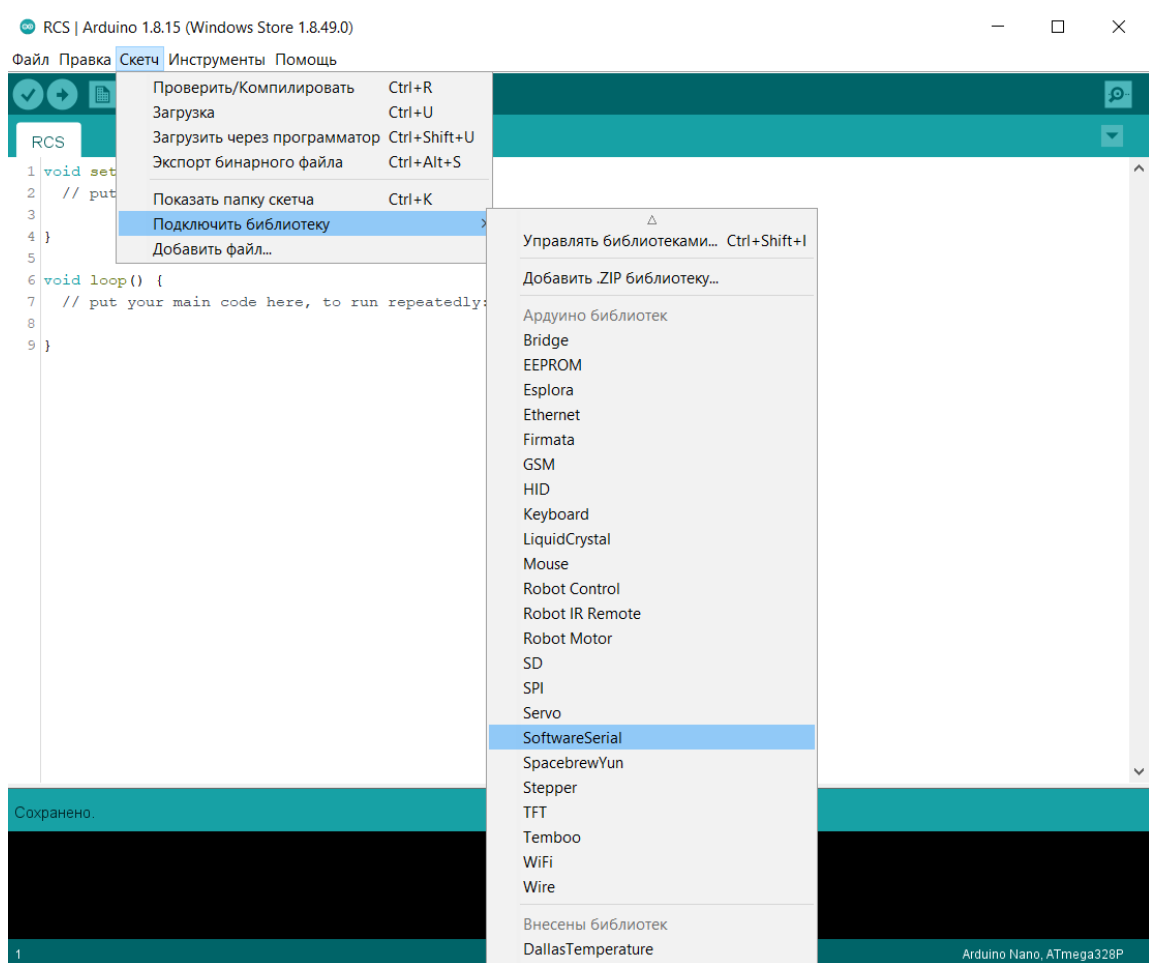


Рис. 3.7. Підключення бібліотеки *SoftwareSerial.h*

Бездротова технологія Bluetooth [8] стоїть на шляху традиційної технології дротового зв'язку малого радіусу дії, що підключає портативні та/або стаціонарні пристрої, зберігаючи високий рівень безпеки. Це застаріло дроти між вашою робочою станцією, мишею, портативним комп'ютером, музичними навушниками тощо. Ключовими особливостями технології Bluetooth є надійність, низька потужність і низька вартість. Специфікація Bluetooth встановлює стандарт для широкого діапазону пристроїв для підключення та обміну даними один з одним. Фундаментальною перевагою бездротової технології Bluetooth є здатність одночасно обробляти передачу даних і голосу, що робить її найбільш можливою для портативних пристроїв.

Розуміння зв'язку Bluetooth: пристрої Bluetooth повинні мати можливість виявляти пристрої Bluetooth поблизу. Коли виявляється новий пристрій Bluetooth, може бути ініційовано пошук служби, щоб визначити, які послуги пропонує пристрій. У специфікації Bluetooth операція виявлення пристрою називається запитом. Під час процесу запиту пристрій Bluetooth, який запитує, отримує адресу та годинник Bluetooth від пристроїв, які можна знайти поблизу. Потім пристрій, який запитує, ідентифікує інші пристрої за їхньою адресою Bluetooth, а також може синхронізувати стрибок частоти з виявленими пристроями, використовуючи їхню адресу Bluetooth і годинник. Пристрої стають видимими, увійшовши в режим запиту сканування. Виявлені пристрої використовують код доступу запиту (IAC) для виявлення пристрою. Загальний код доступу до запиту (GIAC) і обмежений код доступу до запиту (LIAC) є типами коду доступу до запиту. GIAC використовується, коли пристрій є загальнодоступним, що означає, що його можна буде виявити протягом невизначеного періоду часу. LIAC використовується, коли пристрій можна буде виявити лише протягом обмеженого періоду часу. Різні пристрої Bluetooth пропонують різні набори послуг. Отже, пристрій Bluetooth має виконати виявлення служби на віддаленому пристрої, щоб отримати інформацію про доступні послуги. Пошук послуг може мати загальний характер шляхом опитування пристрою щодо всіх доступних послуг, але також може бути звужений, щоб знайти лише одну послугу. Процес виявлення служби використовує протокол виявлення служби (SDP). Клієнт SDP повинен надсилати SDP-запити до SDP-сервера, щоб отримати інформацію із службових записів сервера. Пристрої Bluetooth зберігають інформацію про свої послуги Bluetooth у базі даних виявлення служб (SDDB). SDDB містить записи про службу, де кожна запис про послугу містить атрибути, що описують певну послугу. Кожна служба має власний запис у SDDB. Віддалені пристрої можуть отримувати записи послуг під час виявлення служби та володіють усією інформацією, необхідною для використання описаних послуг. Універсальний унікальний ідентифікатор (UUID) використовується для ідентифікації служб, протоколів і профілів. UUID — це 128-бітний ідентифікатор, який гарантовано буде унікальним у будь-якому

часі та просторі. Діапазон значень UUID був попередньо призначений для часто використовуваних служб, протоколів і профілів і вказано в документі Bluetooth Assigned Numbers на веб-сайті Bluetooth Membership. Загальний профіль доступу (GAP) Основа для всіх профілів у системі Bluetooth. GAP визначає основні функції Bluetooth, як-от налаштування L2CAP посилання, обробка режимів безпеки та режимів видимості та простих передач даних. Тому це потрапляє в сферу наших інтересів. Автентифікація виконується за допомогою з'єднання та створення пари

З'єднання — це процедура автентифікації Bluetooth-пристроєм іншого Bluetooth-пристрою та залежить від спільного ключа автентифікації. Якщо пристрої не мають спільного ключа автентифікації, перед завершенням процесу з'єднання необхідно створити новий ключ. Генерація ключа автентифікації називається створенням пари. Процес сполучення передбачає створення ключа ініціалізації та ключа автентифікації з подальшою взаємною автентифікацією. Ключ ініціалізації базується на введенні користувачем, випадковому числі та адресі Bluetooth одного з пристроїв. Введені користувачем дані називають персональним ідентифікаційним номером (PIN) або ключем доступу та можуть мати довжину до 128 біт. Ключ доступу є спільною таємницею між двома пристроями. Ключ автентифікації базується на випадкових числах і адресах Bluetooth з обох пристроїв. Ключ ініціалізації використовується для шифрування під час обміну даними для створення ключа автентифікації, а потім відкидається. Після завершення процесу сполучення пристрої автентифікують один одного. Обидва пристрої використовують один і той же ключ автентифікації, який часто називають комбінованим ключем, оскільки обидва пристрої брали участь у створенні ключа. Коли два пристрої завершать процес сполучення, вони можуть зберегти ключ автентифікації для подальшого використання. Потім пристрої об'єднуються в пару та можуть автентифікувати один одного в процесі з'єднання без використання ключа доступу. Пристрої залишатимуться сполученими, доки один із них не попросить новий процес з'єднання або ключ автентифікації не буде видалено на будь-якому з пристроїв. Авторизація — це процес надання дозволу віддаленому пристрою Bluetooth на доступ до певної служби. Щоб

отримати авторизацію, віддалений пристрій має спочатку пройти автентифікацію за допомогою процесу зв'язування. Тоді доступ може бути надано на тимчасовій або постійній основі. Атрибут довіри пов'язаний з авторизацією, пов'язуючи дозволи авторизації з певним пристроєм. Довірений пристрій може підключитися до служби Bluetooth, і процес авторизації завершиться успішно без участі користувача.

Мобільний зв'язок: Нам потрібно створити мобільну програму Bluetooth-клієнта та комп'ютерну програму-сервер Bluetooth, щоб встановити двонаправлений зв'язок між ними. Ми обираємо JAVA для розробки як клієнта, так і сервера, оскільки J2ME є найбільш використовуваною мовною платформою для розробки мобільних додатків. Java надає Bluetooth Application Programmer Interface (API), який відомий як JSR-82. Він також відомий як JABWT (Java API для бездротової технології Bluetooth). Основні концепції будь-якої програми Bluetooth (Java чи іншої) складаються з ініціалізації стека, виявлення пристрою, керування пристроєм, виявлення служби та зв'язку.

Ініціалізація стека: На самому початку потрібно ініціалізувати стек Bluetooth. Стек — це частина програмного забезпечення (або мікропрограми), яка керує пристроєм Bluetooth. Ініціалізація стека може складатися з кількох речей, але її основна мета — підготувати пристрій Bluetooth до початку бездротового зв'язку. Кожен постачальник по-різному обробляє ініціалізацію стека. Основними функціями для ініціалізації стека є `getLocalDevice ()`, `setDiscoverable ()`, `getDiscoveryAgent ()`.

Управління пристроєм: `LocalDevice` і `RemoteDevice` є двома основними класами в специфікації Java Bluetooth, які дозволяють виконувати керування пристроями. Ці класи дають можливість запитувати статистичну інформацію про локальний пристрій Bluetooth (`LocalDevice`) та інформацію про пристрої у віддаленій області (`RemoteDevice`). Статичний метод `LocalDevice.getLocalDevice ()` повертає створений об'єкт `LocalDevice` для використання. Щоб отримати унікальну адресу радіо Bluetooth, потрібно викликати `getBluetoothAddress ()` на об'єкті локального пристрою. Адреса Bluetooth служить для тих же цілей, що й MAC-адреса на мережева карта комп'ютера. Кожен пристрій Bluetooth має

унікальну адресу. Якщо пристрій хоче, щоб інші пристрої Bluetooth у цій зоні знайшли його, йому потрібно викликати метод `setDiscoverable ()` в об'єкті `LocalDevice`. У двох словах, це майже все, що потрібно для керування пристроями за допомогою `Java Bluetooth Specification API`.

Виявлення пристрою: пристрій Bluetooth не знає, які інші пристрої Bluetooth знаходяться в цьому місці. Можливо, поблизу є ноутбуки, настільні ПК, принтери, мобільні телефони чи КПК. Щоб дізнатися, пристрій Bluetooth використовуватиме класи `Device Discovery`, які надаються в `Java Bluetooth API`, щоб побачити, що там є. Два класи, відповідальні за пристрій Bluetooth для виявлення віддалених пристроїв Bluetooth у зоні, це `DiscoveryAgent` і `DiscoveryListener`. Після отримання об'єкта `LocalDevice` за допомогою функції `getLocalDevice ()` потрібно лише створити екземпляр `DiscoveryAgent`, викликавши `LocalDevice.getDiscoveryAgent ()`. По-перше, об'єкт повинен реалізувати інтерфейс `DiscoveryListener`. Цей інтерфейс працює як будь-який слухач, тому сповіщатиме про подію. У цьому випадку він повідомить, коли пристрої Bluetooth знаходяться в зоні. Щоб розпочати процес виявлення, потрібно викликати метод `startInquiry ()` на `DiscoveryAgent`. Цей метод не блокує, тож можна вільно робити інші речі, поки ви чекаєте, поки будуть знайдені інші пристрої Bluetooth. Коли пристрій Bluetooth знайдено, JVM викличе метод `deviceDiscovered ()` класу, який реалізував інтерфейс `DiscoveryListener`. Цей метод передасть об'єкт `RemoteDevice`, який представляє пристрій, виявлений у запиті.

Відкриття служби: Після пошуку інших пристроїв Bluetooth потрібно буде побачити, які послуги пропонують ці пристрої. Ніколи не можна бути впевненим, які послуги може запропонувати `RemoteDevice`. Сервіс `Discovery` дозволяє дізнатися, що це таке. `Service Discovery` схоже на `Device Discovery` в тому сенсі, що для «виявлення» використовується `DiscoveryAgent`. Метод `searchServices ()` класу `DiscoveryAgent` дозволяє шукати служби на `RemoteDevice`. Коли служби знайдено, JVM буде викликати `servicesDiscovered ()`, якщо об'єкт реалізував інтерфейс `DiscoveryListener`. Цей метод зворотного виклику також передає об'єкт `ServiceRecordObject`, який відноситься до служби, для якої шукали. Маючи під

рукою `ServiceRecord`, можна робити багато речей, але найімовірніше підключитися до `RemoteDevice`, звідки цей `ServiceRecord` походить.

Реєстрація послуги: Перш ніж клієнтський пристрій Bluetooth зможе використовувати функцію виявлення служб на пристрої-сервері Bluetooth, сервер Bluetooth має зареєструвати свої служби всередині бази даних виявлення служб (SDDB). Цей процес називається реєстрацією послуги. У одноранговій програмі, такій як програма для передачі файлів або програма для чату, слід пам'ятати, що будь-який пристрій може діяти як клієнт або сервер, тому потрібно включити цю функціональність (як клієнта, так і сервера) у код у для обробки обох сценаріїв виявлення служби (тобто клієнта) і реєстрації служби (тобто сервера). Ось сценарій того, що необхідно для того, щоб реєструвати та зберігати службу в SDDB. Спочатку викличте `Connector.open ()` і передайте результуюче підключення до `StreamConnectionNotifier`. `Connector.open ()` створює новий `ServiceRecord` і встановлює деякі атрибути. По-друге, використовуйте об'єкт `LocalDevice` і `StreamConnectionNotifier`, щоб отримати `ServiceRecord`, створений системою. По-третє, додайте або змініть атрибути в `ServiceRecord` (необов'язково). По-четверте, використовуйте `StreamConnectionNotifier` і викличте `acceptAndOpen ()` і зачекайте, поки клієнти Bluetooth виявлять цю службу та підключаться. Система створює службовий запис у SDDB. По-п'яте, дочекайтеся підключення клієнта. По-шосте, тоді сервер готовий до виходу, виклик `close ()` у `StreamConnectionNotifier`. Система видаляє службовий запис із SDDB. Використовуйте `StreamConnectionNotifier` і викличте `acceptAndOpen ()` і зачекайте, поки клієнти Bluetooth виявлять цю службу та підключаться. Система створює службовий запис у SDDB. По-п'яте, дочекайтеся підключення клієнта. По-шосте, тоді сервер готовий до виходу, виклик `close ()` у `StreamConnectionNotifier`. Система видаляє службовий запис із SDDB.

StreamConnectionNotifier і Connector походять із пакета javax.microedition.io платформи J2ME. Це все, що потрібно для реєстрації послуги в Bluetooth.

спілкування: Bluetooth — це протокол зв'язку. Java Bluetooth API надає три способи надсилання та отримання даних. У цьому документі ми розглядаємо один із них під назвою RFCOMM. RFCOMM — це рівень протоколу, який профіль послідовного порту використовує для зв'язку, але ці два елементи майже завжди використовуються як синоніми. Для сторони сервера,

спілкування потребує унікального UIUD. URL-адреса: <btsp://localhost:>+<UIUD>+<name=appName>, де appName — це назва програми, яку вибрав користувач. Потрібні StreamConnectionNotifier і StreamConnection. Нам потрібно перевести Connector.open (URL) до екземпляра StreamConnectionNotifier. Метод acceptAndOpen () StreamConnectionNotifier повертає екземпляр StreamConnection, який необхідний для зв'язку. StreamConnection містить функції openDataInputStream () і openDataOutputStream (), які використовуються для відкриття потоків. Функції readUTF () і writeUTF () StreamConnection використовуються для читання та запису. На стороні клієнта URL-адреса отримується для підключення до пристрою з об'єкта ServiceRecord, який отримується з Service Discovery. Метод getConnectionURL () запису служби повертає URL-адресу. Метод open (URL) Connector повертає екземпляр StreamConnection, який необхідний для зв'язку. StreamConnection містить функції openDataInputStream () і openDataOutputStream (), які використовуються для відкриття потоків. Функції readUTF () і writeUTF () StreamConnection використовуються для читання та запису. Для отримання додаткової інформації про Bluetooth див. [12] [13].

Плюси

1. Bluetooth пропонує глобальний стандарт для підключення широкого спектру пристроїв із різними службами. Отже, використання протоколу Bluetooth у нижній частині робить одну програму дещо універсальною з точки зору сумісності.

2. Bluetooth доступний на більшості кишенькових пристроїв, таких як стільникові телефони, музичні плеєри та камери, які відповідають визначеному

стандарту. Отже, щоб використовувати Bluetooth з комп'ютером, не потрібно думати про накладні витрати, такі як Інтернет і веб-сервери.

3. Технологія дуже проста у використанні. Пристрої, як правило, мають вбудовану підтримку програмного забезпечення для роботи Bluetooth, і це програми, які найчастіше використовуються для портативних пристроїв.

4. Найбільш бажаною властивістю для протоколу зв'язку є здатність вимірювати підтримку безпеки, яку він забезпечує. А Bluetooth обмежує зловмисні атаки за допомогою 128-бітних спільних ключів і після безпечного запуску підтримує його, доки не буде зазначено інше.

Мінуси

1. Одним із головних обмежень технології Bluetooth є короткий діапазон операцій. Тому система не справляється з мобільністю.

2. Bluetooth не повністю розроблений та інтегрований у всі задіяні продукти. Нам ще потрібно почекати, поки він повністю розвинеться.

3. У програмі J2ME передача даних будь-яким спеціальним програмним забезпеченням вимагає втручання людини, що робить рішення Bluetooth неможливим для нашого випадку.

3.3. Розробка програмного модуля

У системних сценаріях ми розробили систему автоматичних дверей для допомоги людям з обмеженими можливостями та людям похилого віку, яка помічає та розпізнає їх, а також автоматично відкриває та закриває двері. Два датчики розміщуються всередині та зовні будинку. Система виявлення витoku газу піклується про хворих на хворобу Альцгеймера та аносмію, яка попереджає про витік газу. Датчик детектора газу відповідає за активацію периферійної схеми датчика газу з підключеним до нього сервомотором. Якщо є витік газу, схема датчика газу виявляє це, перериває ручку та передає попереджувальне повідомлення про витік. Актори через кухонну сигналізацію отримують повідомлення. Контур сервомотора обертається, щоб вимкнути ручку газу.

Дверний дзвінок був розроблений спеціально для допомоги глухим. Невеликий вібраційний двигун підключений до радіощита, який зв'язується з Arduino, тоді як схема дверного дзвінка також підключається до радіоприймача на Arduino. Ця система дозволяє фізично повідомити про цих людей, запускаючи вібраційний двигун на людину, коли натискається дверний дзвінок. Система виявлення вторгнень інтегрована в розумний дім, щоб покращити безпеку та допомогти незрячим розпізнавати присутність інших людей у домі.

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).

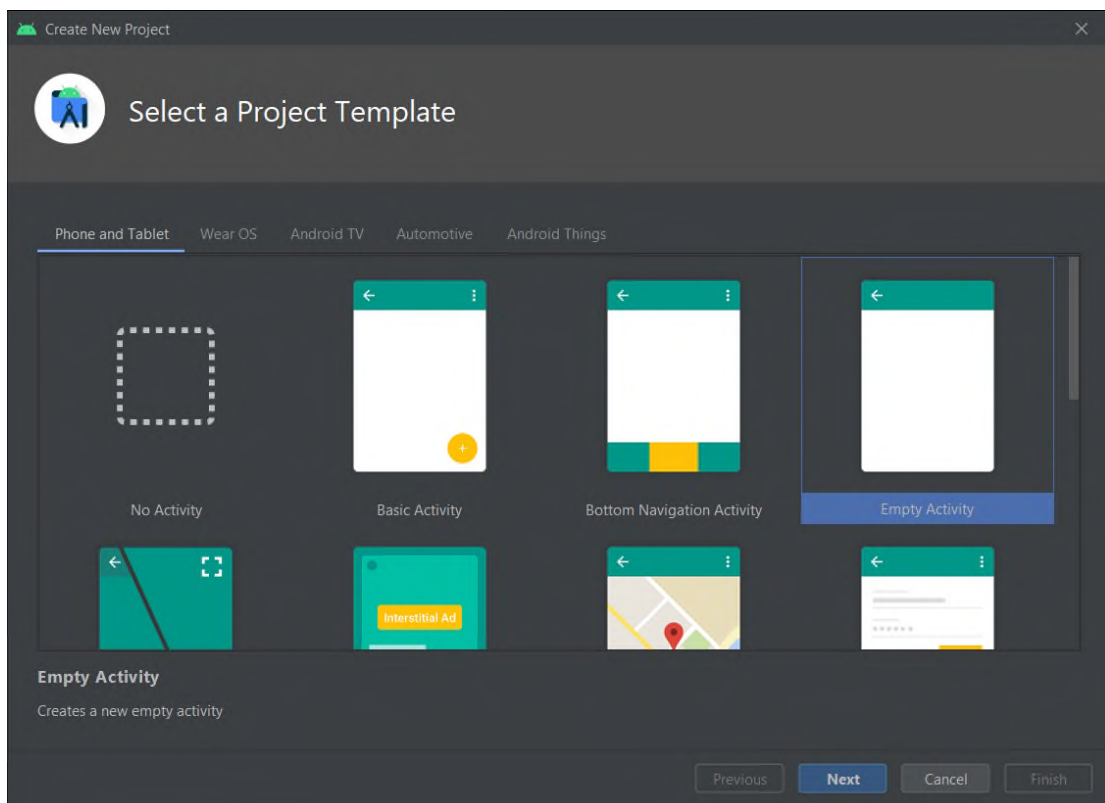


Рис. 3.5. Вибір шаблону *Empty Activity*

Датчик руху розгортається навколо будинку та активується, щоб виявити присутність інших людей у домі. Спрацьовує система тривоги, яка, у свою чергу, запускає систему сповіщень для органів безпеки для виконання необхідних дій у разі виявлення. Система віддалено контролюється через веб-браузер. У період, коли люди з обмеженими фізичними можливостями або люди похилого віку можуть бути не в змозі пересуватися, функції віддаленого моніторингу та

керування забезпечують функціональність увімкнення та вимкнення пристроїв у домі. Включає систему освітлення, кондиціонування та інші пристрої в оселі. Світлозалежний резистор (LDR) використовується для регулювання як зовнішнього, так і внутрішнього освітлення на основі інтенсивності навколишнього освітлення. Це економить енергію та полегшує життя інвалідам.

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).

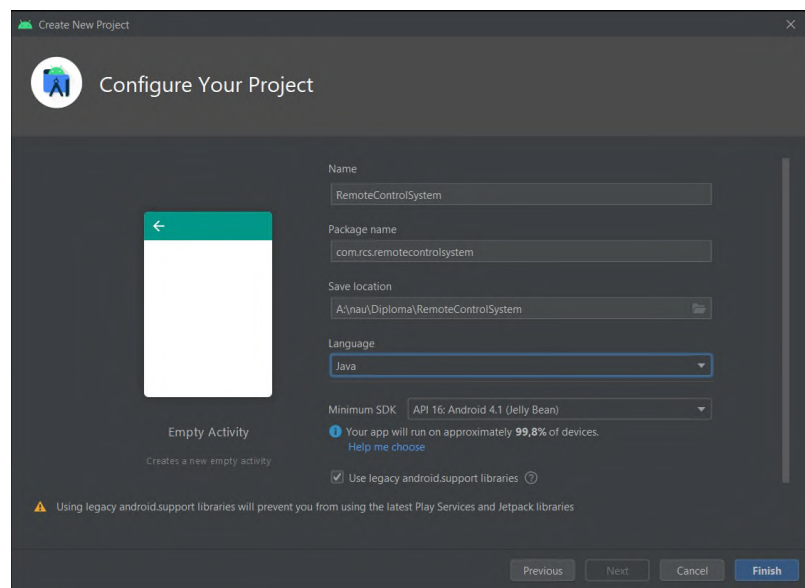


Рис. 3.6. Створення класу у проекті *Android Studio*

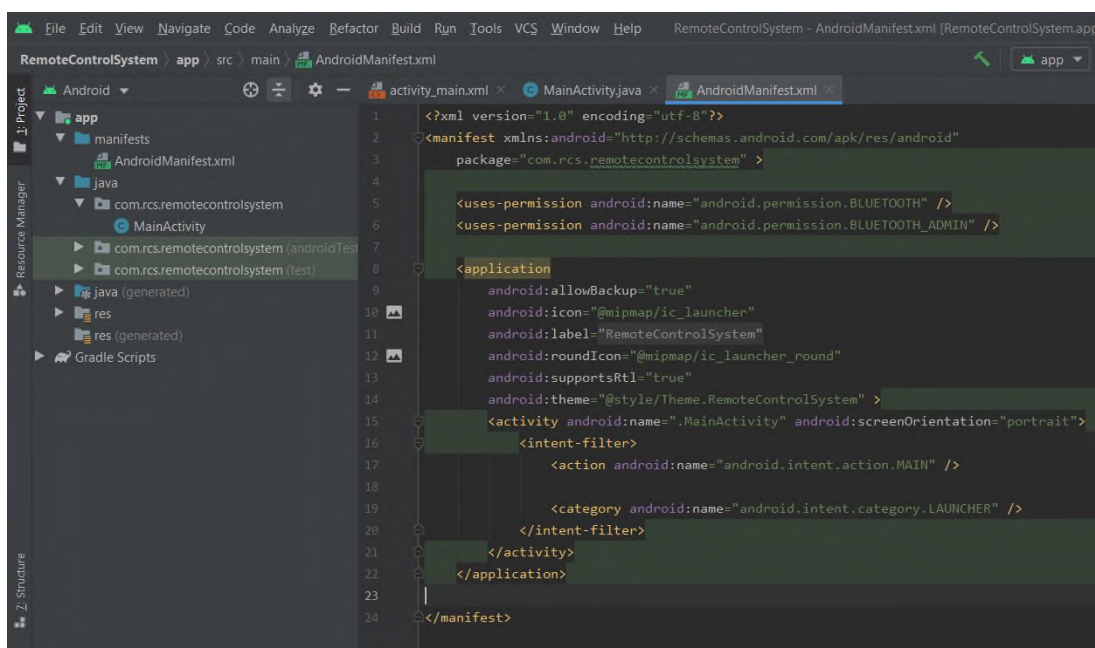


Рис. 3.7. Файл *AndroidManifest.xml*

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

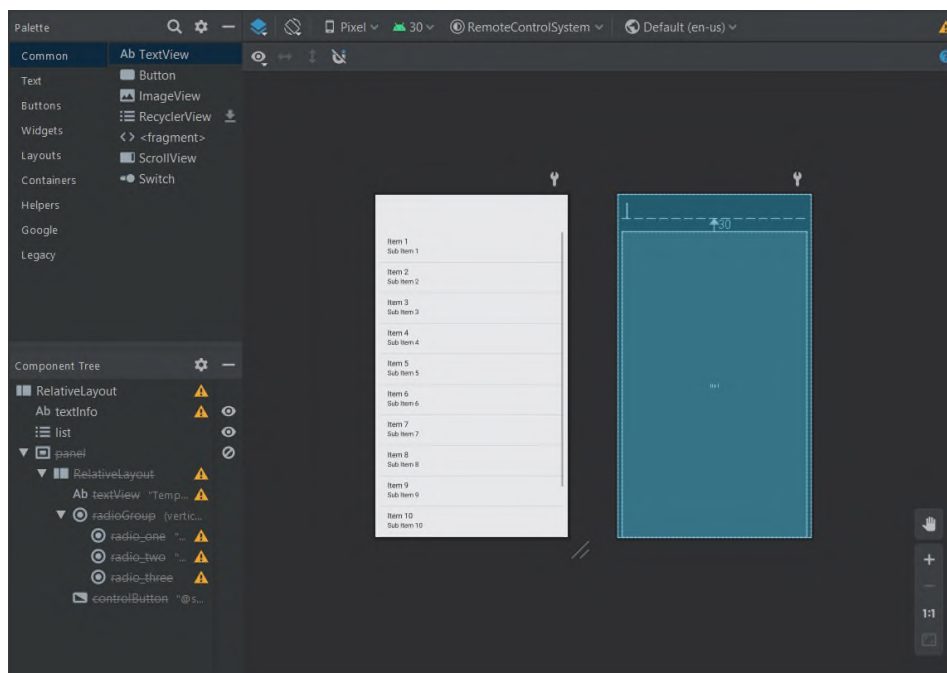


Рис. 3.8. Файл *activity_main.xml*

Основними стандартами для дротового режиму передачі є INSTEON, KNX і X10. Стандарт INSTEON використовує сітчасту топологію та розроблений на основі моделі X10, що дозволяє об'єднувати прості пристрої в мережу за допомогою лінії електропередач або радіочастот. Пристрої ліній електропередач INSTEON можуть передавати дані на частоті 131,65 кГц з відповідними бездротовими пристроями на частоті 904 МГц. Пристрої INSTEON є одноранговими, тому головний контролер або програмне забезпечення для маршрутизації не потрібні. KNX — це стандарт мережевого протоколу зв'язку на основі OSI для розумних будівель.

Цей стандарт базується на комунікаційному стеку EIB із розширенням фізичних рівнів і режимів конфігурації. KNX підтримує різні типи комунікаційних засобів, як-от кручена пара, радіочастота, лінія електропередач та IP. Нарешті, X10 є одним із стандартів домашньої автоматизації. Він використовує проводку лінії електропередач як несучий режим. X10 залишається популярним у домашньому середовищі через недорогу доступність нових компонентів. X10 передає пакети з чотирибітним домашнім кодом, за яким

слідуює один або більше чотирибітних одиничних кодів, а потім чотирибітна команда.

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

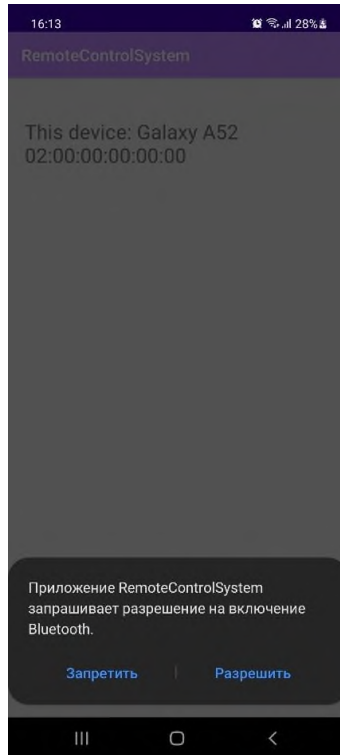


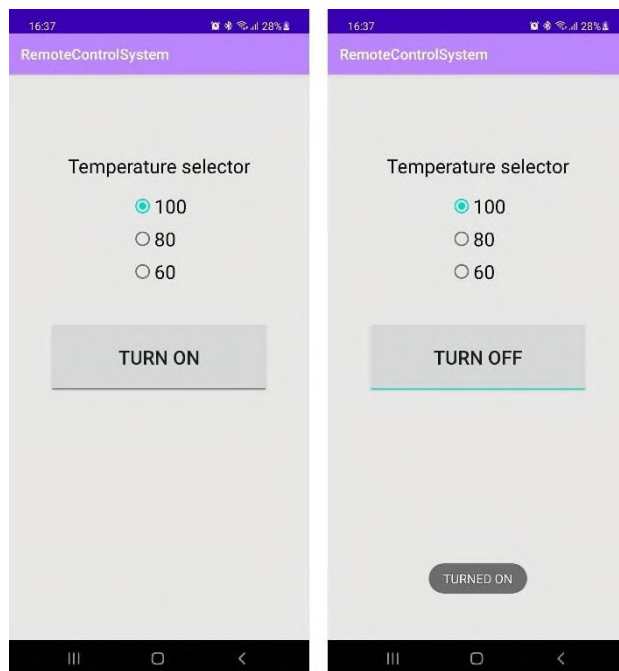
Рис. 3.10. Запит на ввімкнення *Bluetooth*

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 увімкнено, б) при стані вимкнено

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).

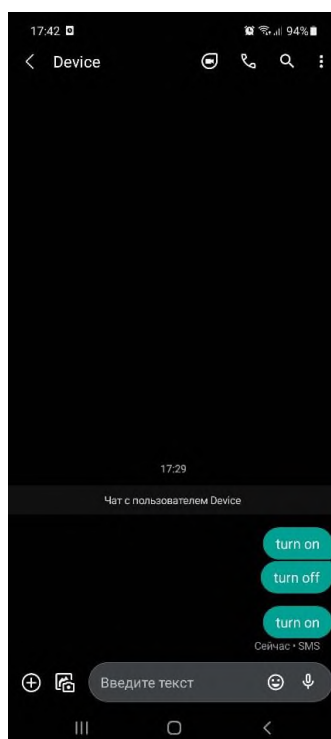


Рис. 3.13. Керування апаратною частиною з допомогою *GSM*

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

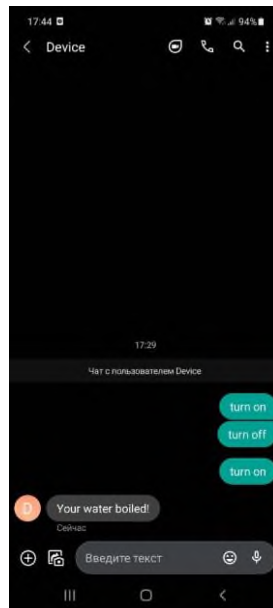


Рис. 3.14. *SMS* повідомлення про завершення роботи

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.

Система управління побутовою технікою управляється або мобільним телефоном, або долонею, або ПК. Вони з'єднані через протокол бездротової мережі (*WAP*), Інтернет або інтрамережу. Він керує різними приладами, такими як мікрохвильова піч, телевізор тощо. Система *HACS* отримує команди від віддалених пристроїв, якими керує користувач. Система, у свою чергу, надсилає команди відповідним пристроям, які виконуватимуть дії. *HACS* відповідає за відстеження стану пристроїв. Якщо щось піде не так, він повідомить користувача, надіславши повідомлення на віддалені пристрої, а також у відділ екстреної допомоги, якщо це необхідно.

ВИСНОВКИ

Під час написання даної роботи було розроблено систему 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..

ІЧ промені частотою 38 кГц, що генеруються пультом дистанційного керування, приймаються модулем ІЧ-приймача *TSOP1738* схеми. Вивід 1 *TSOP1738* підключений до землі, висновок 2 підключений до джерела живлення через резистор *R5*, а вихід береться з виводу 3. Вихідний сигнал посилюється транзистором *T1 (BC558)*.

Посилений сигнал подається на висновок 14 тактового лічильника *IC CD4017 (IC1)*. Контакт 8 мікросхеми *IC1* заземлений, контакт 16 підключений до *Vcc*, а контакт 3 підключений до *LED1* (червоний), який світиться, вказуючи, що прилад вимкнено.

Вихідний сигнал *IC1* береться з його контакту 2. *LED2* (зелений), підключений до контакту 2, використовується для вказівки «увімкненого» стану приладу. Транзистор *T2* (*BC548*), підключений до контакту 2 *IC1*, керує реле *RL1*. Діод *1N4007* (*D1*) виконує роль вільного ходу. Пристрій, яким потрібно керувати, підключається між полюсом реле та нейтральною клемою мережі. Він підключається до живої клеми мережі змінного струму через нормально відкритий (*N/O*) контакт, коли реле подається під напругу.

Система управління побутовою технікою управляється або мобільним телефоном, або долонею, або ПК. Вони з'єднані через протокол бездротової мережі (*WAP*), Інтернет або інтрамережу. Він керує різними приладами, такими як мікрохвильова піч, телевізор тощо. Система *HACS* отримує команди від віддалених пристроїв, якими керує користувач. Система, у свою чергу, надсилає команди відповідним пристроям, які виконуватимуть дії. *HACS* відповідає за відстеження стану пристроїв. Якщо щось піде не так, він повідомить користувача, надіславши повідомлення на віддалені пристрої, а також у відділ екстреної допомоги, якщо це необхідно.

Системний адміністратор системи *HACS* має можливість додати новий пристрій або видалити існуючий. Системний адміністратор має можливість додати новий віддалений пристрій і налаштувати його за допомогою *HACS* або видалити існуючий, коли він не використовується. Також системний адміністратор може створити обліковий запис для нового користувача або видалити наявний обліковий запис, якщо він більше не використовується.

У цьому дослідженні запропоновано різні можливі способи використання мобільної телефонії за допомогою існуючих послуг, але переосмислюючи тривіальні цілі, яким вони служать. Досліджено різні способи використання стільникових телефонів, щоб вийти за межі здійснення дзвінків і надсилання SMS, і розробили кілька способів впровадження дистанційного керування, яке є «дистанційним» і може використовуватися для керування побутовою технікою розумного дому. Існує два підходи до управління побутовою технікою. Один з

них – створити з нуля контролер для керування побутовою технікою за допомогою дротового з'єднання. Основна проблема полягає в тому, що підключення цього контролера стають незграбними, а також ненадійними. Ми з'ясували, що для досягнення цієї мети краще використовувати наявні домашні контролери, ніж розробляти їх з нуля, оскільки на ринку є стандартизовані домашні контролери, які пропонують широкий спектр керованих приладів. X10, Insteon, Z-Wave і ZigBee є доступними кандидатами для виробників домашніх контролерів. Ми залишили частину керування побутовою технікою X10 і зосередилися на зв'язку між мобільним телефоном і контролером X10 для дистанційного керування контролером X10. Ми обираємо X10 перед іншими через його широку доступність.

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. *Işilak, A. H. (2010), "Smart home applications for disabled people by using wireless sensor network", Unpublished BSc Project Report, Yeditepe University, Istanbul 57pp.*
2. *Chander, V. R. (2012), "Novel Ubiquitous Interoperable Context-aware Smart Environments through Web Services",*
3. *Proceedings of International Multi Conference of Engineers and Computer Scientists IMECS 2012, Hong Kong March 14-16, Vol. 1, pp. 45 – 46.*
4. *Weiser, M. (1991), "The computer of the twenty-first century", Scientific American, Vol. 265, No. 3 pp. 94 - 104.*
5. *Piyare, R and Lee S.R. (2013), "Smart Home-Control and Monitoring System Using Smartphone", Proceedings, The 1st International Conference on Convergence and it's Application ICCA 2013, ASTL, Vol. 24, pp. 83 - 86.*
6. *Kelly, S.D.T., Suryadevara N.K., and Mukhopadhyay S.C. (2013), "Towards the Implementation of IoT for Environmental Condition Monitoring in Homes",*
7. *Institute of Electrical and Electronic Engineers IEEE, Vol. 13, pp. 3846 – 3853.*
8. *Yan, M. Shi, H. (2013), "Smart Living Using Bluetooth-Based Android Smartphone", International Journal of Wireless & Mobile Networks (IJWMN), Vol. 5, No. 5, pp. 65*
9. *72. Condition Monitoring in Homes", Institute of Electrical and Electronic Engineers IEEE, Vol. 13, pp. 3846 – 3853.*
10. *Park, B. (2012), "Mobile IP-based architecture for smart homes", International Journal of Smart Home, Vol. 6, No.1, pp. 29 – 36.*
11. *Polaris PWK 1725CGLD [Електронний ресурс] // tekhnik.top - Режим доступу: <https://tekhnik.top/rejting-umnyh-chajnikov-2019/#Polaris-PWK-1725CGLD-WIFI-IQ-Home>*

12. *How to Achieve Ranges of over 1 Km using Bluetooth Low Energy* [Электронный ресурс] // www.novelbits.io - Режим доступа: <https://www.novelbits.io/long-range-bluetooth-coded-phy/>

13. 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.

14. Beazley D. *Python essential reference* // BHV. – 2015. – 734 с.

15. Jones C.A., Drake F.L. *Python & XML* // BHV. – 2014. – 807 с.

16. F Mattern, C Floerkemeier. *From the Internet of Computers to the Internet of Things. From active data management to event-based systems and more*. Springer. 2010: 242-259.

17. R Piyare. *Internet of things: ubiquitous home control and monitoring system using android based smart phone*. *International Journal of Internet of Things*. 2013; 2: 5-11.

18. YP Tsou, JW Hsieh, CT Lin, CY Chen. *Building a remote supervisory control network system for smart home applications*. In *Systems, Man and Cybernetics, SMC'06. IEEE International Conference on*. 2006: 1826-1830.

19. A Brush, B Lee, R Mahajan, S Agarwal, S Saroiu, C Dixon. *Home automation in the wild: challenges and opportunities*. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011: 2115-2124.

20. S Bhattacharjee, A Kumar, J Roy Chowdhury. *Appliance classification using energy disaggregation in smart homes*. In *Computation of Power, Energy, Information and Communication (ICCPEIC), 2014 International Conference on*. 2014: 1-6.

21. J Choi, D Shin, D Shin. *Research and implementation of the context-aware middleware for controlling home appliances*. *IEEE Transactions on Consumer Electronics*. 2005; 51: 301-306.

22. J Nichols, BA Myers. *Controlling home and office appliances with smart phones*. *IEEE Pervasive Computing*. 2006; 5: 60-67.

23. D Pishva, K Takeda. *Product-based security model for smart home appliances*. *IEEE Aerospace and Electronic Systems Magazine*. 2008; 23.

24. SH Baeg, JH Park, J Koh, KW Park, MH Baeg. *Building a smart home environment for service robots based on RFID and sensor networks. In Control, Automation and Systems, 2007. ICCAS'07. International Conference on. 2007: 1078-1082.*
25. L Zhang, H Leung, KC Chan. *Information fusion based smart home control system and its application. IEEE Transactions on Consumer Electronics. 2008; 54.*
26. W ZhenXing, S LinXiang, W ShuTao. *Embedded Web Server Architecture For Mobile Phone. In Future Networks, 2009 International Conference on. 2009: 208-211.*
27. C Felix, IJ Raglend. *Home automation using GSM. In Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on. 2011: 15-19.*
28. ДСТУ 2851-94 Програмні засоби ЕОМ. Документування результатів випробувань.
29. НД ТЗІ 1.1-003-99. Термінологія у області захисту інформації в комп'ютерних системах від несанкціонованого доступу. // Департамент спеціальних телекомунікаційних систем і захисту інформації Служби безпеки України. – Київ, 1999.
30. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.

Додаток А

Програмний код розробленого пакета