

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота магістра

**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА ТЕХНОЛОГІЯ КЕРУВАННЯ
МЕРЕЖАМИ НА ОСНОВІ КОНЦЕПЦІЇ ІНТЕРНЕТУ РЕЧЕЙ**

Здобувач освіти гр. ІН.м-13

Тимофій ЧЕЧІЛЬ

Науковий керівник,
к.ф.-м.н., ст.викладач

Дмитро ВЕЛИКОДНИЙ

В.о. завідувача кафедри
к.т.н., доцент

Ігор ШЕЛЕХОВ

Суми-2022

Факультет ЕІТ Кафедра Комп'ютерних наук
Спеціальність «122 - Комп'ютерні науки»

Затверджую:
зав.кафедрою _____
“ _____ ” _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ
Чечіль Тимофій Юрійович
(прізвище, ім'я, по батькові)

1. Тема проекту Інформаційно-комунікаційна технологія керування мережами на основі концепції інтернету речей
затверджена наказом по університету від «__» _____ 2022 р. № _____

2. Термін здачі студентом закінченого проекту _____

3. Вхідні дані до проекту _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз предметної області; 2) Постановка задачі та методи дослідження технологій; 3) Вибір інструменту для створення інформаційно-комунікаційних мереж; 4) Вибір засобів для програмної реалізації; 5) Проектування інформаційно-комунікаційної мережі в ПЗ Cisco Packet Tracer; 6) Тестування розробленої мережі.

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

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Науковий керівник _____

Завдання прийняв до виконання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проєкту (роботи)	Термін виконання проєкту (роботи)	Примітка
1.	Огляд можливих варіацій рішень розробки		
2.	Постановка задачі та групування за напрямками завдань дослідження		
3.	Програмна реалізація		
4.	Тестування створеної мережі		
4.	Оформлення пояснювального звіту		

Здобувач вищої освіти _____

Науковий керівник _____

РЕФЕРАТ

Записка: 60 стр., 47 рис., 1 додаток, 26 джерел.

Об'єкт дослідження — розробка мережі та її використання.

Мета роботи — створити інформаційно-комунікаційну технологію керування мережами на основі концепції інтернету речей..

Методи дослідження — розробка мережі за допомогою використання інструментів Cisco.

Результати — спроектована та розроблена інформаційно-комунікаційної технології керування мережами на основі концепції інтернету речей.

CISCO, WIRELESS, РОЗУМНИЙ ОФІС, LAN, МОДУЛЬ, МЕРЕЖА, CONFIGURATION, CONDITION, МІКРОКОНТРОЛЕРИ, ІНТЕРНЕТ-СИСТЕМА.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Аналіз предметної області.....	7
1.2 Аналіз програмних продуктів розроблених систем керування мереж.....	9
1.3. Мета та задачі дослідження	13
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ.....	15
2.1. Вибір інструменту для створення інформаційно-комунікаційних мереж .	15
2.2. Вибір засобів для програмної реалізації.....	19
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	22
3.1 Проектування інформаційно-комунікаційної мережі Розумного офісу в ПЗ Cisco Packet Tracer.....	22
3.2 Тестування за різних умов розробленої мережі та модулів	34
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТОК А. ЛІСТИНГ КОДУ ДЛЯ МІКРОКОНТРОЛЛЕРА	49

ВСТУП

Технології змінюють світ. Отже, чому б не використати сучасні інструменти та передові технології, щоб зробити свій виставковий стенд привабливішим?

В останні роки відбулися серйозні зміни у використанні технологій, і бренди отримали величезну перевагу в охопленні своєї аудиторії. Це чудова можливість привернути увагу, вільно залучати та перетворювати потенційних клієнтів на постійних клієнтів.

Протягом багатьох років торгові виставки були успішною платформою для компаній, щоб зв'язатися зі своєю цільовою аудиторією та зміцнити свою ділову мережу. Оскільки технології перевершують традиційні методи, все більше експонентів звертаються до більш привабливих та інтерактивних концепцій дизайну стендів для просування своїх брендів. Оскільки технології продовжують розвиватися, відвідувачі очікують побачити дизайн стендів, який творчо включає новітні технології. У минулому конструкція виставкового стенду включала прості коробки модульної форми з зображеннями бренду, але з часом все змінилося.

Сьогодні відвідувачі шукають досвіду, який викликає справжні емоції, і використання технологій занурення зробить його незабутнім. Отже, було прийнято рішення по розробці інформаційно-комунікаційної технології керування мережами на основі концепції інтернету речей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

За останні два роки в офісному дизайні відбулися неймовірні інновації. Тепер ми краще усвідомлюємо, як ми використовуємо простір на робочому місці. Звичайно, ергономічний простір завжди був елегантним та зручним у використанні. Для створення плавного гібридного робочого процесу та повернення співробітників в офіс ефективний дизайн робочого місця є ключовим. Це означає приватні та продезінфіковані приміщення, коли потрібна приватність, і гарні добре провітрювані приміщення для групової та спільної роботи.

Саме тут дебютували розумні кабінки. Розумна кабінка — це акуратно сконструйований корпус будь-якого розміру зі звукоізоляційних панелей з оргскла, які поєднуються в будь-якому офісному, чи іншому, приміщенні [1]. Ваші розумні кабінки можуть бути окремо стоячими або вбудованими в кутовий конференцзал. Кожна з них має панель керування на дверях для ексклюзивного, запланованого та легко відстежуваного доступу та розроблена, щоб відповідати найсучаснішим потребам будь-якого сучасного дизайну офісу.

Розумні кабінки — це не просто корпуси, це портативні розумні офіси з вбудованим резервуванням і відстеженням. Багато професіоналів вже знайомі з ідеєю бронювання конференц-залу або особистого кабінету, коли виникає така необхідність [2]. Кожна розумна кабінка має вбудовану клавіатуру, яка підключається до вбудованого програмного забезпечення для бронювання. Співробітники можуть налаштувати розклад, записатися на час роботи в кабінці та знати, коли кабінка зайнята або збирається бути зайнятою, на основі підключеної веб-програми.

Керівники підприємств також можуть відстежувати розклад і доступ до кожної кабінки за допомогою унікального контролю за тим, як використовується робоче місце. Це дає вам можливість побачити, наскільки ваш поточний макет і

робочий процес корисні для команди та як ваші розумні кабінети використовуються групами та окремими особами в рамках повсякденної роботи [3]. Якщо ви вирішите змінити розташування своїх розумних кабінетів, зміни у відстеженні доступу можуть дати вам зрозуміти, чи новий макет більш чи менш ергономічний, ніж попередній.

Окрім цього часто використовуються технологія Smart-Home на основі програми використовує вашу домашню мережу для зв'язку з хмарою. Хмарні технології є життєво важливою частиною Інтернету речей, і обидві набули великої популярності за останні кілька років.

Потенціал Home Automation продовжує зростати. Такі технології можуть використовувати для забезпечення безпеки будинку, керування побутовою технікою, моніторингу споживання енергії, як особистого помічника, для догляду за людьми похилого віку чи інвалідами тощо (рис.1.1).

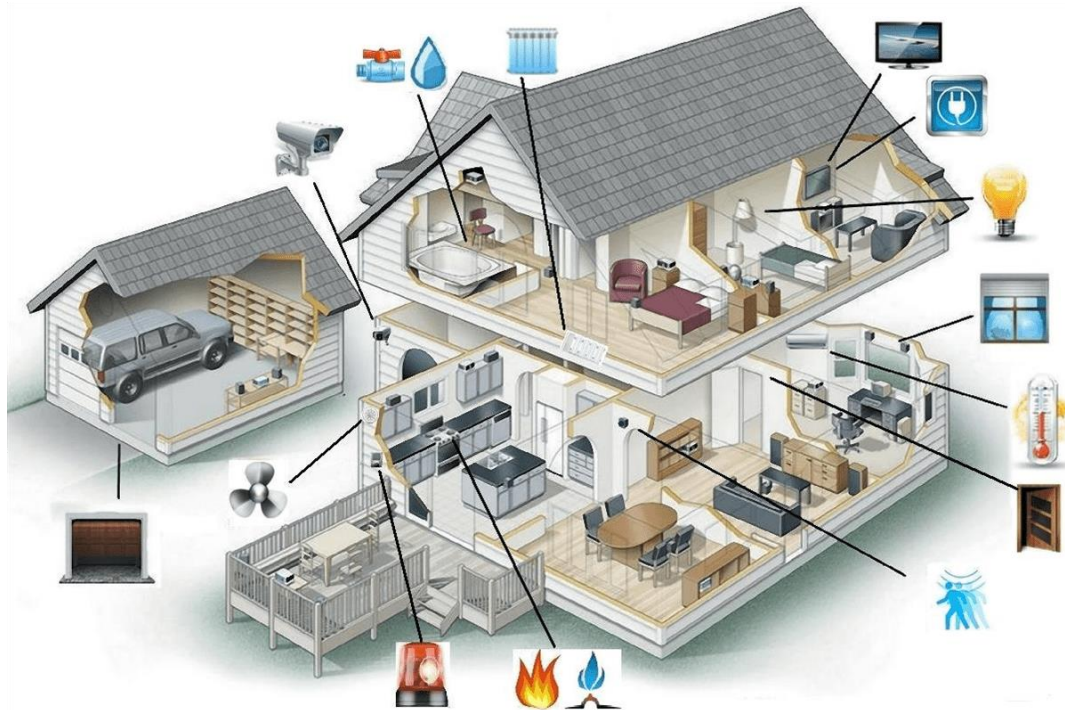


Рисунок 1.1 – Приклад об'єму можливих варіацій у розумному будинку

Дана тематика також є актуальною у демонстрації продуктів. Адже такі продукти, як автомобілі та велосипеди, важко представити, оскільки вони зазвичай потребують великих площ для демонстрації транспортних засобів. AR і VR є рішенням для таких продуктів [4]. Ця техніка дозволяє відвідувачам досліджувати продукт компанії та взаємодіяти з ним на мобільному пристрої або в окулярах AR.

І останнє, але не менш важливе це те, що дані технології також є ідеальним рішенням для роботодавців, яким потрібна швидка відповідь щодо безпеки на місці COVID. Багато професіоналів все ще стурбовані поверненням на робоче місце з командними офісами, гарячими робочими столами та робочими просторами відкритого планування.

1.2 Аналіз програмних продуктів розроблених систем керування мереж

На сьогоднішній час, коли більшість домівок наповнені дорогими гаджетами, а смартфон або планшет вартістю вимірюється тисячами, безпека дому завжди є головним пріоритетом. Оскільки розумні будинки з роками зростали, люди бачили різноманітні пристрої, як-от камери, регулятори опалення та дверні дзвінки. Єдине, чого не вистачає, це система сигналізації, яка принаймні є бездротовою, але все ще застрягла зі старими датчиками та головною панеллю керування, з незначною інтеграцією з розумним будинком або без неї.

Для порівняння програмних продуктів-аналогів було обрано декілька систем, а саме Ajax та Hikvision AX Pro.

Розглянемо систему Ajax. Безпека власної оселі завжди була великою проблемою. На щастя, це пішло в минуле з появою розумних систем сигналізації від різних виробників [5]. Популярним на ринці представником такого типу систем є Ajax. Ajax – це українська компанія, заснована в 2011 році, зі штаб-квартирою, науково-дослідними центрами та виробництвом у Києві, Україна. Компанія також постачає професійні системи безпеки в 93 країни світу (рис.1.2).

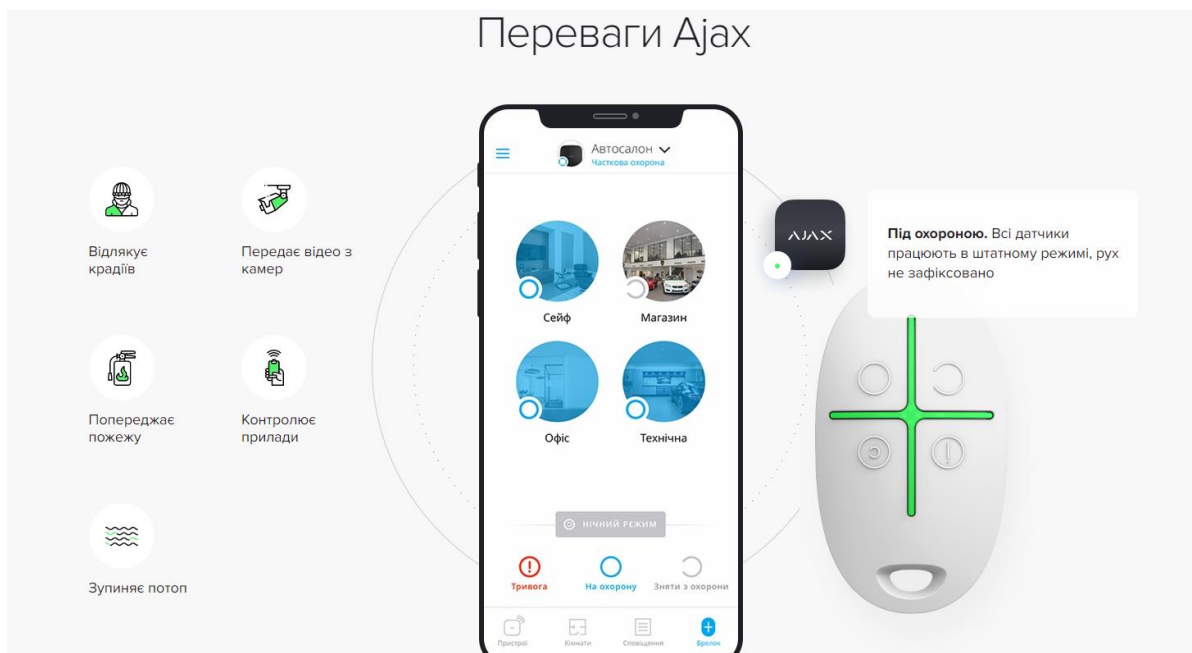


Рисунок 1.2 – Інформація про Ajax

Перейдемо до використання пристроїв Ajax. Після простого встановлення та налаштування можна протестувати всі пристрої та їх різноманітні функції. Компанія Ajax створила пристрої, які працюють ідеально, точно так, як рекламується. Головна перевага в тому, що вони надзвичайно прості у використанні.

Від Hub 2, дверних датчиків, PIR-детектора з фотоперевіркою та всіх інших аксесуарів, які представлені у інтернет магазинах і офлайн магазинах, не виявлено жодних проблем чи збоїв у додатку, непрацюючих функцій чи будь-яких негативних наслідків із системою.

Розглянемо, як саме це працює. Коли спрацьовує будь-яка подія, будь то зняття кришки з пристрою, втрата живлення концентратора, підключення до Інтернету, або активація будь-якої іншої функції, користувач миттєво отримує сповіщення на свій смартфон [6]. Крім того, із додатковими налаштуваннями інформація може передаватися й на розумний годинник. Коли система поставлена на охорону, якщо спрацьовує PIR-датчик разом із миттєвим сповіщенням,

приблизно через 10 секунд ви зможете переглянути перше зображення, зроблене PIR-датчиком.

Підсумуємо інформацію та сформує головні переваги систем безпеки [7] та інших пристроїв Ajax:

- Простий у використанні та багатофункціональний додаток;
- Вражаючий технічний та візуальний дизайн;
- Компоненти професійного рівня та високої якості;
- Надзвичайно комплексний у кожному відділі;
- Величезний робочий діапазон;
- Працює ідеально, без помилок;
- Можливість зовнішнього моніторингу.

Не дивлячись на це, системи безпеки та інших пристроїв продуктової компанії Ajax має й певні мінуси:

- Висока ціна закупівлі;
- Тільки професійне встановлення;
- Відсутня реєстрація роботи датчика, коли система знята з охорони;
- Не може працювати планшетами.

Бездротова сигналізація Hikvision AX Pro. Розглянемо також бездротову сигналізацію Hikvision AX Pro. Серія AXE від Hikvision — це потужна та безпечна бездротова система охоронної сигналізації, яка розроблена таким чином, щоб її було легко встановити та розширити. На відміну від інших провідних систем сигналізації, лінійка AXE PRO повністю контролюється та налаштовується за допомогою додатків Hik-Connect і Hik-ProConnect(рис.1.3). Це пропонує централізований багатосайтовий і багатокористувацький портал керування [8], який дає змогу віддалено працювати з інтерфейсом, додавати та налаштовувати пристрої, ставити й зняти з охорони одним дотиком, статус системи та пристрою, а також переглядати відео та GIF, чи інші формати.



Рисунок 1.3 – Інформація про варіації Hikvision AX Pro

Сигналізація AXE PRO починається з панелі керування, яка діє як центр для всієї системи. Далі можна додати широкий спектр пристроїв, включаючи внутрішні та зовнішні сповіщувачі, звукові сигналізатори, повторювачі, реле та пристрої охорони, які можна використовувати для створення спеціальної системи сигналізації, розробленої відповідно до ваших вимог (рис.1.4).

Connectec має на складі великий асортимент бездротових концентраторів сигналізації AX PRO, детекторів, датчиків, реле та пристроїв охорони, які можна придбати окремо або як повноцінний будинковий комплект.

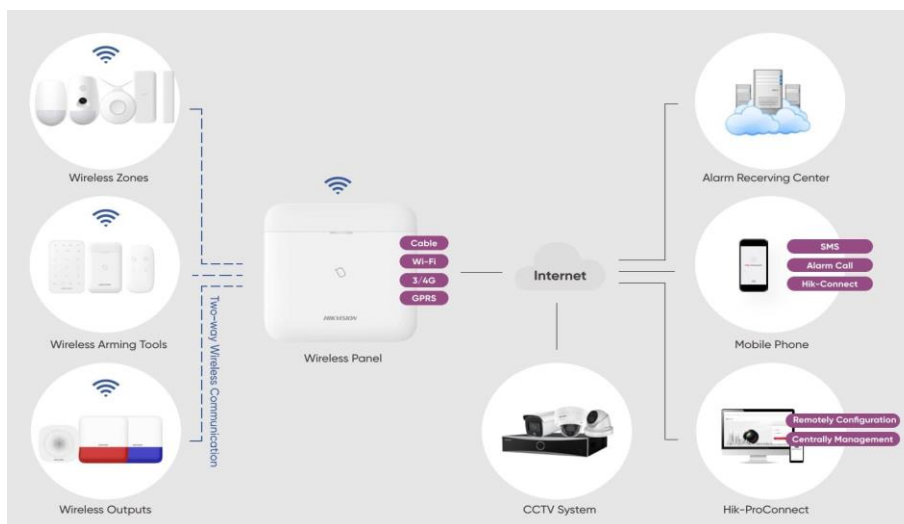


Рисунок 1.4 – Використання Hikvision AX Pro

Система AXE PRO пропонує прості та гнучкі варіанти встановлення без проводів, які обмежують налаштування. Просто встановіть концентратор і пов'язані пристрої за допомогою гвинтів або клею залежно від монтажної поверхні. Після того, як кожен пристрій знайдено, його можна додати до мережі за допомогою програми для смартфона.

Усю систему безпеки Hikvision AX PRO і Hikvision CCTV можна контролювати за допомогою додатків для смартфонів Hik-Connect і Hik-ProConnect [9]. Він пропонує зручний інтерфейс, можливість отримувати сповіщення щоразу про інцидент, швидку реакцію інсталяторів і цілодобовий хмарний сервіс створюють професійний досвід.

Додаток також можна використовувати для перевірки стану системи, віддаленої постановки та зняття з охорони, а також перегляду відеоканалів відеоспостереження, файлів і сповіщень про тривогу в будь-якій точці світу.

1.3. Мета та задачі дослідження

Головною метою розробки є інформаційно-комунікаційної технології керування мережами на основі концепції інтернету речей. Під час планування було поставлені основні задачі та цілі, що повинні бути досягнуті під час роботи.

Об'єкт дослідження: структура керування мережами на основі концепції інтернету речей.

Предмет дослідження: взаємодія усіх блоків один між одним у розробленій мережі.

Практичне значення: спроектована та розроблена інформаційно-комунікаційної технології керування мережа на основі концепції інтернету речей.

Головні задачі, які повинні бути виконані:

- Підготувати модуль cluster Internet, що забезпечує доступ до серверів;
- Побудовано комунікаційну мережу з використанням пристроїв Інтернету речей;

- Налаштувати внутрішній сервер;
- Налаштувати під'єднатися до створеного внутрішнього серверу.

Задачі можна поділити на декілька типів, а саме функціональні та не функціональні. До функціональних можна віднести як те, що пов'язано саме із інструментарем, так його використання при розробці інформаційно-комунікаційної технології керування мережами на основі концепції інтернету речей. Не функціональні – пов'язані із підготовкою даних, інформації та саме підготовкою до здачі кваліфікаційної роботи магістра.

Визначимо перелік не функціональних задач:

- Підготувати всі дані, що необхідні для реалізації інформаційної технології по керування мережами;
- Виконати планування робіт, для поступового виконання задачі;
- Проаналізувати інструменти для створення інформаційно-комунікаційних мереж;
- Реалізація взаємодії блоків у мережі.

Перейдимо до функціональних задач інформаційної технології по керування мережами:

- Підготувати модуль cluster Internet, що забезпечє доступ до серверів;
- Побудовано комунікаційну мережу з використанням пристроїв Інтернету речей;
- Налаштувати внутрішній сервер;
- Налаштувати під'єднання до створеного внутрішнього серверу.

Результатом роботи інформаційної технології повинна бути розроблена структурно вірна мережа, що буде відповідати всім вимогам.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ

2.1. Вибір інструменту для створення інформаційно-комунікаційних мереж

Під час підготовки до роботи з CCNA, CCNP чи CCIE, чи інше, вибір правильного симулятора або емулятора мережі може бути важким рішенням, але вагомим у подальшій роботі. Буває багато способів для вирішення даної задачі. Не дивлячись на кількість інформації, не існує ідеального рішення [10, 14]. Для пошуку найкращого способу вирішення задачі та побудові мережі потрібно проаналізувати та зробити самостійний вибір саме під обрані задачі. Адже на перевірки кожної програми окремо піде занадто багато часу.

Незалежно від того, отримав користувач інформацію та поради від колеги, керівника чи соціальних мереж, кожен знаходить щось специфічне та потрібне саме під обрані задачі. Як наприклад, хто переважно використовує GNS3, завжди стверджуватиме, що він кращий за VIRL, чи навпаки [15].

Основна перевага симуляторів полягає в тому, що вони, як правило, досить легкі у використанні та моделювання мереж. Програмне забезпечення симулятора може працювати практично на будь-якому сучасному комп'ютері, не турбуючись про вимоги до процесора, пам'яті, пам'яті чи іншого.

Для порівняння було обрано декілька симуляторів, а саме:

- Симулятор SNMP;
- Boson Netsim;
- Cisco Packet Tracer.

Симулятор SNMP — це якісна програма на основі Java, яка може імітувати агентів SNMP [12]. Оскільки він повністю написаний «чистою» мовою Java, він добре працює на всіх платформах, на яких встановлено віртуальну машину Java, включаючи Windows, Linux та інші.

Головні переваги даного симулятора:

1. Гнучкий і простий у використанні. Це полягає у тому, що симуляція SNMP дозволяє кожному розробляти, тестувати та навчати застосування додатків SNMP без покупки додаткових інструментів;
2. Це дозволяє легко записувати та відтворювати пристрої SNMP. Можна записати вузли MIB агентів SNMP у файл графічний інтерфейс. Записаний файл даних також модифікується для імітації різних конфігурацій;
3. Він пропонує кілька номерів агентів лише в одній JVM;
4. Включає повну підтримку V3, V2C та SNMPV1;
5. Пропонує підтримку динамічного видалення та створення рядків;

Вони виробляли різні продукти, що можна використовувати при розробці мережі. Набір симуляторів MIMIC, використовується для тестування розширюваності та конфігурації. Обраний симулятор, а саме MIMIC SNMP – продуктом, який керує різними пристроями, в залежності від вказаних налаштувань [13]. Симулятор MIMIC IoT створює велике тестове середовище, що використовується для перевірки датчиків і пристроїв.

SNMP Simulator покращує якість керування додатками та підтримує тестування на сотнях пристроїв. Він також підтримує динамічне створення рядків через новий екземпляр, який дозволяє редагувати екземпляр інформації відповідно до ваших потреб. Це зменшує вартість пропускної здатності, доступної для різних пристроїв.

До недоліків можна віднести – немає способу перевести мережеві топології SNMP у «режим симуляції», як це можна зробити за допомогою інших технологій.

Розглянемо також симулятор NetSim. Налаштування пристрою Boson NetSim запозичив інтуїтивно зрозумілу систему налаштування пристрою GNS3. У NetSim, якщо мережевий пристрій можна налаштувати за допомогою додавання модулів, програмне забезпечення явно запитує, які модулі ви хочете вставити в пристрій, коли пристрій додається до топології мережі.

Подібно до Packet Tracer, кожна лабораторія оцінюється за завершенням і точністю в програмі [15]. Однак Boson NetSim відрізняє себе від Packet Tracer здатністю відстежувати виконання лабораторних робіт із програми. Ця функція дозволяє миттєво переглядати ваші завершені лабораторні роботи, лабораторні роботи, які ви спробували виконати, і лабораторні роботи, які ви не зробили.

До переваг можна віднести реалістичні функції терміналу. Копіювати та вставляти дуже реалістично в NetSim. У реальному терміналі виділений текст автоматично копіюється в системний буфер обміну. Ви вставляєте, клацаючи правою кнопкою миші. Термінал NetSim імітує цю функцію [14-15]. Ця реалістична функція відрізняє NetSim від Packet Tracer, де вона не підтримується.

Тут варто зазначити, що навчальна програма Cisco NetAcad також пропонує лабораторні роботи за допомогою вправ Packet Tracer, а також практичні лабораторні роботи з фізичним обладнанням. Однак у цій статті розглядається саме програмне забезпечення Packet Tracer, а не навчальна програма NetAcad . Програмне забезпечення Boson обертається навколо придбаних лабораторних заходів на додаток до пісочниці. Це особливість і перевага, яку NetSim має перед Packet Tracer, яка вимагає від вас творчих або вихідних лабораторій від NetAcad.

Не дивлячись на це, симулятор NetSim має свої недоліки. Відсутність налаштування топології . На відміну від більшості інших мережевих симуляторів і емуляторів, NetSim не має способу додавання розфарбованих форм до топології мережі. Окрім мережевих пристроїв і з'єднань, до топології можна додавати лише текстові примітки та мітки. Це досить незначний момент, але він важливий під час репетиції деяких тем.

Не дивлячись на це, даний додаток має крос-платформнену сумісності. NetSim доступний лише для операційних систем Windows — зокрема Windows 10, 8, 7, Vista та XP. Він не підтримується в macOS або будь-якому дистрибутиві

Linux. У результаті користувачі macOS і Linux не можуть використовувати NetSim без віртуальної машини Windows.

Без режиму симуляції . Немає способу перевести мережеві топології NetSim у «режим симуляції», як це можна зробити за допомогою Packet Tracer. У NetSim неможливо переглянути вміст окремих пакетів, коли вони проходять мережею. Усі мережеві пристрої моделюються в режимі реального часу. Як наслідок, ви не можете візуалізувати шлях пакета у вашій змодельованій мережі так само, як у Packet Tracer.

Packet Tracer зазвичай використовується як симулятор мережі, адже він має багато переваг. Перш за все, Packet Tracer безкоштовний. Cisco Packet Tracer можна використовувати будь-якому користувачеві [25-26]. Адже даний додаток можна завантажити та використовувати абсолютно на будь-якій системі. Для цього потрібно лише створити безкоштовний обліковий запис Cisco Networking Academy та зареєструвати Packet Tracer.

Також кросплатформено-сумісний із іншими системами. На момент написання Packet Tracer працює на більшості операційних систем, включаючи всі активні операційні системи Windows, macOS High Sierra та Mojave, а також Ubuntu 14.04 LTS.

Packet Tracer може бути використаний користувачами додатку й як мережевий додаток для проектування [24], розробки та тестування-симулювання віртуальної мережі через реальну мережу. Цікава властивість додатку в тому, що користувачі можуть працювати над однією мережевою топологією. Це особливість багатокористувацького режиму.

Окрім цього, Cisco Packet Tracer широко використовується для організації значної командної роботи, як наприклад, проведення ігор чи змагань між віддаленими учасниками.

Cisco Packet Tracer має багато покращень, і версії також є проблемою. Файли Packet Tracer, створені за допомогою версії 7.2.1 [16], несумісні з попередніми версіями Packet Tracer. Коли ви вперше завантажуєте та встановлюєте Packet Tracer

як новий користувач, ви побачите порожню топологію мережі ізольованого програмного середовища.

Для деяких пісочниці сприймаються як запрошення до експериментів і самостійного навчання будь-яких можливих тем. Декого переповнюють порожні пісочниці [17]. Відсутність інтегрованих лабораторій може бути загрозою для студентів, яким потрібна додаткова структура, щоб розпочати роботу з лабораторіями.

Програмне рішення Cisco Packet Tracer [18] дозволяє імітувати поведінку різних мережевих пристроїв.

- маршрутизатор;
- перемикач;
- бездротові точки доступу;
- комп'ютер;
- мережеві принтери;
- IP-телефон тощо.

Інтерактивний симулятор дає користувач додатку дуже реалістичне відчуття побудови справжньої мережі з десятками чи сотнями пристроїв.

2.2. Вибір засобів для програмної реалізації

Інтернет речей — це не просто набір різних пристроїв і датчиків, які з'єднані через дротові та бездротові канали зв'язку та підключені до Інтернету, це більш прямий зв'язок між фізичним і віртуальним світом, у якому здійснюється спілкування між людьми. і машини.

Для створення платформи IoT було обрано Cisco Packet Tracer 8 – інструмент, який надає різні мережеві компоненти, що імітують реальну мережу. Пристрій має різноманітні можливості IoT, містить розумні пристрої, компоненти, датчики, виконавчі механізми та може емулювати мікроконтролери, такі як наприклад

Raspberry Pi. Стаття поповнює розрив між розробкою IoT та його впровадженням [19]. У ньому описується метод швидкого розгортання, який використовує цифрову платформу, яка включає об'єкти IoT і можливості програмування трьома мовами: Blockly, Python і Javascript.

Дві мови програмування Blockly і Python розглядаються для програмування мікроконтролерів (MCU-PT) і одноплатних комп'ютерів (SBC-PT), датчиків.

Python є мовою загального призначення. Тобто з його допомогою можна створювати найрізноманітніші програми, не призначені для вирішення конкретних завдань.

Існує багато причин популярності Python. Давайте детальніше розглянемо, що робить Coder таким універсальним і простим у використанні:

- Простий синтаксис. Подібно до логічної простої мови, яка полегшує читання та розуміння коду [23]. Це дозволяє швидко створювати та вдосконалювати проекти.

- Універсальність. Python можна використовувати для різноманітних завдань, від веб-розробки до машинного навчання.

- Легко для початківців. Тому він дуже популярний серед початківців програмістів.

- Відкритий код. Він має позитивні наслідки для вільного використання та розповсюдження, навіть у комерційних цілях. Існує великий і зростаючий архів коду, модулів і бібліотек, створених сторонніми користувачами для розширення функціональності Python.

Python дозволяє аналітикам даних та іншим фахівцям використовувати мову для виконання складних статистичних обчислень [20], створення візуалізацій даних, створення алгоритмів машинного навчання та керування даними, аналізу даних та виконання інших завдань, пов'язаних із даними. Вона стала невід'ємною частиною науки.

Python може створювати широкий спектр візуалізацій даних, включаючи лінійні діаграми, стовпчасті діаграми, секторні діаграми, гістограми та 3D-

діаграми. У Python є кілька бібліотек, таких як TensorFlow і Keras, які дозволяють програмістам писати швидші й ефективніші програми для аналізу даних і машинного навчання.

Озираючись навколо, щоб дізнатися, які графічні редактори підходять для початківців у програмуванні на Python, я шукав blocky python. Якщо ви раніше не торкалися Blockly [21-22], це бібліотека для створення графічних інтерфейсів на основі браузера на основі взаємопов'язаних блоків із естетикою стилю Scratch: blockly.

Середовище також підтримує переклад блоків у код, а також код у блоки, тому ви можете вставити фрагмент коду в текстове подання, а потім відобразити еквівалент блоків.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

3.1 Проектування інформаційно-комунікаційної мережі Розумного офісу в ПЗ Cisco Packet Tracer

Першим, але важливим кроком є проектування мережі. В програмному забезпеченні Cisco Packet Tracer 8 побудовано комунікаційну мережу з використанням пристроїв Інтернету речей (рис.3.1).

На представленій схемі зображені такі пристрої та датчики IoT:

- маршрутизатори;
- планшети;
- веб-камери;
- системи сигналізації;
- розумні двері;
- датчики вологості;
- детектори руху;
- вуличне освітлення;
- детектори газу;
- вітряки, автомобілі;
- спринклери.

Окрім домашній шлюз використовується для реєстрації розумних пристроїв у мережі та призначення їм IP-адрес. Після встановлення хмарного WAN-з'єднання шлюз автоматично отримує IP-адресу від маршрутизатора провайдера.

Домашній шлюз має чотири порти Ethernet і бездротову точку доступу з SSID. У цій роботі протоколи WEP/WPA-PSK/WPA2 налаштовані для автентифікації бездротового підключення.



Рисунок 3.1 – Побудова мережі

Хотілось би додати також, що на даній схемі розміщені всі необхідні пристрої, які будуть детально налаштовані у подальшому. Всі вони будуть налаштовані на керування за допомогою бездротового маршрутизатора. Приклад бездротового маршрутизатору представлений на рисунку 3.2.



Рисунок 3.2 – Бездротовий маршрутизатор

Перейдемо до налаштування. Створений додатковий модуль «cluster Internet» забезпечує доступ до серверів які надають послуги інтернету (Internet_Server), до мережі стільникового зв'язку (Central Office Server), та до серверу (Smart_Office_Server). Саме з серверу «Smart_Office_Server» відбувається керування приладами які використовуються в мережі Інтернету речей.

Доступ віддаленого керування надається завдяки зв'язку 3G\4G. Детальна схема налаштування модулю «cluster Internet» представлена на рисунку 3.3.

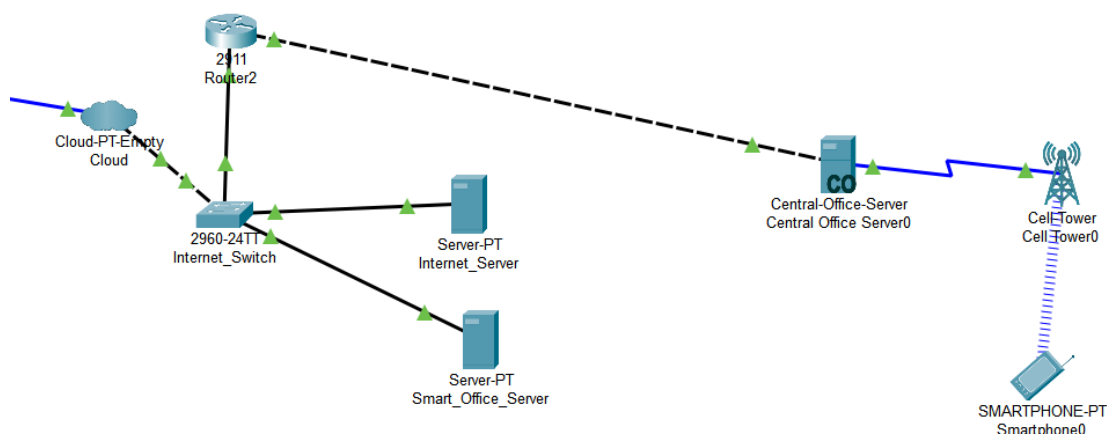


Рисунок 3.3 – Схема налаштування модулю «cluster Internet»

Для налаштування роботи пристроїв, необхідно під'єднатися до серверу «Smart_Office_Server» з будь якого смартфона, ноутбуку чи планшету. Налаштування з'єднання додатково відображається на створеній схемі для кращого розуміння користувачем.

Все це буде складати з себе схему інформаційно-комунікаційної мережі розумного офісу (рис.3.4).

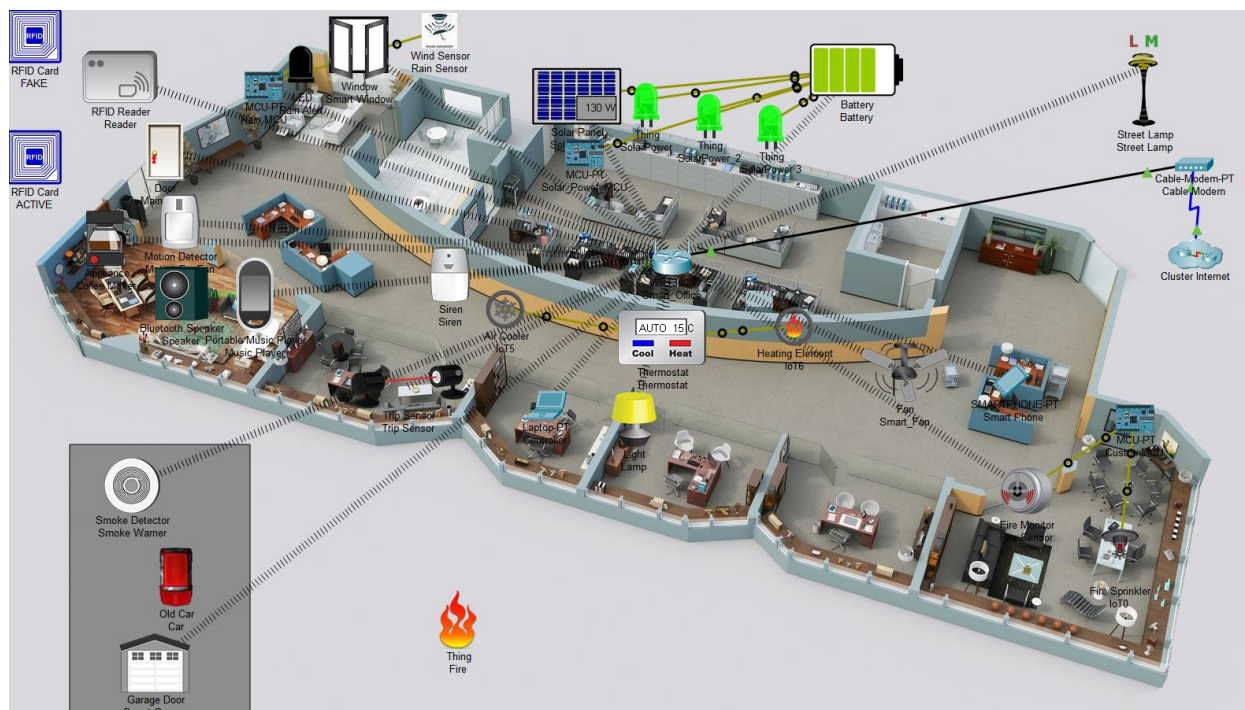


Рисунок 3.4 – Схема інформаційно-комунікаційної мережі розумного офісу

Для подальших дій необхідно перейти на контролер (рис.3.5). Для дізнання статусів необхідно перейти на вкладку «Desktop», тут можна знайти більше інформації за різні налаштування, наприклад:

- Bluetooth;
- TFTP Service;
- loX IDE;
- Email;
- Web Browser.

Для наших задач потрібно переглянути список девайсів які підключені до мережі на вкладці «Home» (рис.3.6).

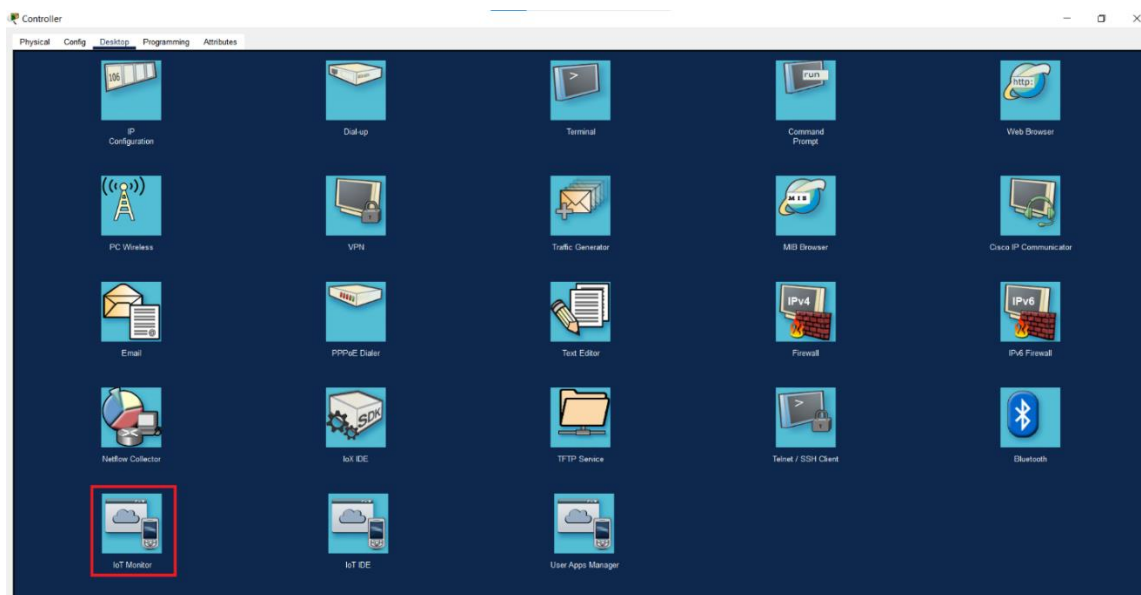


Рисунок 3.5 – «Controller» додатку

На рисунку 3.6 зображено інформація про кожен із девайсів, що використовуються у створеній та представленій раніше схемі.

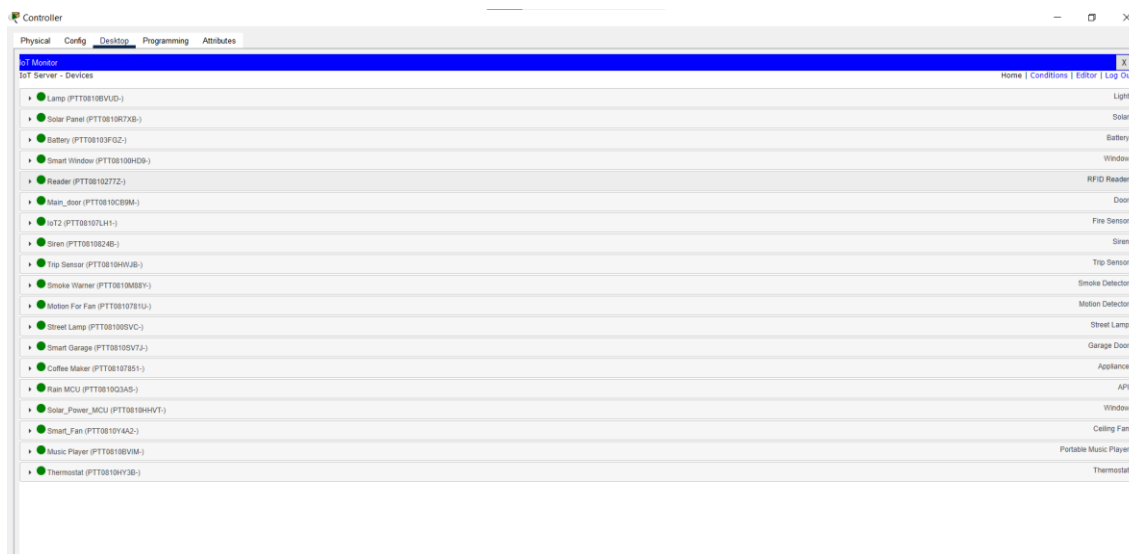


Рисунок 3.6 – Додаткова інформація про пристрої

При розгортанні можна побачити детальну інформацію по кожному із пристроїв. Приклад розгортання та перегляд інформації представлений на рисунку 3.7.

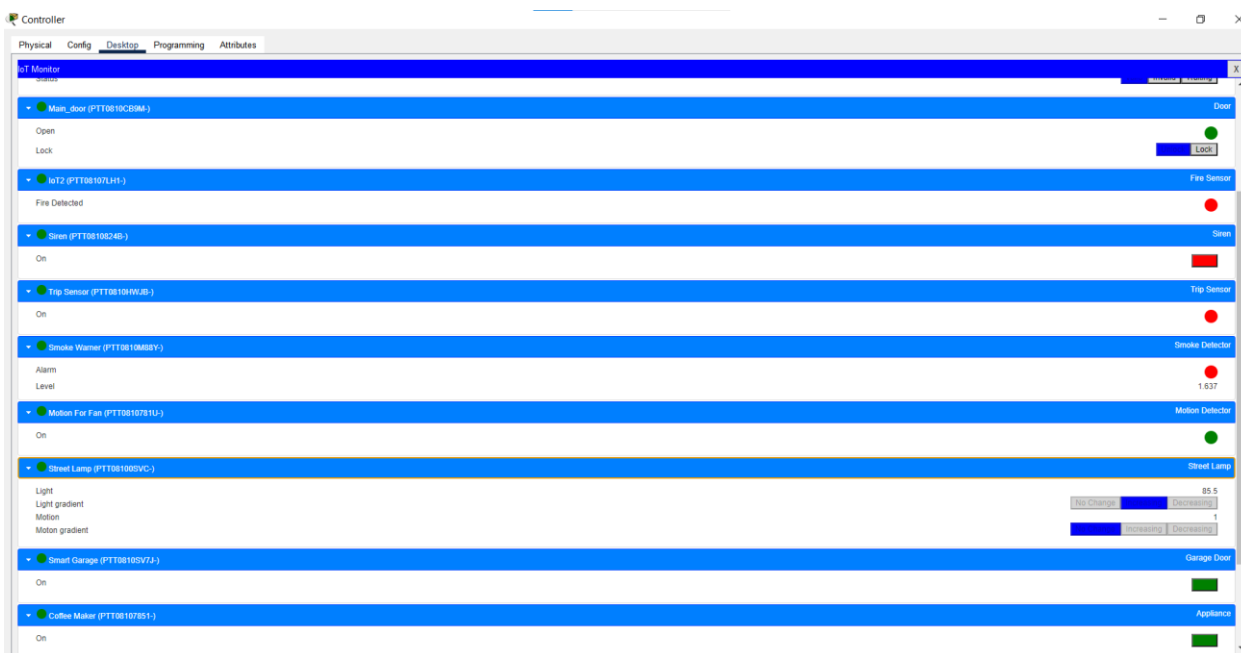


Рисунок 3.7 – Інформація при розгортанні

Для будь яких девайсів, а саме мікрокомп'ютерів, серверів, контролерів налаштувати умови за яких прилади будуть працювати автоматично. Для цього користувачеві необхідно перейти на вкладку «Condition». На цій вкладці користувач може створювати та редагувати умови роботи різних пристроїв (рис.3.8).

Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	rfid_active	Match all: • Reader Card ID >= 555 • Reader Card ID <= 666	Set Reader Status to Valid
Edit Remove	Yes	rfid_disable	Match any: • Reader Card ID < 555 • Reader Card ID > 666	Set Reader Status to Waiting
Edit Remove	Yes	door_open	Reader Status is Valid	Set Main_door Lock to Unlock
Edit Remove	Yes	door_close	Match any: • Reader Status is Invalid • Reader Status is Waiting	Set Main_door Lock to Lock
Edit Remove	Yes	fire_and_siren	IoT2 Fire Detected is true	Set Siren On to true
Edit Remove	Yes	trip_sensor	Trip Sensor On is true	Set Siren On to true
Edit Remove	Yes	smoke_detector	Smoke Warner Alarm is true	Set Siren On to true
Edit Remove	Yes	window_open	Match all: • Street Lams Light >= 10 • Rain MCU Rain Possibility <= 0.0	Set Smart Window On to true
Edit Remove	Yes	window_close	Match any: • Street Lams Light <= 9 • Rain MCU Rain Possibility > 0.0	Set Smart Window On to false
Edit Remove	Yes	garage_open_smoke	Smoke Warner Level > 1	Set Smart Garage On to true
Edit Remove	Yes	garage_close_smoke	Smoke Warner Level <= 1	Set Smart Garage On to false
Edit Remove	Yes	coffee_on	Motion For Fan On is true	Set Coffee Maker On to true
Edit Remove	Yes	motion_fan	Motion For Fan On is true	Set Smart_Fan Status to High

Рисунок 3.8 – Додаткова інформація про стани пристроїв

Для кожного із представлених девайсів у додатку Cisco представлені відповідні налаштування. Як наприклад, для системи опалення можна встановити температури, при яких пристрій буде самостійно тримати значення у вказаних межах (рис.3.9). Відповідно до цього, термостат буде автоматично охолоджувати або опалювати приміщення.

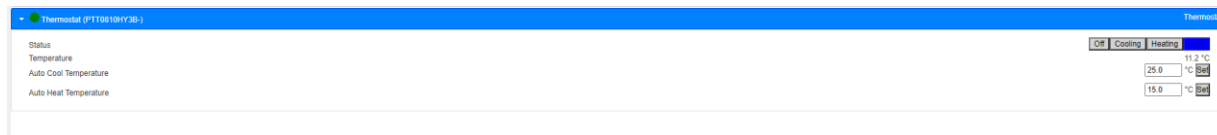


Рисунок 3.9 – Налаштування термостату

На рисунку 3.10 представлено приклад використання налаштованого термостату. Нагріваючи та охолоджуючи пристрої перемикаються автоматично.

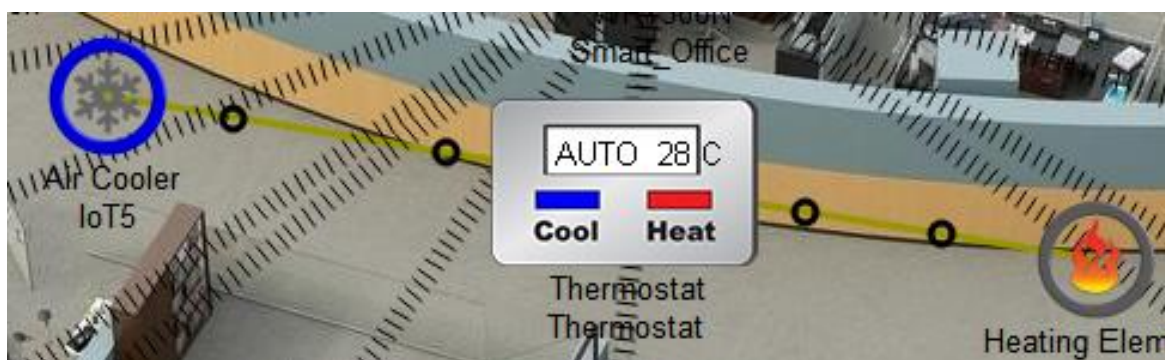


Рисунок 3.10 – Приклад використання налаштованого термостату

Система доступу до приміщення налаштована на роботу RFID зчитувача так, що лише при використанні обмеженого переліку RFID карток, можна буде розблокувати двері (рис.3.11-12).

Враховуючи налаштування, користувач може власноруч вказати перелік ID значень.

Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	rfid_active	Match all: • Reader Card ID >= 555 • Reader Card ID <= 666	Set Reader Status to Valid
Edit Remove	Yes	rfid_disable	Match any: • Reader Card ID < 555 • Reader Card ID > 666	Set Reader Status to Waiting
Edit Remove	Yes	door_open	Reader Status is Valid	Set Main_door Lock to Unlock
Edit Remove	Yes	door_close	Match any: • Reader Status is Invalid • Reader Status is Waiting	Set Main_door Lock to Lock

Рисунок 3.11 – Налаштування системи RFID зчитувача

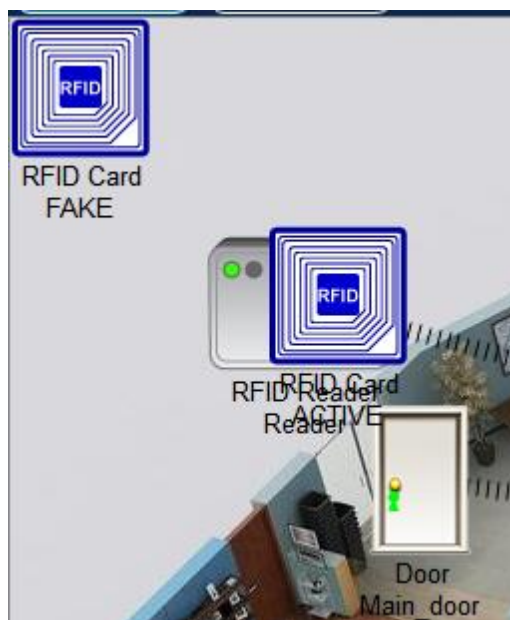


Рисунок 3.12 – Схема RFID зчитувач

Розумне вікно також має умови роботи. Користувач може власноруч встановити години автоматичного відкриття і зачинення вікна. Проте не зважаючи на час, за умови що йде дощ, система сама закриє вікно (рис.3.13-14).

Edit Remove	Yes	window_open	Match all: • Street Lamp Light >= 10 • Rain MCU Rain Possibility <= 0.0	Set Smart Window On to true
Edit Remove	Yes	window_close	Match any: • Street Lamp Light <= 9 • Rain MCU Rain Possibility > 0.0	Set Smart Window On to false

Рисунок 3.13 – Налаштування системи розумного вікна

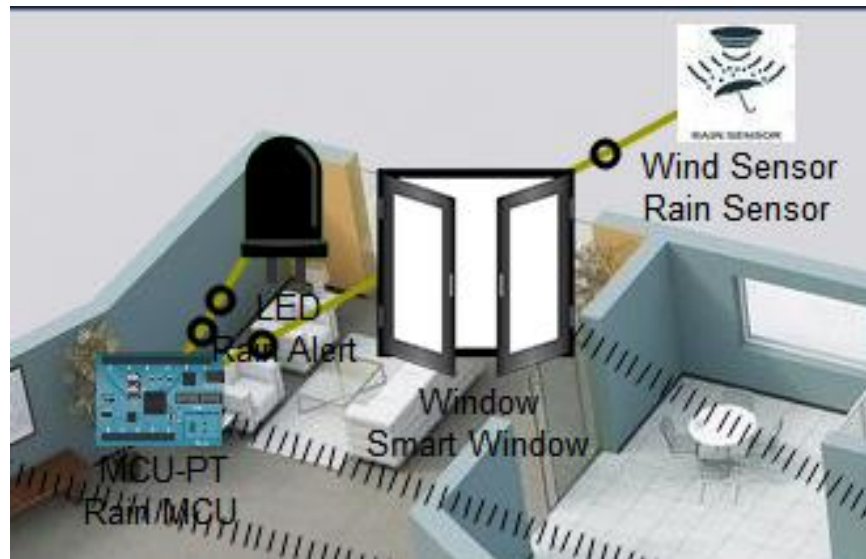


Рисунок 3.14 – Схема розумного вікна

Переходячи до списку девайсів які підключені до мережі на вкладці «Home» та при відкритому вікні, в списку пристроїв, можна буде побачити за зеленим індикатором, що налаштування вказані вірно (рис.3.15).

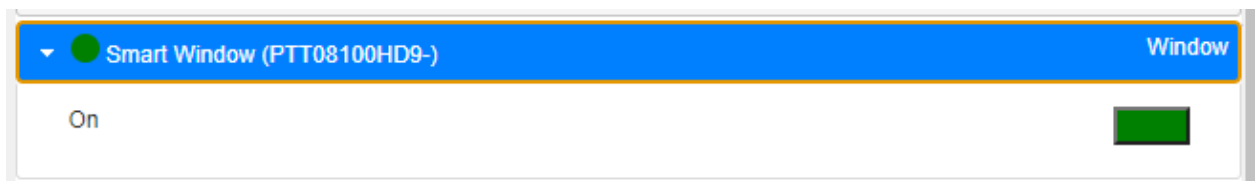


Рисунок 3.15 – Стан системи розумних вікон

Наступним етапом розробки є налаштування системи сповіщення. Вона також використовується для надсилання сповіщення про пожежу, задимленості чи незаконного проникнення в приміщення. Також система налаштовується в умовах роботи пристроїв (рис.3.16-17).

Edit	Remove	Yes	fire_and_siren	IoT2 Fire Detected is true	Set Siren On to true
Edit	Remove	Yes	trip_sensor	Trip Sensor On is true	Set Siren On to true
Edit	Remove	Yes	smoke_detector	Smoke Warner Alarm is true	Set Siren On to true

Рисунок 3.16 – Налаштування системи сповіщення

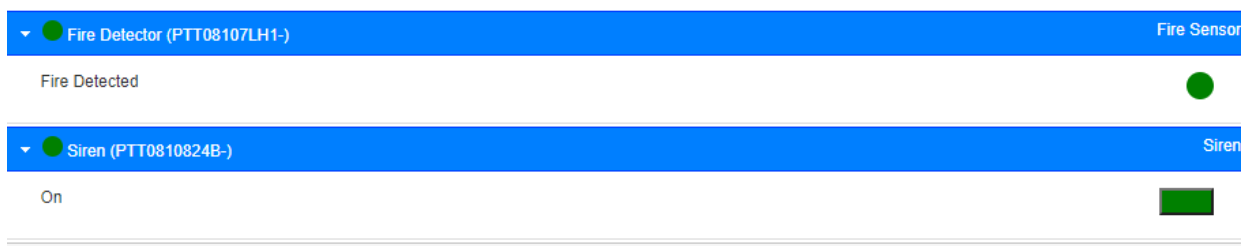


Рисунок 3.17 – Стан системи сповіщення

Відповідно до того яка з ситуацій спрацює, та і буде відображена на вкладці Home напроти цього пристрою. Зображення схеми взаємодії представлено на рисунку 3.18.

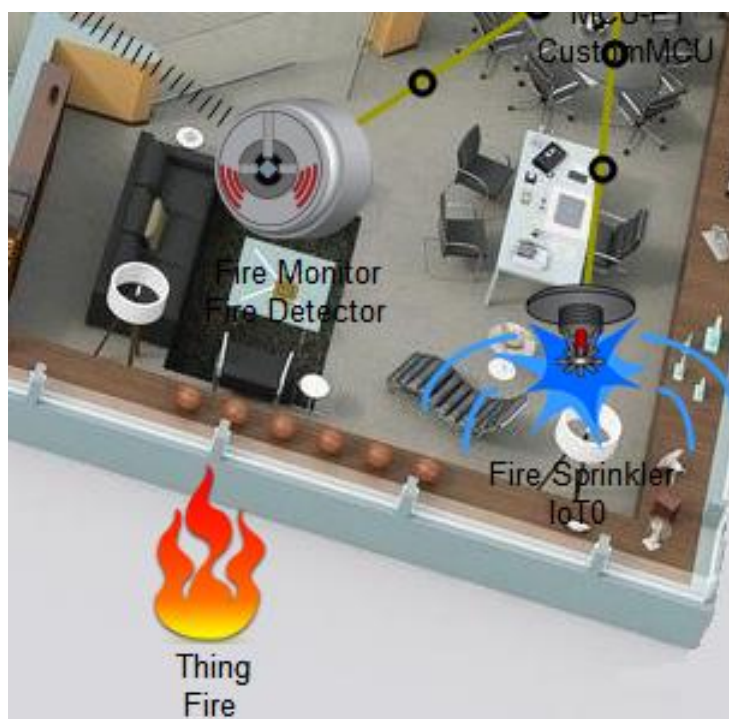


Рисунок 3.18 – Схема системи сповіщення

При умові задимленості, крім сирени будуть автоматично відчинені ворота. Налаштування для цього відповідають а рисунках 3.19-21.

Edit	Remove	Yes	garage_open_smoke	Smoke Warner Level > 1	Set Smart Garage On to true
Edit	Remove	Yes	garage_close_smoke	Smoke Warner Level <= 1	Set Smart Garage On to false

Рисунок 3.19 – Налаштування системи воріт

▼ ● Siren (PTT0810824B-)	Siren
On	
▶ ● Trip Sensor (PTT0810HWJB-)	Trip Sensor
▼ ● Smoke Warner (PTT0810M88Y-)	Smoke Detector
Alarm	
Level	5.4563
▶ ● Motion For Fan (PTT0810781U-)	Motion Detector
▶ ● Street Lamp (PTT08100SVC-)	Street Lamp
▼ ● Smart Garage (PTT0810SV7J-)	Garage Door
On	

Рисунок 3.20 – Стан системи воріт

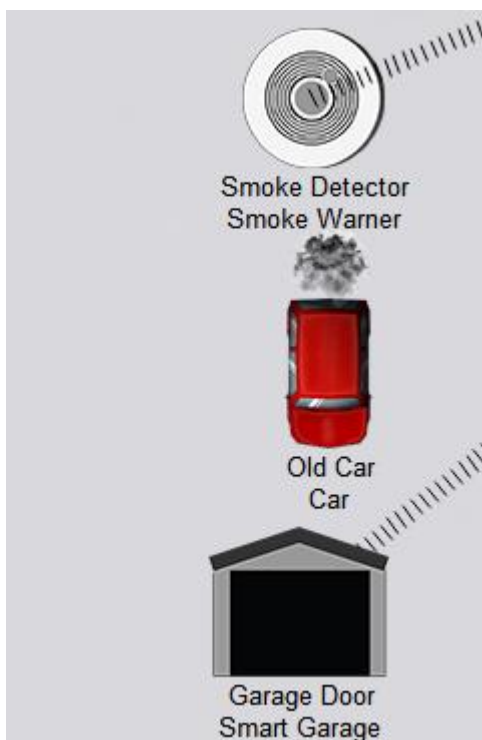


Рисунок 3.21 – Схема системи воріт

Вуличне освітлення працює автоматично на основі вмонтованого датчика освітленості. На вкладці Home користувач може переглядати, роботу датчиків (рис.3.22-23).

Датчик світла (L):

– Реагує на рівень властивості «СВІТЛО» Середовища. Очікуване значення становить від 0 до 1000. Коли стає досить темно ($\text{СВІТЛО} < 330$), загоряється світло лампи. Світло стає яскравішим, чим темніше воно стає, досягаючи повної яскравості, коли $\text{LIGHT} \leq 0$.

– Показує свій стан за допомогою піктограми L із зеленим для увімкнення та червоним для вимкнення.

Датчик руху (M):

– Реагує на наявність поблизу пристроїв у робочому просторі.

– Показує свій стан за допомогою значка M із зеленим для увімкнення та червоним для вимкнення. Стан «Ввімкнено» активний, коли в безпосередній близькості від ліхтаря знаходяться інші пристрої (об'єкти).

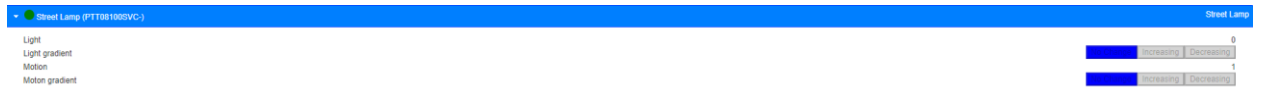


Рисунок 3.22 – Стан системи освітлення



Рисунок 3.23 – Схема системи освітлення

Окрім цього було налаштовано датчик руху налаштовано на вмикання кавомашини та вентиляторів. Детальна інформація за налаштування та схеми кожного із об'єктів схеми представлена на рисунку 3.24-25.

Edit	Remove	Yes	coffee_on	Motion For Fan On is true	Set Coffee Maker On to true
Edit	Remove	Yes	motion_fan	Motion For Fan On is true	Set Smart_Fan Status to High
Add					

Рисунок 3.24 – Налаштування системи кавомашини та вентилятора



Рисунок 3.25 – Схема системи кавомашини та вентилятора

При бажанні, користувач може додати будь які пристрої та запрограмувати на датчик руху, будь-то системи безпеки, освітлення та будь яких інших пристроїв. Все залежить від поставлених цілей та який результат хоче потримати користувач.

3.2 Тестування за різних умов розробленої мережі та модулів

Для успішної роботи мережі, необхідно провести програмування мікрокомп'ютерів, датчиків та контролерів. Додаткова інформація по коду представлена у Додатку А.

Першим етапом є тестування RFID зчитувача. Він був запрограмовано на розпізнання RFID карток за різними ID значеннями.

Враховуючи всі налаштування та вказані умови роботи RFID зчитувача, він загоряється червоним при використанні пропуску з не активним ID. Індикатор відповідно до цього також стає червоним.

Відповідно до цього двері блокуються, що й представлено на рисунку 3.26.



Рисунок 3.26 – Блокування дверей

При використанні карти з відповідним ID RFID до вказаних налаштувань зчитувач загоряється зеленим. Системний індикатор аналогічно до цього також стає зеленим. Якщо ці умови були виконані, то двері будуть розблоковані, та відчинені (рис.3.27-28).

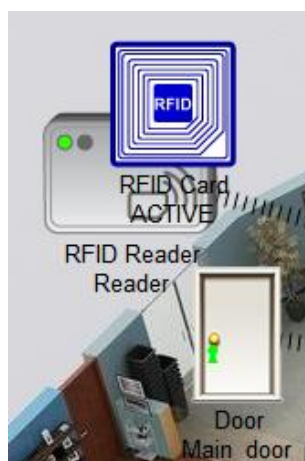


Рисунок 3.27 – Розблоковані двері



Рисунок 3.28 – Відчинення дверей

При налаштуванні RFID можна вказати різні значення. Адже для кожної картки можна встановлювати унікальний ідентифікаційний ключ в атрибутах самої картки. Налаштування карток із RFID Card FAKE та RFID Card ACTIVE представлені на рисунку 3.29 та 3.30.

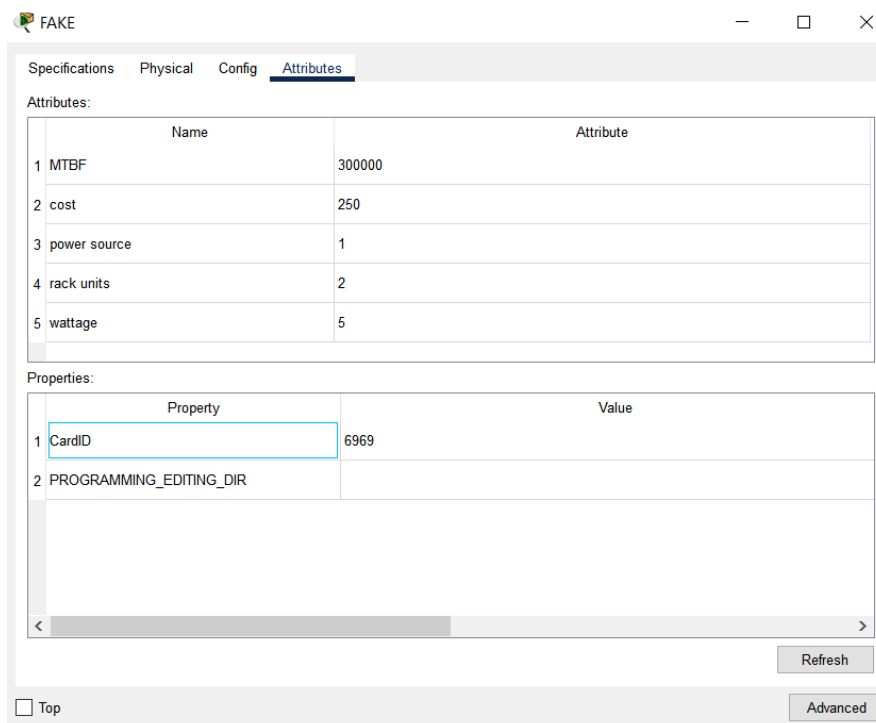


Рисунок 3.29 – RFID Card FAKE

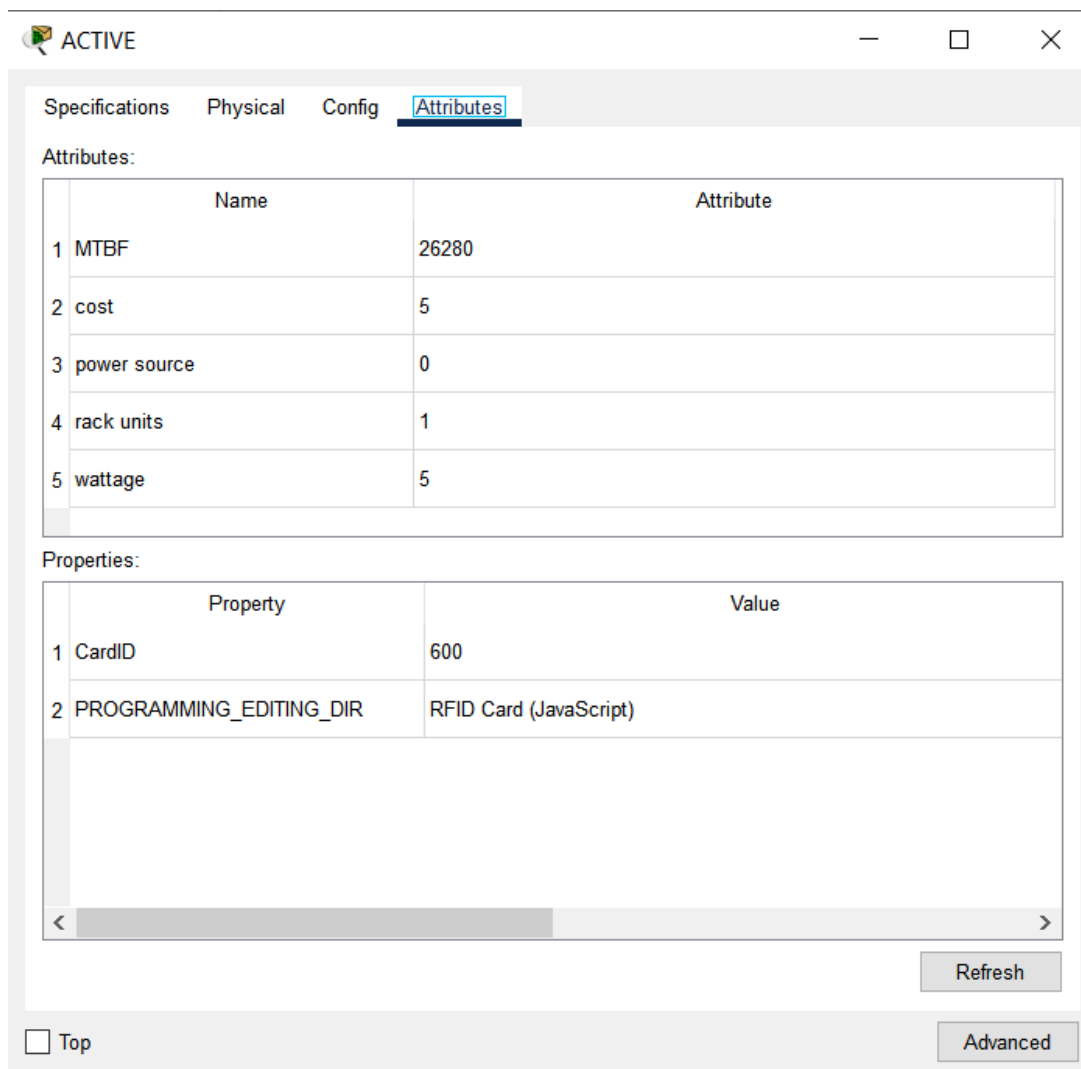


Рисунок 3.30 – RFID Card ACTIVE

Наступним етапом буде тестування Rain MCU. Розумне вікно на основі світла та дощу має головний функціонал по відстеженню погодних умов та вірного реагування та це. Розумне вікно, яке відкриватиметься завдяки запрограмованому мікрокомп'ютеру Rain MCU.

Вікно буде відчинене коли настане ранок і немає дощу який можна відслідкувати за допомогою датчика дощу. Якщо ніч, вікно буде закрито, щоб уникнути комарів, навіть якщо йде дощ.

LED індикатор світиться зеленим кольором, коли на вулиці йде дощ. Відповідно до цього система зачинає вікно (рис.3.31).

Якщо LED індикатор вимкнений, то це може прирівнюватися тому, що відсутні опади (рис.3.32).

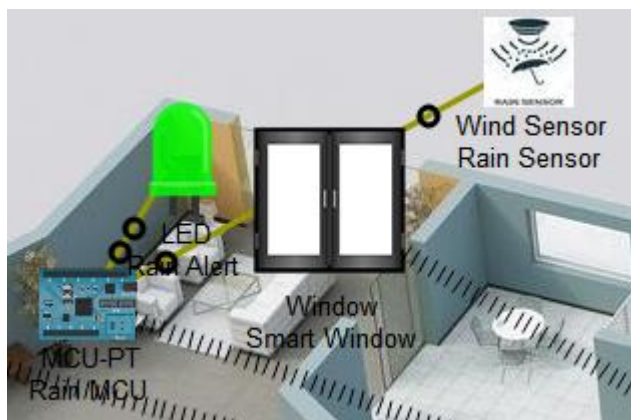


Рисунок 3.31 – Вікно закрите, опади

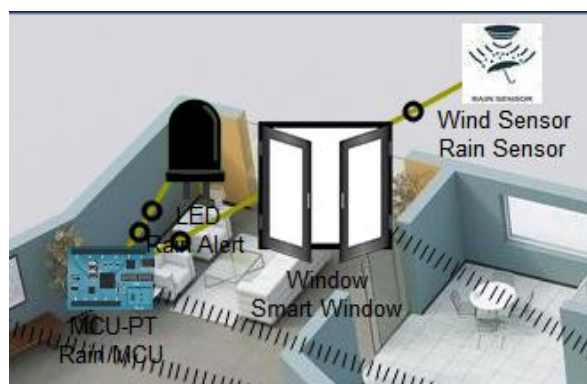


Рисунок 3.32 – Вікно відкрите, опади відсутні

Наступним модулем для тестування є система пожежної сигналізації. Мікрокомп'ютер «CustomMCU» запрограмований на автоматичну роботу пожежних сплінкерів (рис.3.33). При умові, що щось загориться, пожежний сповіщувач подасть сигнал-сирену, щоб попередити всіх оточуючих у будівлі, чи де він встановлений. Також пожежні сплінкери автоматично починають розпиляти воду з метою усунення поширення вогню.

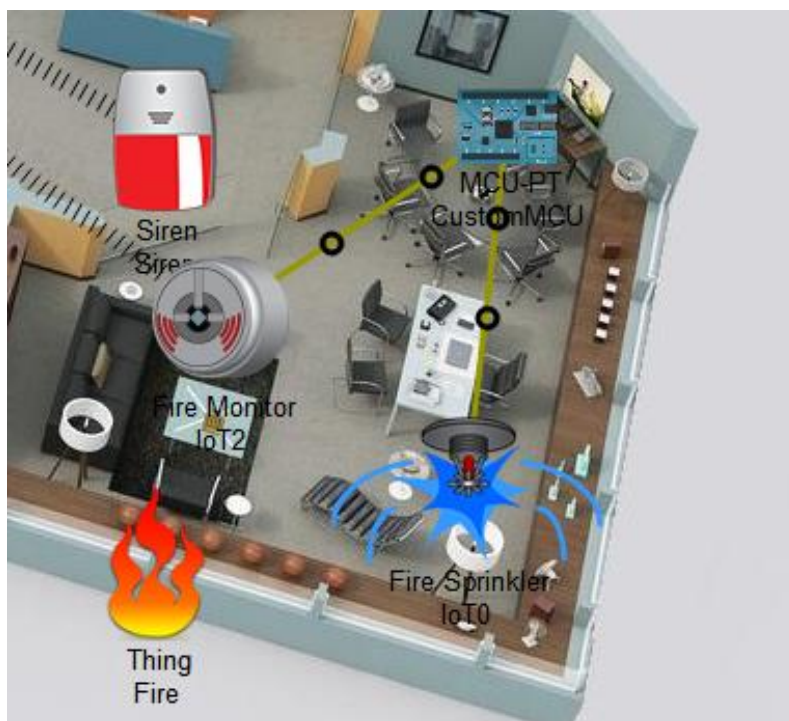


Рисунок 3.33 – Система пожежної сигналізації

Не менш важливим є й система управління опаленням. Пристрій «Thermostat» можна запрограмувати як на опалення так і на охолодження температури (рис.3.34). Також можна встановити автоматичний режим роботи. В залежності від температури в приміщенні, термостат буде вмикатися на обігрів або на охолодження.

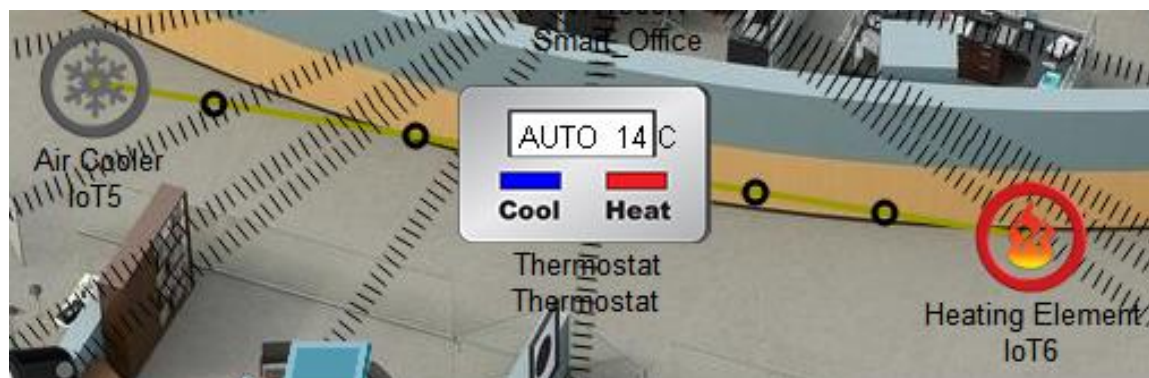


Рисунок 3.34 – Система термостату

В атрибутах пристрою можна встановити температуру для автономної роботи термостату. Приклад налаштування для тестування представлений на рисунку 3.35.

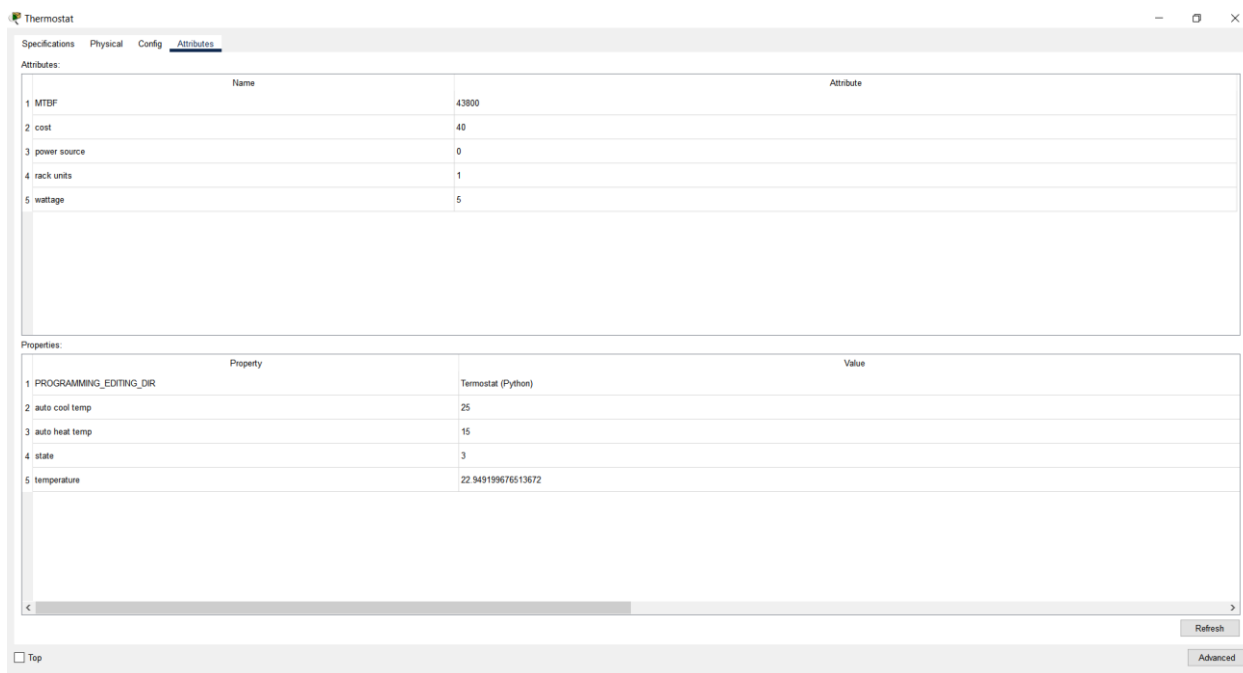


Рисунок 3.35 – Тестові налаштування системи термостату

Наступна частина загальної схеми інформаційно-комунікаційної технології керування є розумний вуличний ліхтар.

Він вмикається вночі за допомогою вмонтованого фотодатчика. За параметрами тут встановлено те, що він має функцію енергозбереження. Дана функція працює в залежності від яскравості сонячного світла.

При базових налаштуваннях ліхтар автоматично вмикається після заходу сонця, та вимикається після світанку.

Приклад роботи вуличного ліхтаря при різних умовах та кількості світла представлено на рисунку 3.36-37. Для цього було використано зміна часу. На кожному із них присутній годинник.



Рисунок 3.36 – Вимкнений вуличний ліхтар



Рисунок 3.37 – Увімкнений вуличний ліхтар

Наступною частиною є розумний гараж. Головний принцип роботи – відкривається сам, коли виявляє дим, означає, що автомобіль заведений. Ворота паркінгу зачиняються самі, коли немає диму, це означає, що або машина знаходиться не в паркінгу, або вона заглушена всередині паркінгу (рис.3.38).



Рисунок 3.38 – Зачинений гараж



Рисунок 3.39 – Відчинений гараж

Також до схеми було додано та налаштовано музичний плеєр. Музику можна відтворювати за допомогою музичного плеєра, як наприклад через Bluetooth, на портативній колонці (рис.3.40). Головна мета даного модулю є тільки розважальна. Також їх зручно використовувати із іншими системами, як телевізор.



Рисунок 3.40 – Портативна мультимедія

Крім того, був доданий датчик руху. Завдяки запрограмованому датчику руху, на створеній схемі це «Motion Detector», коли хтось заходить до їдальні, вентилятор та кавоварка вмикаються автоматично (рис.3.41).

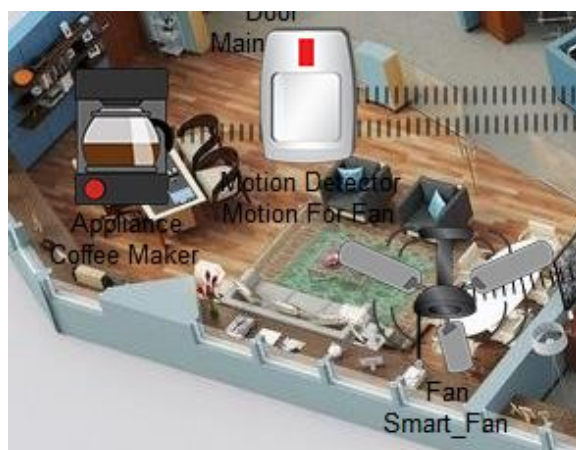


Рисунок 3.41 – Приклад схеми автоматичного вмикання кавоварки та вентилятора

Одним із важливих частин загальної схеми є зарядка батареї від сонячної енергії. Адже на основі сонячної енергії вентилятор і світло у створеній мережі керування працюватимуть автоматично. Акумулятор заряджається лише за умови достатнього освітлення. Також для кращої взаємодії було додано додаткові LED індикатори, що відображають процес зарядки. Приклад зарядки від батареї та активація індикаторів представлена на рисунку 3.42.

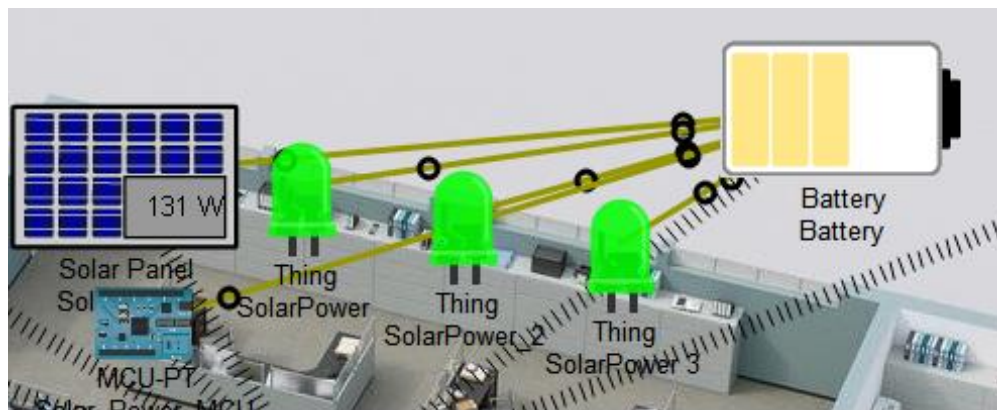


Рисунок 3.42 – Приклад схеми зарядки батареї від сонячної енергії

Останнім компонентом розробленої схеми для кваліфікаційної роботи магістра є система сигналізації. Головне призначення – захист від крадіжки. Загалом, для забезпечення захисту від крадіжки використовується датчик спрацьовування, якщо хтось розіб'є вікно та ввійде всередину, датчик спрацьовує та активується сигнал сирени (рис.3.43).

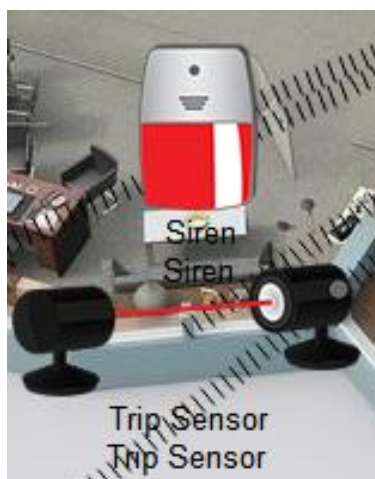


Рисунок 3.43 – Приклад системи сигналізації

Отже, було виконано тестування інформаційно-комунікаційної технології керування мережами. Тестування було пройдено успішно, всі модулі мережі відповідають поставленим задачам та працюють вірно.

ВИСНОВКИ

Під час виконання та розробки випускної роботи магістра, а саме інформаційно-комунікаційної технології керування мережами, було проведено аналіз тематики розробки мереж на основі концепції інтернету речей. Під час роботи був виконаний аналіз переваг та недоліків продуктів-аналогів.

Крім цього, було проаналізовано вибірку технологій для реалізації запропонованої роботи. Також було проаналізовано мови програмування, що допомагають взаємодіяти із обраними програмними продуктами. Для розробки було обрано Cisco та мову програмування Python.

У ході виконання кваліфікаційної роботи магістра була створена та якісно налаштована мережа за допомогою IoT технології, за використання додатку Cisco Packet Tracer. У результаті мережі «розумного офісу» були відтворені з використанням концепцій IoT. При розробці системи використовувався бездротовий зв'язок. Також при розробці ми переконалися, що всіма підключеними пристроями можна керувати та контролювати їх через веб-інтерфейс планшета.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Modern Technology That Can Make Your Booth Attractive [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://vasant.in/modern-technology-that-can-make-booth-attractive/> (дата звернення: 12.09.2022);
2. 7 Technologies to Enhance Interaction at the Booth [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://onestorexpo.com/technologies-to-enhance-interaction-at-the-booth/> (дата звернення: 12.09.2022);
3. Incorporating Advanced Technology in your Stand Design can do Wonders [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.expodisplayservice.com/incorporating-technology-in-your-stand-design-can-do-wonders/> (дата звернення: 12.09.2022);
4. Everything You Need to Know About Smart Home Technology [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.otelco.com/resources/smart-the-home-guide/> (дата звернення: 12.09.2022);
5. Охоронні системи [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://nadzor.ua/blog/signalizacii/ohrannyh-sistem-ajax-starterkit-hikvision-ax-pro-atis-kit-gsmwifi-130t> (дата звернення: 12.09.2022);
6. Ajax Smart Home Alarm System Review [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.avforums.com/reviews/ajax-smart-home-alarm-system-review/> (дата звернення: 12.09.2022);
7. Ajax security system [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://gadget.com/en/smarthethome/24900-ajax-security-system-review/> (дата звернення: 12.09.2022);
8. A Guide to the Hikvision AX Pro Wireless Alarm Range [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://connectec.uk/blog/guide-to-the-hikvision-ax-wireless-alarm-range> (дата звернення: 12.09.2022);

9. Hikvision AX PRO Wireless Alarm System: AX PRO HUB Review | Features & Activation [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://pipl.systems/blogs/blog/hikvision-wireless-alarm-system-ax-pro-hub-the-review-features-activation> (дата звернення: 12.09.2022);

10. An Overview – What is the Advantage of Effective Network Simulator? [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://snmpsimulation.com/tag/snmp-the-simulator/> (дата звернення: 12.09.2022);

11. Pros of Using an IOS and Web Simulator [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <http://snmpsimulation.weebly.com> (дата звернення: 12.09.2022);

12. Free SNMP Agent Simulator [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.topbestalternatives.com/free-snmp-an-agent-simulator/> (дата звернення: 12.09.2022);

13. NetSim [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.g2.com/products/netsim/reviews> (дата звернення: 12.09.2022);

14. Which is the best network simulator for CCNA? [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.edaboard.com/blog/which-is-the-best-network-simulator-for-ccna> (дата звернення: 12.09.2022);

15. 5 Best Network Simulators for Cisco Exams: CCNA, CCNP, CCIE [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.cbttuggets.com/blog/career/career-progression/best-network-simulators-for-cisco-exams-ccna-and-ccie> (дата звернення: 12.09.2022);

16. Cisco Packet Tracer 8.0.0.0212 [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://bestsoft.club/set/cisco-packet-tracer.html> (дата звернення: 12.09.2022);

17. Cisco Packet Tracer using [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.g2.com/products/cisco-packet-tracer/review> (дата звернення: 12.09.2022);

18. Cisco Packet Tracer Reviews & Product Details [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.netacad.com/courses/packet-tracer/faq>
19. J. Blank, K. Deb, Pymoo: Multi-Objective Optimization in Python. IEEE Access. 8, 89497–89509 (2020);
20. 1. A. Meurer et al., SymPy: Symbolic computing in python. PeerJ Computer Science. 2017 (2017), doi:10.7717/peerj-cs.103;
21. Blockly – Introductory Python Programming Blockly Environment [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: <https://blog.ouseful.info/2016/02/18/blockpy-python-blockly-environment/inf> (дата звернення: 12.09.2022);
22. Visual programming in JupyterLab with Blockly [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://blog.jupyter.org/visual-programming-in-jupyterlab-with-blockly> (дата звернення: 12.09.2022);
23. Blockly [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://think.cs.vt.edu/blockpy/> (дата звернення: 12.09.2022);
24. 1. S. Khanagha, S. Ansari, S. Paroutis, L. Oviedo, Mutualism and the dynamics of new platform creation: A study of Cisco and fog computing. Strategic Management Journal. 43, 476–506 (2022).
25. 1. R. Parthasarathy, S. S. Loong, P. Ayyappan, Z. A. Hamid, A. S. Kumar, Implementation of Site-To-Site IPSEC The Virtual Private Network For Enterprise Network Design Using Cisco Packet Tracer Simulation Tools. International Journal of Mechanical Engineering. 7, 1293–1305 (2022).
26. 1. E. Pincemin et al., The Cisco-Acacia Communications. Neophotonics, 6–8 (2022).

ДОДАТОК А. ЛІСТИНГ КОДУ ДЛЯ МІКРОКОНТРОЛЕРА

main.py

RFID зчитувач

```

from options import Options
from time import *
from physical import *
from gpio import *
from ioeclient import IoEClient

DELAY_TIME = 1000    # var DELAY_TIME
current_time = 0     # var current_time

X_READ_DISTANCE = 50    # var X_READ_DISTANCE
Y_READ_DISTANCE = 50    # var Y_READ_DISTANCE
cardID = 0            # var cardID
lastCardID = 0        # var lastCardID
state = 2 # waiting    # var state

def setup ():

    # Registration Server Setup

    IoEClient.setup({
        "type": "RFID Reader",
        "states": [{
            "name": "Card ID",
            "type": "number",
            "unit": "",
            "controllable": False
        }],
        {
            "name": "Status",
            "type": "options",
            "options": {
                "0": "Valid",
                "1": "Invalid",
                "2": "Waiting"
            }
        }
    ]

```

```

        "controllable": True
    }}
})

```

```
IoEClient.onInputReceive( lambda rinput: processData(rinput, True) )
```

```

def loop ():
    global cardID, lastCardID, state
    devices = devicesAt(getCenterX(), getCenterY(), X_READ_DISTANCE,
Y_READ_DISTANCE)    # var devices
    found = False    # var found
    for i in xrange(0, len(devices)) :
        if devices[i] is getName():
            continue

        cardID = getDeviceProperty(devices[i], 'CardID')
        found = True
        break

```

```

if not found:
    cardID = lastCardID = 0
    setState(2)
else:
    if lastCardID != cardID:
        lastCardID = cardID
        sendReport()

```

```
delay(DELAY_TIME)
```

```

def setState (newState):
    global state
    if state != newState:
        state = newState
        analogWrite(A1, state)
        sendReport()

```

```

def sendReport ():
    report = str( int(cardID)) + "," + str(state)    # var report
    IoEClient.reportStates(report)

```

```

def processData (data, bIsRemote):
    if len(data) <= 0:
        return
    data = data.split(",")
    setState(int(data[1]))

if __name__ == "__main__":
    setup()
    while True:
        loop()
        idle()

```

main.py

Мікрокомп'ютер Rain MCU

```

from time import *
from gpio import *

def setup ():
    pinMode(0, INPUT)
    pinMode(1, OUTPUT)

def loop ():
    digitalWrite(1, digitalRead(0))
    delay(1000)

if __name__ == "__main__":
    setup()
    while True:
        loop()
        idle()

```

main.py

Мікрокомп'ютер Custom MCU

```

from gpio import *
from time import *

```

```

def handleSensorData():
    value = digitalRead(0)
    if value == 0:
        customWrite(1, '0')
    else:
        customWrite(1, '1')

def main():
    add_event_detect(0, handleSensorData)

    while True:
        delay(1000)

if __name__ == "__main__":
    main()

```

main.py

Терморегулятор Thermostat

```

from options import Options
from time import *
from physical import *
from gpio import *
from environment import Environment
from ioeclient import IoEClient

autoCoolTemp = 25
autoHeatTemp = 15

METRIC = ""

HEATER_PIN = 1
COOLER_PIN = 2
state = 0 # 0 off, 1 cooling, 2 heating, 3 auto
STATE_NAMES = ["OFF", "COOL", "HEAT", "AUTO"]
temperature = 0
unit = ""

def setup ():
    global autoCoolTemp , autoHeatTemp , temperature , state

```

```

IoEClient.setup({
  "type": "Thermostat",
  "states": [{
    "name": "Status",
    "type": "options",
    "options": {
      0: "Off",
      1: "Cooling",
      2: "Heating",
      3: "Auto"
    },
    "controllable": True
  }, {
    "name": "Temperature",
    "type": "number",
    "unit": "&deg;C",
    "imperialUnit": "&deg;F",
    "toImperialConversion": "x*1.8+32",
    "toMetricConversion": "(x-32)/1.8",
    "decimalDigits": 1
  }, {
    "name": "Auto Cool Temperature",
    "type": "number",
    "unit": "&deg;C",
    "imperialUnit": "&deg;F",
    "toImperialConversion": "x*1.8+32",
    "toMetricConversion": "(x-32)/1.8",
    "decimalDigits": 1,
    "controllable": True,
    "minValue": 10,
    "maxValue": 100
  }, {
    "name": "Auto Heat Temperature",
    "type": "number",
    "unit": "&deg;C",
    "imperialUnit": "&deg;F",
    "toImperialConversion": "x*1.8+32",
    "toMetricConversion": "(x-32)/1.8",
    "decimalDigits": 1,
    "controllable": True,
    "minValue": -100,
    "maxValue": 20
  }
}

```

```

    }}
  })

  IoEClient.onInputReceive(on_input_receive)

  add_event_detect(0, on_event_detect_0)

  temperature = Environment.get("Ambient Temperature")

  measurementSystemChangeEvent()

  state = restoreProperty("state", 0)
  setState(state)

  autoCoolTemp = restoreProperty("auto cool temp", 25)
  autoHeatTemp = restoreProperty("auto heat temp", 15)

  Environment.removeCumulativeContribution("Ambient Temperature")

  updateEnvironment()
  sendReport()
  on_event_detect_0()

def on_event_detect_0 () :
  processData(customRead(0), False)

def on_input_receive(input) :
  processData(input, True)

def restoreProperty (propertyName, defaultValue):
  value = getDeviceProperty(getName(), propertyName)
  if not (value == "" or value == None) :
    if isinstance(defaultValue, (int, float)) :
      value = int(value)

    setDeviceProperty(getName(), propertyName, value)
  return value

return defaultValue

def loop ():

```

```

global temperature
temperature = Environment.get("Ambient Temperature")
updateEnvironment()
measurementSystemChangeEvent()
sendReport()
delay(1000)

```

```

def measurementSystemChangeEvent ():
    global METRIC, unit
    METRIC = Options.isUsingMetric()
    unit = "C" if METRIC else "F"
    setCustomText(27, 15, 200, 100,
        STATE_NAMES[state] + " " +
        convertTemperature(temperature) + " " + unit)

```

```

def mouseEvent (pressed, x, y, firstPress):
    global state
    # if mouse pressed, then flip it
    if firstPress:
        state+=1
        if state >= 4:
            state = 0

    setState(state)

```

```

def processData (data, bIsRemote):
    global autoCoolTemp, autoHeatTemp
    if data:
        # print data
        data = data.split(",")
        # print data[0],data[1],data[2],data[3]
        setState(int(data[0]))
        autoCoolTemp = float(data[2])
        autoHeatTemp = float(data[3])
        if autoCoolTemp < autoHeatTemp:
            autoCoolTemp,autoHeatTemp = autoHeatTemp,autoCoolTemp

```

```

def setState (newState):

```

```

global state
state = newState

analogWrite(A1, state) # change graphic
setCustomText(27, 15, 200, 100,
  STATE_NAMES[state] + " " +
  convertTemperature(temperature) + " " + unit)

sendReport()

```

```

def sendReport ():
    report = str(state) + "," + str(temperature) + "," + str(autoCoolTemp) + "," +
str(autoHeatTemp)
    customWrite(0, report)
    IoEClient.reportStates(report)
    setDeviceProperty(getName(), "state", state)
#    print "SR: ", autoCoolTemp, autoHeatTemp, temperature
    setDeviceProperty(getName(), "temperature", temperature)
    setDeviceProperty(getName(), "auto cool temp", autoCoolTemp)
    setDeviceProperty(getName(), "auto heat temp", autoHeatTemp)

```

```

def updateEnvironment ():
    global autoMode
    autoCoolTemp = float(getDeviceProperty(getName(), "auto cool temp"))
    autoHeatTemp = float(getDeviceProperty(getName(), "auto heat temp"))

```

```

digitalWrite(HEATER_PIN, LOW)
digitalWrite(COOLER_PIN, LOW)

```

```

if state == 0 :
    digitalWrite(HEATER_PIN, LOW)
    digitalWrite(COOLER_PIN, LOW)

```

```

elif state == 1 :
    digitalWrite(COOLER_PIN, HIGH)

```

```

elif state == 2 :
    digitalWrite(HEATER_PIN, HIGH)

```

```

elif state == 3 :
    if temperature >= autoCoolTemp:

```



```

        digitalWrite(COOLER_PIN, HIGH)

    elif temperature < autoHeatTemp:
        digitalWrite(HEATER_PIN, HIGH)

def convertTemperature (value):
    if METRIC:
        return "%.0f"%(value,)
    else:
        return "%.0f"%(value * 1.8 + 32,)

if __name__ == "__main__":
    setup()
    while True:
        loop()

```

main.py

Датчик руху Motion Detector

```

from gpio import *
from time import *
from ioecclient import *
from physical import *
from bluetooth import *
import math

DEACTIVATE_TIMER = 5; # in seconds
state = 0
current_time = 0

def main():
    setup()
    while True:
        loop()
def setup():
    IoEClient.setup({
        "type": "Motion Detector",
        "states": [{
            "name": "On",
            "type": "bool",

```

```

        "controllable": False
    }]
})
global state
state = restoreProperty("state", 0)
setState(state)

```

```

def restoreProperty(propertyName, defaultValue):
    value = getDeviceProperty(getName(), propertyName)
    if not (value is "" or value is None):
        if type(defaultValue) is int :
            value = int(value)

        setDeviceProperty(getName(), propertyName, value)
    return value
return defaultValue

```

```

def mouseEvent(pressed, x, y, firstPress):
    setState(1)

```

```

def loop():
    global state
    global current_time
    if state == 1 :
        current_time = current_time - 1
        if current_time <= 0 :
            setState(0)

```

```

    sleep(1)

```

```

def setState(newState):
    global state
    global DEACTIVATE_TIMER
    global current_time
    state = newState

    if state is 0 :
        digitalWrite(1, LOW)
    else:
        digitalWrite(1, HIGH)
        current_time = DEACTIVATE_TIMER

```

```

IoEClient.reportStates(state)
setDeviceProperty(getName(), "state", state)

if __name__ == "__main__":
    main()

```

main.py

Датчик проникновения Trip Sensor

```

from gpio import *
from time import *
from ioecient import *
from physical import *
import math

state = 0
def main():
    setup()
    while True:
        delay(1000)
def setup():
    IoEClient.setup({
        "type": "Trip Sensor",
        "states": [{
            "name": "On",
            "type": "bool",
            "controllable": False
        }]
    })

state = restoreProperty("state", 0)
setState(state)

def restoreProperty(propertyName, defaultValue):
    value = getDeviceProperty(getName(), propertyName)
    if not (value is "" or value is None):
        if type(defaultValue) is int :
            value = int(value)

    setDeviceProperty(getName(), propertyName, value)
    return value

```

```
return defaultValue

def mouseEvent(pressed, x, y, firstPress):
    if x > 42 and x < 109 and y > 12 and y < 25:
        setState(1)
    else:
        setState(0)

def setState(newState):
    global state
    global DEACTIVATE_TIMER
    global current_time
    state = newState

    if state is 0 :
        digitalWrite(1, LOW)
    else:
        digitalWrite(1, HIGH)
    IoEClient.reportStates(state)
    setDeviceProperty(getName(), "state", state )

if __name__ == "__main__":
    main()
```