

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

Секція інформаційно-комунікаційних технологій

ВИПУСКНА РОБОТА

на тему:

**«Програмно-апаратне забезпечення для керування побутовими
пристроями ІоТ»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Великодний Д.В.

Студент гр. ІН-71

Чечіль Т.Ю.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедри Довбиш А.С.

“ _____ ” _____ 2021 г.

**ЗАВДАННЯ
до випускної роботи**

Студента четвертого курсу, групи ІН-71 спеціальності “Комп'ютерні науки”
денної форми навчання Чечіля Тимофія Юрійовича _____ .

Тема: «Програмно-апаратне забезпечення для керування побутовими пристроями IoT»

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) огляд та аналіз існуючих аналогів «розумного будинку»; 2) постановка завдання та формулювання завдань дослідження; 3) практична реалізація технології IoT.

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Великодний Д.В.

Завдання прийняв до виконання _____ Чечіль Т.Ю.

РЕФЕРАТ

Записка: 60 стр., 34 рис., 4 табл., 1 додаток, 22 джерела.

Об'єкт дослідження — використання та розробка «розумного будинку».

Мета роботи — розробка програмно-апаратного забезпечення для керування побутовими пристроями IoT.

Методи дослідження — використання інструментів Cisco Packet Tracer.

Результати — результатом розробки програмно-апаратного забезпечення для керування побутовими пристроями IoT. Було також представлено веб-додаток із представленням роботи спроектованої системи «розумного будинку».

CISCO, РОЗУМНИЙ БУДИНОК, LAN, WIRELESS, МОДУЛЬ,
CONFIGURATION, МІКРОКОНТРОЛЕРИ, ІНТЕРНЕТ-МЕРЕЖА.

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Дослідження актуальності проблеми	6
1.2 Аналіз аналогів	9
1.3 Постановка задачі проекту	13
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІОТ	14
2.1. Можливості «Cisco Packet Tracer 7» для створення ІоТ	14
2.2. Вибір засобів реалізації	15
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ІОТ	19
3.1 Розробка	19
3.2 Використання системи	30
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	38
ДОДАТОК А. КОД РЕАЛІЗАЦІЇ	41

ВСТУП

На сьогоднішній день мережа Інтернет, яка використовується більшістю людей кожного дня, є одним з найважливіших винаходів за всю історію. Звичайно, кожного дня розвиваються системи для автоматизації та управління електронними пристроями.

Дана тематика ставиться все популярніша як на підприємствах, так і в побуті. Багато людей хотіли би мати можливість виконати онлайн-підключення своїх домашніх пристроїв для керування ними дистанційно. Виконувати управління можна наприклад з персонального мобільного пристрою.

На сьогоднішній день йдуть обговорення та прототипування найрізноманітніших проектів для реалізації «розумних будинків» чи інших варіантів об'єднання розрізнених електронних компонентів в єдину систему. Це все становить досить складною темою, якщо враховувати всі складності. При оснащенні свого житла сучасними розумними пристроями може виникнути проблема узгодженої роботи пристроїв різних виробників.

Для людини, яка не є фахівцем в області електроніки, програмування чи проектуванні мікрокомп'ютерів, поєднання декількох різних екосистем в одну, свою власну, ставиться надзвичайно складною задачею. У результаті чого, ідея створення програмно-апаратного комплексу для використання та управління домашніми пристроями навіть від різних виробників є досить актуальною на 2021 рік.

Отже, мета дипломної роботи – розробка програмно-апаратного забезпечення для керування побутовими пристроями IoT.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

Домашня автоматизація – можна адаптувати як автоматизація можливості контролювати предмети й техніку навколо будинку. До даних предметів можна віднести й вікна, й підгодівлю домашніх тварин, й автоматичний полив рослин. Все це можна виконати простим натисканням кнопки або, в залежності від реалізації, голосовою командою[1-2].

Деякі можливості, наприклад налаштування лампи, світильників чи іншого освітлення, для вмикання та вимикання за вашим бажанням, прості і відносно недорогий для замовлення функціонал. Інші, такі як вдосконалені камери спостереження, можуть вимагати більш вагомих вкладень часу та грошей.

Існує багато категорій розумних будинків, тому ви можете контролювати все, від освітлення та температури до замків та пристроїв домашньої безпеки. Вони також роблять фантастичні подарунки, будь то покупки на свята чи придбання новорічного подарунка.

Розглянемо декілька прикладів напрямлень реалізації «розумного будинку»[3]:

- Розумні колонки та дисплеї.

Можуть робити набагато більше, ніж відтворювати музику, хоча вони це чудово роблять. Вони можуть шукати в Інтернеті, розповісти вам про погоду, дати вам новини розбір польотів, робота в якості особистого помічника, і може виступати в якості центрального вузла управління для всієї вашої Smart-Home system.

- Розумні рішення для освітлення.

Коли мова заходить про автоматизацію світла, є два варіанти. Перший - це встановлення розумних вимикачів світла . Розумні вимикачі світла не тільки можна керувати додатком, але багато хто надає вам різні варіанти освітлення, наприклад,

затемнення. Другий варіант розумного освітлення - це розумні лампочки . Як і розумні розетки, розумні лампочки покращують звичайне освітлення в розумні. Вимкніть свої старі лампочки на розумні, і ви зможете автоматизувати своє освітлення.

- Розумна побутова техніка.

Значно полегшують приготування їжі та прибирання. Ви навіть можете підключити багато з них за допомогою своїх розумних колонок

- Розумні роботи-пилососи.

Ними можна керувати за допомогою смарт-пристроїв, включаючи розумні колонки. Як би не було достатньо мати робот-пилосос, деякі з цих пилососів мають власний штучний інтелект

- Камери та датчики.

Система безпеки, що базується на додатках , дозволить вам підтвердити попередження самостійно, отримавши доступ до кадру на вашому смарт-пристрої з будь-якої камери в приміщенні.

Важко пояснити, як працює система автоматизації дому, оскільки немає єдиного загальноприйнятого протоколу[4]. Технологія залежить від країни, компанії та типу автоматизації. Часто, оскільки ринок домашньої автоматизації настільки прибутковий і стрімкий, компанії затримують нові розробки.

Розглянемо перелік переваг (табл.1.1) та недоліків (табл.1.2) використання розробки «розумного будинку».

Таблиця 1.1 – Переваги «розумного будинку»

№	Напрямок	Опис
1	Зручність	Управління предметами по дому за допомогою однієї кнопки або голосової команди здається мрією[5].
2	Ефективність	створюючи більш комфортні умови для тих періодів часу, коли зазвичай перебувають власники будинків.
3	Заощадження грошей	Менше споживання енергії означає зниження плати за комунальні послуги
4	Персоналізація	Технології розумного будинку адаптуються до потреб своїх власників, звичок та звичок. Вони легко налаштовуються [6].
5	Безпека	Розумні пристрої спостереження можуть дозволити власнику будинків стежити за своїм будинком, коли вони перебувають у відпустці.

Таблиця 1.2 – Недоліки «розумного будинку»

№	Напрямок	Опис
1	Злом	Зловмисні дії можуть загрожувати безпеці будинку. Зловмисники можуть отримати доступ до вашого розкладу входу та виїзду, відключити вашу систему безпеки та грабувати будинок[7].
2	Складна навігація	доведеться бути технічно підкованим, щоб впоратися з налаштуванням та їх використанням.
3	Відсутність взаємодії між пристроями	використовувати кілька різних програм окремо для кожного пристрою

4	Неправильне використання виробниками	технології, можуть таємно збирати інформацію про ваші звички та уподобання
---	--------------------------------------	--

За прогнозами, до 2020 року технологія розумного будинку становитиме мільярд доларів, і вона представляє найдосконаліший сектор Інтернету речей. Апеляція зрозуміла - більше інформації, більше контролю та більше автоматизації вашого будинку дозволяють приймати кращі рішення та оптимізувати використання обмежених ресурсів.

Це корисно для планети і корисно для вашого гаманця. Однак до тих пір, поки зяючі діри в безпеці не будуть принаймні частково усунені, технології розумного будинку все ще потрібно подолати серйозні перешкоди, перш ніж вони досягнуть повсюдного поширення.

1.2 Аналіз аналогів

Розглянемо приклади програм, що реалізують технології «розумного будинку» урахувавши також на багаторічний досвід роботи з подібним обладнанням. Розглянемо такі компанії:

- «Ajax»[8];
- «Fibarо»[9];
- «Orvibo»[10].

Перейдемо до детальнішого перегляду компанії «Ajax». Продукція компанії представлена на рис.1.1.

КАТАЛОГ ПРОДУКЦІЇ АЈАХ

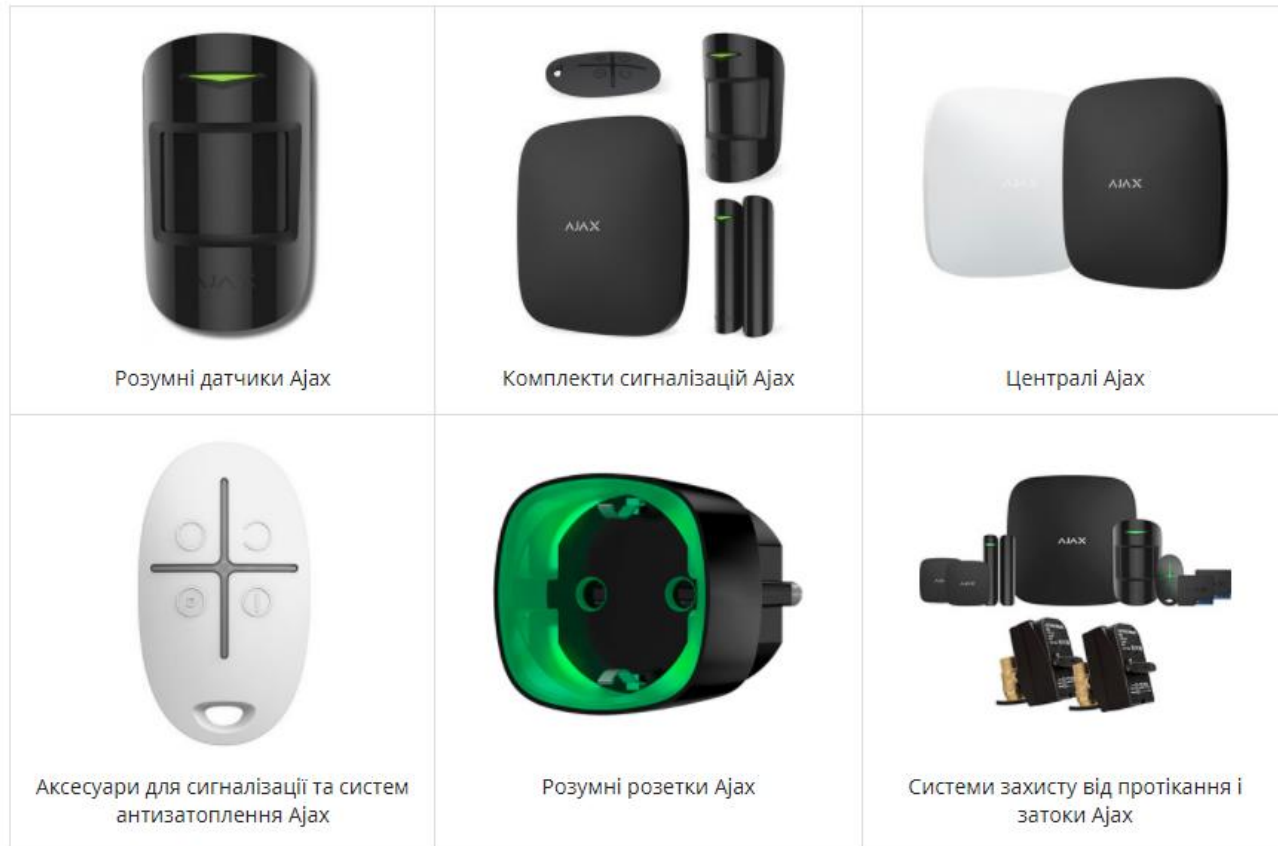


Рисунок 1.1 – Продукція «Ајах»

Дана система автоматизації та розробки «розумного будинку» в повній мірі справляється відразу з декількома важливими завданнями:

1. забезпечення комфорту і зручність в управлінні життєзабезпеченням приміщення;
2. забезпечення безпеки.

Наступний приклад – «розумний будинок» від «Fibaro». Дана компанія відноситься до професійного обладнання із забезпечення автоматизації забезпечення безпеки будинку. Дана ціль досягається за допомогою спектру доступного функціоналу.

Звичайно, представлені системи повинні встановлюватися та налаштовуватися досвідченими фахівцями. Категорії товару даної команди представлені на рис.1.2.

The image shows a screenshot of the FIBARO product page. On the left, there is a sidebar with filters under the heading 'ФИЛЬТРЫ'. The 'КАТЕГОРИЯ:' section lists various smart home categories such as 'Центральные контроллеры', 'Датчики', 'Управление климатом', 'Умная розетка', 'Комплекты для Умного дома', 'Реле', 'Выключатели', 'Реле и диммеры', 'Вызывные панели цветные', 'Распродажа', 'Управление шторами и жалюзи', and 'Дополнительное оборудование для умного дома'. Below this is a price filter section 'ЦЕНА:' with a range from 228 to 28058. The main content area is titled 'FIBARO' and shows 'Подобрано 53' items. It includes sorting options ('Сортировать по: Популярность, Имя, Цена') and display options ('Отображать по: 12, 24, 36'). Three product cards are visible: 1. 'Радиаторный термостат FIBARO Heat Controller для Apple HomeKit — FIB_FCBHT-' with code 12292, price 2957 грн, and a 'КУПИТЬ' button. 2. 'Контроллер умного дома FIBARO Home Center 3 - FIB_HOMEC3' with code 12791, price 18999 грн (discounted from 19499 грн), and a 'ПРЕДЗАКАЗ' button. 3. 'Релейный выключатель со счетчиком электроэнергии FIBARO Single Switch для' with code 12294, price 1971 грн, and a 'КУПИТЬ' button. There are also 'РАСПРОДАЖА' (SALE) banners above the second and third cards, and a 'FLOOD EFFECT' product card partially visible below the first one.

Рисунок 1.2 – Продукція «Fibaro»

Останні приклад – це недорогі комплекти простого в експлуатації обладнання, а саме «Orvibo» (рис.1.3-4). Головне завдання даних комплектів полягає в безпеці будинку. Хоча це також є невідмінною частиною для організації повноцінної системи «розумний будинок».

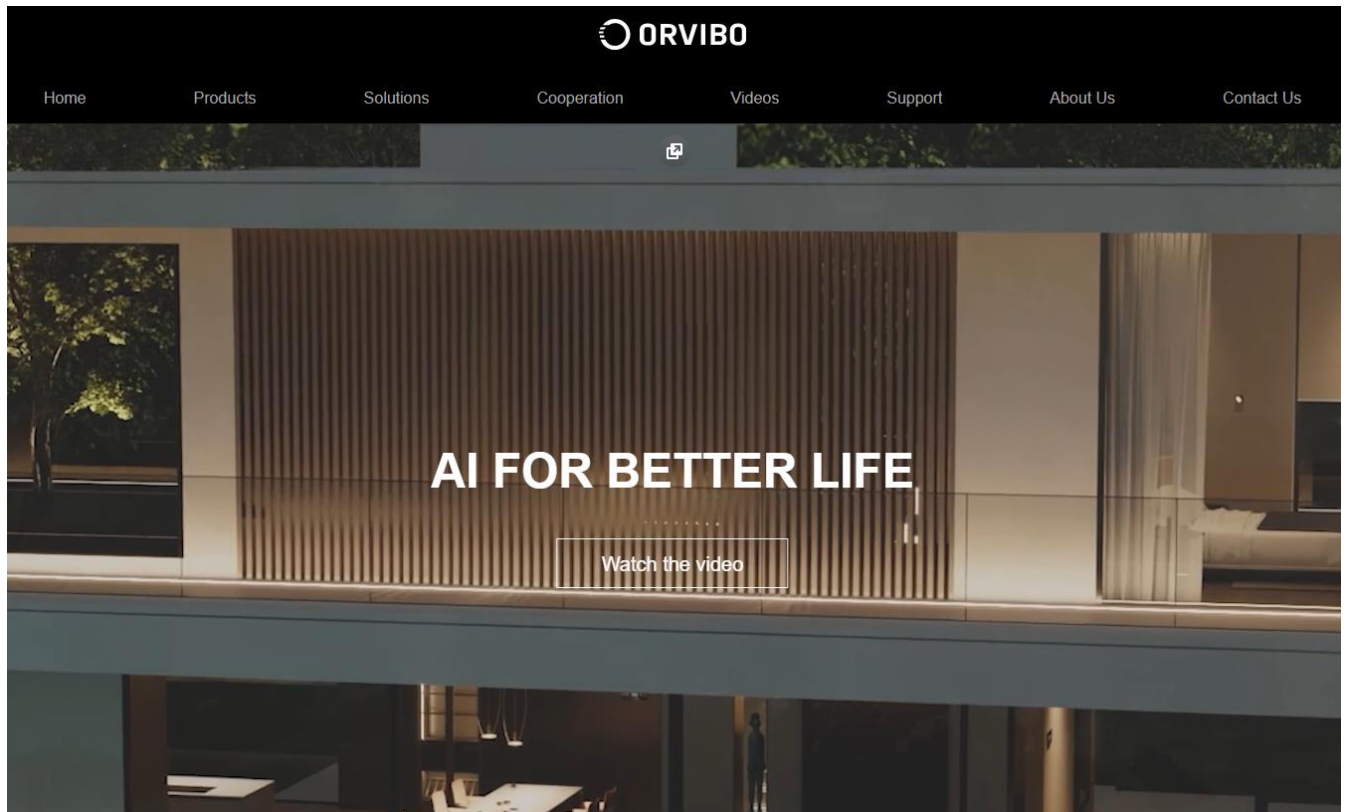


Рисунок 1.3 – Продукція Orvibo

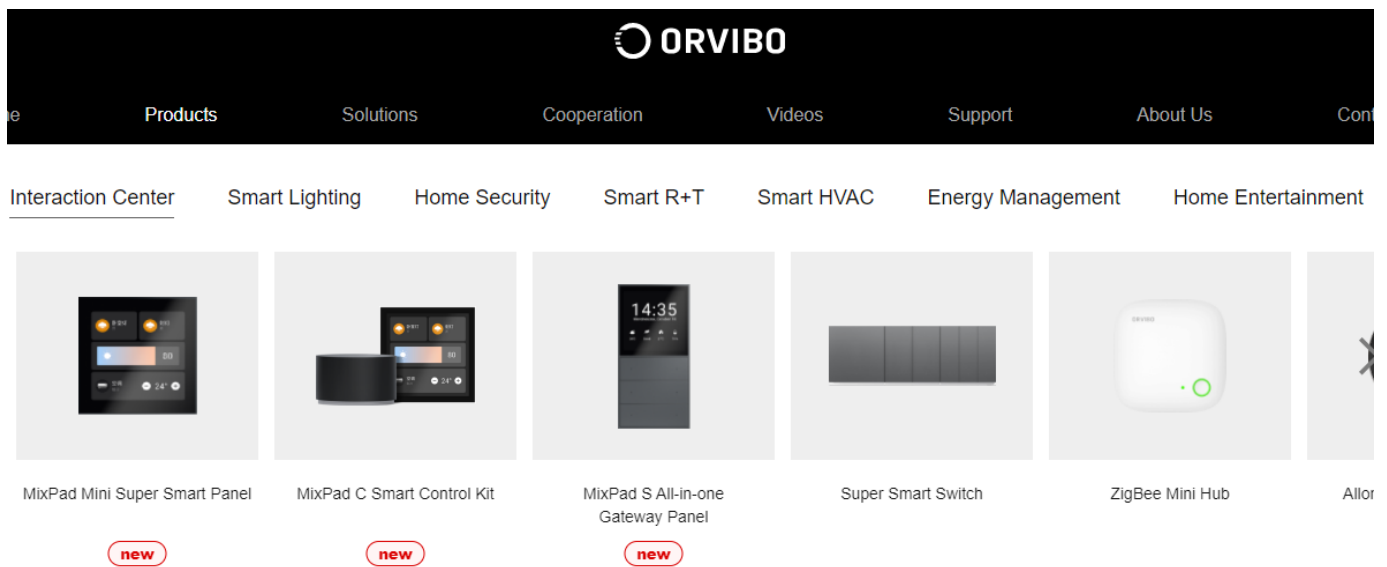


Рисунок 1.4 – Продукція «Orvibo»

1.3 Постановка задачі проекту

Як відомо більшості, головним призначенням та функціоналом представленої системи, системи «розумного будинку», представляє з себе об'єднання та поєднання в єдину еко-систему всіх інженерних систем будинку. Це реалізовується з метою надати єдиний механізм управління інфраструктурою будівлі. Ідеальне представлення даної системи – одна панель управління для всіх підсистем «розумного будинку». Звичайно, розробити єдиний блок для управління можливо, але це потребує значних зусиль. Частіше всього даний блок розділяється на декілька.

Отже, метою бакалаврської роботи є розробка програмно-апаратного забезпечення для керування побутовими пристроями IoT. На основі цього потрібно розробити перелік функціональних блоків та можливостей:

- розробка прикладу управління пристроями та обраною апаратурою;
- налаштування світильників;
- налаштування існуючих систем для забезпечення безпеки;
- налаштування та керування воротами в гаражі із автомобілем;
- управління додатковою технікою в звичайному побуті;
- контроль температури в будівлі та налаштування автополиву;
- автоматичне зачинення дверей;
- взаємодія із додатковими модулями для забезпечення енергії.

Робота присвячена темі налаштування технології IoT. Дана технологія представляє з себе концепцію локальної мережі.

«Розумний будинок» повинен складатися із певних пов'язаних між собою пристроїв. Дані модулі будуть мати власні індивідуальні налаштування. Додатково потрібно виділити, що буде розроблено програмне забезпечення. Саме програмна реалізація дипломної роботи дозволить здійснювати передавати дані та взагалі обмінюватися інформацією окремим модулям між собою та інтернет-мережею.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІОТ

2.1. Можливості «Cisco Packet Tracer 7» для створення ІоТ

«Cisco Packet Tracer 7» - додаток, що був розроблений відомою компанією Cisco. Дана програма була рекомендована для використання при вивченні телекомунікаційних мереж чи додаткових інструментів[11].

Розглянемо перелік основних можливостей «Packet Tracer»:

- GUI інтерфейс. Дана можливість сприяє кращому розумінню принципу організації мереж та роботи пристрою.
- Реалізація декількох мов інтерфейсу.
- Можливість змодельовати логічну топологію.
- Моделювання в режимі реального часу;
- Використання режиму симуляції;
- Якісне зображення та відображеного мережевого обладнання.
- Наявність інструменту «Activity Wizard», який дозволяє створювати професійні шаблони мереж.
- Проектування фізичної топології.

Широкий спектр можливостей обраного інструментарію, що дозволяє мережевим інженерам як конфігурувати, так й налагоджувати мережу. Також даний продукт незамінний в навчальному процесі, оскільки дає наочне відображення роботи мережі, що підвищує освоєння матеріалу учнями[12-13].

«Packet Tracer» здатний моделювати будь-яку кількість пристроїв, а так само використання чимало різних типів поєднань.

Методи моделюються пристрої:

- Комутатори третього рівня: «Router 2620 XM»; «Router 2621 XM»; «Router-PT».
- Комутатори другого рівня: «Switch 2950-24»; «Switch 2950T»; «Switch-PT»; з'єднання типу «міст» Bridge-PT.

- Мережеві концентратори: «Hub-PT»; повторювач «Repeater-PT».
- Кінцеві пристрої: робоча станція «PC-PT»; сервер «Server-PT»; принтер «Printer-PT».
- Бездротові пристрої: точка доступу «AccessPoint-PT».
- Глобальна мережа WAN.

Типи зв'язків для проектування:

- Консольний.
- Мідний кабель без перехрещення або прямий кабель.
- Мідний кабель з перехрещуванням або крос-кабель.
- Телефонна лінія.

Так само доцільно привести ті протоколи, які будь-який користувач може відстежувати: ARP, DHCP, ICMP, CDP, EIGRP;

2.2. Вибір засобів реалізації

«Cisco Packet Tracer» - це комплексний інструмент викладання та навчання мережевих технологій, який пропонує унікальну комбінацію реалістичного досвіду моделювання та візуалізації, оцінки, можливостей створення активності та багатокористувацьких можливостей для співпраці та конкуренції. Інноваційні функції «Packet Tracer» допоможуть студентам і викладачам співпрацювати, вирішувати проблеми та вивчати концепції в захоплюючому та динамічному соціальному середовищі.

- Забезпечує реалістичне навчальне середовище для моделювання та візуалізації, що доповнює обладнання класу, включаючи можливість бачити внутрішні процеси в режимі реального часу, які зазвичай приховані на реальних пристроях

- Забезпечує багатокористувацьку співпрацю в реальному часі та змагання за динамічне навчання

- Дозволяє створювати та локалізувати структуровані навчальні заходи, такі як лабораторії, демонстрації, вікторини, іспити та ігри
- Надає студентам можливість досліджувати концепції, проводити експерименти та перевіряти своє розуміння побудови мережі
- Дозволяє студентам та викладачам проектувати, будувати, налаштовувати та усувати складні мережі за допомогою віртуального обладнання
- Підтримує різноманітні можливості викладання та навчання, такі як лекції, групові та індивідуальні лабораторії, домашні завдання, ігри та змагання
- Підтримує розширення функцій за допомогою зовнішніх додатків, що використовують API, щоб покращити функціональність «Cisco Packet Tracer» у таких сферах, як навчальна програма та оцінка, ігри, доступність та взаємодія з реальним обладнанням.

Крім того, у додатку «Cisco Packet Tracer» можна запрограмувати пристрої, що будуть імітувати поведінку мікроконтролерів. Для програмування використовуються JavaScript [14-16], Python [17-19] або Blockly.

Розглянемо переваги та недоліки обраних мов програмування в табл.2.1-2.

Таблиця 2.1 – JavaScript: переваги та недоліки

№	Переваги	Недоліки
1	JavaScript завжди виконується в клієнтському середовищі.	Об'єктна модель документа працює досить повільно і не має швидкий рендерінг HTML.
2	JavaScript підтримується всіма сучасними популярними браузерами, що допомагає отримувати еквівалентний результат.	Низький захист на стороні клієнта, що може бути використано для зловмисних цілей.

3	Велика кількість прикладів та JavaScript проектів із відкритим кодом.	Помилки при виконанні JavaScript можуть впливати на відображення всього додатку.
4	Великий спектр способів та сфер застосування.	
5	Широко розповсюджений.	

Таблиця 2.1 – Python: переваги та недоліки

№	Переваги	Недоліки
1	бібліотекою та вмістом коду для різних цілей.	Тип роботи, а саме читання рядок за рядком, призводить до повільного виконання проекту.
2	Python може бути розширений до інших мов.	Не такий надійний для використання у мобільних обчислень.
3	Висока продуктивність	Дуже легко спричинити помилки під час написання коду та його виконання.
4	Досить легко вивчити , зрозуміти та кодувати.	Рівні доступу до бази даних Python на даному етапі знаходиться на низькому рівні.
5	підтримує як процедурну, так і об'єктно-орієнтовану парадигми програмування.	

Оскільки мережа та системи ускладнюються, важливо мати нові та вдосконалені методи викладання та вивчення мережевих технологій. Індикатор пакетів Cisco - це інструмент, який може допомогти студентам на місцях навчитися, зрозуміти мережеві технології та вирішити проблеми.

Це чудовий додаток, який може допомогти їм зрозуміти складність інформаційно-комунікаційних технологій. Ця програма моделювання, безсумнівно, допомагає студентам здобути такі здібності, як критичне мислення, прийняття рішень та вирішення проблем.

Хоча це чудова заміна дорогого обладнання, такого як ряд маршрутизаторів та комутаторів, необхідних для створення справжньої мережі.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ІОТ

3.1 Розробка

Інтелектуальна домашня мережа складається з багатьох інфраструктурних пристроїв [20]. Розглянемо приклад реалізації та розробки компоненту «домашній шлюз». Приклад додавання моделі представлено на рис.3.1.

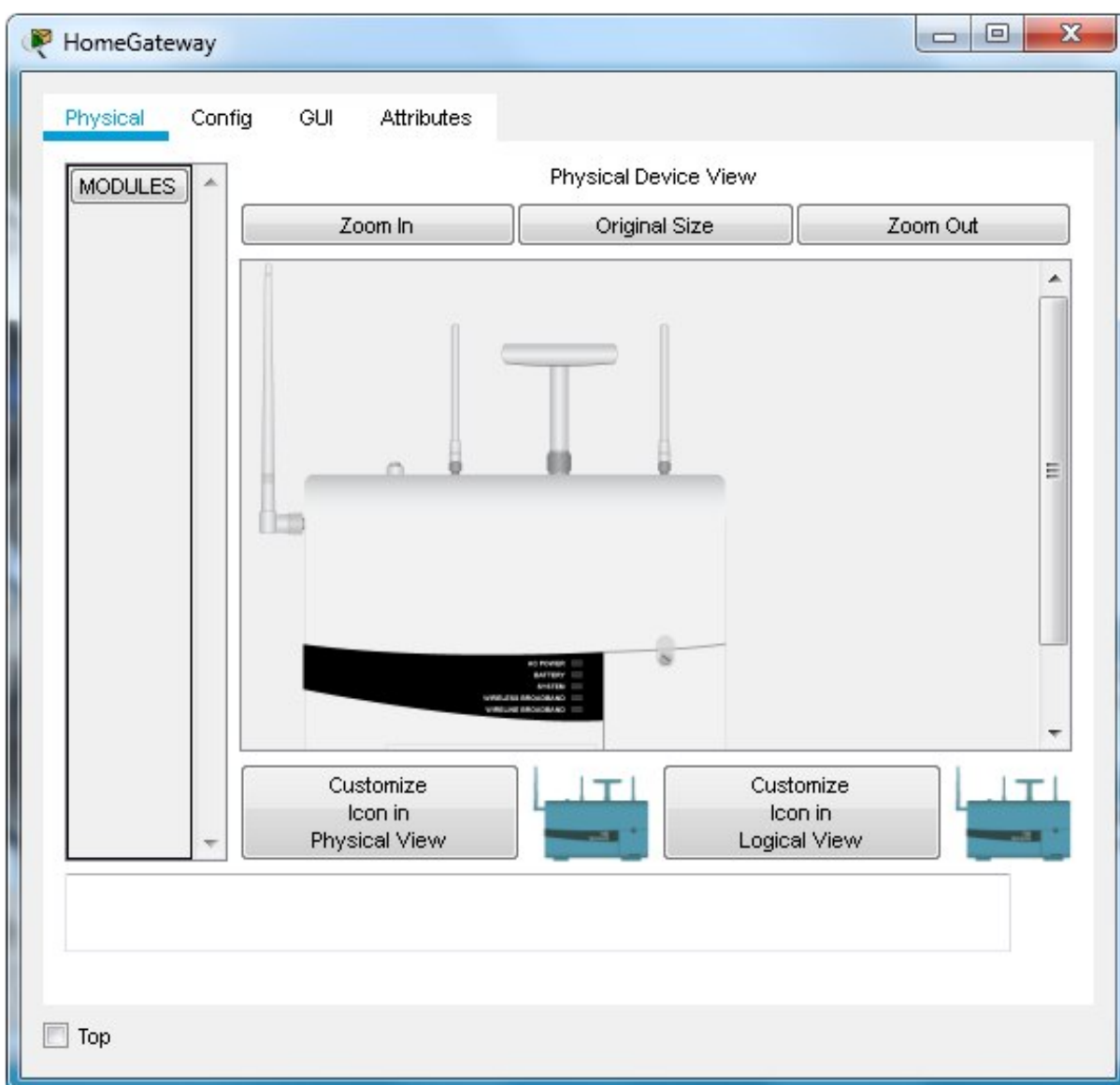


Рисунок 3.1 – Зображення головного шлюза

Для перегляду додаткових налаштувань необхідно перейти на вкладку «Config» (рис.3.2). Змінюємо його іменування (рис.3.3.)

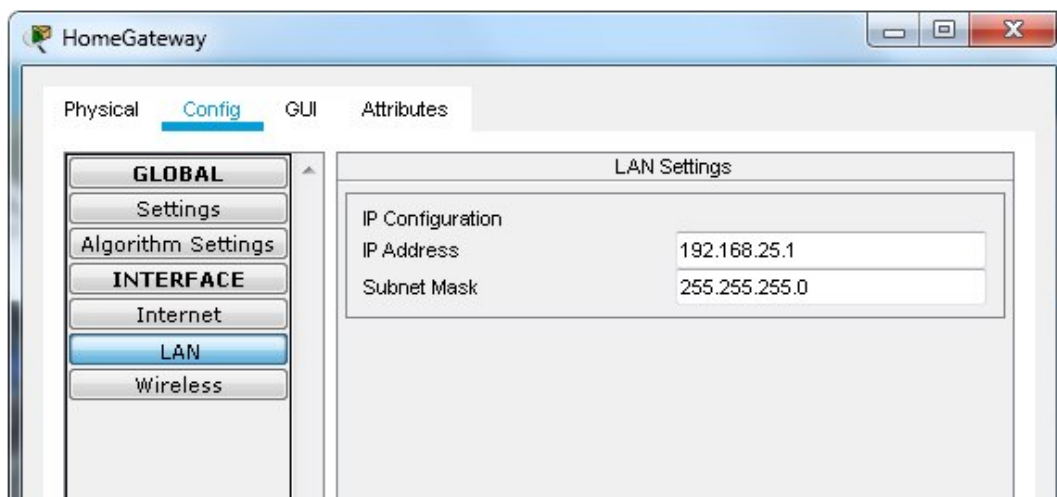


Рисунок 3.2 – Вікно властивостей головного шлюза

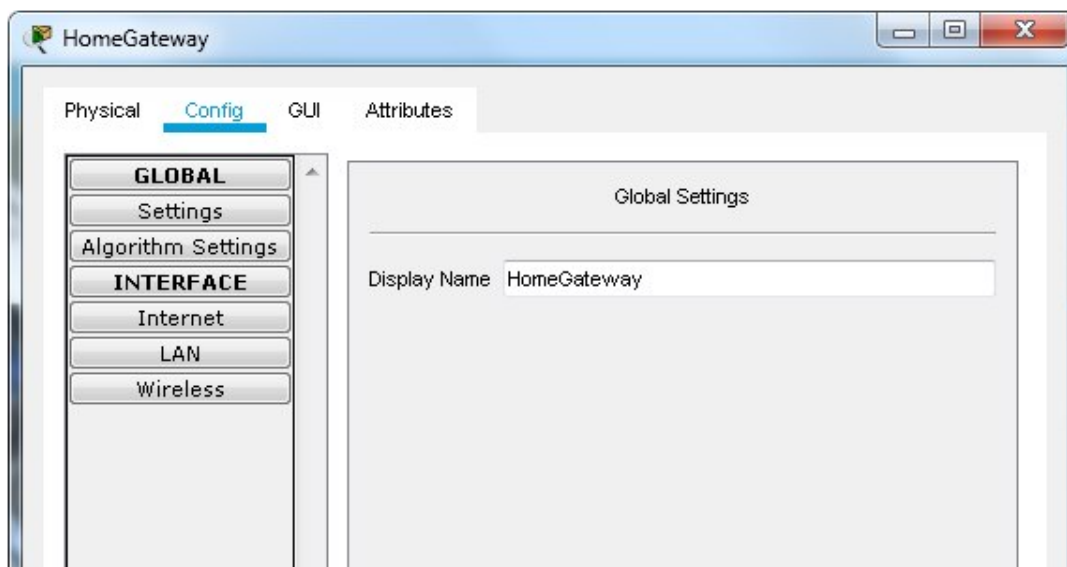


Рисунок 3.3 – Назва головного шлюза

Відкривши інтерфейс «Wireless» ми повинні виконати додаткові налаштування захисту. Для цього на необхідно обрати специфікації шифрування «WPA2-PSK» та додати пароль [21].

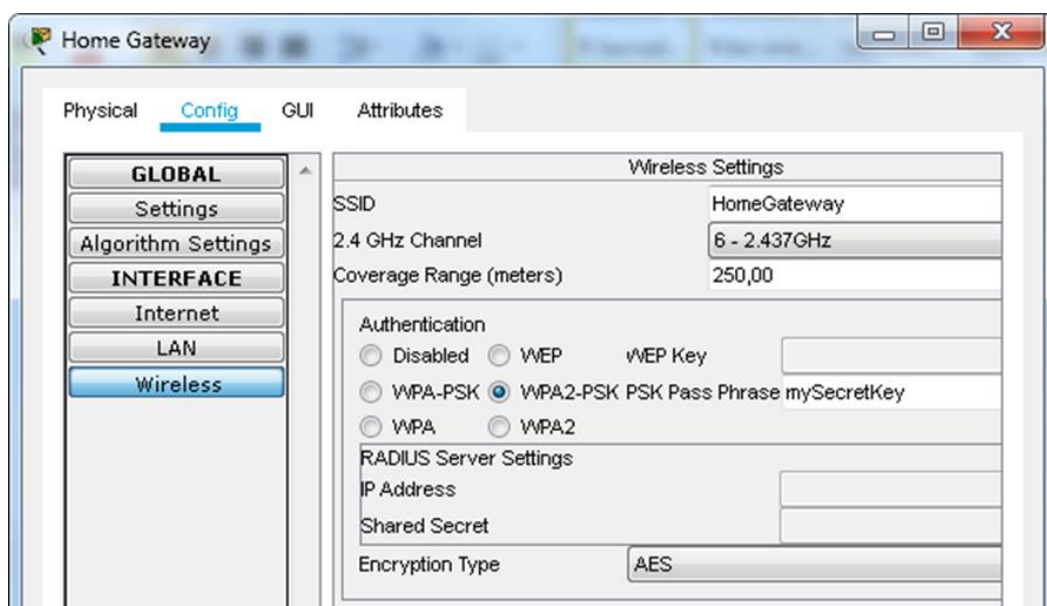


Рисунок 3.4 – Налаштування безпеки головного шлюза

Для керування мережею «Розумний Дім» необхідно додати планшет. Планшет необхідно під'єднати до головного шлюзу. Відкривши інтерфейс «Wireless0» на додатковій вкладці «Config», вказуємо SSID – «HomeGateway». Обираємо шифрування «WPA2-PSK» та додаємо пароль (рис.3.5).

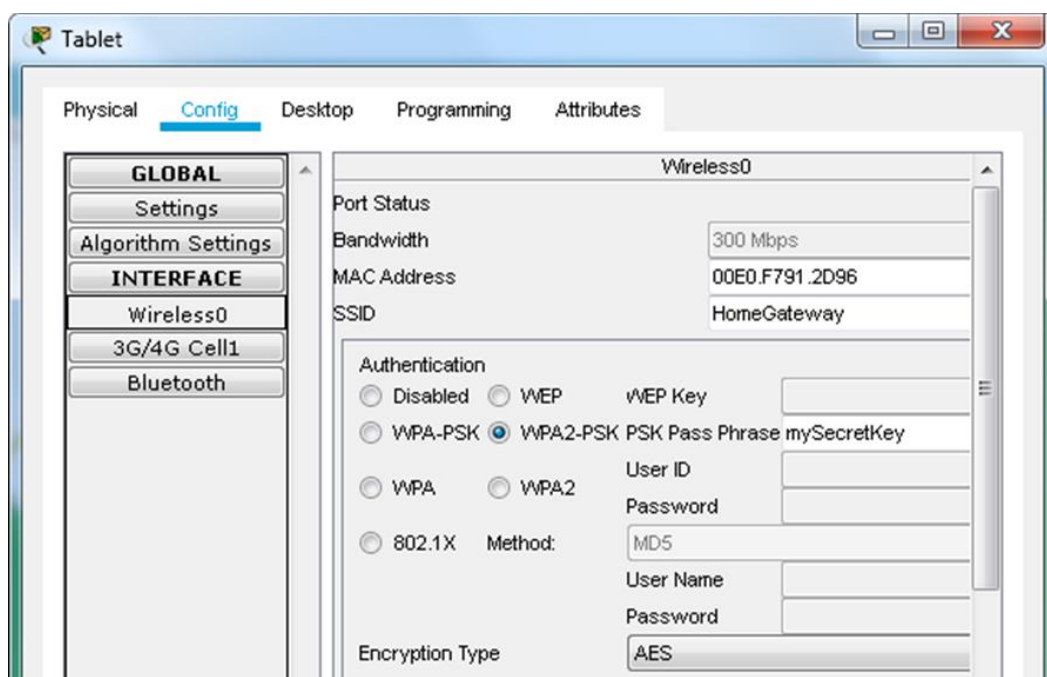


Рисунок 3.5 – Налаштування планшету

Додаємо новий пристрій, а саме пристрій для поливу. Для налаштування роботи системи поливу через мережи переходимо в меню конфігурацій «Advanced» на вкладку «Config». Всі налаштування представлені на рис.3.6.

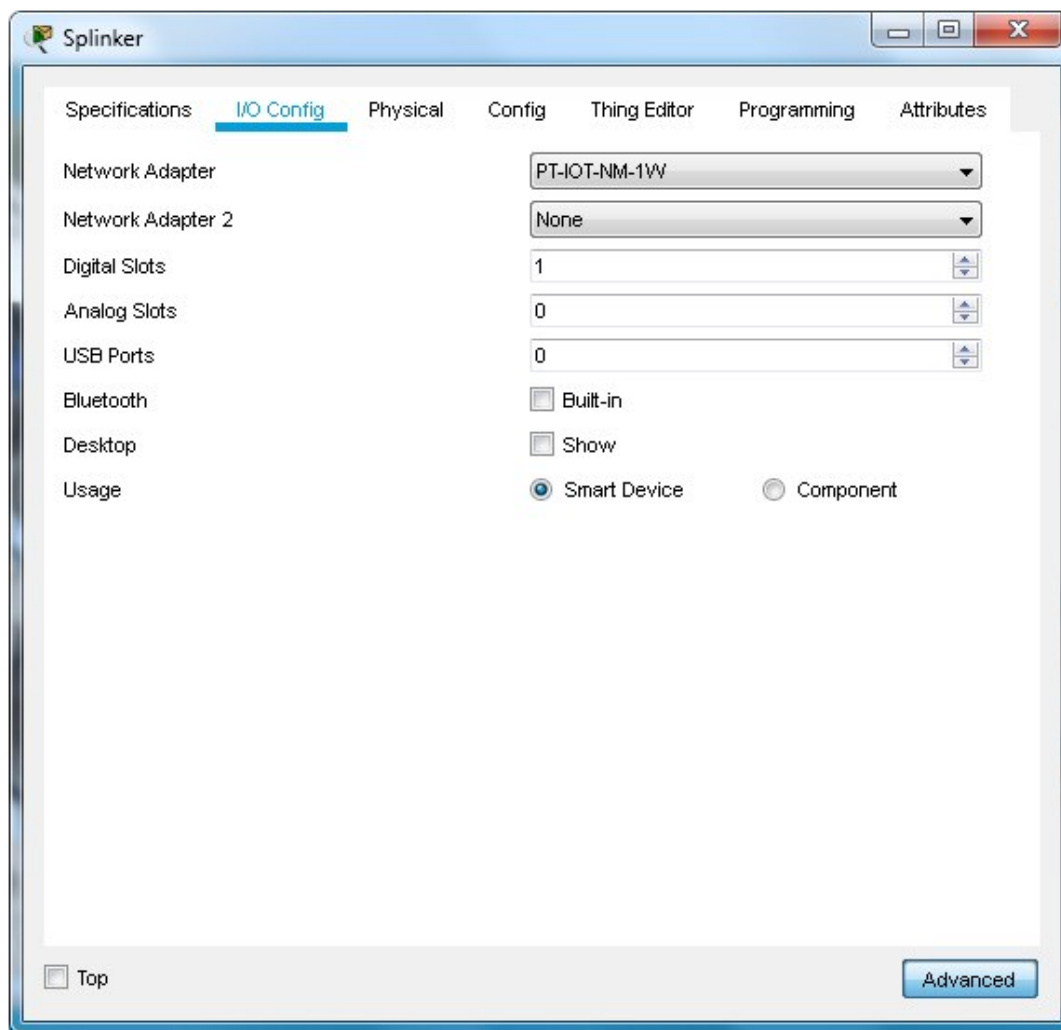


Рисунок 3.6 – Налаштування мережевого адаптеру розприскувача

Для того щоб перейменувати пристрій необхідно перейти на вкладку «Config». В полі імені прописуємо «Splinker». В налаштуваннях «Server» обираємо підключення через «Home Gateway» [22]. Відкривши інтерфейс «Wireless0», вказуємо SSID та тип шифрування.

Детальне налаштування представлено на рис.3.7.

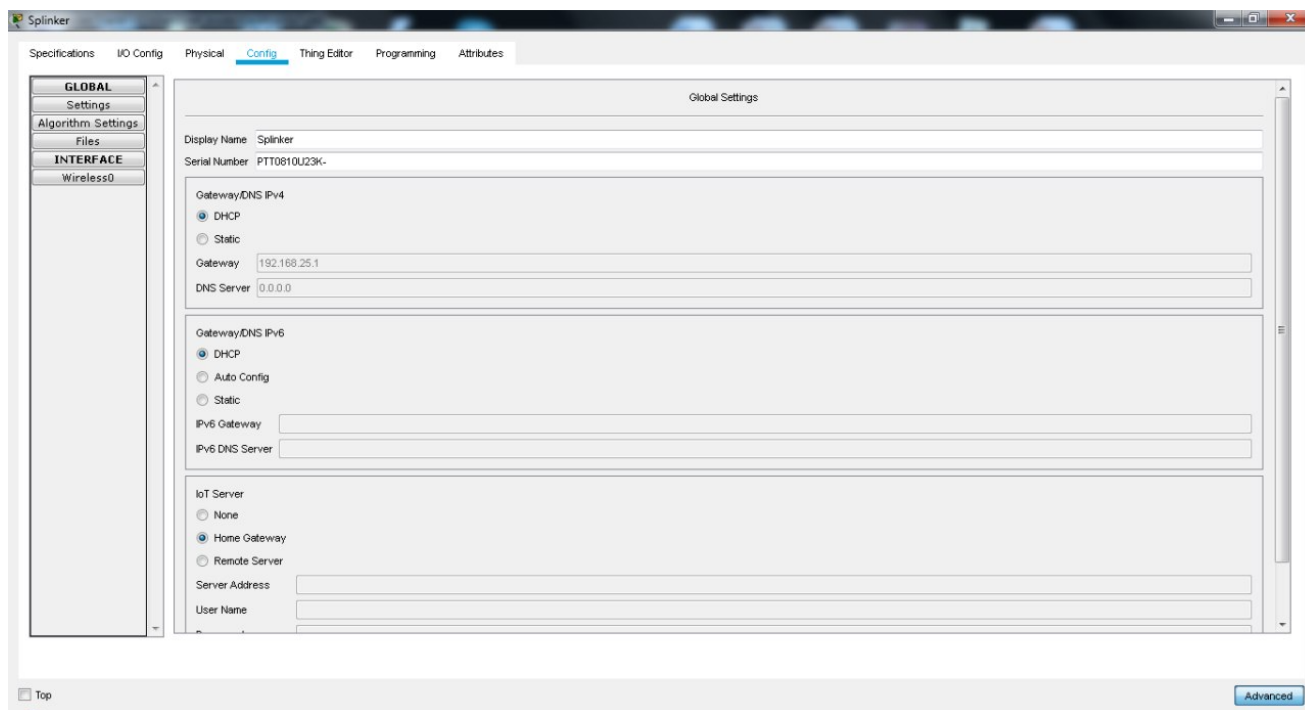


Рисунок 3.7 – Налаштування конфігурацій пристрою поливу

Додаємо ще два пристрої: двері та лампу. Виконаємо аналогічні налаштування. Наступний крок – додаємо температурний датчик. Виконуємо налаштування конфігурацій температурного датчику (рис.3.8).

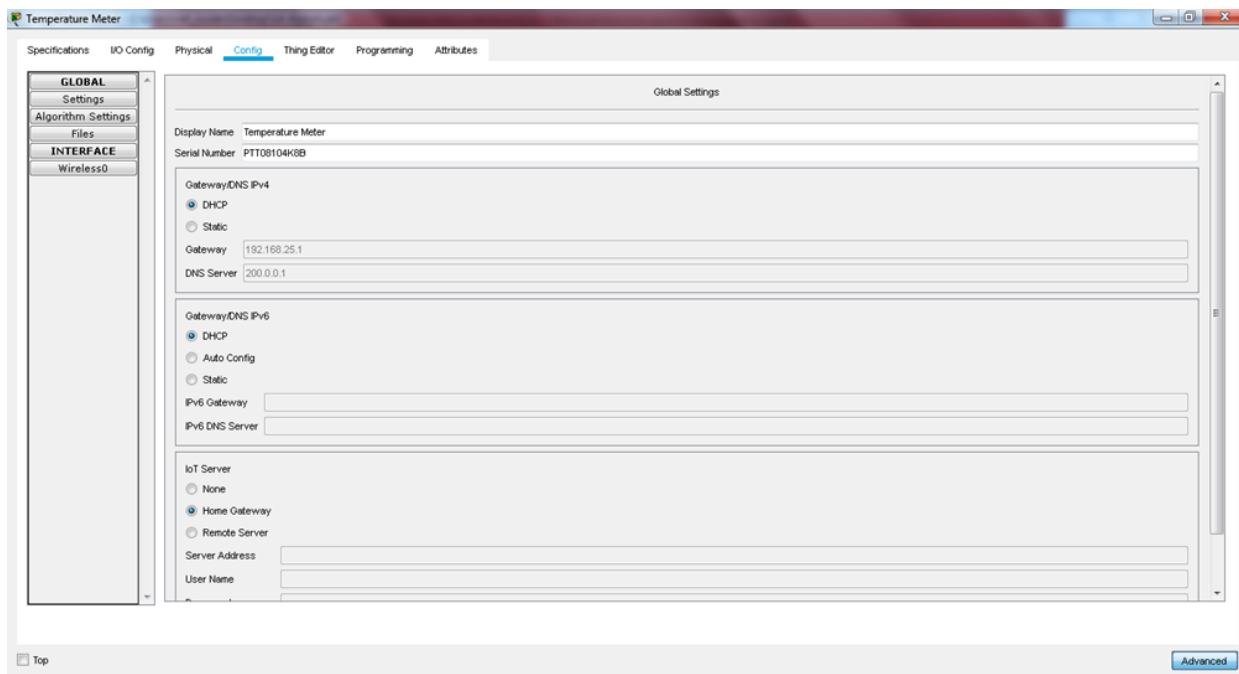


Рисунок 3.8 – Налаштування конфігурацій температурного датчику

На інтерфейсі «Wireless0» виконуємо налаштування, як на рис.3.9.

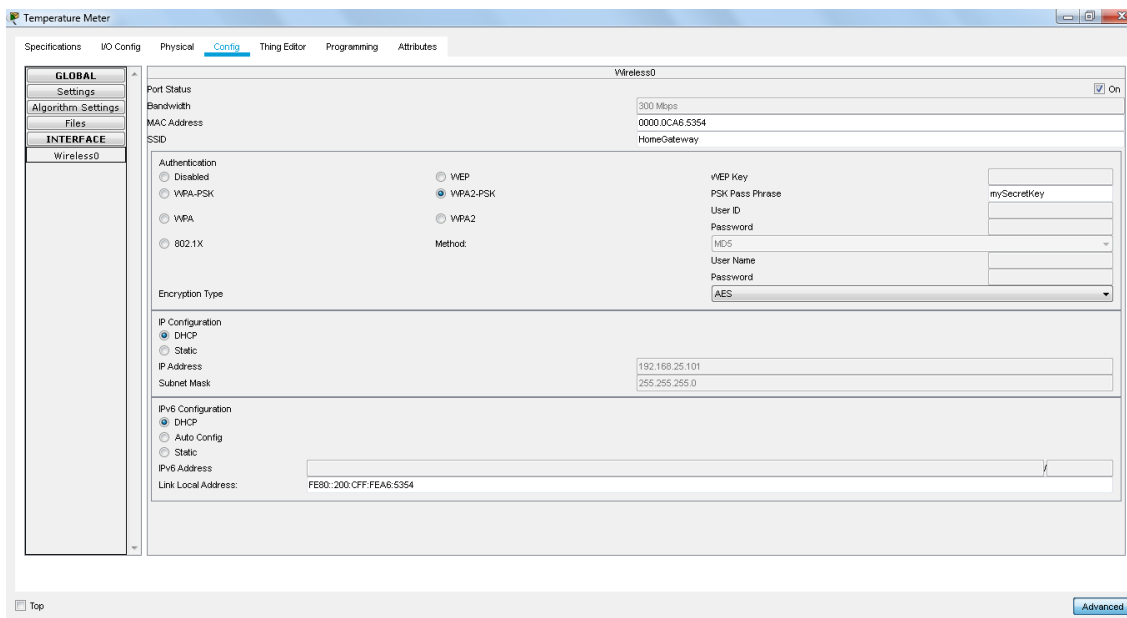


Рисунок 3.9 – Налаштування аутентифікації температурного датчика

Щоб система поливу працювала автоматично, необхідно встановити температурний режим на вкладці «Environment» (рис.3.10-11).

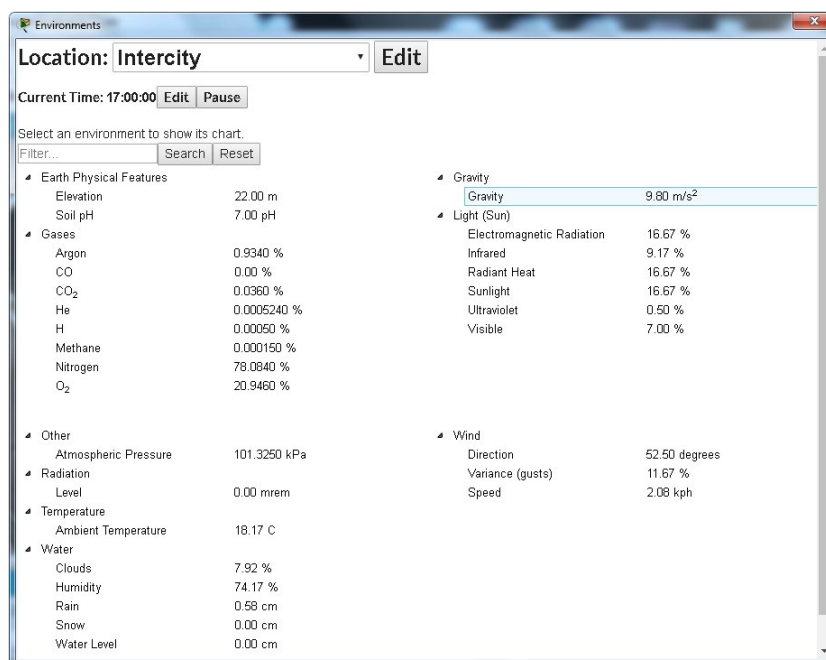


Рисунок 3.10 – Данні навколишнього середовища

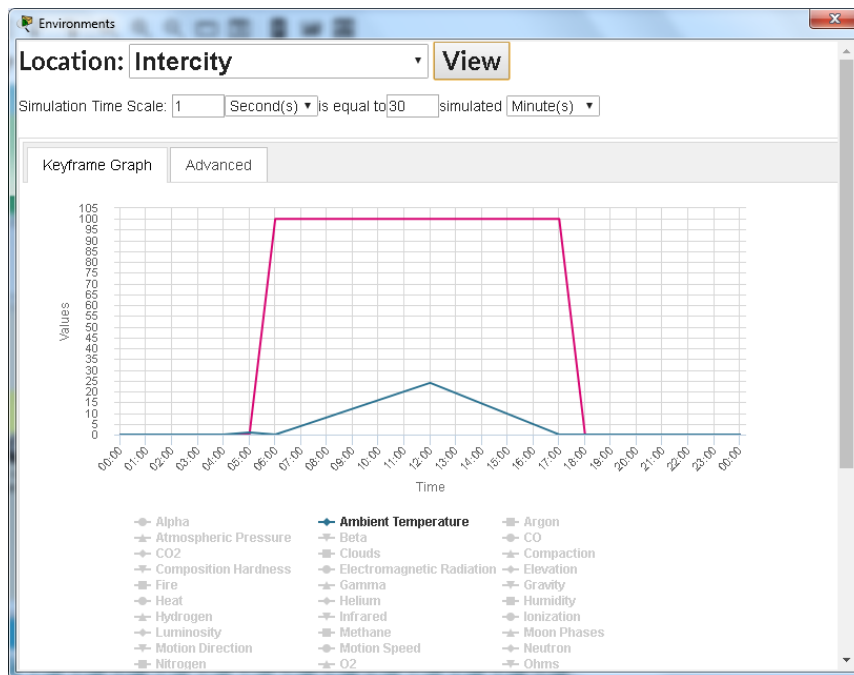


Рисунок 3.11 – Графік щоденної температури

Для налаштування умов роботи розприскувача відкриваємо на планшеті «WebBrowser» на вкладці Desktop. Підключаємося до шлюзу (рис.3.12).

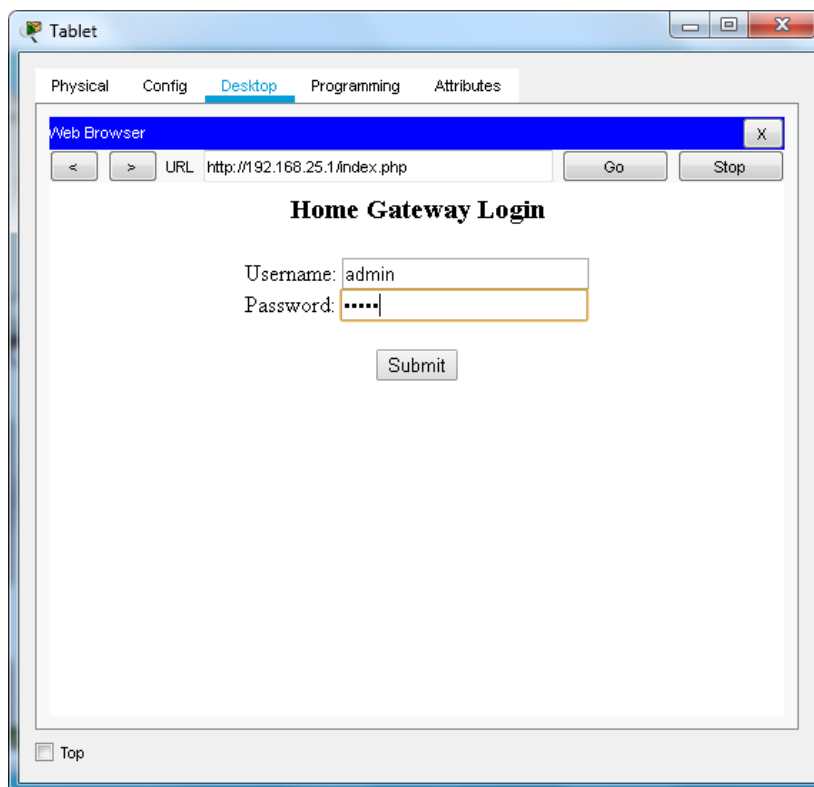


Рисунок 3.12 – Авторизація шлюзу

Підключившись ми побачимо список під'єднаних пристроїв (рис.3.13).

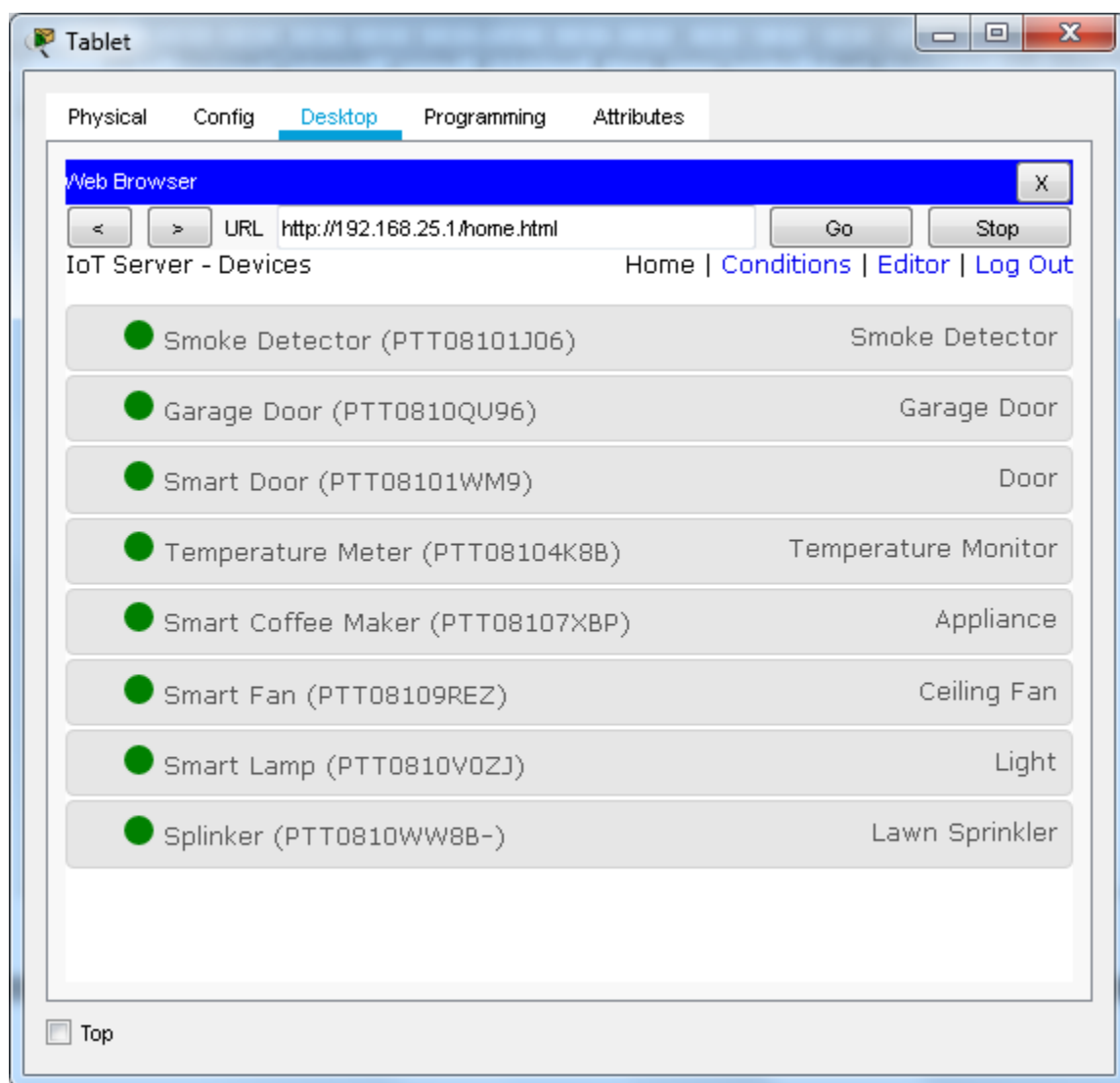


Рисунок 3.13 – Список під'єднаних пристроїв

Для створення умов роботи розприскувача та лампи виконуємо редагування в «Condition». Потрібно відтворити умову вмикання поливу Add. Налаштування представлені на рис.3.14.

Виконаємо також аналогічні налаштування для інших температурних режимів (рис.3.15-16).

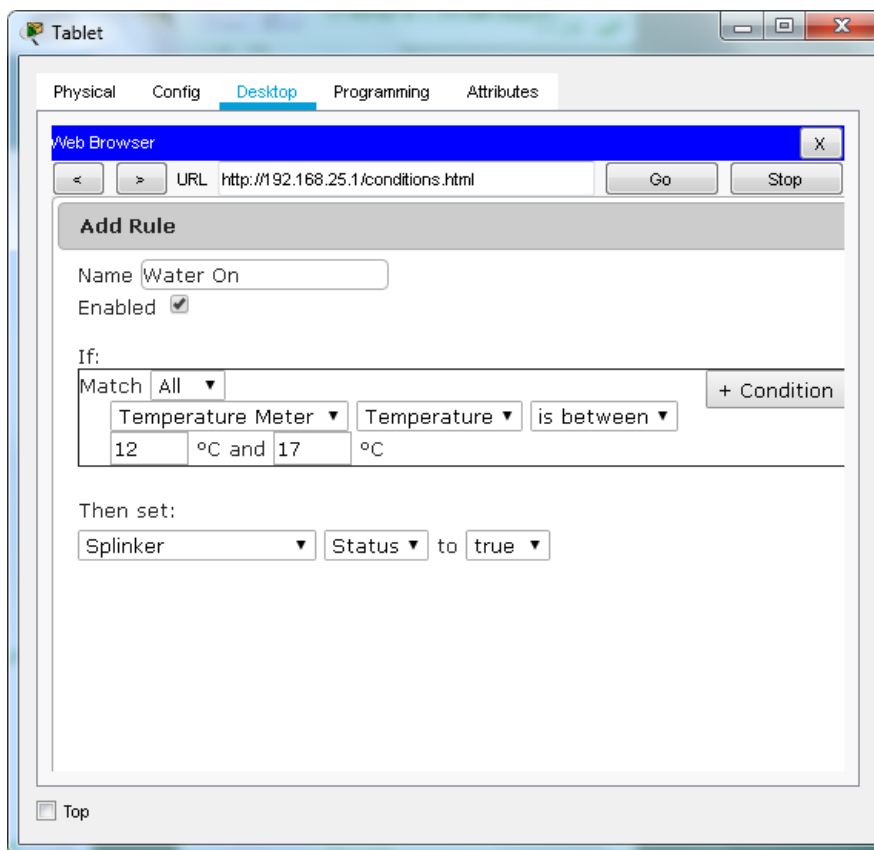


Рисунок 3.14 – Налаштування умови та дії приладу поливу

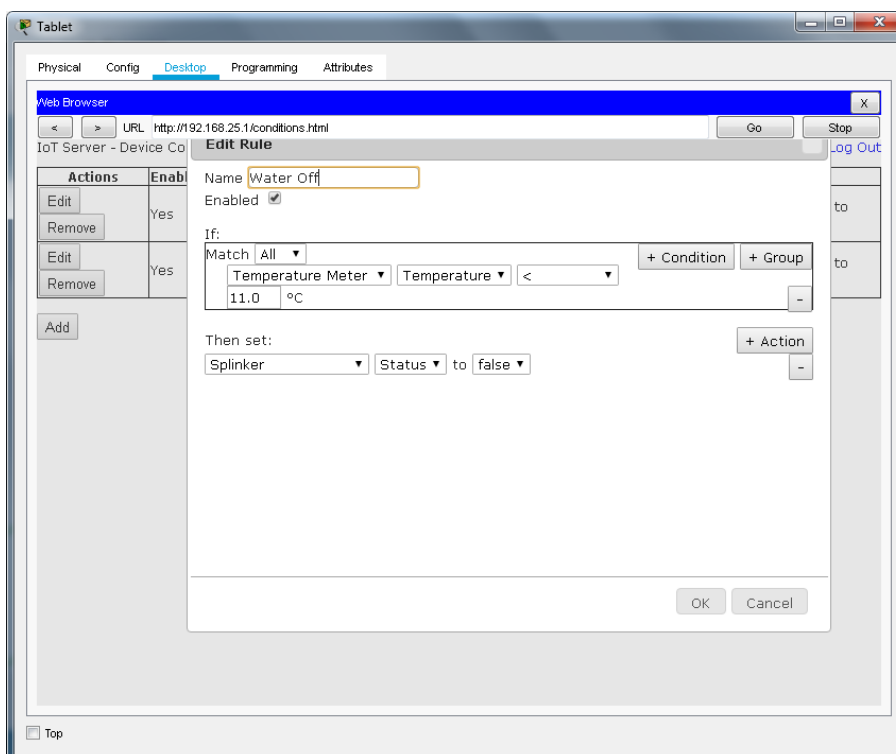


Рисунок 3.15 – Налаштування умови та дії приладу поливу (при температурі менше ніж 11 градусів за Цельсієм)

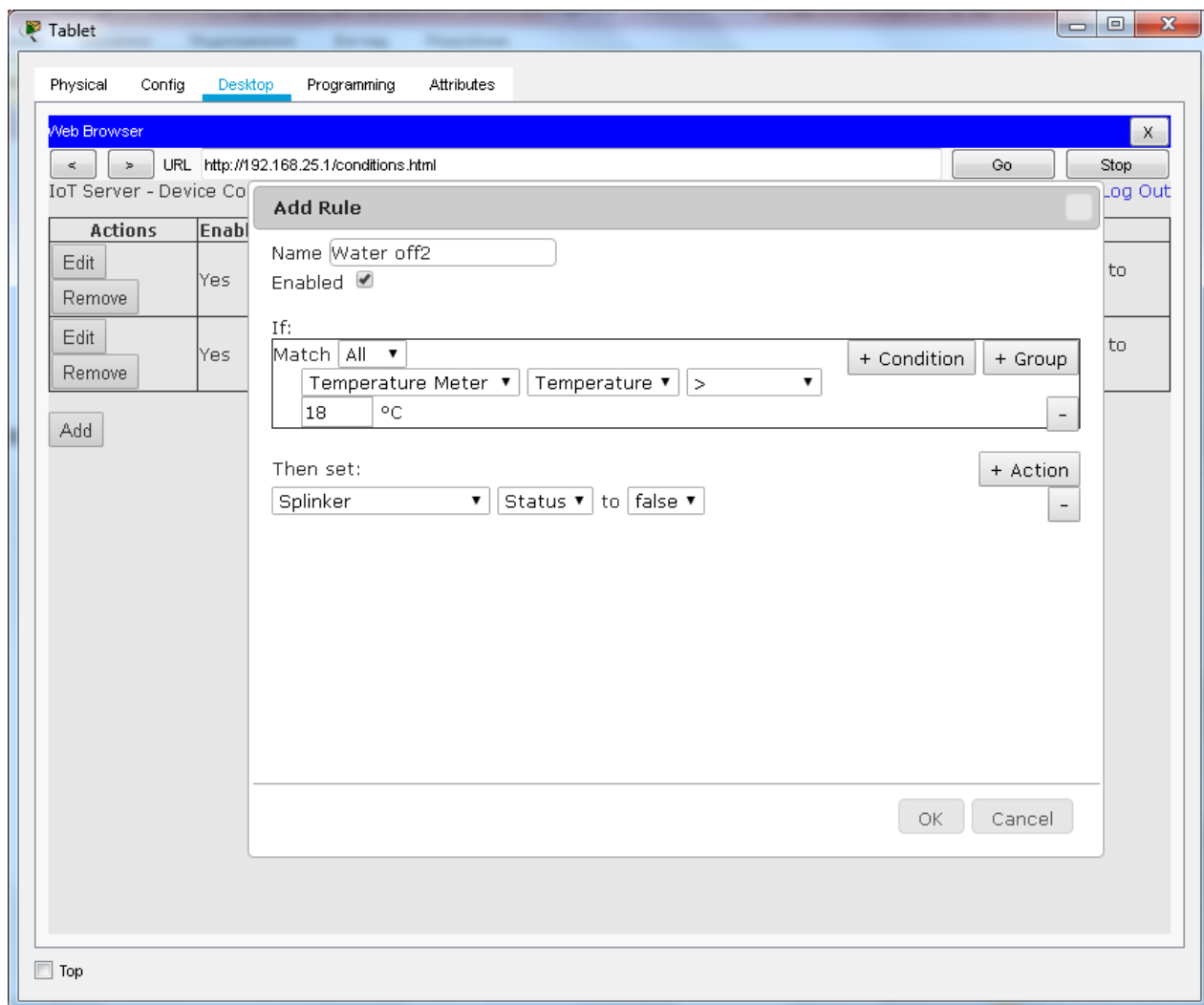
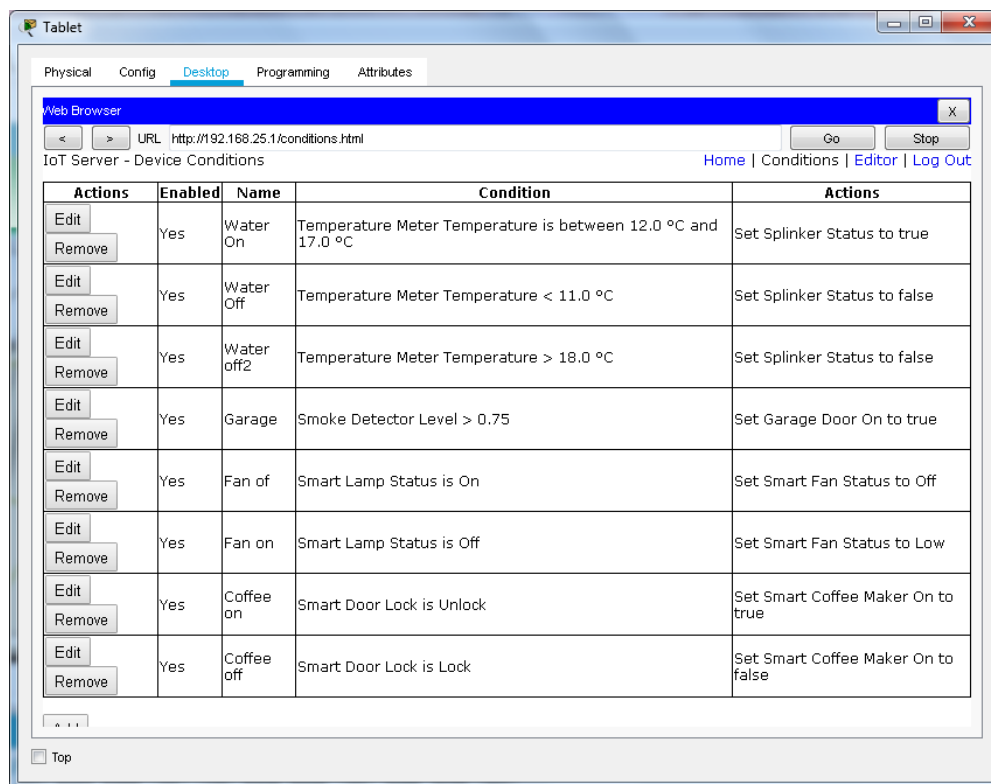


Рисунок 3.16 – Налаштування умови та дії приладу поливу (при температурі більшої ніж 18 градусів за Цельсієм)

Для налаштування умов роботи лампи також потрібно також додати додаткових три правила:

1. Відчиняються двері – світло вмикається;
2. Зачиняються двері, але не замкненні на ключ – світло ввімкнене;
3. Двері зачинені на ключ –вимкнуті світло.

Приклади налаштування умови та умови для світла в приміщенні на рис.3.17. Як бачимо, система «розумний дім» налаштована та працює в правильному режимі (рис.3.18).



Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	Water On	Temperature Meter Temperature is between 12.0 °C and 17.0 °C	Set Splinker Status to true
Edit Remove	Yes	Water Off	Temperature Meter Temperature < 11.0 °C	Set Splinker Status to false
Edit Remove	Yes	Water off2	Temperature Meter Temperature > 18.0 °C	Set Splinker Status to false
Edit Remove	Yes	Garage	Smoke Detector Level > 0.75	Set Garage Door On to true
Edit Remove	Yes	Fan of	Smart Lamp Status is On	Set Smart Fan Status to Off
Edit Remove	Yes	Fan on	Smart Lamp Status is Off	Set Smart Fan Status to Low
Edit Remove	Yes	Coffee on	Smart Door Lock is Unlock	Set Smart Coffee Maker On to true
Edit Remove	Yes	Coffee off	Smart Door Lock is Lock	Set Smart Coffee Maker On to false

Рисунок 3.17 – Приклад налаштування умови та дій для світла в приміщенні

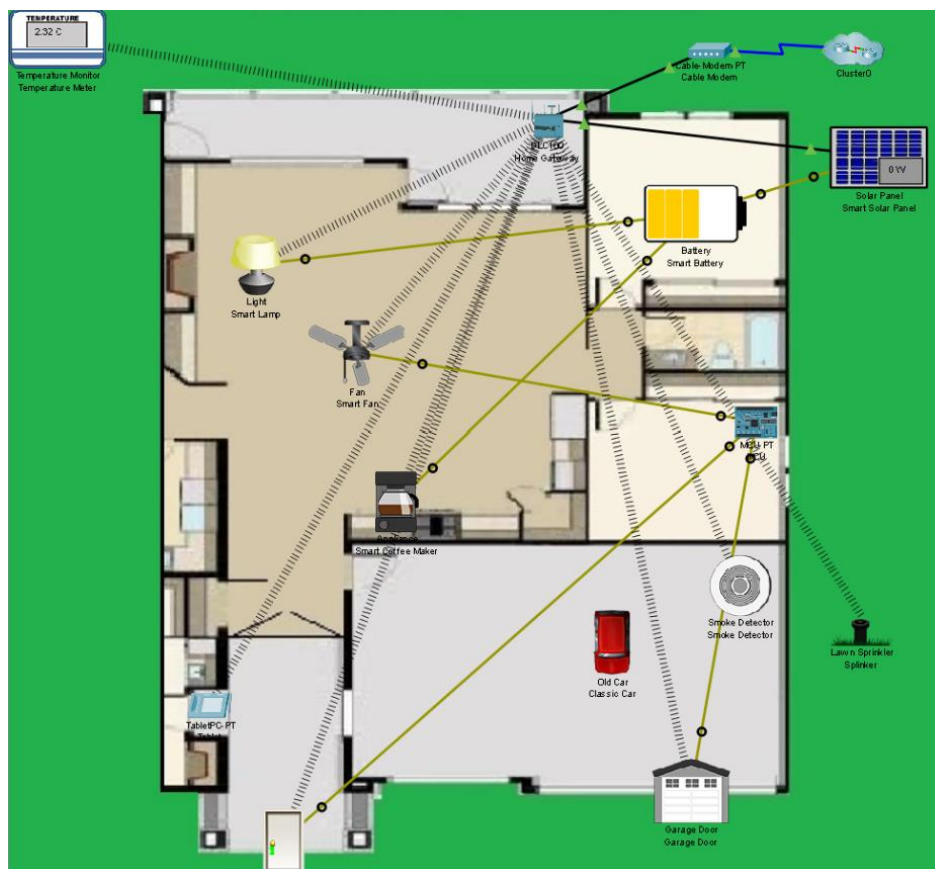


Рисунок 3.18 – Приклад роботи системи «Розумний дім»

3.2 Використання системи

Схема була розроблена та продемонстрована у вигляді веб-додатку. Сама система представлена на рис.3.19.

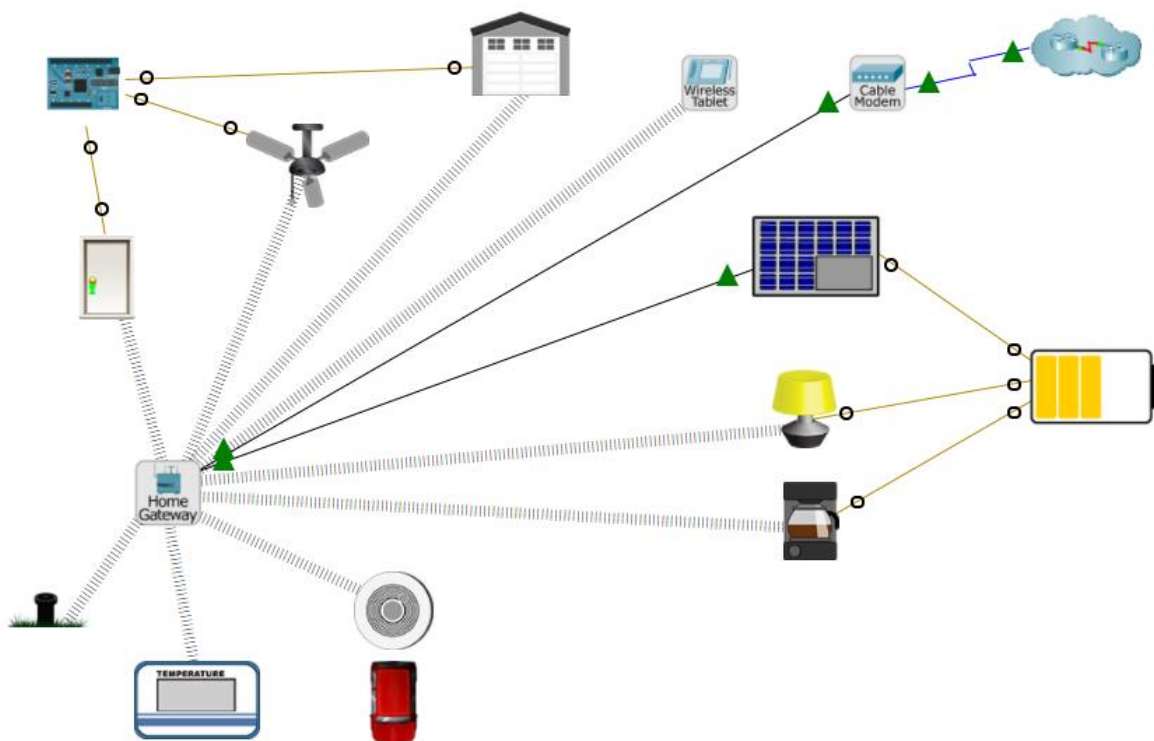


Рисунок 3.19 – Приклад роботи веб-додатку

Щоб продемонструвати роботу технології IoT була побудована схема у якій використані:

1. Шлюз «DLC100»;
2. Планшет «Wireless TabletPC-PT»;
3. Прилад для поливу газону «Lawn Sprinkler»;
4. Температурний датчик «Temperature Monitor»;
5. Лампа «SmartLight»;
6. Двері «SmartDoor»;
7. Вентилятор «SmartFan»;
8. Кофе машина «SmartCoffeeMaker»;
9. Гараж «SmartGarage»;
10. Датчик диму;
11. Сонячна панель «Smart Solar Panel»;

12. Батарея «SmartBattery» заряджаються від сонячної панелі, та використовуються для живлення лампи і кофе машини.

Для прикладу розглянемо налаштування «SmartLamp» (рис.3.20-22). Для їх перегляду потрібно натиснути на відповідне зображення на створеній схемі «розумного будинку».

I/O Config	Config(Advanced)	Interface(Wireless0)
Network Adapter	PT-IOT-NM-1W	
Network Adapter 2	None	
Digital Slots	1	
Analog Slots	0	
USB Ports	0	
Bluetooth	<input type="checkbox"/> Built-in	
Desktop	<input type="checkbox"/> Show	
Usage	<input checked="" type="radio"/> Smart Device <input type="radio"/> Component	

Рисунок 3.20 – Вкладка «I/O Config» з «SmartLamp»

I/O Config	Config(Advanced)	Interface(Wireless0)
Display Name	SmartLamp	
Serial Number	PTT0810V0ZJ	
Gateway/DNS IPv4	<input checked="" type="radio"/> DHCP <input type="radio"/> Static	
Gateway	192.168.25.1	
DNS Server	200.0.0.1	
Gateway/DNS IPv6	<input type="radio"/> DHCP <input checked="" type="radio"/> Static	
IPv6 Gateway		
IPv6 DNS Server		
IoT Server	<input checked="" type="radio"/> Home Gateway <input type="radio"/> Remote Server <input type="radio"/> Home Gateway	

Рисунок 3.21 – Вкладка «Config (Advanced)» з «SmartLamp»

I/O Config	Config(Advanced)	Interface(Wireless0)
Port Status	<input checked="" type="checkbox"/> On	
Bandwidth	<input type="text" value="200"/>	
MAC Address	<input type="text" value="0002.4A13.D4DD"/>	
SSID	<input type="text" value="HomeGateway"/>	
Authentication	<input type="radio"/> Disabled <input type="radio"/> WEP	
WEP Key	<input type="text"/>	
	<input type="radio"/> WPA-PSK <input checked="" type="radio"/> WPA2-PSK	
PSK Pass Phrase	<input type="text" value="mySecretKey"/>	
	<input type="radio"/> WPA <input type="radio"/> WPA2	
User ID	<input type="text"/>	
	<input type="radio"/> 802.1X	
Method	<input type="text" value="MD5"/>	

Рисунок 3.22 – Вкладка «Interface(Wireless)» з «SmartLamp»

Розглянемо також «Smart Door» (рис.3.23) та «HomeGateway» (рис.3.24-25).

I/O Config	Config(Advanced)	Interface(Wireless0)
Display Name	<input type="text" value="SmartDoor"/>	
Serial Number	<input type="text" value="PTT08101WM9"/>	
Gateway/DNS IPv4	<input checked="" type="radio"/> DHCP <input type="radio"/> Static	
Gateway	<input type="text" value="192.168.25.1"/>	
Gateway/DNS IPv6	<input type="radio"/> DHCP <input checked="" type="radio"/> Static	
IPv6 Gateway	<input type="text"/>	
IPv6 DNS Server	<input type="text"/>	
IoT Server	<input checked="" type="radio"/> Home Gateway <input type="radio"/> Remote Server <input type="radio"/>	
	<input type="text" value="Home Gateway"/>	

Рисунок 3.23 – Вкладка «Config (Advanced)» з «Smart Door»

Config(Advanced)	Interface(FastEthernet0)	Interface(LAN)
SSID	HomeGateway	
2.4Ghz Channel	6-2.437GHz	
Coverage Range(meters)	250.00	
Authentication	<input type="radio"/> Disabled <input type="radio"/> WEP	
WEP Key		
	<input type="radio"/> WPA-PSK <input checked="" type="radio"/> WPA-PSK2	
PSK Pass Phrase	mySecretKey	
	<input type="radio"/> WPA <input type="radio"/> WPA2	
IP Address		
Shared Secret		
Encryption Type	AES	

Рисунок 3.24 – Вкладка «Interface(FastEthernet)» з «HomeGateway»

Config(Advanced)	Interface(FastEthernet0)	Interface(LAN)
IP Address	192.168.25.1	
Subnet Mask	255.255.255.0	

Рисунок 3.25 – Вкладка «Interface(LAN)» з «HomeGateway»

Розглянемо також «Splinker» (рис.3.26-27) та «TemperatureMeter» (рис.3.24-25).

I/O Config	Config(Advanced)	Interface(Wireless0)
Display Name	<input type="text" value="Splinker"/>	
Serial Number	<input type="text" value="PTT0810WW8B-"/>	
Gateway/DNS IPv4	<input checked="" type="radio"/> DHCP <input type="radio"/> Static	
Gateway	<input type="text" value="192.168.25.1"/>	
DNS Server	<input type="text" value="200.0.0.1"/>	
Gateway/DNS IPv6	<input type="radio"/> DHCP <input checked="" type="radio"/> Static	
IPv6 Gateway	<input type="text"/>	
IPv6 DNS Server	<input type="text"/>	
IoT Server	<input checked="" type="radio"/> Home Gateway <input type="radio"/> Remote Server	
Server Address	<input type="text"/>	
User Name	<input type="text"/>	
Password	<input type="text"/>	

Рисунок 3.26 – Вкладка «Config (Advanced)» з «Splinker»

I/O Config	Config(Advanced)	Interface(Wireless0)
Port Status	<input checked="" type="checkbox"/> On	
Bandwidth	<input type="text" value="300"/>	
MAC Address	<input type="text" value="000D.BDB4.7AA1"/>	
SSID	<input type="text" value="HomeGateway"/>	
Authentication	<input type="radio"/> Disabled <input type="radio"/> WEP	
WEP Key	<input type="text"/>	
	<input type="radio"/> WPA-PSK <input checked="" type="radio"/> WPA2-PSK	
PSK Pass Phrase	<input type="text" value="mySecretKey"/>	
	<input type="radio"/> WPA <input type="radio"/> WPA2	
User ID	<input type="text"/>	
	<input type="radio"/> 802.1X	
Method	<input type="text" value="MD5"/>	
User Name	<input type="text"/>	
Password	<input type="text"/>	
Encryption Type	<input type="text" value="AES"/>	

Рисунок 3.27 – Вкладка «Interface(Wireless)» з «Splinker»

I/O Config	Config(Advanced)	Interface(Wireless0)
MAC Address	0000.0CA6.5354	
SSID	HomeGateway	
Authentication	<input type="radio"/> Disabled <input type="radio"/> WEP	
WEP Key		
	<input type="radio"/> WPA-PSK <input checked="" type="radio"/> WPA2-PSK	
PSK Pass Phrase	mySecretKey	
	<input type="radio"/> WPA <input type="radio"/> WPA2	
User ID		
	<input type="radio"/> 802.1X	
Method	MD5	
User Name		
Password		
Encryption Type	AES	
IP Configuration	<input checked="" type="radio"/> DHCP <input type="radio"/> Static	
IP Address	192.168.25.116	
Subnet Mask	255.255.255.0	

Рисунок 3.27 – Вкладка «Interface(Wireless)» з «TemperatureMeter»

Модуль «Cable Modem» використовується для доступу до розумного дому із мережі Інтернет, що представлено на рис.3.26.

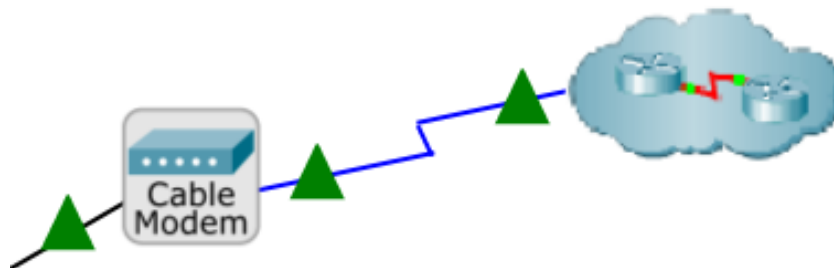


Рисунок 3.26 – Модуль «Cable Modem»

«Home Controller MCU» виконує роль синхронізатора між усіма елементами розумного дому (рис.3.27).

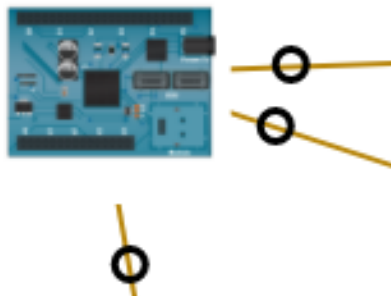


Рисунок 3.27 – Модуль «Home Controller MCU»

У ході створення веб-додатку було реалізовано технологію Інтернету речей для керування мережею «розумний будинок». Працездатність системи успішно перевірена у додатку-симуляторі Cisco Packet Tracer 7.

ВИСНОВКИ

Під час виконання випускної роботи було проведено аналіз тематики розробки «розумного будинку». Були проаналізовані переваги та недоліки даної тематики.

Проаналізовано технології для реалізації дипломної роботи та сучасні мови програмування, що допомагають взаємодіями із іншими інструментами. Для розробки було обрано JavaScript.

У ході виконання роботи був створений приклад налаштування технології IoT у програмному додатку Cisco Packet Tracer. В результаті за допомогою IoT концепцій було відтворено мережу «розумного будинку». Для проектування системи було використано бездротовий зв'язок. Перевірено, що усі з'єднані пристрої доступні для керування та моніторингу через веб-інтерфейс планшета.

Також розроблено веб-додаток для представлення роботи розробленої мережі «розумного будинку».

СПИСОК ЛІТЕРАТУРИ

1. The Best Smart Home Devices for 2021 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.pcmag.com/news/the-best-smart-home-devices> (дата звернення: 18.04.2021);
2. Types of Smart Home Technology Optionshttps [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: [//www.homestratosphere.com/smart-home-technology-ultimate-guide/](https://www.homestratosphere.com/smart-home-technology-ultimate-guide/) (дата звернення: 18.04.2021);
3. Everything You Need to Know About Smart Home Technology [Електронний ресурс]. – 2021. – Режим доступу до ресурсу:<https://www.otelco.com/resources/smart-home-guide/> (дата звернення: 18.04.2021);
4. What is Smart Home Technology? [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://payspacemagazine.com/tech/smart-home-technology-pros-and-cons/> (дата звернення: 03.05.2021);
5. Smart home technology: pros and cons [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.engineess.io/insights/what-is-smart-home-technology> (дата звернення: 03.05.2021);
6. Smart Home Technology [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://axiomq.com/blog/smart-home-technology-pros-and-cons/> (дата звернення: 03.05.2021);
7. C. Wilson, T. Hargreaves, R. Nauxwell-Baldwin, Benefits and risks of smart home technologies. Energy Policy. 103, 72–83 (2017);
8. Ajax [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://ajax.systems/ru-ua/> (дата звернення: 12.05.2021);
9. Fibaro [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.fibaro.com/ru/products/all-domotica-devices/> (дата звернення: 12.05.2021);

10. Orviro [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: https://www.orvibo.com/en/product/productList_index.html (дата звернення: 12.05.2021);
11. What is Cisco Packet Tracer? [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.netacad.com/courses/packet-tracer/faq> (дата звернення: 12.05.2021);
12. Everything You Need To Know About Cisco Packet Tracer [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.mymagicfundas.com/cisco-packet-tracer/> (дата звернення: 18.05.2021);
13. T. Hargreaves, C. Wilson, R. Hauxwell-Baldwin, Learning to live in a smart home. Building Research and Information. 46, 127–139 (2018);
14. The Advantages and Disadvantages of JavaScript [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/> (дата звернення: 18.05.2021);
15. JavaScript [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-javascript/> (дата звернення: 18.05.2021);
16. Using JavaScript for everything [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://medium.com/@bellomuba.rak0/using-javascript-for-everything-the-pros-and-cons-dd69e6a49712> (дата звернення: 25.05.2021);
17. Why use Python for building your next project [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://proxify.io/de/articles/python-pros-and-cons> (дата звернення: 25.05.2021);
18. ADVANTAGES AND DISADVANTAGES OF PYTHON FOR YOUR BUSINESS [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://inoxoft.com/advantages-and-disadvantages-of-python-for-your-business/> (дата звернення: 25.05.2021);

19. Python [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://data-flair.training/blogs/advantages-and-disadvantages-of-python/> (дата звернення: 25.05.2021);

20. R. R. Chaudhari, K. K. Joshi, N. Joshi, M. Kumar, Smart and secure home using IOT Simulations with Cisco Packet Tracer. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 88–93 (2020);

21. M. Elsaid, S. Altuwaijri, N. Aljammaz, A. Ara, Design and Analysis of Secure Smart Home for Elderly People. International Journal of Distributed and Parallel systems. 10, 1–13 (2019);

22. G. L. P. Ashok, P. Saleem Akram, M. Sai Neelima, J. Nagasaikumar, A. Vamshi, Implementation of smart home by using packet tracer. International Journal of Scientific and Technology Research. 9, 678–685 (2020).

ДОДАТОК А. КОД РЕАЛІЗАЦІЇ

index.html

```

<!DOCTYPE html>
<html>
<head>
  <title>IoT</title>
  <link rel="stylesheet" href="styles.css">
  <script src="script.js"></script>
  <script src="data.js"></script>
</head>
<body>
<div class="main">
  <div class="container">
    <div class="schema_container" id="schema">
      
      
      
      
      
      
      
      
      
      
      
      
      
      
      
      
      
    </div></div>
  </div>
</body></html>

```

styles.css

```
* {  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
  padding: 0;  
  margin: 0;  
}
```

```
.main {  
  width: 100vw;  
  height: 100vh;  
}
```

```
.container {  
  display: flex;  
}
```

```
.schema_container {  
  width: 100vw;  
  height: 100vh;  
  position: relative;  
}
```

```
.main_btn {  
  width: 200px;  
  height: 64px;  
  border-radius: 12px;  
  margin-bottom: 16px;  
  border: none;  
  outline: none;  
  cursor: pointer;  
  font-size: 20px;  
  color: floralwhite;  
  background-color: dodgerblue;  
}
```

```
.router {  
  width: 80px;  
  height: 80px;
```

```
background-color: goldenrod;
cursor: pointer;
position: relative;
}

.table {
  position: absolute;
  width: 600px;
  height: 600px;
  background-color: azure;
  border-radius: 12px;
  border: 1px solid lightblue;
  cursor: initial;
}

.header_container {
  width: 100%;
  height: 40px;
  display: flex;
  border-bottom: 1px solid lightblue;
  background-color: whitesmoke;
  border-radius: 11px 11px 0 0;
  overflow: hidden;
}

.red {
  background-color: red;
}

.header_item {
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
}

.active {
  background-color: lightskyblue;
}

.header_item:first-child {
```

```
border-right: 1px solid lightblue;
}
```

```
.fields_container {
  width: 100%;
  height: 560px;
  display: none;
  overflow: auto;
}
```

```
.field_wrapper {
  display: flex;
  align-items: center;
  width: 100%;
  height: 36px;
}
```

```
.field_item {
  width: 50%;
  padding-left: 12px;
}
```

```
code {
  display: block;
  height: 560px;
  background-color: #eee;
  border-radius: 3px;
  font-family: courier, monospace;
  font-size: 12px;
  padding: 0 3px;
  overflow: auto;
}
```

```
.code-btn {
  width: 100px;
  background-color: darkseagreen;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 6px 8px;
```

```
border-radius: 8px;
cursor: pointer;
}
```

```
.home_gateway {
  position: absolute;
  top: 500px;
  left: 150px;
  width: 60px;
  height: 60px;
}
```

```
.door {
  display: block;
  position: absolute;
  top: 300px;
  left: 100px;
}
```

```
.mcu {
  position: absolute;
  top: 100px;
  left: 40px;
  transform: scale(0.6)
}
```

```
.fan {
  position: absolute;
  top: 200px;
  left: 250px;
}
```

```
.garage {
  position: absolute;
  top: 100px;
  left: 450px;
}
```

```
.sprinkler {
```

```
    position: absolute;
    top: 580px;
    left: 20px;
}
```

```
.temperature {
    position: absolute;
    top: 680px;
    left: 150px;
}
```

```
.car {
    position: absolute;
    top: 680px;
    left: 360px;
}
```

```
.detector {
    position: absolute;
    top: 600px;
    left: 346px;
}
```

```
.cable-modem {
    position: absolute;
    top: 100px;
    left: 750px;
    transform: scale(0.4)
}
```

```
.tablet {
    position: absolute;
    top: 100px;
    left: 600px;
    transform: scale(0.4)
}
```

```
.clust {
    position: absolute;
    top: 100px;
```



```
    left: 950px;
  }

.solar {
  position: absolute;
  top: 280px;
  left: 700px;
}

.light {
  position: absolute;
  top: 420px;
  left: 720px;
}

.coffee {
  position: absolute;
  top: 520px;
  left: 730px;
}

.battery {
  position: absolute;
  top: 400px;
  left: 950px;
}

.led {
  position: absolute;
  top: 342px;
  left: 107px;
}
```

script.js

```
const getRandomInt = (max) => Math.floor(Math.random() * max)
```

```
const handleCopyCode = () => {
  let range = document.createRange()
  range.selectNode(document.getElementById('code'))
}
```

```

window.getSelection().removeAllRanges()
window.getSelection().addRange(range)
document.execCommand('copy')
window.getSelection().removeAllRanges()
alert('Code copied!')
}

```

```

const handleSaveCode = () => {
  const tempLink = document.createElement('a')
  let code = document.getElementById('code')
  const file = new Blob([code.innerText], {type: 'text/plain'})

  tempLink.href = URL.createObjectURL(file)
  tempLink.download = 'code.js'
  tempLink.click()

  URL.revokeObjectURL(tempLink.href)
}

```

```

const createPipeLine = (id, width, size, rotateDeg, left, top) => {
  let elemContainer = document.getElementById(id), html = "";

  let container = document.createElement('div')
  container.style.position = 'absolute'
  container.style.height = '14px'
  container.style.width = width + 'px'
  container.style.top = top + 'px'
  container.style.left = left + 'px'
  container.style.transform = `rotate(${rotateDeg}deg)`
  container.style.transformOrigin = 'top left'
  container.style.zIndex = '-1'

  for (let i = 0; i < size; i++) {
    html += '<div style="display: inline-block; font-size: 10px; width: 4px">|</div>'
  }
  container.innerHTML = html

  elemContainer.appendChild(container)
}

```

```
const createSolidLine = (id, width, rotateDeg, left, top, color) => {
  let elemContainer = document.getElementById(id), html = "

  let container = document.createElement('div')
  container.style.position = 'absolute'
  container.style.width = width + 'px'
  container.style.left = left + 'px'
  container.style.top = top + 'px'
  container.style.transform = `rotate(${rotateDeg}deg)`
  container.style.transformOrigin = 'top left'
  container.style.zIndex = '-1'
  container.style.borderBottom = `2px solid ${color}`

  container.innerHTML = html

  elemContainer.appendChild(container);
}
```

```
const createCircle = (id, left, top) => {
  let elemContainer = document.getElementById(id), html = "

  let container = document.createElement('div')
  container.style.position = 'absolute'
  container.style.width = '6px'
  container.style.height = '6px'
  container.style.left = left + 'px'
  container.style.top = top + 'px'
  container.style.zIndex = '-1'
  container.style.borderRadius = '50%'
  container.style.border = `3px solid black`

  container.innerHTML = html

  elemContainer.appendChild(container)
}
```

```
const createTriangle = (id, rotateDeg, left, top) => {
  let elemContainer = document.getElementById(id), html = "

  let container = document.createElement('div')
```

```

    container.style.position = 'absolute'
    container.style.width = '0px'
    container.style.height = '0px'
    container.style.left = left + 'px'
    container.style.top = top + 'px'
    container.style.zIndex = '0'
    container.style.borderLeft = '10px solid transparent'
    container.style.borderRight = '10px solid transparent'
    container.style.borderTop = `20px solid green`
    container.style.transform = `rotate(${rotateDeg}deg)`
    container.style.transformOrigin = 'top left'

    container.innerHTML = html

    elemContainer.appendChild(container)
  }

  const createHeader = (tabs, nameInt, nameInt2) => {
    let tabWidth = 'width:' + 100 / tabs.length + '%;'

    console.log(tabs)
    let tabsHtml = ""
    if (tabs.includes(ETabTypes.IO_CONFIG))
      tabsHtml += `<div class='header_item' style=${tabWidth} onclick="openTab(event,
ETabTypes.IO_CONFIG)">I/O Config</div>`
    if (tabs.includes(ETabTypes.CONFIG))
      tabsHtml += `<div class='header_item' style=${tabWidth} onclick="openTab(event,
ETabTypes.CONFIG)">Config(Advanced)</div>`
    if (tabs.includes(ETabTypes.CODE))
      tabsHtml += `<div class='header_item' style=${tabWidth} onclick="openTab(event,
ETabTypes.CODE)">Code</div>`
    if (tabs.includes(ETabTypes.INTERFACE))
      tabsHtml += `<div class='header_item' style=${tabWidth} onclick="openTab(event,
ETabTypes.INTERFACE)">Interface(${nameInt})</div>`
    if (tabs.includes(ETabTypes.INTERFACE2))
      tabsHtml += `<div class='header_item' style=${tabWidth} onclick="openTab(event,
ETabTypes.INTERFACE2)">Interface(${nameInt2})</div>`

    return `<div class="header_container">${tabsHtml}</div>`
  }
}

```

```

const createTextField = (name, value) => {
  return `
```

```

`<div class="field_item">` + `${radio1}` + `${radio2}` + `${radio3}` +
`</div></div>`
}

const generateTabHtml = (data) => {
  return data.map((field => {
    if (field.type === EFieldTypes.SELECT) {
      return createSelectField(field.name, field.value)
    }
    if (field.type === EFieldTypes.CHECKBOX) {
      return createCheckboxField(field.name, field.label, field.checked)
    }
    if (field.type === EFieldTypes.RADIO) {
      return createRadioField(field.name, field.value1, field.checked1, field.value2, field.checked2, field.value3,
field.checked3)
    }
    if (field.type === EFieldTypes.TEXT) {
      return createTextField(field.name, field.value)
    }
  })).join("")
}

const generateCodeHtml = (data) => {
  let btnContainer = document.createElement('div')
  btnContainer.style.display = 'flex'
  btnContainer.style.justifyContent = 'flex-start'
  btnContainer.style.width = '100%'
  btnContainer.style.height = '45px'

  let saveBtn = `<div onclick="handleSaveCode()" class="code-btn">Save</div>`
  let copyBtn = `<div onclick="handleCopyCode()" class="code-btn">Copy</div>`

  btnContainer.innerHTML = saveBtn + copyBtn

  return btnContainer.outerHTML + `<code style="white-space: pre-line" id="code">${data}</code>`
}

const createFields = (data) => {
  let html = ""
  if (data.IOConfig)

```

```

    html += `<div class="fields_container"
id=${ETabTypes.IO_CONFIG}>${generateTabHtml(data.IOConfig)}</div>`
    if (data.Config)
      html += `<div class="fields_container" id=${ETabTypes.CONFIG}>${generateTabHtml(data.Config)}</div>`
    if (data.code)
      html += `<div class="fields_container" id=${ETabTypes.CODE}>${generateCodeHtml(data.code)}</div>`
    if (data.Interface)
      html += `<div class="fields_container"
id=${ETabTypes.INTERFACE}>${generateTabHtml(data.Interface.data)}</div>`
      if (data.Interface2)
        html += `<div class="fields_container"
id=${ETabTypes.INTERFACE2}>${generateTabHtml(data.Interface2.data)}</div>`

    return html
  }

let isShowingTable = false

const showTable = (event, data) => {
  let elemContainer = document.getElementById('schema')
  let container = document.createElement('div')
  container.id = 'table'
  container.className = 'table'
  container.style.top = '50%'
  container.style.left = '50%'
  container.style.transform = 'translate(-50%, -50%)'

  let name = data.Interface ? data.Interface.name : ""
  let name2 = data.Interface2 ? data.Interface2.name : ""
  container.innerHTML = createHeader(data.tabs, name, name2) + createFields(data)
  elemContainer.appendChild(container)
  document.getElementsByClassName("fields_container")[0].style.display = "block"
  document.getElementsByClassName("header_item")[0].className += " active"

  isShowingTable = true
}

window.addEventListener("load", () => {
  createPipeLine('schema', 160, 38, 75, 155, 370)

```

```
createPipeLine('schema', 130, 32, 125, 175, 550)
createPipeLine('schema', 150, 38, 80, 195, 550)
createPipeLine('schema', 195, 46, 25, 200, 530)
createPipeLine('schema', 595, 140, 3, 200, 518)
createPipeLine('schema', 585, 140, -5, 200, 505)
createPipeLine('schema', 580, 145, -37, 190, 505)
createPipeLine('schema', 480, 120, -48, 180, 505)
createPipeLine('schema', 350, 75, -68, 180, 505)
createSolidLine('schema', 550, -20, 190, 515, 'black')
createSolidLine('schema', 700, -30, 195, 515, 'black')
createSolidLine('schema', 120, 80, 105, 180, 'darkgoldenrod')
createSolidLine('schema', 160, 18, 105, 160, 'darkgoldenrod')
createSolidLine('schema', 360, -2, 105, 160, 'darkgoldenrod')
createSolidLine('schema', 220, 215, 960, 420, 'darkgoldenrod')
createSolidLine('schema', 220, 170, 960, 430, 'darkgoldenrod')
createSolidLine('schema', 220, 150, 960, 445, 'darkgoldenrod')
createSolidLine('schema', 75, 168, 970, 130, 'blue')
createSolidLine('schema', 12, 65, 898, 145, 'blue')
createSolidLine('schema', 75, 168, 903, 157, 'blue')
createCircle('schema', 105, 215)
createCircle('schema', 115, 270)
createCircle('schema', 150, 172)
createCircle('schema', 230, 198)
createCircle('schema', 155, 153)
createCircle('schema', 430, 144)
createCircle('schema', 930, 396)
createCircle('schema', 930, 427)
createCircle('schema', 930, 452)
createCircle('schema', 780, 453)
createCircle('schema', 820, 320)
createCircle('schema', 790, 532)
createTriangle('schema', 180, 780, 190)
createTriangle('schema', 180, 870, 172)
createTriangle('schema', 180, 945, 145)
createTriangle('schema', 180, 240, 500)
createTriangle('schema', 180, 240, 510)
createTriangle('schema', 180, 690, 345)
});
```



```

function openTab(event, type) {
  let i, content, links

  content = document.getElementsByClassName("fields_container")
  for (i = 0; i < content.length; i++) {
    content[i].style.display = "none"
  }
  links = document.getElementsByClassName("header_item")
  for (i = 0; i < links.length; i++) {
    links[i].className = links[i].className.replace(" active", "")
  }
  document.getElementById(type).style.display = "block"

  event.currentTarget.className += " active"
}

document.addEventListener("click", (e) => {
  const table = document.getElementById("table")
  let targetEl = e.target

  if (!isShowingTable) {
    do {
      if (targetEl === table) return

      targetEl = targetEl.parentNode
    } while (targetEl)

    table.remove()
  }
  isShowingTable = false
})

```

data.js

```

const EFieldTypes = {
  BASIC: 'BASIC',
  SELECT: 'SELECT',
  CHECKBOX: 'CHECKBOX',
  RADIO: 'RADIO',
  TEXT: 'TEXT'
}

```

```
}

```

```
const ETabTypes = {
  IO_CONFIG: 'IO_CONFIG',
  CONFIG: 'CONFIG',
  CODE: 'CODE',
  INTERFACE: 'INTERFACE',
  INTERFACE2: 'INTERFACE2'
}
```

```
const DoorFieldsData = {
  tabs: [ETabTypes.IO_CONFIG, ETabTypes.CONFIG, ETabTypes.INTERFACE],
  IOConfig: [{
    type: EFieldTypes.SELECT,
    name: 'Network Adapter',
    value: 'PT-IOT-NM-1W'
  }, {
    type: EFieldTypes.SELECT,
    name: 'Network Adapter 2',
    value: 'None'
  }, {
    type: EFieldTypes.SELECT,
    name: 'Digital Slots',
    value: '1'
  }, {
    type: EFieldTypes.SELECT,
    name: 'Analog Slots',
    value: '0'
  }, {
    type: EFieldTypes.SELECT,
    name: 'USB Ports',
    value: '0'
  }, {
    type: EFieldTypes.CHECKBOX,
    name: 'Bluetooth',
    label: 'Built-in',
    checked: ""
  }, {
    type: EFieldTypes.CHECKBOX,
    name: 'Desktop',

```

```

label: 'Show',
checked: "
}, {
type: EFieldTypes.RADIO,
name: 'Usage',
value1: 'Smart Device',
checked1: 'checked',
value2: 'Component',
checked2: ",
}],
Config: [{
type: EFieldTypes.TEXT,
name: 'Display Name',
value: 'SmartDoor'
}, {
type: EFieldTypes.TEXT,
name: 'Serial Number',
value: 'PTT08101WM9'
}, {
type: EFieldTypes.RADIO,
name: 'Gateway/DNS IPv4',
value1: 'DHCP',
checked1: 'checked',
value2: 'Static',
checked2: ",
}, {
type: EFieldTypes.TEXT,
name: 'Gateway',
value: '192.168.25.1'
}, {
type: EFieldTypes.RADIO,
name: 'Gateway/DNS IPv6',
value1: 'DHCP',
checked1: ",
value2: 'Static',
checked2: 'checked',
}, {
type: EFieldTypes.TEXT,
name: 'IPv6 Gateway',
value: "

```

```

}, {
  type: EFieldTypes.TEXT,
  name: 'IPv6 DNS Server',
  value: ""
}, {
  type: EFieldTypes.RADIO,
  name: 'IoT Server',
  value1: 'Home Gateway',
  checked1: 'checked',
  value2: 'Remote Server',
  checked2: "",
  value3: 'Home Gateway',
  checked3: "",
}, {
  type: EFieldTypes.TEXT,
  name: 'Server Address',
  value: ""
}, {
  type: EFieldTypes.TEXT,
  name: 'User Name',
  value: ""
}, {
  type: EFieldTypes.TEXT,
  name: 'Password',
  value: ""
}],
Interface: {
  name: 'Wireless0',
  data: [{
    type: EFieldTypes.CHECKBOX,
    name: 'Port Status',
    label: 'On',
    checked: 'checked'
  }, {
    type: EFieldTypes.TEXT,
    name: 'Bandwidth',
    value: '300 Mbps'
  }, {
    type: EFieldTypes.TEXT,
    name: 'MAC Address',

```

```

    value: '00D0.971A.50A5'
  }, {
    type: EFieldTypes.TEXT,
    name: 'SSID',
    value: 'HomeGateway'
  }, {
    type: EFieldTypes.RADIO,
    name: 'Authentication',
    value1: 'Disabled',
    checked1: "",
    value2: 'WEP',
    checked2: "",
  }, {
    type: EFieldTypes.TEXT,
    name: 'WEP Key',
    value: ""
  }, {
    type: EFieldTypes.RADIO,
    name: "",
    value1: 'WPA-PSK',
    checked1: "",
    value2: 'WPA2-PSK',
    checked2: 'checked',
  }, {
    type: EFieldTypes.TEXT,
    name: 'PSK Pass Phrase',
    value: 'mySecretKey'
  }, {
    type: EFieldTypes.RADIO,
    name: "",
    value1: 'WPA',
    checked1: "",
    value2: 'WPA2',
    checked2: "",
  }, {
    type: EFieldTypes.TEXT,
    name: 'User ID',
    value: ""
  }, {
    type: EFieldTypes.RADIO,

```

```
name: ",
value1: '802.1X',
checked1: ",
}, {
  type: EFieldTypes.TEXT,
  name: 'Method',
  value: 'MD5'
}, {
  type: EFieldTypes.TEXT,
  name: 'User Name',
  value: ""
}, {
  type: EFieldTypes.TEXT,
  name: 'Password',
  value: ""
}, {
  type: EFieldTypes.SELECT,
  name: 'Encryption Type',
  value: 'AES'
}, {
  type: EFieldTypes.RADIO,
  name: 'IP Configuration',
  value1: 'DHCP',
  checked1: 'checked',
  value2: 'Static',
  checked2: "",
}, {
  type: EFieldTypes.TEXT,
  name: 'IP Address',
  value: '192.168.25.102'
}
```