

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра математичного забезпечення комп'ютерних систем

(повна назва кафедри (предметної, циклової комісії))

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

(освітньо-кваліфікаційний рівень)

Проектування і розробка мережі Інтернет речей

Design and development of the Internet of Things network

Виконав: студент денної форми навчання

спеціальності 123 – Комп'ютерна інженерія.

(шифр і назва напрямку підготовки, спеціальності)

Хомік Микита Володимирович

(прізвище, ім'я, по-батькові)

Керівник к.т.н., доцент Волощук Л.А.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент ст.викл. Лісіцина І.М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ 12 від « 09 » 06 2023 р.

Завідувач кафедри

Євгеній МАЛАХОВ

(підпис)

(ім'я, прізвище)

Захищено на засіданні ЕК № ____

протокол № ____ від « ____ » ____ 2023
р.

Оцінка _____ /
/ _____

(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

Алла КОБОЗЄВА

(підпис)

(ім'я, прізвище)

Одеса – 2023

АНОТАЦІЯ

Мета даної дипломної роботи – розробка IoT мережі автоматичного поливу для присадибної ділянки. Об'єкт складається з семи зон поливу, які треба забезпечити системою автоматичного поливу.

В дипломній роботі досліджені сучасні системи автоматичного поливу. Проведено огляд технологій реалізації IoT мереж автополиву. Спроековано фізичну та логічну топологію територіальної локальної IoT мережі присадибної ділянки. Проведено моделювання і тестування роботи спроектованої IoT мережі в середовищі Packet Tracer компанії Cisco.

Обрано активне та пасивне обладнання для реалізації IoT мережі присадибної ділянки. В якості контролера клапанами та сервера IoT обраний мікрокомп'ютер Raspberry Pi 3+. Для створення пристроїв аналізатора дощу та ґрунту обраний мікроконтролер NodeMCU v3. Обрано протокол MQTT для обміну повідомлень між вузлами IoT мережі. Створено сервер MQTT. Проведена розробка принципових схем IoT вузлів. Розроблений дослідний зразок системи автополиву з можливістю дистанційного керування. Вартість реалізації проекту 15 385 грн.

ABSTRACT

The purpose of this thesis is the development of an IoT network for automatic irrigation for the homestead. The object consists of seven irrigation zones, which must be provided with an automatic irrigation system.

Modern automatic irrigation systems were investigated in the thesis. An overview of the technologies for the implementation of IoT self-irrigation networks was conducted. The physical and logical topology of the territorial local IoT network of the homestead was designed. Modeling and testing of the designed IoT network in Cisco's Packet Tracer environment was carried out.

Active and passive equipment were selected for the implementation of the IoT network of the homestead. A Raspberry Pi 3+ microcomputer was chosen as the valve controller and IoT server. The NodeMCU v3 microcontroller was chosen to create the rain and soil analyzer devices. The MQTT protocol is selected for the exchange of messages between IoT network nodes. MQTT server created. Development of schematic diagrams of IoT nodes was carried out. A pilot model of an automatic watering system with the possibility of remote control has been developed. The cost of project implementation is UAH 15 385 uah.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД І АНАЛІЗ МЕРЕЖІ ІНТЕРНЕТ РЕЧЕЙ ДЛЯ ОРГАНІЗАЦІЇ ОПТИМАЛЬНОГО СЕРЕДОВИЩА ВИРОЩУВАННЯ РОСЛИН ПРИСАДИБНОЇ ДІЛЯНКИ	10
1.1 Поняття і визначення технології Інтернет речей, можливості IoT мережі для віддаленого керування автополивом.....	10
1.2 Огляд технологій підключення елементів мережі інтернет речей автополиву.....	14
1.3 Огляд сучасних IoT мереж автополиву	17
1.4 Постановка завдання:.....	21
2 ПРОЕКТУВАННЯ МЕРЕЖІ ІНТЕРНЕТ РЕЧЕЙ ДЛЯ ОРГАНІЗАЦІЇ ОПТИМАЛЬНОГО СЕРЕДОВИЩА ВИРОЩУВАННЯ РОСЛИН ПРИСАДИБНОЇ ДІЛЯНКИ	23
2.1 Проектування територіального плану розміщення мережі інтернет речей.....	23
2.2 Моделювання роботи IoT мережі в середовищі PacketTracer.	26
2.3 Проектування функцій керування мережею інтернет речей	41
3 РОЗРОБКА МЕРЕЖІ ІНТЕРНЕТ РЕЧЕЙ ДЛЯ ОРГАНІЗАЦІЇ ОПТИМАЛЬНОГО СЕРЕДОВИЩА ВИРОЩУВАННЯ РОСЛИН ПРИСАДИБНОЇ ДІЛЯНКИ	46
3.1 Вибір компонентів для мережі інтернет речей	46
3.2 Розробка принципів схем вузлів мережі інтернет речей	54
3.3 Розробка програмного забезпечення.....	56
3.4 Розробка дослідного зразка та перевірка працездатності.....	61
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67

ДОДАТОК А КОД ПРОГРАМИ КОНТРОЛЕРА У CISCO PACKET TRACER	68
ДОДАТОК Б ФІЗИЧНА ТОПОЛОГІЯ ІОТ МЕРЕЖІ.....	71

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

MQTT – Message Queuing Telemetry Transport.

JSON – JavaScript Objective Notation.

IoT – Internet of Things.

ВСТУП

В добу швидкого технологічного прогресу та зростаючої автоматизації, Інтернет речей (IoT) стає все більш популярним концептом. IoT дозволяє зв'язати фізичні об'єкти з мережею Інтернет і забезпечити взаємодію між ними. У зв'язку з цим виникає безліч можливостей для удосконалення різних аспектів нашого повсякденного життя.

Одним із таких аспектів є автополив, який використовується для забезпечення оптимального зростання рослин, економії води, та повна автоматизація процесу поливу. Але існуючі системи автополиву мають обмежений функціонал. Вони не піддаються модифікаціям і не можуть бути інтегровані в існуючу мережу інтернету речей.

Таким чином, мета даної дипломної роботи проектування та розробка IoT мережі для присадибної ділянки типового будинку. Для досягнення поставленої мети потрібно вирішати наступні завдання:

1. Виконати огляд та аналіз існуючих систем автополиву та типових його елементів.
2. Виконати огляд технологій реалізації IoT мережі автополиву.
3. Провести проектування фізичної та логічної топології територіальної локальної IoT мережі присадибної ділянки.
4. Провести моделювання і тестування роботи спроектованої IoT мережі в середовищі симулятора Packet Tracer компанії Cisco.
5. Провести проектування функцій керування автополивом в IoT мережі присадибної ділянки.
6. Провести вибір активного та пасивного обладнання для реалізації IoT мережі присадибної ділянки.
7. Провести розробку принципів схем підключення IoT вузлів та вибрати системну архітектуру керуючих вузлів.

8. Програмування функцій керування автополивом на присадибній ділянці.
9. Розробити дослідний зразок системи автополиву з можливістю дистанційного керування
10. Розрахувати вартість реалізації IoT мережі.

1 ОГЛЯД І АНАЛІЗ МЕРЕЖІ ІНТЕРНЕТ РЕЧЕЙ ДЛЯ ОРГАНІЗАЦІЇ ОПТИМАЛЬНОГО СЕРЕДОВИЩА ВИРОЩУВАННЯ РОСЛИН ПРИСАДИБНОЇ ДІЛЯНКИ

1.1 Поняття і визначення технології Інтернет речей, можливості IoT мережі для віддаленого керування автополивом.

Технологія Інтернету речей [1] (IoT) – це система з'єднаних пристроїв, які здатні збирати, обмінювати та аналізувати дані за допомогою мережі Інтернет. Іншими словами, IoT передбачає підключення будь-яких речей до Інтернету, що дозволяє їм обмінювати даними із іншими пристроями та системами. На рисунку 1.1 зображена структура IoT системи.

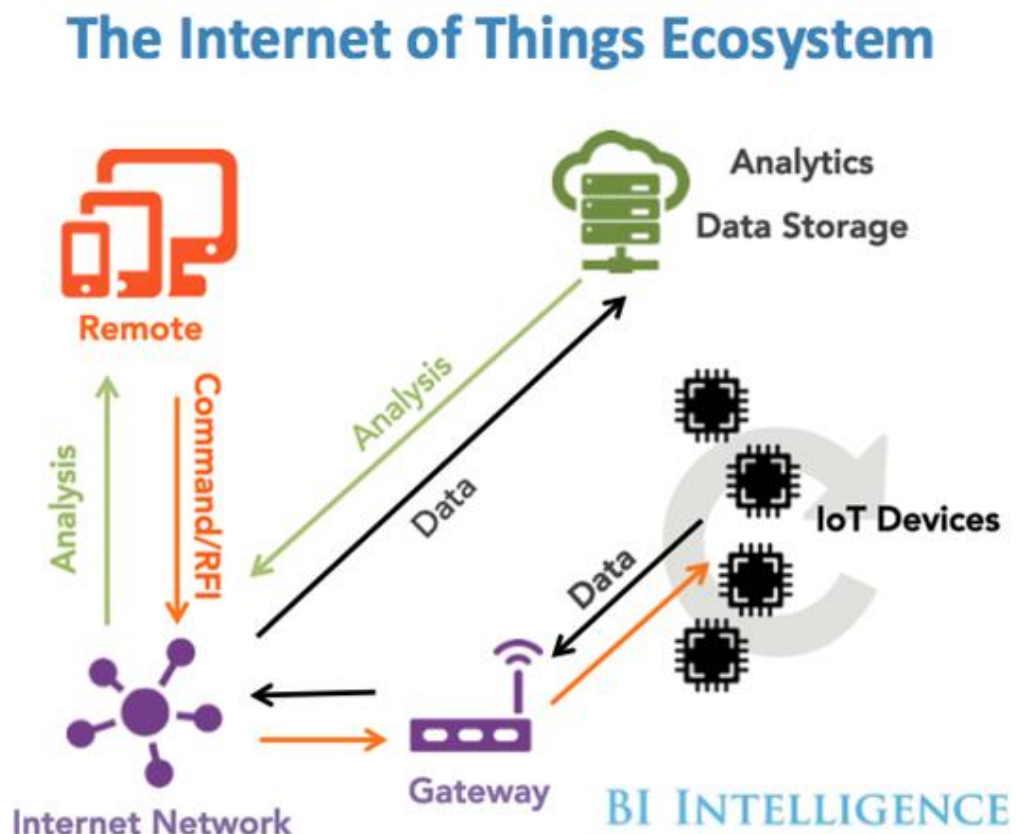


Рисунок 1.1 – Структура IoT системи

Інтернет речей може бути використаний для віддаленого керування автополивом з високою ефективністю. Завдяки датчикам температури, вологості, тиску та іншим параметрам, підключеним до IoT мережі, можна збирати та аналізувати дані про клімат у різних місцях. Ці дані можуть бути передані до централізованої системи керування, яка може віддалено керувати системами автополиву, щоб забезпечити комфортні умови на ділянках для вирощування рослин. Крім того, IoT може допомогти відстежувати та прогнозувати зміни клімату в різних регіонах, що дозволяє більш ефективно планувати та управляти ресурсами.

Автоматизована система управління поливом [2] – це найпростіший і найзручніший спосіб догляду за присадибною ділянкою, що складається з наступних елементів:

- блок керування;
- датчики погодних умов;
- електромагнітні клапани;
- спринклери;

Ключовий елемент обладнання – контролер чи блок керування (рисунк 1.2). Саме він приводить у дію по черзі електромагнітні клапани. Завдяки контролеру зрошення лужка відбувається безперервно і рівномірно. Це найефективніший спосіб поливу.



Рисунок 1.2 – Приклад контролерів для систем автополиву

Система укомплектована таймером, який дозволяє запрограмувати обладнання на включення у певний час (день, годину та хвилину). Крім того, автополив передбачає можливість використання сезонного регулювання поливу за умови, що контролер оснащений опцією дистанційної зміни режиму.

Датчики дощу (рисунок 1.3) застосовують для економії споживаної води в зрошувальних системах. Головне призначення – відключати систему автоматичного поливу, якщо на вулиці йдуть дощі.



Рисунок 1.3 – Вигляд датчику дощу

Блок керування поливом контролює роботу електромагнітних клапанів (рисунок 1.4), які відповідають за функціонування дощувачів. До кожного клапана «прив'язано» кілька спринклерів, що обертаються, здійснюють зрошення газону. Роторні елементи можуть мати різний розмір, працювати з різною потужністю – їх вибір визначається параметрами забору води, площею зрошуваної території, рельєфом газону, а також властивостями ґрунту, глибиною ґрунтових вод.



Рисунок 1.4 – Вигляд магнітних клапанів

Безпосередньо до рослин вода надходить із спринклерів, здатних виконувати зрошення різного рівня складності. Завдяки різноманітності використовуваних насадок для дощів, можна обробляти як маленьку клумбу, так і сад. Основні два види спринклерів це дощувачі та форсунки. Вони бувають ротаційні чи ротатори, віялові чи щілинні, баблери для прикореневого поливу (малий радіус поливу). Ротатори та віялові форсунки відрізняються сектором та радіусом поливу. Спринклери зображені на рисунку 1.5.



Рисунок 1.5 – Вигляд спринклерів

Сучасні технології досягли такого рівня, що сьогодні ви можете керувати зрошенням своєї ділянки зі свого комп'ютера, планшета та навіть з мобільного телефону. Крім цього, система автополива має такі переваги:

- 1) Обладнання працює автономно, йому не потрібна обов'язкова участь людини;
- 2) Ви заощаджуєте особистий час та гроші на оплату роботи садівника;
- 3) Системою легко керувати;
- 4) Обладнання відрізняється надійністю та довговічністю;
- 5) Рослини отримують належний догляд, що є запорукою гарного ландшафту;

1.2 Огляд технологій підключення елементів мережі інтернет речей автополиву

Існує набір стандартів і правил, які використовуються для забезпечення комунікації [3] та обміну даними між різними пристроями в мережі Інтернет Речей. Ці технології визначають формати даних, способи передачі та обробки інформації, а також забезпечують безпеку та конфіденційність даних.

Wi-Fi: Wi-Fi є найпоширенішою технологією підключення до IoT мереж. Вона забезпечує швидку передачу даних та дозволяє віддалено керувати системою автополиву з будь-якої точки з Wi-Fi підключенням.

Bluetooth: Bluetooth - це технологія бездротового зв'язку на короткій відстані. Вона дозволяє підключати до 30 пристроїв до одного контролера. Використання Bluetooth для системи автополиву може бути зручним, якщо планується керування з невеликої відстані.

Zigbee: Zigbee є технологією бездротової мережі, яка спеціально розроблена для IoT-пристроїв. Вона пропонує низьке споживання енергії та

широкий діапазон покриття. Zigbee дозволяє підключати до 65 000 пристроїв до одного контролера.

LoRaWan: LoRWana є дуже енергоефективною технологією, яка дозволяє передавати дані на великі відстані. Вона використовується для зв'язку віддалених датчиків, які вимагають довгого життєвого циклу батареї та низької пропускну здатності.

NB-IoT: Narrowband IoT (NB-IoT) є технологією мережі, яка спеціально розроблена для IoT-пристроїв. Вона пропонує низьке споживання енергії та дозволяє передавати дані на великі відстані. NB-IoT забезпечує зручний і безпечний зв'язок з мережім оператором мобільного зв'язку.

Z-Wave – бездротова технологія з низьким енергоспоживанням, яка працює на частоті до 1 ГГц. Вона призначена для передавання простих команд управління з малими затримками. Ця технологія дозволяє мати повний контроль над домашньою безпекою, енергією та автоматизацією. Вона є простою у використанні, енергозберігаючою та економить час. Система працює з дистанційного керування та використовує радіохвилі низької потужності, що покривають всі області будинку. Z-Wave використовує частоту 908,42 МГц в США і 868,42 МГц в Європі. В порівнянні з іншими технологіями, вона проста у використанні та доступна за ціною. Порівняння технологій підключення елементів IoT мережі наведені у таблиці 1.2.

Таблиця 1.2 – Порівняння технологій підключення.

Технологія	LoRaWan	NB-IoT	BLE	Wi-Fi	Z-Wave	ZigBee
Стандарт зв'язку	Свій	3GPP	Bluetooth SIG	802.11	Свій	IEEE 802.15.4
Частота	433 МГц, 868МГц, 915 МГц	700- МГц-2,2 ГГц	2,4 ГГц	2,4/5,0 ггц	868-926 МГц	2.4 ГГц

Продовження таблиці 1.2

Швидкість передачі даних	50-290 біт/с	200 Кбіт/с	125 Кбіт/с-2 Мбіт/с	До 150 Мбіт/с	100 Кбіт/с	250 Кбіт/с
Топологія мережі	Зірка	Зірка	P2P, broad	Зірка	Mesg	Mesh
Кількість учасників мережі	1000+	1000+	20	100	232	250+
Дальність	5-15 км	5 км	40-840 м	20-100м	40-100 м	40-100 м

1.3 Огляд сучасних IoT мереж автополиву

На ринку вже давно існують рішення для автополиву, які здатні без втручання людини управляти своєчасним поливом рослин, або давати можливість управляти поливом рослин з будь-якої точки світу.

У ході виконання роботи було проведено огляд існуючих на ринку України систем поливу рослин. Найпопулярнішим рішенням на сьогоднішній день мають такі компанії:

- Hunter;
- Rain-bird.

Порівняння було зроблено за такими параметрами:

- кількість зон для поливу,
- кількість програм які можуть працювати одночасно,
- віддалене управління,
- максимальна тривалість поливу.

Також розглянуті конкретні особливості кожного контролера. Для порівняння було обрано рішення від компаній Hunter та Rain-Bird, та їх контролери в одному ціновому діапазоні 7500 грн.

Таблиця 1.1 – Характеристика контролерів автополиву

Назва моделі контролера	Параметри			
	Кількість доступних зон	Кількість одночасних програм	Підключення wi-fi	ПО для доступу до хмари
SP-RZX-6i	4	2	Так (Треба придбати додатковий модуль)	Так (Безкоштовне)
X2-801-e	8	1	Так (Треба придбати додатковий модуль)	Так (Платне)

Розглянемо контролер X2-801-e [4] (рисунок 1.6) від компанії Hunter.



Рисунок 1.6 – Контролер Hunter X2-801-e

Особливості для X2-801: Автономне налаштування режимів та графіків всіх зон і кожної окремо (тривалість та запуск/ закінчення поливу). Шість незалежних робочих режимів у кожному сегменті території. Чотири опції у графіках зрошення (дні тижня, інтервали циклів, необхідні періоди) Можливість запуску всіх або окремої зони вручну. Миттєве налаштування програми за допомогою інформації, що зберігається у пам'яті «за замовчуванням». За необхідності ігнорування датчика дощу для різних ділянок. Контролер має можливість встановити Wi-Fi модуль WAND (рисунок 1.7), на жаль, який треба купляти окремо.



Рисунок 1.7 – Вигляд Wi-Fi модуля WAND для контролеру Hunter

Модуль дозволяє за допомогою програмного забезпечення управляти автополивом через пристрої типу Android, IOS та WEB. Найголовніше, що модуль дає можливість користуватись хмарною платформою Hydrowise [5]. При поєднанні системи з Hydrowise, користувач отримає можливість управляти автополивом через мережу Інтернет з будь якої точки світу. Для цього треба лише щоб контролер мав постійний доступ до мережі інтернет за допомогою WAND.

Розглянемо тепер Rain-Bird SP-RZX-6i (рисунок 1.8)



Рисунок 1.8 – Вигляд контролеру Rain-Bird SP-RZX-6i

Особливості контролеру: Функція Contractor Default дозволяє легко зберігати та відновлювати ваш графік поливу. Припинення поливу до 14 днів та його автоматичне відновлення після закінчення заданого періоду часу. Можливість обходу датчиком дощу будь-якої зони дозволяє налаштувати те, як кожна із зон повинна реагувати на показання датчика. Можливість програмування постійних днів без поливу гарантує, що полив ніколи не включиться, коли на об'єкті працює обслуговуючий персонал (для графіків поливу по парних та непарних днях та періодичного поливу). Налаштування "Seasonal Adjust" (Сезонне налаштування) для кожної програми дозволяє з легкістю зменшувати або збільшувати інтенсивність поливу.

Rain-bird також надає можливість дистанційного керування автополивом за допомогою модуля LNK-WIFI [7] (рисунок 1.9).



Рисунок 1.9 – Вигляд модуля модуля LNK-WIFI

На відміну від контролеру Hunter, Rain-Bird не має безкоштовного хмарного ПЗ, і це є головним мінусом для цієї системи. Користувач не зможе «з коробки» керувати автополивом зі будь-якої точки світу.

1.4 Постановка завдання:

Розробити IoT мережу для реального об'єкта, який являє собою будинок із земельною ділянкою. Об'єкт розташований у одеській області, населений пункт Вигода. Вигляд об'єкта зображен на рисунку 1.10.



Рисунок 1.10 – Супутниковий знімок об'єкта

Об'єкт поділено на 7 зон поливу. Розміщення ділянок зображено на рисунку 1.10.

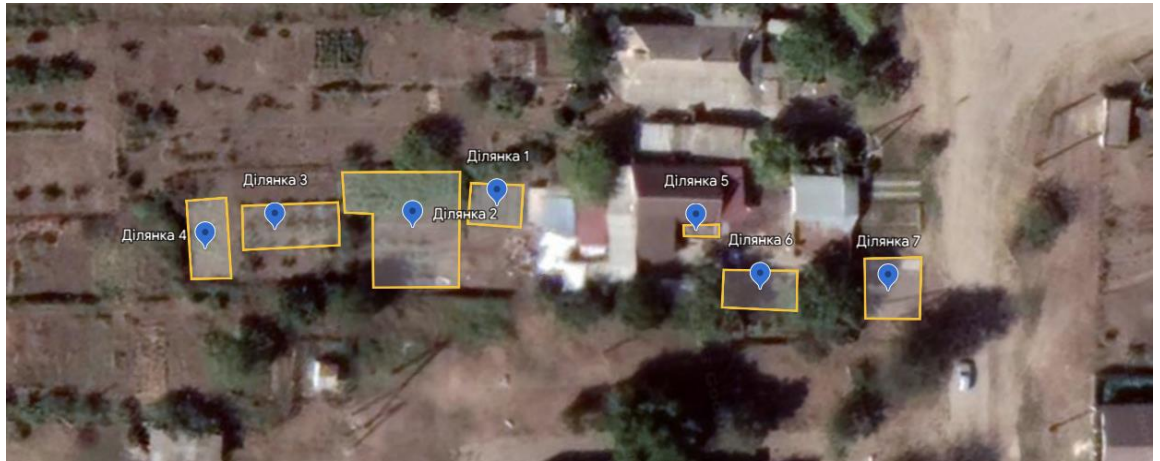


Рисунок 2.2 – розміщення ділянок об'єкта

Кожна зона являє собою окрему земельну ділянку, яку і буде обслуговувати мережа автополиву. Кожна ділянка повинна мати свій аналізатор ґрунту, для вимірювання значення вологості. Розробити моніторинг за погодою, а саме слідкування за температурою та дощем. Розробити функції поливу для ділянок.

2 ПРОЕКТУВАННЯ МЕРЕЖІ ІНТЕРНЕТ РЕЧЕЙ ДЛЯ ОРГАНІЗАЦІЇ ОПИТИМАЛЬНОГО СЕРЕДОВИЩА ВИРОЩУВАННЯ РОСЛИН ПРИСАДИБНОЇ ДІЛЯНКИ

2.1 Проектування територіального плану розміщення мережі інтернет речей

Для об'єкта спроектована структурно-логічно топологія IoT мережі, яка зображена на рисунку 2.1.

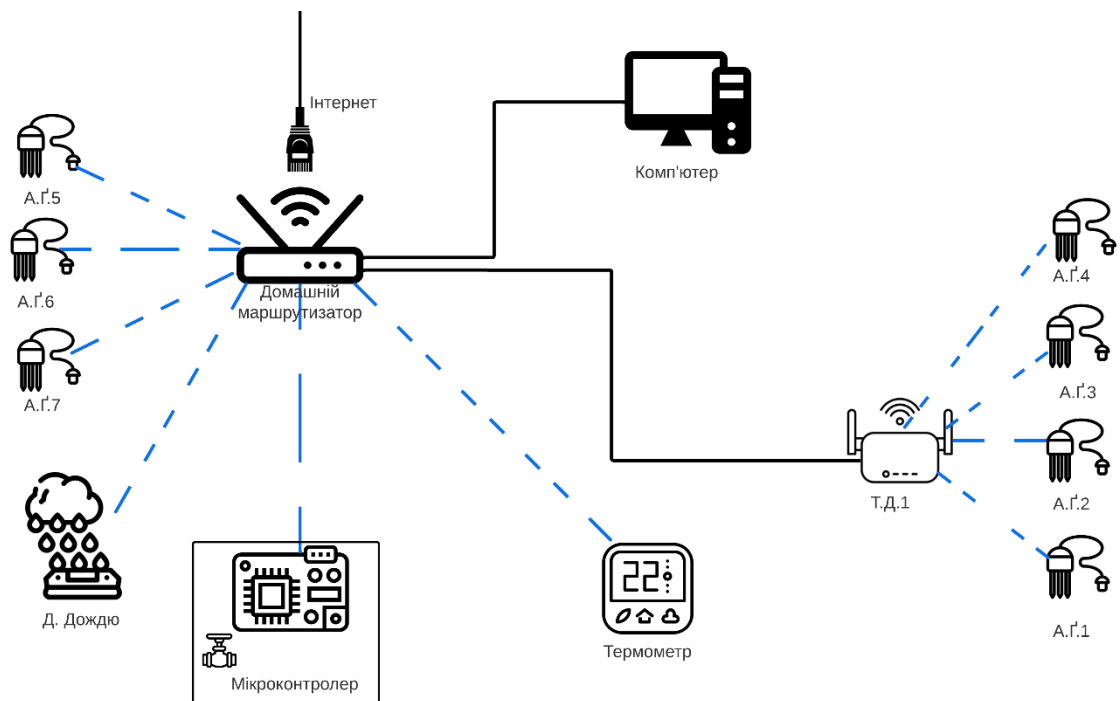
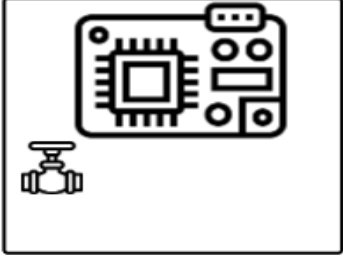


Рисунок 2.1 – Структурно-логічна топологія мережі інтернет речей

Опис елементів топології наведений у таблиці 2.1

Таблиця 2.1 – Опис елементів мережі.

Зображення елемента	Опис
	Домашній маршрутизатор. Представляє собою комбінований пристрій комутатора, маршрутизатора та точки доступу.
	Контролер поливу. Представляє собою мікроконтролер та сім магнітних електроклапанів. Приєднується по WiFi мережі до домашнього маршрутизатора.
	Кабель, який здійснює підключення мережі IoT до глобальної мережі Ethernet .
	Комп'ютер. Стаціонарний комп'ютер, з'єднується з маршрутизатором по ethernet протоколу(кабелю).
	Точка доступу. Станція, призначена для забезпечення бездротового доступу до вже існуючої мережі. З'єднується з маршрутизатором по ethernet протоколу(кабелю).

Продовження таблиці 2.1

	Термометр. Використовується для вимірювання температури та передачі цих даних для подальшого аналізу або керування. Приєднується по WiFi мережі до домашнього маршрутизатора.
	Аналізатор ґрунту. Використовується для вимірювання вологості ґрунту на ділянці для подальшого аналізу або керування. Приєднується по WiFi мережі до домашнього маршрутизатора та точки доступу.
	Датчик дощу. Використовується для виявлення наявності дощу та передачі цієї інформації через мережу для подальшого аналізу або керування. Приєднується по WiFi мережі до домашнього маршрутизатора.
	Графічне зображення бездротового WiFi з'єднання.
	Графічне зображення дротового Ethernet з'єднання.

Головний вузол IoT мережі – домашній маршрутизатор. Він керує комунікацією між пристроями, він розташований у будинку, там він забезпечує оптимальне покриття об'єкта.

В будинку розташований і комп'ютер, через який можливо здійснювати моніторинг та керування системи автополиву.

Аналізатори ґрунту розташовані в кожній зоні на відповідних ділянках. Їх місцезнаходження обрано таким чином, щоб вони ефективно могли вимірювати рівень вологості ґрунту та передавати цю інформацію до головного вузла.

Для забезпечення зв'язку між пристроями, обрана бездротова технологія Wi-Fi. Для ділянок 1, 2, 3, 4 встановлена додаткова точка доступу, оскільки ділянки розташовані віддалено від домашнього маршрутизатора. Точка доступу з'єднана з маршрутизатором по кабелю ethernet.

Датчик дощу та термометр розташовані на відкритій площі, що забезпечить їм точне вимірювання стану клімату.

Контролер поливу розташований поблизу резервуару з водою. Усі клапани знаходяться під мікроконтролером, який забезпечує автоматичне управління поливом залежно від вимірів аналізаторів ґрунту та стану погоди.

Фізична топологія IoT мережі автополиву присадибної ділянки зображена на додатку Б.

2.2 Моделювання роботи IoT мережі в середовищі PacketTracer.

Packet Tracer — це кросплатформний інструмент візуального моделювання, розроблений Cisco Systems, який дозволяє користувачам створювати мережеві топології та імітувати сучасні комп'ютерні мережі, та мережі інтернет речей. Відповідно до територіального плану розміщення мережі, побудуємо модель у середовищі PacketTracer. Побудована модель зображена на рисунку 2.2:

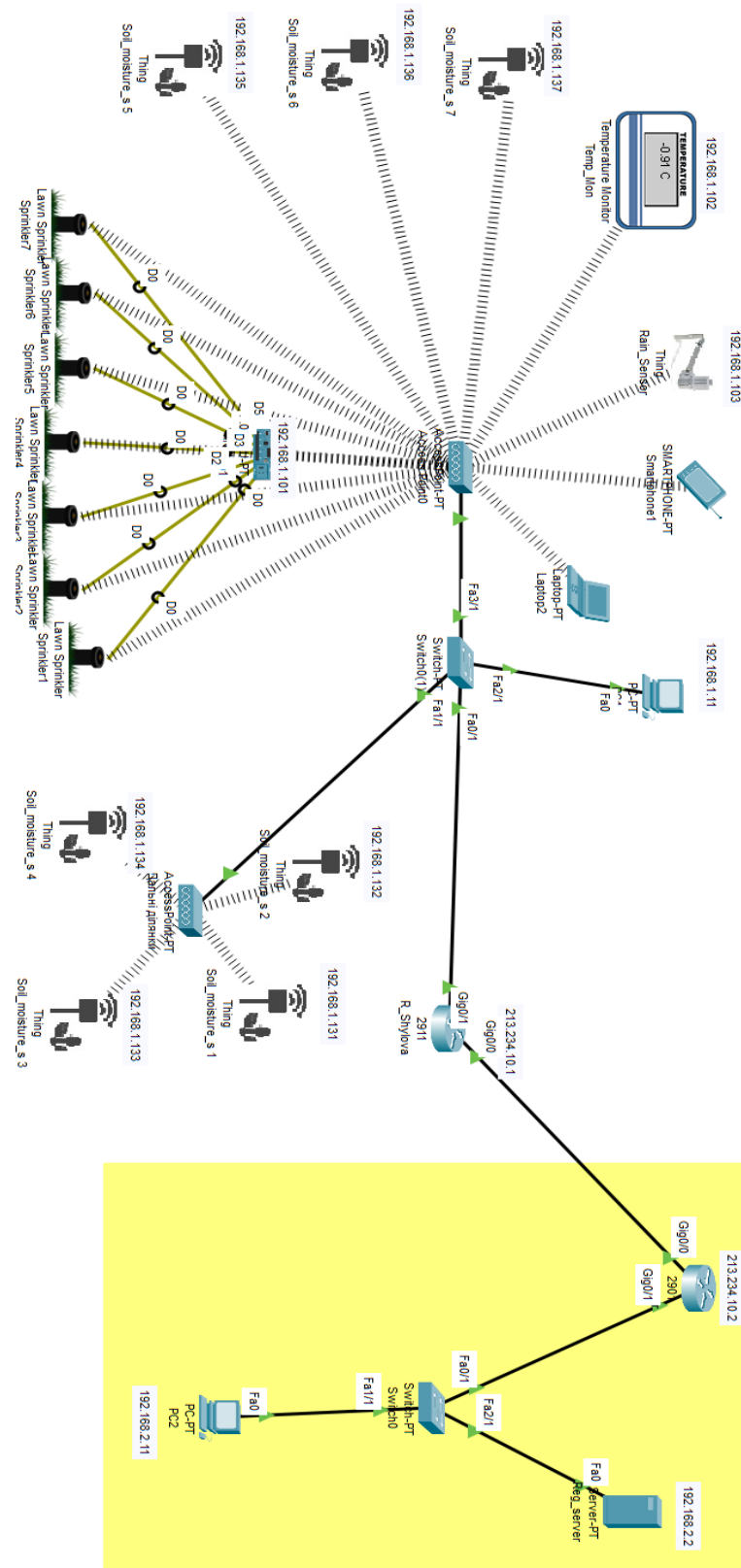


Рисунок 2.2 – Модель мережі інтернет речей для автополиву
присадибної ділянки

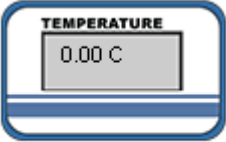
Треба зазначити, що для зображення домашнього маршрутизатора, який являю собою один комбінований пристрій, використано три різні елементи: комутатор, точка доступу та маршрутизатор.

Усі розміщені пристрої та їх опис наведені у таблиці 2.2

Таблиця 2.2– Графічне роз’яснення

Графічне зображення пристрою	Опис
	Маршрутизатор (Router). Одна з частин домашнього маршрутизатора. Маршрутизатор є мережевим пристроєм, який використовується для пересилання пакетів даних між локальною мережею та мережею інтернет.
	Комутатор (Switch). Одна з частин домашнього маршрутизатора. Це мережевий пристрій, який використовується для з'єднання пристроїв в локальній мережі передачі даних між ними.
	Персональний комп'ютер.
	Контролер. Пристрій, який використовується для управління клапанами автополиву.
	Точка доступу (Access Point). Мережевий пристрій, який дозволяє підключати бездротові пристрої до проводової мережі з'єднання для пристроїв пристрої з підтримкою Wi-Fi.

Продовження таблиці 2.2

 SMARTPHONE-PT	Смартфон
 Lawn Sprinkler	Спринклер. Управляється контролером.
 Temperature Monitor	Термометр. Елемент мережі інтернет речем. Вимірює температуру повітря.
 Thing Soil_moisture_s 1	Аналізатор ґрунту. Елемент мережі інтернет речей. Вимірює вологість ґрунту.
 Thing RainSensor	Датчик дощу. Елемент мережі інтернет речей. Вимірює рівень вологості у повітрі.
 Server-PT Registration	Сервер. Сервер потрібен для реєстрації пристроїв у мережі інтернет речей та для їх керування.

Кожному інтернет пристрою у мережі треба надати IP адреса. IP-адреса (Internet Protocol Address) надаються елементам мережі для їхньої ідентифікації та забезпечення комунікації в Інтернеті або локальній мережі. У даній мережі використовуються тільки статичні адреса.

Для того, щоб надати пристрою IP-адрес, треба виконати наступні кроки:

- 1) Клацнути правою кнопкою миші на пристрої, якому хочемо надати IP-адрес, і вибрати "Configure" (Налаштувати) або "CLI" (командний рядок) для відкриття конфігураційного інтерфейсу пристрою.

2) Якщо ми використовуємо маршрутизатор, тоді треба перейти до режиму конфігурації. В командному рядку ввести команду "enable" для входу в режим привілейованого доступу, а потім команду "configure terminal" для входу в режим конфігурації.

3) Ввести команду "interface <ім'я інтерфейсу>", де <ім'я інтерфейсу> - це ім'я інтерфейсу, до якого треба надати IP-адрес. Наприклад, "interface FastEthernet0/0" або "interface GigabitEthernet0/0".

4) Після вибору інтерфейсу ввести команду "ip address <IP-адрес> <Маска підмережі>", де <IP-адрес> - це бажаний IP-адрес для пристрою, а <Маска підмережі> - це маска підмережі, яку треба використовувати. Наприклад, "ip address 192.168.1.1 255.255.255.0".

5) Після всього треба зберегти зміни, ввести команду "exit" для виходу з режиму конфігурації і перевірити налаштування, використовуючи команду "show running-config" або "show ip interface brief".

Присвоювати адресу пристроям можна також через графічний інтерфейс. Наприклад, на рисунку 2.3 зображено надання Ір адреса комп'ютеру PC1 у мережі.

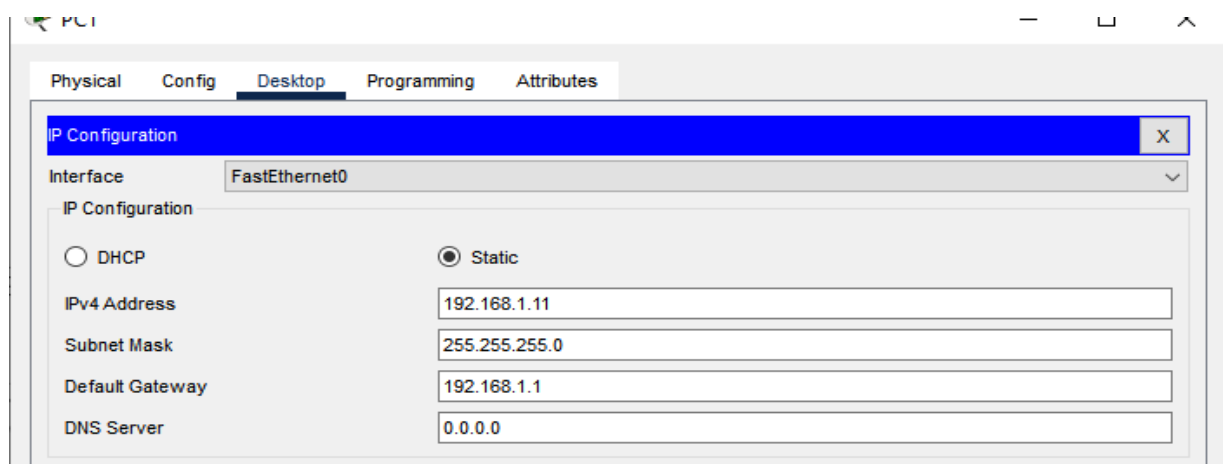


Рисунок 2.3 – Надання Ір адреса комп'ютеру PC

Таблиця адресації вузлів IoT мережі наведена у таблиці 2.3.

Таблиця 2.3 – Адресації вузлів IoT мережі

Назва пристрою	IP-адреса	Маска підмережі	Адрес шлюза
Мережа об'єкта			
PC_Sh	192.168.1.11	255.255.255.0	192.168.1.1
Laptop_Sh	192.168.1.13	255.255.255.0	192.168.1.1
Smарphone 1	192.168.1.12	255.255.255.0	192.168.1.1
MCU_W	192.168.1.101	255.255.255.0	192.168.1.1
Temp_Mon	192.168.1.102	255.255.255.0	192.168.1.1
Rain_Sensor	192.168.1.103	255.255.255.0	192.168.1.1
Soil_moist_s 1	192.168.1.131	255.255.255.0	192.168.1.1
Soil_moist_s 2	192.168.1.132	255.255.255.0	192.168.1.1
Soil_moist_s 3	192.168.1.133	255.255.255.0	192.168.1.1
Soil_moist_s 4	192.168.1.134	255.255.255.0	192.168.1.1
Soil_moist_s 5	192.168.1.135	255.255.255.0	192.168.1.1
Soil_moist_s 6	192.168.1.136	255.255.255.0	192.168.1.1
Soil_moist_s 7	192.168.1.137	255.255.255.0	192.168.1.1
Sprinkler 1	192.168.1.141	255.255.255.0	192.168.1.1
Sprinkler 2	192.168.1.142	255.255.255.0	192.168.1.1
Sprinkler 3	192.168.1.143	255.255.255.0	192.168.1.1
Sprinkler 4	192.168.1.144	255.255.255.0	192.168.1.1
Sprinkler 5	192.168.1.145	255.255.255.0	192.168.1.1
Sprinkler 6	192.168.1.146	255.255.255.0	192.168.1.1
Sprinkler 7	192.168.1.147	255.255.255.0	192.168.1.1
Shylova_R			
Gigabit 0/0	213.234.10.1	255.255.255.0	

Продовження таблиці 2.3

Gigabit 0/1	192.168.1.1	255.255.255.0	
Provider_R			
Gigabit 0/0	213.234.10.2	255.255.255.0	
Gigabit 0/1	192.168.2.1	255.255.255.0	
Мережа інтернет			
Reg_server	192.168.2.2	255.255.255.0	192.168.2.1
PC2	192.168.2.11	255.255.255.0	192.168.2.1
SmartPhone	192.168.2.12	255.255.255.0	192.168.2.1

Для доступу мережі у інтернет, треба налагодити NAT між мережами ділянки та інтернет-провайдером.

NAT (Network Address Translation) є технологією, яка дозволяє перетворювати IP-адреси та порти між різними мережами. Це корисний механізм для забезпечення доступу до Інтернету та комунікації між приватною внутрішньою мережею і глобальною мережею.

Основна функція NAT полягає в перетворенні приватних IP-адрес на публічні IP-адреси. Це дозволяє приватним мережевим пристроям комунікувати з глобальною мережею, використовуючи єдину публічну IP-адресу. NAT налаштовується між двома, це Shylova_R (маршрутизатор об'єкта) та Provider_R (маршрутизатор провайдера) Щоб налаштувати NAT треба зробити такі дії:

Налаштувати інтерфейс, що з'єднує роутер Shylova_r з локальною мережею, та інтерфейс що з'єднує роутер Shylova_r з Інтернетом. Код налаштування зображений на рисунку 2.4.

```
R(config)# interface g 0/1
R(config-if)# ip nat inside
R(config)# interface g 0/0
R(config-if)# ip nat outside
```

Рисунок 2.4 – Налаштування інтерфейсів

Визначити список доступу (ACL), який визначає, які IP-адреси будуть перетворюватись NAT (рисунок 2.5):

```
ip access-list standard FOR-NAT
permit 192.168.1.0 0.0.0.255
```

Рисунок 2.5 – Визначення списку доступу

Ввести команду, яка налаштовує перетворення адрес мережевого шлюзу (NAT) з використанням ACL "FOR-NAT" на інтерфейсі g 0/0 (рисунок 2.6):

```
ip nat inside source list FOR-NAT int g 0/0 overload
```

Рисунок 2.6 – Перетворення адрес мережевого шлюзу

Після треба зареєструвати кожен розумний пристрій на IoT сервері для віддаленого управління мережею інтернет речей. Для цього треба виконати наступні кроки:

1) у веб-браузері увійти на сторінку серверу, ввівши адресу сервера 192.168.2.2, після чого відкриється вікно входу користувача (рисунок 2.7):

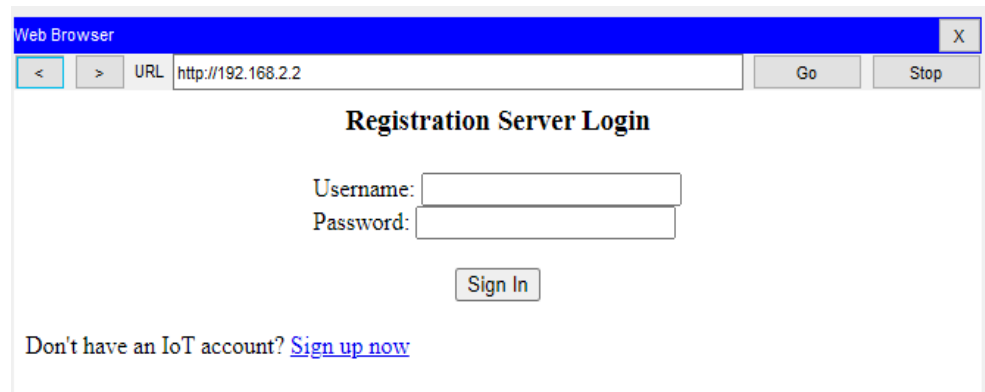


Рисунок 2.7 – Вікно реєстрації користувача

- 2) зареєструвати нового користувача з логіном «nekeka» та паролем «1234»
- 3) Підключити необхідні пристрої до IoT серверу (рисунок 2.8).

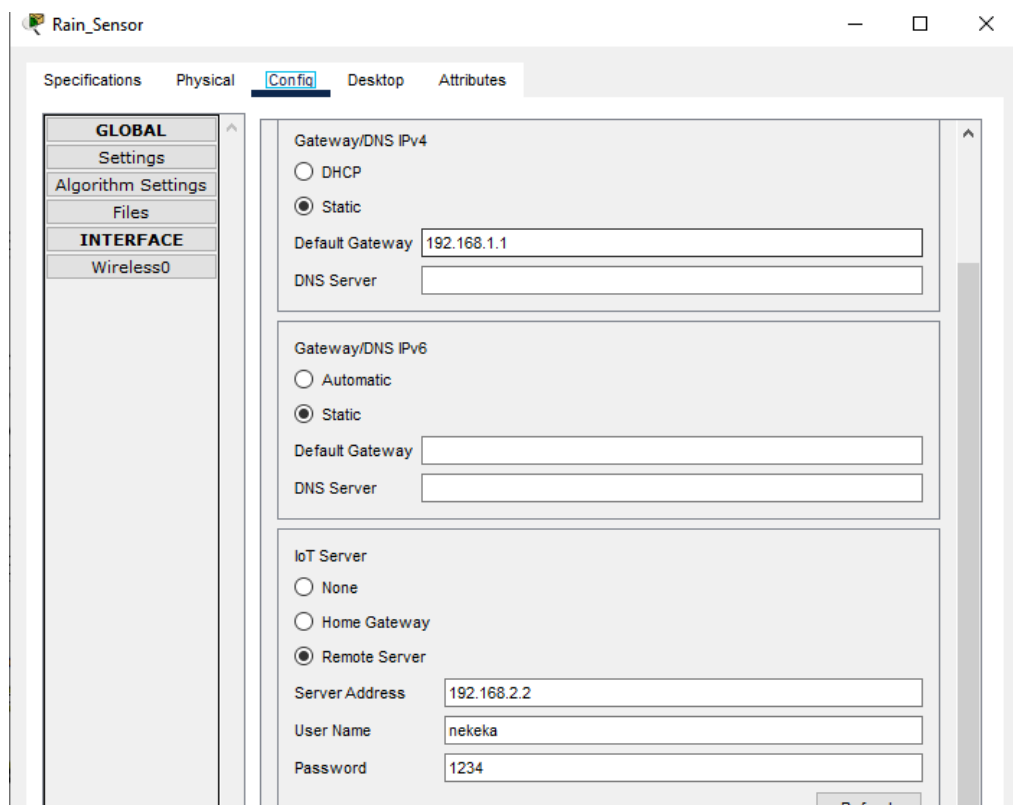


Рисунок 2.8 – Підключення пристрою до серверу IoT

Через застосунок IoT Monitor на PC1 можна здійснювати управління мережею. Усі підключені пристрої до IoT серверу наведені на 2.9.

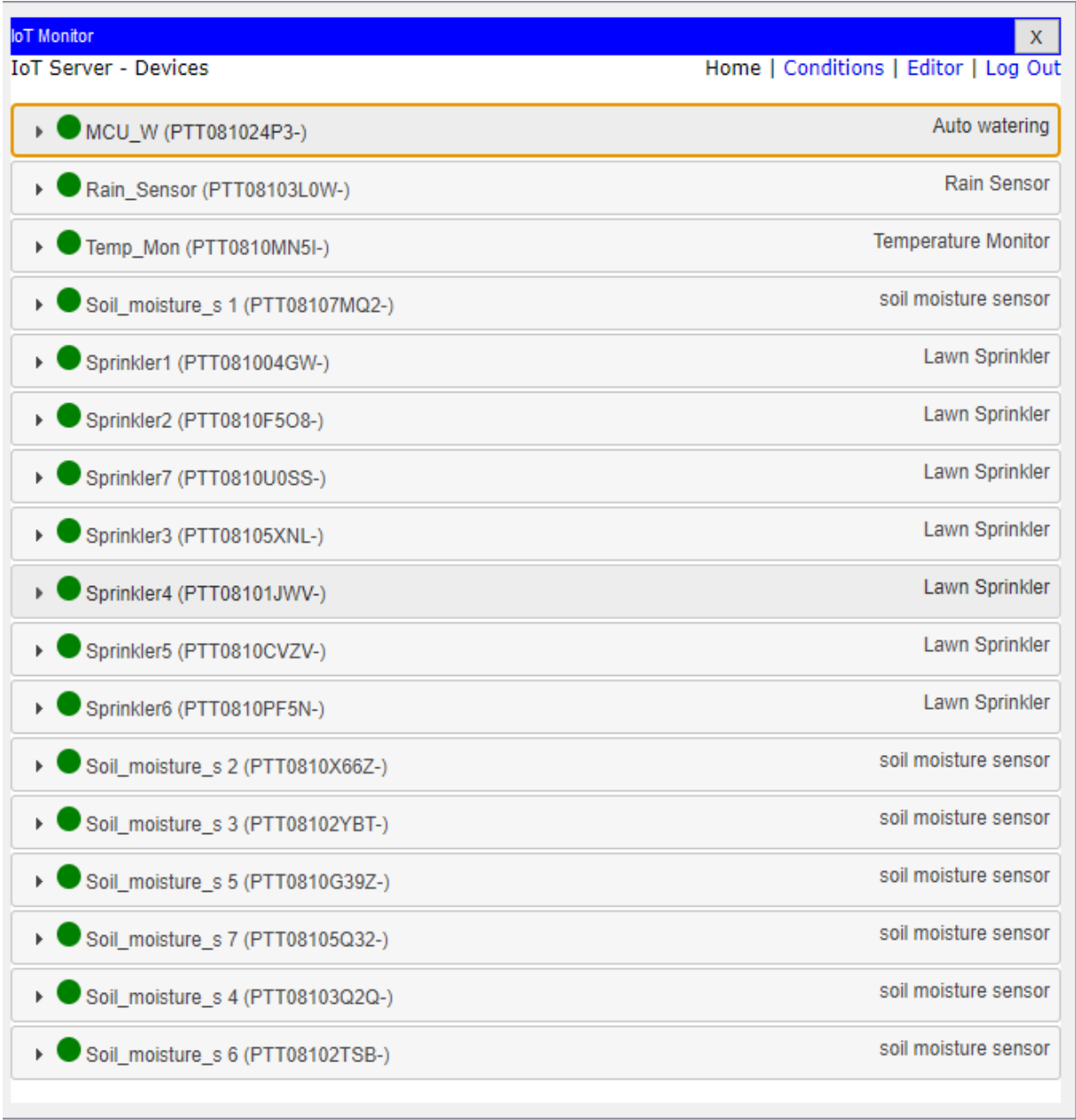


Рисунок 2.9 – Усі підключені пристрої до мережі IoT

Для датчика дощу написана [8] програма, яка відображає на сервері інформацію про те, чи йде зараз дощ (рисунок 2.10).

```
var ENVIRONMENT_NAME = "Rain";
var value;
var boolVar = false;
function setup()
{
    IoEClient.setup({
        type: "Rain Sensor",
        states: [{
            name: "rain",
            type: "bool",
        }]
    });
}
function loop() {
    value = Environment.get(ENVIRONMENT_NAME);
    if (value > 0.1) {
        digitalWrite(0, HIGH);
        console.log("rain");
        boolVar = true;
        IoEClient.reportStates(boolVar);
    }
    else {
        digitalWrite(0, LOW);
        console.log("no rain");
        boolVar = false;
        IoEClient.reportStates(boolVar);
    }
    delay(1000);
}
```

Рисунок 2.10 – Код датчика дощу Rain_Sensor

Цей код описує роботу датчика, а саме, якщо йде дощ, то Rain_Sensor відправляє сигнал false, а якщо ні, то сигнал true.

Щоб перевірити роботу датчика, у моделюванні навколишнього середовища (Environments) налаштували погодні умови так, що дощ буде йти цілодобово. Налаштування погодних умов зображено на рисунку 2.11.

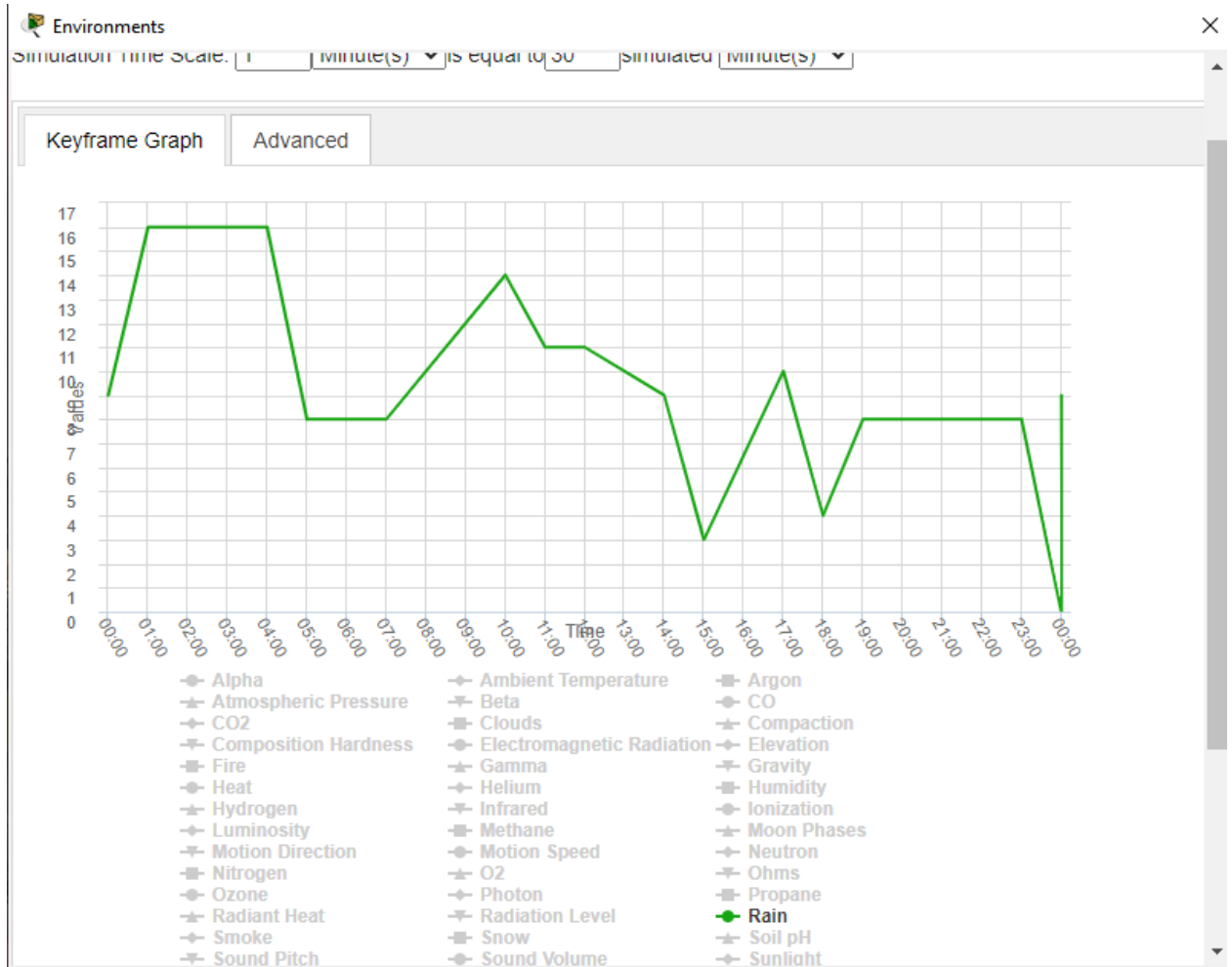


Рисунок 2.11 – Налаштування погодних умов на дощ.

На 2.12 зображено реакцію датчика на дощ. Він надіслав сигнал true, а на застосунку це позначається у виді зеленого кружечка.

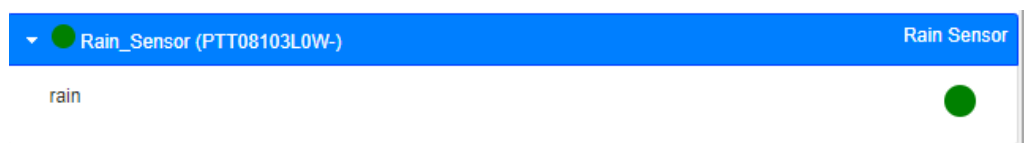


Рисунок 2.12 – Сигнал true від Rain_Sensor

Якщо дощу не буде, то на застосунку це позначиться у вигляді красного кружечка (рисунок 2.13).



Рисунок 2.13 – Сигнал false від Rain_Sensor

Датчик вологості ґрунту Soil_moisture_s відправляє на сервер рівень вологості у відсотках (рисунок 2.14). Рівень вологості підвищується коли йду дощ, або коли працюють спринклери.

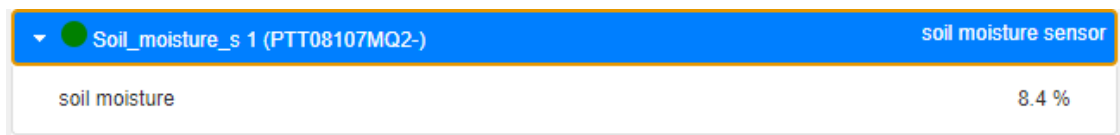


Рисунок 2.14 – Soil_moisture_s

Сам датчик дощу Rain_Sensor та датчик вологості ґрунту Soil_moisture_s потрібен не тільки для моніторингу дощу, а й для регулювання роботи спринклерів. На рисунку 2.15 зображено встановлення правил для роботи спринклеру першої ділянки.

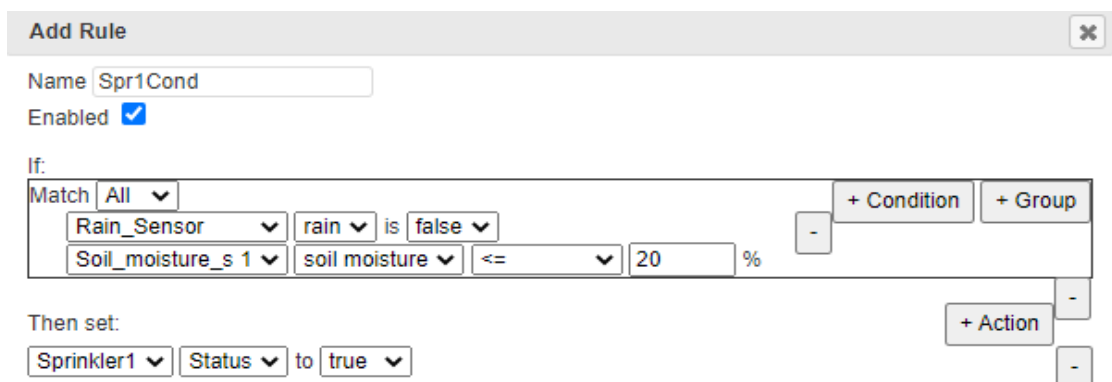
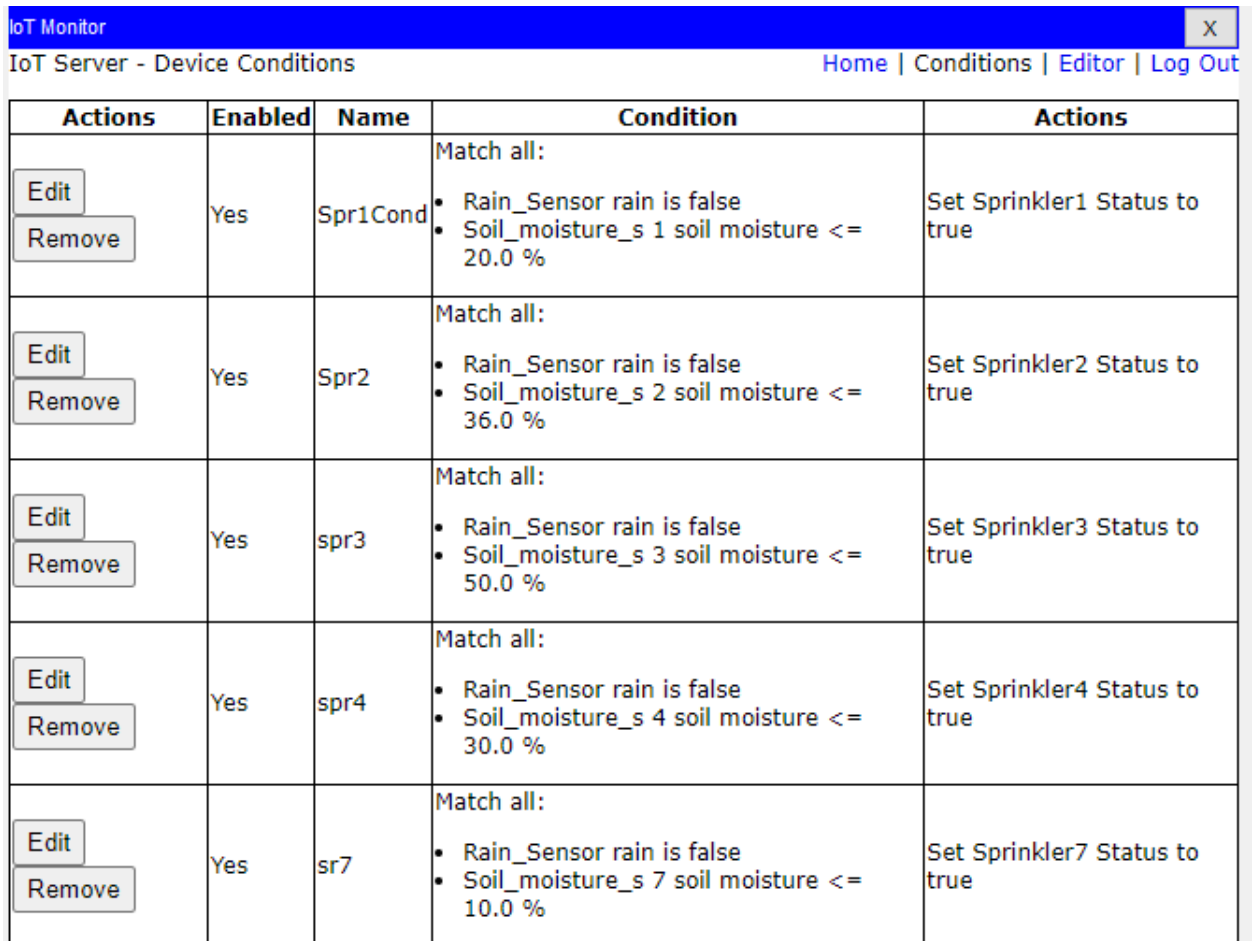


Рисунок 2.15 – Створення правил роботи для першого спринклера

Це правило означає, що спринклер Sprinkler 1 буде активований лише тоді, коли немає дощу та відсоток вологості ґрунту на цій ділянці менше 20%. Встановлені правила зображені на рисунку 2.16.



Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	Spr1Cond	Match all: - Rain_Sensor rain is false - Soil_moisture_s 1 soil moisture <= 20.0 %	Set Sprinkler1 Status to true
Edit Remove	Yes	Spr2	Match all: - Rain_Sensor rain is false - Soil_moisture_s 2 soil moisture <= 36.0 %	Set Sprinkler2 Status to true
Edit Remove	Yes	spr3	Match all: - Rain_Sensor rain is false - Soil_moisture_s 3 soil moisture <= 50.0 %	Set Sprinkler3 Status to true
Edit Remove	Yes	spr4	Match all: - Rain_Sensor rain is false - Soil_moisture_s 4 soil moisture <= 30.0 %	Set Sprinkler4 Status to true
Edit Remove	Yes	sr7	Match all: - Rain_Sensor rain is false - Soil_moisture_s 7 soil moisture <= 10.0 %	Set Sprinkler7 Status to true

Рисунок 2.16 – Встановлені правила увімкнення спринклерів

Для мережі також створена програма автополиву за допомогою контролера. Через програмування мікроконтролера MCU здійснюється управління спринклерами. Блок-схема алгоритму автополива зображена на 2.17. Активувати програму можна через сервер IoT.

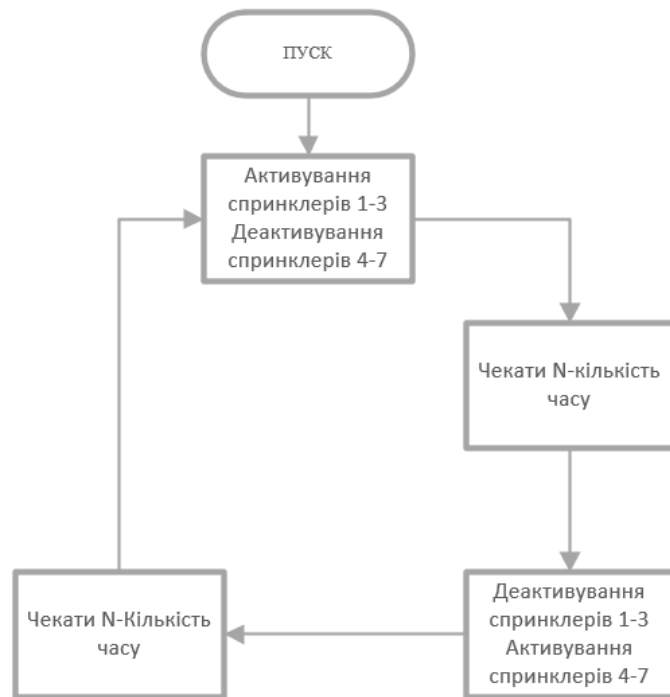


Рисунок 2.17 – Блок-схема алгоритму автополива

Робота контролера зображена на рисунку 2.18.

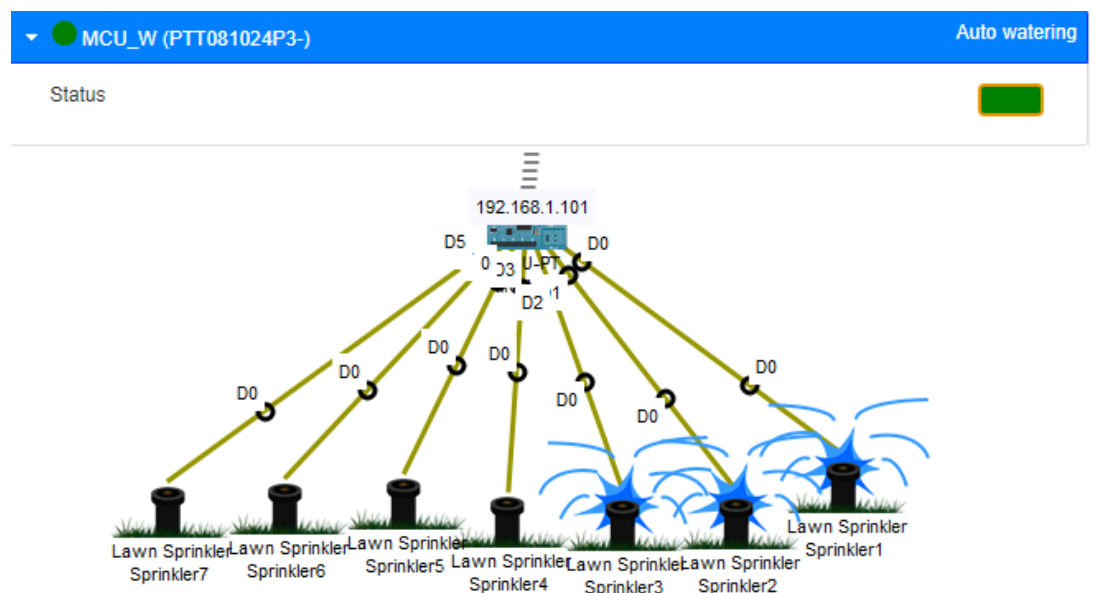


Рисунок 2.18– Робота контролера.

Повний код для контролера у симуляції PacketTracer наведений у додатку А.

2.3 Проектування функцій керування мережею інтернет речей

Для мережі автополиву спроектована загальна функція автополиву (рисунок 2.19), яка являє собою стандартну програму поливу для обраної користувачем ділянки:

1) Спочатку відбувається з'єднання усіх пристроїв мережі з MQTT сервером. У разі неможливості з'єднання пристроїв, клієнту буде відправлення повідомлення про неможливість почати програму автополиву, а програма не буди виконуватись до тих пір, поки проблема не буде усунена.

2) Оновлення даних про стан дощу. Датчиком дощу перевіряється, чи йде зараз дощ. Інформація про стан дощу відправляється до клієнта.

3) Оновлення даних вологості ґрунту. Аналізатори ґрунту кожної ділянки відправляють інформацію про стан ґрунту до клієнта.

3) Оновлення даних температури повітря. Термометр відправляє дані про температуру повітря до клієнта.

4) Перевірка на дощ. Програма чекає поки завершиться дощ для подальшого виконання алгоритма.

5) Аналізатор ґрунту порівнює значення вологості на ділянці з установленим автоматично, або користувачем, порогом N. При достатньому значенні вологості, полив працювати не буде.

6) Перевірка температури повітря з пороговим значенням температури N. Якщо температура повітря не більше C, то полив буде працювати поки вологість ґрунту на ділянці не досягне значення N. В іншому випадку, поки вологість ґрунту не досягне значення M.

7) Після досягнення потрібного рівня вологості, полив припиняється.

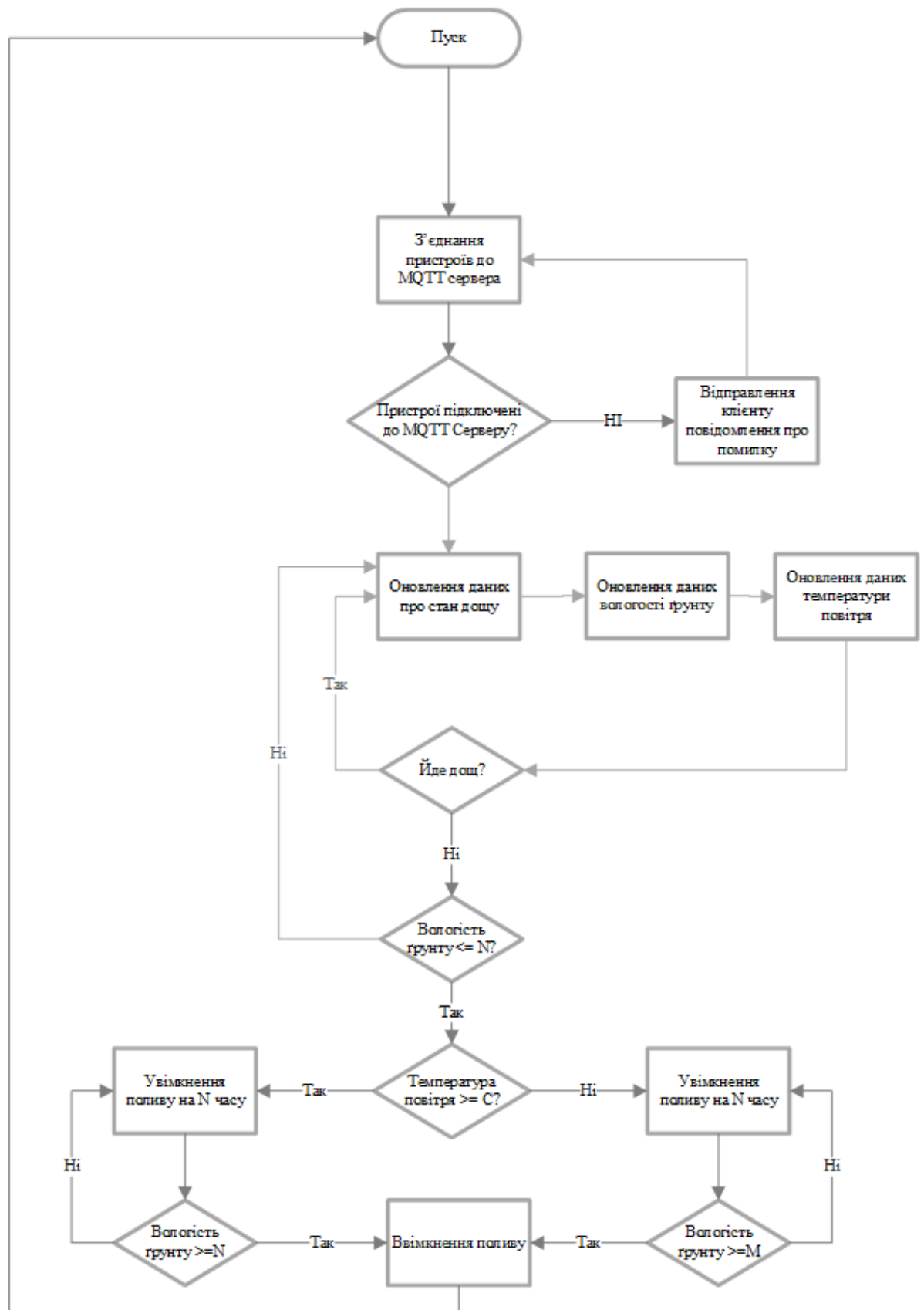


Рисунок 2.19 – Блок-схема основного алгоритму керування автополивом

Для деяких видів рослин, таких як полуниця, потрібен особливий догляд. Для таких рослин написаний адаптивний алгоритм (рисунок 2.20),:

1) Пуск. Встановлення параметрів роботи, такі як оптимальний поріг вологості для полуниці, час поливу, інтервали між вимірюваннями вологості та мінімальний інтервал між поливами.

2) Виміряти вологість ґрунту за допомогою датчика.

3) Перевірити умову. якщо рівень вологості нижче за заданий поріг для полуниці, продовжити дії для поливу.

4) Перевірити стан датчика дощу.

5) Якщо виявлено дощ, пропустити полив і перейти до наступного циклу.

6) Перевірити час останнього поливу. Якщо минуло менше часу, ніж мінімальний інтервал між поливами, перейдіть до наступного циклу без збільшення порогу. Якщо минуло більше часу, увімкнути полив.

7) Періодично перевіряти вологість ґрунту під час поливу. Якщо рівень вологості перевищує певний поріг (наприклад, вище за оптимальне значення), зупинити полив і перейти до наступного циклу.

У функції встановлений адаптивний поріг вологості N . Це змінний поріг вологості, який враховує фази росту та розвитку полуниці. Наприклад, у період цвітіння або плодоношення, коли полуниця потребує більшої кількості води, встановлюється вищий поріг вологості, щоб забезпечити оптимальні умови для рослини. Поріг залежить від поточного циклу, та встановленим інтервалом між поливом.

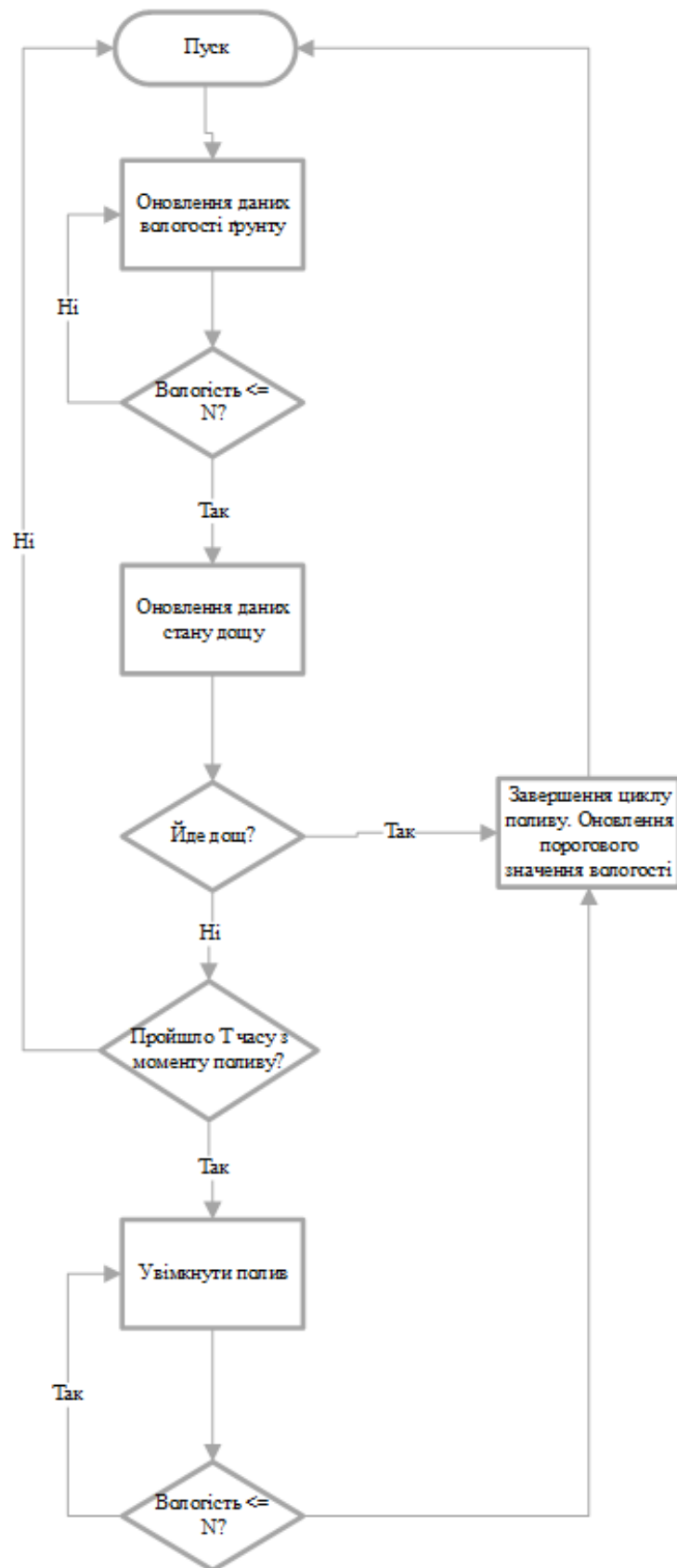


Рисунок 2.20 – Блок-схема адаптивного алгоритму поливу

Також кожний проміжок часу буде спрацьовувати функція оновлення даних з пристроїв да перевірки зв'язку з пристроєм, блок-схема якої зображена на рисунку 2.21 .

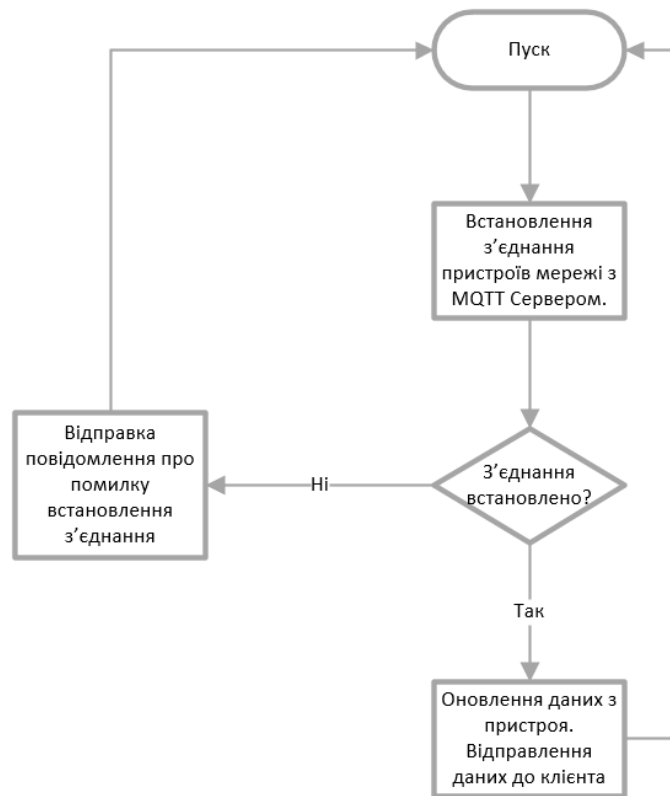


Рисунок 2.21 – Блок-схема алгоритму оновлення даних та перевірки з'єднання.

Функція дійсна для кожного аналізатора ґрунту, для датчика дощу, термометра та мікроконтролера.

З РОЗРОБКА МЕРЕЖІ ІНТЕРНЕТ РЕЧЕЙ ДЛЯ ОРГАНІЗАЦІЇ ОПТИМАЛЬНОГО СЕРЕДОВИЩА ВИРОЩУВАННЯ РОСЛИН ПРИСАДИБНОЇ ДІЛЯНКИ

3.1 Вибір компонентів для мереж інтернет речей.

В якості маршрутизатора обраний TP-LINK Archer C1200[9]. Вигляд маршрутизатора зображений на рисунку 3.1.



Рисунок 3.1 – Вигляд маршрутизатора TP-LINK Archer C1200

Основні характеристики маршрутизатора:

- частота роботи Wi-Fi 5 ГГц + 2.4 ГГц (дводіапазонний);
- стандарт зв'язку Wi-Fi (802.11n);

- 4 порти 10/100/1000 мбіт/с LAN;
- 1 порт 10/100/1000 мбіт/с WAN ;
- швидкість Wi-Fi 1000 мбіт/с.

У якості додаткової точки доступу обраний ретранслятор TP-LINK TL-WA850RE[10], який зображений на рисунку 3.2.



Рисунок 3.2 – Вигляд ретранслятора LINK TL-WA850RE

Основні характеристики ретранслятора:

- частота роботи Wi-Fi 5 ГГц + 2.4 ГГц (дводіапазонний);
- стандарт зв'язку Wi-Fi (802.11n);
- 1 порти 10/100 мбіт/с LAN;
- швидкість Wi-Fi 5 ГГц 433 мбіт/с.
- швидкість Wi-Fi 2.4 ГГц 300 мбіт/с.

Був обраний розумний термометр Shelly Plus H&T[11]. Вигляд термометра зображений на рисунку 3.3.



Рисунок 3.3 – Вигляд термометра Shelly Plus H&T

Цифровий термометр стежить як за температурою, так і за вологістю повітря. Має вбудований веб-сервер, за допомогою якого підключається до мережі Wi-Fi. Працює на батарейках.

Оскільки на ринку не існує пристроїв для вимірювання стану дощу з Wi-Fi з'єднанням, спроектовано датчик дощу на основі мікроконтролера NodeMCU v3 на базі чіпа ESP8266 та модуля датчику дощу. Живлення мікроконтролеру здійснюються звичайними батарейками.

NodeMCU являє собою плату на базі чіпа ESP8266 який являє собою UART-WiFi модуль з ультра-низьким споживанням енергії. Характеристики контролера:

- Wi-fi стандарту 802.11 b / g / n;
- підтримка STA / AP / STA + AP режимів;
- вбудований стек протоколів TCP / IP ;
- 11 портів вводу-виводу ;

Розпіновка контролера зображена на рисунку 3.4:

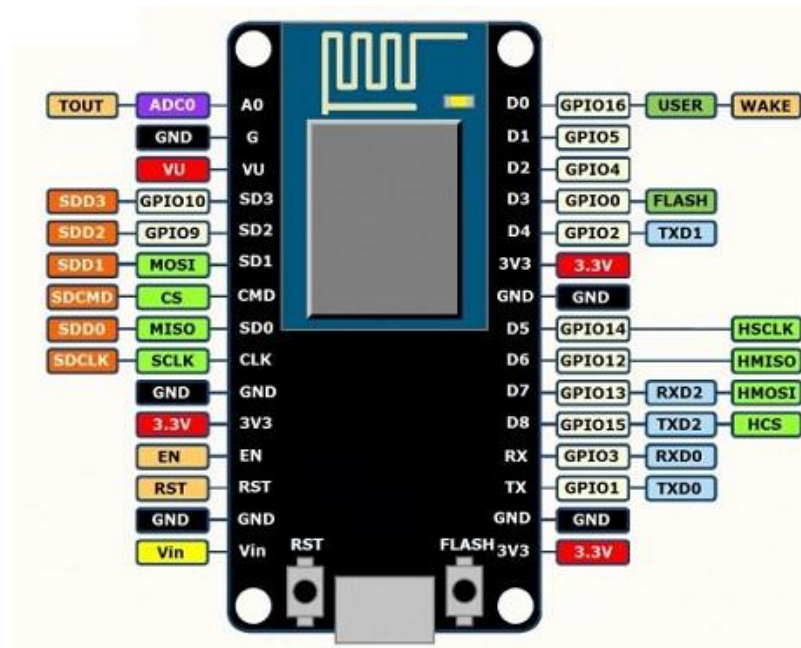


Рисунок 3.4 – Розпіновка контролера NodeMCU v3

Датчик дощу (рисунок 3.5) складається з 2 компонентів:

- плата керування для обробки даних від модуля дощу;
- компаратор, задача якого перетворення значення з сенсора на аналоговий сигнал від 0 до 5 вольт.

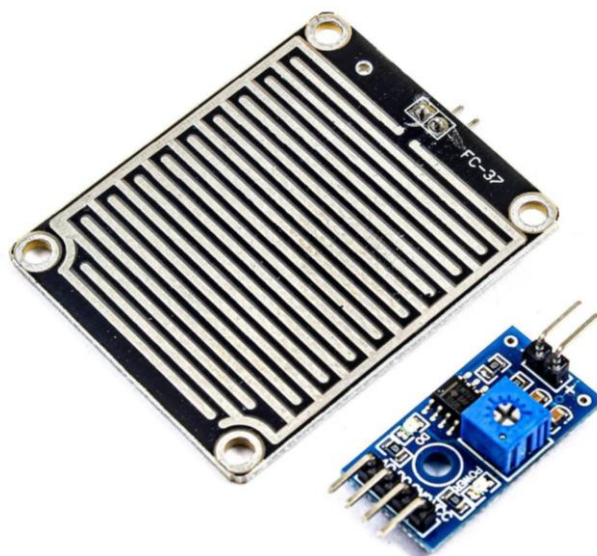


Рисунок 3.5 – Зображення датчику дощу

Сенсор являю собою простий змінний резистор, що замикається водою в різних місцях, що викликає зміну опору. Принцип роботи зображений на рисунку 3.6.

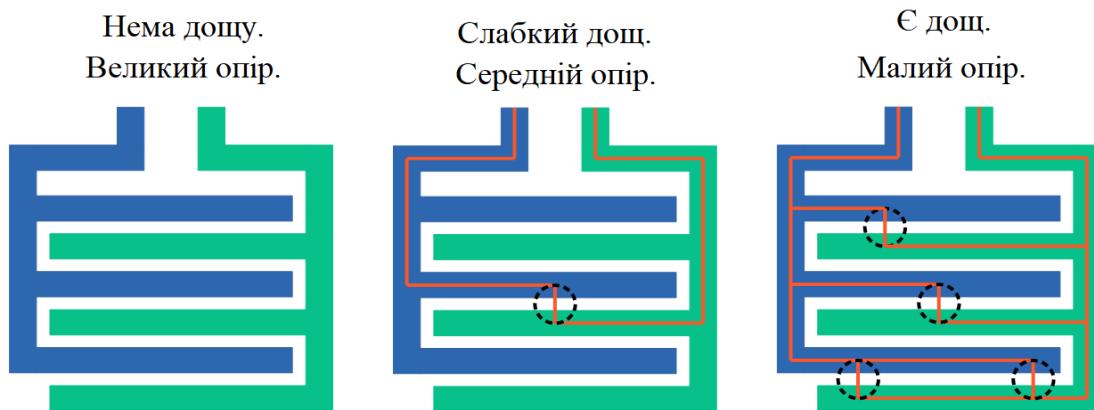


Рисунок 3.6 – Принцип роботи датчику

Для створення аналізатора ґрунту теж обраний мікроконтролер NodeMCU v3. Основний елемент це ємнісний аналоговий датчик вологості ґрунту (рисунок 3.7)



Рисунок 3.7 – Датчик вологості

Датчик повертає інверсне значення: що більше вологість, то нижче значення. На рисунку 3.8 зображений графік залежності вихідної напруги від вологості ґрунту.

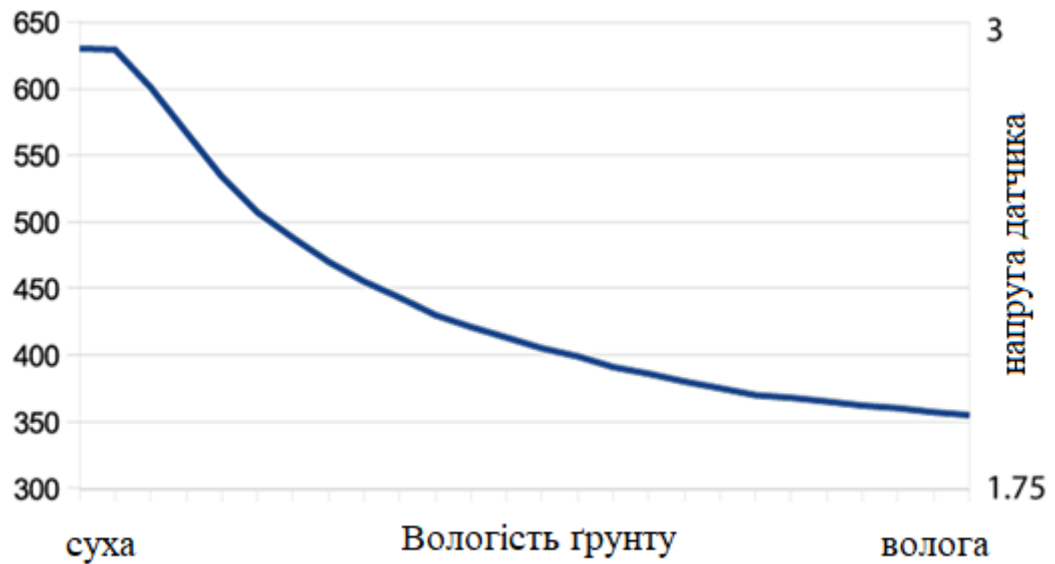


Рисунок 3.8 – Графік залежності вихідної напруги датчику від вологості ґрунту.

У якості mqtt сервера та контролера клапанами обраний мікрокомп'ютер Raspberry Pi 3 Model B+. Raspberry Pi [12] (рисунок 3.9) - це комп'ютер невеликого розміру, але з повноцінною функціональністю. Він відноситься до категорії вбудованих або одноплатних комп'ютерів. У відмінність від мікроконтролерів, таких як NanoMcu, Raspberry Pi має повноцінну операційну систему, що дає йому здатність виконувати складні завдання. Це означає, що на ній можливо встановлювати та виконувати різноманітні програми і додатки, які потребують більшої обробки даних та ресурсів. Також мікрокомп'ютер зберігає дані на SD картці. Raspberry. Технічні характеристики Raspberry Pi 3 model B+.

- Процесор: SOC Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit на частоті 1,4 ГГц ;
- Відеопроцесор Broadcom Videocore-IV;
- RAM 1 Гб LPDDR2 SDRAM;
- Мережа Gigabit Ethernet, Wi-Fi 2,4 ГГц та 5 ГГц 802.11b/g/n/ac;
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE);
- Картка: Micro-SD;
- GPIO 40 контактів;



Рисунок 3.9 – Вигляд мікрокомп'ютера Raspberry PI 3 model B+

GPIO – це 40-контактний роз'єм вводу-виводу загального призначення. Розпіновка GPIO роз'єму зображена на рисунку 3.10.



Рисунок 3.10 – Розпіновка GPIO роз'єму Raspberry PI 3 model B+

Електроклапан обраний Saier SEN-SZ 43WA (рисунок 3.11). Його основні характеристики:

- діаметр різьблення 3/4
- Живлення 12 вольт 500мА
- Тиск 0,02-0,8 мПа

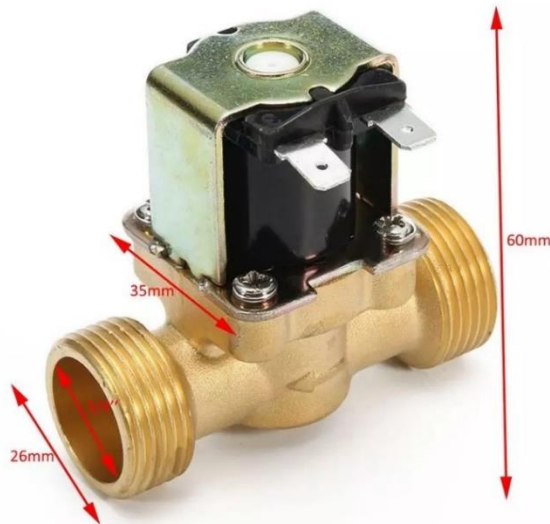


Рисунок 3.11 – Вигляд електромагнітного клапана Saier

Для взаємодії клапанів та Raspberry Pi потрібно реле. Реле дозволяє управляти ланцюгами з більш високою напругою, одночасно використовуючи нижчу напругу для перемикання реле. Було обрано 8-канальний модуль реле з напругою у 5в, та максимальним навантаженням 10 А. Реле зображено на рисунку 3.12.



Рисунок 3.12 – Вигляд реле

Для живлення клапанів аптополиву обраний блок живлення з такими характеристиками:

- вхідна напруга 180-264 VAC;
- вихідна напруга 12 $\pm$ 0,5 VDC;
- струм на виході 5 А;
- вихідна потужність – 60 Вт;

Для живлення аналізаторів ґрунту та дощу обрані дві звичайні батарейки AAA на пристрій.

3.2 Розробка принципових схем вузлів мережі інтернет речей

Для таких пристроїв як контролер поливу, аналізатор ґрунту та аналізатор дощу розроблені принципові схеми. На рисунку 3.13 зображено принципову схему для контролеру поливу.

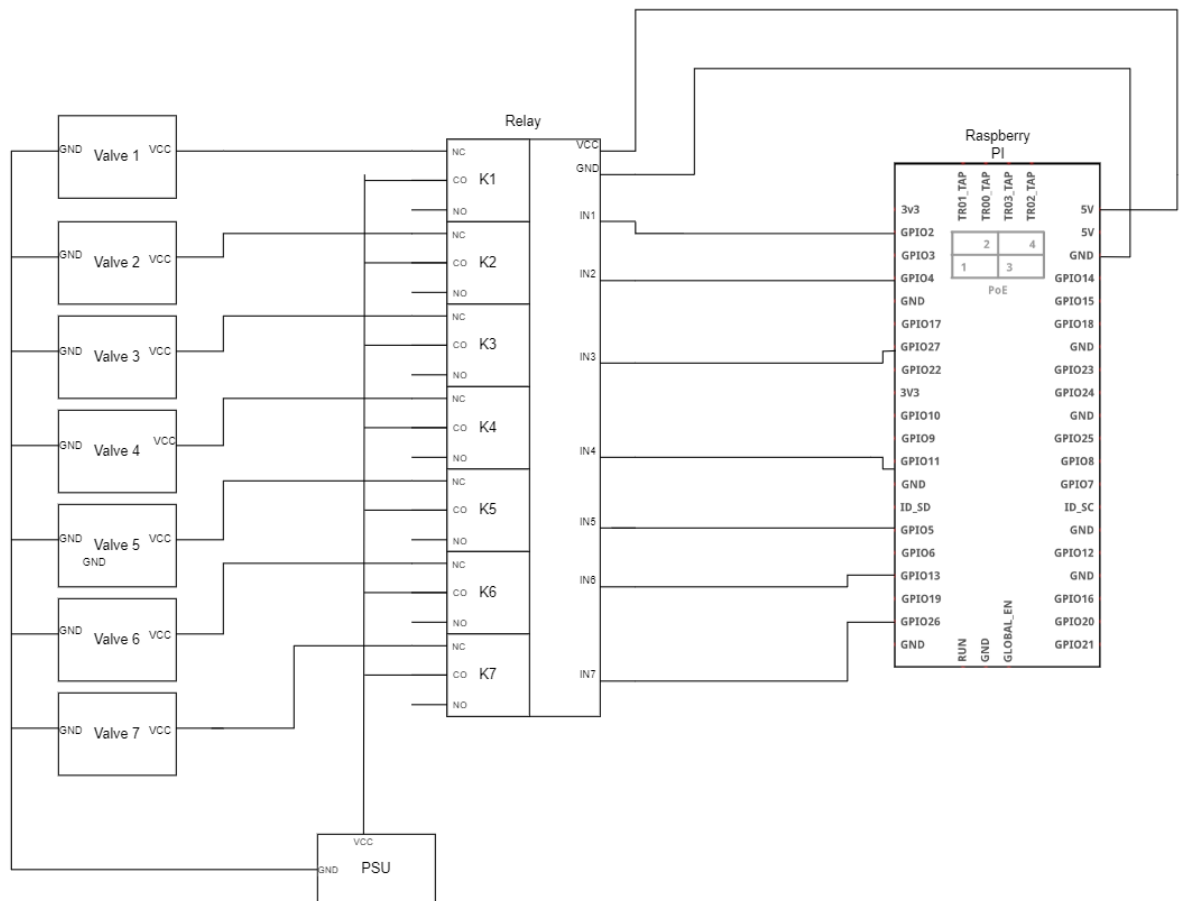


Рисунок 3.13 – Принципова схема контролера поливу.

На схемі контролеру поливу зображений сам мікрокомп'ютер Raspberry Pi 3 b+, реле Relay, електроклапани Valve та блок живлення клапанів PSU. Умовні позначення:

- VCC (DC+, +). Живлення;
- GND (DC-, -). "Земля";
- IN (K). Логічний керуючий сигнал;
- COM (Common). Спільний контакт;
- NO (Normal Open). Нормально розімкнений щодо COM контакт;
- NC (Normal Close). Нормально замкнутий щодо COM контакт;
- GPIO. Контакти Raspberry Pi.

Кожну землю клапана Valve приєднано до землі блока живлення. Живлення клапанів та блока живлення треба приєднати до реле. У неактивного реле замкнуті контакти COM та NC. При активації реле контакт перемикається та COM замикається з NO. Після чого, кожен контакт IN приєднати до контактів GPIO.

Принципова схема аналізатора ґрунту зображена на рисунку 3.14

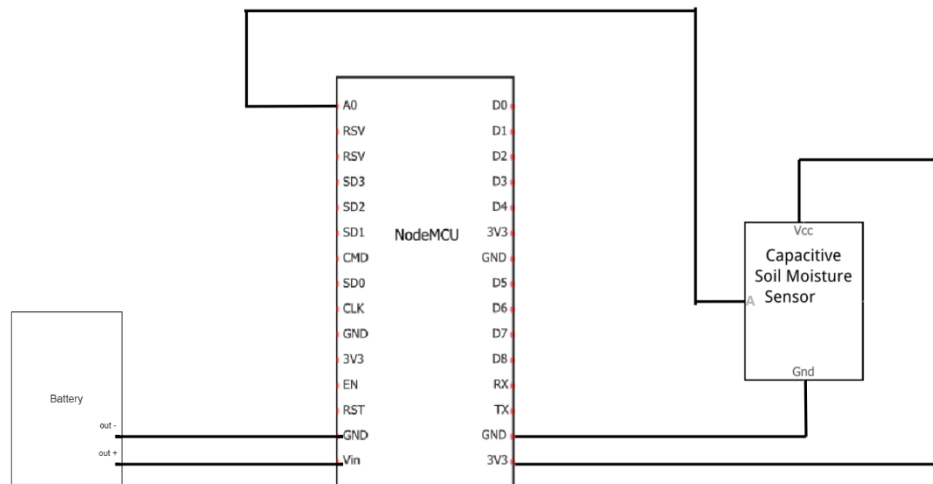


Рисунок 3.14 – Принципова схема аналізатора ґрунту

На схемі зображені батарейки Battery, мікроконтролер NodeMCU та датчик вологості ґрунту Capacitive Soil Moisture Sensor.

3.3 Розробка програмного забезпечення

Для того, щоб Raspberry Pi міг використовувати протокол MQTT, встановлено серверне програмне забезпечення під назвою Mosquitto. Mosquitto - це програма-посередник повідомлень, яка підтримує різні версії протоколу MQTT, включаючи останню версію 5.0.

Raspberry Pi присвоєний статичний IP-адрес 192.168.0.201. На рисунку 3.12 зображено, що MQTT сервер працює.

```
pi@raspberrypi:~$ systemctl status mosquitto.service
● mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre>
   Active: active (running) since Wed 2023-06-07 10:17:07 BST; 9h ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 550 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exi>
   Process: 572 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exi>
   Process: 579 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited,>
   Process: 583 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited,>
  Main PID: 587 (mosquitto)
    Tasks: 1 (limit: 4915)
       CPU: 2.160s
    CGroup: /system.slice/mosquitto.service
            └─587 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Jun 07 10:17:06 raspberrypi systemd[1]: Starting Mosquitto MQTT Broker...
Jun 07 10:17:07 raspberrypi systemd[1]: Started Mosquitto MQTT Broker.
lines 1-17/17 (END)
```

Рисунок 3.15 – Активний MQTT сервер.

Тепер мережа готова до передачі інформації між собою. Інформація між пристроями передається JSON файлами. Це структура, яка має набір пари ключ / значення. Приклад JSON файла, який відправляє аналізатор ґрунту ділянки зображений на рисунку 3.16.

```
{
  "device": "soilsensor",
  "status": "on",
  "soilcount": "22"
  "Dilyanka" : "2"
}
```

Рисунок 3.16 – JSON файл аналізатора ґрунту

Цей файл передає дані клієнту про те, що аналізатор ґрунту з ділянки 2 отримав нові значення вологості ґрунту. Значення вологості ґрунту передається у відсотках від 0 до 100. У даному прикладі значення дорівнює 22%. Також файл говорить, що полив на цій ділянці увімкнений.

Мовою програмування Raspberry Pi обрана Python. Для того щоб програма здійснювати публікацію повідомлень, а також для підписки на теми та отримання опублікованих повідомлень, необхідно підключити бібліотеку Paho-MQTT.

На рисунку 3.17 зображена функція для підключення пристрою до серверу MQTT.

```
def connect_mqtt() -> mqtt_client:
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect, return code %d\n", rc)
client = mqtt_client.Client(client_id)
client.username_pw_set(username, password)
client.on_connect = on_connect
client.connect(broker, port)
return client
```

Рисунок 3.17 – Підключення підключення пристрою до серверу MQTT.

Визначається функція `on_connect`, яка буде викликана при успішному підключенні до брокера MQTT.

Створюється об'єкт клієнта MQTT за допомогою `mqtt_client.Client(client_id)`.

Задається обробник підключення `client.on_connect`, який буде використовуватися при виклику `client.connect()`. Тобто, при підключенні клієнта до брокера MQTT буде викликана функція `on_connect`.

Викликається метод `client.connect(broker, port)`, щоб підключитися до брокера MQTT.

Повертається об'єкт клієнта MQTT `client`, який готовий до використання для взаємодії з брокером MQTT.

Функція для публікації повідомлень зображена на рисунку 3.18.

```
def publish(client, msg):
    result = client.publish(topic, msg)
    status = result[0]
    if status == 0:
        print(f"Send `{msg}` to topic `{topic}`")
    else:
        print(f"Failed to send message to topic {topic}")
```

Рисунок 3.18 – Публікація повідомлень

`Publish(client, msg)` використовується для публікації повідомлення `msg` на заданий топик у брокері MQTT

Функція для підписки на топик та отримання повідомлень зображена на рисунку 3.19.

```
def subscribe(client: mqtt_client):
    def on_message(client, userdata, msg):
        json_str = msg.payload.decode('utf-8')
        print(f"Received `{msg.payload.decode()}` from  

`{msg.topic}` topic")
        str = msg.payload.decode()
        client.subscribe(topicD1)
        client.subscribe(topicD2)
        client.on_message = on_message
```

Рисунок 3.19 – Функція підписки на топик

`Subscribe(client)` використовується для підписки клієнта MQTT на певні топіки та встановлення функції зворотного виклику `on_message` для обробки отриманих повідомлень.

Основна функція поливу `check_weather_conditions` зображена на рисунку 3.20.

```
def check_weather_conditions():
    # Перевірка з'єднання з MQTT сервером
    if not is_connected_to_mqtt():
        send_error_message()
        return

    while True:
        # Отримання даних про дощ
        while is_raining():
            process_rain_data()
            time.sleep(1)

        # Отримання даних про вологість ґрунту
        while is_soil_moisture_high():
            process_soil_moisture_data()
            time.sleep(1)

        # Отримання даних про температуру
        if is_air_temperature_high():
            start_irrigation(20)
            while is_soil_moisture_low():
                process_soil_moisture_data()
                time.sleep(1)
        else:
            start_irrigation(dilyanka, 20)
            while is_soil_moisture_low():
                process_soil_moisture_data()
                time.sleep(1)
```

Рисунок 3.20 – Функція `check_weather_conditions`

Функція `check_weather_conditions` виконує перевірку погодних умов для системи поливу, і якщо всі умови виконані, полив вмикається.

В якості графічного інтерфейса обраний застосунок `MQTTPanel`. Він дозволяє легко візуалізувати отриману з пристроїв інформацію, та здійснювати їх контроль. Зображення інтерфейсу застосунка наведено на рисунку 3.21.

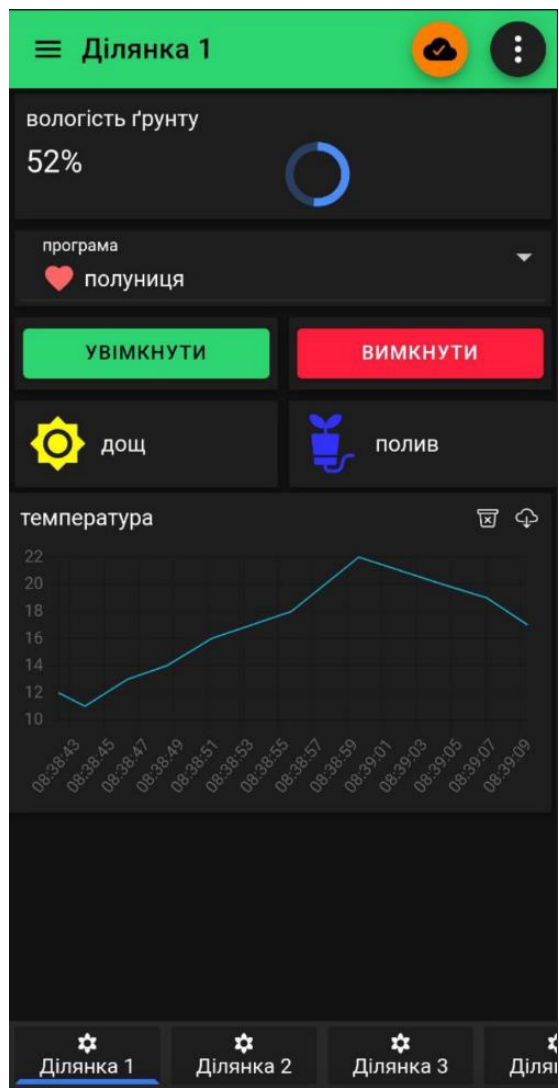


Рисунок 3.21 – Мобільний застосунок MQTTPanel

3.4 Розробка дослідного зразка та перевірка працездатності.

Для проекту створений макет, який складається з таких елементів:

- Raspberry Pi 4;
- NodeMCU v3;
- водяна помпа;
- датчик вологості ґрунту;
- датчик температури;

- реле 5в;
- три блока живлення.

Слід зазначити, що для імітації поливу замість електромагнітного клапану, обрано водяну помпу. Вона підключена до Raspberry PI і керується нею. Для тестування працездатності, макет здійснював керування поливом кімнатних рослин (рисунок 3.22).



Рисунок 3.22 – Зібраний макет

Також для перевірки функцій поливу використана програма MQTTX. Завдяки неї, можливо відправляти повідомлення на обрані топіки мережі. Відправимо повідомлення від аналізатора ґрунту на топік controller. На рисунку 3.23 зображено відправлення JSON повідомлення на топік controller у програмі MQTTX.

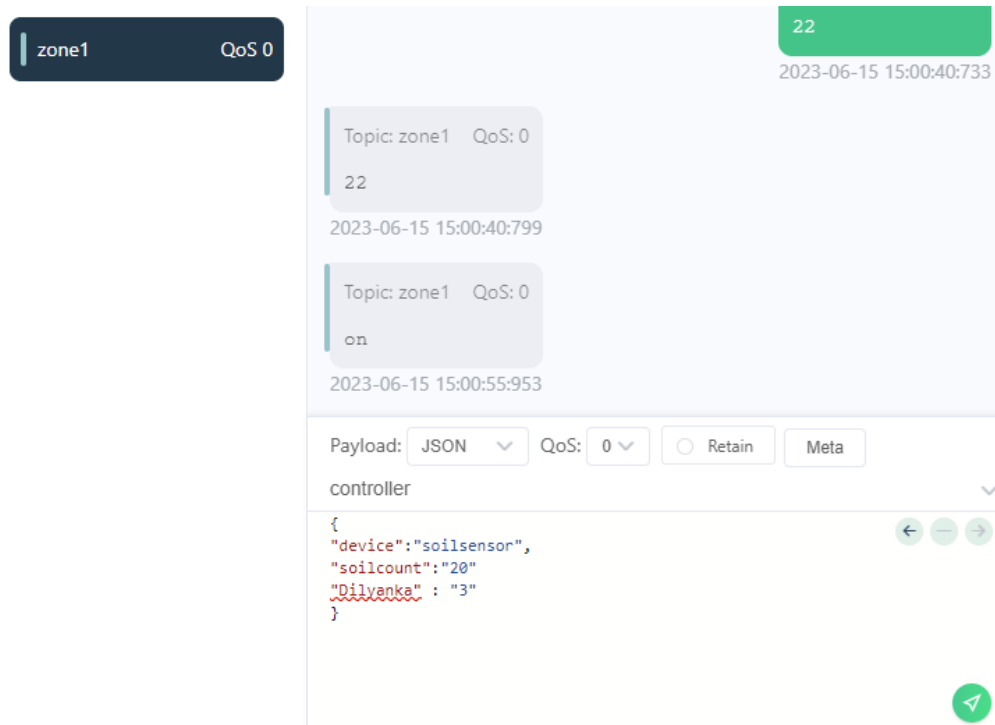


Рисунок 3.23 – Інтерфейс програми MQTTX

На топик контролер відправляється JSON файл, який містить в собі інформацію про те, що на ділянці 3 вологість ґрунту складає 20 відсотків. Контролер повинен відреагувати на повідомлення, і увімкнути полив. Реакцію контролера зображено на інтерфейсі додатку (рисунок 3.24).

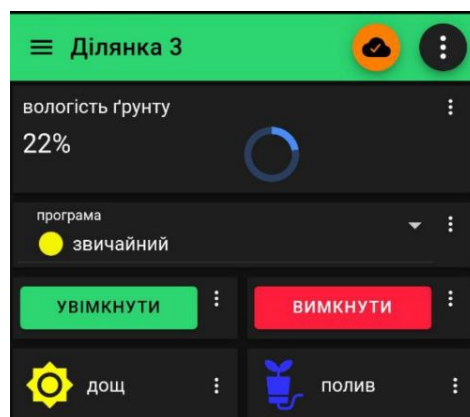


Рисунок 3.24 – Статус ділянки 3 на інтерфейсі додатка

3.5 Кошторис проекту IoT мережі

Кошторис розробленої IoT мережі для автоматичного поливу присадибної ділянки наведений у таблиці 3.1

Таблиця 3.1 – Кошторис IoT мережі.

Назва компонента	Ціна (грн за шт)	Кількість (шт)	Ціна (разом грн)
Реле 8-канальний	200	1	200
Маршрутизатор TP-LINK Archer C1200	1300	1	1300
Ретранслятор TP-LINK AC750 RE200	750	1	750
Датчик вологості ґрунту Capacitive 1.2	65	7	455
Розумний термометр Shelly Plus H&T	1100	1	1100
Блок живлення S-15-3	250	1	250
Датчик дощу YL-42	40	1	40
Raspberry PI 3 B+	3200	1	3200
NodeMCU v3	200	8	1600
Електроклапан Saier 3/4	750	7	5250
Батарейки	70	16	1120
Батарейний відсік	40	8	320
Ціна разом	15 385		

ВИСНОВКИ

В процесі виконання дипломної роботи проведено аналіз існуючих систем автополиву та типових його елементів. Вияснено, що автоматизована система управління поливом складається з таких елементів, як пульт керування, датчики погодних умов, спринклери, електромагнітні клапани.

Проведено огляд технологій реалізації IoT мереж автополиву, таких як BLE, ZigBee, Z-wave, LoRaWan, Wi-Fi, Np-IoT.

Спроековано фізичну та логічну топології територіальної локальної IoT мережі присадибної ділянки для реального об'єкта.

Проведено моделювання і тестування роботи спроекованої IoT мережі в середовищі симулятора Packet Tracer компанії Cisco. Створені правила, при яких розумні пристрої виконували певну роботу. Створені спеціальні розумні пристрої для моделювання, такі як датчик дощу, аналізатор ґрунту. Також створено контролер поливу з мікроконтролера та спрінклерів. Контролер запрограмований на простий алгоритм поливу. Код датчиків та контролера написаний мовою Python

Проведено проектування функцій керування автополивом в IoT мережі присадибної ділянки. Створені функції звичайного поливу, адаптивного поливу, алгоритм для оновлення даних з датчиків.

Обрано активне та пасивне обладнання для реалізації IoT мережі присадибної ділянки. В якості контролера клапанами та сервера MQTT обраний мікрокомп'ютер Raspberry Pi 3+. Для створення пристроїв аналізатора дощу та ґрунту обраний мікроконтролер NodeMCU v3.

Обрано протокол MQTT для обміну повідомлень між вузлами IoT мережі. Створено сервер MQTT.

Проведена розробка принципових схем підключення IoT вузлів. Розроблені принципові схеми для контролеру поливу, аналізатору дощу, аналізатору ґрунту.

Запрограмовано функції керування автополиву. Розроблений дослідний зразок системи автополиву з можливістю дистанційного керування. Для зразку створений аналізатор ґрунту з мікроконтролера та датчику вологості ґрунту, створено примітивну систему поливу з помпи. Протестована функціональність системи на кімнатній рослині.

Розрахована вартість реалізації IoT мережі. Вартість складає 15 385 грн.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Опис IoT [Електронний ресурс] – Режим доступу: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>
2. Системи автополиву [Електронний ресурс] – Режим доступу: <https://sad.ukr.bio.ua/articles/6285/>
3. Технології реалізації IoT Мереж [Електронний ресурс] – Режим доступу: <https://www.emnify.com/blog/iot-connectivity>
4. Контролер Hunter [Електронний ресурс] – Режим доступу: <https://www.aquaprice.com.ua/avtopoliv/hunter/kontrollery/kontroller-hunter-x2-801-e-naruzhnyy>
5. ПО Hydrowise [Електронний ресурс] – Режим доступу: <https://www.hunterindustries.com/ru/product/oblachnoe-programmnoe-obespechenie-hydrowise> облако для хантера
6. Контролер RainBird [Електронний ресурс] – Режим доступу: <https://rain-bird.com.ua/shop/kontrollery/kontroller-esp-rzx-6>
7. WiFi модуль RainBird [Електронний ресурс] – Режим доступу: https://rain-bird.com.ua/shop/aksessuary-dlya-kontrollerov/rain_bird_wifi_modul_lnkwifi
8. Мова Python у Cisco Packet Tracer [Електронний ресурс] – Режим доступу: <https://www.linkedin.com/pulse/how-use-python-cisco-packet-tracer-luciano-martins-fagundes>
9. Маршрутизатор TP-LINK Archer C1200 [Електронний ресурс] – Режим доступу: <https://rozetka.com.ua/ua/12254999/p12254999/>
10. Ретранслятор TP-LINK AC750 RE200 [Електронний ресурс] – Режим доступу: https://rozetka.com.ua/ua/tp_link_ac750_re200/p4674775
11. Цифровий термометр Shelly Plus RE200 [Електронний ресурс] – Режим доступу: <https://www.tradeinn.com/techinn/ru/shelly-plus-h-t>

12.Документація Raspberry PI [Електронний ресурс] – Режим доступу:
<https://www.raspberrypi.com/documentation/>

ДОДАТОК А

КОД ПРОГРАМИ КОНТРОЛЕРА У CISCO PACKET TRACER

```

from options import Options
from time import *
from gpio import *
from ioeclient import IoEClient
state = 0
def setup ():
    global state
    IoEClient.setup({
        "type": "Auto watering",
        "states": [{
            "name": "Status",
            "type": "bool",
            "controllable": True
        }]
    })
    IoEClient.onInputReceive(lambda rinput:
processData(rinput, True))

def on_event_detect():
    processData(customRead(0), False)
    add_event_detect(0, on_event_detect)
    state = restoreProperty("state", 0)
    setState(state)

def restoreProperty (propertyName, defaultValue):
    value = getDeviceProperty(getName(), propertyName)
# var value
    if value:
        if isinstance(defaultValue, (int, float)):
            value = int(value)
        setDeviceProperty(getName(), propertyName, value)
        return value

    return defaultValue

def mouseEvent (pressed, x, y, firstPress):
    if firstPress:
        setState(( 0 if state else 1 ) )

```

```

def processData (data, bIsRemote):
    if len(data) <= 0:
        return
    setState(int(data))

def setState (newState):
    global state
    state = newState
    IoEClient.reportStates(state)
    setDeviceProperty(getName(), "state", state)
    offWatering()

def offWatering():
    if state == 0:
        customWrite(0, 0)
        customWrite(1, 0)
        customWrite(2, 0)
        customWrite(3, 0)
        customWrite(4, 0)
        customWrite(5, 0)
        customWrite(6, 0)
        print("stop")

def watering ():
    if state == 1:
        print("start watering")
        customWrite(0, 0)
        customWrite(1, 0)
        customWrite(2, 0)
        customWrite(3, 1)
        customWrite(4, 1)
        customWrite(5, 1)
        customWrite(6, 1)
        delay(100)
        customWrite(0, 1)
        customWrite(1, 1)
        customWrite(2, 1)
        customWrite(3, 0)
        customWrite(4, 0)
        customWrite(5, 0)
        customWrite(6, 0)
        delay(100)
        print("end watering")

if __name__ == "__main__":

```

```
setup()  
while True:  
    watering()
```

ДОДАТОК Б
ФІЗИЧНА ТОПОЛОГІЯ ІОТ МЕРЕЖІ

