МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ОСНОВИ ІОТ»
ДЛЯ СТУДЕНТІВ УСІХ ФОРМ НАВЧАННЯ
ЗІ СПЕЦІАЛЬНОСТІ 123 – «КОМП’ЮТЕРНА ІНЖЕНЕРІЯ»
ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ
«КОМП’ЮТЕРНА ІНЖЕНЕРІЯ»
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»
ЧАСТИНА 1

КРЕМЕНЧУК 2023

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Основи IoT» для студентів усіх форм навчання зі спеціальності 123 – «Комп'ютерна інженерія» освітньо-професійної програми «Комп'ютерна інженерія» освітнього ступеня «Бакалавр» (частина 1)

Укладачі: д. т. н., проф. А. Л. Перекрест,
асист. К. О. Вадурін

Рецензент к. т. н., доц. В. М. Сидоренко

Кафедра комп'ютерної інженерії та електроніки

Затверджено методичною радою Кременчуцького національного університету імені Михайла Остроградського
Протокол № 7 від 27 03 2023 року
Голова методичної ради проф. В. В. Костін

ЗМІСТ

ВСТУП

Метою вивчення навчальної дисципліни «Основи IoT» є отримання кваліфікації для проєктування архітектури та створення програмного забезпечення для вбудованих інформаційних систем з урахуванням обмеженості обчислювальних ресурсів, можливостей обміну інформацією з іншими цифровими пристроями як у локальній мережі, так і в Інтернеті.

Завдання вивчення навчальної дисципліни «Основи IoT» полягає у: накопиченні теоретичних знань про основні види будови програмного забезпечення для вбудованих систем, вимоги до такого забезпечення й засоби, якими можливо задовільнити поставлені вимоги; набутті практичних навичок у розв'язанні типових задач, що зустрічаються від час роботи із засобами IoT чи пристроями типу Smart.

Програмні результати після виконання запропонованих лабораторних робіт такі:

− Знати і розуміти наукові положення, що є підґрунтям функціонування комп'ютерних засобів, систем і мереж.

− Мати навички проведення експериментів, збирання даних і моделювання в комп'ютерних системах.

− Знати новітні технології в галузі комп'ютерної інженерії.

− Знати й розуміти вплив технічних рішень у суспільному, соціальному та екологічному контексті.

− Уміти застосовувати знання для ідентифікації, формулювання й розв'язування технічних задач зі спеціальності, використовуючи методи, що є найпридатнішим для досягнення поставлених цілей.

− Уміти розв'язувати задачі аналізу й синтезу засобів, характерних для спеціальності.

− Уміти системно мислити й застосовувати творчі здібності до формування нових ідей.

– Уміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп’ютерних систем та мереж для розв’язання технічних задач зі спеціальності.

– Уміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

– Уміти здійснювати пошук інформації в різних джерелах для розв’язання задач комп’ютерної інженерії.

– Уміти ефективно працювати як індивідуально, так і у складі команди.

– Уміти ідентифікувати, класифікувати й описувати роботу комп’ютерних систем та їх компонентів.

– Уміти поєднувати теорію і практику, а також приймати рішення й виробляти стратегію діяльності для розв’язування завдань зі спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних і виробничих інтересів.

– Уміти оцінювати отримані результати й аргументовано захищати прийняті рішення.

1 ПЕРЕЛІК ЛАБОРАТОРНИХ РОБІТ

Лабораторна робота № 1

Тема. Взаємодія з Інтернет-речами. Модель передачі даних у IoT

Мета: ознайомитися із системою Cisco Packet Tracer і засобами налаштування мережної інфраструктури.

1.1 Короткі теоретичні відомості

Packet Tracer – симулятор мережі передачі даних, що випускається фірмою Cisco Systems. Дозволяє робити працездатні моделі мережі, налаштовувати (командами Cisco IOS) маршрутизатори та комутатори, взаємодіяти між кількома користувачами (через хмару). Packet Tracer в основному орієнтований для учасників Програми Мережної Академії Cisco як безкоштовний навчальний посібник, що допомагає їм вивчити основні концепції Сертифікації Cisco.

Packet Tracer надає можливість учням проєктувати складні й великі мережі, що часто неможливо з фізичним обладнанням через великі витрати. Packet Tracer зазвичай використовується студентами, оскільки є безкоштовним. Packet Tracer надає додаткові компоненти для навчання, зокрема авторську систему, моделювання мережного протоколу, поліпшення знань і систему оцінювання. Однак, через функціональні обмеження, Cisco має намір використовувати його тільки як навчальний посібник, а не як заміну маршрутизаторам і комутаторам Cisco. Сам додаток має лише невелику кількість функцій, присутніх у реальному обладнанні, на якому працює поточна версія Cisco IOS. Отже, Packet Tracer не підходить для моделювання виробничих мереж. Він має обмежений набір команд, що означає, що він не дозволяє використовувати всі команди IOS, які можуть знадобитися. Packet Tracer може бути корисний для розуміння абстрактних мережних концепцій, як-от Enhanced Internal Gateway Routing Protocol способом анімації цих елементів у візуальній формі.

1.2 Порядок виконання роботи

Реєструємося на курсах, доступних на кафедрі, для отримання безкоштовного доступу до Cisco Packet Tracer.

Частина 1. Запустіть Packet Tracer.

a. Запустіть Packet Tracer на своєму ПК або ноутбуці.

Двічі натисніть піктограму Packet Tracer на своєму робочому столі або перейдіть до провідника на вашому комп'ютері, який містить файл Packet Tracer, і запустіть Packet Tracer. Packet Tracer слід відкрити за допомогою порожньої робочої області логічної топології за замовчуванням, як показано на рисунку 1.1.

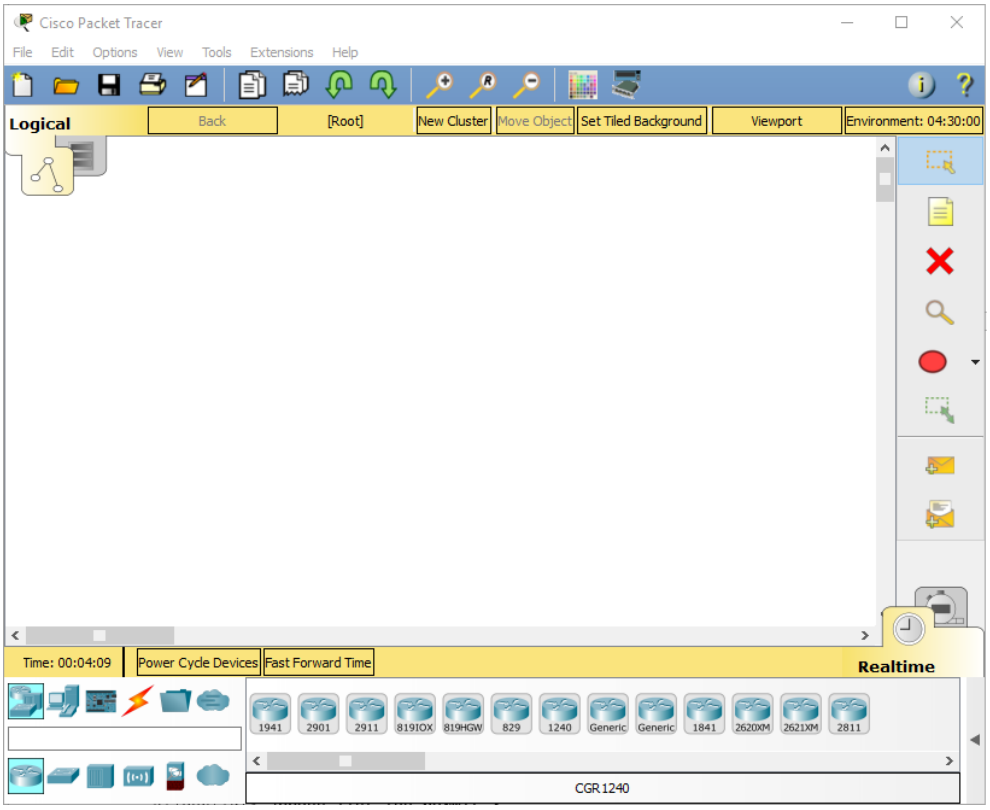

Рисунок 1.1 – Packet Tracer

Крок 2. Побудуйте топологію.

a. Додайте мережні пристрої до робочої області.

На полі вибору пристрою додайте мережні пристрої до робочої області, як показано на схемі топології.

Щоб помістити пристрій у робоче середовище, спочатку виберіть тип пристрою з поля **Вибір типу пристрою**. Потім натисніть потрібну модель пристрою з поля **Вибір конкретного пристрою**. Нарешті, натисніть на робочу область, щоб розмістити свій пристрій у цьому місці. Якщо ви хочете скасувати вибір, натисніть значок **Скасувати** для цього пристрою. Крім того, ви можете натиснути та перетягнути пристрій із вікна **} Вибір конкретного пристрою** на робоче середовище.

b. Додайте мережні пристрої до робочої області.

Використовуючи вікно вибору пристрою, додайте мережні пристрої на робочий простір, як показано на діаграмі топології.

Щоб помістити пристрій на робоче середовище, спочатку виберіть тип пристрою з вікна **Вибір типу пристрою**. Потім натисніть потрібну модель пристрою з поля **Вибір конкретного пристрою**. Нарешті, натисніть на робочу область, щоб розмістити свій пристрій у цьому місці. Якщо ви хочете скасувати вибір, натисніть значок **Скасувати** для цього пристрою. Крім того, ви можете натиснути та перетягнути пристрій із вікна **Вибір конкретного пристрою** на робоче середовище.

c. Змінити відображення назв мережних пристроїв.

Щоб змінити відображувані назви мережних пристроїв, натисніть на іконку пристрою на робочому місці Packet Tracer Logical, після натисніть вкладку **Config** у вікні налаштування пристрою. На вкладці Config уведіть нову назву пристрою у вікно **Відображуване ім’я**, як показано на рисунку 1.2.

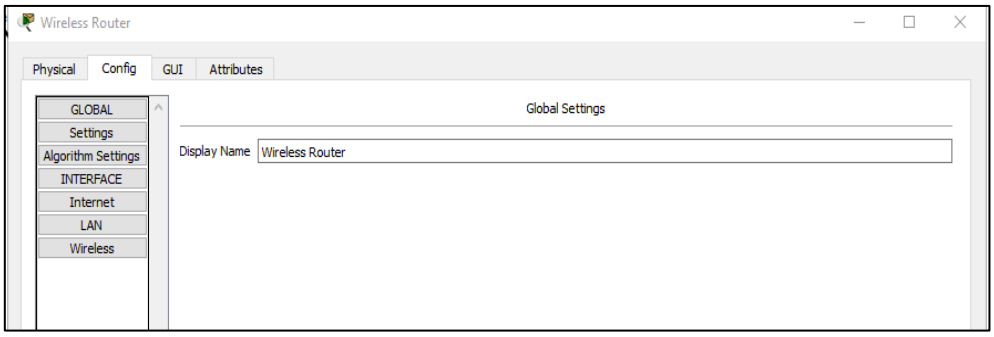

Рисунок 1.2 – Відображуване ім’я

d. Додайте фізичну проводку між пристроями на робочій області.

Використовуючи поле вибору пристрою, додайте фізичну проводку між пристроями в робочій області, як показано на діаграмі топології.

Для підключення до бездротового маршрутизатора ПК знадобиться мідний наскрізний кабель Виберіть мідний наскрізний кабель у вікні «Вибір пристрою» і прикріпіть його до інтерфейсу FastEthernet0 на ПК і інтерфейсу Ethernet 1 бездротового маршрутизатора.

Для підключення до кабельного модема для бездротового маршрутизатора потрібен мідний прямоточний кабель. Виберіть мідний прямоточний кабель у вікні Device-Selection і прикріпіть його до інтерфейсу Інтернету Wireless Router та інтерфейсу Port 1 кабельного модема.

Кабельний модем повинен мати коаксіальний кабель для підключення до хмарної мережі Інтернет. Виберіть коаксіальний кабель у вікні вибору пристрою та прикріпіть його до порту 0 кабельного модема й коаксіального інтерфейсу хмарної мережі Інтернет.

Для підключення до сервера Cisco.com для Інтернету необхідно мати мідний прямий кабель. Виберіть мідний прямоточний кабель у вікні Device-Selection і прикріпіть його до інтерфейсу Ethernet в Інтернеті та інтерфейсу FastEthernet0 сервера Cisco.com.

Частина 2. Налаштуйте мережні пристрої.

Крок 1. Налаштуйте бездротовий маршрутизатор.

e. Створіть бездротову мережу на бездротовому маршрутизаторі.

Натисніть іконку Бездротового маршрутизатора на робочому полі Packet Tracer Logical, щоб відкрити вікно налаштування пристрою.

У вікні налаштування бездротового маршрутизатора натисніть вкладку GUI, щоб переглянути параметри конфігурації бездротового маршрутизатора.

Далі натисніть на вкладку **Wireless** у графічному інтерфейсі, щоб переглянути параметри бездротового зв’язку. Єдиним параметром, який

потрібно змінити, є **Назва мережі (SSID)**. Тут уведіть ім’я «HomeNetwork», як показано на рисунку 1.3.

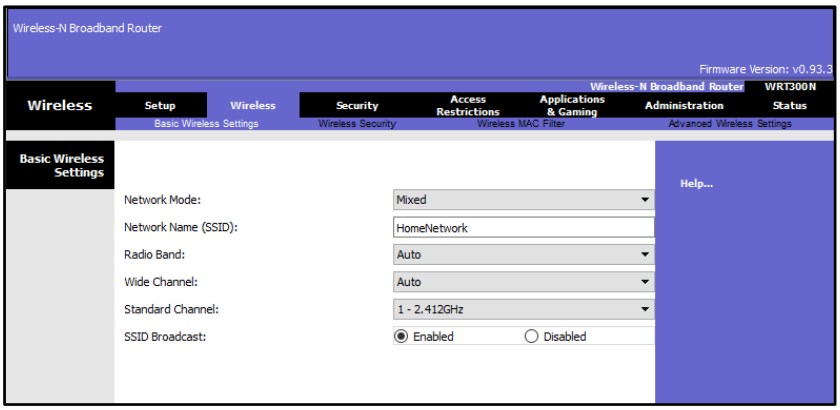

Рисунок 1.3 – Панель керування мережею

b. Налаштуйте підключення до Інтернету на бездротовому маршрутизаторі.

Натисніть вкладку **Налаштування** на графічному інтерфейсі бездротового маршрутизатора.

У налаштуваннях сервера DHCP перевірте, чи вибрано кнопку **Увімкнено,** та налаштуйте статичну IP-адресу DNS-сервера як 208.67.220.220, як показано на рисунку.

c. Натисніть вкладку **Зберегти параметри**.

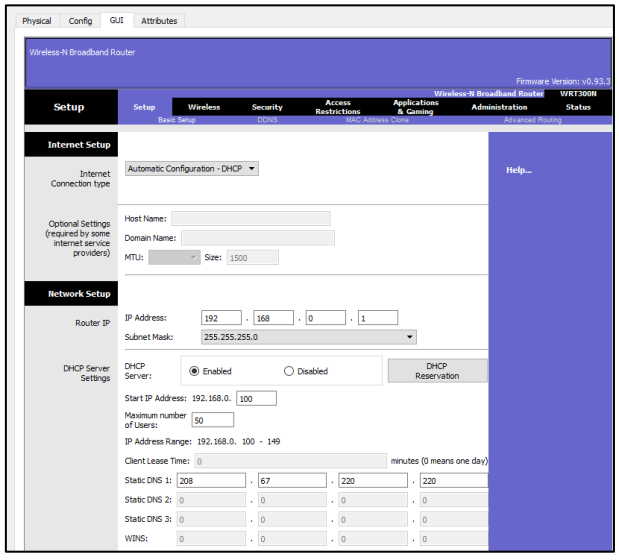

Рисунок 1.4 – Панель параметрів

Крок 2. Налаштуйте ноутбук.

a. Налаштуйте ноутбук для доступу до бездротової мережі.

Натисніть значок ноутбука на робочому столі Packet Tracer Logical і у вікні налаштування ноутбука виберіть вкладку **Physical**.

На вкладці Physical вам потрібно буде видалити мідний модуль Ethernet і замінити його модулем Wireless WPC300N.

Для цього спочатку вимкніть ноутбук, натиснувши кнопку живлення на боці ноутбука. Потім видаліть установлений зараз мідний модуль Ethernet, натиснувши модуль збоку ноутбука й перетягнувши його в панель **MODULES** зліва від вікна ноутбука. Потім установіть модуль Wireless WPC300N, натиснувши на панель **MODULES** і перетягнувши його в пустий модульний порт на стороні ноутбука. Увімкніть ноутбук знову, натиснувши кнопку живлення ще раз.

Після встановлення бездротового модуля наступне завдання – підключити ноутбук до бездротової мережі.

Натисніть вкладку **Desktop** у верхній частині вікна налаштування ноутбука й виберіть значок **PC Wireless**.

Коли налаштування адаптера для ноутбуків Wireless-N відображатимуться, виберіть вкладку **Connect**. Бездротова мережа «HomeNetwork» має бути видимою у списку бездротових мереж, як показано на рисунку.

Виберіть мережу й натисніть вкладку **Підключити**, розташовану нижче **Інформації про сайт.**

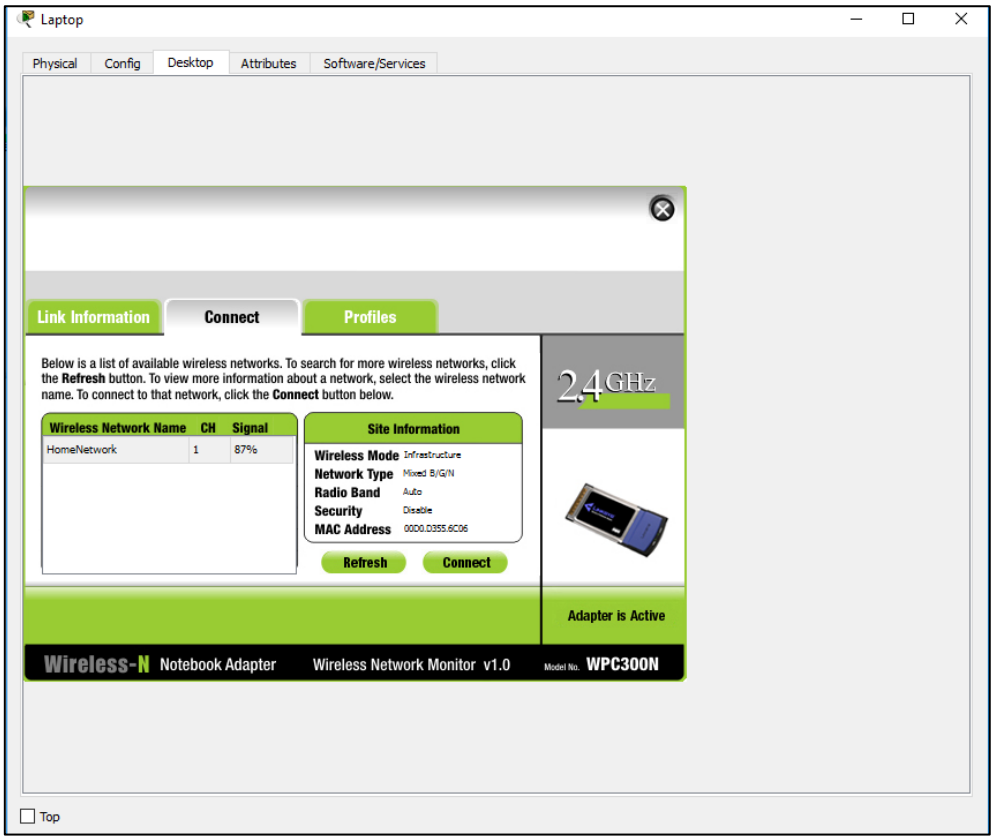

Рисунок 1.5 – Ноутбук

Крок 3. Налаштуйте ПК.

a. Налаштуйте ПК для дротової мережі.

Натисніть іконку ПК на робочому столі Packet Tracer Logical і виберіть вкладку **Desktop**, а потім **іконку IP Configuration**.

У вікні «Конфігурація IP» виберіть перемикач **DHCP**, як показано на рисунку, щоб ПК використовував DHCP для отримання адреси IPv4 з бездротового маршрутизатора. Закрийте вікно налаштування IP.

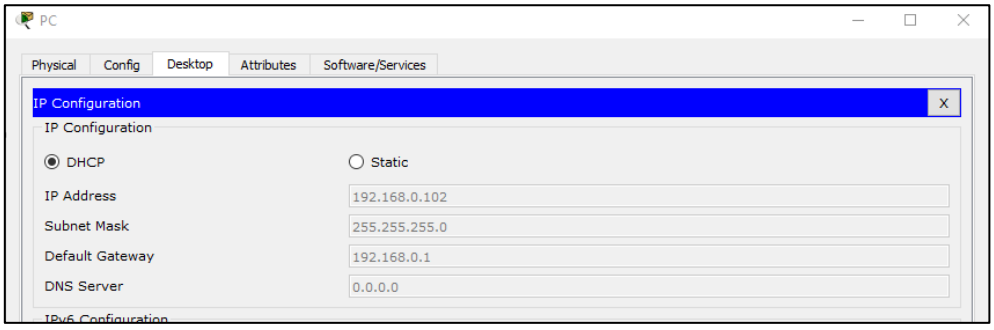

Рисунок 1.6 – Панель комп’ютера

Натисніть значок командного рядка. Переконайтеся, що ПК отримав адресу IPv4, видавши команду ipconfig/all command, як показано на рисунку. ПК має отримати адресу IPv4 у діапазоні 192.168.0.x.

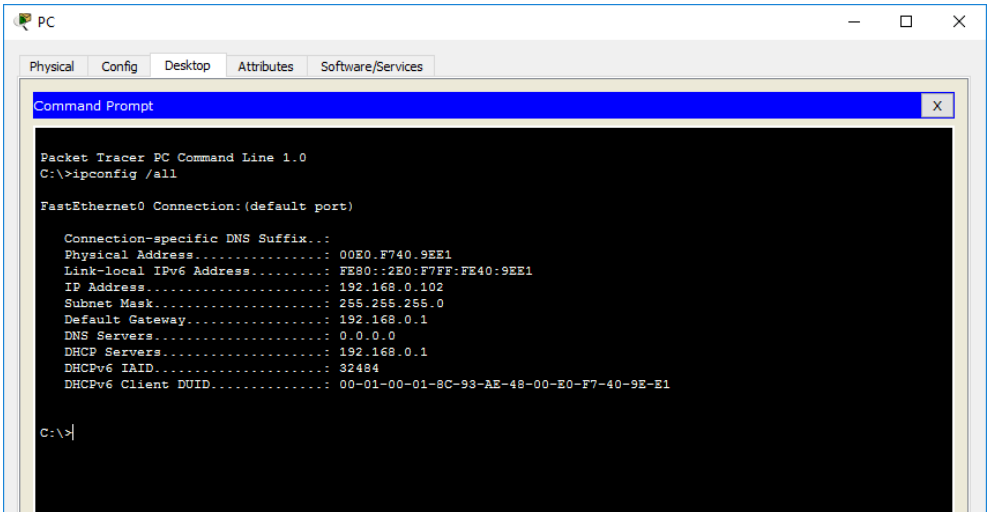

Рисунок 1.7 – Командний рядок

Крок 4. Налаштуйте Інтернет-хмару.

a. У разі потреби встановіть мережні модулі.

Натисніть іконку Packet Tracer Logical, а потім на вкладку Physical. Обладнання для хмар потребує двох модулів, якщо вони ще не встановлені. PT-CLOUD-NM-1CX, який призначений для підключення кабельного модему, та PT-CLOUD-NM-1CFE для кабельного з'єднання з міддю Ethernet. Якщо ці модулі відсутні, вимкніть фізичні пристрої хмари, натиснувши кнопку живлення, та перетягніть кожен модуль у пустий порт модуля пристрою, а потім знову ввімкніть пристрій.

b. Ідентифікація портів «Від і до».

Натисніть вкладку **Config** у вікні пристрою Cloud. На лівій панелі натисніть **Cable** під **CONNECTIONS**. У першому вікні, який випав, виберіть Coaxial, а у другому виберіть Ethernet, після чого натисніть кнопку **Add**, щоб додати їх, як показано на рисунку 1.8.

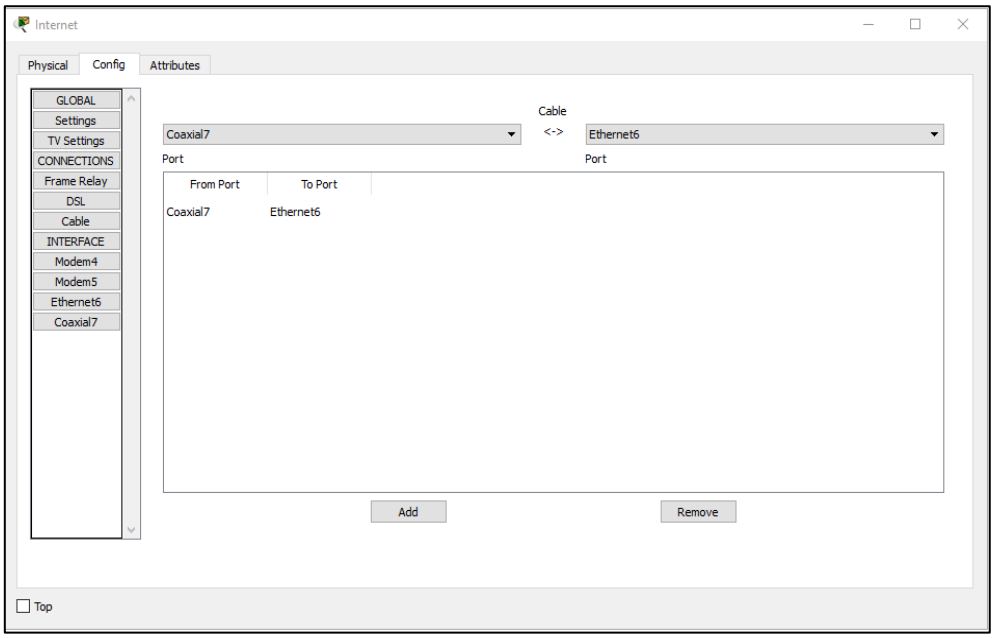

Рисунок 1.8 – Панель керування мережею

d. Визначте тип постачальника.

Перебуваючи на вкладці **Config**, натисніть кнопку Ethernet у полі **INTERFACE** на лівій панелі. У вікні налаштування Ethernet виберіть **Cable** як мережу постачальників, як показано на рисунку 1.9.

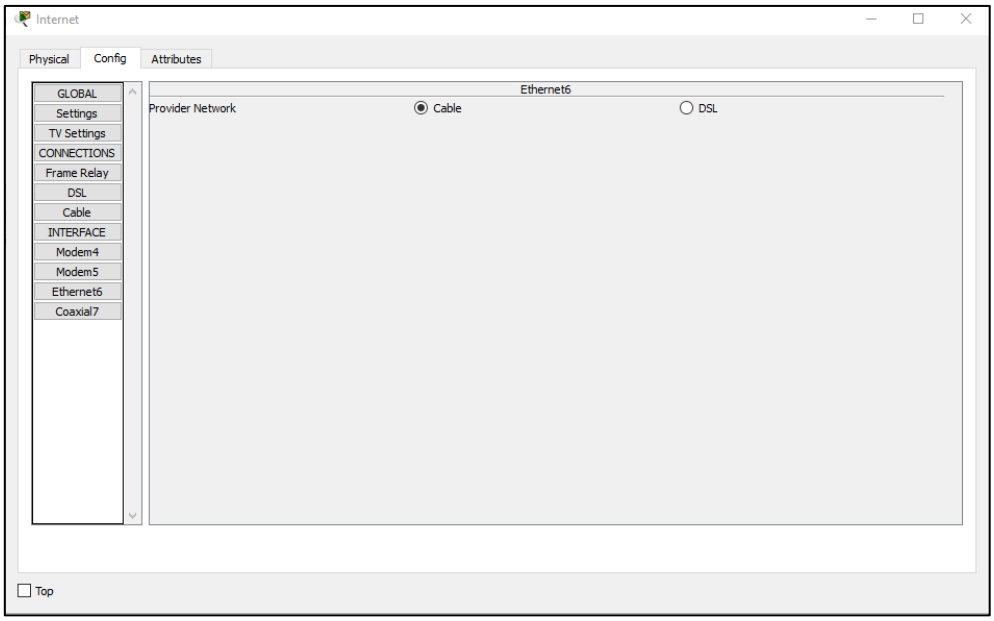

Рисунок 1.9 – Панель параметрів

Крок 5. Налаштуйте сервер Cisco.com.

a. Налаштуйте сервер Cisco.com як сервер DHCP.

Натисніть іконку сервера Cisco.com на робочому середовищі Packet Tracer Logical та виберіть вкладку **Services**.

Виберіть **DHCP** зі списку **SERVICES** на лівій панелі.

У вікні налаштування DHCP налаштуйте DHCP, як показано на рисунку, з такими настройками.

- Натисніть **Включити**, щоб включити службу DHCP.
- Назва: DHCPpool.
- Шлюз за замовчуванням: 208.67.220.220.
- За замовчуванням.
- Стартова IP-адреса: 208.67.220.1.
- Маска підмережі 255.255.255.0.
- Максимальна кількість користувачів: 50.

Натисніть **Add**, щоб додати pool.

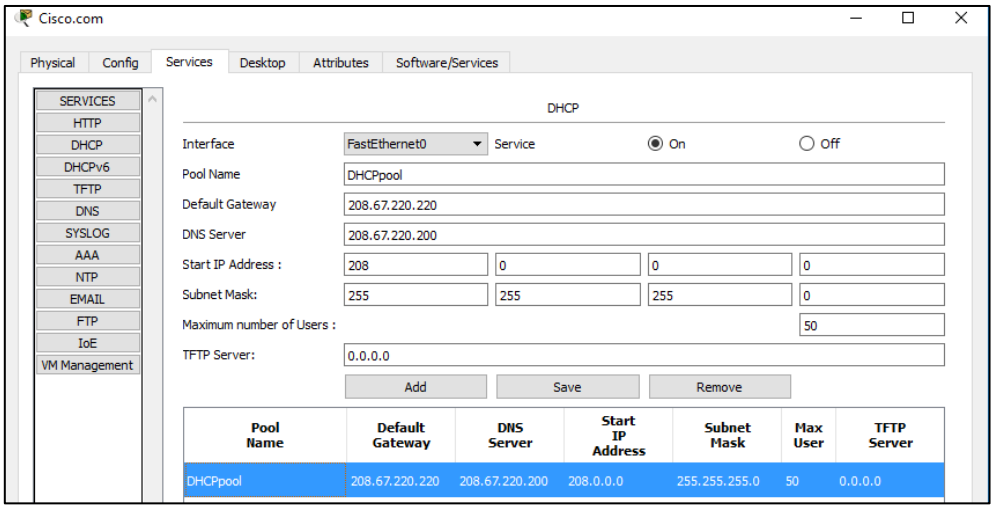

Рисунок 1.10 – Панель Cisco

b. Налаштуйте сервер Cisco.com як DNS-сервер, щоб указати назву домену для розв’язання адреси IPv4.

На вкладці **Служби** виберіть **DNS** із **SERVICES,** перелічених на лівій панелі.

Налаштуйте службу DNS за допомогою наступних параметрів, як показано на рисунку.

- Натисніть **On,** щоб включити службу DNS.
- Ім’я: Cisco.com.
- Тип: A Record.
- Адреса: 208.67.220.220.

Натисніть **Add,** щоб додати налаштування служби DNS.

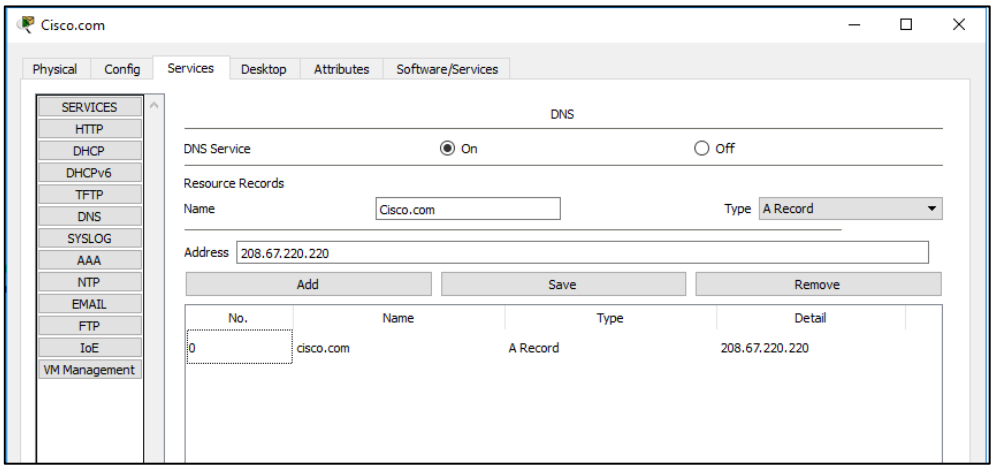

Рисунок 1.11 – Панель Cisco сервіси

c. Налаштувати глобальні налаштування сервера Cisco.com.

Виберіть вкладку **Config**.

Натисніть на **Налаштування** на лівій панелі.

Налаштуйте глобальні налаштування сервера так.

- Виберіть **Static**.
- Шлюз: 208.67.220.1.
- DNS-сервера: 208.67.220.220.

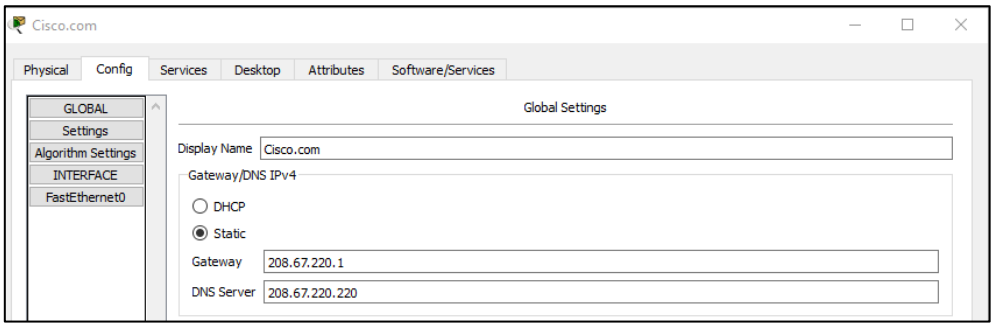

Рисунок 1.12 – Панель Cisco конфігурації

d. Налаштуйте параметри інтерфейсу сервера Cisco.com FastEthernet0.

Click on **FastEthernet** in left pane of the **Config** tab.

Налаштуйте параметри інтерфейсу FastEthernet сервера так.

- Виберіть **Static** під IP Configuration.
- IP-адреса: 208.67.220.220.
- Маска підмережі: 255.255.255.0.

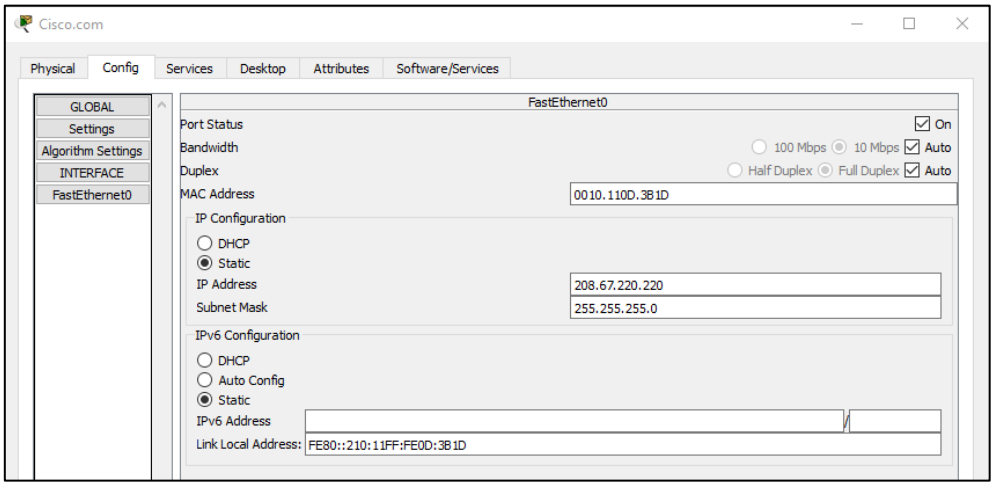

Рисунок 1.13 – Панель Cisco

Частина 3. Перевірте підключення.

Крок 1. Оновіть налаштування IPv4 на ПК.

a. Перевірте, чи ПК отримує інформацію про конфігурацію IPv4 з DHCP.

Натисніть **PC** на робочому середовищі Packet Tracer Logical, а потім виберіть вкладку **Desktop** вікна налаштування ПК.

Натисніть іконку **Command Prompt.**

У командному рядку оновіть параметри IP, виписавши команди **ipconfig/release,** а потім **ipconfig/update.** Результат має показати, що на ПК є IP-адреса в діапазоні 192.168.0.x, маска підмережі, шлюз за замовчуванням і адреса DNS-сервера, як показано на рисунку.

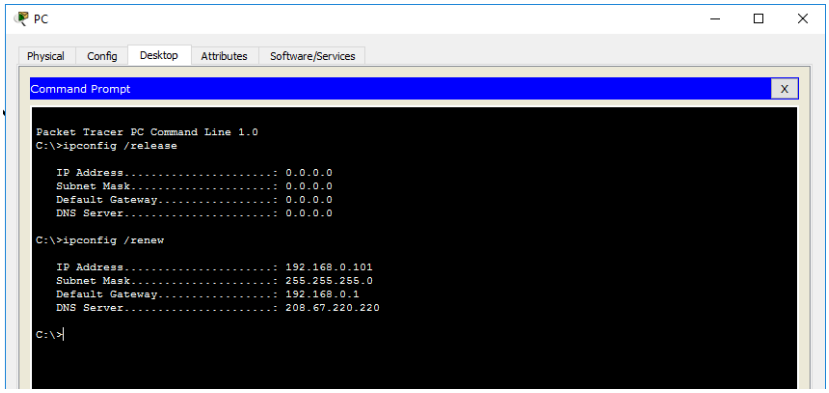

Рисунок 1.14 – Командний рядок ПК

b. Перевірте підключення до сервера Cisco.com з ПК.

У командному рядку видається команда **ping Cisco.com.** Для повернення ping може знадобитися кілька секунд. Необхідно отримати чотири відповіді, як показано на рисунку.

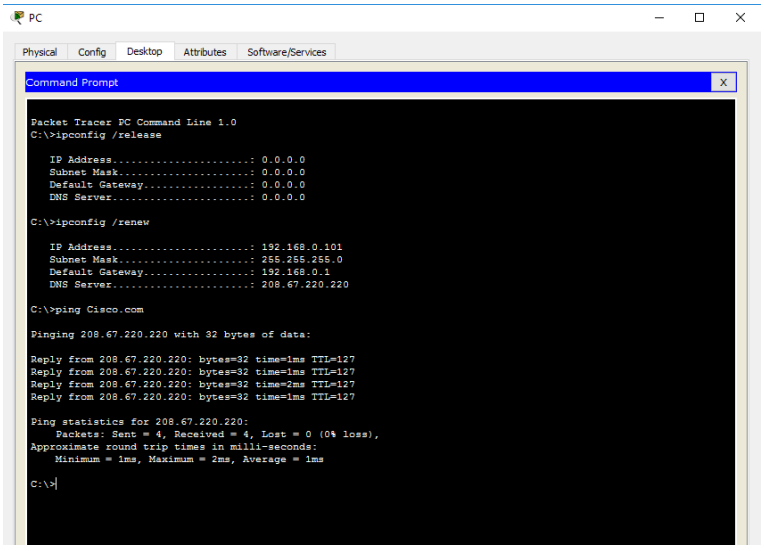

Рисунок 1.15 – Командний рядок ПК

Частина 4. Збережіть файл і закрийте Packet Tracer.

Крок 1. Збережіть файл як Packet Tracer Activity File (*.pkt).

Щоб зберегти завершену мережу, натисніть на **Файл** у смужці Packet Tracer, а потім у спадному меню виберіть **Save As...** У вікні «Зберегти файл» виберіть провідник, щоб зберегти файл і надати йому відповідне ім’я. Значення типу «Зберегти як» за замовчуванням відносяться до файлу Packet Tracer (*.pkt). Натисніть **Save,** щоб зберегти файл.

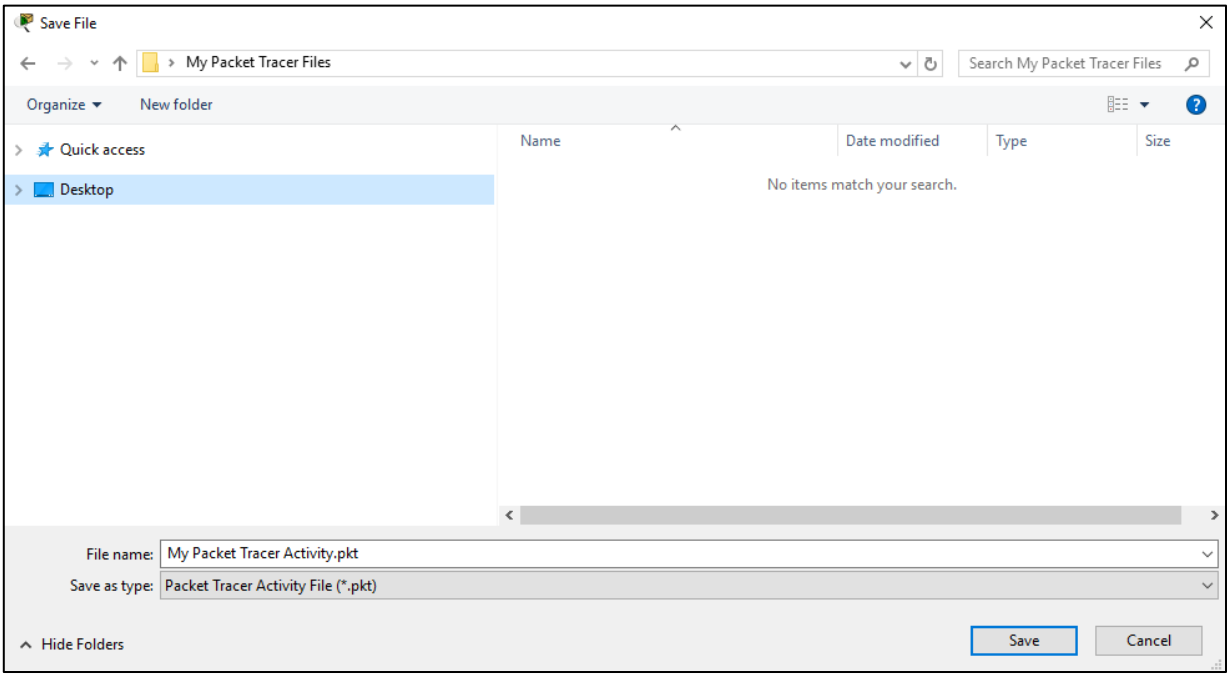

Рисунок 1.16 – Файлова система ПК

Крок 2. Закрийте Packet Tracer.

Щоб закрити Packet Tracer, ви можете або натиснути кнопку «**X**» у верхньому правому куті вікна Packet Tracer, або клацнути на **Вихід** у спадному меню Файл.

1.3 Завдання для самостійного виконання

Ознайомитися з порядком виконання роботи й виконати аналогічні операції для діагностики мережі за даними, наведеними у таблиці 1.2. Варіант обирати згідно з алфавітною позицією у списку групи.

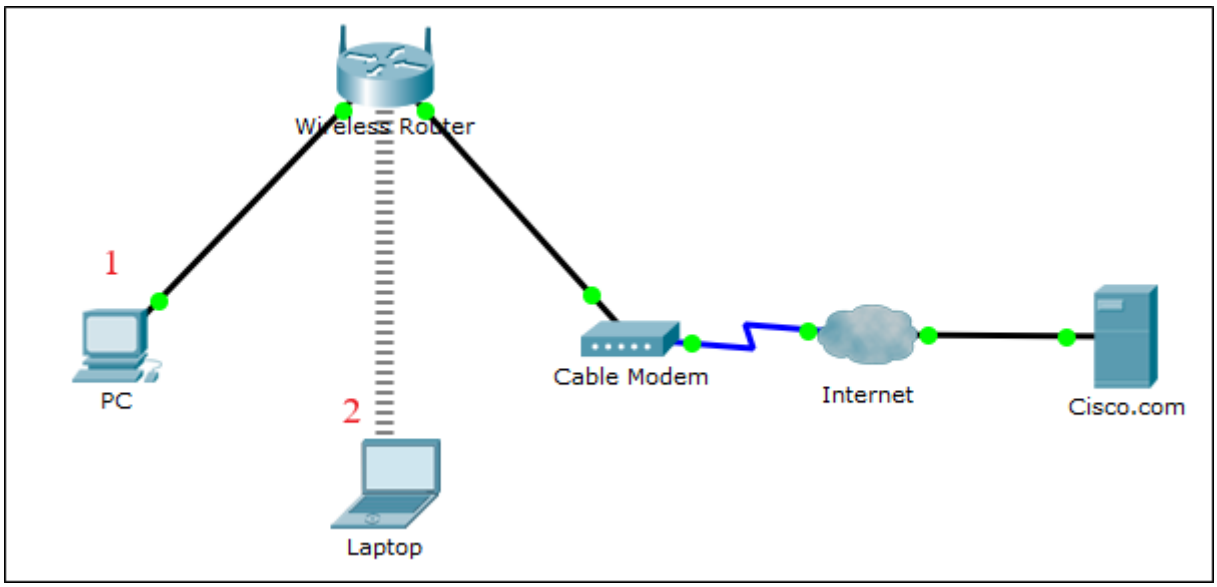

Рисунок 1.16 – Топологія модельованої мережі

Таблиця 1.1 – Таблиця адресації

Пристрій	Інтерфейс	IP-адреса	маска підмережі	Шлюз за замовчуванням
ПК	Ethernet0	DHCP		192.168.0.1
Wireless Router	LAN	192.168.0.1	255.255.255.0	
	Internet	DHCP		
Cisco.com Server	Ethernet0	208.67.220.220	255.255.255.0	
Ноутбук	Wireless0	DHCP		

Таблиця 1.2 – Варіанти індивідуальних завдань

Вар.	Вихідні дані	Вар.	Вихідні дані
1	У топології до роутера приєднано 7 ПК та 7 ноутбуків	16	У топології до роутера приєднано 8 ПК та 6 ноутбуків
2	У топології до роутера приєднано 6 ПК та 8 ноутбуків	17	У топології до роутера приєднано 9 ПК та 5 ноутбуків
3	У топології до роутера приєднано 5 ПК та 9 ноутбуків	18	У топології до роутера приєднано 10 ПК та 4 ноутбуків
4	У топології до роутера приєднано 4 ПК та 10 ноутбуків	19	У топології до роутера приєднано 11 ПК та 3 ноутбуків
5	У топології до роутера приєднано 3 ПК та 11 ноутбуків	20	У топології до роутера приєднано 12 ПК та 2 ноутбуків
6	У топології до роутера приєднано 4 ПК та 12 ноутбуків	21	У топології до роутера приєднано 13 ПК та 1 ноутбуків

7	У топології до роутера приєднано 3 ПК та 13 ноутбуків	22	У топології до роутера приєднано 14 ПК та 2 ноутбуків
8	У топології до роутера приєднано 2 ПК та 14 ноутбуків	23	У топології до роутера приєднано 15 ПК та 1 ноутбуків
9	У топології до роутера приєднано 1 ПК та 15 ноутбуків	24	У топології до роутера приєднано 16 ПК та 0 ноутбуків
10	У топології до роутера приєднано 7 ПК та 0 ноутбуків	25	У топології до роутера приєднано 11 ПК та 0 ноутбуків
11	У топології до роутера приєднано 0 ПК та 8 ноутбуків	26	У топології до роутера приєднано 10 ПК та 2 ноутбуків
12	У топології до роутера приєднано 5 ПК та 0 Ноутбуків	27	У топології до роутера приєднано 13 ПК та 1 ноутбуків
13	У топології до роутера приєднано 0 ПК та 10 ноутбуків	28	У топології до роутера приєднано 14 ПК та 0 ноутбуків
14	У топології до роутера приєднано 3 ПК та 0 ноутбуків	29	У топології до роутера приєднано 0 ПК та 3 ноутбуків
15	У топології до роутера приєднано 0 ПК та 12 ноутбуків	30	У топології до роутера приєднано 12 ПК та 0 ноутбуків

1.4 Зміст звіту

1. Короткий опис змісту виконання роботи.
2. Скріншоти й аналітичні дані проведеної роботи.
3. Висновки.

1.5 Контрольні запитання

1. Як додати мережні пристрої до робочої області?
2. Як змінити відображення назв мережних пристроїв?
3. Як додати фізичну проводку між пристроями на робочій області?
4. Як налаштувати бездротовий маршрутизатор?
5. Як налаштувати ноутбук?

Лабораторна робота № 2

Тема. Моделювання архітектури IoT-системи

Мета: навчитися працювати з IoT-речами в системі Cisco Packet Tracer.

2.1 Короткі теоретичні відомості

Розумний дім (розумний будинок/smart home, digital house) – система домашніх пристроїв, здатних виконувати дії та розв'язувати певні повсякденні завдання без участі людини. Функціонально пов'язуються між собою усі електроприлади будівлі, якими можна керувати централізовано – з пульта-дисплею. Прилади можуть бути під'єднані до комп'ютерної мережі, що дозволяє керувати ними за допомогою ПК та надає віддалений доступ до них через Інтернет. Завдяки інтеграції інформаційних технологій у домашні умови, усі системи та прилади узгоджують виконання функцій між собою, порівнюючи задані програми та зовнішні показники (обстановки).

Для визначення високотехнологічних особливостей приміщення також вживають терміни: intelligent building, smart-house, digital home.

Розумний дім створюється за допомогою професійного проєктування та програмування компаніями, що займаються розробкою проєктів smart-home. Програми, що вводяться до алгоритмів multi-room розумного дому, розраховані на певні потреби мешканців і ситуації, пов'язані зі зміною середовища або безпекою. Особливістю smart-home є керування за допомогою пульта, на якому людина може натиснути одну-єдину клавішу для створення певної обстановки. До того ж, сама система мульти-рум аналізує навколишню ситуацію та параметри у середині приміщення та, керуючись власними висновками, виконує задані користувачем команди із відповідними налаштуваннями. Окрім того, електронні побутові прилади, встановлені у розумному будинку, можуть бути об'єднані в домашню Universal Plug'n'Play – мережу із виходом до Інтернету.

2.2 Порядок виконання роботи

Частина 1. Дослідіть інтелектуальну домашню мережу.

Крок 1. Відкрийте файл Smart_Home_Network.pkt.

Відкрийте файл **Smart_Home_Network.pkt**.

Збережіть файл на своєму комп'ютері.

Частина 2. Дослідіть інтелектуальну домашню мережу.

Дослідіть кінцеві пристрої IoT.

У лівому нижньому кутку Packet Tracer знайдіть і натисніть значок **[End Devices]** у верхньому рядку, а значок **[Home]** у нижньому рядку **Вибір типу пристрою**.

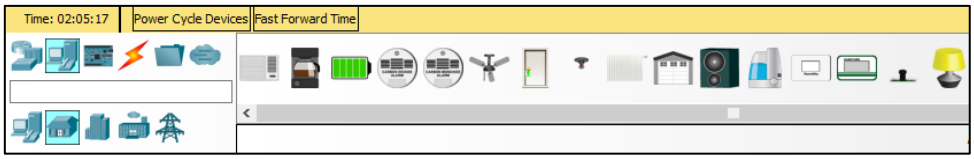

Рисунок 2.1 – Панель Інтернет-речей

У нижній частині вікна Packet Tracer поле **Device-specific Selection** відображає багато доступних пристроїв Smart Home IoT.

Перемістіть указівник миші на кожний пристрій і помітите, що описане ім'я пристрою відображається в нижній частині вікна **Device-specific Selection.** Подивіться на кожний тип пристрою.

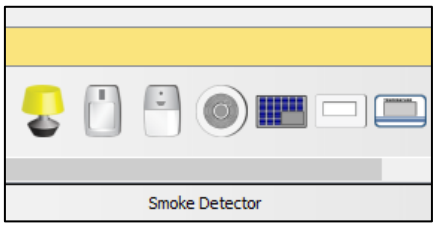

Рисунок 2.2 – Панель Інтернет-речей

Дослідіть мережу «Розумного будинку».

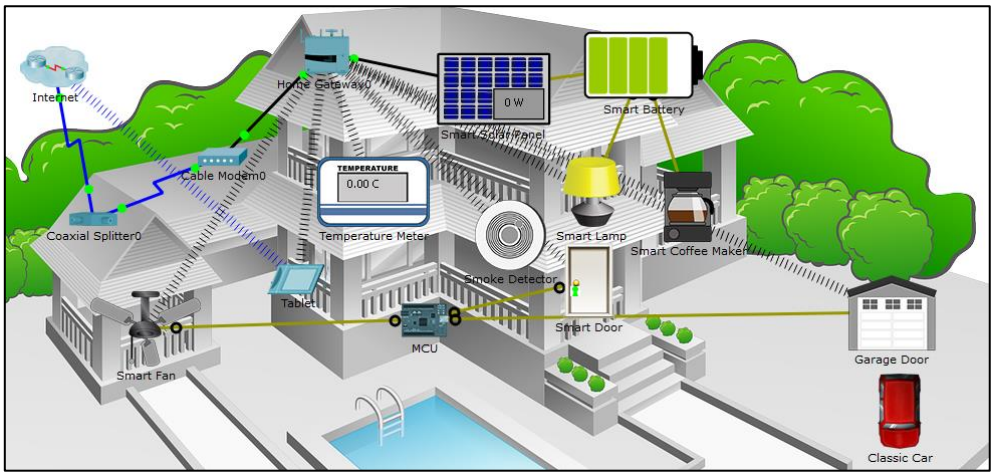

Рисунок 2.3 – Модель «Розумного будинку»

У робочому середовищі є попередньо побудована розумна домашня мережа, яка складається з багатьох провідних та бездротових пристроїв IoT та пристроїв мережної інфраструктури.

Коли ви наводите курсор на пристрій, наприклад Smart Fan, відкривається інформаційне вікно, що містить основну інформацію про мережу на цьому пристрої.

Рисунок 2.3 – Параметри «Розумного будинку»

Щоб увімкнути або активувати пристрій, просто утримуйте клавішу **Alt** на клавіатурі, а потім **клацніть лівою кнопкою миші** на пристрої. Спробуйте це на кожному зі смарт-пристроїв, щоб спостерігати, що вони роблять.

Інтелектуальна домашня мережа складається з інфраструктурних пристроїв, таких як домашній шлюз.

Натисніть на **Home Gateway,** щоб відкрити вікно Home Gateway.

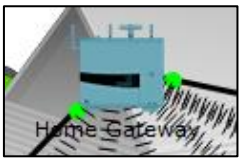

Рисунок 2.4 – Іконка Home Gateway

Вкладка Physical вибрана за замовчуванням та показує зображення домашнього шлюзу.

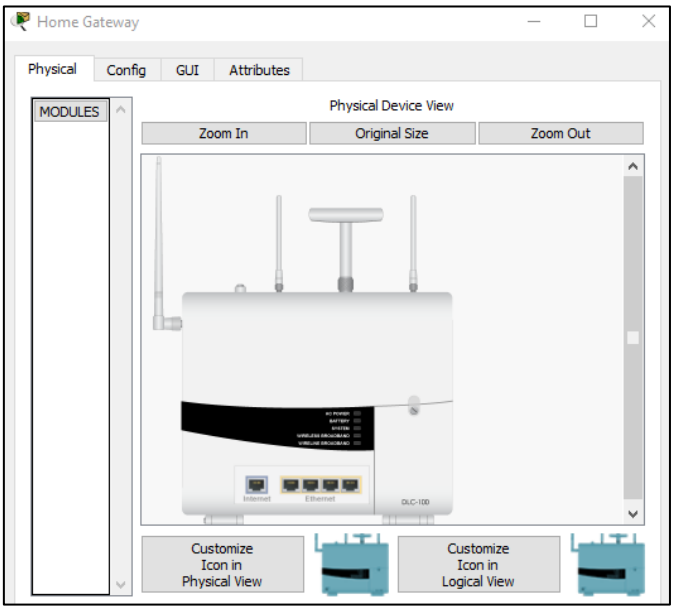

Рисунок 2.5 – Панель Home Gateway

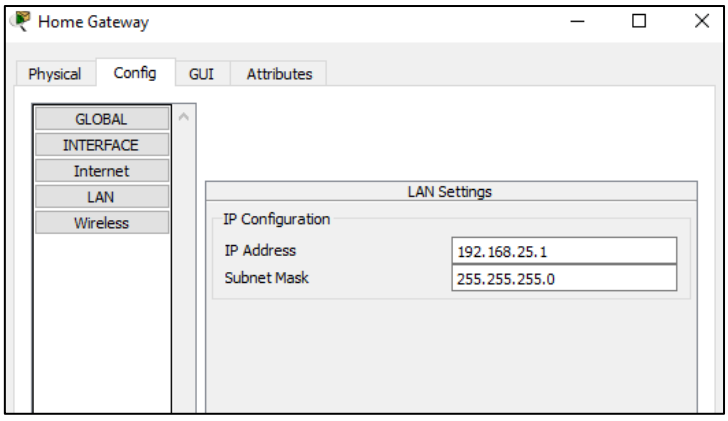

Рисунок 2.6 – Параметри у панелі Home Gateway

Натисніть **Wireless** на лівій панелі, щоб переглянути параметри бездротового зв’язку домашнього шлюзу.

Запишіть SSID домашньої мережі та WPA2-PSK Pass Phrase для подальшого використання.

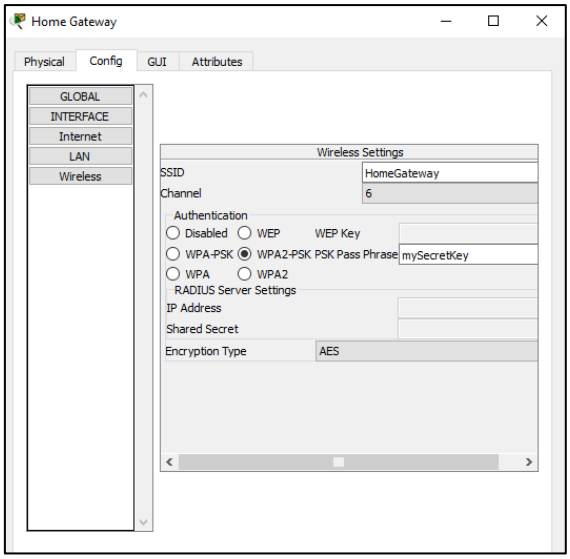

Рисунок 2.7 – Панель Home Gateway, налаштування

Закрийте вікно Home Gateway.

Потім натисніть значок пристрою **Tablet,** щоб відкрити вікно планшета.

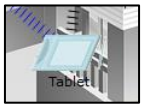

Рисунок 2.8 – Планшет

У вікні планшета виберіть вкладку **Desktop** та натисніть значок веб-браузера.

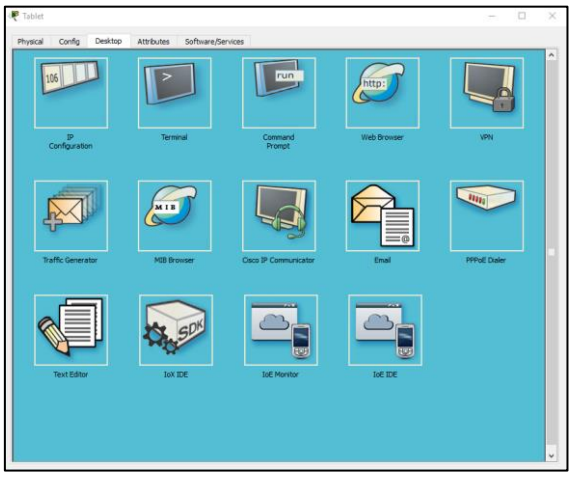

Рисунок 2.9 – Меню планшета

У вікні **Web Browser** введіть IP-адресу домашнього шлюзу **192.168.25.1** у поле URL-адреси та натисніть **Go.** На екрані входу в головний шлюз уведіть admin як для імені користувача, так і для пароля та натисніть **Надіслати.**

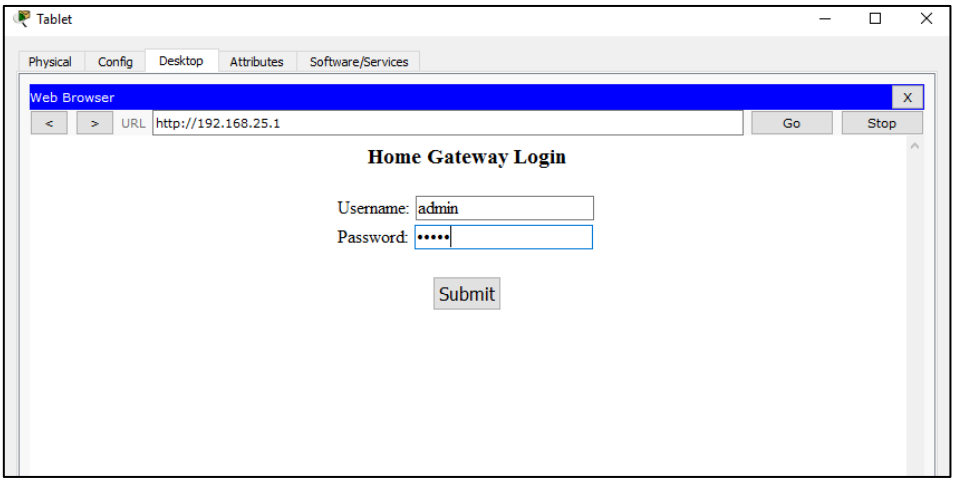

Рисунок 2.10 – Браузер планшета

Після підключення до веб-інтерфейсу домашнього шлюзу з’явиться список усіх підключених пристроїв IoT.

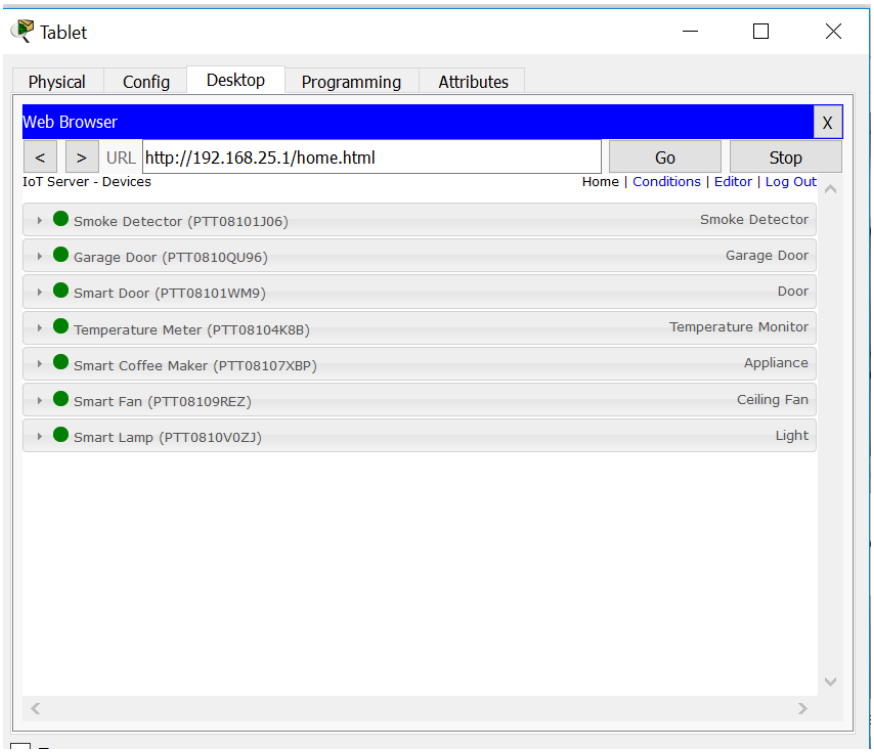

Рисунок 2.11 – Список під’єднаних IoT-пристроїв

Коли ви натискаєте на пристрій у списку, відображається статус та установки цього пристрою.

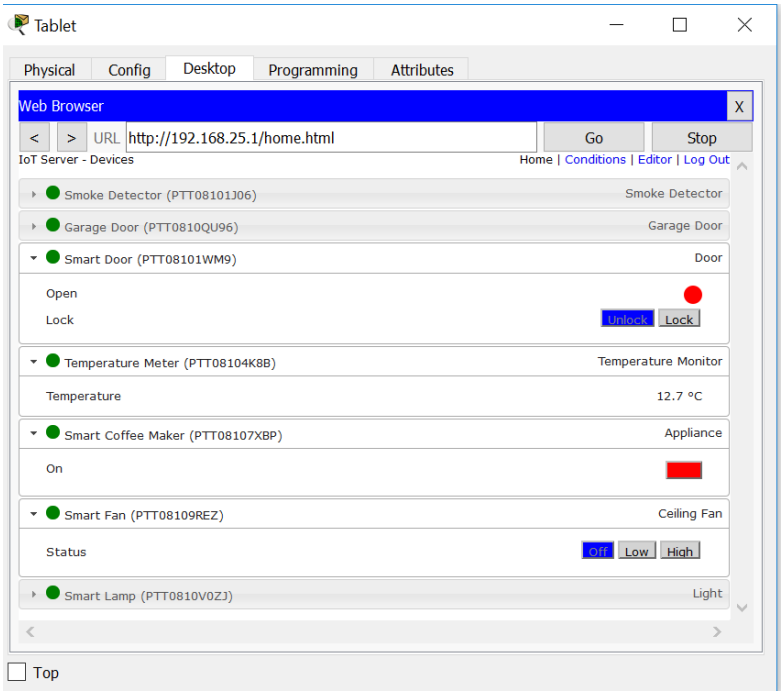

Рисунок 2.12 – Стани IoT речей

Закрийте вікно планшета.

Додайте Wired IoT пристрої до Smart Home Network.

Крок 1. Підключіть пристрій до мережі.

У вікні **Device-specific Selection** натисніть значок **}Lawn Sprinkler,** а потім натисніть робочу область, де ви хочете розташувати **Lawn Sprinkler**.

Підключіть Fire Sprinkler до Home Gateway.

In the **Device-Type Selection** box, click the **[Connections]** icon (this looks like a lightning bolt). Натисніть на тип з'єднувача Copper Straight Through в полі **Device-Specific Selection.** Потім натисніть на значок Sprinkler і підключіть один кінець кабелю до FastEthernet0 Sprinkler. Далі натисніть значок Home Gateway і підключіть інший кінець кабелю до доступного Ethernet.

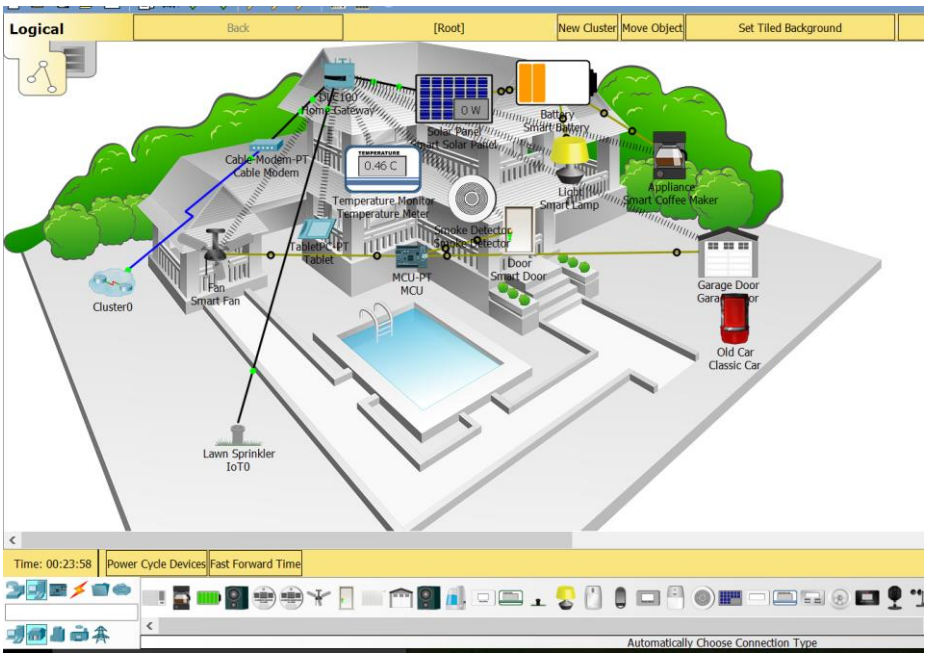

Рисунок 2.13 – Модель «Розумного будинку»

Крок 2. Налаштуйте розбризгувач для підключення до мережі.

а. Натисніть на **Lawn Sprinkler** у робочому середовищі, щоб відкрити вікно пристрою. Зверніть увагу, що прямо зараз назва розбризгувача є загальним IoT0.

Вікно пристрою відкриється на вкладці **Специфікація,** в якому міститься інформація про пристрій, який можна редагувати.

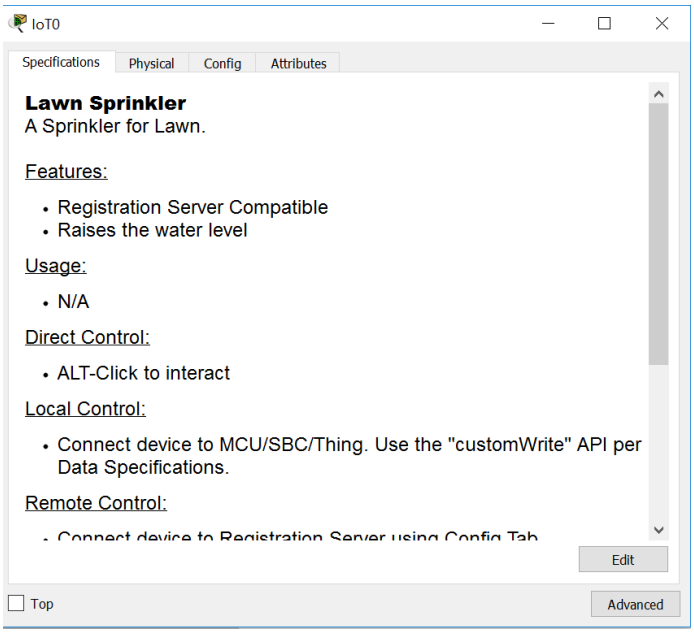

Рисунок 2.14 – IoT0 специфікація

b. Натисніть вкладку **Config,** щоб змінити налаштування конфігурації пристрою.

На вкладці «Конфігурація» внесіть такі зміни до **Налаштування** :

– Установіть **Display Name** до Sprinkler1 (ім’я вікна змінюється на Sprinkler1).

– Установіть IoT-сервер на **Home Gateway.**

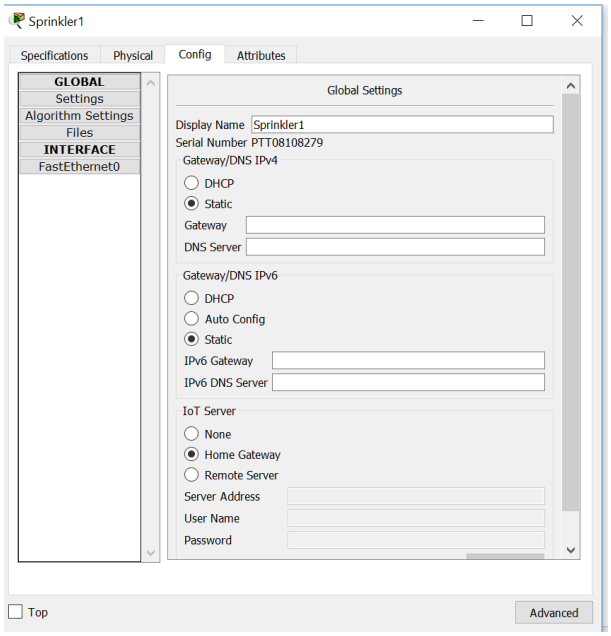

Рисунок 2.15 – Конфігурація мережі

Натисніть **FastEthernet0** та змініть **IP Configuration** на **DHCP**.

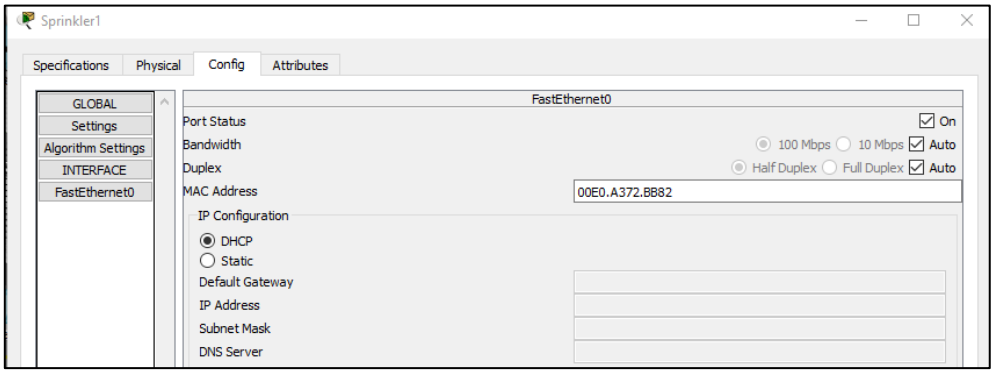

Рисунок 2.16 – Панель корнфігурації

Закрийте вікно **Sprinkler1.**

с. Перевірте, чи є розбризкувач у мережі.

Увійдіть у **домашній браузер** з планшета.

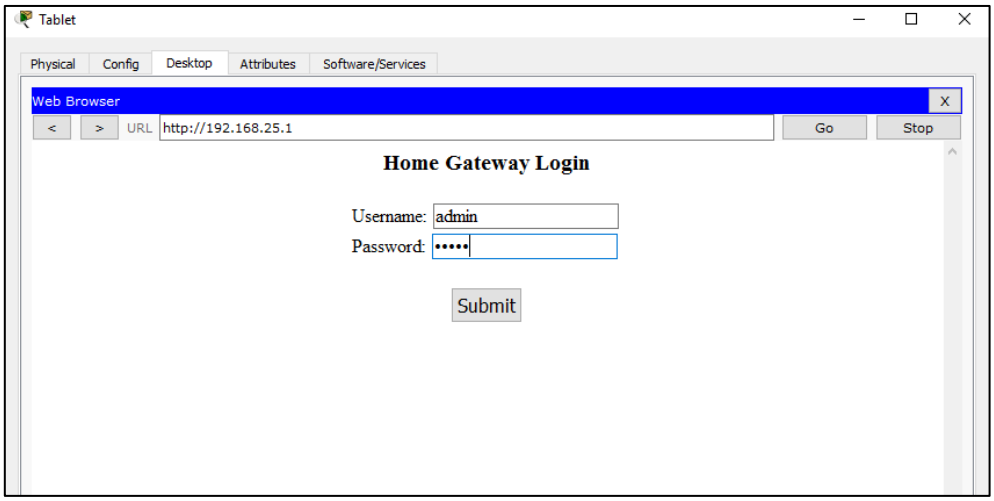

Рисунок 2.17 – Браузер планшета

Пристрій Sprinkler 1 тепер має з’явитися в списку IoT Server – Devices.

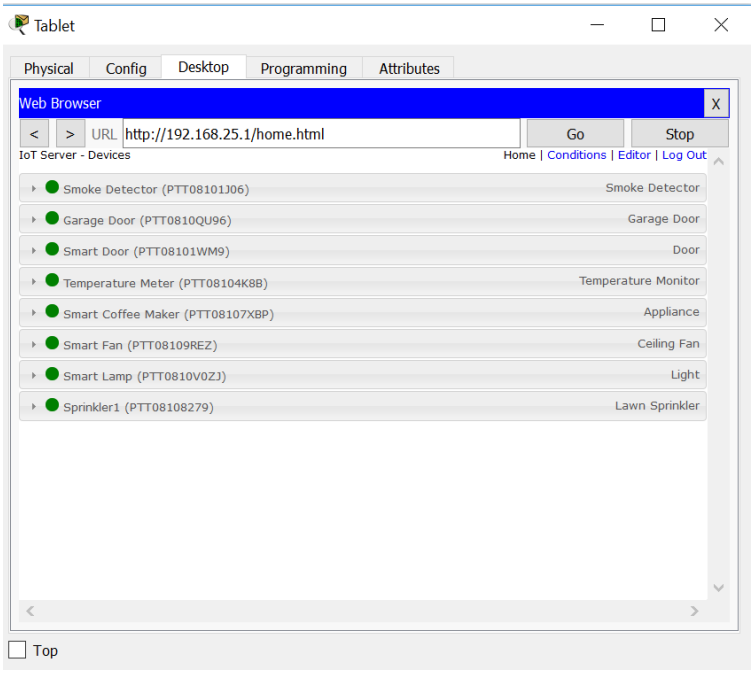

Рисунок 2.18 – Панель Home Gateway

Закрийте вікно планшета.

Крок 3. Експериментуйте, додаючи інші види пристроїв IoT до розумної домашньої мережі. Додайте бездротові пристрої IoT до Smart Home Network.

Крок 1. Додайте бездротовий пристрій до мережі.

У вікні **Device-specific Selection** натисніть значок **Wind Detector,** а потім натисніть на робочу область, де ви хочете розмістити **детектор вітру.**

Спеціальний вибір пристрою.

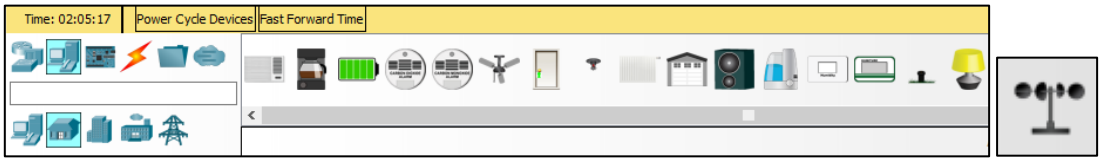

Рисунок 2.19 – Вибір детектору вітру

Додайте бездротовий модуль до детектора вітру.

Натисніть на **Wind Detector** у робочому середовищі, щоб відкрити вікно пристрою IoT. У нижньому правому куті вікна пристрою IoT натисніть кнопку **Додатково.** Зверніть увагу, що більше з'являється у верхній частині вікна. Натисніть **вкладку введення/виведення Config**.

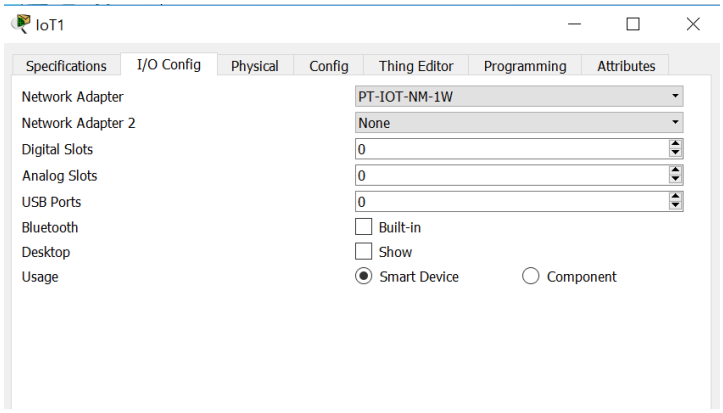

Рисунок 2.20 – Конфігурація IoT1

Змініть випадаючий список **Мережний адаптер** у **PT-IOT-NM-1W,** який є бездротовим адаптером.

Налаштуйте детектор вітру для бездротової мережі.

Натисніть вкладку **Config.**

Change the **Display Name** to **Wind_Detector** and change the **IoT Server** to **Home Gateway**.

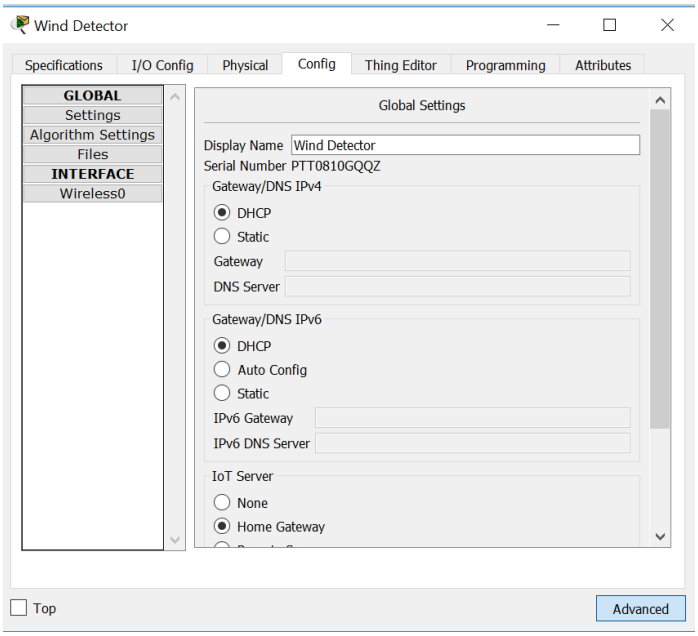

Рисунок 2.21 – Панель керування доданого датчика

Потім нати Wireless0 на лівій панелі. Потім клацніть Wireless0 на лівій панелі. Змініть тип аутентифікації на **WPA2-PSK** і в поле PSC Pass Phrase **mySecretKey.** Це налаштування бездротової мережі домашнього шлюзу, записаного у частині 1.

Бездротовий зв’язок має бути сформований між детектором вітру та домашнім шлюзом.

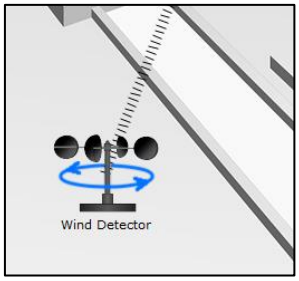

Рисунок 2.22 – Формування бездротового зв’язку

d. Перевірте детектор вітру в мережі.

Увійдіть у **домашній браузер** з планшета.

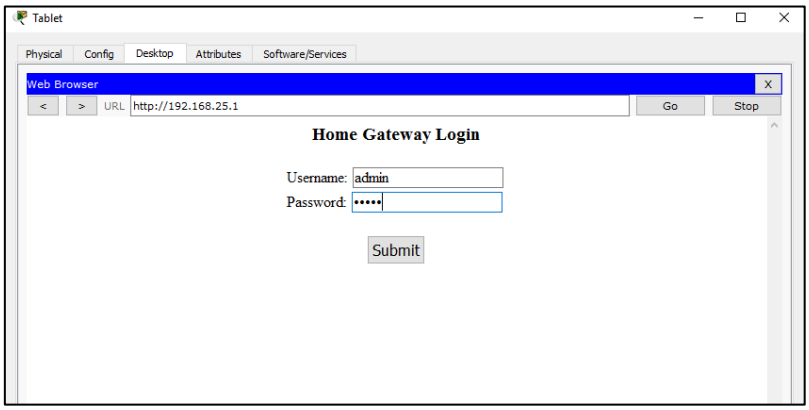

Рисунок 2.23 – Вхід у систему IoT

Вітровий детектор тепер має з’явитися в списку IoT Server – Devices.

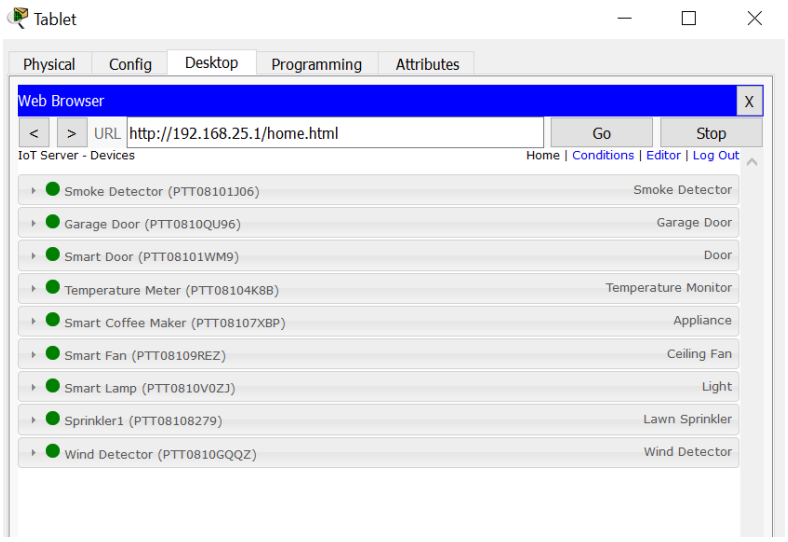

Рисунок 2.24 – Меню IoT у планшеті

2.3 Завдання для самостійного виконання

Ознайомитися з порядком виконання роботи й додати до топології будинку бездротові пристрої, наведені в таблиці 2.1. Варіант обирати згідно з алфавітною позицією у списку групи.

Таблиця 2.1 – Варіанти індивідуальних завдань

Вар.	Вихідні дані	Вар.	Вихідні дані
1	Детектор диму й датчик освітленості	16	Детектор диму та двері
2	Мікрохвильовку й вентилятор	17	Мікрохвильовку й вікно
3	Двері та батарею	18	Двері й мікрохвильовку
4	Музичну систему й датчик температури	19	Музичну систему й кавоварку
5	Лампу й вікно	20	Лампу та ПК
6	Детектор диму й вікно	21	Детектор диму й мікрохвильовку
7	Мікрохвильовку й лампу	22	Мікрохвильовку й лампу
8	Двері й мікрохвильовку	23	Двері й вікно
9	Датчик вітру й ПК	24	Датчик вітру та ПК
10	Лампу й кавоварку	25	Лампу й кавоварку
11	Планшет і кавоварку	26	Планшет і датчик температури
12	Планшет і мікрохвильовку	27	Планшет і батарею
13	Планшет і вікно	28	Планшет і вентилятор
14	ПК і вікно	29	ПК і вентилятор
15	ПК і двері	30	ПК і датчик освітленості

2.4 Зміст звіту

1. Короткий опис змісту виконання роботи.
2. Скріншоти та аналітичні дані проведеної роботи.
3. Висновки.

2.5 Контрольні запитання

1. Як підключити пристрій до мережі?
2. Як налаштувати підключений до мережі пристрій?
3. Як додати бездротовий пристрій до мережі?
4. Як налаштувати бездротовий пристрій для роботи в мережі?
5. Що таке «Розумний будинок» і в чому полягає суть такої концепції?

Лабораторна робота № 3

Тема. Використання Arduino Uno як IoT-платформу

Мета: ознайомлення з віртуальними моделями пристроїв IOT у програмному середовищі Proteus, отримання практичних навичок швидкого налагодження програм для AVR-мікроконтролерів у середовищі Proteus.

3.1 Короткі теоретичні відомості

Розробником пакету Proteus є фірма Labcenter Electronics у Великобританії. Сайт розробника можна знайти за посиланням http://www.labcenter.co.uk/.

Відмінність Proteus від аналогічних за призначенням пакетів програмних засобів, наприклад, Electronics Workbench, Multisim, MicroCap, Tina та інших, у розвиненій системі симуляції роботи (інтерактивного налагодження в режимі реального часу і покроково) для різних сімейств мікроконтролерів (МК): 8051, PIC (Microchip), AVR (Atmel) та інших. У Proteus наявні бібліотеки компонентів, у тому числі й периферійних пристроїв: світлодіодні й рідиннокристалічні індикатори, температурні датчики, годинники реального часу – RTC, інтерактивні елементи введення-виведення: кнопки, перемикачі, віртуальні порти й віртуальні вимірювальні прилади, інтерактивні графіки, які не завжди присутні в інших аналогічних програмах.

Proteus VSM складається з двох самостійних програм: ISIS і ARES. ARES – це трасувальник друкованих плат. Основною програмою є ISIS, у ній передбачений гарячий зв’язок з ARES для розведення друкованих доріжок плати.

Загалом, робота в середовищі моделювання Proteus ISIS складається з таких пунктів: 1) вибір електричної принципової схеми й задання її параметрів функціонування; 2) розміщення віртуальних приладів аналогічно до принципової схеми, з урахуванням умовностей моделювання, та вибір режимів їх роботи; 3) симуляція роботи ланцюгів чи проведення їх спеціалізованого аналізу; 4) перевірка працездатності й налагодження програм, завантажуваних до МК.

Для того, щоб сформувати уявлення про можливості Proteus, як середовища моделювання роботи електронних ланцюгів, необхідно розглянути деякі прості приклади, які за замовчуванням наявні у файлах програми. Також слід відзначити VSM Studio IDE, у якому була реалізована підтримка наборів інструментів Arduino AVR, що дозволяє розробляти прототипи проєктів Arduino середині середовища Proteus, нова функція проєкту дуже корисна, оскільки проєкт можна легко створити для різних Arduino-платформ.

3.2 Порядок виконання роботи

1. Вимірювання температури за допомогою датчика DS18B20.

Розглянемо процеси, що відбуваються під час вимірюванні температури. Відкриємо файл з набору прикладів: File → Open Sample Project → VSM for AVR → Arduino → Arduino DS18B20 OneWire (рисунок 3.1).

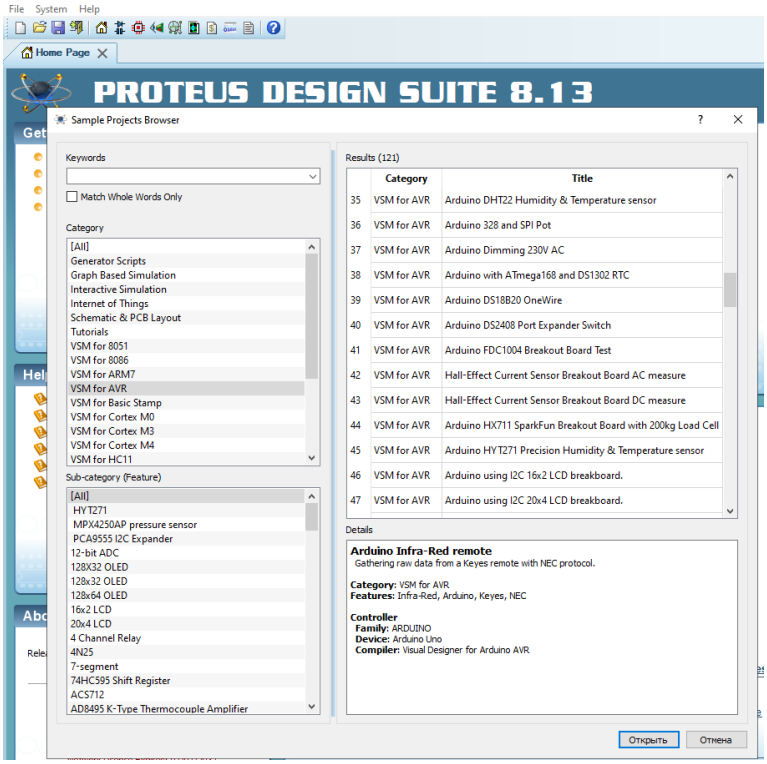

Рисунок 3.1 – Вибір базового проєкту в середовищі Proteus

На рисунку 3.2 наведено схему моделювання цифрового термометра, що базується на МК ATmega328 та цифровому датчику температури DS18B20 фірми Dallas Semiconductor. Виведення інформації у схемі здійснюється на

семисегментний індикатор. Мікросхема DS18B20 забезпечує здійснення вимірювань температури за шкалою Цельсія.

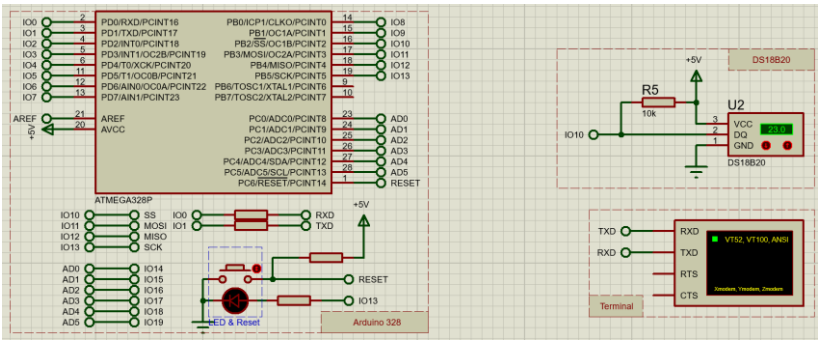

Рисунок 3.2 – Схема моделювання зміни температури

Мікросхема DS18B20 підключається до МК через однодротову шину, що вимагає тільки однієї лінії даних для взаємодії з МК. Вона має робочий діапазон температур від – 55 °С до + 125 °С і точність ± 0,5 °С у діапазоні від – 10 °С до + 85 °С. Модель термометра DS18B20 дозволяє задавати температуру термодатчика.

За допомогою вбудованого в Proteus осцилографа можна отримати осцилограму інформаційного обміну мікроконтролера з датчиком. Тимчасова діаграма сигналів (за протоколом 1-Wire) надана на рисунку 3.3.

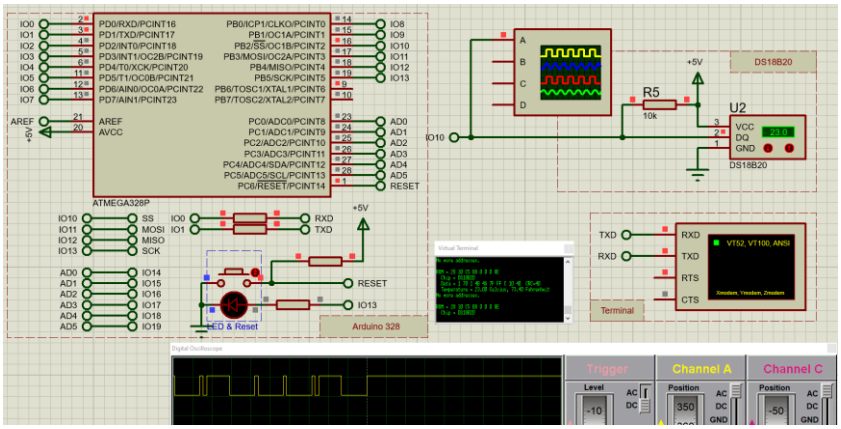

Рисунок 3.3 – Сигнал на однодротовій шині датчика при 25 °С

Значення напруги можна визначити за кольором міток на всіх виводах компонентів (лог. 1 – червоний колір, лог. 0 – синій, непідключений вивід – сірий колір).

2. Налаштування мікроконтролера.

Панель редагування властивостей МК, що зображена на рисунку 1.4, відкривається подвійним натисканням на нього лівою кнопкою миші або через праву кнопку і опцію Edit Properties (можна, виділивши МК, натиснути на клавіатурі комбінацію Ctrl + E).

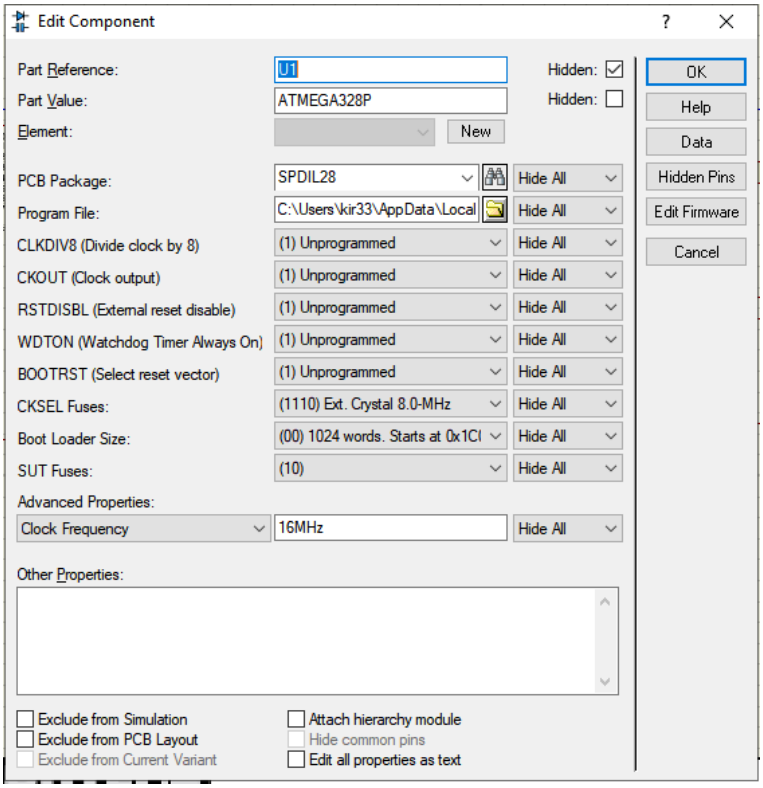

Рисунок 3.4 – Вікно налаштувань МК

У полі «Program File» потрібно вибрати файл типу «*.elf» (для налагодження) або «*.hex», що створюються під час компіляції програми в Arduino IDE, WinAVR чи інших програмних засобах. «*.hex» – файл «прошивки», що може бути завантажений у реальний МК. Під час його вибору можна налагоджувати пристрій без необхідності перегляду виконуваного коду.

Щоб застосувати файли програми з розширенням «*.hex», потрібно натиснути символ МК правою кнопкою миші, вибрати пункт «правка властивостей», далі в цьому пункті вибрати «Program file» і ввести шлях до файлу або просто натиснути значок «відкрити папку» і вказати потрібний файл. Після цього ніяких додаткових дій робити не потрібно, крім натискання кнопки «ОК».

Clock Frequency – визначає будь-яку частоту тактування мікроконтролера під час симуляції, яка відповідає частоті мікроконтролера в проєкті. За замовчанням, базова частота синхронізації – 8 МГц. В Arduino Uno, Nano частота синхронізації – 16 МГц. У деяких Arduino Mini – 8 МГц. У середовищі програмування Arduino IDE за замовчуванням обрана частота 16 МГц, але є можливість установлення частоти 8 МГц. Для зміни тактової частоти в списку CKSEL Fuses потрібно вибрати Ext.

Clock і в Advanced Properties замість (Default) необхідно встановити частоту в герцах.

Після цих налаштувань проєкт із прикладів Proteus функціонує згідно із заданою програмою. Це надає можливість використовувати приклади проєктів у Proteus demo для налагодження власних програм.

Proteus може використовувати компілятори, які встановлені в System – Compilers, зокрема:

– Arduino AVR – <Path IDE Arduino>;

– WinAVR – <Path WinAVR>.

3.3 Завдання для самостійного виконання

Ознайомитися з прикладами проєктів працюючих схем у програмному середовищі Proteus для Arduino:

– Arduino 4 Channel Relay;

– Arduino Cyrillic LCD;

– Arduino Motor Shield.

3.4 Зміст звіту

1. Короткий опис змісту виконання роботи.

2. Опис прикладів симуляції в Proteus:

– принцип роботи;

– приклади осцилограм;

– дані обміну в терміналі;

– оцінювання використовуваних ресурсів (пам’яті програм і даних, виводів).

3. Висновки.

3.5 Контрольні запитання

1. Який порядок створення проєкту в Proteus?
2. Як відкрити приклад для AVR МК?
3. Як визначити розмір програми в пам’яті МК?
4. Що таке «*.hex» файл? Як його завантажити в МК?
5. Як вибрати віртуальний інструмент?

Лабораторна робота № 4

Тема. Побудова простого IoT-пристрою на платформі Node MCU

Мета: навчитися працювати з Arduino IDE, Node MCU та здобути досвід у розробці IoT-пристроїв.

4.1 Короткі теоретичні відомості

NodeMCU – це прошивка з відкритим вихідним кодом, для якої доступні проєкти прототипів плати з відкритим кодом. Назва «NodeMCU» поєднує в собі «вузол» і «MCU» (блок мікроконтролера). Термін «NodeMCU» стосується мікропрограми, а не відповідних наборів для розробки.

Як мікропрограмне забезпечення, так і проєкти прототипів плат є відкритими.

Прошивка використовує мову сценаріїв Lua. Прошивка заснована на проєкті eLua та побудована на Espressif Non-OS SDK для ESP8266. Він використовує багато відкритих проєктів, таких як lua-cjson і SPIFFS. Через обмеження ресурсів користувачам потрібно вибрати модулі, що відповідають їхньому проєкту, і створити мікропрограмне забезпечення відповідно до своїх потреб. Також реалізовано підтримку 32-розрядного ESP32.

Апаратне забезпечення для створення прототипів, зазвичай, являє собою друковану плату, що функціонує як дворядний пакет (DIP), який інтегрує контролер USB з меншою платою поверхневого монтажу, що містить MCU та антену. Вибір формату DIP дозволяє легко створювати прототипи на макетних платах. Спочатку дизайн базувався на модулі ESP-12 ESP8266, який є Wi-Fi SoC, інтегрованим з ядром Tensilica Xtensa LX106, широко використовуваним у додатках IoT (див. відповідні проєкти).

Існує дві доступні версії NodeMCU: версії 0.9 і 1.0, де версія 0.9 містить ESP-12, а версія 1.0 містить ESP-12E, де E означає «Enhanced».

NodeMCU був поданий 30 грудня 2013 року, коли Espressif Systems розпочала виробництво ESP8266. NodeMCU розпочався 13 жовтня 2014 року, коли Хонг зафіксував перший файл прошивки nodemcu на GitHub. Через два місяці проєкт розширився, щоб включити відкриту апаратну платформу, коли розробник Huang R зафіксував gerber-файл плати ESP8266 під назвою devkit v0.9. Пізніше, того ж місяця, Tuan PM переніс клієнтську бібліотеку MQTT з Contiki на платформу SoC ESP8266 і приєднався до проєкту NodeMCU, тоді NodeMCU зміг підтримувати протокол MQTT IoT, використовуючи Lua для доступу до брокера MQTT. Ще одне важливе оновлення було зроблено 30 січня 2015 року, коли Devsaurus переніс u8glib до проєкту NodeMCU, що дозволило NodeMCU легко керувати РК-дисплеями, екранами, OLED-дисплеями та навіть VGA.

Улітку 2015 року розробники покинули проєкт мікропрограми, і група незалежних учасників взяла на себе роботу. До літа 2016 року NodeMCU включав понад 40 різних модулів.

4.2 Порядок виконання роботи

Робота детально надана за посиланням її розробником Rui Santos [1].

Реле – це перемикач з електричним приводом, і, як і будь-який інший вимикач, його можна вмикати чи вимикати, пропускаючи струм чи ні. Ним можна керувати за допомогою низьких напруг, як-от 3,3 В, що забезпечуються GPIO ESP8266, і дозволяє нам контролювати високі напруги, наприклад 12 В, 24 В або напругу в мережі (230 В у Європі та 120 В у США).

1, 2, 4, 8, 16-канальні релейні модулі

Існують різні релейні модулі з різною кількістю каналів. Можна знайти релейні модулі з одним, двома, чотирма, вісьмома і навіть шістнадцятьма каналами. Кількість каналів визначає кількість виходів, якими ми зможемо керувати.

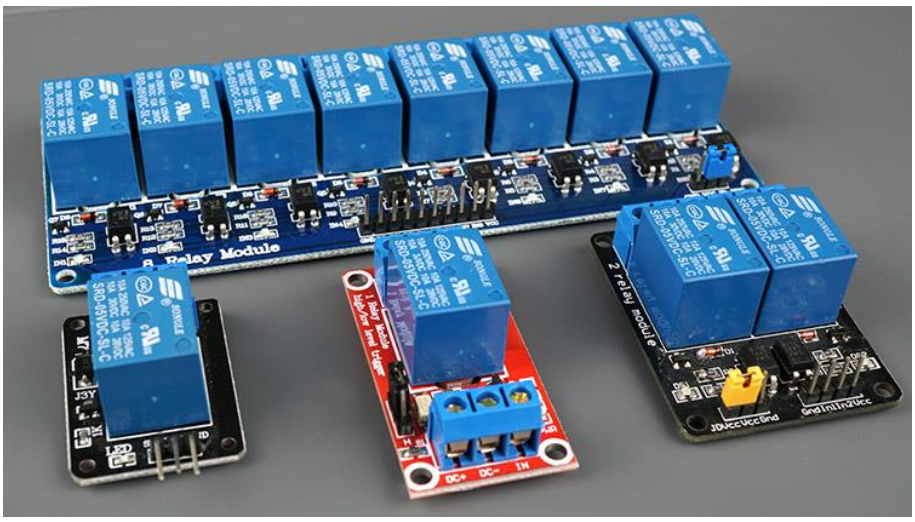

Рисунок 4.1 – Ілюстративний вигляд реле

Існують релейні модулі, електромагніт яких може живитися від 5 В і від 3,3 В. Обидва можна використовувати з ESP8266 – ви можете використовувати контакт Vin (який забезпечує 5 В) або контакт 3,3 В.

Окрім того, деякі оснащені вбудованою оптроною, яка додає додатковий «рівень» захисту, оптично ізолюючи ESP8266 від схеми реле.

Розпіновка реле. Для демонстрації можна подивитися на цоколівку 2-канального релейного модуля. Використання релейного модуля з іншою кількістю каналів аналогічно.

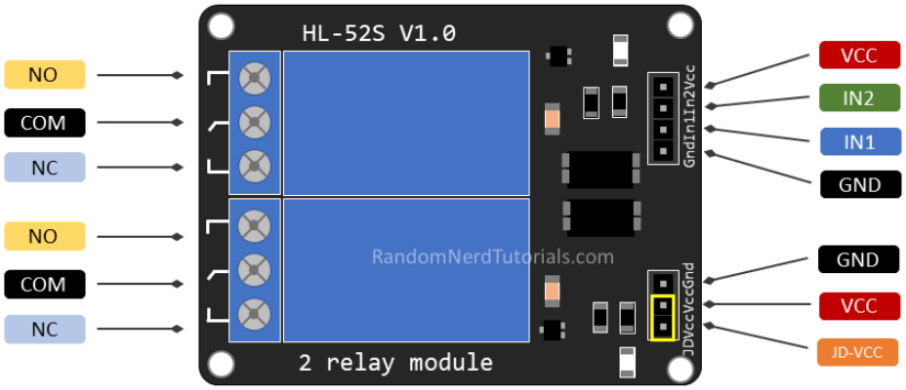

Рисунок 4.2 – Розпіновка реле

Два роз’єми (з трьома гніздами кожен) на лівій стороні релейного модуля підключають високу напругу, а контакти на правій стороні (низька напруга) підключаються до модулів уводу/виводу ESP8266 GPIO.

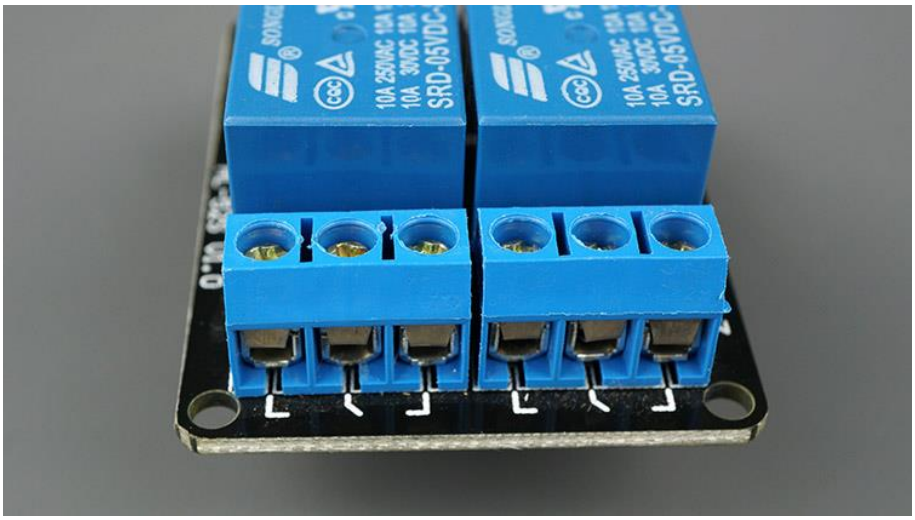

Рисунок 4.3 – Подвійний реле-модуль

Підключення до мережі. Релейний модуль, показаний на попередній фотографії, має два роз’єми, кожен з яких має три гнізда: загальний (COM), нормально закритий (NC), і нормально відкритий (HI).

– **COM:** підключіть струм, який ви хочете контролювати (напруга мережі).

– **NC (Normally Closed):** нормально замкнута конфігурація використовується, якщо потрібно, щоб реле було закрито за замовчуванням. NC є штифтами COM підключені, тобто струм тече, якщо ви не надішлете сигнал від ESP8266 до модуля реле, щоб розімкнути ланцюг і припинити потік струму.

– **NO (нормально відкритий):** нормально розімкнута конфігурація працює навпаки: немає зв’язку між контактами NO та COM, тому ланцюг розірвано, якщо ви не надішлете сигнал від ESP8266 для замикання ланцюга.

Контрольні штифти. Сторона низької напруги має набір із чотирьох і трьох контактів. Перший набір складається з VCC і GND, щоб включити живлення модуля і введіть 1 (IN1) і вхід 2 (IN2) для керування нижнім і верхнім реле відповідно.

Якщо ваш релейний модуль має лише один канал, ви матимете лише один контакт IN. Якщо у вас чотири канали, у вас буде чотири контакти IN, і так далі.

Сигнал, який ви посилаєте на контакти IN, визначає, активне реле чи ні. Реле спрацьовує, коли вхідна напруга падає нижче приблизно 2 В. Це означає, що ви матимете такі сценарії.

- **Нормально закрита конфігурація (NC)**:
- Сигнал HIGH – тече струм.
- Сигнал LOW – струм не тече.
- **Нормально відкрита конфігурація (NO):**
- Сигнал HIGH – струм не тече.
- Сигнал LOW – тече струм.

Ви повинні використовувати нормально закриту конфігурацію, коли струм має протікати більшу частину часу і ви хочете лише час від часу його зупиняти.

Використовуйте нормально відкриту конфігурацію, якщо ви хочете, щоб струм час від часу проходив (наприклад, час від часу вмикайте лампу).

Вибір джерела живлення. Другий набір шпильок складається з GND, VCC, і JD-VCC шпильки. JD-VCC штифт живить електромагніт реле. Зверніть увагу, що модуль має перемичку, яка з'єднує контакти VCC і JD-VCC.

З капюшоном VCC і JD-VCC контакти підключені. Це означає, що електромагніт реле живиться безпосередньо від контакту живлення ESP8266, тому модуль реле та схеми ESP8266 фізично не ізольовані один від одного.

Без перемички вам потрібно забезпечити незалежне джерело живлення для живлення електромагніту реле через JD-VCC шпильки. Ця конфігурація фізично ізолює реле від ESP8266 за допомогою вбудованої в модуль оптронної пари, що запобігає пошкодженню ESP8266 у разі стрибків напруги.

ESP8266. Найбезпечніші контакти для використання з реле. Деякі контакти ESP8266 видають сигнал 3,3 В під час завантаження ESP8266. Це може бути проблематично, якщо у вас є реле або інші периферійні пристрої, підключені до цих GPIO.

Окрім того, для завантаження ESP8266 деякі контакти мають бути висунуті HIGH або LOW.

Ураховуючи це, найбезпечніші контакти ESP8266 для використання з реле: GPIO 5, GPIO 4, GPIO 14, GPIO 12 і GPIO 13.

Для отримання додаткової інформації про ESP8266 GPIO читайте: ESP8266 Pinout Reference: Які контакти GPIO слід використовувати?

Підключення релейного модуля до плати ESP8266 NodeMCU. Підключіть релейний модуль до ESP8266, як показано на схемі. На схемі показано підключення 2-канального релейного модуля, підключення іншої кількості каналів аналогічне.

Попередження: у цьому прикладі ми маємо справу з напругою в мережі. Неправильне використання може призвести до серйозних травм. Якщо ви не знайомі з напругою в мережі, попросіть когось, хто вам допоможе. Під час програмування ESP або проводки вашої схеми переконайтеся, що все відключено від напруги мережі.

Окрім того, ви можете використовувати джерело живлення 12 В для керування приладами 12 В.

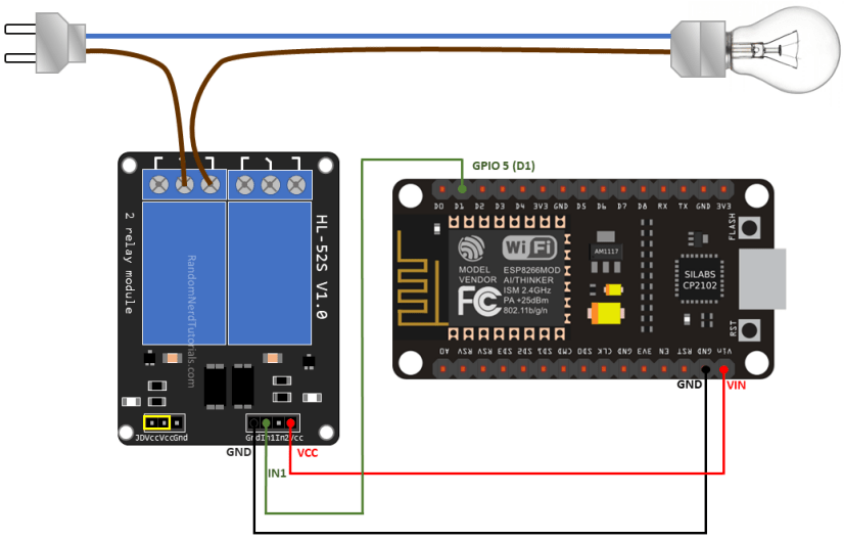

Рисунок 4.4 – Схема розмикання живлення лампи

У цьому прикладі ми керуємо лампою. Ми просто хочемо час від часу запалювати лампу, тому краще використовувати нормально відкриту конфігурацію.

Підключаємо контакт IN1 до GPIO 5, ви можете використовувати будь-який інший відповідний GPIO.

Керування релейним модулем за допомогою ESP8266 NodeMCU – Arduino Sketch. Код для керування реле за допомогою ESP8266 такий же простий, як керування світлодіодом або будь-яким іншим виходом. У цьому прикладі, оскільки ми використовуємо нормально відкриту конфігурацію, нам потрібно надіслати сигнал LOW, щоб протікав струм, і сигнал HIGH, щоб зупинити протікання струму.

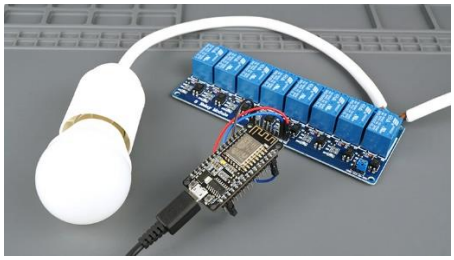

Рисунок 4.5 – Увімкнена реле лампа

Такий код засвітить лампу на 10 секунд і вимкне її ще на 10 секунд.

```
/*********
 Rui Santos
 Complete project details at https://RandomNerdTutorials.com/esp8266-relay-module-ac-web-server/
 The above copyright notice and this permission notice shall be included in all
 copies or substantial portions of the Software.
*********/
const int relay = 5;
void setup() {
 Serial.begin(115200);
 pinMode(relay, OUTPUT);
}
void loop() {
 // Normally Open configuration, send LOW signal to let current flow
 // (if you’re usong Normally Closed configuration send HIGH signal)
 digitalWrite(relay, LOW);
 Serial.println(«Current Flowing»);
 delay(5000);
 // Normally Open configuration, send HIGH signal stop current flow
 // (if you’re usong Normally Closed configuration send LOW signal)
 digitalWrite(relay, HIGH);
 Serial.println(«Current not Flowing»);
 delay(5000);
}
```

Керування кількома ретрансляторами за допомогою веб-сервера ESP8266 NodeMCU.

Рисунок 4.6 – Вигляд Інтернет-інтерфейсу керування

У цьому розділі ми створено приклад веб-сервера, який дозволяє керувати будь-якою кількістю реле через веб-сервер, незалежно від того, налаштовані вони як нормально відкриті чи як нормально закриті. Просто потрібно змінити кілька рядків коду, щоб визначити кількість реле, якими ви хочете керувати, і призначення контактів.

Для створення цього веб-сервера ми використовуємо бібліотеку ESPAsyncWebServer.

Установлення бібліотеки ESPAsyncWebServer. Виконайте наступні кроки, щоб установити бібліотеку ESPAsyncWebServer:

1. Завантажте бібліотеку ESPAsyncWebServer. У папці завантажень має бути папка.zip.

2. Розархівуйте папку.zip, і ви повинні отримати папку *ESPAsyncWebServer-master*.

3. Перейменуйте папку у *ESPAsyncWebServer*.

4. Перемістіть папку *ESPAsyncWebServer* до папки інсталяційних бібліотек Arduino IDE.

Окрім того, у вашій Arduino IDE ви можете перейти до **Sketch** > **Include Library** > **Add.ZIP library…** і вибрати бібліотеку, яку ви щойно завантажили.

Установлення бібліотеки ESPAsyncTCP для ESP8266. Для роботи бібліотеки ESPAsyncWebServer потрібна бібліотека ESPAsyncTCP. Виконайте наступні дії, щоб установити цю бібліотеку:

1. Завантажте бібліотеку ESPAsyncTCP. У папці завантажень має бути папка.zip.

2. Розархівуйте папку.zip, і ви повинні отримати папку *ESPAsyncTCP-master.*

3. Перейменуйте у *ESPAsyncTCP*.

4. Перемістіть папку *ESPAsyncTCP* у папку інсталяційних бібліотек Arduino IDE.

5. Нарешті, знову відкрийте Arduino IDE.

Окрім того, у вашій Arduino IDE ви можете перейти до **Sketch** > **Include Library** > **Add.ZIP library…** і вибрати бібліотеку, яку ви щойно завантажили.

Після встановлення необхідних бібліотек скопіюйте наведений далі код у вашу Arduino IDE.

```
/*********
 Rui Santos
 Complete project details at https://RandomNerdTutorials.com/esp8266-relay-module-ac-web-server/
 The above copyright notice and this permission notice shall be included in all
 copies or substantial portions of the Software.
*********/
// Import required libraries
#include «ESP8266WiFi.h»
#include «ESPAsyncWebServer.h»
// Set to true to define Relay as Normally Open (NO)
#define RELAY_NO  true
// Set number of relays
#define NUM_RELAYS 5
// Assign each GPIO to a relay
int relayGPIOs[NUM_RELAYS] = {5, 4, 14, 12, 13};
// Replace with your network credentials
const char* ssid = «REPLACE_WITH_YOUR_SSID»;
const char* password = «REPLACE_WITH_YOUR_PASSWORD»;
const char* PARAM_INPUT_1 = «relay»;
const char* PARAM_INPUT_2 = «state»;
// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
const char index_html[] PROGMEM = R»rawliteral(
<!DOCTYPE HTML><html>
<head>
 <meta name=«viewport» content=«width=device-width, initial-scale=1»>
```

```
<style>
 html {font-family: Arial; display: inline-block; text-align: center;}
 h2 {font-size: 3.0rem;}
 p {font-size: 3.0rem;}
 body {max-width: 600px; margin:0px auto; padding-bottom: 25px;}
.switch {position: relative; display: inline-block; width: 120px; height: 68px}
.switch input {display: none}
.slider {position: absolute; top: 0; left: 0; right: 0; bottom: 0; background-color: #ccc; border-radius: 34px}
.slider:before {position: absolute; content: ««; height: 52px; width: 52px; left: 8px; bottom: 8px; background-color: #fff; -webkit-transition:.4s; transition:.4s; border-radius: 68px}
 input:checked+.slider {background-color: #2196F3}
 input:checked+.slider:before {-webkit-transform: translateX(52px); -ms-transform: translateX(52px); transform: translateX(52px)}
</style>
</head>
<body>
 <h2>ESP Web Server</h2>
 %BUTTONPLACEHOLDER%
<script>function toggleCheckbox(element) {
 var xhr = new XMLHttpRequest();
 if(element.checked){ xhr.open(«GET», «/update?relay=«+element.id+»&state=1», true); }
 else { xhr.open(«GET», «/update?relay=«+element.id+»&state=0», true); }
 xhr.send();
}</script>
</body>
</html>
)rawliteral»;
// Replaces placeholder with button section in your web page
String processor(const String& var){
 //Serial.println(var);
 if(var == «BUTTONPLACEHOLDER»){
  String buttons =««;
  for(int i=1; i<=NUM_RELAYS; i++){
   String relayStateValue = relayState(i);
   buttons+= «<h4>Relay #» + String(i) + « – GPIO « + relayGPIOs[i-1] + «</h4><label class=\»switch\»><input type=\»checkbox\» onchange=\»toggleCheckbox(this)\» id=\»« + String(i) + «\» «+ relayStateValue +»><span class=\»slider\»></span></label>«;
  }
  return buttons;
 }
 return String();
}
String relayState(int numRelay){
 if(RELAY_NO){
  if(digitalRead(relayGPIOs[numRelay-1])){
   return ««;
  }
  else {
   return «checked»;
  }
 }
 else {
  if(digitalRead(relayGPIOs[numRelay-1])){
   return «checked»;
  }
  else {
   return ««;
  }
 }
 return ««;
}
```

```
void setup(){
 // Serial port for debugging purposes
 Serial.begin(115200);
 // Set all relays to off when the program starts – if set to Normally Open (NO), the relay is off when you set the relay to HIGH
 for(int i=1; i<=NUM_RELAYS; i++){
  pinMode(relayGPIOs[i-1], OUTPUT);
  if(RELAY_NO){
   digitalWrite(relayGPIOs[i-1], HIGH);
  }
  else{
   digitalWrite(relayGPIOs[i-1], LOW);
  }
 }
 // Connect to Wi-Fi
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println(«Connecting to WiFi..»);
 }
 // Print ESP8266 Local IP Address
 Serial.println(WiFi.localIP());
 // Route for root/web page
 server.on(«/», HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, «text/html», index_html, processor);
 });
 // Send a GET request to <ESP_IP>/update?relay=<inputMessage>&state=<inputMessage2>
 server.on(«/update», HTTP_GET, [] (AsyncWebServerRequest *request) {
  String inputMessage;
  String inputParam;
  String inputMessage2;
  String inputParam2;
  // GET input1 value on <ESP_IP>/update?relay=<inputMessage>
  if (request->hasParam(PARAM_INPUT_1) & request->hasParam(PARAM_INPUT_2)) {
   inputMessage = request->getParam(PARAM_INPUT_1)->value();
   inputParam = PARAM_INPUT_1;
   inputMessage2 = request->getParam(PARAM_INPUT_2)->value();
   inputParam2 = PARAM_INPUT_2;
   if(RELAY_NO){
    Serial.print(«NO «);
    digitalWrite(relayGPIOs[inputMessage.toInt()-1], !inputMessage2.toInt());
   }
   else{
    Serial.print(«NC «);
    digitalWrite(relayGPIOs[inputMessage.toInt()-1], inputMessage2.toInt());
   }
  }
  else {
   inputMessage = «No message sent»;
   inputParam = «none»;
  }
  Serial.println(inputMessage + inputMessage2);
  request->send(200, «text/plain», «OK»);
 });
 // Start server
 server.begin();
}

void loop() {
}
```

Визначте конфігурацію реле. Змініть наведену далі змінну, щоб указати, чи використовуєте ви свої реле в нормально розімкненій (NO) або нормально закритій (NC) конфігурації. Установіть значення змінної RELAY_NO у стан true для нормально розімкнених контактів.

#define RELAY_NO true.

Визначити кількість реле (каналів). Ви можете визначити кількість реле, якими ви хочете керувати NUM_ RELAYS змінна. Для демонстрації ми встановлюємо значення 5.

#define NUM_RELAYS 5.

Визначте призначення контактів реле. У наведеній далі змінній масиву ви можете визначити GPIO ESP8266, які керуватимуть реле.

int relayGPIOs[NUM_RELAYS] = {5, 4, 14, 12, 13}; кількість реле, встановлених у NUM_ RELAYS змінній, мають відповідати кількості GPIO, призначених у relay-GPIO масиві.

Облікові дані мережі. Вставте свої мережні облікові дані в наведені далі змінні.

const char* ssid = «REPLACE_WITH_YOUR_SSID»;

const char* password = «REPLACE_WITH_YOUR_PASSWORD».

Підключення 8-канального реле до ESP8266 NodeMCU. Для демонстрації ми керуємо 5 релейними каналами. Підключіть плату ESP8266 NodeMCU до релейного модуля, як показано на рисунку 4.7.

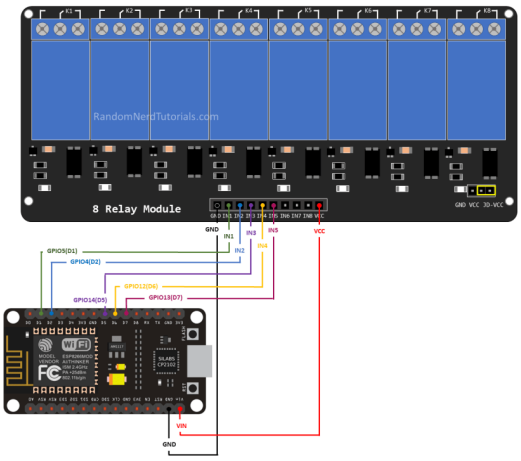

Рисунок 4.7 – Схема підключення реле

Демонстрація реалізованого рішення. Після внесення необхідних змін завантажте код на свій ESP8266.

Відкрийте Serial Monitor на швидкості 115200 бод і натисніть кнопку ESP8266 RST, щоб отримати його IP-адресу.

Потім відкрийте браузер у локальній мережі та введіть IP-адресу ESP8266, щоб отримати доступ до веб-сервера.

Ви повинні отримати щось таке з такою кількістю кнопок, скільки реле ви визначили у своєму коді.

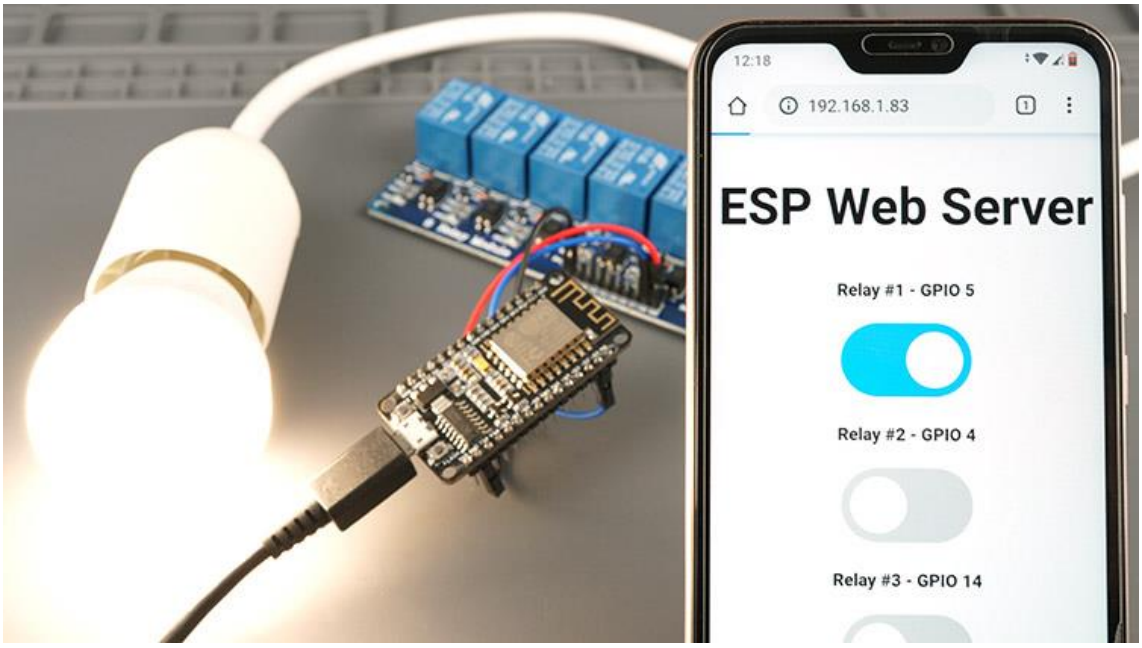

Рисунок 4.8 – Вигляд робочої IoT-системи

Тепер за допомогою кнопок можна дистанційно керувати реле за допомогою смартфона.

4.3 Завдання для самостійного виконання

Ознайомитися з порядком виконання роботи та, підключивши реалізувати програму для Node MCU, яка під час підключення до реле, що знаходиться на моделі Smart Home, виконуватиме задачі наведені в таблиці 4.1. Варіант обирати згідно з алфавітною позицією у списку групи.

Таблиця 4.1 – Варіанти індивідуальних завдань

Вар.	Вихідні дані	Вар.	Вихідні дані
1	Вимикатиме за натисканням світлодіод на 15 с	16	Вмикатиме за натисканням світлодіод на 15 с
2	Вимикатиме за натисканням світлодіод на 14 с	17	Вмикатиме за натисканням світлодіод на 14 с
3	Вимикатиме за натисканням світлодіод на 13 с	18	Вмикатиме за натисканням світлодіод на 13 с
4	Вимикатиме за натисканням світлодіод на 12 с	19	Вмикатиме за натисканням світлодіод на 12 с
5	Вимикатиме за натисканням світлодіод на 11 с	20	Вмикатиме за натисканням світлодіод на 11 с
6	Вимикатиме за натисканням світлодіод на 10 с	21	Вмикатиме за натисканням світлодіод на 10 с
7	Вимикатиме за натисканням світлодіод на 9 с	22	Вмикатиме за натисканням світлодіод на 9 с
8	Вимикатиме за натисканням світлодіод на 8 с	23	Вмикатиме за натисканням світлодіод на 8 с
9	Вимикатиме за натисканням світлодіод на 7 с	24	Вмикатиме за натисканням світлодіод на 7 с
10	Вимикатиме за натисканням світлодіод на 6 с	25	Вмикатиме за натисканням світлодіод на 6 с
11	Вимикатиме за натисканням світлодіод на 5 с	26	Вмикатиме за натисканням світлодіод на 5 с
12	Вимикатиме за натисканням світлодіод на 4 с	27	Вмикатиме за натисканням світлодіод на 4 с
13	Вимикатиме за натисканням світлодіод на 3 с	28	Вмикатиме за натисканням світлодіод на 3 с
14	Вимикатиме за натисканням світлодіод на 2 с	29	Вмикатиме за натисканням світлодіод на 2 с
15	Вимикатиме за натисканням світлодіод на 1 с	30	Вмикатиме за натисканням світлодіод на 1 с

4.4 Зміст звіту

1. Короткий опис змісту виконання роботи.
2. Скріншоти та аналітичні дані проведеної роботи.
3. Висновки.

4.5 Контрольні запитання

1. Що таке реле?
2. Що таке Node MCU?
3. Як об’єднати Node MCU та реле?
4. Яка схема підключення керованого пристрою до реле?
5. Якими бібліотеками реалізується сервер на Node MCU?

Лабораторна робота № 5

Тема. Побудова сенсорної мережі на основі Node MCU

Мета: вивчення особливостей реалізації сенсорних мереж на основі ESP-NOW.

5.1 Короткі теоретичні відомості

Технологія ESP-NOW – це спрощений протокол зв’язку WiFi з передачею коротких пакетів між парами парних пристроїв, розроблений і випущений *Espressif* у липні 2016 року для мікроконтролерів ESP8266 та ESP32. Додаткові процедури, пов’язані з підтримкою протоколу WiFi, не використовуються, що прискорює процес обміну пакетами.

ESP-NOW може застосовуватися в Інтернеті Речей для керування інтелектуальними джерелами світла, реле, розетками, іншими пристроями дистанційного керування, отримання інформації від датчиків та інших програм.

ESP-NOW підтримує такі функції.

− Зашифрований і незашифрований зв’язок між парами пар.

− Змішані зашифрований і незашифрований зв’язок між сполученими пристроями.

− Передача до 250 б корисної інформації.

− Налаштування функції зворотного дзвінка для інформування прикладного рівня, зокрема, про успішність або збій передачі.

ESP-NOW також має такі особливості й обмеження.

− Швидкість передачі – трохи більше 1 Мбіт/с з частотою 2,4 ГГц, тобто. ESP-NOW працює на тій же частоті та каналах, що і ваш роутер WiFi.

− Протокол Wi-Fi не використовується.

− Аналогічний протоколу з низьким енергоспоживанням, який використовується в бездротовій миші 2,4 ГГц.

− Потрібне лише початкове сполучення.

− Після сполучення з’єднання не розривається.

− Широкомовна передача не підтримується – тільки множинна роздача парам парних пристроїв.

− Максимум 20 пар, включаючи зашифровані, підтримуються на одному пристрої, включаючи зашифровані пари.

− Максимум 10 пар зашифрованих підтримуються в режимі Station.

− Максимум 6 в режимі SoftAP або SoftAP+Station.

− Шифрування багатоадресної розсилки не підтримується.

Безпека. ESP-NOW застосовує технологію кадрів IEEE802.11 Action Vendor разом з функцією IE, розробленою Espressif, та технологією шифрування CCMP, реалізуючи безпечне комунікаційне рішення без установлення з'єднання. Пристрій з Wi-Fi підтримує основний майстер-ключ (PMK) та кілька локальних майстер-ключів (LMK).

− PMK використовується для шифрування LMK за допомогою алгоритму AES-128.

− LMK парного пристрою використовується для шифрування інформації користувача методом CCMP. Максимальна кількість різних LMK – 6. Якщо LMK для парного пристрою не заданий, дані користувача шифруватися не будуть.

Базовий рівень. На нижньому рівні протоколу ESP_NOW підтримується пов'язаний список, що містить інформацію про локальний пристрій та про сполучений пристрій, у тому числі MAC-адреси та ключі. ESP-NOW також зберігає дані, що часто використовуються, для прикладного рівня, щоб уникнути накладних витрат на повторну обробку пов'язаного списку. Інформація про пристрої використовується для надсилання та отримання даних і передбачає інформацію про локальний пристрій.

− PMK : 16 байт – основний майстер-ключ, який використовується для шифрування даних на приєднаному пристрої (KOK в API). ESP_NOW підтримує PMK за замовчуванням, тому налаштування не потрібне. Якщо необхідно, переконайтеся, що PMK збігається з локальним пристроєм.

Режим: 1 байт – режим локального пристрою, що визначає передавальний WiFi інтерфейс (SoftAP або STA) ESP-NOW. Режим пари не впливає на будь-яку функцію, а лише зберігає інформацію про режим для прикладного рівня. У режимі STA WiFi застосовується лише Station і SoftAP WiFi – лише SoftAP.

Інформацію про пару в парі (включаючи інформацію, що часто використовується, та іншу користувальницьку інформацію):

− **LMK:** 16 байт – локальний майстер-ключ, який використовується для шифрування ключа корисної інформації під час зв'язку в цій парі.

− **MAC-адреса:** 6 байт – адреса сполученого пристрою, збігається з адресою відправника. Наприклад, якщо пакет відправляється зі Station, MAC-адреса має збігатися з адресою Station.

− **Режим:** 1 байт – режим локального пристрою, що визначає передавальний інтерфейс (SoftAP або STA) ESP-NOW.

− **Канал:** 1 байт – канал, через який обмінюються даними пристрою, з'єднані в пару. Може мати значення 0..255. Канал не впливає на жодну функцію, а лише зберігає інформацію про канал для прикладного рівня. Значення визначається прикладним рівнем. Наприклад, 0 означає, що канал не визначено; 1~14 означає дійсні канали; всім іншим значенням можуть бути призначені функції, визначені прикладним рівнем.

Espressif не рекомендує використовувати тривалі операції у функціях зворотного виклику під час надсилання/відправки пакетів, що пов'язано, ймовірно, з реалізацією алгоритмів, які використовують механізм переривань. На користь цього припущення також говорять і проблеми, пов'язані з динамічним виділенням пам'яті у функціях зворотного виклику, що розв'язкуються кращим використанням статичних змінних, а також неоднозначність застосування механізму виключень *MicroPython*.

Реалізація асинхронності процесів запуску/завершення, встановлення зв'язку в парі, отримання/передачі пакетів *Espressif* не описується, що не полегшує застосування ідеології ***asyncio*** *MicroPython*.

Формат пакету ESP-NOW.

− **Заголовок MAC**: 24 байти.

− **Категорія**: 1 байт, що вказує на категорію творця пакета. Встановлено значення (127).

− **ID організації**: 3 байти, містить унікальний ідентифікатор, який є першими трьома байтами MAC-адреси, застосованої Espressif. Установлено значення (0x18fe34).

− **Випадкове значення**: 4 байти, використовується для захисту даних.

− **Дані творця пакету**: 7-255байт.

Дані автора пакета містять такі поля.

− **ID**: 1 байт, установлено значення (221).

− **Довжина**: 1 байт, загальна довжина ID організації, типу, версії та даних користувача.

− **ID організації**: 3 байти, містить унікальний ідентифікатор, який є першими трьома байтами MAC-адреси, застосованої Espressif. Установлено значення (0x18fe34).

− **Тип**: 1 байт, протокол ESP-NOW. Установлено значення (4).

− **Версія**: 1 байт, поточна версія ESP-NOW. Установлено (1).

− **Вміст**: 0–250 байт, дані користувача.

− **FCS**: 4 байти, контрольна сума.

Оскільки ESP-NOW не використовує протокол WiFi, заголовок MAC трохи відрізняється від заголовка стандартних пакетів. Біти FromDS та ToDS поля FrameControl дорівнюють 0. У першому полі адреси задана адреса призначення. У другому полі адреси вказано адресу джерела. Третє поле адреси встановлено як широкомовна адреса (0xff: 0xff: 0xff: 0xff: 0xff: 0xff).

Подана інформація та технічна реалізація детально надана її розробником Rui Santos.

ESP-NOW підтримує такі функції.

− Зашифрований і незашифрований одноадресний зв’язок.

– Змішані зашифровані й незашифровані однорангові пристрої.

– Можна переносити корисне навантаження **розміром до 250 байт.**

– Надсилання функції зворотного виклику, яку можна налаштувати для інформування прикладного рівня про успішну або невдалу передачу.

Технологія ESP-NOW також має такі обмеження.

– Обмежені зашифровані вузли. У станційному режимі підтримуються щонайбільше 10 зашифрованих вузлів; не більше 6 у режимі SoftAP або SoftAP+Station;

– Підтримуються кілька незашифрованих однорангових вузлів, однак їх загальна кількість має бути менше 20, включаючи зашифровані однорангові вузли.

– **Корисне навантаження обмежене 250 байтами**.

Простими словами, ESP-NOW – це швидкий протокол зв’язку, який можна використовувати для обміну невеликими повідомленнями (до 250 байт) між платами ESP8266.

ESP-NOW дуже універсальний, і ви можете мати односторонній або двосторонній зв’язок у різних налаштуваннях.

Рисунок 5.1 – Однонаправлений зв’язок

Односторонній зв’язок ESP-NOW. Одна плата ESP8266 надсилає дані на іншу плату ESP8266. Цю конфігурацію дуже легко реалізувати, і це чудово

для надсилання даних з однієї плати на іншу, як-от показання датчиків або команди ON і OFF для керування GPIO.

«Головний» ESP8266 надсилає дані кільком «підлеглим» ESP8266. Одна плата ESP8266 надсилає однакові або різні команди на різні плати ESP8266. Ця конфігурація ідеальна для створення чогось на зразок пульта дистанційного керування. Ви можете мати кілька плат ESP8266 навколо будинку, які керуються однією головною платою ESP8266.

Рисунок 5.2 – Один хаб та багато керованих пристроїв

Один «підлеглий» ESP8266 отримує дані від кількох «головних». Ця конфігурація ідеальна, якщо ви хочете збирати дані з кількох вузлів датчиків на одній платі ESP8266. Його можна налаштувати як веб-сервер для відображення даних з усіх інших плат, наприклад.

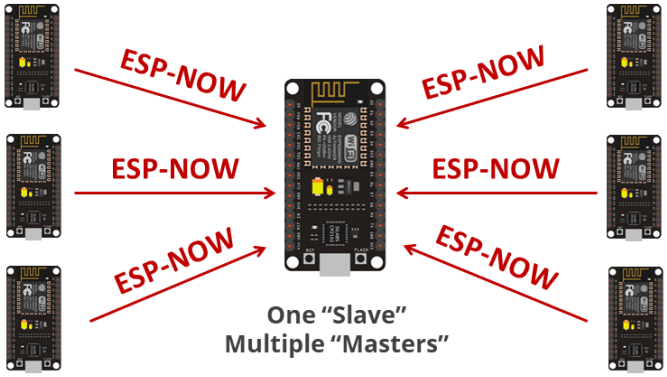

Рисунок 5.3 – Багато керівників, але один підпорядкований пристрій

Примітка: у документації ESP-NOW немає такого поняття, як «відправник/головний» і «одержувач/підлеглий». Кожна дошка може бути відправником або отримувачем. Однак, щоб все було зрозуміло, ми будемо використовувати терміни «відправник» і «одержувач» або «головний» і «підлеглий».

Двосторонній зв'язок ESP-NOW. З ESP-NOW кожна дошка може бути відправником і отримувачем одночасно. Отже, можна встановити двосторонній зв'язок між платами.

Наприклад, ви можете мати дві дошки, які спілкуються одна з одною.

Рисунок 5.4 – Двостороння комунікація

Ви можете додати більше плат до цієї конфігурації та мати щось схоже на мережу (усі плати ESP8266 спілкуються одна з одною).

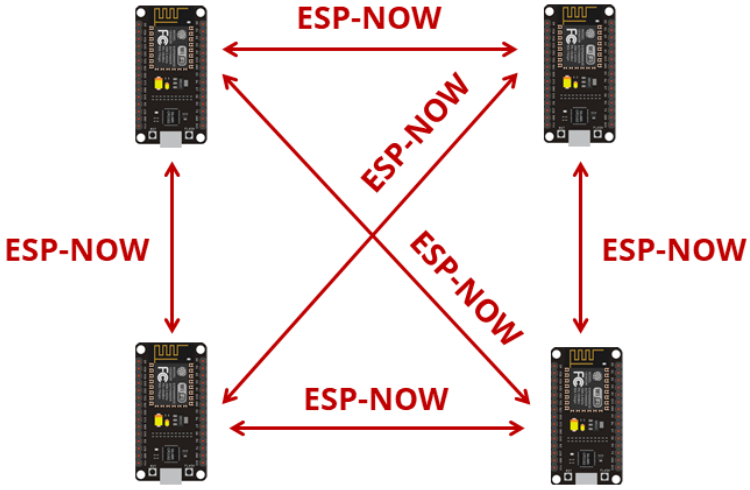

Рисунок 5.5 – Мережна комунікація

Отже, ESP-NOW ідеально підходить для побудови мережі, в якій ви можете мати кілька плат ESP8266, які обмінюються даними одна з одною.

5.2 Порядок виконання роботи

ESP8266: отримання MAC-адреси плати. Щоб спілкуватися через ESP-NOW, вам потрібно знати MAC-адресу приймача ESP8266. Так ви дізнаєтесь, на який пристрій надішлете інформацію.

Кожен ESP8266 має унікальну MAC-адресу, і саме так ми ідентифікуємо кожну плату для надсилання даних на неї за допомогою ESP-NOW (дізнайтеся, як отримати та змінити MAC-адресу ESP8266).

Щоб отримати MAC-адресу дошки, завантажте такий код.

```
// Complete Instructions to Get and Change ESP MAC Address: https://RandomNerdTutorials.com/get-change-esp32-esp8266-mac-address-arduino/
#include <ESP8266WiFi.h>
void setup(){
 Serial.begin(115200);
 Serial.println();
 Serial.print(«ESP8266 Board MAC Address: «);
 Serial.println(WiFi.macAddress());
}
 void loop(){
}
```

Після завантаження коду відкрийте Serial Monitor на швидкості 115200 бод і натисніть кнопку ESP8266 RESET. MAC-адреса має бути надрукована так.

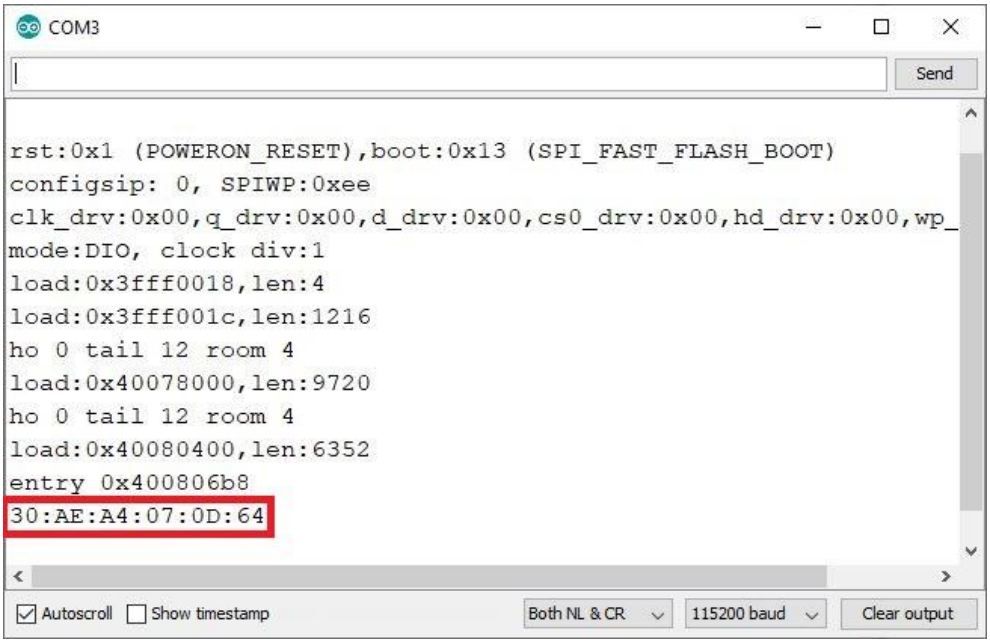

Рисунок 5.6 – Інтерфейс COM-порту

Збережіть MAC-адресу дошки, оскільки вона знадобиться для надсилання даних на потрібну дошку через ESP-NOW.

ESP-NOW Односторонній зв'язок «точка-точка» з ESP8266. Щоб почати роботу з бездротовим зв'язком ESP-NOW, ми створимо простий проєкт, який покаже, як надіслати повідомлення з одного ESP8266 на інший. Один ESP8266 буде «відправником», а інший ESP8266 буде «одержувачем».

Ми надішлемо структуру, яка містить змінну типу *char, int, float, String* і *boolean*. Потім ви можете змінити структуру, щоб надсилати будь-які типи змінних, які підходять для вашого проєкту (наприклад, показання датчиків або логічні змінні, щоб щось увімкнути чи вимкнути).

Для кращого розуміння ми будемо називати «відправником» ESP8266 #1 і «одержувачем» ESP8266 #2.

Ось що ми повинні включити в ескіз **відправника**.

1. Ініціалізувати ESP-NOW.

2. Зареєструйте функцію зворотного виклику після надсилання даних – OnDataSent функція буде виконана під час надсилання повідомлення. Це може сказати нам, чи було повідомлення успішно доставлено.

3. Додати одноранговий пристрій (приймач). Для цього вам потрібно знати MAC-адресу приймача.

4. Надіслати повідомлення на одноранговий пристрій.

З боку **ствольної** коробки ескіз має містити таке.

1. Ініціалізувати ESP-NOW.

2. Зареєструватися для отримання функції зворотного виклику (OnDataRecv). Це функція, яка буде виконана під час отримання повідомлення.

3. У цій функції зворотного виклику зберегти повідомлення у змінній, щоб виконати будь-яке завдання з цією інформацією.

ESP-NOW працює з функціями зворотного виклику, які викликаються, коли пристрій отримує повідомлення або коли повідомлення надсилається (ви отримуєте інформацію про те, чи було доставлено повідомлення успішно).

Корисні функції ESP-NOW. Ось короткий перелік найважливіших функцій ESP-NOW.

Назва та опис функці.

esp_now_init()Ініціалізує ESP-NOW. Ви повинні ініціалізувати Wi-Fi перед ініціалізацією ESP-NOW. Повертає 0 у разі успіху.

esp_now_set_self_role(роль)роль може бути:ESP_NOW_ROLE_IDLE = 0, ESP_NOW_ROLE_CONTROLLER,ESP_NOW_ROLE_SLAVE,ESP_NOW_ROLE_COMBO,ESP_NOW_ROLE_MAX.

esp_now_add_peer(uint8 mac_addr, uint8 роль, uint8 канал, uint8 ключ, uint8 key_len). Викличте цю функцію, щоб створити пару з пристроєм.

esp_now_send(uint8 mac_address, uint8 дані, int len). Надсилайте дані за допомогою ESP-NOW.

esp_now_register_send_cb(). Зареєструйте функцію зворотного виклику, яка запускається після надсилання даних. Коли повідомлення надсилається, викликається функція – ця функція інформує, чи була доставка успішною.

esp_now_register_rcv_cb(). Зареєструйте функцію зворотного виклику, яка запускається після отримання даних. Коли дані надходять через ESP-NOW, викликається функція.

Для отримання додаткової інформації про ці функції:

Ескіз відправника NodeMCU ESP8266 (ESP-NOW). Ось код для плати відправника ESP8266 NodeMCU. Скопіюйте код до Arduino IDE, але поки не завантажуйте його. Вам потрібно внести кілька змін, щоб він працював на вас.

```
/*
 Rui Santos
 Complete project details at https://RandomNerdTutorials.com/esp-now-esp8266-nodemcu-arduino-ide/
  Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files.
 The above copyright notice and this permission notice shall be included in all
 copies or substantial portions of the Software.
*/
#include <ESP8266WiFi.h>
#include <espnow.h>
// REPLACE WITH RECEIVER MAC Address
uint8_t broadcastAddress[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
// Structure example to send data
// Must match the receiver structure
typedef struct struct_message {
```

```
 char a[32];
 int b;
 float c;
 String d;
 bool e;
} struct_message;
// Create a struct_message called myData
struct_message myData;
unsigned long lastTime = 0;
unsigned long timerDelay = 2000; // send readings timer
// Callback when data is sent
void OnDataSent(uint8_t *mac_addr, uint8_t sendStatus) {
 Serial.print(«Last Packet Send Status: «);
 if (sendStatus == 0){
  Serial.println(«Delivery success»);
 }
 else{
  Serial.println(«Delivery fail»);
 }
}

void setup() {
 // Init Serial Monitor
 Serial.begin(115200);
 // Set device as a Wi-Fi Station
 WiFi.mode(WIFI_STA);
 // Init ESP-NOW
 if (esp_now_init() != 0) {
  Serial.println(«Error initializing ESP-NOW»);
  return;
 }
 // Once ESPNow is successfully Init, we will register for Send CB to
 // get the status of Trasnmitted packet
 esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);
 esp_now_register_send_cb(OnDataSent);
 // Register peer
 esp_now_add_peer(broadcastAddress, ESP_NOW_ROLE_SLAVE, 1, NULL, 0);
}

void loop() {
 if ((millis() – lastTime) > timerDelay) {
  // Set values to send
  strcpy(myData.a, «THIS IS A CHAR»);
  myData.b = random(1,20);
  myData.c = 1.2;
  myData.d = «Hello»;
  myData.e = false;
  // Send message via ESP-NOW
  esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
  lastTime = millis();
 }
}
```

Як працює код.

По-перше, включіть ESP8266WiFi.h і espnow.h бібліотеки.

#include <ESP8266WiFi.h>

#include <espnow.h>

У наступному рядку ви повинні вставити MAC-адресу приймача ESP8266.

uint8_t broadcastAddress[] = {0x5C, 0xCF, 0x7F, 0x99, 0x9A, 0xEA}.

У нашому випадку MAC-адреса одержувача: 5C:CF:7F:99:9A:EA, але вам потрібно замінити цю змінну своєю власною MAC-адресою.

Потім створіть структуру, яка містить тип даних, які ми хочемо надіслати. Ми назвали цю структуру struct_message і, вона містить 5 різних типів змінних. Ви можете змінити це, щоб надсилати будь-які типи змінних.

```
typedef struct struct_message {
 char a[32];
 int b;
 float c;
 String d;
 bool e;
} struct_message;
```

Створіть нову змінну типу struct_message, що називається myData який зберігатиме значення змінних.

struct_message myData.

Далі визначте OnDataSent() функцію. Це функція зворотного виклику, яка буде виконана під час надсилання повідомлення. У цьому разі це повідомлення просто друкується, якщо повідомлення було успішно надіслано.

```
void OnDataSent(uint8_t *mac_addr, uint8_t sendStatus) {
 Serial.print(«Last Packet Send Status: «);
 if (sendStatus == 0){
  Serial.println(«Delivery success»);
 }
 else{
  Serial.println(«Delivery fail»);
 }
}
```

В setup(), ініціалізуйте послідовний монітор для цілей налагодження.

Serial.begin(115200).

Налаштуйте пристрій як станцію Wi-Fi.

WiFi.mode(WIFI_STA);

Ініціалізація ESP-NOW.

```
if (esp_now_init() != 0) {
 Serial.println(«Error initializing ESP-NOW»);
 return;
}
```

Установіть роль ESP8266:

esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);

Він приймає такі ролі: ESP_NOW_ROLE_CONTROLLER, ESP_NOW_ROLE_SLAVE, ESP_NOW_ROLE_COMBO, ESP_NOW_ROLE_MAX.

Після успішної ініціалізації ESP-NOW зареєструйте функцію зворотного виклику, яка буде викликана під час надсилання повідомлення. У цьому випадку ми реєструємося на OnDataSent() функції, створеної раніше.

esp_now_register_send_cb(OnDataSent).

Потім підключіться до іншого пристрою ESP-NOW, щоб надіслати дані:

esp_now_add_peer(broadcastAddress, ESP_NOW_ROLE_SLAVE, 1, NULL, 0).

Функція esp_now_add_peer приймає такі аргументи в такому порядку: mac-адреса, роль, канал Wi-Fi, ключ і довжина ключа.

у Loop(), ми надсилатимемо повідомлення через ESP-NOW кожні 2 секунди (ви можете змінити час затримки за допомогою зміни timerDelay).

Спочатку ми встановлюємо значення змінних наступним чином:

```
strcpy(myData.a, «THIS IS A CHAR»);
myData.b = random(1,20);
myData.c = 1.2;
myData.d = «Hello»;
myData.e = false.
```

Пам’ятайте, що myData – це структура. Тут ми призначаємо значення, які хочемо надіслати в середину структури. Наприклад, перший рядок призначає char, другий рядок призначає випадкове число Int, Float, String і булеву змінну.

Ми створюємо таку структуру, щоб показати вам, як надсилати найпоширеніші типи змінних. Ви можете змінити структуру, щоб надсилати будь-який інший тип даних.

Нарешті, надішліть повідомлення так:

esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));

Цикл виконується кожні 2000 мілісекунд (2 секунди).

if ((millis() – lastTime) > timerDelay) {

```
// Set values to send
strcpy(myData.a, «THIS IS A CHAR»);
myData.b = random(1,20);
myData.c = 1.2;
myData.d = «Hello»;
myData.e = false;
// Send message via ESP-NOW
esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
lastTime = millis();
}
```

Ескіз приймача ESP8266 NodeMCU (ESP-NOW). Завантажте наступний код на плату приймача ESP8266 NodeMCU.

```
/*
 Rui Santos
 Complete project details at https://RandomNerdTutorials.com/esp-now-esp8266-nodemcu-arduino-ide/
 Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files.
 The above copyright notice and this permission notice shall be included in all
 copies or substantial portions of the Software.
*/
#include <ESP8266WiFi.h>
#include <espnow.h>
// Structure example to receive data
// Must match the sender structure
typedef struct struct_message {
  char a[32];
  int b;
  float c;
  String d;
  bool e;
} struct_message;
// Create a struct_message called myData
struct_message myData;
// Callback function that will be executed when data is received
void OnDataRecv(uint8_t * mac, uint8_t *incomingData, uint8_t len) {
 memcpy(&myData, incomingData, sizeof(myData));
 Serial.print(«Bytes received: «);
 Serial.println(len);
 Serial.print(«Char: «);
 Serial.println(myData.a);
 Serial.print(«Int: «);
 Serial.println(myData.b);
 Serial.print(«Float: «);
 Serial.println(myData.c);
 Serial.print(«String: «);
 Serial.println(myData.d);
 Serial.print(«Bool: «);
 Serial.println(myData.e);
```

```
 Serial.println();
}
void setup() {
 // Initialize Serial Monitor
 Serial.begin(115200);
 // Set device as a Wi-Fi Station
 WiFi.mode(WIFI_STA);
 // Init ESP-NOW
 if (esp_now_init() != 0) {
  Serial.println(«Error initializing ESP-NOW»);
  return;
 }
 // Once ESPNow is successfully Init, we will register for recv CB to
 // get recv packer info
 esp_now_set_self_role(ESP_NOW_ROLE_SLAVE);
 esp_now_register_recv_cb(OnDataRecv);
}
void loop() {
}
```

Як працює код. Подібно до відправника, почніть із додавання бібліотек:

```
#include <esp_now.h>
#include <WiFi.h>
```

Створіть структуру для отримання даних. Ця структура має відповідати визначенню в ескізі відправника.

```
typedef struct struct_message {
 char a[32];
 int b;
 float c;
 String d;
 bool e;
} struct_message;
```

Створити struct_message зі зміною myData.

struct_message myData.

Створіть функцію зворотного виклику, яка буде викликана, коли ESP8266 отримає дані через ESP-NOW. Функція називається onDataRecv() і має прийняти кілька параметрів:

void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {

Ми копіюємо вміст вхідних даних у змінну myData.

memcpy(&myData, incomingData, sizeof(myData));

Тепер, myData структура містить кілька змінних зі значеннями, надісланими відправником ESP8266. Щоб отримати доступ до змінної a, наприклад, нам просто потрібно звернутись до myData.a.

У цьому прикладі ми просто друкуємо отримані дані, але в практичному застосуванні ви можете роздрукувати дані, наприклад, на дисплеї.

```
 Serial.print(«Bytes received: »);
 Serial.println(len);
 Serial.print(«Char: »);
 Serial.println(myData.a);
 Serial.print(«Int: »);
 Serial.println(myData.b);
 Serial.print(«Float: »);
 Serial.println(myData.c);
 Serial.print(«String: »);
 Serial.println(myData.d);
 Serial.print(«Bool: »);
 Serial.println(myData.e);
 Serial.println();
}
```

У setup(), ініціалізувати послідовний зв’язок для цілей налагодження.

Serial.begin(115200);

Налаштуйте пристрій як станцію Wi-Fi.

WiFi.mode(WIFI_STA);

Ініціалізація ESP-NOW:

```
if (esp_now_init() != ESP_OK) {
 Serial.println(«Error initializing ESP-NOW»);
 return;
}
```

Установіть роль ESP8266:

esp_now_set_self_role(ESP_NOW_ROLE_SLAVE).

Зареєструйтеся для функції зворотного виклику, яка буде викликана, коли дані будуть отримані. У цьому разі ми реєструємося на OnDataRecv() функція, яка була створена раніше.

esp_now_register_recv_cb(OnDataRecv).

Тестування зв’язку ESP-NOW. Завантажте ескіз відправника на одну плату, а ескіз одержувача – на іншу. Не забудьте вставити MAC-адресу одержувача в ескіз відправника.

Тепер відкрийте два вікна Arduino IDE, один для одержувача, інший – для відправника. Відкрийте серійний монітор для кожної плати. Це має бути окремий COM-порт для кожної плати.

Це те, що ви повинні отримати на стороні відправника.

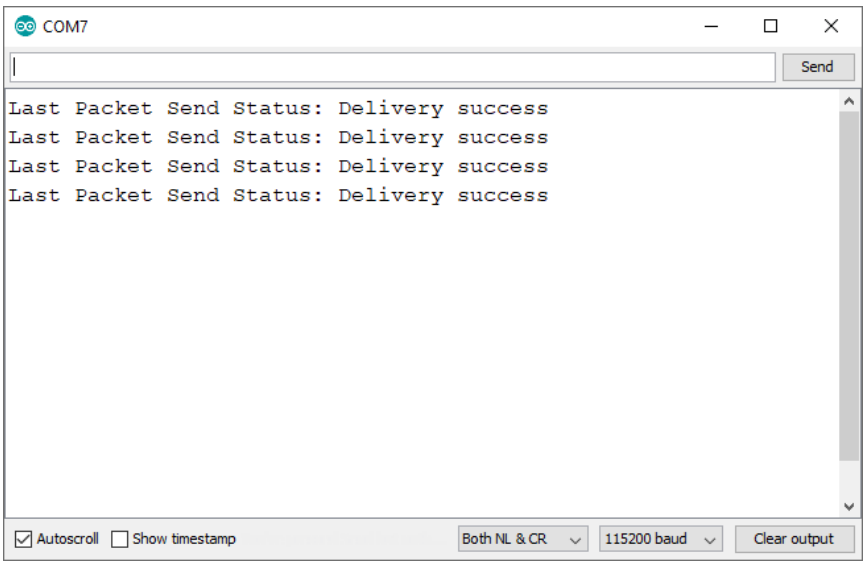

Рисунок 5.6 – Монітор порту

І це те, що ви повинні отримати на стороні приймача. Зауважте, що змінна Int змінюється між кожним отриманим читанням (оскільки ми встановлюємо для неї випадкове число на стороні відправника).

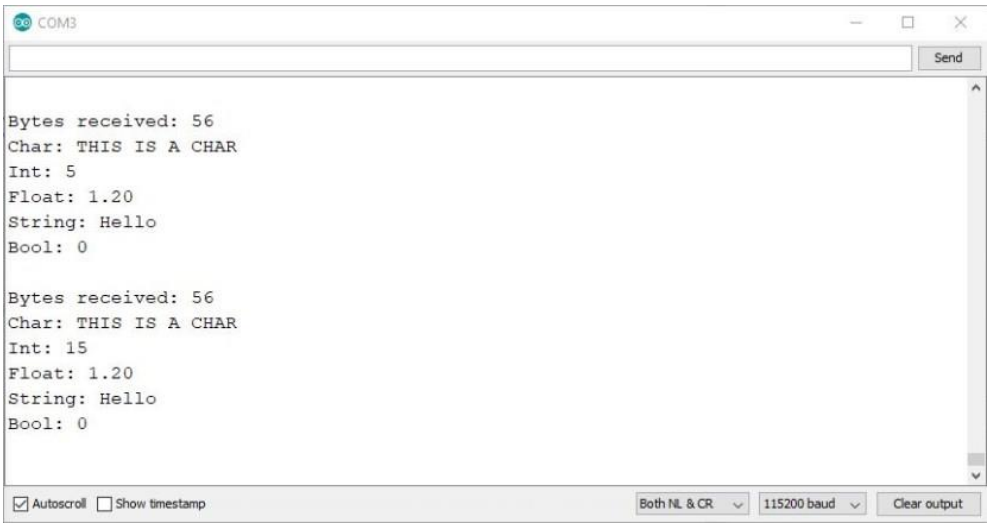

Рисунок 5.7 – Статус посилань

Ми перевірили дальність зв’язку між двома платами, і ми можемо отримати стабільний зв’язок на відстані до 140 метрів (приблизно 459 футів) у відкритому полі. У цьому експерименті обидві бортові антени ESP8266 були спрямовані одна на одну.

Рисунок 5.7 – Перевірена дальність зв’язку у мережі

5.3 Завдання для самостійного виконання

Ознайомитися з порядком виконання роботи та виконати аналогічні операції для передачі текстового повідомлення з власним ім’ям та прізвищем у прямому та зворотньому напрямку мережею ESP-NOW.

5.4 Зміст звіту

1. Короткий опис змісту виконання роботи.
2. Скріншоти та аналітичні дані проведеної роботи.
3. Висновки.

5.5 Контрольні запитання

1. Які типи зв’язку підтримуються мережею ESP-NOW?
2. Як реалізується двосторонній зв’язок у мережі ESP-NOW?
3. Які є функції у ESP-NOW?
4. Як реалізується передача даних у ESP-NOW?
5. Як реалізується прийом даних у ESP-NOW?

2 КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

2.1 Розподіл балів та критерії оцінювання за семестрами

Навчальна дисципліна викладається у 5 семестрі. Формою семестрового контролю є іспит. Критерії оцінювання знань наведені в таблиці 2.1.

Таблиця 2.1 – Розподіл балів за видами занять

Вид занять	Змістовий модуль № 1					Змістовий модуль № 2					Змістовий модуль № 3					Змістовий модуль № 4					Сума
	Т1	Т2	Т3	Т4	Т5	Т6	Т7	Т8	Т9	Т10	Т11	Т12	Т13	Т14	Т15	Т16	Т17	Т18	Т19	Т20	
Лекції	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	10
Лабораторні роботи	1,15	1,15	1,15	4,67	1,15	1,15	1,15	1,15	1,15	1,15	1,15	2,31	2,31	1,15	1,15	1,15	1,15	1,15	1,15	2,31	30
Тести (за змістовими модулями)	10					10					10					10					40
Підсумковий тест (іспит)	20																				20
Усього																					100

2.2 Методика оцінювання й розрахунку балів

Бал за відвідування лекційних занять ураховує кількість лекційних занять, фактично відвіданих студентом.

Для лабораторного практикуму, практичного заняття, розрахунково-графічного завдання кожна контрольна точка (розрахункове завдання, тестове завдання, лабораторна робота) оцінюється за чотирибальною шкалою: B_i = 2, 1,5, 1, 0.

Таблиця 2.2 – Критерії оцінювання завдань з різних видів занять

№	Завдання	Критерії оцінювання
1	Перевірка виконання практичних робіт (у тому числі питання для самостійного опрацювання)	Самостійність, правильність, вчасність виконання завдань, розуміння матеріалу, творчість
2	Виконання й захист лабораторних робіт	Самостійність, правильність, вчасність виконання завдань, розуміння матеріалу та комплексність звіту

Таблиця 2.3 – Відповідність балів і критерії оцінювання розв’язання або захисту окремого завдання

Розподіл балів	Критерії оцінювання
2–1,5 «відмінно»	Повна відповідь, не менше 90 % потрібної інформації, (повне, безпомилкове розв’язування завдання)
1,5–1,0 «добре»	Достатньо повна відповідь, не менше 75 % потрібної інформації, є незначні неточності (повне розв’язування завдання з незначними неточностями)
1,0–0,5 «задовільно»	Неповна відповідь, не менше 60 % потрібної інформації, є деякі помилки (завдання виконане з певними недоліками)
0,5–0 «незадовільно»	Відповідь не відповідає умовам до «задовільно»

До того ж використовується ваговий коефіцієнт τ, що враховує своєчасність виконання навчального плану (τ = 1 – завдання виконане у термін; τ = 0,9 – протягом тижня; τ = 0,8 – пізніше, ніж через тиждень).

Підсумковий бал з чотирибальної шкали перераховується до Nmax бальної.

У разі модульного контролю у формі тестових завдань максимальна відведена кількість балів (15 балів) розподіляється залежно від рівня складності між тестовими завданнями. Також ураховується коефіцієнт τ – своєчасність виконання навчального плану.

Семестровий контроль знань у формі іспиту з використанням екзаменаційних білетів. Білет містить 2 теоретичні питання і 1 практичне завдання. Кожне питання (завдання) оцінюється за чотирибальною шкалою: В = 5, 4, 3, 0. У такому разі використовується ваговий коефіцієнт k, що враховує складність питання (завдання).

Підсумковий екзаменаційний бал з чотирибальної шкали перераховується до 20-бальної. Також ураховується коефіцієнт τ – своєчасність складання іспиту.

Семестрова оцінка студента – арифметична сума балів, набраних студентом з усіх видів контролю.

СПИСОК ЛІТЕРАТУРИ

1. Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi. Random Nerd Tutorials. URL: https://randomnerdtutorials.com/ (date of access: 10.04.2022).
2. 1st V. C., 2nd J. S. R. Internet of Things (iot) in Realtime Applications. INSC International Publisher (IIP), 2020.
3. Afsaruddin A. Jihad: What Everyone Needs to Know: What Everyone Needs to Know ®. Oxford University Press, Incorporated, 2022. 216 p.
4. Business Intelligence for Enterprise Internet of Things/ed. by A. Haldorai, A. Ramu, S. A. R. Khan. Cham: Springer International Publishing, 2020. URL: https://doi.org/10.1007/978-3-030-44407-5 (date of access: 31.01.2022).
5. Data Science and Internet of Things/ed. by G. Fortino et al. Cham: Springer International Publishing, 2021. URL: https://doi.org/10.1007/978-3-030-67197-6 (date of access: 31.01.2022).
6. Deshmukh S. G., Karande K. J., Kolhe M. L. Artificial Intelligence, Internet of Things (IoT) and Smart Materials for Energy Applications. Taylor & Francis Group, 2022.
7. Gupta A. The IoT Hacker’s Handbook: A Practical Guide to Hacking the Internet of Things. Apress, 2019. 340 p.
8. Internet of Things and Cyber Physical Systems/K. Kaushik et al. Boca Raton: CRC Press, 2022. URL: https://doi.org/10.1201/9781003283003 (date of access: 31.10.2022).
9. Internet of Things and Its Applications/ed. by S. Nandan Mohanty, J. M. Chatterjee, S. Satpathy. Cham: Springer International Publishing, 2022. URL: https://doi.org/10.1007/978-3-030-77528-5 (date of access: 31.10.2022).
10. Internet of Things Security and Data Protection/ed. by S. Ziegler. Cham: Springer International Publishing, 2019. URL: https://doi.org/10.1007/978-3-030-04984-3 (date of access: 31.01.2022).

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Основи IoT» для студентів усіх форм навчання зі спеціальності 123 – «Комп'ютерна інженерія» освітньо-професійної програми «Комп'ютерна інженерія» освітнього ступеня «Бакалавр» (частина 1)

Укладачі: д. т. н., проф. А. Л. Перекрест
асист. К. О. Вадурін

Відповідальний за випуск д. т. н., проф. М. І. Гученко

Підп. до др._____. Формат 60×84 1/16. Папір тип. Друк ризографія.
Ум. друк. арк._____. Наклад_____прим. Зам. №. Безкоштовно.

Редакційно-видавничий відділ
Кременчуцького національного університету імені Михайла Остроградського

вул. Першотравнева 20, м. Кременчук, 39600