

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**Л.М. Олещенко, Я.В. Хіцко**

# **ПРОГРАМУВАННЯ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ**

## **ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для студентів, які навчаються  
за спеціальністю 121 «Інженерія програмного забезпечення»  
(освітня програма «Програмне забезпечення комп'ютерних та  
інформаційно-пошукових систем»)

Київ  
КПІ ім. Ігоря Сікорського  
2019

Рецензенти: Бідюк Петро Іванович, д-р техн. наук, проф.  
Полторак Вадим Петрович, канд. техн. наук, доц.

Відповідальний редактор: Лєгеца Віктор Петрович, д-р. техн. наук, проф.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського  
(протокол № 10 від 20.06.2019 р.)  
за поданням Вченої ради факультету прикладної математики  
(протокол № 11 від 18.06.2019 р.)

Електронне мережне навчальне видання

Олещенко Любов Михайлівна, канд. техн. наук  
Хіцко Яна Володимирівна, канд. техн. наук

## **ПРОГРАМУВАННЯ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ**

### **ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Програмування пристроїв Інтернету речей: лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» (освітня програма «Програмне забезпечення комп'ютерних та інформаційно-пошукових систем») / КПІ ім. Ігоря Сікорського ; уклад.: Л.М. Олещенко, Я.В. Хіцко. – Електронні текстові дані (1 файл: 2,46 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 47 с.

Навчальний посібник розроблено для ознайомлення студентів з теоретичними відомостями та практичними прийомами програмування пристроїв Інтернету речей, методами захисту хмарних сервісів Інтернету речей за допомогою середовища Cisco Packet Tracer, а також методами аналізу даних мережі Інтернет та вимогами до виконання лабораторних робіт. Навчальне видання призначене для студентів, які навчаються за спеціальністю 121 Інженерія програмного забезпечення (освітня програма «Програмне забезпечення комп'ютерних та інформаційно-пошукових систем») факультету прикладної математики КПІ ім. Ігоря Сікорського.

© Л.М. Олещенко, 2019

© Я.В. Хіцко, 2019

© КПІ ім. Ігоря Сікорського, 2019

## ЗМІСТ

Вступ.....	4
ЛАБОРАТОРНА РОБОТА № 1. ПРОГРАМУВАННЯ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ ЗАСОБАМИ CISCO PACKET TRACER .....	5
Завдання для лабораторної роботи.....	5
Теоретичні відомості.....	5
Вказівки до виконання лабораторної роботи.....	9
Питання для самоперевірки.....	29
Рекомендована література.....	30
ЛАБОРАТОРНА РОБОТА № 2. ЗАХИСТ ХМАРНИХ СЕРВІСІВ ІНТЕРНЕТУ РЕЧЕЙ.....	31
Завдання для лабораторної роботи.....	31
Теоретичні відомості.....	31
Вказівки до виконання лабораторної роботи.....	32
Питання для самоперевірки.....	36
Рекомендована література.....	36
ЛАБОРАТОРНА РОБОТА № 3. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ВІДКРИТИХ ДАНИХ МЕРЕЖІ ІНТЕРНЕТ .....	37
Завдання для лабораторної роботи.....	37
Теоретичні відомості.....	37
Вказівки до виконання лабораторної роботи.....	37
Питання для самоперевірки.....	47
Рекомендована література.....	47

## ВСТУП

Учбова дисципліна "Організація комп'ютерних мереж" є нормативною і входить до складу циклу професійно-орієнтованих дисциплін навчального плану підготовки бакалаврів, що навчаються за спеціальністю 121 «Інженерія програмного забезпечення» освітньої програми «Програмне забезпечення комп'ютерних та інформаційно- пошукових систем».

Предмет дисципліни – теоретичні і практичні основи організації комп'ютерних мереж, програмування пристроїв Інтернету речей та аналіз даних мережі Інтернет. Мета дисципліни – забезпечити знання теоретичних і практичних основ з організації та функціонування комп'ютерних мереж, уміння застосовувати в професійній діяльності розподілені дані, програми і ресурси мереж, забезпечити знання теоретичних і практичних основ використання мережевих протоколів для програмування пристроїв Інтернету речей.

Метою лабораторного практикуму є ознайомлення студентів з особливостями розроблення програмного забезпечення для роботи у мережі пристроїв Інтернету речей за допомогою середовища моделювання Cisco Packet Tracer, ознайомлення з методами захисту хмарних сервісів Інтернету речей та методами аналізу даних мережі Інтернет.

Навчальний посібник складається з 3 розділів, кожен з яких присвячений виконанню певної лабораторної роботи з дисципліни «Організація комп'ютерних мереж». У кожному розділі надаються теоретичні відомості з певної теми, завдання на лабораторну роботу цієї теми, вказівки до виконання завдання, а також наводяться вимоги до оформлення звіту з виконаної лабораторної роботи, контрольні питання для самоперевірки та список використаної та рекомендованої літератури.

Лабораторний практикум розрахований для студентів, які навчаються у мережевій академії Cisco Networking Academy.

## ЛАБОРАТОРНА РОБОТА № 1. МОДЕЛЮВАННЯ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ ЗАСОБАМИ CISCO PACKET TRACER

**Мета роботи:** навчитися створювати та налаштовувати підключення пристроїв Інтернету речей, використовуючи середовище моделювання Cisco Packet Tracer.

### Завдання для лабораторної роботи

У середовищі Cisco Packet Tracer змоделювати топологію для домашньої мережі згідно сценарію за варіантом та запрограмувати пристрої для їх функціонування у мережі та взаємодії з мережевими пристроями (рис. 1.18). Усі пристрої, домашні планшети та сервери IoT підключені до домашньої бездротової мережі. Кожен пристрій підключений до бездротового маршрутизатора. Налаштувати DHCP, WLAN. Усі бездротові пристрої необхідні для використання однакових ідентифікаторів SSID, пароля та стандартних налаштувань DHCP, крім локального сервера, що використовує 10-клас статичних IP. Також потрібно налаштувати DNS-сервер для перекладу URL-адреси домашньої сторінки IoT у власну IP-адресу сервера IoT. Виконати тестування запрограмованих пристроїв.

### Теоретичні відомості

**Інтернет речей** (англ. Internet of Things, IoT) є концепцією мережі, яка є системою взаємопов'язаних обчислювальних пристроїв, механічних і цифрових машин, об'єктів, тварин або людей, які мають програмне забезпечення, що дозволяє здійснювати передачу та обмін даними між фізичним світом та комп'ютерними системами за допомогою стандартних протоколів зв'язку. Така мережа може мати датчики та виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої можуть зчитувати та приводити в дію, програмувати та ідентифікувати пристрої без участі людини за рахунок використання інтелектуальних інтерфейсів.

Основною концепцією протоколу IP є можливість підключення різних об'єктів (речей), які людина може використовувати у повсякденному житті. Це можуть бути такі пристрої, як кавоварка, холодильник, кондиціонер, автомобіль. Ці об'єкти (речі) повинні мати вбудовані сенсори для обробки інформації, що

надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії в залежності від отриманої інформації. Прикладом є система «розумний будинок» або «розумна ферма». Ця система аналізує дані навколишнього середовища і, наприклад, в залежності від показників, регулює температуру в приміщенні. У зимовий період регулюються інтенсивність опалення, у спеку будинок має механізми відкриття і закриття вікон, завдяки чому будинок провітрюється, при чому все це відбувається без втручання людини.

Для об'єднання побутових речей у мережу необхідно використати декілька технологій. Система унікальної ідентифікації дозволяє збирати та накопичувати інформацію про предмети. Це можна забезпечити за допомогою мікросхем RFID (Radio-Frequency IDentification). Вони здатні без власного джерела струму передавати інформацію приладам зчитування. Кожна мікросхема має індивідуальний номер. Альтернативою даної технології для ідентифікації об'єктів може бути використання ів (Quick Response Code). Для визначення точного місцеположення пристрою використовується технологія GPS (Global Positioning System). Для відслідковування змін у стані елемента або оточуючого середовища об'єкти оснащуються сенсорами.

Для обробки та накопичення даних з сенсорів використовується вбудований комп'ютер (наприклад, Raspberry Pi, Intel Edison тощо). Для обміну інформацією між пристроями можуть бути використані технології бездротових мереж (Wi-Fi, Bluetooth, ZigBee, 6LoWPAN).

Інтеграція з мережею Інтернет передбачає використання IP-адреси як унікального ідентифікатора. Через обмеження адресного простору в IPv4 пристроям потрібно використовувати IPv6, який забезпечує унікальними адресами мережевого рівня не менше 300 млн. пристроїв на одного жителя Землі. Об'єктами в IP будуть не тільки пристрої із сенсорними можливостями, але й пристрої, які виконують дії (наприклад, лампочки або замки, якими можна керувати через мережу Інтернет).

Важливими характеристиками бездротової передачі даних для побудови IoT є ефективність, відмовостійкість, адаптивність та можливість самоорганізації. Стандарт IEEE 802.15.4 управляє доступом для організації енергоефективних

персональних мереж. Наприклад, сонячне випромінювання заряджає сонячний акумулятор, що надсилає електроенергію на акумулятор для зберігання та розподілу енергії. Вимірник потужності, з'єднаний між ними, зчитує та відображає кількість енергії, яку захоплює сонячна панель. Оскільки усі пристрої підключені, вони реєструються на сервері реєстрації, дозволяючи користувачеві контролювати всю систему з веб-браузера. Коли пристрої належним чином підключені, вони повинні бути налаштовані. Оскільки система покладається на IP-мережу, пристрої повинні бути налаштовані з правильною інформацією про IP-адреси. Оскільки сервер налаштований також на роль сервера DHCP, пристрої IoT повинні бути налаштовані як клієнти DHCP, щоб автоматично вивчати інформацію IP. Користувач може контролювати енергоспоживання системи не тільки локально, але і дистанційно. Ще один крок – підключити мікроконтролер та написати код для вимкнення одного або декількох світлодіодів, коли рівень заряду акумулятора стане нижчим заздалегідь визначеного порогу. Цей режим енергозбереження дозволить батареї деякий час перезаряджатись.

Середовище моделювання Cisco Packet Tracer (CPT) має наступні можливості для створення та налаштування IoT рішень:

- підтримка SPAN / RSPAN;
- підтримка L2NAT;
- підтримка PTP;
- підтримка REP;
- підтримка LLDP;
- підтримка розподілу адрес на основі портів DHCP-сервера;
- PoEsupport;
- підтримка протоколу IoTSwitch.

На рис. 1.1 зображені пристрої, які можна використовувати як рішення IoT для моделювання та тестування їх роботи в середовищі Cisco Packet Tracer.

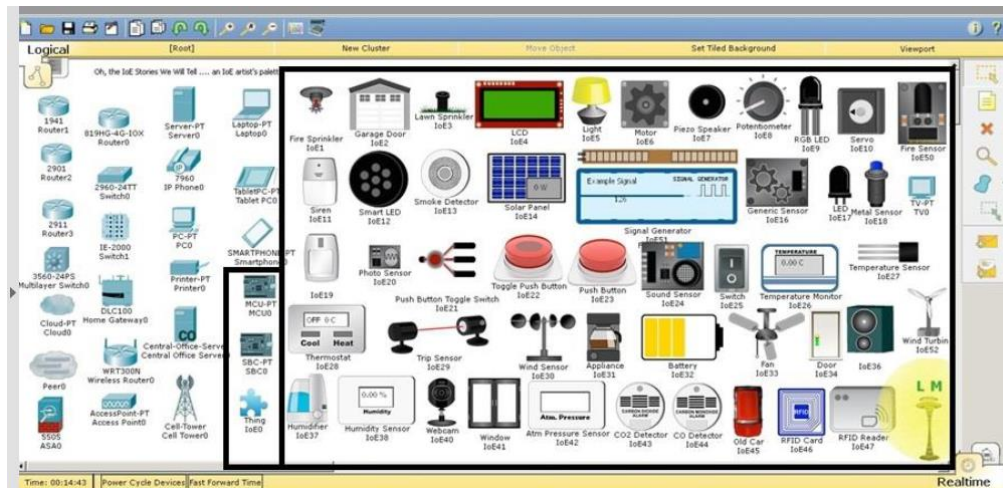


Рис.1.1. Пристрої IoT у середовищі моделювання Cisco Packet Tracer

Cisco Packet Tracer дозволяє використовувати різні фізичні середовища, «розумні пристрої», датчики і активатори, Smart Home, Smart City, Industrial, Power Grid, редагувати існуючі або програмувати власні пристрої, використовуючи технології Python, Javascript, Blockly, SBC та MCU, Home Gateway та RegServer, налаштовувати правила роботи пристроїв. Cisco Packet Tracer містить приклади файлів, навчальні посібники та підручники для моделювання IoT рішень (рис.1.2).

 Two screenshots from the Cisco Packet Tracer interface. The top one shows the 'File' menu with 'Open Samples...' highlighted. The bottom one shows the 'Contents' dropdown menu with 'Tutorials' selected.
 

JavaScript API			
Program Structure and Events			
Function	Return Type	Description	Example
setup()	N/A	If defined, this function is called once when the program starts.	<pre>function setup() {   pinMode(0, INPUT); }</pre>
loop()	N/A	If defined, this function is called continuously when the program is running. The frequency of the calls depends on the complexity of this function, the number of	<pre>function loop() {   Serial.println(digitalRead(0)); }</pre>

Рис. 1.2. Довідкові матеріали Cisco Packet Tracer



Блок мікроконтролера MCU (multipoint control units) – це невеликий комп'ютер, побудований на чіпі SoC (System on a Chip). Він містить ядро процесора, пам'ять та периферійні пристрої введення / виведення. Мікроконтролери призначені для вбудованих програм або програм, які не потребують багато ресурсів комп'ютера.

Мікропроцесори, що використовуються в персональних комп'ютерах, зазвичай використовуються для підтримки інших програм загального призначення, що вимагають більшого обсягу ресурсів комп'ютера.

Прикладами програм, що використовують мікроконтролери, є системи керування автомобільними двигунами, медичні пристрої, пульти дистанційного керування, офісні машини, електричні інструменти, іграшки.

Мікроконтролери змішаного типу також є інтегруючими аналоговими компонентами, вони необхідні для управління нецифровими електронними системами. Cisco Packet Tracer забезпечує підтримку емулятора MCU. Користувач може запрограмувати RT MCU для виконання завдань, подібних до MCU у реальному світі. Щоб спростити процес, RT MCU може бути запрограмований за допомогою Java і Python.

RT MCU – це плата, що має порт USB, шість цифрових портів вводу-виводу та чотири аналогові порти вводу-виводу. Цифрові входи / виходи на RT MCU дозволяють користувачеві підключати цифрові датчики та виконавчі пристрої. Аналогові порти вводу-виводу дозволяють користувачеві підключати аналогові датчики та виконавчі пристрої. Python працює на різних комп'ютерних платформах. Cisco Packet Tracer використовує цю перевагу та реалізує пристрої IoT, що підтримують Python.

### **Вказівки до виконання лабораторної роботи**

Для налаштування роботи «розумного пристрою» – Smart Greenhouse у середовищі моделювання Cisco Packet Tracer встановимо такі умови відносно змін температури:

- якщо температура занадто висока, відкривається шарнірна вентиляція;
- порогова температура може бути встановлена з сервера.

У нижньому меню вікна Cisco Packet Tracer створюємо нову річ – натискаємо "Thing", задаємо Name.

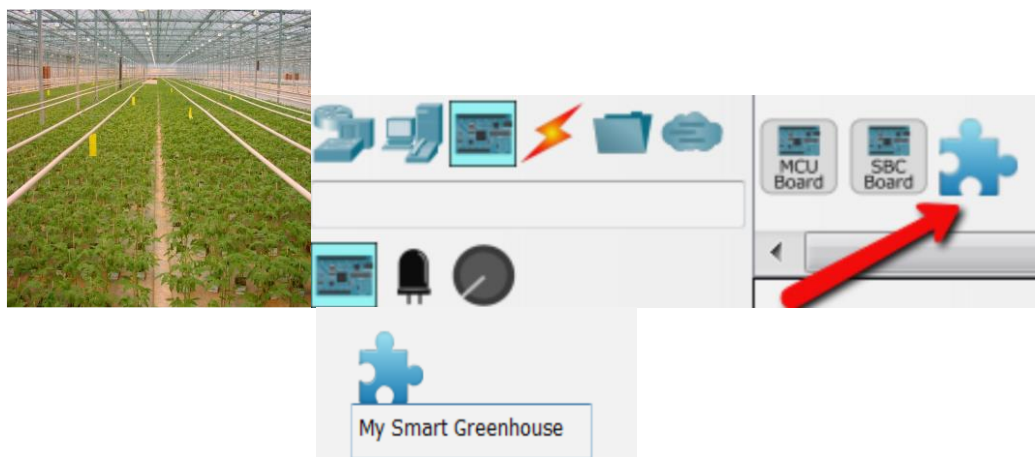


Рис. 1.3. Додавання нового «розумного пристрою» та його імені в Cisco Packet Tracer

Виконуємо налаштування вводу / виводу:

- переходимо на вкладку Конфігурація вводу / виводу;
- виконуємо налаштування інтерфейсу:
- обираємо дротову мережу;
- вибираємо один цифровий слот;
- немає аналогових або USB-портів;
- обираємо тип: смарт-пристрій (рис.1.4).

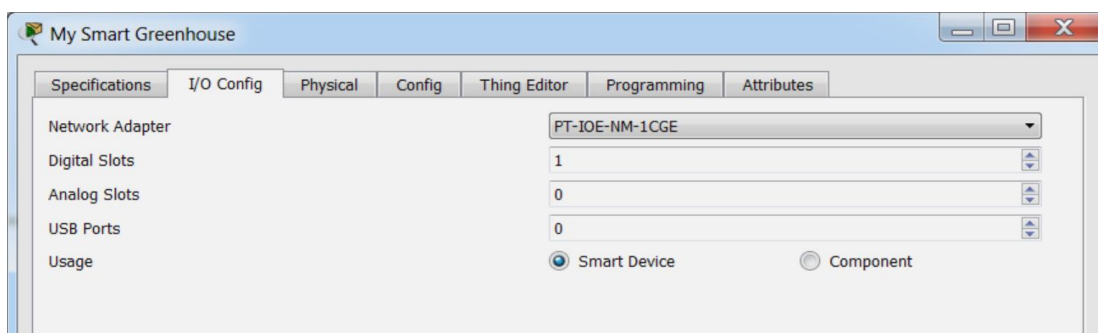


Рис. 1.4. Налаштування конфігурації пристрою в Cisco Packet Tracer

Для наочного відображення роботи пристрою створимо зображення закритої та відкритої теплиці. Для цього перейдемо на вкладку "Редактор теми", пристрій буде містити лише один компонент. Створимо простий значок для закритого стану, змінимо його для відкритого стану, натиснемо «Нове», щоб додати зображення в цифровий слот 0 (рис.1.5).

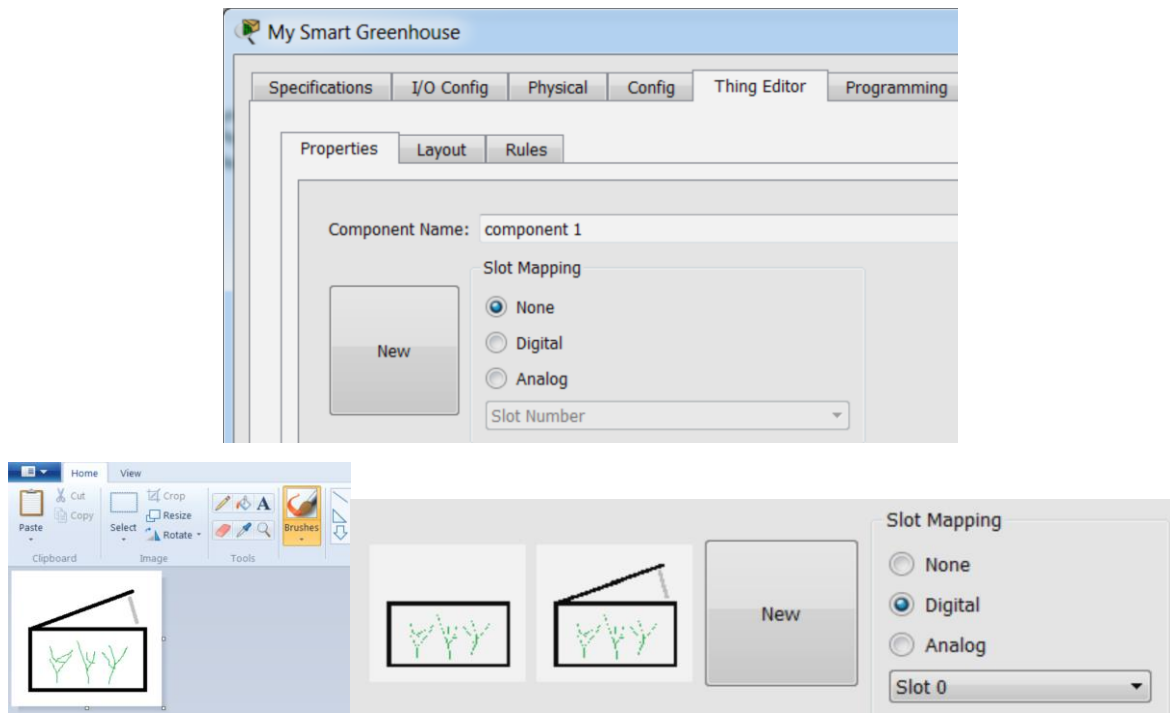


Рис.1.5. Додавання зображень різного стану пристрою

Призначимо зображення для значень слотів:

- зображення змінюватимуться, коли змінюється значення слоту;
- перейдемо до Редактор теми – Правила.

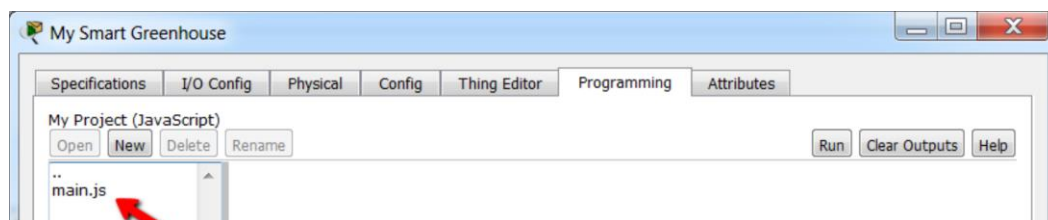
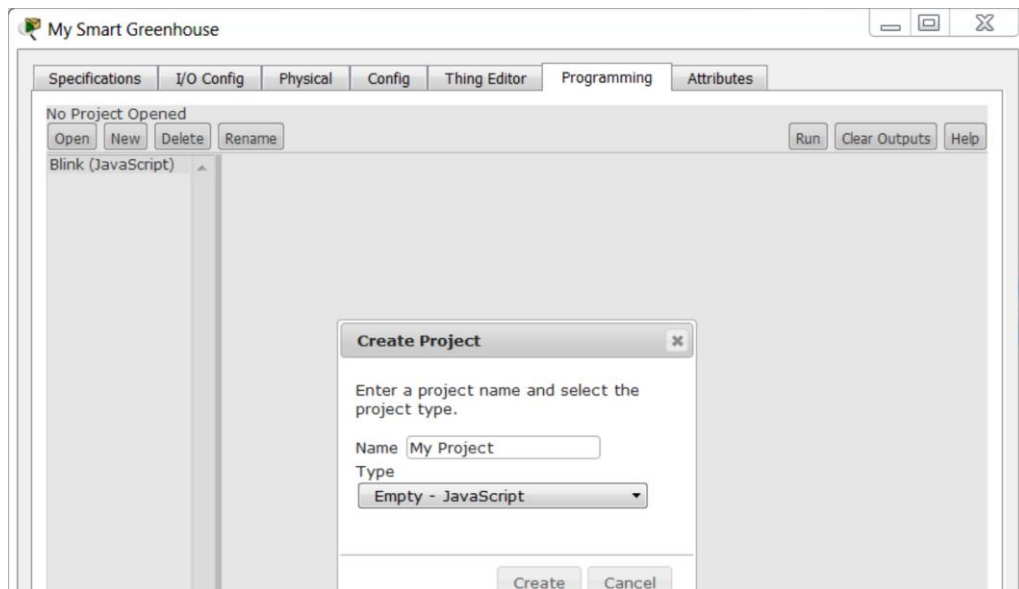
Створимо два правила:

- перше правило відображає значення LOW для "закритого" зображення;
- друге правило називає високе значення "відкритим" зображенням (рис. 1.6).



Рис. 1.6. Створення правил для різних станів пристрою

На вкладці «Програмування» створимо новий проект у JavaScript або Python (у даному випадку використовується JavaScript). Відкриваємо проект, потім файл main.js, який буде порожнім та запишемо код для відкриття та закриття парнику залежно від зміни температури (рис.1.7).



```

var state = 0; // 0 закрито, 1 відкрито
var temp = 0; // поточна температура
var threshold = 30; // значення температури для відкриття парнику

function setup () {
  digitalWrite (0, state) ;
}

function loop () {
  var temp = Environment.get ("Ambient Temperature");
  // зчитувати значення температури з навколишнього середовища
  Serial.println ("Current temperature: "+temp);
  // вивести значення температури в консоль
  if (temp > threshold) {
    // якщо значення температури завелике, змінити стан
    state = 1;
  } else {
    state = 0 ;
  }
  digitalWrite (0, state);
  // зробити поточний стан успішним для запису в слот
  delay (1000); // очікування 1 секунду
}

```

Рис. 1.7. Створення проекту у JavaScript для «розумного пристрою» Smart Greenhouse

Пристрій повинен автоматично відкривати вентиляційний отвір, якщо занадто спекотно, для цього можна встановити порогову температуру до 30 ступенів.

Щоб протестувати це, потрібно зробити середовище гарячим:

- запускаємо новий екземпляр Packet Tracer;
- переходимо до файлу - Відкрити зразки - 7,0 - Environment -thermostat.pkt;
- вибираємо і копіюємо [Ctrl-C] усі три пристрої;
- вставляємо [Ctrl-V] у свій PT поруч із «розумним пристроєм»;
- клацаємо Alt на пристрій термостату, щоб перейти в режим HEAT (рис.1.8);
- нагріваємо і охолоджуємо приміщення та спостерігаємо за тим, як на це реагує пристрій.

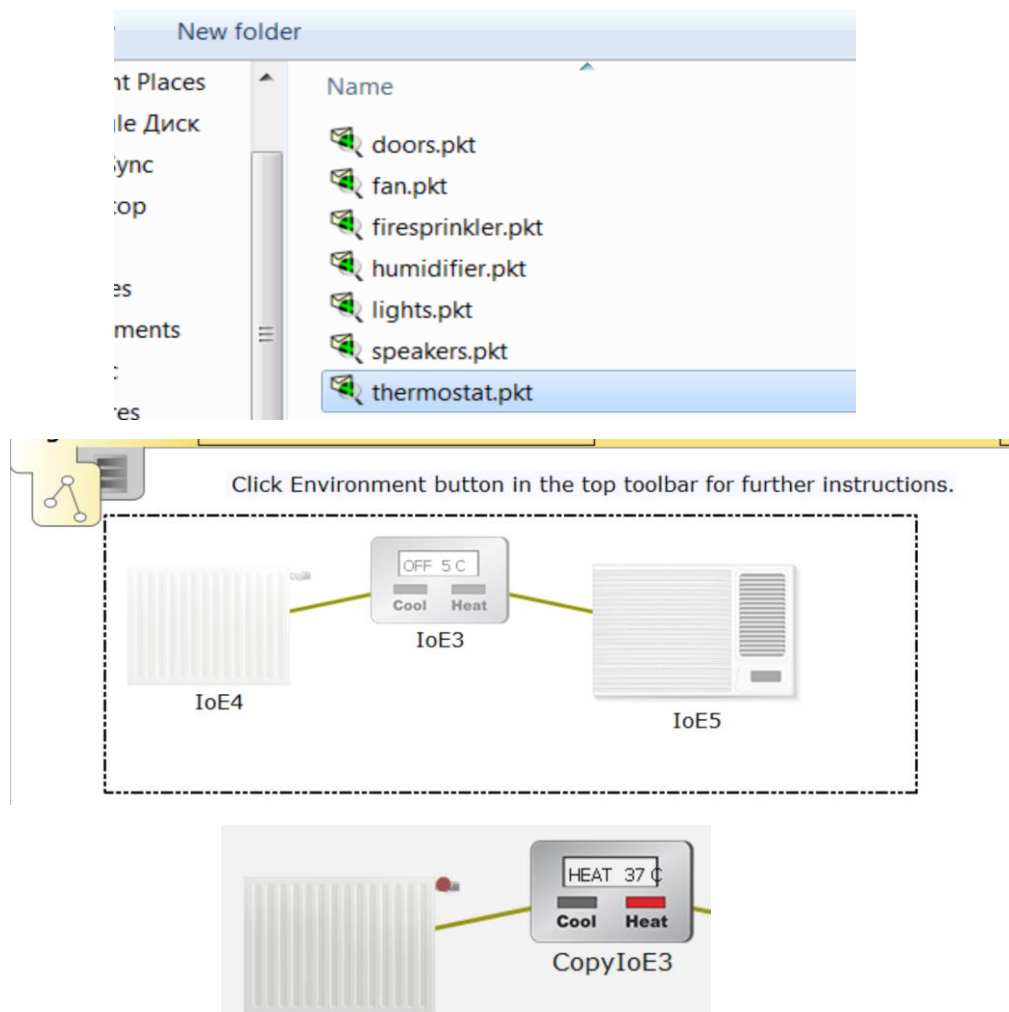


Рис. 1.8. Тестування роботи «розумного пристрою» в Cisco Packet Tracer шляхом зміни температури у приміщенні

Щоб пристрій міг взаємодіяти з реєстраційним сервером (Reg Server), потрібно додати додаткові вказівки – параметри обміну (поточний стан «відкрито»/ «закрито», поточну температуру та поріг температури, який може бути змінений з реєстраційного серверу):

```
IoEClient.setup(  
  {  
    type: "Smart Greenhouse",  
    states: [{  
      name: "Status",  
      type: "options",  
      options: {  
        "0": "Closed",  
        "1": "Open",  
      }  
    }],  
    {  
      name: "Temperature",  
      type: "number",  
      unit: "&deg;C",  
      decimalDigits: 1  
    }  
  },  
  {  
    name: "Threshold Temperature",  
    type: "number",  
    unit: "&deg;C",  
    decimalDigits: 1;  
    controllable: true,  
    minValue: 10,  
    maxValue: 100  
  }  
});  
IoEClient.onInputReceive = function (input) {  
  if (input) {  
    input = input.split(",");  
    threshold = parseInt(input[2]);  
  }  
};
```

Для повідомлення поточних значень трьох параметрів (стан, температура, поріг) до сервера Reg необхідно ввести рядок:

```
IoEClient.reportStates(state+ "," +temp+ "," +threshold);
```

Налаштуємо шлюз для отримання IP-адреси, щоб шлюз міг взаємодіяти з IoT пристроєм, також налаштуємо шлюз для DHCP. На ПК перейдемо до 192.168.25.1, використаємо "адміністратор" як ім'я користувача та пароль.

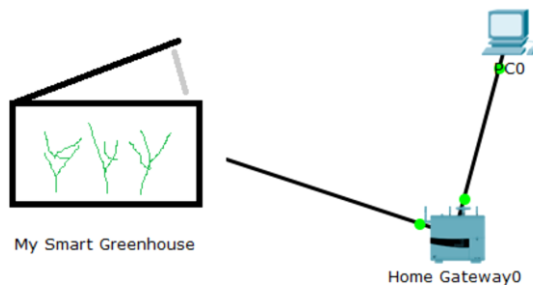


Рис. 1.9. Підключення «розумного пристрою» до домашнього шлюзу в Cisco Packet Tracer

The screenshot shows the configuration interface for the 'My Smart Greenhouse' device. The top navigation bar includes tabs for Physical, Config, Desktop, Attributes, and Software/Services. The 'Config' tab is active, showing the 'FastEthernet0' interface configuration. The 'IP Configuration' section is set to 'DHCP' with an IP address of 192.168.25.102 and a subnet mask of 255.255.255.0. Below this, another navigation bar includes Specifications, I/O Config, Physical, Config, Thing Editor, Programming, and Attributes. The 'Config' tab is active, showing 'Global Settings' for the device. The 'Display Name' is 'My Smart Greenhouse' and the 'Serial Number' is 'PTT08100016'. The 'IoE Server' is set to 'Home Gateway'. At the bottom, a summary panel for 'My Smart Greenhouse (PTT08100016)' shows its status as 'Closed', current temperature as 16.1 °C, and a threshold temperature of 20.0 °C.

Рис. 1.10. Налаштування шлюза для отримання IP-адреси

Підключимося до сервера.

- Підключимось до мережі Wi-Fi.
- SSID: netacad.
- Задаємо пароль.

Додамо багатокористувацьку хмару до робочої області (Multiuser Connection). Налаштування багатокористувацької хмари:

- Тип (Connection Type): вихідний (Outgoing);
- Адреса (Peer Address): 192.168.1.101;
- Порт (Peer Port Number): 38000;
- Назва мережі (Peer Network Name): виберіть унікальне ім'я.

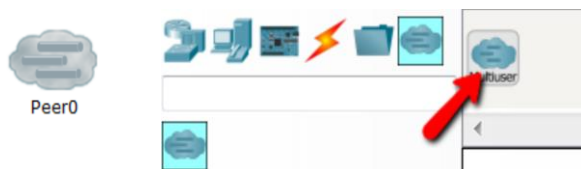
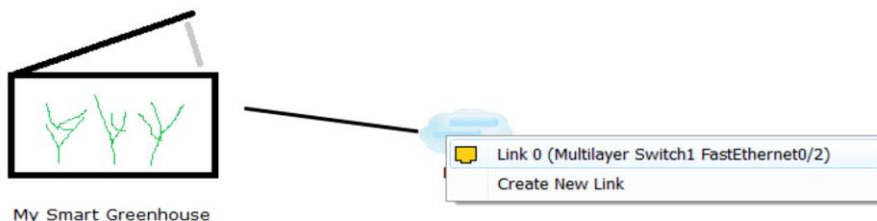


Рис. 1.11. Налаштування багатокористувацької хмари в Cisco Packet Tracer

Підключимо «розумний пристрій» до багатокористувацької хмари та налаштуємо його для використання DHCP.

Реєстраційний сервер:

- Адреса: 1.1.1.1.
- Ім'я користувача: IoT.
- Пароль: system.
- Натискаємо "Підключити".



IP Configuration	
<input checked="" type="radio"/> DHCP	
<input type="radio"/> Static	
Default Gateway	192.168.25.1
IP Address	192.168.25.101

IoE Server	
<input type="radio"/> None	
<input type="radio"/> Home Gateway	
<input checked="" type="radio"/> Remote Server	
Server Address	1.1.1.1
User Name	IoT
Password	system
<input type="button" value="Connect"/>	

Рис. 1.12. Підключення і налаштування Reg Server в Cisco Packet Tracer



Пристрій IoT повинен стати видимим на IoT Server. Звітами пристрою будуть поточна температура і статус. Порогова температура може бути встановлена з сервера.



Рис. 1.13. Управління станом температури «розумного пристрою» з веб-браузера

У Cisco Packet Tracer можна взаємодіяти з реальним світом, використовуючи протоколи TCP, UDP, HTTP. Функції, які допомагають це зробити, описані у JavaScript API (PT → Help → Contents):



Рис. 1.14. Функції JavaScript API для програмування «розумних пристроїв»

Знайдемо рядок, у якому ми читаємо значення температури у середовищі моделювання CPT та замінимо його на виклик HTTP-сервера:

```
//var temp = Environment.get("Ambient Temperature");
RealHTTPClient.get("http://192.168.1.101:5000/temp", function(status, data) {temp = data;});
```

Використаємо IP-адресу 192.168.1.101 та порт 5000. Тепер пристрій повинен діяти залежно від зміни реальної температури приміщення. Пристрій також може безпосередньо відображати поточну температуру:

```
setCustomText (50, 130, 100, 20, temp) ;
```



Здійснимо візуалізацію даних за допомогою одноплатного комп'ютера SBC (single-board computer), Для цього потрібно виконати наступні кроки.

- Відкрити графічний зразок з "7.\* - ioe - graphing.pkt".
- Вибрати SBC0.
- Скопіювати SBC0 у файл PKT.
- Змінити введення / виведення пристрою та додати ще один цифровий слот для підключення до SBC.
- Змінити код пристрою в SBC для візуалізації.

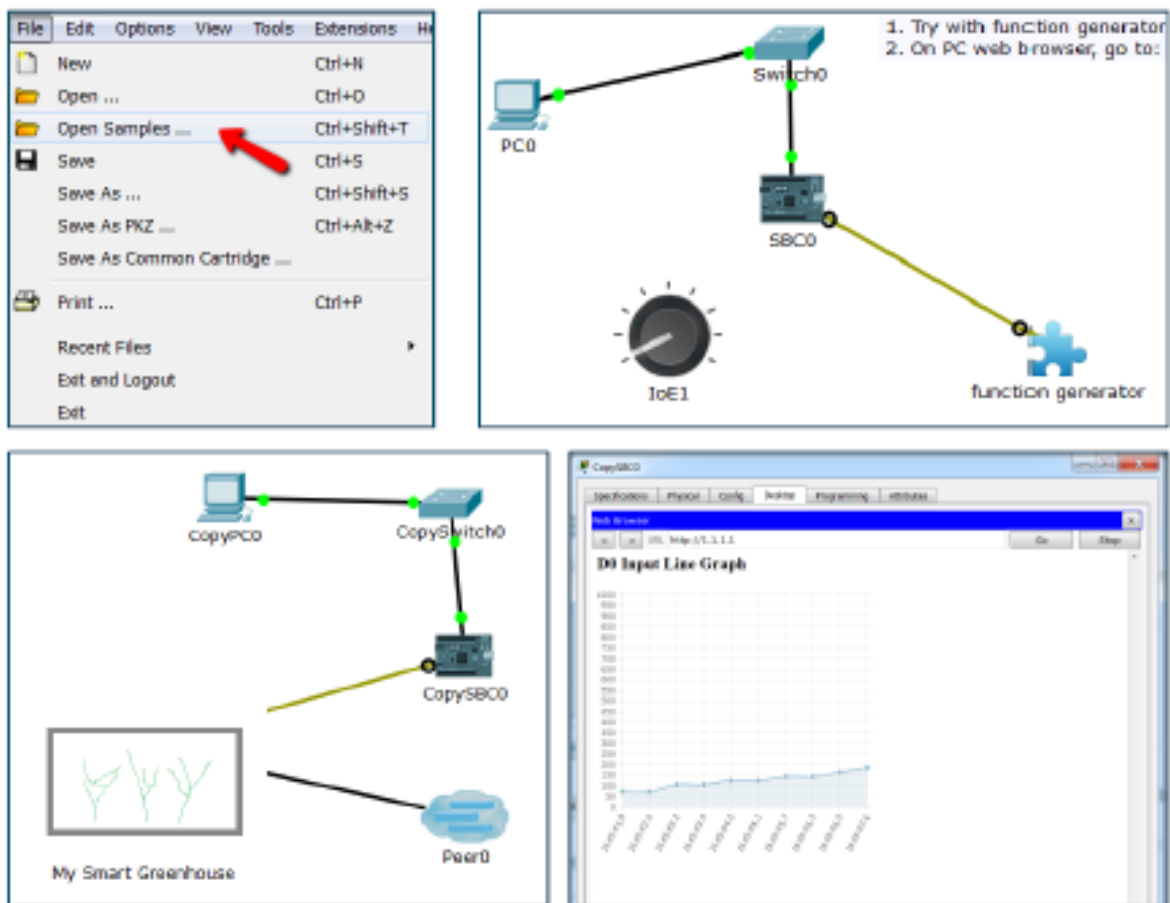


Рис. 1.15. Візуалізація даних за допомогою одноплатного комп'ютера SBC

Створимо RT MCU, підключений до чотирьох датчиків (цифровий вимикач, цифрова топографічна кнопка, аналоговий потенціометр та аналоговий датчик) і чотирьох виконавчих механізмів (світло, світлодіодний індикатор, сирена та аналоговий двигун) (рис.1.16). RT MCU програмується за допомогою Python для неперервного читання значень датчика і запису на виконавчі елементи, керовані

умовною логікою. Співвідношення між датчиками та виконавчими елементами можна визначити за допомогою перемикача, який керує світлом; кнопки натискання, що керує світлодіодним індикатором; потенціометра, що контролює сирену та Flex Sensor, що керує двигуном.

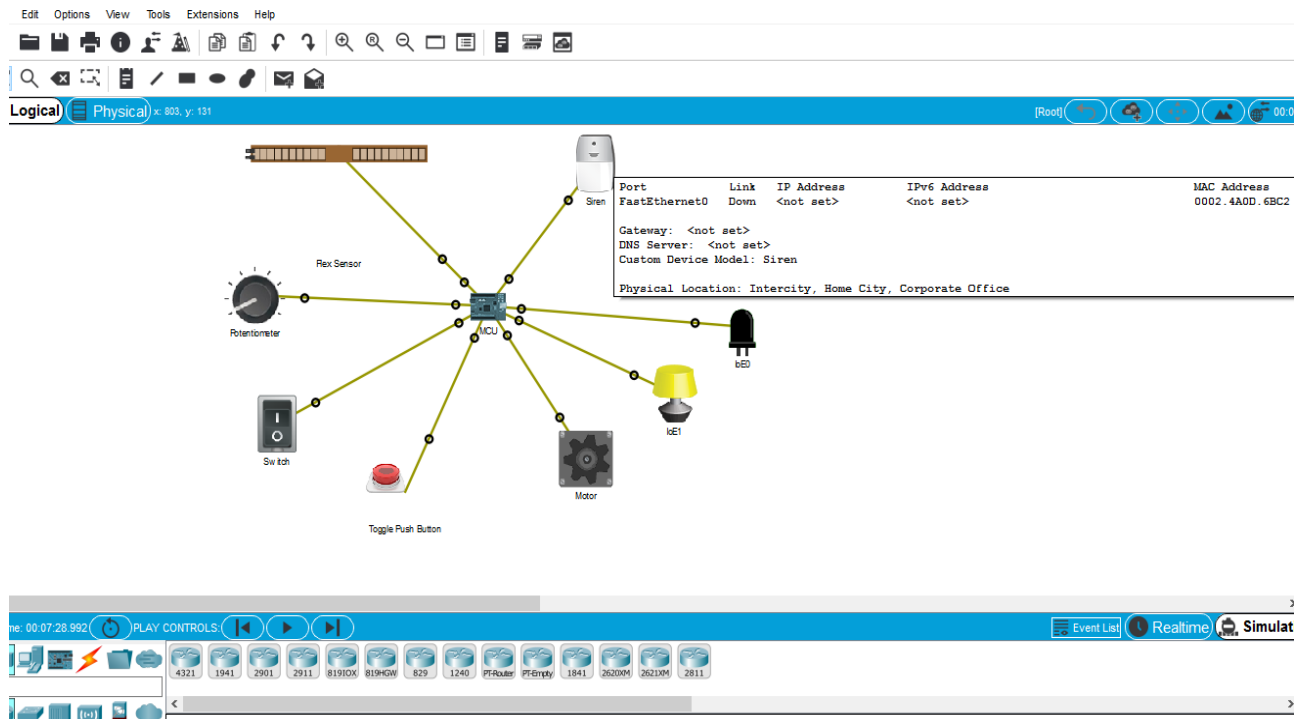


Рис. 1.16. MCU, підключений до чотирьох датчиків і чотирьох виконавчих механізмів в СРТ

На рис.1.16. MCU розміщений у центрі. Вхідні пристрої (датчики та перемикачі) розташовані ліворуч, а пристрої виходу – праворуч.

Виберемо MCU, датчики, перемикачі та керовані пристрої, щоб відкрити їх вікна налаштування. На різних пристроях доступні різні вкладки.

У вікні MCU вкладка «Програмування» містить код Python, який визначає поведінку «розумного пристрою». Утримуючи клавішу ALT під час натискання на пристрій, ми можемо з ним взаємодіяти: ALT + перемикач (для увімкнення / вимкнення світла); ALT + кнопка (для увімкнення / вимкнення світлодіоду); ALT + клацнути та утримувати потенціометр (для контролю гучності сирени); ALT + клацнути та утримувати сенсорний регулятор (для керування швидкістю двигуна).

Відкриємо MCU, розташований у центрі топології. Виберемо вкладку «Програмування», щоб отримати доступ до коду Python, що працює на MCU.

Нижче наведено короткий опис задач, які виконує даний код.

```

1 from gpio import * # imports all modules in the GPIO library
2 from time import * # imports all modules in the time library
3
4 switchValue = 0 # initialize Switch sensor value global variable to 0
5 togglePushButtonValue = 0 # initialize Toggle Push Button sensor value global variable to 0
6 potentiometerValue = 0 # initialize Potentiometer sensor value global variable to 0
7 flexSensorValue = 0 # initialize Flex Sensor value global variable to 0
8
9 def readFromSensors():
10     global switchValue # declare switchValue as global
11     global togglePushButtonValue # declare togglePushButtonValue as global
12     global potentiometerValue # declare potentiometerValue as global
13     global flexSensorValue # declare flexSensorValue as global
14
15     switchValue = digitalRead(0) # read Switch sensor value
16     togglePushButtonValue = digitalRead(1) # read Toggle Push Button sensor value
17     potentiometerValue = analogRead(A0) # read Potentiometer sensor value
18     flexSensorValue = analogRead(A1) # read Flex Sensor value
19
20 def writeToActuators():
21     if (switchValue == HIGH): # evaluates to True if the Switch sensor value is digital HIGH
22         customWrite(2, 1) # turn on the Light
23     else:
24         customWrite(2, 0) # turn off the Light

```

У рядках 1 і 2 усі класи з бібліотек **time** та **gpio** імпортуються в програму для надання доступу до функцій **time** та **gpio**.

У рядках 4, 5, 6 і 7 оголошуються та ініціалізуються чотири глобальні змінні для зберігання значень датчика.

Створюється функція **readFromSensors()** (рядки від 9 до 18), яка готує змінні датчика (рядки з 10 по 13) та запускає дві інші функції **digitalRead()** і **analogRead()** для визначення стану датчиків і зберігає їх у відповідних змінних (рядки від 15 до 18). Підключивши датчики, програма зможе захоплювати конкретний сенсорний статус.

У рядках від 20 до 39 створюється функція **writeToActuators()** для зміни статусу виконавчих елементів на основі стану датчиків. У рядках з 21 по 24 програма перевіряє вміст змінної **switchValue**. Оскільки **switchValue** зберігає статус (рядок 15), програма може вирішити, чи повинен він включити світло; якщо значення, збережене в **switchValue**, дорівнює HIGH (застосовується напруга або перемикач увімкнено), програма вмикає світло, записуючи значення 2 на привід 2. Якщо **switchValue** дорівнює LOW (без напруги або перемикача, якщо OFF), програма вимикає світло, записуючи 0 на виконавчий механізм 2.

Рядки 26 - 39 відповідають за випробовання і модифікацію інших виконавчих елементів на основі їх відповідних контрольних датчиків.

```

26 ▾ if (togglePushButtonValue == HIGH): # evaluates to True if the Toggle Push Button sensor value
27 is digital HIGH, otherwise false
28 ▾     digitalWrite(3, HIGH) # turn on the LED
29 else:
30     digitalWrite(3, LOW) # turn off the LED
31 ▾
32 if (potentiometerValue > 512): # evaluates to True if the Potentiometer is turned at least half way
33 ▾     customWrite(4, HIGH) # turn on the Siren
34 else:
35     customWrite(4, LOW) # turn off the Siren
36 ▾
37 if (flexSensorValue > 0): # evaluates to True if the Flex Sensor is bent, otherwise false
38 ▾     analogWrite(5, flexSensorValue) # turn on the motor with speed equal to the Flex Sensor value
39 else:
40     analogWrite(5, 0) # turn off the motor

```

Рядки з 41 по 54 визначають функцію **main()**, яка автоматично виконується, коли MCU увімкнено. Рядки з 42 по 46 ініціалізують піни; піни 0 і 1 налаштовуються як INPUT (лінії 42 і 43), тоді як піни 2, 3 і 4 налаштовуються як OUTPUT. Це важливо, оскільки піни INPUT отримують напругу, а піни OUTPUT випромінюють напругу, що генерується самим MCU.

```

41 ▾ def main(): # defines the main function
42     pinMode(0, IN) # sets digital slot 0 (Switch) to input
43     pinMode(1, IN) # sets digital slot 1 (Toggle Push Button) to input
44     pinMode(2, OUT) # sets digital slot 2 (Light) to output
45     pinMode(3, OUT) # sets digital slot 3 (LED) to output
46     pinMode(4, OUT) # sets digital slot 4 (Siren) to output
47     pinMode(5, OUT) # sets digital slot 5 (Motor) to output
48
49 ▾     while True: # loop indefinitely
50         readFromSensors() # call the readFromSensors function
51         writeToActuators() # call the writeToActuators function
52         delay(1000) # delay script execution for 1000 ms
53
54 ▾ if __name__ == "__main__": # Evaluates to True if this module is the script
55     main() # call the main function

```

У рядках 48-51 створюється нескінченний цикл **while**. Оскільки умова у циклі просто визначає істину, MCU буде виконувати рядки 49, 50 і 51 завжди. Цей безмежний цикл підсилює мікроконтролер. Для цього потрібно:

1. запустити функцію **readFromSensors()** у рядку 49;
2. запустити функцію **writeToActuators()** у рядку 50;
3. зачекати 1 секунду у рядку 50 (1000 мс = 1 секунда);
4. перезапустити цикл зверху, повернувшись до рядка 49 та знову запустивши функцію **readFromSensors()**.

Хоча нескінченні цикли небажані, проте це корисно у даній програмі, оскільки нескінченний цикл гарантує, що MCU завжди перевіряє стан датчиків і перемикачів (запускаючи **readFromSensors()** кожну секунду) і завжди виконує

відповідні дії до виконавчих механізмів (запустивши `writeToActuators()`), виходячи з стану датчиків.

Виконаємо налаштування дозволу кавоварці почати готувати каву, коли хтось заходить на кухню (коли рух виявлено), при цьому також повинна вмикатися лампа. Топологія даної мережі зображена на рис. 1.17. Додамо пристрої до робочого простору Packet Tracer.

Виберемо SBC у розділі Components >> Boards >> SBC-PT.



Кавоварку та лампу вибираєм у розділі End Devices >> Home >> Appliance / Light.



Датчик руху можна знайти в розділі Components >> Sensors >> Motion Sensor.

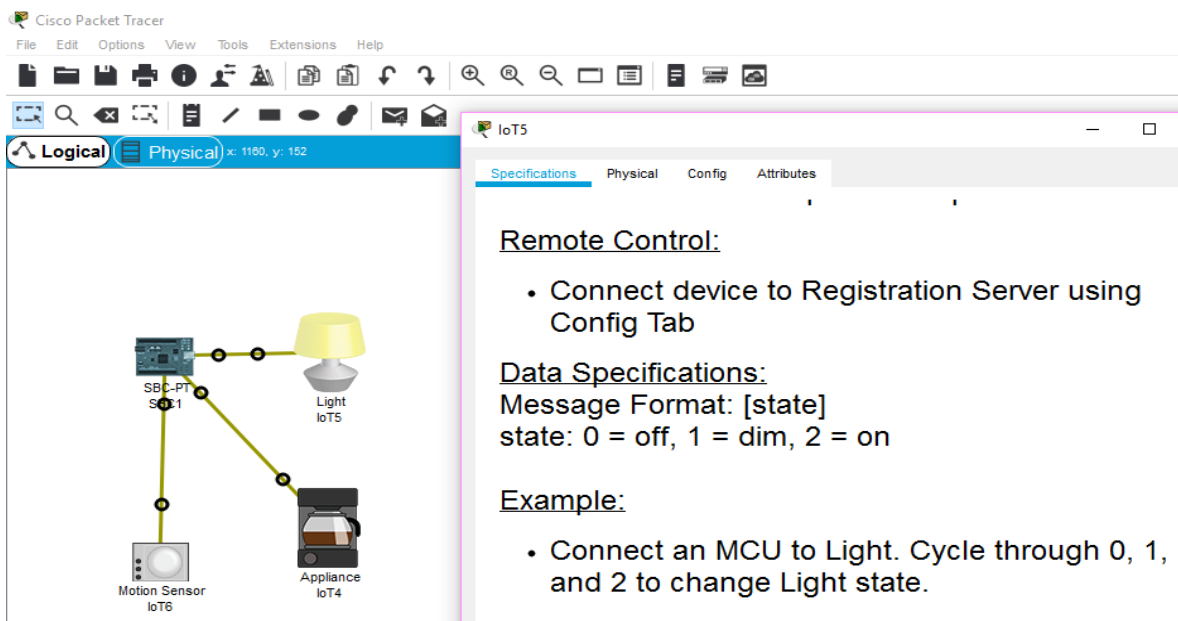


Рис.1.17. Топологія мережі IoT- пристроїв в СРТ та перегляд специфікацій пристрою

Змінимо назви пристроїв та співставимо пристрої SBC портам згідно табл. 1.1. Використовуючи спеціальні кабелі ІоЕ, підключимо пристрої до плати РТ-SBC. Спеціальний кабель ІоЕ знаходиться у розділі «Підключення».

Таблиця 1.1. Співставлення пристроїв SBC портам

Пристрій	SBC Порт
Кавоварка (Coffee Maker)	D1
Лампа (Lamp)	D2
Датчик руху (Motion Sensor)	D9

SBC імітує реальні одноплатні комп'ютери, такі як Raspberry Pi. Однією з переваг SBC-PT є те, що вона може бути запрограмована на Python.

Натиснемо на SBC і виберемо вкладку «Програмування» для перегляду коду. Вміст вкладки «Програмування» розділений на дві частини. На лівій панелі відображаються усі файли, що містять створені програми. На правій панелі відображається вміст вибраного файлу в лівій частині. Натиснемо кнопку "Створити", розташовану ліворуч, щоб створити новий файл. Введемо ім'я main.py і виберемо Empty - Python для свого типу. На лівій панелі вказуємо файл main.py і натискаємо Open; кнопка Відкрити розташована безпосередньо над лівою панеллю. Права панель тепер готова отримати код для програми main.py.

Використовуючи праву панель, напишемо програму для SBC, щоб включати кавоварку та світло, коли детектор виявляє рух.

```
1. from gpio import*
2. from time import*
3. def main():
4. while True:
5. motion_sensor = digitalRead(9)
6. if motion_sensor ==HIGH:
7. print("Someone's awake.");
8. print("Making Coffe...");
9. customWrite(1,1)
10. customWrite(2,1)
11. delay(6000)
12. print("Done. Coffee is ready.");
13. customWrite(1,0)
14. customWrite(2,0)
15. delay(500)
16. if _name_ == "_main_":
17. main ()
```

Рядки 1 та 2 включають програмні модулі **gpio** та **time**, які необхідні для функцій **digitalRead ()**, **customWrite ()** та **delay ()**.

Рядок 3 визначає функцію **main()**.

Рядок 4 визначає цикл **while**. Оскільки умова завжди відповідає дійсності (логічне значення True), цикл буде виконуватися завжди.

У рядку 5 зчитується цифровий сигнал з контакту 9 (сигнал від детектора PT-Motion) і зберігається у змінній **motion\_sensor**. PT-Motion Detector надсилає HIGH або LOW, щоб повідомити про рух або його відсутність, відповідно.

Рядок 6 перевіряє вміст змінної **motion\_sensor**. Якщо він дорівнює HIGH (є рух), рядки 7 - 14 виконуються.

Рядки 7 і 8 друкують повідомлення на консолі Python.

Рядки 9 та 10 надсилають цифрове значення 1 на цифрові порти 1 і 2. Ці порти підключаються до PT-Coffee Maker та PT-Light, відповідно. Це вмикає два пристрої.

Рядок 11 відображає очікування 6000 мілісекунд (6 секунд).

Рядок 12 друкує інше повідомлення на консолі Python.

Рядки 13 та 14 надсилають цифровий сигнал 0 на цифрові порти 1 і 2. Ці порти підключаються відповідно до PT-Coffee Maker та PT-Light. Це призведе до вимкнення двох пристроїв. Рядок 15 відображає очікування 500 мілісекунд (0,5 секунди).

Для перевірки коду потрібно натиснути клавішу ALT, переміщаючи курсор миші над датчиком руху. Це означає рух, який буде захоплений датчиком PT-Motion. Після виявлення руху високочастотний цифровий сигнал надсилається до SBC, який, у свою чергу, вмикає кавоварку та світло. Повідомлення також з'являються в консолі SBC.

### **Варіанти завдань для виконання лабораторної роботи**

У середовищі Cisco Packet Tracer потрібно створити модель домашньої мережі згідно сценарію та запрограмувати пристрої для їх функціонування у мережі та взаємодії з мережевими пристроями (рис. 1.18). Усі домашні пристрої та сервери IoT підключені до домашньої бездротової мережі. Кожен пристрій



підключений до бездротового маршрутизатора. Налаштувати DHCP, WLAN. Усі бездротові пристрої необхідні для використання ідентифікаторів SSID, пароля та налаштувань DHCP, крім локального сервера, що використовує 10-клас статичних IP. Налаштувати DNS сервер для перекладу URL-адреси домашньої сторінки IoT у власну IP-адресу сервера IoT. Виконати тестування запрограмованих пристроїв.

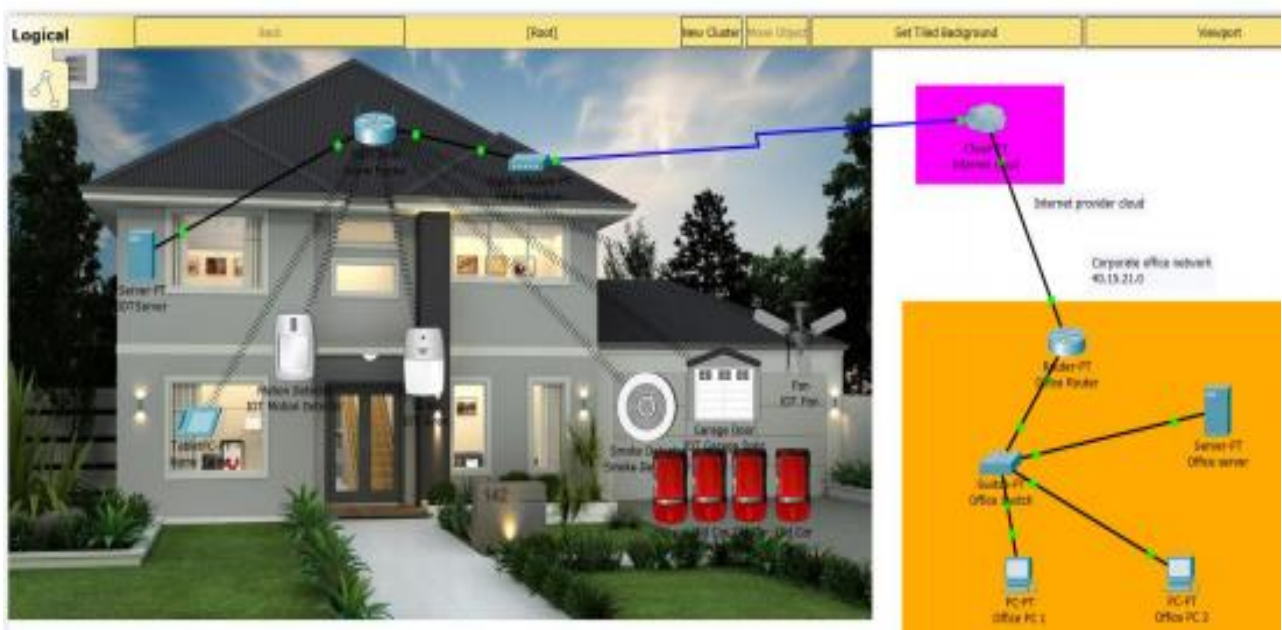
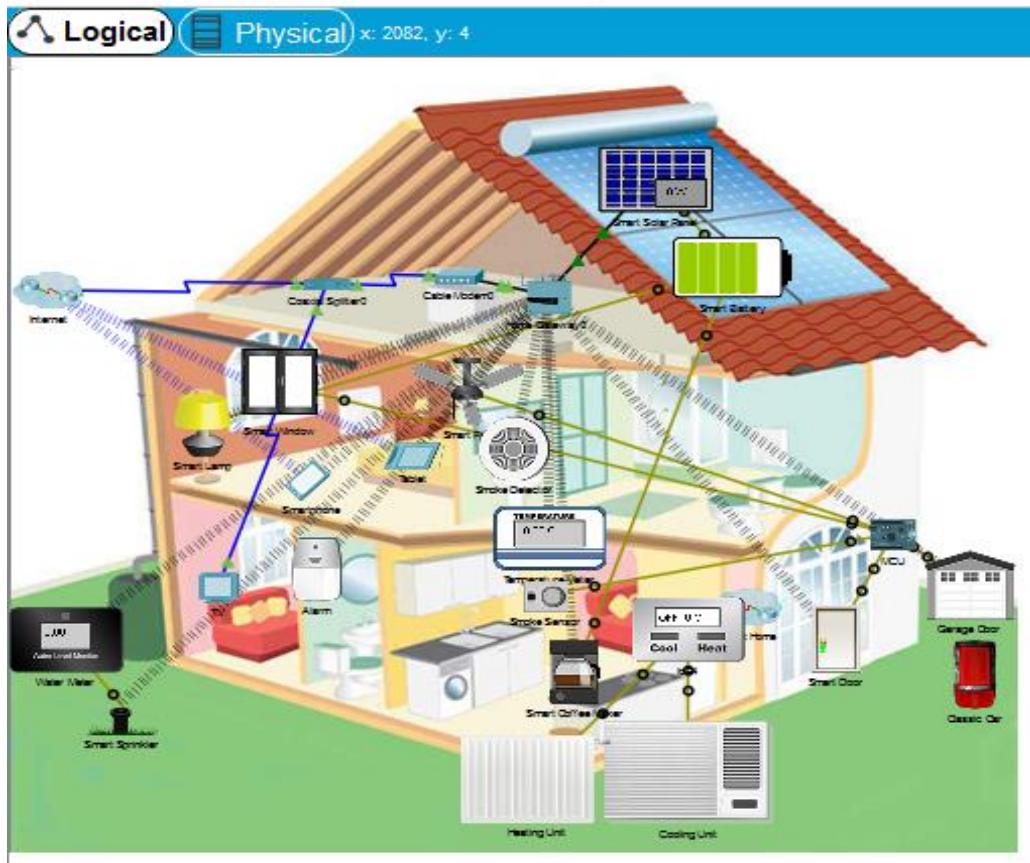


Рис. 1.18. Приклади моделювання домашньої IoT- мережі в СРТ

Варіант	Сценарій
1.	Власник будинку після підключення через браузер повинен пройти аутентифікацію, він може управляти дверима гаражу, вентиляцією, перевіряти поточний стан системи сигналізації та рівень двоокису вуглецю у гаражі. Усі пристрої і сервер IoT підключені до однієї мережі WLAN. Сервер IoT налаштований для сервісів DNS.
2.	Змоделювати відкриття дверей гаража разом з включенням світла. Датчик безпосередньо підключений до входу SBC. Коли об'єкти виявлені, датчик подає значення на вхідний контакт SBC, тоді мікроконтролер порівнює його з попередньо встановленим пороговим значенням.
3.	Використати смарт-пристрої IoT: детектор руху, сирена, двері гаража. Використати датчик руху для тимчасової активації сирени сигналізації. При виявленні датчиком руху спрацьовує сирена. Коли датчик не виявляє будь-якого руху після попередньо встановленого тайм-ауту, сирена налаштовується на вимкнення.
4.	Змоделювати включення системи клімат-контролю у приміщенні. Задати допустимий діапазон температури та включення і виключення системи, якщо температура виходить за межі заданого діапазону.
5.	Налаштувати мережу, у якій датчик налаштований на включення вентилятора, відкриття дверей гаража і увімкнення сигналізаційної сирени у випадку досягнення рівня діоксиду вуглецю вище попередньо встановленої межі.
6.	Використати смарт-пристрої IoT: детектор руху, світло, двері. Використати датчик руху для тимчасової активації джерела світла. При виявленні датчиком руху вмикається світло. Коли датчик не виявляє руху зазначений період часу, світло вимикається.

7.	<p>Встановити сонячні батареї та акумулятор. Вдень, коли сонце світить, сонячні панелі виробляють електроенергію, відбувається заряджання акумулятора. Якщо сонце збільшує вологість в будинку, запускається кондиціонер. Акумулятор зберігає електропостачання, коли кондиціонер розряджається.</p>
8.	<p>Встановити детектор руху, який запускає трансляцію веб-камери, якщо виявлено рух. Перевірити результат моделювання, натиснувши клавішу ALT на клавіатурі та виконати обробку на датчику детектора руху за допомогою миші. Веб-камера транслює зображення з домашньої сторінки IoT.</p>
9.	<p>Налаштувати інтелектуальне світло в саду. Використати датчик руху і датчик світла. Ліхтар вмикається автоматично, якщо поблизу рухається об'єкт та при цьому виявлено рівень видимого світла, нижчий заданого.</p>
10.	<p>Налаштувати детектори для безпеки будинку: детектор для диму, детектор виявлення оксиду вуглецю та встановити пожежний сповіщувач. Налаштувати приведення в дію додаткової вентиляції та відкриття дверей і вікон у разі виявлення небезпек.</p>
11.	<p>Встановити датчик вологості IoT. Якщо виявлена вологість перевищує або не досягає попередньо встановленого порогового значення, автоматично вмикається пристрій – зволожувач повітря, який регулює рівень вологості.</p>
12.	<p>Створити рішення для управління доступом до будинку за допомогою RFID-зчитувача, використовуючи декілька карток RFID і «розумні двері». Якщо авторизована RFID-картка поміщається на RFID-зчитувач, відкриваються двері, якщо ж використовується неавторизована RFID-карта, двері залишаються заблокованими.</p> <p>Для досягнення такого сценарію визначити параметри карти RFID. Логіка авторизації виконується сервері IoT, як умова встановлюється код для розблокування дверей.</p>

13.	Змоделювати систему зрошення саду, де датчик виявляє рівень вологості і, якщо показник вологості нижчий заданого діапазону, запускається система зрошування. Налаштувати час регулярного поливу 03.00 до 06.00 та з 20.00 до 23.00 щодня.
14.	Змоделювати датчик руху і веб-камеру, яка повинна включатися при спрацьовуванні датчика руху і вимикатися, коли рух припиняється. Створити правила включення і виключення для веб-камери, використовуючи головний шлюз через веб-браузер і вкладку Conditions (правила If – Else).
15.	Змоделювати роботу газонного розприскувача. Використати тип з'єднання Copper Straight Through FastEthernet0 Sprinkler з будь-яким вільним портом Ethernet головного шлюзу Home Gateway. Використати в Gateway / DNS Ip v4 – DHCP.
16.	Встановити в будинку датчик рівня води. Якщо датчик фіксує перевищення заданого порогу, вмикається сирена та підключається пристрій «розумний водостік».
17.	Забезпечити в будинку систему реагування на вогонь з метою запобігання пожежі. При виявленні диму чи вогню спрацьовує система гасіння та сирена.
18.	Налаштувати детектор вимірювання швидкості вітру для бездротової мережі з домашнім шлюзом. Налаштувати закриття вікна в будинку при перевищенні значення рівня швидкості вітру вище значень заданого діапазону.
19.	Встановити датчик руху при відкритті дверей будинку і веб-камеру так, щоб веб-камера включалася при спрацьовуванні датчика руху і вимикалася, коли рух припинявся.
20.	Встановити перехід на енергозберігаюче світло, якщо знижується рівень звуку у приміщенні. При нульовому рівні шуму заданого періоду часу світло вимикається.

21.	Встановити датчик руху в приміщенні. При виявленні руху при рівні освітлення, нижчого заданому, вмикається світло та кавоварка, якщо рух відбувається між 6-00 та 7-00 годинами ранку.
22.	Забезпечити увімкнення звукового динаміку, кавоварки та вимкнення системи опалення будинку в 7-00 при виявленні руху детектором руху.
23.	Забезпечити систему відео-спостереження приміщення у період з 23-00 до 6-00. У разі виявлення руху детектором руху вмикати світло та сигналізацію.
24.	Змодельовати систему обігріву приміщення, де датчик виявляє рівень тепла і, якщо показник рівня тепла нижче заданого, запускається система обігріву. Налаштувати час регулярного обігріву з 18.00 до 23.00 щодня.
25.	Встановити в будинку веб-камеру, яка починає трансляцію, коли вмикається кавоварка. Якщо показник рівня світла при цьому нижче заданого значення, вмикається лампа.

### **Вимоги до оформлення звіту**

Звіт має включати:

1. Титульний аркуш.
2. Завдання на лабораторну роботу.
3. Хід роботи. Цей розділ складається з послідовного опису виконуваних кроків згідно інструкцій до лабораторної роботи.
4. Висновки.

### **Питання для самоперевірки**

1. Пояснити термін та концепцію IoT. Навести приклади.
2. Пояснити призначення RFID.
3. Який стандарт управляє доступом для організації енергоефективних персональних мереж?
4. Які технології використовуються для обміну інформацією між пристроями?

5. Які можливості середовища Cisco Packet Tracer для створення та налаштування IoT рішень ви знаєте?
6. Яке призначення мікроконтролера?
7. Яким чином встановлюється зв'язок між датчиками та виконавчими елементами?
8. Як відбувається програмування MCU?
9. Як відбувається програмування SBC?
10. Які зміни слід внести в SBC (прикладу 2), якщо детектор руху був перенесений на порт D5?

### **Рекомендована література**

1. Introduction to IoT (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.
2. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli. Fog Computing and Its Role in the Internet of Things // Електронний ресурс. Режим доступу: <http://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf>
3. Leading the IoT // Електронний ресурс. Режим доступу: [https://www.gartner.com/imagesrv/books/iot/iotEbook\\_digital.pdf](https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf)
4. Changing the programming paradigm for the embedded in the IoT domain // Електронний ресурс. Режим доступу: <http://ieeexplore.ieee.org/document/7389059/?arnumber=7389059>
5. IoT Fundamentals Connecting Things (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.
6. Finardi A. IoT Simulations with Cisco Packet Tracer // Електронний ресурс. Режим доступу: <https://www.theseus.fi/bitstream/handle/10024/150158/Andrea%20Finardi%20%20Master%20of%20Engineering%20%20Information%20technology.pdf>
7. Things and Components available in Packet Tracer 7.2 // Електронний ресурс. Режим доступу: <https://www.packettracernetwork.com/internet-of-things/pt7-iot-devices-configuration.html>

## ЛАБОРАТОРНА РОБОТА № 2.

### ЗАХИСТ ХМАРНИХ СЕРВІСІВ ІНТЕРНЕТУ РЕЧЕЙ

**Мета роботи:** навчитися здійснювати основні заходи безпеки для IoT-рішень з хмарними сервісами у середовищі моделювання Cisco Packet Tracer.

#### Завдання до лабораторної роботи

- У середовищі моделювання Cisco Packet Tracer зареєструвати чотири пристрої IoT на складі компанії ABC, детектор руху, спрямоване світло та веб-камеру. Додати умови в реєстраційному сервері, щоб при активації детектору руху вмикались націлене спрямоване світло та веб-камера.
- Налаштувати на маршрутизаторі складу аутентифікацію для безпечного входу на консоль та для віддаленого доступу.
- Налаштувати ACL, щоб обмежити мережевий трафік між реєстраційним сервером та складом компанії ABC.
- Налаштувати веб-сервер у мережі постачальника послуг хмарної мережі, щоб забезпечити безпеку передачі даних.

Припустимо, що компанія ABC розробляє рішення IoT у своєму основному складі. Необхідно розгорнути фізичні пристрої безпеки на складі для того, щоб коли склад буде закритий, ці пристрої контролювали двері та вікна. Коли буде виявлено вторгнення, світло увімкнеться, а веб-камери почнуть відео-трансляцію.

У середовищі Packet Tracer необхідно побудувати мережу заданої топології (рис.2.1) та виконати відповідні налаштування. Додатково потрібно виконати налаштування IoT-системи з Лабораторної роботи №1 згідно варіанту завдання.

#### Теоретичні відомості

Інтернет речей може забезпечити новий рівень комфорту у повсякденному житті звичайних користувачів, проте, якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання за допомогою надійного криптографічного алгоритму, замість користі вони можуть принести шкоду, надавши кіберзлочинцям лазівку для підриву інформаційної безпеки. Оскільки пристрої із вбудованими комп'ютерами зберігають дуже багато

інформації про свого власника, зокрема, можуть знати його точне місцезнаходження, доступ до такої інформації може допомогти зловмисникам здійснити злочин. Відсутність на даний час стандартів для захисту таких автономних мереж сповільнює впровадження Інтернету речей у повсякденне життя. Безпека є важливим аспектом для систем IoT з хмарними сервісами.

### Вказівки до виконання лабораторної роботи

Виконаємо реєстрацію IoT пристроїв на сервері реєстрації.

Додайте користувача до реєстраційного сервера [www.registrar1.pka](http://www.registrar1.pka) з надійним паролем. Використовуйте ПК в офісі WH. Вкладка "Робочий стіл", відкрийте веб-браузер, введіть [www.registrar1.pka](http://www.registrar1.pka) та виберіть "Перейти". Відображається вікно входу реєстраційного сервера.

Клацніть Зареєструватися зараз і створіть свій власний обліковий запис із надійним паролем (переконайтеся, що пароль має принаймні 8 символів).

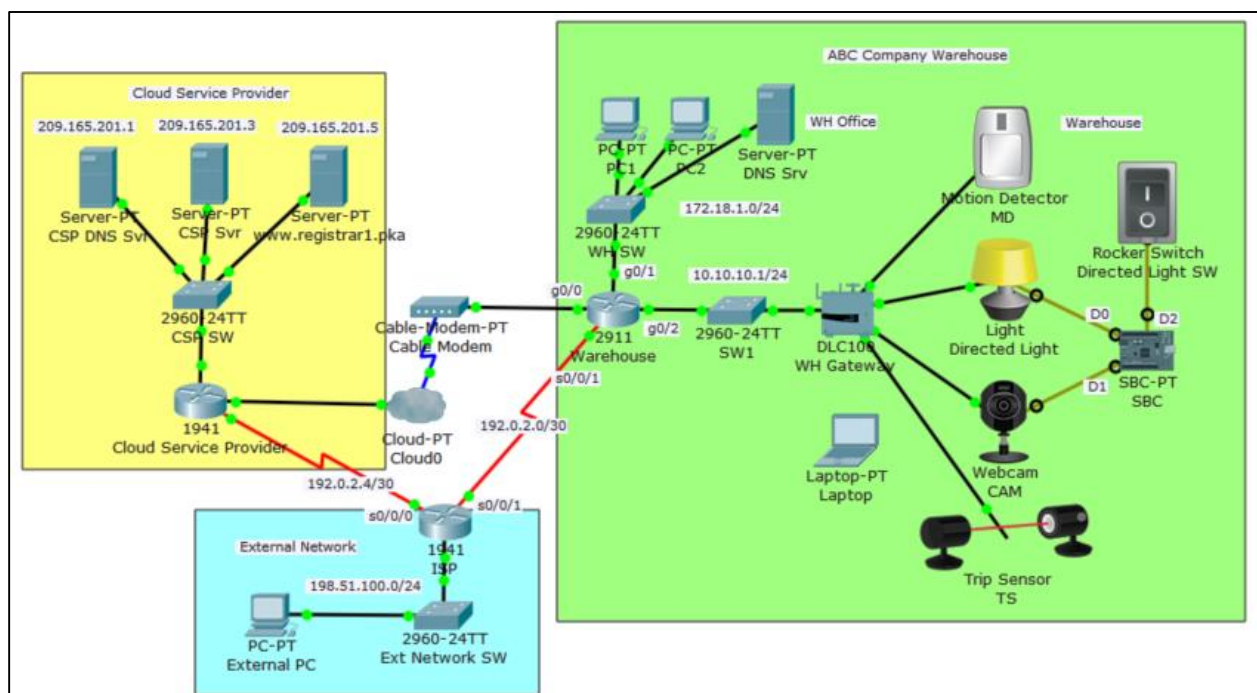


Рис.2.1. Топологія системи IoT компанії ABC

Зареєструйте пристрої IoT на сервер реєстрації. Натисніть «Детектор руху», на вкладці «Налаштування» виберіть «Віддалений сервер», введіть [www.registrar1.pka](http://www.registrar1.pka) як адресу сервера та ім'я користувача та пароль, які ви щойно створили. Клацніть кнопку Connect і почекайте завершення з'єднання. Перевірте,



чи з'являється детектор руху на сервері реєстрації. Повторіть кроки 1 і 2, щоб зареєструвати світло, веб-камеру та датчик руху.

Додайте умови на реєстраційному сервері, так щоб при активації будь-якого детектору руху кероване світло та веб-камера вмикались. Увійдіть на сервер реєстрації за допомогою імені користувача / пароля, який ви створили. Натисніть "Умови" та додайте три наступні умови.

Назвіть LightsOn1, якщо стан MD ON включений, встановіть статус "Referred Light" на "On" і встановіть CAM статус On на true.

Назвіть LightsOn2, якщо статус TS On («Увімкнено») є правильним, встановіть статус «Referred Light» на «On» і встановіть CAM статус On на true.

Назвіть LightsOff, якщо обидві значення MD статус On є помилковими, а статус TS on Is false, а потім встановіть статус Directed Light на Off і встановіть CAM статус On на false.

Перевірте умови. Тримайте клавішу ALT і посуньте курсор миші над детектором руху. Перевірте, чи увімкнено режими прямого світла та веб-камери. Наведіть курсор миші та зачекайте кілька секунд. Перевірте, чи вимкнено напрямом світла та веб-камери?

Налаштуйте надійну аутентифікацію для бездротового з'єднання на шлюзовому пристрої WH. Клацніть WH Gateway. На вкладці Конфігурація, параметр «Бездротовий зв'язок», встановіть SSID на WhGateway1, встановіть аутентифікацію на WPA2-PSK з паролем IoTWh001. Залиште тип шифрування AES. Натисніть на ноутбук. На вкладці «Конфігурація» опції Wireless0 встановіть SSID на WhGateway1, встановіть аутентифікацію на WPA2-PSK з паролем IoTWh001. Залиште тип шифрування як AES. Перевірте, чи успішно підключено ноутбук до шлюзу WH.

На маршрутизаторі складу налаштуйте маршрутизатор для відображення попереджувального повідомлення про незаконний доступ. Введіть зашифрований пароль для входу в режим Eexec. Налаштуйте локальний обліковий запис користувача для командного рядка та віддаленого доступу. Клацніть на маршрутизатор складу, потім клацніть на вкладці CLI та введіть команди:

```
Warehouse> enable
```

```
Warehouse# config terminal
```

```
Warehouse(config)# banner login %Login with valid password%
```

```
Warehouse(config)# banner motd %Authorized Access Only! Unauthorized access is  
subject to Federal Prosecution.%
```

```
Warehouse(config)# exit
```

```
Warehouse#
```

Встановіть пароль для безпечного режиму EXEC:

```
Warehouse(config)# enable secret AbcWh001
```

```
Warehouse(config)# exit
```

Встановіть локальне ім'я користувача для доступу до консолі та ліній  
VTY:

```
Warehouse# configure terminal
```

```
Warehouse(config)# username WhAdmin secret AbcLine001
```

```
Warehouse(config)# line console 0
```

```
Warehouse(config-line)# login local
```

```
Warehouse(config-line)# exit
```

```
Warehouse(config)# line vty 0 4
```

```
Warehouse(config-line)# login local
```

```
Warehouse(config-line)# end
```

```
Warehouse#
```

Виконаємо налаштування списків доступу для обмеження трафіку між пристроями IoT компанії ABC та мережею постачальників Cloud Services. На маршрутизаторі складу налаштуйте та застосуйте список доступу 10 для дозволу трафіку лише з DNS-сервера та реєстраційного сервера та входу в мережу пристроїв IoT компанії ABC:

```
Warehouse# configure terminal
```

```
Warehouse(config)# access-list 10 permit host 172.18.1.5
```

```
Warehouse(config)# access-list 10 permit host 209.165.201.5
```

```
Warehouse(config)# interface g0/2
```

```
Warehouse(config-if)# ip access-group 10 out
```

```
Warehouse(config-if)# end
```

На маршрутизаторі постачальника хмарних сервісів налаштуйте та застосуйте список доступу 110, щоб дозволити трафік лише з мережі пристроїв IoT складу компанії ABC для доступу до реєстраційного сервера:

```
CSP# configure terminal
CSP(config)# access-list 110 permit ip host 209.165.200.226 host 209.165.201.5
CSP(config)# access-list 110 deny ip any host 209.165.201.5
CSP(config)# access-list 110 permit ip any any
CSP(config)# interface g0/0
CSP(config-if)# ip access-group 110 out
CSP(config-if)# end
```

Налаштуємо безпечну веб-комунікацію на веб-сервері в мережі Cloud Provider Service. Компанія ABC використовує веб-сервер у постачальника хмарних послуг для частини своєї операції. Налаштуйте веб-сервер у мережі постачальника хмарних сервісів, до якого можна отримати доступ лише через HTTPS. Клацніть CSP Svr, потім клацніть вкладку Services. Клацніть HTTP на лівій панелі. Переконайтеся, що HTTP вимкнено, а HTTPS увімкнено.

Виконайте тестування. З ноутбука у складській мережі зверніться на сервер реєстрації. Увімкніть детектор руху та спостерігайте за дією спрямованого світла та веб-камери. З PC1 або PC2 відкрийте веб-браузер. Перевірте, чи може він отримати доступ до реєстраційного сервера. З PC1 або PC2 відкрийте веб-браузер. Перевірте, чи може він отримати доступ до веб-сервера 209.165.201.3 через HTTP. З PC1 або PC2 відкрийте веб-браузер. Перевірте, чи може він отримати доступ до веб-сервера 209.165.201.3 через HTTPS.

### **Вимоги до оформлення звіту**

Звіт має включати:

1. Титульний аркуш.
2. Завдання на лабораторну роботу.
3. Хід роботи. Цей розділ складається з послідовного опису виконуваних кроків згідно інструкцій до лабораторної роботи.
4. Висновки.

### **Питання для самоперевірки**

1. Які заходи безпеки IoT рішень ви знаєте?
2. Які заходи безпеки слід враховувати при розгортанні рішень IoT з використанням хмарних обчислень?
3. Як відбувається налаштування надійної аутентифікації на мережевих пристроях?
4. Як відбувається налаштування списків доступу для обмеження трафіку між пристроями IoT компанії та мережею постачальників?
5. Яким чином здійснюється налаштування безпечної веб-комунікації на веб-сервері у мережі ?

### **Рекомендована література**

1. Top cloud security controls you should be using // Електронний ресурс. Режим доступу: <https://www.csoonline.com/article/3208905/cloud-security/top-cloud-security-controls-you-should-be-using.html>
2. Introduction to IoT (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.
3. IoT Fundamentals Connecting Things (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.

## **ЛАБОРАТОРНА РОБОТА № 3.**

### **ДОСЛІДЖЕННЯ ТА АНАЛІЗ ВІДКРИТИХ ДАНИХ МЕРЕЖІ ІНТЕРНЕТ**

**Мета роботи:** проаналізувати веб-сайти, які надають відкриті дані, здійснити аналіз даних, використовуючи технології Pandas і Python.

#### **Завдання до лабораторної роботи**

У цій лабораторній роботі необхідно дослідити можливості візуалізації даних, доступні на веб-сайтах для того, щоб створити статистику для подальшої обробки даних. Необхідно отримати статистику швидкості інтернету і зберегти дані в реальному часі у файлі CSV, завантажити збережені дані з файлу CSV в структуру даних, використати технології Python, Pandas у DataFrame для вивчення та обробки даних.

#### **Теоретичні відомості**

Технології Pandas і Python пропонують ряд корисних операцій для роботи з даними, зокрема, з датою і часом.

Багато організацій, таких як уряди, університети та неприбуткові організації, публікують свої відкриті дані. Наприклад, веб-сайт <http://www.data.gov> є сховищем відкритих даних уряду США. На цьому веб-сайті можна знайти дані та ресурси для досліджень, розроблення програмного забезпечення та інших проектів, пов'язаних із даними. З цього веб-сайту також можна перейти на оригінальні веб-сайти, на яких надано опубліковану інформацію.

#### **Вказівки до виконання лабораторної роботи**

Для того, щоб відфільтрувати дані сайту <http://www.data.gov> за різними категоріями, наприклад, вибрати лише дані, надані міською владою, потрібно натиснути кнопку "Дані" / "Місцевий уряд" (у верхній частині екрану), прокрутити список до типів організацій на лівій панелі, натиснути «Міська влада», тут можна переглядати статистику злочинів за вказаний період. Потім можна завантажити цей набір даних, перейшовши до розділу «Завантаження та ресурси».

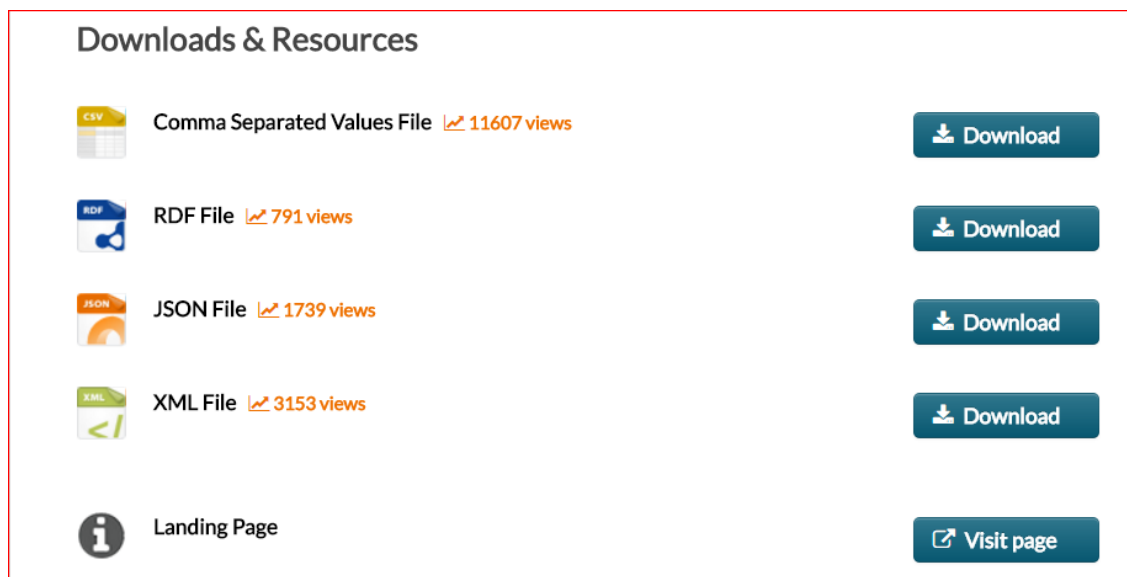


Рис. 3.1. Доступні формати файлів для завантаження

На цьому сайті також є можливість перегляду даних на карті. Для цього потрібно обрати потрібне місто у верхній частині сторінки, щоб перейти до списку наборів даних, доступних на сайті порталу даних міста. У розділі «Типи перегляду» на панелі ліворуч необхідно натиснути «Карти», щоб відобразити список карт. При натисненні на наборі даних "Злочини - Карта" відобразиться інтерактивна карта міста з місцями злочину за рік.

Подібні веб-сайти для інших країн можна знайти за допомогою пошуку в мережі Інтернет. Наприклад:

Австралія: <http://data.gov.au/>

У.К.: <http://data.gov.uk>

Канада: <http://data.gc.ca>

Візуалізація даних може допомогти "побачити" тенденції, які можуть бути пропущені. Розглянемо веб-сайти, які надають візуалізацію даних.

Наприклад, фонд Garminder забезпечує збір статистичних даних з різних джерел. Перейдіть до [www.garminder.org](http://www.garminder.org). Натисніть "Відео", щоб переглянути презентації, які оживляють статистичні дані та роблять дані більш релевантними для вас. Оберіть відео. Для того, щоб взаємодіяти з бульбашковими діаграмами, які представлені у відео, натисніть Garminder World. Натисніть кнопку "Грати". Спостерігайте за змінами розсіювання на тлі між тривалістю життя (роками) та доходом на людину. Кожна з цих бульбашок являє собою прогрес країни. При

цьому можна спостерігати деякі закономірності. Наприклад, для осі  $Y$  Очікувана тривалість життя (років) для населення у віці 60+ років, обох статей (%), тому на графіку порівнюється відсоток населення старше 60 років з доходом на людину.

Виберіть кілька країн, щоб виділити їх шлях за певний період часу. Наприклад, Канаду, Німеччину та Японію. Наприклад, згідно прогнозу, Японія матиме найвищий відсоток їх населення у віці 60+ років у 2030 році відповідно до візуалізації даних з <http://knoema.com/awppbnd/how-well-ageing-population-is-faring>.

Натисніть кнопку «Відтворити», щоб спостерігати за країнами та виділіть основні закономірності. Джерела даних, які використовуються в цих діаграмах, можна завантажити, натиснувши символ сітки біля осі  $X$  або  $Y$  на графіку. Натисніть символ сітки, щоб завантажити та переглянути вихідні дані.

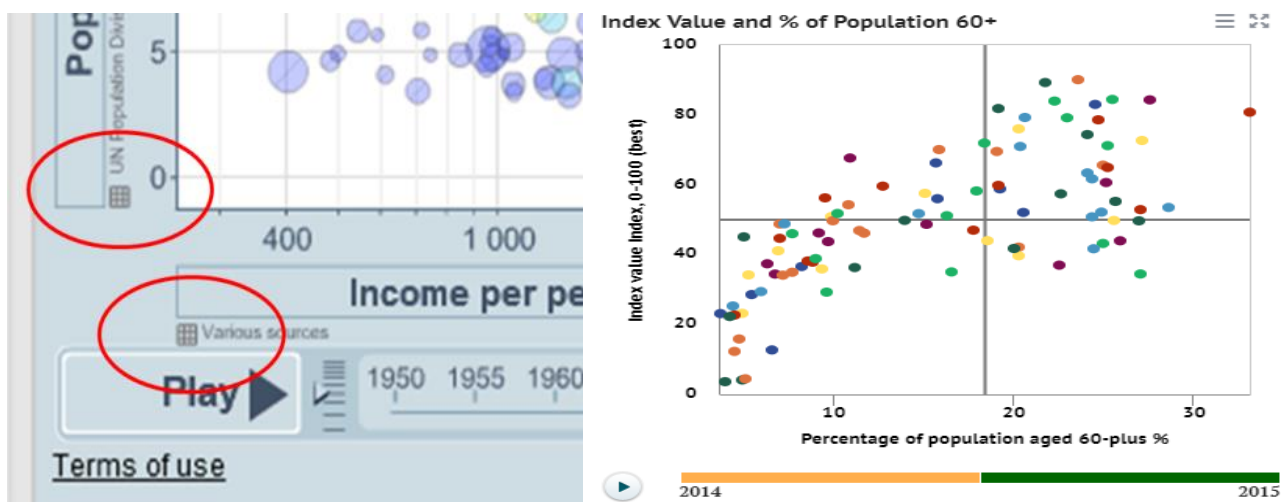


Рис. 3.2. Засоби візуалізації даних knoema.com

Розглянемо обмеження електронних таблиць у аналізі даних, використовуючи відкритий набір даних із міста Сіетл. Завантажте Data Set Weather Weather Stations із <https://data.seattle.gov>.

Перейдіть до <https://data.seattle.gov/Transportation/Road-Weather-Information-Stations/egc4-d24i?> та натисніть **View Data**.

Даний набір містить температуру дорожнього покриття і повітря від датчиків, вбудованих на мости та поверхні вулиці в межах міста Сіетл. Дані оновлюються кожні п'ятнадцять хвилин (рис. 3.3).

	StationName	StationLocation	DateTime	RecordId
1	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:42:00 PM	67
2	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:43:00 PM	67
3	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:44:00 PM	67
4	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:45:00 PM	67
5	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:46:00 PM	67
6	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:47:00 PM	67
7	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:48:00 PM	67
8	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:49:00 PM	67
9	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:50:00 PM	67
10	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:51:00 PM	67
11	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:52:00 PM	67
12	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:53:00 PM	67
13	35thAveSW_SWMyrtleSt	(47.53918°, -122.37658°)	03/03/2014 12:54:00 PM	67

Рис. 3.3. Дані Data Set Weather Weather Stations

Необхідно перевірити, скільки полів (стовпчиків) є у наборі даних, скільки записів (рядків) та які імена полів. Натисніть кнопку Експортувати. Натисніть CSV, щоб завантажити цей набір даних у форматі CSV (Comma-Separated Values). Завантаження займе декілька хвилин. Якщо не вдалося повністю завантажити файл Road\_Weather\_Information\_Stations.csv, знайдіть Road\_Weather\_Information\_Stations.csv.part у вашій файловій системі. У папці "Завантаження" змініть ім'я файлу на Road\_Weather\_Information\_Stations.csv.

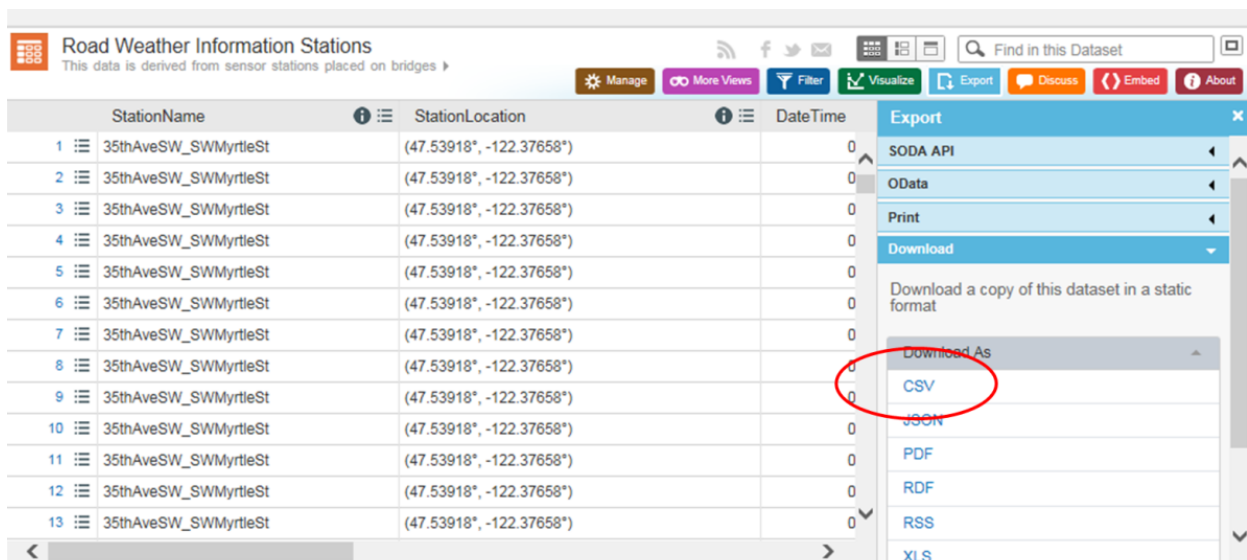


Рис. 3.4. Завантаження набору даних у форматі CSV

Установіть SpeedTest та імпортуйте бібліотеки Python. Speedtest-cli – це скрипт на мові Python, який вимірює завантаження і швидкість завантаження



підключення до Інтернету. Для отримання додаткової інформації про SpeedTest, перейдіть за посиланням <https://github.com/sivel/speedtest-cli>,

а) Установіть speedtest-cli.,

```
!pip install speedtest-cli
```

Це дозволяє редактору Jupyter підключитися до веб-сайту і зберегти дані.

б) Імпорт необхідних бібліотек Python.

```
# Python бібліотеки для керування датою і часом даних
```

```
import datetime
```

```
#бібліотека Python для читання і запису файлів CSV
```

```
import csv
```

```
# бібліотека Python для виконання команд із записної книжки
```

```
# Якщо ви хочете дізнатися більше про це, перегляньте ресурс:
```

```
# http://www.pythonforbeginners.com/os/subprocess-for-system-administrators
```

```
import subprocess
```

Виконаємо генерування статистичних даних швидкості мережі Інтернет. Найважливішим кроком в отриманні даних для більшості аналітичних даних додатків є зв'язування мітки часу для вимірювань.

а) Для того, щоб згенерувати мітку часу, використовуйте `datetime.now` функцію з `datetime` пакету:

```
date_time = datetime.datetime.now()
```

```
print(date_time, type(date_time))
```

б) Примірник класу `datetime` не може бути записаний безпосередньо у текстовому вигляді. Функція `strftime` перетворює інформацію дати в рядок. Аргументи цієї функції визначають формат. Опис цих параметрів можна знайти в документації з функції `strftime` за посиланням <https://docs.python.org/2/library/time.html>,

```
date_time.strftime('%a, %d %b %Y %H:%M:%S')
```

Після прочитання документації з `strftime` треба згенерувати мітку час і розпарсити його в рядок у форматі: YYYY-MM-DD HH: MM: SS.

Speedtest-cli команда, запустившись з терміналу, повертає рядок вимірів швидкості. Для запуску команди з цього редактора необхідно використовувати

модуль Python subprocess, який дозволяє запускати процес безпосередньо з коду редактора.

а) Виконайте тест швидкості, використовуючи speedtest-cli з Python.

Результат буде зберігатися у змінній process\_output.

```
# Цей рядок містить команди для взаємодії з speedtest.net
speedtest_cmd = "speedtest-cli --simple"
# Виконати процес
process = subprocess.Popen(speedtest_cmd.split(), stdout=subprocess.PIPE)
# Вивід команди
process_output = process.communicate()[0]
```

б) Виведіть результат процесу. Зверніть увагу на тип для process\_output змінної.

```
print(process_output, type(process_output))
```

в) Результат тесту швидкості аналізується, і відмітка часу додається до результатів.

```
# Зберігайте час, в якому виконувалася SpeedTest
date_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
process_output = process_output.split()
process_output.append(date_time)
print(process_output, type(process_output))
```

г) Функція SpeedTest () створюється для повернення результатів від команди speedtest-cli.

```
# Функція для перетворення додатків тесту швидкості
```

```
захист SpeedTest():
```

```
# Потрібно зберегти час, при якому виконувалася SpeedTest
date_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
# Це рядок містить те, що ми будемо писати в командному рядку
# з інтерфейсу з speedtest.net
speedtest_cmd = "speedtest-cli --simple"
# Тепер виконаємо процес:
process = subprocess.Popen(speedtest_cmd.split(), stdout=subprocess.PIPE)
```

```

process_output = process.communicate()[0]
process_output = process_output.split()
# додамо дату і час
process_output.append(date_time)
return process_output

```

Значення, розділені комами (CSV), є найбільш поширеним форматом імпорту та експорту для електронних таблиць і баз даних. Щоб дізнатися більш детальну інформацію про роботу з CSV в Python, перейдіть за посиланням <https://docs.python.org/2/library/csv.html>,

а) Створіть файл з ім'ям test.txt в каталог / TMP і напишіть «test\_msg» у файлі.

```

with open("/tmp/test.txt",'w') as f:
    f.write('test_msg')

```

б) За допомогою команди Linux cat перевірити створення файлу.

```
!cat /tmp/test.txt
```

в) перевірити, що файл був успішно відкритий:

```

with open("/tmp/test.txt",'r') as f:
    str = f.read()
print(str)

```

г) Значення with, особливо в поєднанні try, описане у ресурсі <http://effbot.org/zone/python-with-statement.htm>,

Для запису в файл CSV, необхідно створити об'єкт csv.writer.

```
# Функція збереження даних в форматі CSV
```

```
def save_to_csv(data, filename):
```

```
    try:
```

```
        # Якщо файл існує, додайте новий рядок до нього, з
```

```
        #results поточного експерименту
```

```
    with open(filename + '.csv', 'a') as f:
```

```
        # Створення об'єкта CSV
```

```
        wr = csv.writer(f)
```

```
        # Зберегти (запис) у файл
```

```
wr.writerow(data)
```

**except:**

```
# Якщо він не існує, створіть файл
```

```
    with open(filename + '.csv', 'w') as f:
```

```
        # Це схоже на додавання нових рядків у файл
```

```
        # Створення об'єкта CSV
```

Напишіть функцію, щоб відкрити файл CSV і роздрукувати його вміст на екран. Можна знайти приклад в ресурсі <https://docs.python.org/2/library/csv.html>

```
def print_from_csv(filename):
```

```
    with open(filename + '.csv', 'r') as f:
```

```
        re = csv.read(f)
```

```
        # 1. Цикл по рядках
```

```
        # 2. Друк
```

Усі функції, необхідні для збору даних про швидкість Інтернету, виконані.

Виконаємо запуск SpeedTest і збереження даних.

а) Напишіть цикл for, який викликає speedTest 5 раз, друкує висновок тестів і зберігає дані у файлі CSV.

```
for i in xrange(5):
```

```
    speedtest_output = speedtest()
```

```
    # Вивести номер тесту
```

```
    print 'Test number {}'.format(i)
```

```
    # Вивести результат (Необхідна змінна speedtest_output)
```

```
    save_to_csv (speedtest_output, '/ TMP / rpi_data_test')
```

б) Відобразіть файл для перевірки того, що дані були збережені правильно.

```
print_from_csv('/tmp/rpi_data_test')
```

Якщо потрібно більше наборів даних, SpeedTest може працювати у фоновому режимі для збору більшої кількості зразків.

Приклад запуску коду 100 разів:

```
for i in xrange(100):
```

```
    speedtest_output = speedtest()
```

```
    print 'Test number: {}'.format(i)
```

```
print speedtest_output
save_to_csv(speedtest_output, '/tmp/rpi_data')
```

Бібліотека Python pandas дуже корисна для роботи зі структурованими даними. Повну документацію можна знайти за посиланням: <http://pandas.pydata.org/pandas-docs/version/0.14.1/>

Імпорт Pandas та інших бібліотек використовується для виконання завдань:

```
import datetime
```

```
import csv
```

```
import pandas as pd
```

```
# NumPy бібліотека, яка додає підтримку для великих багатовимірних масивів і матриць
```

```
# Поряд з високим рівнем математичних функцій для роботи на цих масивах
```

```
import numpy as np
```

Pandas DataFrame є 2-мірною структурою даних різних типів. Бібліотечна функція pandas read\_csv автоматично перетворює CSV в DataFrame об'єкт. Переглянути документацію можна тут: [http://pandas.pydata.org/pandas-docs/version/0.14.1/generated/pandas.read\\_csv.html](http://pandas.pydata.org/pandas-docs/version/0.14.1/generated/pandas.read_csv.html),

Ця функція містить багато параметрів. Тільки один є обов'язковим – це шлях до файлу, тобто місце з CSV файлом. Всі інші параметри не є обов'язковими.

Необхідно імпортувати і переглянути вміст файлу CSV, rpi\_data\_long.csv. Цей файл CSV знаходиться в тому ж каталозі, що і редактор.

а) Призначення файлу rpi\_data\_long.csv змінної data\_file,

```
data_file = './Data/rpi_data_long.csv'
```

б) За допомогою команди Linux head() перегляньте перші 10 рядків файлу CSV.

```
!head -n5 ./Data/rpi_data_long.csv
```

в) За допомогою names параметра read\_csv вкажіть ім'я DataFrame.

```
column_names = [ 'Type A', 'Measure A', 'Units A',
                 'Type B', 'Measure B', 'Units B',
                 'Type C', 'Measure C', 'Units C',
                 'Datetime']
```

d) За допомогою `read_csv` функції читання з `file`, призначте `COLUMN_NAMES` як імена стовпців в `dataframe`.

**with** `open(data_file, 'r')` **as** `f`:

```
df_redundant = pd.read_csv(f, names = column_names)
```

е) Команда `head()` відображає перші кілька рядків з `dataframe`.

```
df_redundant.head()
```

Створіть більш компактне представлення, використовуючи копію даних `df_redundant`.

Скопіюйте `df_redundant` в інший `dataframe` - `df_compact` за допомогою `copy()`,

```
df_compact = df_redundant.copy()
```

б) Перейменуйте стовпці:

Measure A -> Ping (мс)

Measure B -> Завантажити (Мбіт / с)

Measure C -> Вивантажити (Мбіт / с)

```
df_compact.rename(columns={'Measure A': 'Ping (ms)',
```

```
'Measure B': 'Download (Mbit/s)',
```

```
'Measure C': 'Upload (Mbit/s)'}, inplace=True)
```

```
df_compact.head(3)
```

в) Оскільки типи і стовпці більше не потрібні, ці стовпці можуть бути відкинуті.

```
df_compact.drop(['Type A', 'Type B', 'Type C',
```

```
'Units A', 'Units B', 'Units C'], axis=1, inplace=True)
```

```
df_compact.head()
```

Збережіть Pandas `dataframe` `df_compact` як CSV файл `rpi_data_compact.csv`,

```
df_compact.to_csv('./Data/rpi_data_compact.csv')
```

## **Вимоги до оформлення звіту**

Звіт має включати:

1. Титульний аркуш.
2. Індивідуальне завдання на лабораторну згідно варіанту.
3. Хід роботи. Цей розділ складається з послідовного опису виконуваних кроків згідно інструкцій до лабораторної роботи.
4. Висновки.

## **Питання до лабораторної роботи**

1. Які джерела відкритих даних ви знаєте?
2. Які обмеження електронних таблиць не дозволяють обробляти великі набори даних?
3. Які технології Python використовуються для збору та аналізу Інтернет даних?
4. Що робить SpeedTest функція ()? Що таке код для перегляду результатів з функції SpeedTest ()?
5. Який код дозволяє прочитати перші 20 рядків файлу CSV?

## **Рекомендована література**

1. Introduction to IoT (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.
2. IoT Fundamentals Big Data & Analytics (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.
3. Python data analysis library // Електронний ресурс. Режим доступу: <https://pandas.pydata.org>