**Національний технічний університет України**

**«Київський політехнчний інститут імені Ігоря Сікорського»**

**Навчально-науковий інститут атомної та теплової енергетики**

**Кафедра цифрових технологій в енергетиці**

**ЗВІТ**

**з лабораторної роботи №2**

**з дисципліни «Основи технології Інтернету речей (ІоТ)»**

Виконав:

студент групи ТР-23

Ровний Г.О.

Дата здачі 21.09.2024

КИЇВ - 2024

Тема 2. «Програмування пристроїв SBC із використанням Python»

**Мета роботи:**

ознайомлення з основами програмування одноплатного комп'ютера (SBC) мовою Python у середовищі Cisco Packet Tracer та отримання базових практичних навичок.

**Частина 1**

**Поставлене завдання:**

1. Додати та підключити необхідні елементи.
2. Запрограмувати одноплатний комп'ютер (SBC).
3. Виконати тестування.

**Результат виконання роботи**

1. Додаємо та підключаємо до робочого простору CPT наступні пристрої:

* Пристрій SBC: (розділ Components -> SBC-PT).
* Кавоварку: (розділ Home –> Appliance).
* Настільну лампу: (розділ End Devices -> Home –> Light).
* Датчик руху: (розділ Components -> Sensors -> Motion Sensor).
* За допомогою IoT Custome Cable підключимо пристрої до плати SBC.

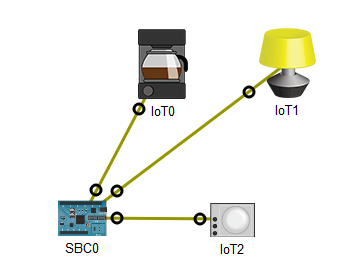


Рис.1. Топологія мережі

1. Натиснемо на SBC і перейдемо на «Programming». Зміст вкладки «Programming» поділяється на дві основні частини. На лівій панелі відображаються всі файли, що містять створені програми. Права панель відображає вміст обраного файлу. Натиснемо «New» над лівою панеллю, щоб створити новий файл main.py та оберемо шаблон Empty -> Python.

На лівій панелі обрати main.py та натиснути Open. Тепер права панель готова до введення коду програми. Використовуючи праву панель, напишемо програму, яка вмикає кавоварку та світло, коли датчик руху виявляє присутність.

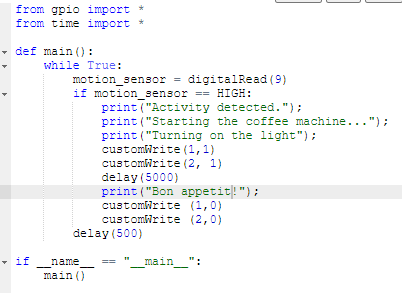


Рис.2. Код для роботи з датчиком руху

1. Для перевірки коректності коду запустимо файл main.py. Після цього натискаючи клавішу ALT і перемістивши курсор миші на датчик руху відбудеться імітація руху. В консолі отримаємо повідомлення що кавомашина та лампа спрацювали.

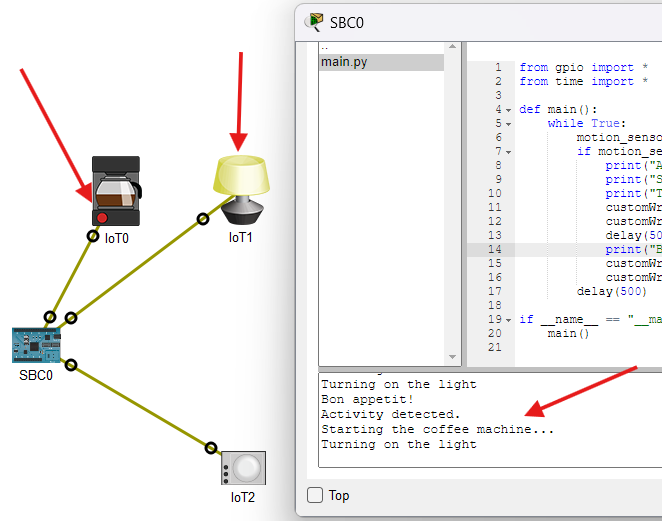


Рис.3. Тестування роботи програми

**Частина 2.**

**Поставлене завдання:**

1. Створити систему контролю доступу з оповіщенням.
2. Використати мессенджер Telegram.
3. Запрограмувати та протестувати SBC.
4. Зробити розширення топології мережі.

**Результат виконання роботи**

1. Побудуємо схему контролю доступу, що складається з таких елементів:

* Пристрій SBC (Components - SBC-PT).
* Датчик руху (Components - Sensors - Motion Sensor).
* За допомогою IoT Custom Cable підключимо пристрої до PT-SBC.

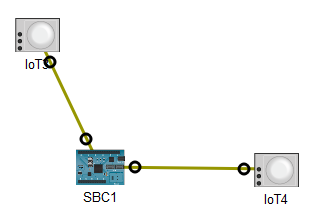


Рис.4. Типологія новоствореної мережі

1. Використаємо Telegram. Для цього кроку необхідно звернутися до офіційної сторінки створення ботів @BotFather. Створимо новий бот, що і буде отримувати інформацію про спрацювання датчиків руху. Просто перейдемо до боту та введемо команду /newbot. Після чого потрібно задати name i username для бота і ми отримаємо токен.

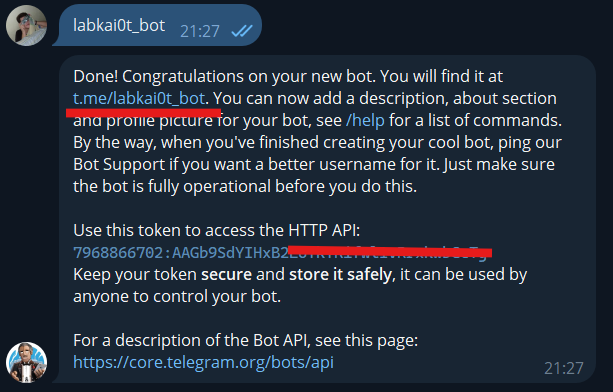


Рис.5. Повідомленням з унікальним токеном для роботи телеграм-бота

Наступним кроком необхідно відкрити чат та отримати chat\_id. Для цього в адресній стрічці браузера необхідно ввести: https://api.telegram.org/bot<YourBOTToken>/getUpdates, де <YourBOTToken> необхідно замінити на токен бота. У відповідь згенерується JSON об’єкт в якому і потрібно знайти chat\_id.

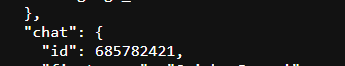


Рис.6. Отримання chat\_id

1. Програмування та тестування SBC. Відкриємо вкладинку Programing та створимо файл main.py. В створений файл main.py пишемо наступний код:

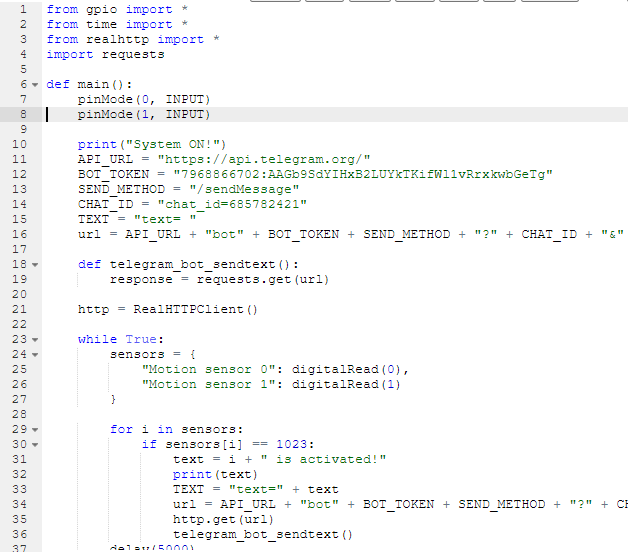


Рис.7. Код для роботи з датчиками руху та телеграм ботом

Датчик руху в CPT працює в значені від 0 до 1023. Тому значення 1023 відповідає за наявність руху. Затримка в 5с. гарантує, що після спрацювання кожного датчика буде отримано тільки одне відповідне повідомлення.

Для активації датчика затиснемо Alt і проведемо курсором миші перед датчиком. Після цього в чат боту надійде сповіщення щодо спрацювання відповідних датчиків:



Рис.8. Результат реагування датчиків руху

1. Зробимо розширення топології мережі. Додамо, наприклад, датчик пожежної сигналізації та системи гасіння пожежі, а також створимо об’єкт Fire, який буде імітувати вогонь. Маємо наступну топологію:

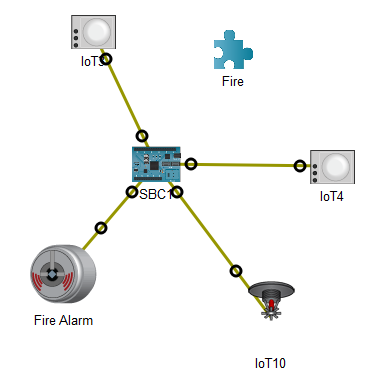


Рис.9. Удосконалена топологія мережі

Тепер нам потрібно додати наступні рядки коду для нашої програми. Потрібно підключити нові піни (pinMode) відповідно портів до яких ми підключались, а також вказати pinMode(3, OUT) для системи гасіння пожежі. Це потрібно для того, щоб ця система спрацьовувала лише при активації пожежної сигналізації у нашому випадку. Через блок коду if та функцію customWrite будемо керувати ввімкненням та вимкненням системи пожежогасіння.

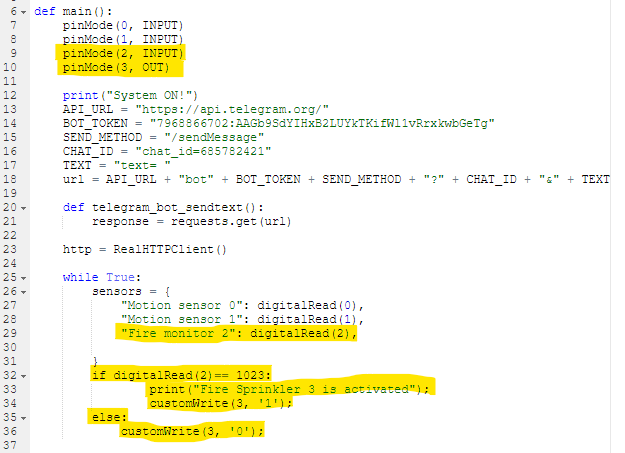


Рис.10. Оновлений програмний код

А також пропишемо наступні властивості для нашого об’єкта Fire, які необхідні для того щоб спрацювала пожежна сигналізація.

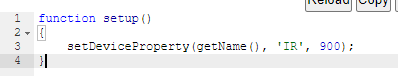


Рис.11. Задання властивосте об’єкту Fire

Запустимо код на Fire та перемістимо цей об’єкт на пожежні сигналізацію. Як результат спрацбовує пожежна сигналізація, активується система пожежогасіння, а ми отримуємо відповідне сповіщення в наш бот.

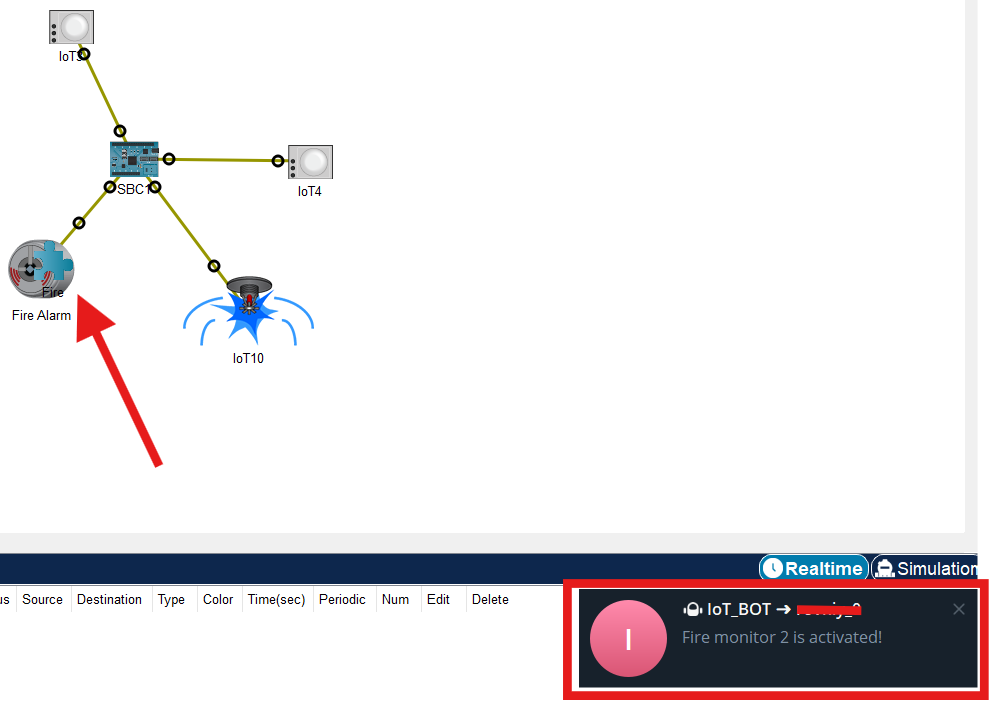


Рис.12. Результат реагування датчика на вогонь

**Висновок:**

У результаті виконання лабораторної роботи було набуто практичні навички роботи зі взаємодії IoT систем та їх програмування за допомогою мови Python а також було розроблено систему оповіщення з використанням можливостей мессенджера Telegram та IoT сенсорів. Розробка подібних систем дозволяє розширити функціональні можливості IoT пристроїв та може бути використана у розробці систем «Розумного будинку».