

Міністерство Освіти і Науки України
Київський Національний Університет імені Тараса Шевченка
Факультет Інформаційних Технологій
Кафедра Інформаційних систем та технологій

Звіт з лабораторної роботи № 1
з дисципліни « **Програмування IoT систем** »

Виконав студент 1-го курсу магістратури
групи ІРма-12
Гаврасієнко Є.О.

Київ – 2024

Мета роботи: Ознайомитися з принципами розробки та реалізації IoT-систем для автоматизації кліматичних умов, енергозбереження та управління

рівнем вологості. Навчитися підключати сенсори до мікроконтролерів, обробляти отримані дані та реалізовувати автоматичне керування виконавчими пристроями (система клімат-контролю, кондиціонер, зволожувач повітря).

Завдання для практичної роботи

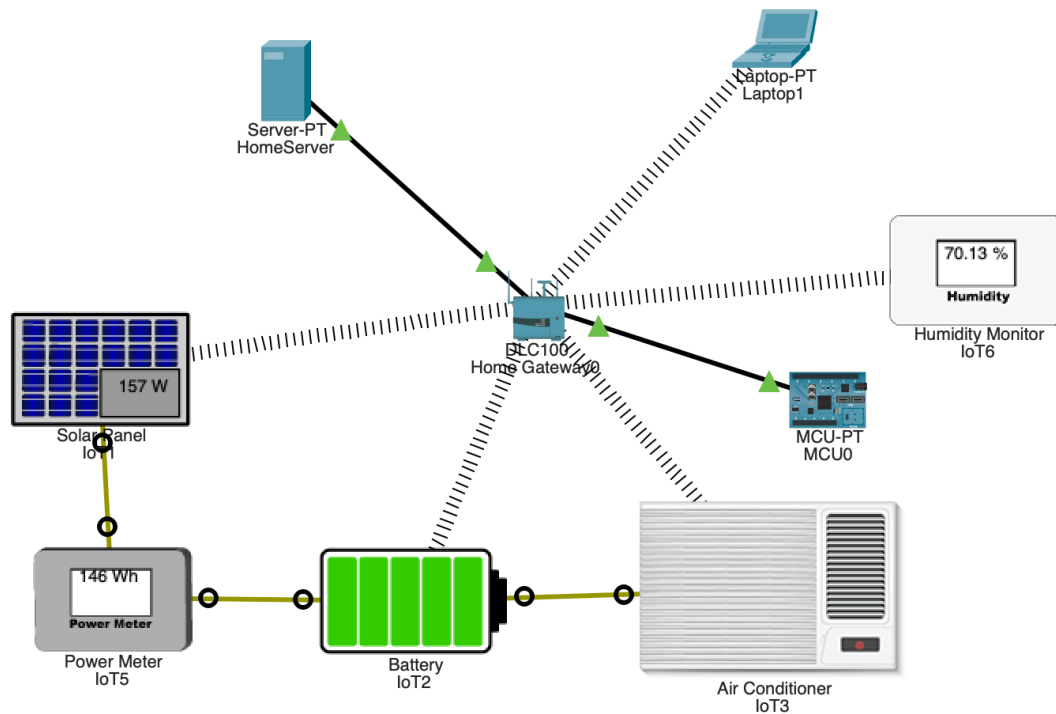
У середовищі Cisco Packet Tracer змодельовати топологію для домашньої мережі згідно сценарію за варіантом та запрограмувати пристрої для їх функціонування у мережі та взаємодії з мережевими пристроями (рис. 1.18). Усі пристрої, домашні планшети та сервери IoT підключені до домашньої бездротової мережі. 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.

Варіант 2: 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.

Хід роботи

Згідно з варіантом роботи, необхідно побудувати автоматичну систему, яка буде регулювати вологість в приміщенні шляхом використання кондиціонера, використовувати сонячну енергію для його функціонування, та заряджати

батарею, яку потім буде використовувати під-час не сонячних годин. Також всі ці девайси повинні бути об'єднанні єдину мережу, мати власні IP в ній, та користувач має мати змогу керувати цими IoT девайсами.



Структурна схема виглядає таким чином і складається з таких компонентів:

- Кондиціонер, який підключений до роутеру, та яким буде змога керувати або з серверу або з мікроконтроллера.
- Батарея, яка живить кондиціонер
- Сонячна панель
- Індикатор потужності
- Датчик вологості
- Домашній сервер
- Ноутбук користувача

IoT Server

☐ None
☐ Home Gateway
☒ Remote Server

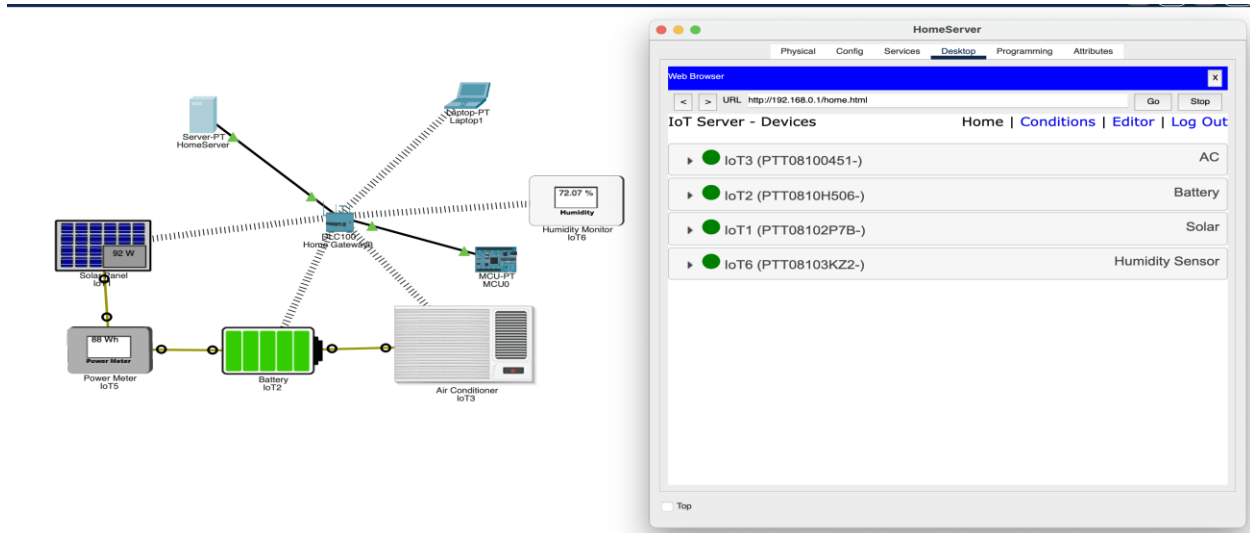
Server Address
192.168.0.1

User Name
admin

Password
admin

Refresh

Сервер буде виступати у ролі IoT серверу, для якого будуть підключенні всі наші компоненти і ми можемо мати змогу керувати їхнім станом звідси, як це виглядає тут:



Тепер залишається сконфігурувати наш мікроконтроллер для виконання основної задачі: “Якщо сонце збільшує вологість в будинку, запускається кондиціонер. Акумулятор зберігає електропостачання, коли кондиціонер розряджається.”

Код для мікро-контроллера буде на **JS**:

```
var humidityDevice = "IoT6";
var acDevice = "AirConditioner";
```

```
var lastState = "";

function getHumidity() {
    var humidity = getDeviceProperty(humidityDevice,
    "level");

    if (humidity !== null && humidity !== "") {
        return parseFloat(humidity);
    }

    Serial.println("Failed to get humidity");

    return null;
}

function setAirConditioner(state) {
    if (state !== lastState) {
        setDeviceProperty(acDevice, "power", state);
        Serial.println("Air Conditioner " + state);
        lastState = state;
    }
}

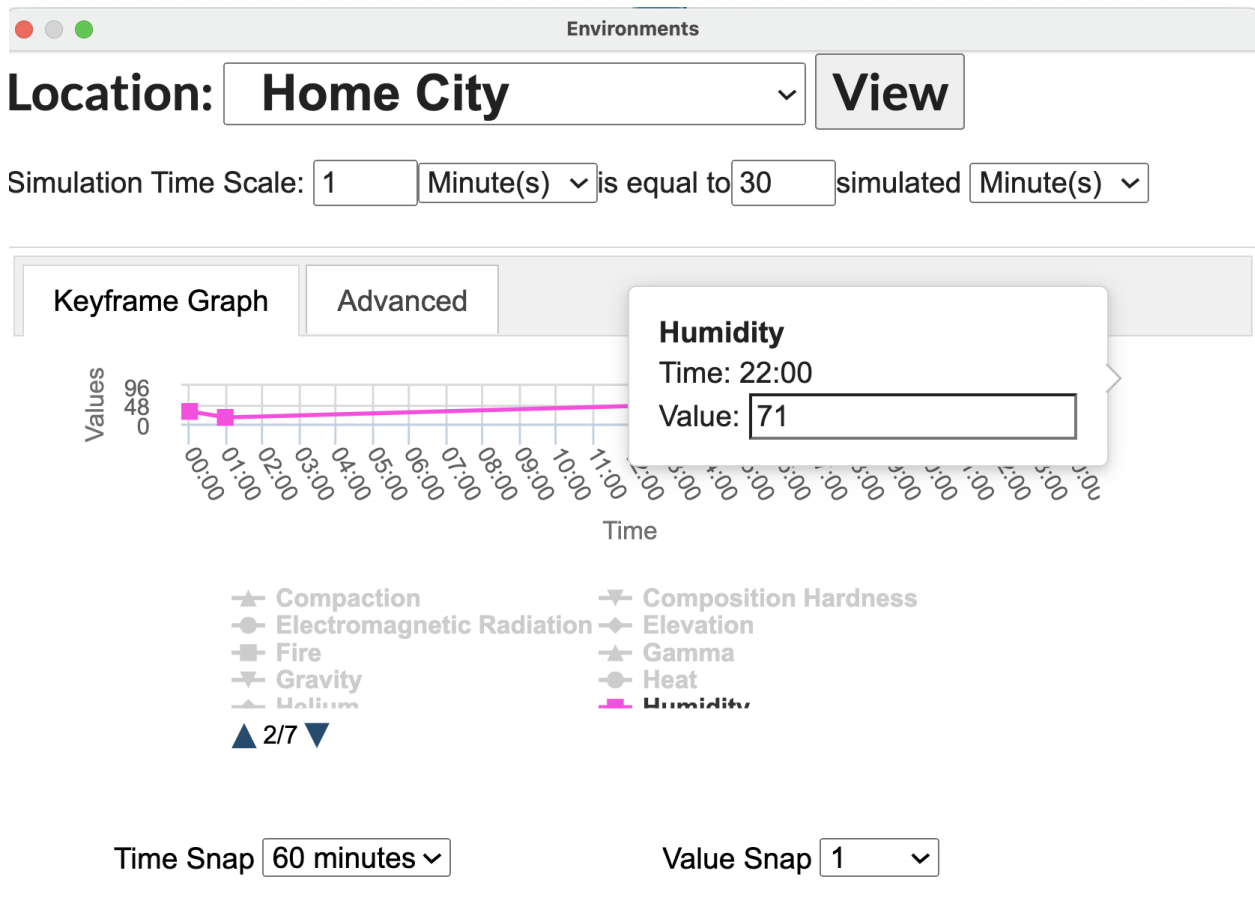
function loop() {
    var humidity = getHumidity();

    Serial.println("Humidity: " + humidity);
```

```
if (humidity != null) {  
  
    if (humidity > 60) {  
        setAirConditioner("on");  
    } else {  
        setAirConditioner("off");  
    }  
} else {  
    Serial.println("Failed to get humidity data");  
}  
  
delay(5000);  
}
```

Алгоритм роботи: підключаємось по мережі до датчика вологості -- фіксуємо значення вологості -- якщо воно вище 60 -- включаємо кондиціонер, якщо менше -- вимикаємо кондиціонер

Симуляція роботи системи:



Виконаємо конфігурацію нашої вологості, аби вона змінювалась декілька разів протягом дня

Запускаємо код:

```
..  
main.js  
26 var humidity = getHumidity();  
27  
28 Serial.println("Humidity: " + humidity);  
29  
30 if (humidity !== null) {  
31  
32     if (humidity > 70) {  
33         setAirConditioner("on");  
34     } else {  
35         setAirConditioner("off");  
36     }  
37 } else {  
38     Serial.println("Failed to get humidity");  
39 }  
40  
41 delay(5000);  
42
```

Air Conditioner off
Humidity: 49.72650146484375
Humidity: 68.45490264892578
Humidity: 75.49240112304688
Air Conditioner on
Humidity: 77.84220123291016

☐ Top

```
Humidity: 11.978599548339844  
Air Conditioner off  
Humidity: 49.72650146484375  
Humidity: 68.45490264892578  
Humidity: 75.49240112304688  
Air Conditioner on  
Humidity: 77.84220123291016
```

☐ Top

Бачимо зміну статусу кондиціонера, в залежності від показу датчика вологості

Висновки

Ознайомився з принципами розробки та реалізації IoT-систем для автоматизації кліматичних умов, енергозбереження та управління рівнем вологості. Навчився підключати сенсори до мікроконтролерів, обробляти

отримані дані та реалізовувати автоматичне керування виконавчими пристроями (система клімат-контролю, кондиціонер, зволожувач повітря).

Контрольні запитання

1. Які основні компоненти необхідні для реалізації системи з використанням сонячних батарей та акумулятора?

У Cisco Packet Tracer для моделювання такої системи можна використовувати:

- Solar Panel – сонячна батарея для генерації енергії.
 - Battery – акумулятор для збереження енергії.
 - Power Controller (IoT Power Meter) – контролер живлення, що керує перемиканням між джерелами енергії.
 - IoT-обладнання (датчики, контролери, актуатори) – пристрої, які використовують енергію.
 - Home Gateway – шлюз для управління енергоспоживанням через програмні алгоритми.
2. Як визначити оптимальну потужність сонячних батарей для IoT-пристроїв

У Cisco Packet Tracer потрібно перевіряти, чи поточна генерація енергії достатня для живлення всіх пристроїв.

3. Механізм автоматичного перемикання між сонячною батареєю та акумулятором

Реалізується через Power Controller, який моніторить рівень заряду акумулятора та інтенсивність сонячного випромінювання.

Алгоритм:

- Якщо сонячна батарея генерує достатньо енергії, пристрої працюють від неї.

- Якщо сонячної енергії недостатньо, система автоматично перемикається на акумулятор.
- Коли рівень заряду акумулятора падає нижче критичного значення, система може вимкнути другорядні пристрої або повідомити користувача.

У Cisco Packet Tracer можна використовувати скрипти на JavaScript у вбудованому контролері для перемикання живлення.

4. Фактори, що впливають на рівень заряду акумулятора, та способи його покращення

Основні фактори:

- Інтенсивність сонячного випромінювання – змінюється залежно від погоди та часу доби.
- Температура – впливає на ефективність акумулятора (при низьких температурах ємність знижується).
- Кількість підключених пристроїв – чим більше споживачів, тим швидше розряджається акумулятор.
- Цикли заряду-розряду – старіння акумулятора зменшує його ефективність.

Способи покращення:

- Використовувати енергоефективні IoT-пристрої.
- Реалізувати динамічне управління навантаженням (наприклад, вимикати другорядні пристрої вночі).
- Використовувати високоякісні акумулятори (LiFePO₄ замість Li-ion).
- Оптимізувати кут нахилу сонячної панелі для максимального виробництва енергії.

