Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №6

з дисципліни «Інженерія систем IoT»

Тема: «Використання реле в IoT застосунках»

Варіант: 65

Виконав:
студент групи ІА-33
Заранік Максим

Перевірив:
асистент кафедри ІСТ
Головатенко І. А.

Київ 2025

Мета: Навчитися використовувати реле у IoT застосунках, та зрозуміти доцільність його використанян у

тих чи інших випадках

Хід роботи

Налаштування датчиків у CounterFit Відповідно до завдання було запущено середовище CounterFit. У конфігурації створено два пристрої:

Датчик вологості ґрунту (Soil Moisture Sensor): Тип – Analog, Пін – 65.

Реле (Relay): Тип – Digital, Пін – 66.

Розробка програмного забезпечення Було написано скрипт мовою Python, який підключається до CounterFit, зчитує значення вологості та керує станом реле.

Результати роботи Програма успішно зчитує дані з віртуального піну 65. При зміні значень у веб-інтерфейсі CounterFit (імітація висихання ґрунту), програма автоматично перемикає стан реле на піні 66, що відображається у консолі та зміною статусу LED-індикатора віртуального реле.

Код клієнтської частини:

app.py

```python
from counterfit_connection import CounterFitConnection
import time
from counterfit_shims_grove.adc import ADC
from counterfit_shims_grove.grove_relay import GroveRelay
import paho.mqtt.client as mqtt
import json

CounterFitConnection.init('127.0.0.1', 5001)
adc = ADC()
relay = GroveRelay(66)

id = '9e9e2074-64b1-4dd2-8ee3-ad2ab0359a77'
client_name = id + 'soil_moisture_sensor_client'
client_telemetry_topic = id + '/telemetry'


mqtt_client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2, client_name)
mqtt_client.connect('test.mosquitto.org')
mqtt_client.loop_start()
print("MQTT connected!")

while True:
    soil_moisture = adc.read(65)
    print("Soil moisture:", soil_moisture)
    telemetry = json.dumps({'soil_moisture': soil_moisture})
    print("Sending telemetry:", telemetry)
    mqtt_client.publish(client_telemetry_topic, telemetry)
    time.sleep(10)
    if soil_moisture > 486:
        print("Soil Moisture is too low, turning relay on.")
        relay.on()
    else:
        print("Soil Moisture is ok, turning relay off.")
        relay.off()
```

Код серверної частини:

app.py

```python
import json
import time
import paho.mqtt.client as mqtt
import threading

id = '9e9e2074-64b1-4dd2-8ee3-ad2ab0359a77'
client_telemetry_topic = id + '/telemetry'
server_command_topic = id + '/command'
client_name = id + 'soil_moisture_sensor_server'

mqtt_client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2, client_name)
mqtt_client.connect('test.mosquitto.org')
mqtt_client.loop_start()

water_time = 5
wait_time = 20

def send_relay_command(client, state):
    command = {'relay_on': state}
    print("Sending message:", command)
    client.publish(server_command_topic, json.dumps(command))

def control_relay(client):
    mqtt_client.unsubscribe(client_telemetry_topic)
    send_relay_command(client, True)
    time.sleep(water_time)
    send_relay_command(client, False)
    time.sleep(wait_time)
    mqtt_client.subscribe(client_telemetry_topic)

def handle_telemetry(client, userdata, message):
    payload = json.loads(message.payload.decode())
    print("Message received:", payload)
    if payload['soil_moisture'] > 486:
        threading.Thread(target=control_relay, args=(client,)).start()

mqtt_client.subscribe(client_telemetry_topic)
mqtt_client.on_message = handle_telemetry

while True:
    time.sleep(2)
```

mosquitto conf

```
listener 1883 0.0.0.0
allow_anonymous true
```

Результат роботи:

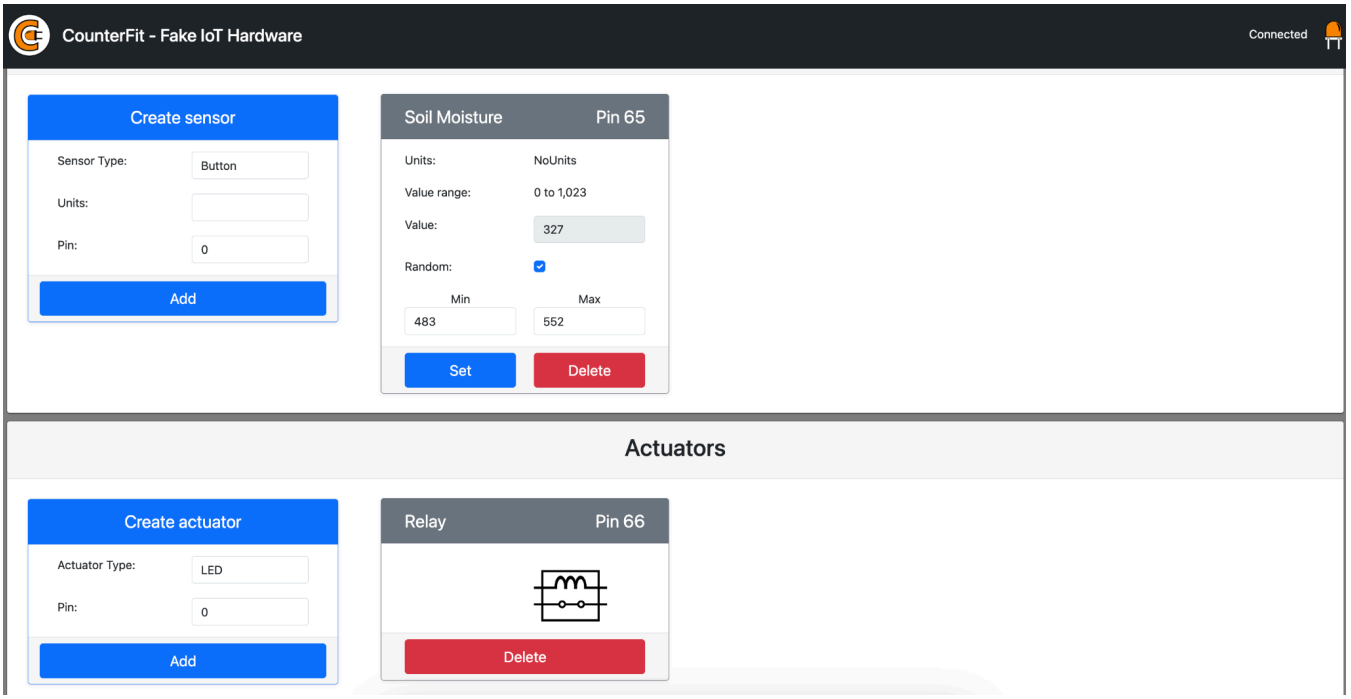

Рисунок 1.1 - інтерфейс Counterfit

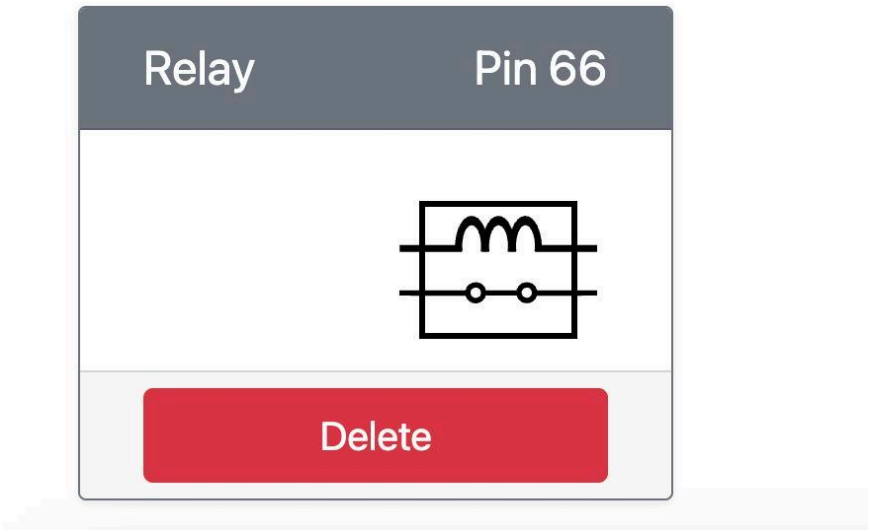

Рисунок 1.2 - стан реле “замкнутий”

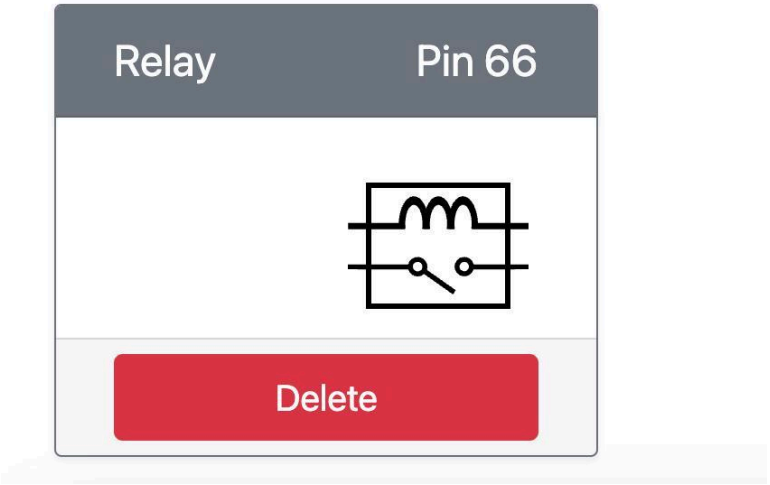

Рисунок 1.3 - стан реле “розімкнутий”

```
Soil moisture: 503
Sending telemetry: {"soil_moisture": 503}
Soil Moisture is too low, turning relay on.
Soil moisture: 483
Sending telemetry: {"soil_moisture": 483}
Soil Moisture is ok, turning relay off.
Soil moisture: 499
```

Рисунок 1.4 - код клієнтської частини який відповідає цьому стану

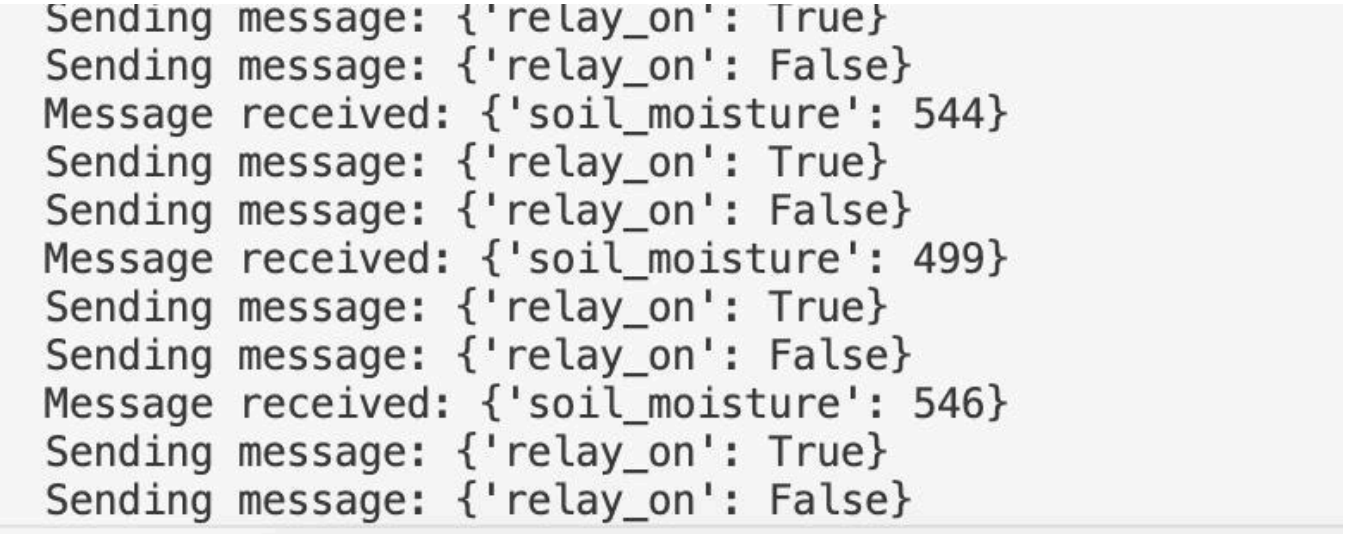

Рисунок 1.5 - код серверної частини який відповідає цьому стану

Посилання на github: https://github.com/makszaranik/IOT-6

Висновок:

У ході виконання лабораторної роботи було набуто практичних навичок роботи з середовищем емуляції CounterFit, що дозволяє розробляти та тестувати IoT-рішення без необхідності використання фізичного обладнання. Було успішно сконфігуровано віртуальні периферійні пристрої: аналоговий датчик вологості ґрунту та цифрове реле. У програмній частині роботи, виконаній мовою Python, було реалізовано взаємодію з віртуальними портами введення-виведення. Зокрема, відпрацьовано алгоритм зчитування аналогових даних з датчика та налаштовано логіку прийняття рішень на основі порогових значень. Створена програма імітує реальну систему розумного поливу: вона моніторить стан ґрунту в режимі реального часу та автоматично активує виконавчий механізм (реле) при зниженні вологості, що демонструє базові принципи побудови автоматизованих систем керування в Інтернеті речей.