|  |  |  |
| --- | --- | --- |
|  | **BỘ CÔNG THƯƠNG** | |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM** | |
|  | **KHOA CÔNG NGHỆ ĐIỆN TỬ** |

A red and white logo

Description automatically generated with low confidence

**BÁO CÁO BUỔI 3**

**BÁO CÁO KẾT QUẢ THỰC HÀNH NHÓM 2**

**Giảng viên : Phạm Quang Trí**

|  |  |
| --- | --- |
| **Họ và Tên** | **MSSV** |
| * Nguyễn Thành Đạt: | 21057761 |
| * Nguyễn Mạnh Đức : | 21039681 |
| * Vi Đại Thái Dũng: | 21033991 |

**Yêu cầu thực hành:**

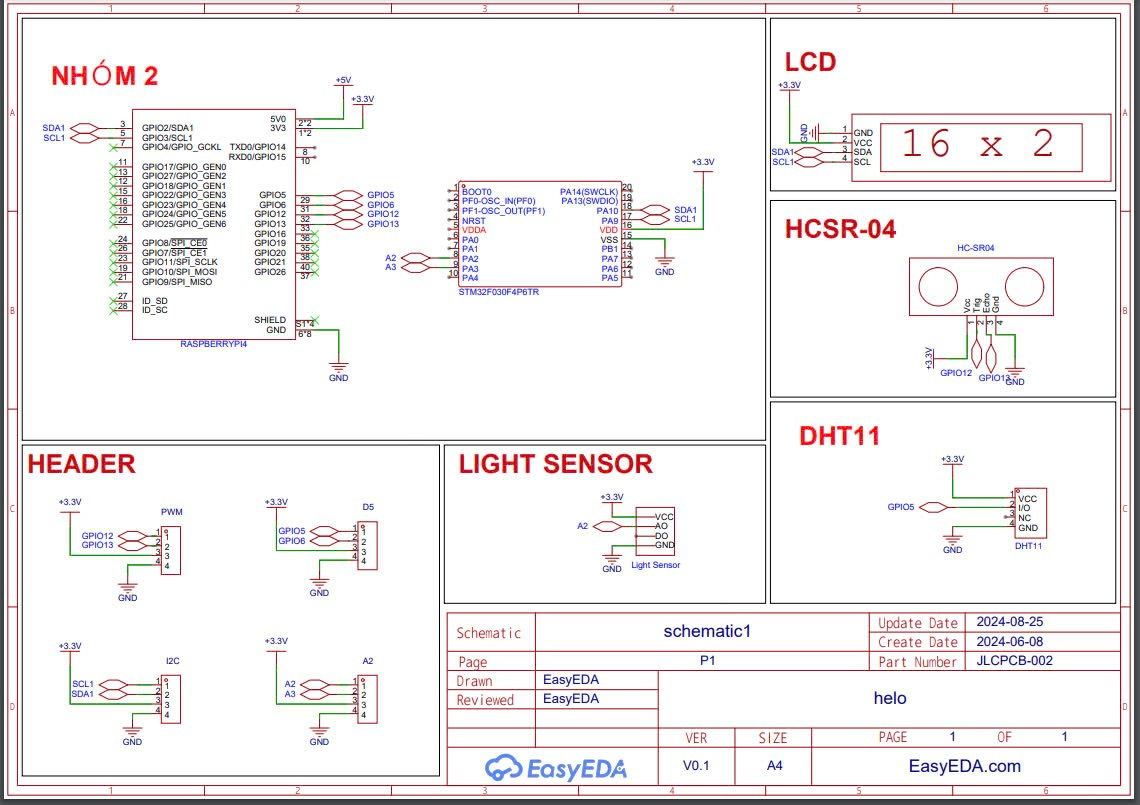
* Viết chương trình điều khiển mô-đun Raspberry Pi kết hợp với Grove Base Hat và các mô-đun Grove phù hợp để thực hiện việc đọc giá trị từ cảm biến nhiệt độ, độ ẩm của môi trường, cảm biến ánh sáng, cảm biến siêu âm đo khoảng cách và hiển thị các giá trị này lên màn hình LCD 16x2. Đồng thời gửi các dữ liệu này lên Server của ThingSpeak mỗi 20 giây cho một gói tin và thực hiện liên tục trong ít nhất là 45 phút. Bắt buộc phải sử dụng đồng thời cả hai giao thức HTTP và MQTT.
* ***Yêu cầu 1:*** Bố cục, phân công nhiệm vụ từng thành viên, trình bày nội dung báo cáo.
* ***Yêu cầu 2:*** Sơ đồ nguyên lý chi tiết cảu hệ thống( Bắt buộc vẽ bằng phần mềm).
* ***Yêu cầu 3:*** Lưu đồ giải thuật.
* ***Yêu cầu 4:*** Mã nguồn chương trình.
* ***Yêu cầu 5:*** Giải thích tác dụng của từng dòng mã nguồn được bổ sung vào.
* ***Yêu cầu 6:*** Minh chứng kết quả thực nghiệm
* ***Yêu cầu 7:*** Minh chứng kết quả thực nghiệm bằng cách quay một video clip và tải lên Youtube

**BT MỨC ĐỘ 3 (10 điểm):**

* ***Yêu cầu 1:*** Bố cục, phân công nhiệm vụ từng thành viên, trình bày nội dung báo cáo.

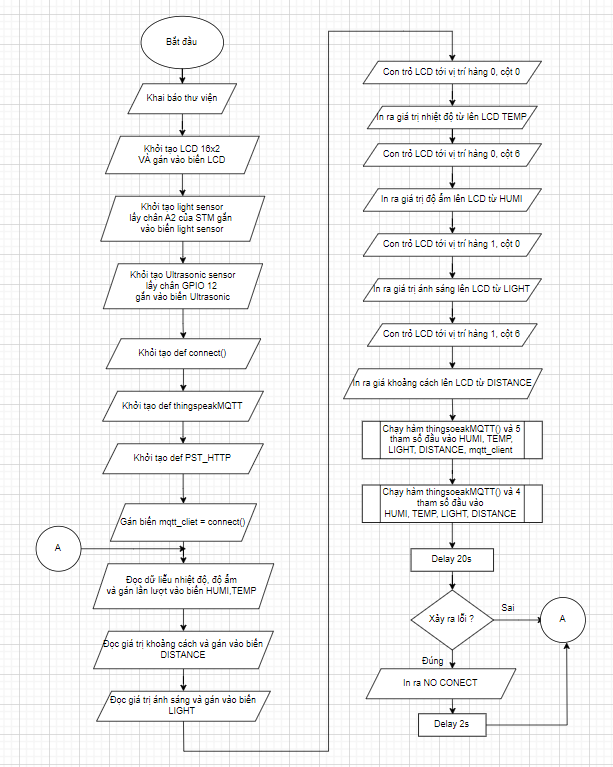
|  |  |
| --- | --- |
| Nguyễn Thành Đạt | Làm báo cáo,lưu đồ giải thuật |
| Nguyễn Mạnh Đức | Lưu đồ giải thuật, giải thích code |
| Vi Đại Thái Dũng | Làm video thuyết trình, vẽ sơ đồ nguyên lý |

* ***Yêu cầu 2:*** Sơ đồ nguyên lý chi tiết cảu hệ thống( Bắt buộc vẽ bằng phần mềm).

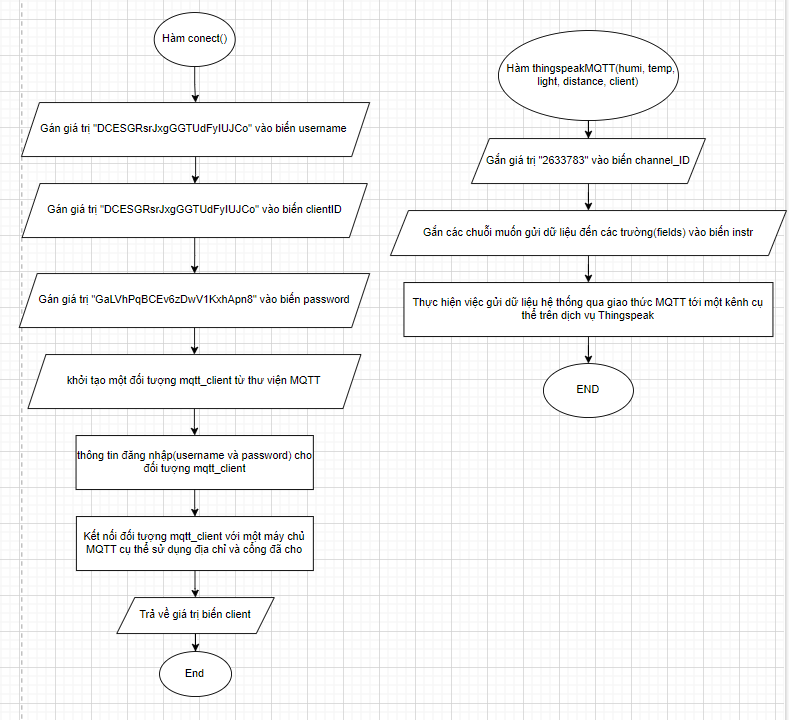
****

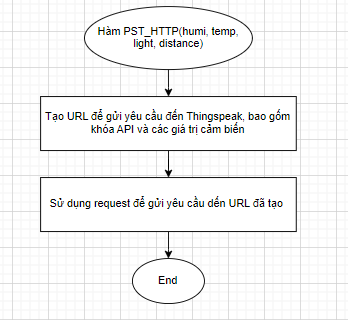
***Yêu cầu 3:*** Lưu đồ giải thuật.

**- Giao thức HTTP và MQTT:**



**Chương trình con:**





* ***Yêu cầu 4:*** Mã nguồn chương trình.

***Code:***

import paho.mqtt.client as mqtt

from time import sleep

from grove.grove\_light\_sensor\_v1\_2 import GroveLightSensor

from grove.grove\_ultrasonic\_ranger import GroveUltrasonicRanger

from grove.display.jhd1802 import JHD1802

from seeed\_dht import DHT

from urllib import request, parse

#KHAI BAO CHAN MODULE WITH RASPI4

lcd = JHD1802()

sensor = DHT('11',5)

Distance = GroveUltrasonicRanger(12)

Light\_led = GroveLightSensor(2)

#CONNETED MQTT

def connect():

    username = "DCESGRsrJxgGGTUdFyIUJCo"

    clientId = "DCESGRsrJxgGGTUdFyIUJCo"

    password = "GaLVhPqBCEv6zDwV1KxhApn8"

    client = mqtt.Client(clientId)

    client.username\_pw\_set(username=username, password=password)

    client.connect("mqtt3.thingspeak.com", 1883, 60)

    return client

def thingspeakMQTT(humi, temp, light, distance, client):

    channel\_ID = "2633783"

    instr = f"&field1={humi}&field2={temp}&field3={light}&field4={distance}&status=MQTTPUBLISH"

    client.publish(f"channels/{channel\_ID}/publish", instr)

def PST\_HTTP(humi, temp, light, distance):

    url = f"https://api.thingspeak.com/update?api\_key=RK84OE4D9FP7EJ3L&field1={humi}&field2={temp}&field3={light}&field4={distance}"

    request.urlopen(url)

mqtt\_cliet = connect()

while 1:

    try:

        HUMI,TEMP =  sensor.read()

        DISTANCE = int(Distance.get\_distance())

        LIGHT = Light\_led.light

        #In LEN LCD

        lcd.clear()

        lcd.setCursor(0,0)

        lcd.write('{0:0.1f}C'.format(TEMP))

        lcd.setCursor(0,6)

        lcd.write('{0:0.1f}%'.format(HUMI))

        lcd.setCursor(1,0)

        lcd.write('{0:2}'.format(LIGHT))

        lcd.setCursor(1,6)

        lcd.write('{0:2}Cm'.format(DISTANCE))

        #GUI DATA LEN MQTT\_HTTP

        thingspeakMQTT(HUMI, TEMP, LIGHT, DISTANCE, mqtt\_client)

        PST\_HTTP(HUMI, TEMP, LIGHT, DISTANCE)

        sleep(20)

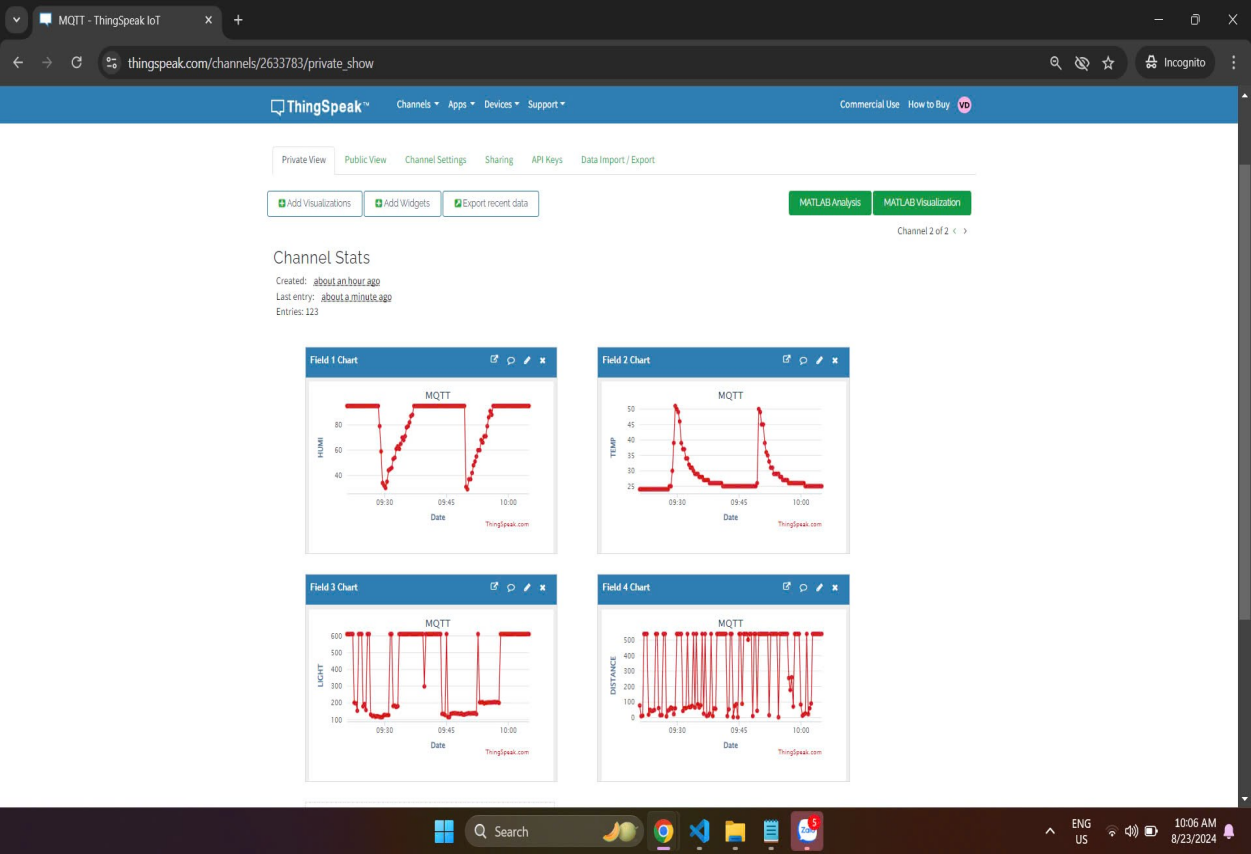
    except:

      print("NO CONNECT")

      sleep(2)

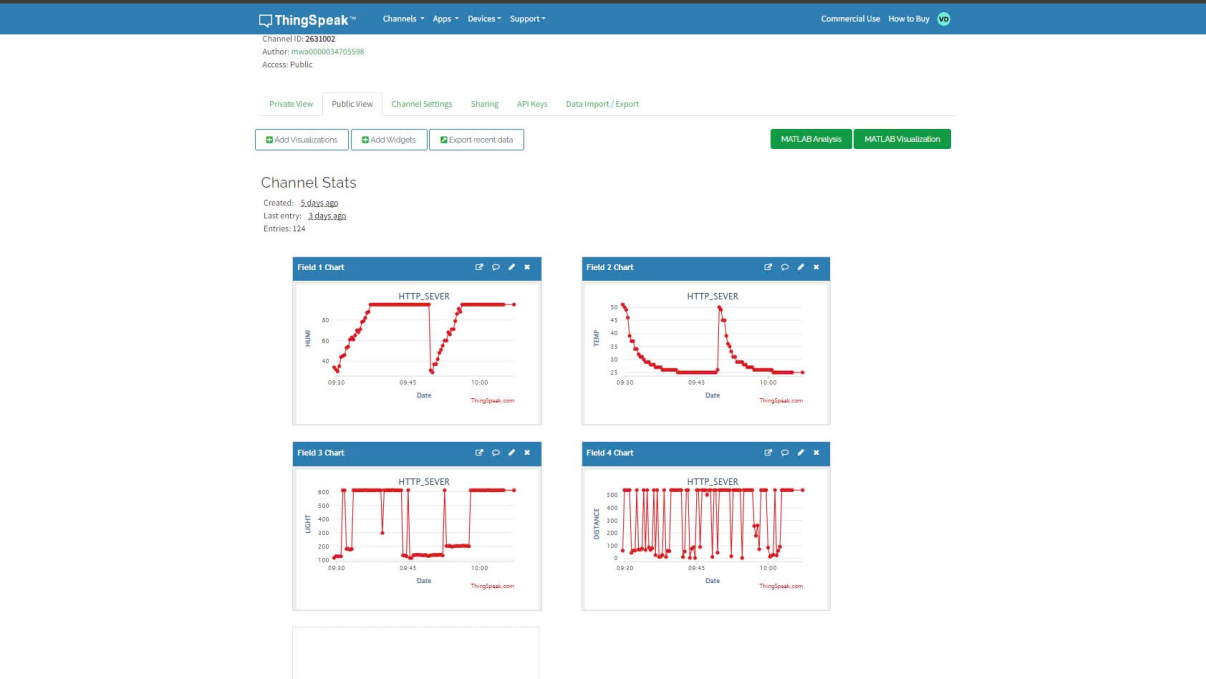
* ***Yêu cầu 5:*** Giải thích tác dụng của từng dòng mã nguồn được bổ sung vào.
* import paho.mqtt.client as mqtt
* from time import sleep
* from grove.grove\_light\_sensor\_v1\_2 import GroveLightSensor
* from grove.grove\_ultrasonic\_ranger import GroveUltrasonicRanger
* from grove.display.jhd1802 import JHD1802
* from seeed\_dht import DHT
* from urllib import request, parse
* #KHAI BAO CHAN MODULE
* lcd = JHD1802() #Khởi tạo LCD
* sensor = DHT('11',5) #Khai báo cảm biến DHT11 tại chân số 5
* Distance = GroveUltrasonicRanger(12) #Khai báo cảm biến siêu âm tại chân số 12
* Light\_led = GroveLightSensor(2) #Khai báo cảm biến ánh sáng tại chân analog số 2
* #Kết nối với MQTT
* def connect():
* username = "DCESGRsrJxgGGTUdFyIUJCo" #Tên đăng nhập MQTT
* clientId = "DCESGRsrJxgGGTUdFyIUJCo" #ID của client MQTT
* password = "GaLVhPqBCEv6zDwV1KxhApn8" #Mật khẩu MQTT
* client = mqtt.Client(clientId)  #Tạo một đối tượng client MQTT
* client.username\_pw\_set(username=username, password=password)  #Thiết lập tên đăng nhập và mật khẩu cho MQTT
* client.connect("mqtt3.thingspeak.com", 1883, 60)  #Kết nối tới MQTT broker qua cổng 1883
* return client  #Trả về client đã kết nối
* #Gửi dữ liệu lên MQTT
* def thingspeakMQTT(humi, temp, light, distance, client):
* channel\_ID = "2633783"  #ID của kênh ThingSpeak
* instr = f"&field1={humi}&field2={temp}&field3={light}&field4={distance}&status=MQTTPUBLISH"  #Chuẩn bị chuỗi dữ liệu
* client.publish(f"channels/{channel\_ID}/publish", instr)  #Gửi dữ liệu lên kênh ThingSpeak
* #Kết nối với HTTP
* def PST\_HTTP(humi, temp, light, distance):
* url=f"https://api.thingspeak.com/update?api\_key=RK84OE4D9FP7EJ3L&field1={humi}&field2={temp}&field3={light}&field4={distance}"  #Chuẩn bị URL với dữ liệu
* request.urlopen(url)  #Gửi yêu cầu HTTP GET tới ThingSpeak
* mqtt\_client = connect() #Tạo biến kết nối với hàm connect
* while 1:
* try:
* HUMI, TEMP = sensor.read()  #Đọc độ ẩm và nhiệt độ từ cảm biến DHT11
* DISTANCE = int(Distance.get\_distance())  #Đọc khoảng cách từ cảm biến siêu âm
* LIGHT = Light\_led.light  #Đọc mức độ ánh sáng từ cảm biến ánh sáng
* # In len LCD
* lcd.clear()  #Xóa màn hình LCD
* lcd.setCursor(0, 0)  #Đặt con trỏ đến hàng đầu tiên, cột đầu tiên
* lcd.write('{0:0.1f}C'.format(TEMP))  #Hiển thị nhiệt độ trên LCD
* lcd.setCursor(0, 6)  #Đặt con trỏ đến hàng đầu tiên, cột thứ bảy
* lcd.write('{0:0.1f}%'.format(HUMI))  #Hiển thị độ ẩm trên LCD
* lcd.setCursor(1, 0)  #Đặt con trỏ đến hàng thứ hai, cột đầu tiên
* lcd.write('{0:2}'.format(LIGHT))  #Hiển thị mức độ ánh sáng trên LCD
* lcd.setCursor(1, 6)  #Đặt con trỏ đến hàng thứ hai, cột thứ bảy
* lcd.write('{0:2}Cm'.format(DISTANCE))  #Hiển thị khoảng cách trên LCD
* #Gửi dữ liệu lên ThinkSpeak
* thingspeakMQTT(HUMI, TEMP, LIGHT, DISTANCE, mqtt\_client) #Gửi dữ liệu qua MQTT
* PST\_HTTP(HUMI, TEMP, LIGHT, DISTANCE) #Gửi dữ liệu qua HTTP
* sleep(20) #Chờ 20 giây
* except:
* print("NO CONNECT") #In ra màn hình "No Connect" nếu kết nối thất bại
* sleep(2) #Chờ 2 giây
* ***Yêu cầu 6:*** Minh chứng kết quả thực nghiệm
* **Giao thức MQTT**

**ID: 2633783**

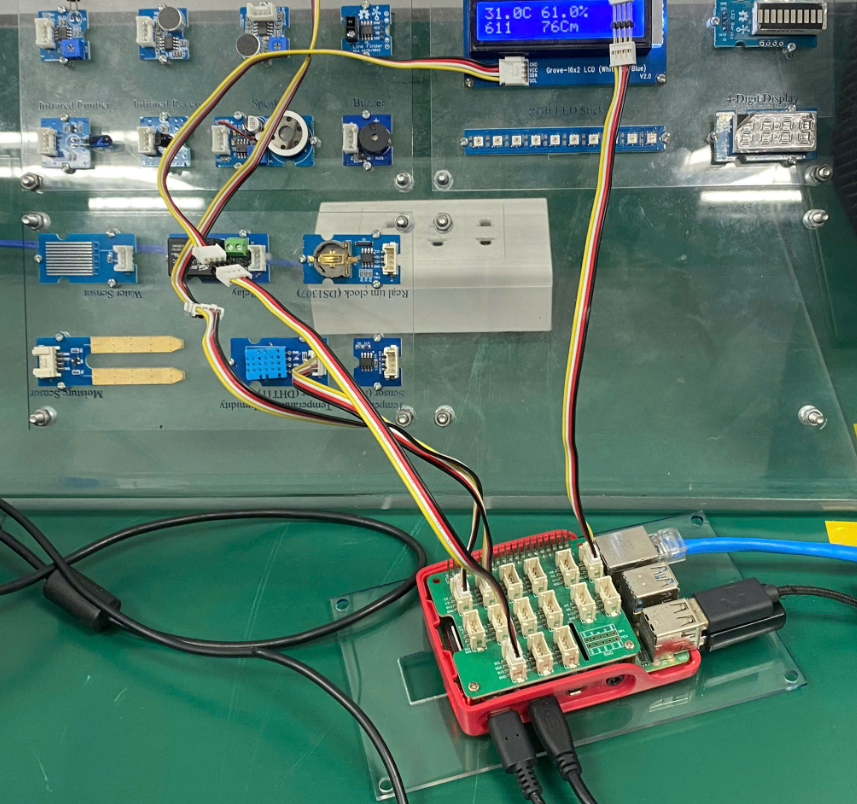


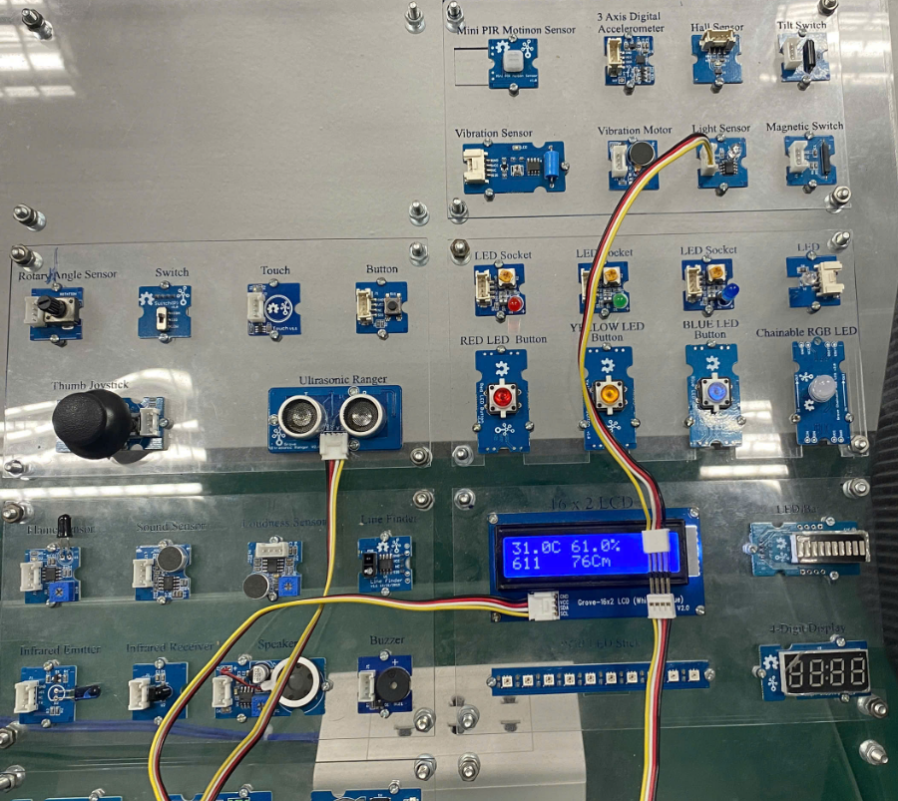
* **Giao thức HTTP**

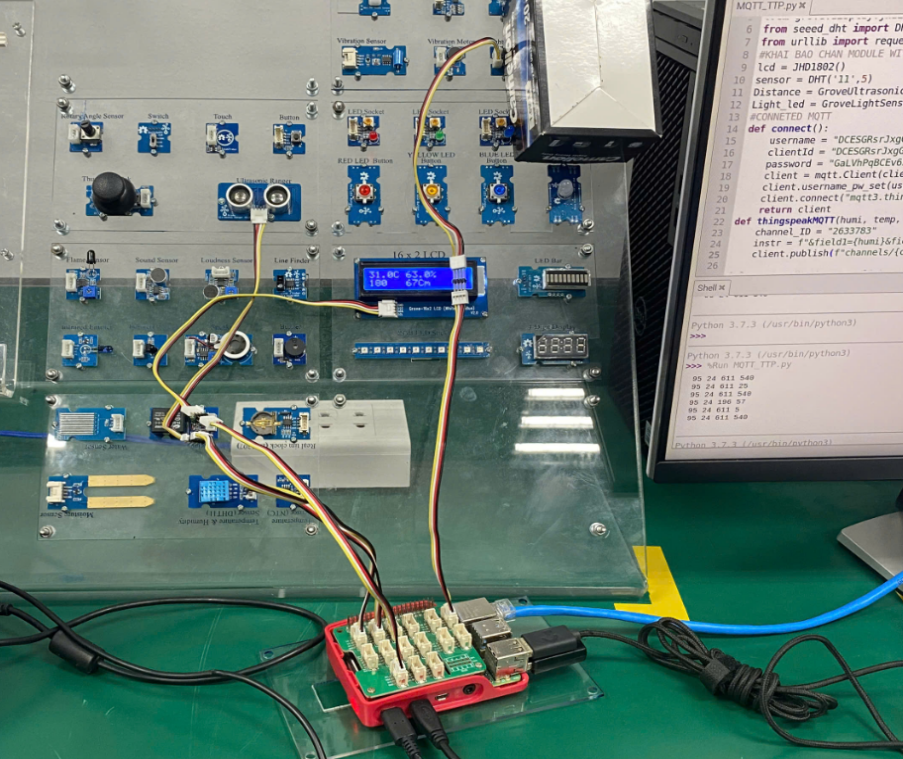
**ID: 2631002**



* **Minh chứng các modul:**







* ***Yêu cầu 7:*** Minh chứng kết quả thực nghiệm bằng cách quay một video clip và tải lên Youtube

**Link : https://youtu.be/13KCQz8RChU**