



IOT LAB
REPORT

2024

(CO3038) Phát triển Ứng dụng
Internet Of Things
Bài thí nghiệm số 3

GVHD: VŨ TRỌNG THIÊN

Người viết: Tô Hoàng Phong - 2112012

Ho Chi Minh University of Technology (HCMUT)

Khoa khoa học và kỹ thuật máy tính

Mục lục

1	Đường dẫn github và một số đường dẫn khác (nếu có)	3
2	Yêu cầu	3
3	Kết nối bắt tay và giao thức Stop-and-Wait ARQ	4
3.1	Khái niệm	4
3.2	Máy trạng thái	6
3.3	Hiện thực trên python	6
4	Kết quả	7

1 Đường dẫn github và một số đường dẫn khác (nếu có)

- Đường dẫn github của bài thí nghiệm: [github](#)
- Đường dẫn dashboard của bài thí nghiệm: [dashboard](#)
- Đường dẫn video demo: [youtube](#)

2 Yêu cầu

Các yêu cầu đã thực hiện được trong phần thí nghiệm này:

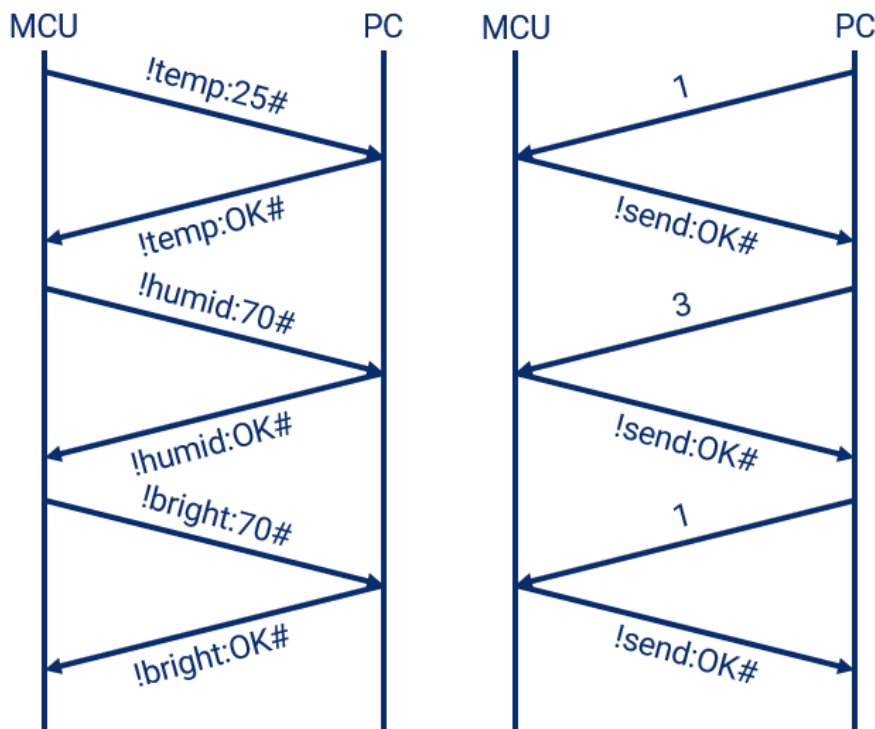
- Nhận tín hiệu từ server
- Chạy code trên môi trường Window.
- Chạy code trên môi trường Linux.

3 Kết nối bắt tay và giao thức Stop-and-Wait ARQ

3.1 Khái niệm

Giao thức Stop and wait ARQ yêu cầu bên gửi (sender) và bên nhận (receiver) tuân thủ quy tắc như sau: Bên gửi mỗi lần chỉ gửi một gói tin, và chỉ gửi tiếp gói tin khác khi nhận được ACK của gói đã gửi trước đó.

Đây là giao thức cơ bản nhất về kỹ thuật bắt tay, giúp bên nhận và bên gửi có thể xác định được kết nối có còn tồn tại hay không.

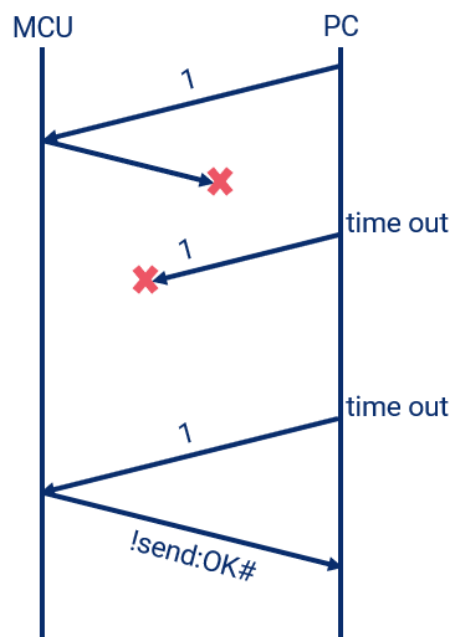


Hình 1: Giao thức stop-and-wait

Hình trên mô tả chi tiết về cách thức và nguyên lý hoạt động của giao thức giữa MCU và PC:

- MCU lấy dữ liệu từ cảm biến, gửi cho IOT gateway (PC) gói tin cú pháp: `!<name>:<data>#`. Nếu PC đã nhận được thì sẽ phản hồi: `!<name>:OK#`
- PC lấy tín hiệu điều khiển từ server, gửi cho MCU lệnh điều khiển. Nếu kết nối ổn định thì MCU sẽ phản hồi: `!send:OK#`

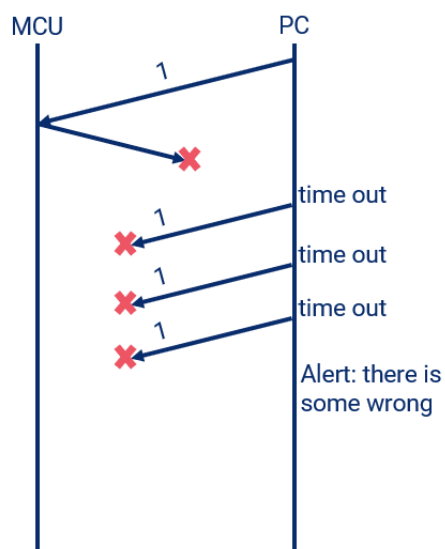
Trường hợp lỗi và xử lý: Lỗi mất tín hiệu trong quá trình gửi (cả bên gửi và bên nhận)



Hình 2: Lỗi mất gói tin

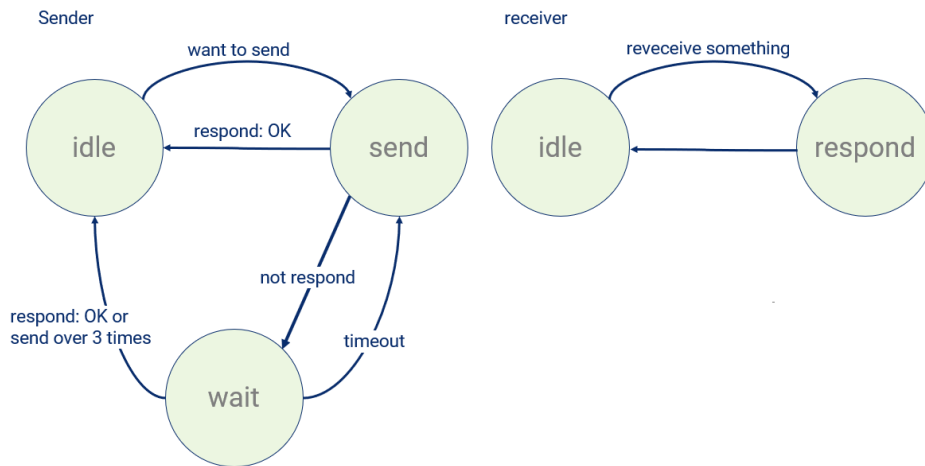
Xử lý: bên gửi sẽ gửi lại gói tin khi hết thời gian chờ phản hồi (round trip time)

Đặc biệt: Trong trường hợp gửi lại quá số lần quy định (thường là 3) mà vẫn chưa thấy phản hồi, ta sẽ thông báo lỗi về kết nối cho người dùng, và sẽ dừng gửi.



Hình 3: Lỗi mất kết nối

3.2 Máy trạng thái



Hình 4: Máy trạng thái cho bên nhận và gửi

Bảng chức năng:

Tên trạng thái	Mô tả
idle	Trạng thái rảnh rỗi, chờ tác vụ
send	Bên gửi gửi gói tin Tăng số lần gửi
wait	Đếm thời gian, nếu hết thời gian chờ mà chưa thấy phản hồi thì chuyển trạng thái Nếu gửi quá số lần thì cảnh báo cho người dùng
resond	Bên nhận sau khi nhận gói tin sẽ xử lý, rồi gửi phản hồi

Bảng 1: Bảng mô tả trạng thái

3.3 Hiện thực trên python

```

1 fsm_communication = "idle"
2 ...
3 if fsm_communication == "idle":
4     if my_server.received == True:
5         fsm_communication = "send"
6 elif fsm_communication == "send":
7     # send data
8     send_time += 1
9     mess = my_server.data
10    my_serial.send_data(mess)
11    if my_serial.check_connection == True:
12        # if receiving OK from sender
13        my_server.received = False
14        fsm_communication = "idle"
15        send_time = 0
16    else:
17        # haven't not responded yet
18        timer_connection = 2
19        fsm_communication = "wait"
20 elif fsm_communication == "wait":
21     timer_connection -= 1
22     if my_serial.check_connection == True:
23         # if receiving OK from sender
24         send_time = 0
25         my_server.received = False
26         fsm_communication = "idle"
27     elif timer_connection <= 0:
28         fsm_communication = "send"
29     elif send_time >= 3:

```

```

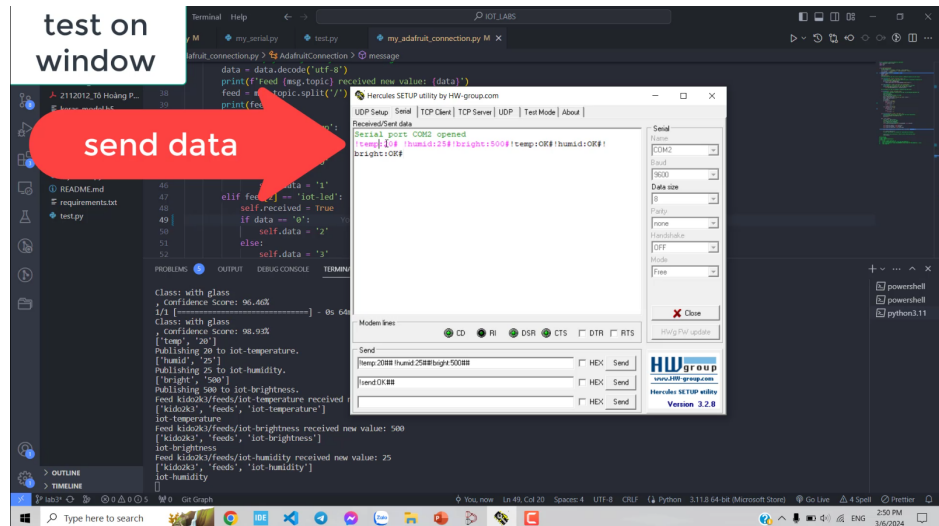
30 # sending over 3 times
31 print("THERE IS SOME WRONG IN SERIAL")
32 my_server.received = False
33 fsm_communication = "idle"
34 send_time = 0

```

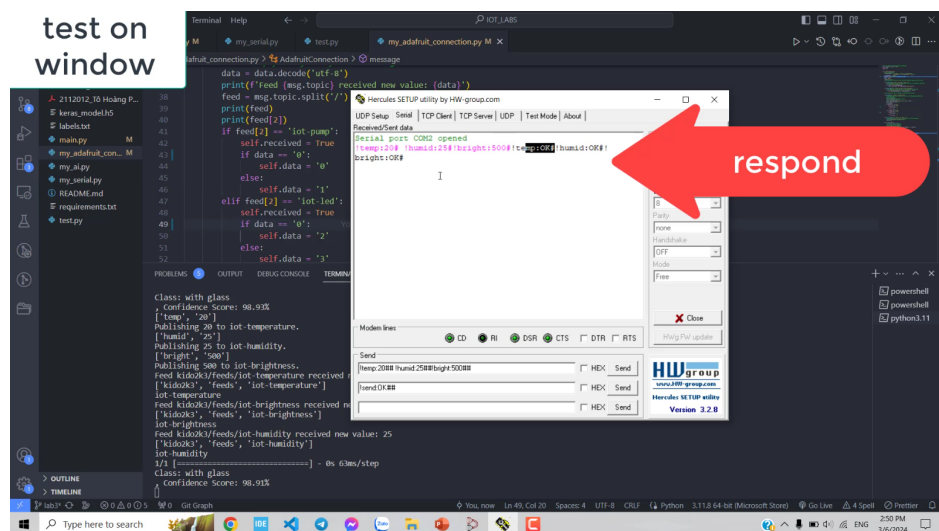
Listing 1: máy trạng thái

4 Kết quả

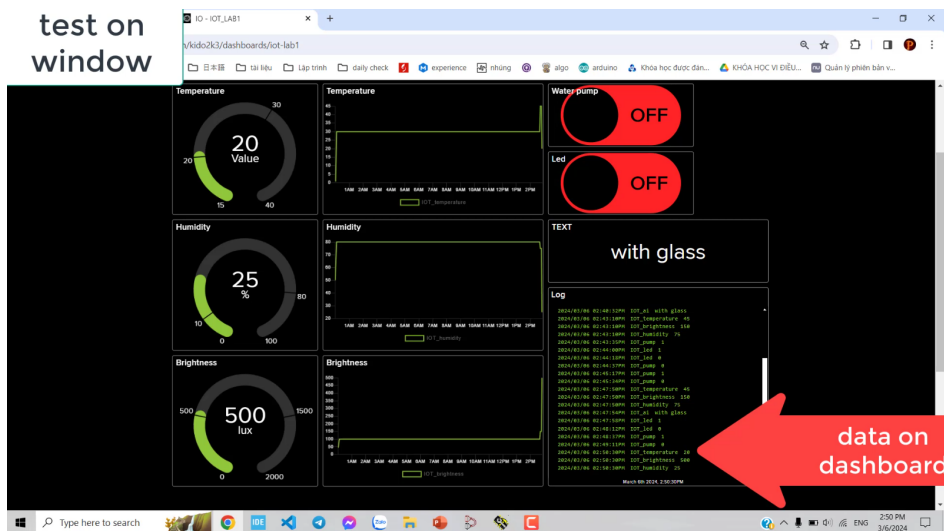
Xem chi tiết tại: [đây](#)



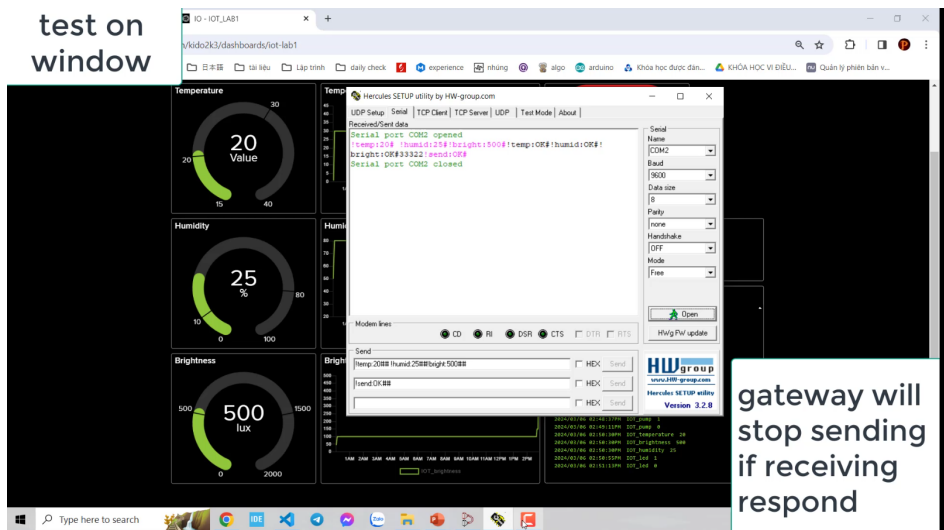
Hình 5: Kết quả trên window 1



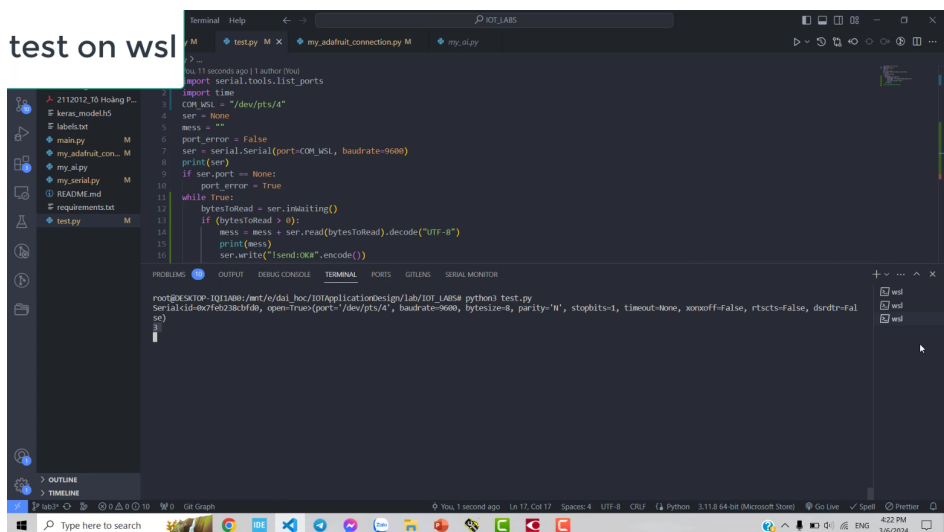
Hình 6: Kết quả trên window 2



Hình 7: Kết quả trên window 3



Hình 8: Kết quả trên window 4



Hình 9: Kết quả trên wsl 1

test on wsl

```

1 #!/usr/bin/env python
2 import serial.tools.list_ports
3
4 import time
5 COM_MSI = "/dev/pts/4"
6 ser = None
7 mess = ""
8 port_error = False
9 ser = serial.Serial(port=COM_MSI, baudrate=9600)
10 print(ser)
11 if ser.port == None:
12     port_error = True
13 while True:
14     bytesToRead = ser.inWaiting()
15     if (bytesToRead > 0):
16         mess = mess + ser.read(bytesToRead).decode("UTF-8")
17         print(mess)
18         ser.write("send:OK".encode())

```

root@DESKTOP-IQ1IAB8:/mnt/e/dai_hoc/IOT/ApplicationDesign/lab/IOT_LAB8# python3 main.py

Connected to Adafruit IOT

Subscribed successfully to iot-pump with qos 0

Subscribed successfully to iot-led with qos 0

Subscribed successfully to iot-brightness with qos 0

Subscribed successfully to iot-humidity with qos 0

Subscribed successfully to iot-temperature with qos 0

Subscribed successfully to iot-ai with qos 0

read k1d0k3/fredu/iot-led received new value: 1

["k1d0k3", "fredu", "iot-led"]

iot-led

["send", "OK"]

Hình 10: Kết quả trên wsl 2