



# Samsung Innovation Campus

| Vạn vật kết nối – IoT

Together for Tomorrow!  
**Enabling People**

Education for Future Generations

Chương 6.

# Truyền thông Mạng Và Lập trình Mạng

**Vạn vật kết nối – IoT**

# Mô tả Chương

---

## Mục tiêu của các chương

- ✓ Tìm hiểu về đặc điểm cần có của cơ sở hạ tầng và các thiết bị phần cứng để sử dụng Mạng Dữ liệu.
- ✓ Tìm hiểu về cấu trúc liên kết mạng và cấu hình mạng theo khoảng cách.
- ✓ Tìm hiểu khái niệm về lập trình socket và hiểu lập trình mạng bằng Raspberry Pi.
- ✓ Có thể lưu trữ dữ liệu đã thu thập bằng cách sử dụng cơ sở dữ liệu và tìm kiếm dữ liệu đã thu thập được.
- ✓ Trong thực tế, có thể sử dụng Raspberry Pi làm máy chủ cơ sở dữ liệu.

## Nội dung của các chương

- ✓ Bài 1. Tìm hiểu về Mạng
- ✓ Bài 2. Các giao thức
- ✓ Bài 3. Lập trình Socket
- ✓ Bài 4 Hướng dẫn truyền thông socket với Raspberry Pi
- ✓ Bài 5. Sử dụng Raspberry Pi làm máy chủ cơ sở dữ liệu

Bài 1.

# Tìm hiểu về Mạng

- | **1.1. Tổng quan về Mạng**
- | 1.2. Thiết bị Mạng
- | 1.3. Chuyển đổi Mạng
- | 1.4. Phương thức truyền dữ liệu của Mạng
- | 1.5. Phân loại và Tính năng của giao tiếp có dây
- | 1.6. Cấu trúc liên kết mạng
- | 1.7. Sử dụng Mạng

# Tổng quan về Mạng?

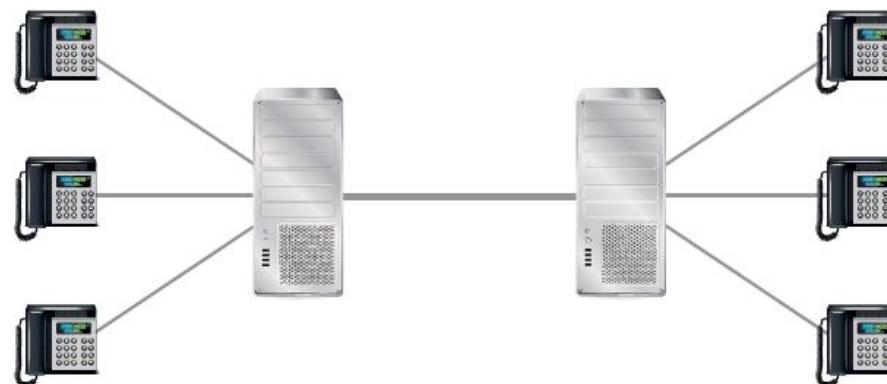
| Một hệ thống các máy tính được kết nối với nhau bằng phương tiện truyền dẫn nhằm trao đổi thông tin.

- ▶ Mạng lưới máy tính
  - Nơi gửi, nơi nhận, phương tiện truyền dẫn
- ▶ phương tiện truyền dẫn
  - Dây nối, không dây
- ▶ Giao thức
  - Một quy trình hoặc giao thức xác định đối tượng và cách thức giao tiếp dữ liệu khi truyền dữ liệu giữa hai thực thể giao tiếp.
  - Kiểu giao thức
    - TCP, IP, UDP, HTTP

## Mạng Điện thoại

| Loại viễn thông đầu tiên là mã Morse do Samuel Morse phát minh vào năm 1837, và điện thoại do Alexander Bell phát minh vào năm 1876.

- ▶ Về cơ bản, mạng điện thoại truyền các tín hiệu analog
  - Nếu khoảng cách truyền xa, tín hiệu sẽ yếu đi và sẽ cần đến một bộ khuếch đại.
  - Các mạng máy tính đời đầu sử dụng các mạng điện thoại này.
- ▶ Ban đầu, điện thoại chỉ có thể kết nối một - một.
  - Khi số lượng điện thoại tăng lên, kết nối một - một trở nên đắt đỏ và không hiệu quả bởi số lượng đường dẫn đã tăng lên.



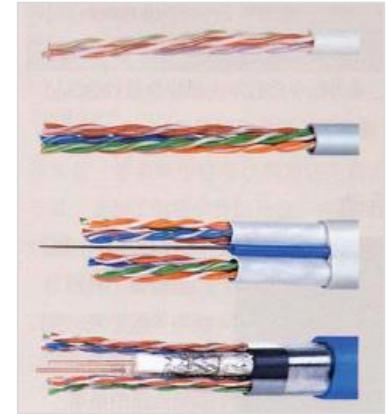
## Lịch sử Mạng

| Lịch sử xây dựng mạng sử dụng máy tính không quá dài, vào những năm 1960, một phương pháp thô sơ để kết nối nhiều thiết bị đầu cuối vào một máy tính đã được sử dụng.

- ▶ ARPANET (Mạng lưới cơ quan với các đề án nghiên cứu tiên tiến)
  - Năm 1969, Bộ Quốc phòng Hoa Kỳ đã tạo ra mạng ARPANET để chia sẻ và truyền tải tài nguyên của các dự án trên khắp nước Mỹ.
- ▶ SNA (Kiến trúc Mạng Hệ thống)
  - Năm 1972, IBM
- ▶ Ethernet
  - Năm 1974, Xerox
- ▶ NSFNET
  - Năm 1986, Quỹ Khoa học Quốc gia (NFS) đã kết nối mạng của họ, NFSNET, với mạng ARPANET.
- ▶ TCP/IP
  - Năm 1982, giao thức TCP/IP được tạo và được sử dụng làm giao thức mạng Internet.
  - WWW
    - Năm 1992, WWW được phát triển và khiến mạng Internet lan rộng nhanh chóng.

## Phương tiện Truyền dẫn

- | Mạng nào cũng cần một phương tiện truyền dẫn để kết nối nơi gửi và nơi nhận.
- | Mỗi phương tiện truyền dẫn đều có một tính chất đặc trưng như băng thông, độ trễ khi truyền dẫn, v.v.
  - ▶ **Băng thông**
    - Khoảng cách giữa tần số tối đa và tần số tối thiểu của tín hiệu đi qua phương truyền dữ liệu
    - Băng thông rộng hơn giúp truyền nhiều dữ liệu hơn trên một đơn vị thời gian
  - ▶ **Cáp xoắn đôi**
    - Phương tiện truyền dẫn được cấu tạo bởi hai sợi đồng xoắn đôi và được bọc nhựa bên ngoài
  - ▶ **Cáp đồng trực**
    - Một dây dẫn ở giữa được bọc một lớp cách điện và một dây dẫn khác phía ngoài lớp cách điện đó
    - Băng thông rộng hơn và việc tốc độ truyền dữ liệu nhanh hơn nhờ tần số cao hơn cáp xoắn đôi, nhưng đắt hơn



Cáp xoắn đôi



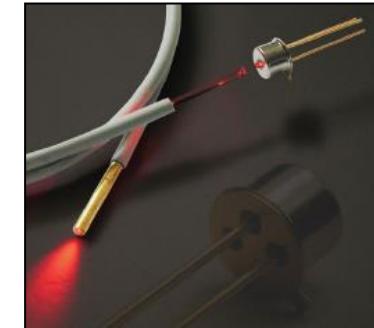
Cáp đồng trực

## Các loại phương tiện truyền dẫn

- | Sợi quang truyền ánh sáng thông qua các sợi thủy tinh mỏng hơn tóc chứ
- | không truyền dữ liệu bằng điện tới các loại dây đồng như cáp xoắn.

- ▶ **Sợi quang**

- Tia sáng được truyền đi nhanh hơn nhiều so với tín hiệu điện
- Băng thông rất cao và ít bị ảnh hưởng bởi các bước sóng điện từ
- Độ bảo mật tín hiệu xuất sắc so với các phương tiện truyền dẫn khác vì không bị ảnh hưởng từ các thiết bị liên lạc



Optical fiber

- ▶ **Vệ tinh**

- Vệ tinh được trang bị một bộ phát đáp và hoạt động như một thiết bị chuyển tiếp để truyền dữ liệu giữa hai trạm mặt đất.



Satellite

Bài 1.

# Tìm hiểu về Mạng

- | 1.1. Tổng quan về Mạng
- | **1.2. Thiết bị Mạng**
- | 1.3. Chuyển đổi Mạng
- | 1.4. Phương thức truyền dữ liệu của Mạng
- | 1.5. Phân loại và Tính năng của giao tiếp có dây
- | 1.6. Cấu trúc liên kết mạng
- | 1.7. Sử dụng Mạng

## Thiết bị truy cập mạng: Modem

| Modem là một thiết bị giúp chuyển tín hiệu digital thành tín hiệu analog và ngược lại.

▶ Modem

- Khi mới ra mắt công nghệ giao tiếp bằng máy tính, nó được sử dụng để liên lạc giữa các máy tính bằng mạng điện thoại có sẵn.
- Vì mạng điện thoại truyền được các tín hiệu analog như giọng nói, nên để các máy tính giao tiếp với nhau qua mạng điện thoại thì cần một bộ chuyển đổi để chuyển tín hiệu analog sang tín hiệu digital.
- Bộ điều chế
  - Modem chuyển dữ liệu digital của máy tính sang tín hiệu analog.
- Bộ giải điều chế
  - Có chức năng phục hồi tín hiệu analog về tín hiệu digital



## Card giao tiếp mạng và thiết bị trung tâm

| Card giao tiếp mạng (NIC) thường được gọi là card LAN và card Ethernet giúp kết nối máy tính với các mạng bên ngoài.

▶ Card giao tiếp mạng (NIC)

- Không giống các modem kết nối các đường dây điện thoại truyền tín hiệu analog, card mạng trực tiếp truyền tín hiệu digital.

▶ Thiết bị trung tâm

- Gửi dữ liệu từ nơi này đến nơi khác

- Thiết bị trung tâm giả

- Truyền dữ liệu đến từ máy tính này sang máy tính khác qua mạng một cách đơn giản.

- Vì các máy chủ đều sử dụng chung một băng thông nên việc tăng số lượng máy chủ sẽ làm giảm tốc độ truyền dữ liệu.

- Thiết bị trung tâm chuyển đổi

- Ngoài chức năng chuyển tiếp, thiết bị này còn có khả năng chuyển đến một địa chỉ đích.

- Mạng này hiệu quả hơn rất nhiều vì băng thông không bị chia ra như một thiết bị trung tâm giả mà được kết nối điểm - điểm.

→ Đắt hơn

## Bộ lặp và Cầu nối

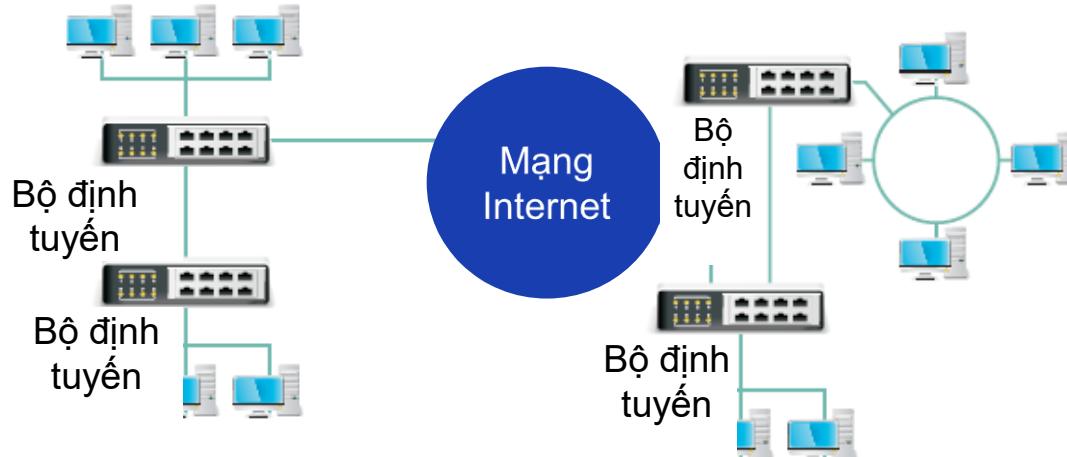
Bộ lặp là một thiết bị mở rộng khoảng cách truyền dữ liệu của một mạng, và cầu nối là một thiết bị giúp kết nối hai hoặc nhiều mạng LAN với nhau để tạo thành một mạng.

- ▶ Bộ lặp
  - Bộ lặp tái tạo và truyền các tín hiệu yếu trong quá trình truyền tín hiệu
- ▶ Cầu nối
  - Cầu nối tìm kiếm địa chỉ của một khung (frame) lưu thông trong mạng, chấp nhận frame nếu đó là frame của một địa chỉ trong cùng một mạng LAN, và gửi nó đến mạng LAN phù hợp thông qua cầu nối nếu cần gửi đến một mạng LAN đã kết nối.
  - Giảm lưu lượng trên toàn mạng.

## Bộ định tuyến

| Bộ định tuyến là một thiết bị kết nối các mạng như mạng LAN, MAN, và WAN với nhau.

- ▶ Bộ định tuyến định tuyến các gói dữ liệu theo địa chỉ logic của chúng (địa chỉ IP)
- ▶ Bộ định tuyến có trách nhiệm kết nối mạng
- ▶ định tuyến
- ▶ Bộ định tuyến tìm kiếm các địa chỉ Internet đích của gói dữ liệu được truyền đến bộ định tuyến và xác định lộ trình tiếp theo.
- ▶ Gói dữ liệu được chuyển tiếp giữa hai mạng
- ▶ trả lên (Tầng 3 OSI)

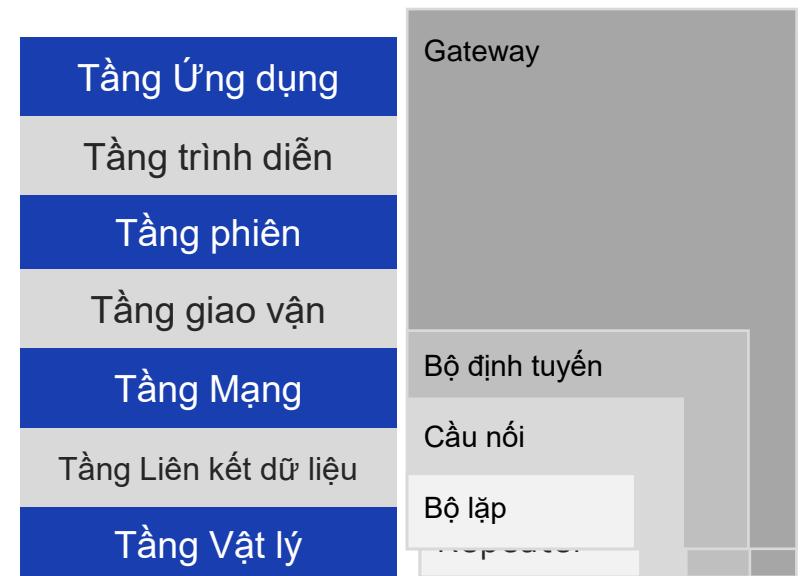
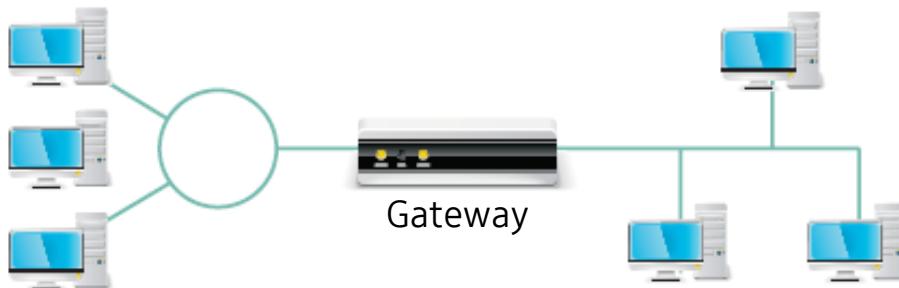


## Gateway

| Gateway là một điểm kết nối của một mạng và có chức năng như một cổng vào hoặc cổng ra của một mạng khác.

▶ Gateway

- Vai trò của Bộ chuyển đổi giao thức
  - Kết nối hai mạng có giao thức khác nhau
- Ngày nay, gateway và bộ định tuyến được sử dụng đồng thời



Mô hình OSI và các Thiết bị đi kèm

Bài 1.

# Tìm hiểu về Mạng

- | 1.1. Tổng quan về Mạng
- | 1.3. Thiết bị Mạng
- | **1.2. Chuyển đổi Mạng**
- | 1.4. Phương thức truyền dữ liệu của Mạng
- | 1.5. Phân loại và Tính năng của giao tiếp có dây
- | 1.6. Cấu trúc liên kết mạng
- | 1.7. Sử dụng Mạng

## Chuyển mạch

| Sau khi xác định bên gửi và bên nhận, một tuyến thích hợp trong số các đường truyền sẽ được thiết lập.

▶ Chuyển mạch

- Mạch: Tập hợp các đường truyền (các kênh giao tiếp)
- Sau khi thiết lập đường truyền, các máy tính khác sẽ không thể sử dụng đường truyền này dù không truyền dữ liệu nào, cho đến khi đường truyền này được giải phóng. Nhược điểm về mặt tối ưu hóa đường truyền
- Sau khi thiết lập đường truyền và bắt đầu truyền dữ liệu thì không cần phải tìm lại tuyến đó nữa.
  - Lý tưởng để truyền dữ liệu theo thời gian thực, ví dụ như giọng nói
  - Chỉ được sử dụng để truyền khối lượng dữ liệu lớn ở tốc độ cao
- Ví dụ điển hình như mạng điện thoại

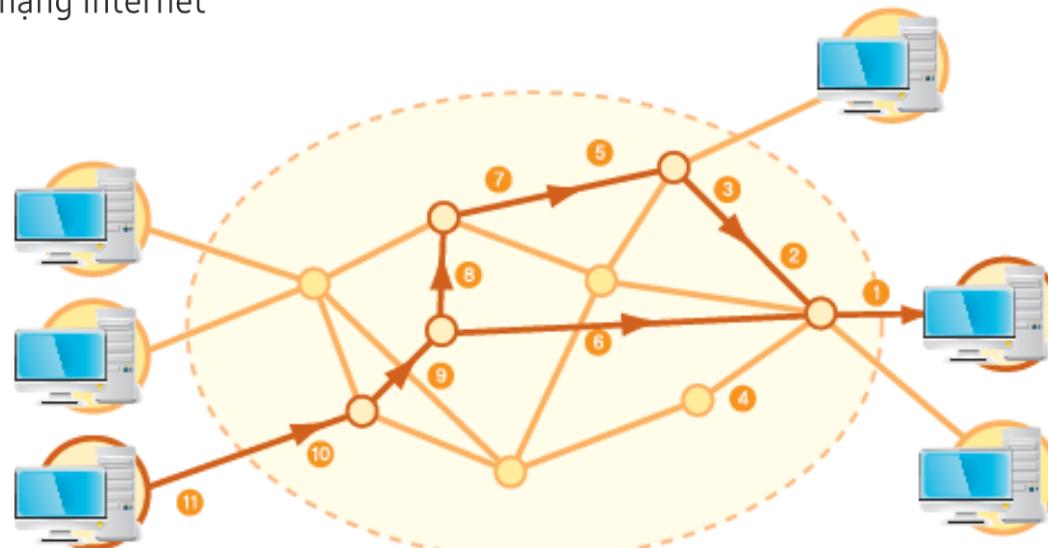


## Chuyển gói

| Không thiết lập trước tuyến cố định, truyền dữ liệu độc lập trong gói dữ liệu

▶ Chuyển gói

- Mỗi gói dữ liệu có đường truyền dữ liệu khác nhau dựa trên tình trạng lưu lượng của mạng.
- Không thiết lập một tuyến cố định
  - Sử dụng đường truyền hiệu quả hơn vì nhiều gói dữ liệu đến các đích khác nhau cùng dùng chung một tuyến
- Ví dụ điển hình là mạng Internet



Bài 1.

# Tìm hiểu về Mạng

- | 1.1. Tổng quan về Mạng
- | 1.2. Thiết bị Mạng
- | 1.3. Chuyển đổi Mạng
- | **1.4. Phương thức truyền dữ liệu của Mạng**
- | 1.5. Phân loại và Tính năng của giao tiếp có dây
- | 1.6. Cấu trúc liên kết mạng
- | 1.7. Sử dụng Mạng

### Đơn công, Bán song công, Song công toàn phần

| Dựa theo hướng và tính đồng thời của luồng dữ liệu, phương thức giao tiếp của mạng được phân chia như sau.

▶ đơn công

- Kiểu giao tiếp chỉ được thực hiện theo một hướng.
- Một thiết bị chỉ có chức năng truyền thông tin và thiết bị còn lại chỉ có chức năng nhận thông tin.

▶ bán song công

- Hai thiết bị đều có thể giao tiếp theo hai hướng, nhưng không đồng thời.
- Sẽ xảy ra xung đột nếu cả hai thiết bị đều truyền dữ liệu cùng một lúc.
  - Để tránh xung đột, trước khi giao tiếp, đầu tiên hãy kiểm tra xem có thể truyền dữ liệu được không.

▶ song công toàn phần

- Hai thiết bị có thể đồng thời truyền dữ liệu theo cả hai hướng
  - Nhận và gửi dữ liệu đồng thời.

### Truyền dữ liệu analog và digital

| Dữ liệu analog là dữ liệu có giá trị thay đổi liên tục theo thời gian, và dữ liệu digital có các giá trị riêng biệt và có thể xử lý bởi số nguyên của các giá trị bất kỳ.

- ▶ Truyền dữ liệu analog
  - Dữ liệu analog là thông tin có độ lớn thay đổi liên tục theo thời gian
    - Âm thanh, áp suất, nhiệt độ
  - Các tín hiệu analog sẽ yếu đi khi khoảng cách truyền dữ liệu tăng lên
    - Để tín hiệu mạnh trở lại, cần phải sử dụng bộ khuếch đại ở các vị trí nhất định
- ▶ Truyền dữ liệu digital
  - Truyền các tín hiệu digital 0 và 1
  - Sử dụng bộ lặp để giải quyết các hạn chế về khoảng cách

### Truyền tín hiệu nối tiếp và song song

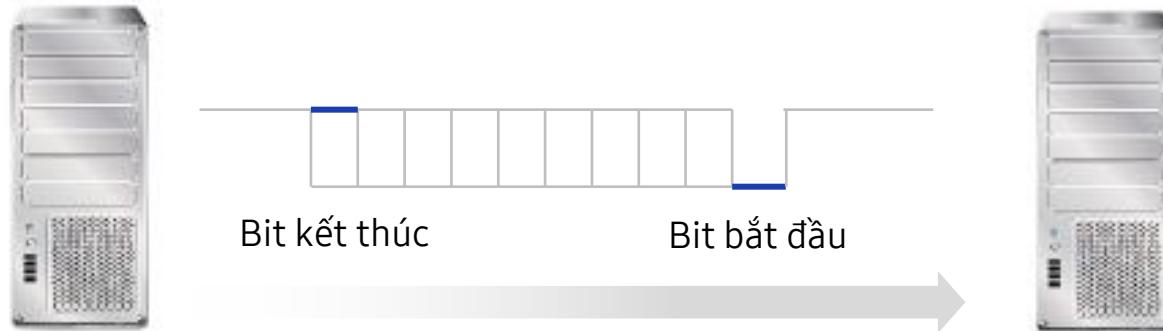
- | Truyền tín hiệu nối tiếp là phương thức truyền từng bit một theo thứ tự qua một đường truyền giao tiếp.
- | Truyền dữ liệu song song truyền nhiều bit cùng một lúc thông qua nhiều đường truyền
  - ▶ Truyền dữ liệu nối tiếp
    - Tốc độ truyền chậm vì phải truyền từng bit một, tuy nhiên chi phí đường truyền giao tiếp cũng rất thấp
    - Chủ yếu được sử dụng trong truyền tải thông tin khoảng cách xa và cho các điểm giao tiếp như RS-232C và RS-423
  - ▶ Truyền dữ liệu song song
    - So với truyền nối tiếp, truyền song song truyền dữ liệu nhanh hơn nhưng lại cần chi phí cao hơn để xây dựng đường truyền giao tiếp.
    - Ít được sử dụng để kết nối giữa các thiết bị đầu cuối và thường được dùng để kết nối ở tầm gần từ một máy tính đến một thiết bị ngoại vi (như máy in).

## Truyền dữ liệu đồng bộ và bất đồng bộ

| Truyền dữ liệu bất đồng bộ là phương pháp truyền dữ liệu theo từng ký tự khi dữ liệu được gửi từ bên truyền đến bên nhận mà không cần làm việc cùng nhịp.

### ▶ Truyền dữ liệu bất đồng bộ

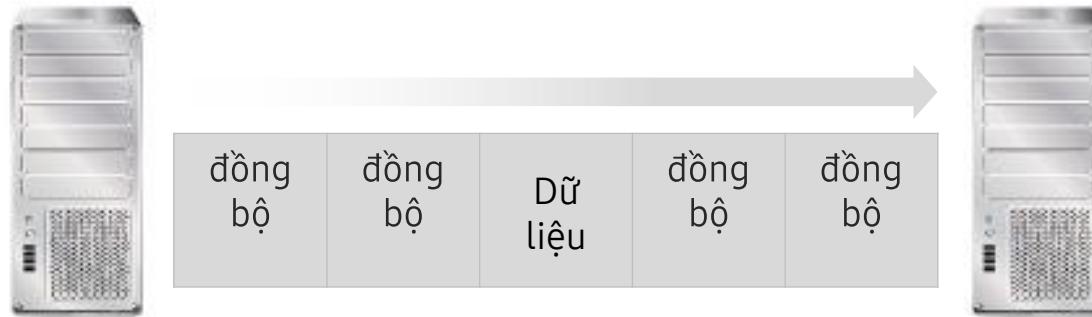
- Trong phương pháp truyền dữ liệu bất đồng bộ, tín hiệu dữ liệu được phân biệt bằng bit bắt đầu, dữ liệu và bit kết thúc
  - Bên truyền và bên nhận sử dụng các xung đồng hồ hệ thống độc lập nhưng chúng đều đồng bộ ở các bit bắt đầu và kết thúc để nhận diện dữ liệu.
- Truyền dữ liệu bất đồng bộ có chi phí truyền tải thấp vì cấu trúc của các thiết bị truy cập đơn giản.
- Dữ liệu càng dài sẽ càng làm tăng nguy cơ xuất hiện các lỗi do cộng dồn chênh lệch xung đồng hồ hệ thống.



| Truyền dữ liệu đồng bộ khôi phục từ các lỗi bằng cách truyền lại dữ liệu y hệt nếu có lỗi xảy ra trong quá trình truyền dữ liệu.

▶ Truyền dữ liệu đồng bộ

- Để cải thiện hiệu suất truyền dữ liệu, bên truyền và bên nhận truyền dữ liệu theo một định dạng dữ liệu nhất định mà hai bên đã thống nhất.
  - Bên gửi sẽ chèn và truyền các thông tin điều khiển trước và sau khi khối dữ liệu được truyền đi.
- Truyền dữ liệu đồng bộ khôi phục từ các lỗi bằng cách truyền lại dữ liệu y hệt nếu có lỗi xảy ra trong quá trình truyền dữ liệu.



Bài 1.

# Tìm hiểu về Mạng

- | 1.1. Tổng quan về Mạng
- | 1.2. Thiết bị Mạng
- | 1.3. Chuyển đổi Mạng
- | 1.4. Phương thức truyền dữ liệu của Mạng
- | 1.5. Phân loại và Tính năng của giao tiếp có dây
- | 1.6. Cấu trúc liên kết mạng
- | 1.7. Sử dụng Mạng

## Mạng LAN

| Mạng LAN là một mạng kết nối và cấu hình các thiết bị trong phạm vi tương đối gần.

▶ Mạng LAN (mạng cục bộ)

- Một mạng giúp kết nối và cấu hình máy tính, máy in và các thiết bị mạng khác trong phạm vi gần
  - Một khu vực do một tổ chức quản lý
  - Một mạng được cài đặt trong khuôn viên trường, tòa nhà công ty, nhà máy, v.v.
- Ban đầu, mạng Ethernet với tốc độ từ 10 đến 100 Mbps được sử dụng
- Gần đây, để có tốc độ cao hơn, nhiều mạng như mạng ethernet Gigabit, ATM, FDDI và mạng LAN không dây đã được sử dụng



### I Mạng LAN có chất lượng giao tiếp tương đối tốt và tốc độ nhanh

#### ▶ Tính năng

- Truyền dữ liệu hiếm khi bị chậm. Sử dụng và quản lý mạch giao tiếp chất lượng tốt
  - Chất lượng giao tiếp tốt. Ít xảy ra lỗi khi truyền. Tốc độ truyền nhanh
- Các thiết bị như máy tính và máy in có thể được kết nối và sử dụng dễ dàng. Dễ mở rộng.

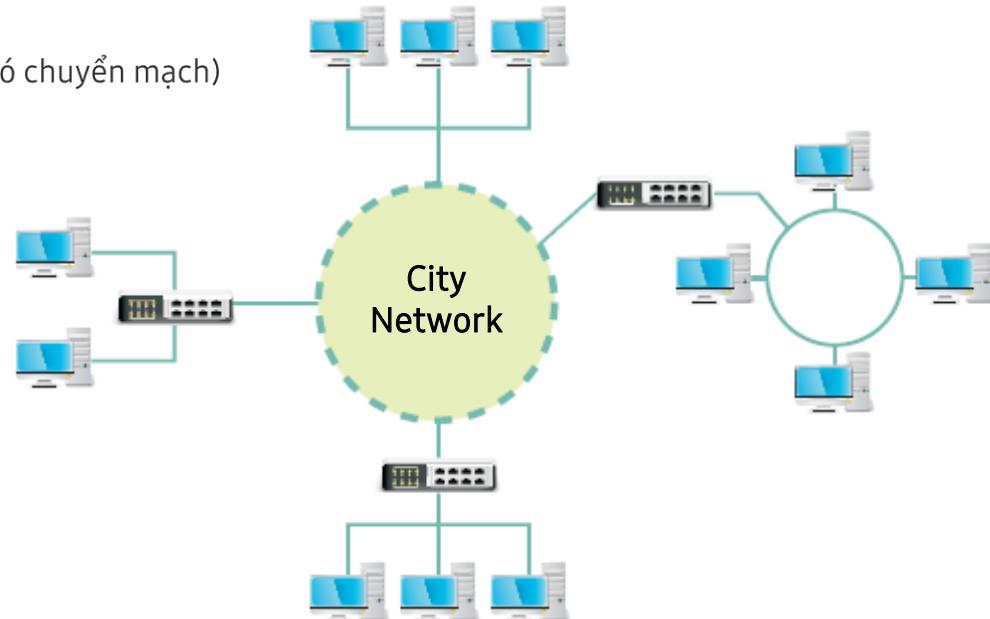
#### ▶ Chuẩn mạng LAN

- Ethernet
  - Phát triển bởi Xerox vào năm 1976
  - tốc độ 10 Mbps, thuật toán CSMA/CD
- Tốc độ mạng Ethernet nhanh
  - 100Mbps, 100BASE-T
- Ethernet Gigabit
  - 1 Gbps
- FDDI (Giao diện dữ liệu phân tán sợi quang)
  - Mạng này thường dùng sợi quang như một phương tiện truyền dữ liệu để lắp đặt mạng LAN tốc độ cao
  - Mạng này thường được dùng ở các mạng xương sống cần băng thông dài và tốc độ truyền tải nhanh

## Mạng MAN

Mạng MAN (Mạng đô thị) là một mạng rộng hơn mạng LAN một chút

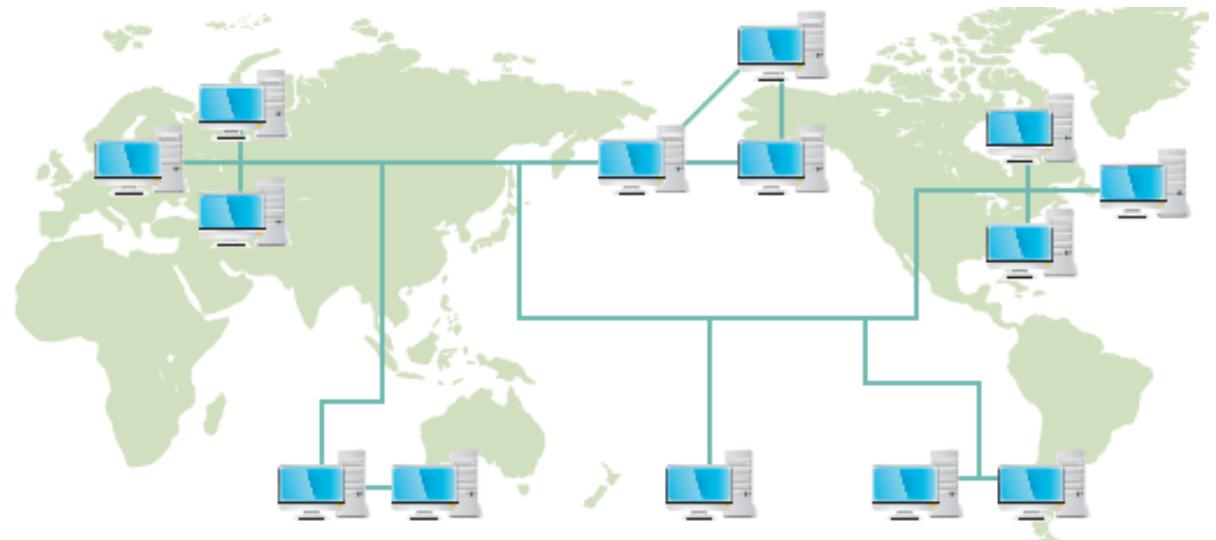
- ▶ MAN (Mạng đô thị)
  - Một mạng giúp kết nối trong phạm vi một ngôi làng hoặc một thành phố với mạng LAN mở rộng và được kết nối
  - Các nhà mạng thông thường cung cấp và quản lý hoạt động này
- ▶ Ngành điện thoại qua mạng MAN
  - Cung cấp SMDS (dịch vụ số liệu nhiều megabit có chuyển mạch)



## Mạng WAN

WAN là một mạng kết nối các quốc gia với nhau

- ▶ WAN (Mạng diện rộng)
  - Một mạng có phạm vi rất rộng
    - Ví dụ về mạng WAN là mạng Internet kết nối toàn thế giới
  - Nói chung, mạng WAN có phạm vi kết nối rộng hơn mạng LAN
    - Cần rất nhiều tiền để xây dựng mạng này.
  - Tốc độ chậm hơn mạng LAN



Bài 1.

# Tìm hiểu về Mạng

- | 1.1. Tổng quan về Mạng
- | 1.2. Thiết bị Mạng
- | 1.3. Chuyển đổi Mạng
- | 1.4. Phương thức truyền dữ liệu của Mạng
- | 1.5. Phân loại mạng dựa trên kích cỡ
- | **1.6. Cấu trúc liên kết mạng**
- | 1.7. Sử dụng Mạng

## Cấu trúc liên kết Mạng

| Cấu trúc liên kết mạng là sự sắp xếp vật lý hoặc logic của các nút và đường dẫn kết nối tới mạng.

▶ Cấu trúc liên kết mạng

- Nút

- Một thiết bị giao tiếp có địa chỉ được kết nối với một mạng máy tính.
  - Nó được gọi là một máy chủ.
- Mỗi máy tính, bộ định tuyến, máy in, v.v., là một nút.

▶ Các dạng

- Dạng tuyến
- Dạng sao
- Dạng vòng
- Dạng cây
- Dạng lưới

## Dạng tuyệ́n

| Trong dạng tuyệ́n, tất cả nút mạng đều được kết nối tới trên một đường dây trên đường truyền, gọi là một tuyệ́n (bus).

- ▶ Dạng tuyệ́n

- Cấu trúc này khá đơn giản và chi phí cho dây cáp rất nhỏ.
- Nếu một nút truyền dữ liệu khi một nút khác cũng đang truyền dữ liệu thì sẽ xảy ra xung đột.
  - Nếu xảy ra xung đột, thì sau đó sẽ phải truyền lại dữ liệu. Nếu có nhiều nút hoặc lưu lượng dữ liệu cao, xung đột sẽ xảy ra thường xuyên và hiệu suất của mạng sẽ giảm.



## Dạng sao

- ▶ Các nút khác kết nối đến một thiết bị trung tâm bằng liên kết điểm - điểm
- ▶ Khả năng xử lý và độ tin cậy của mạng con phụ thuộc vào thiết bị trung tâm này
  - ▶ Các thiết bị trung tâm thông minh có thể kiểm soát lượng giao tiếp trên mạng hoặc ngăn xung đột
- ▶ Cấu trúc tập trung giúp phát hiện lỗi đơn giản, bảo trì dễ dàng và kiểm soát đường truyền đơn giản hơn

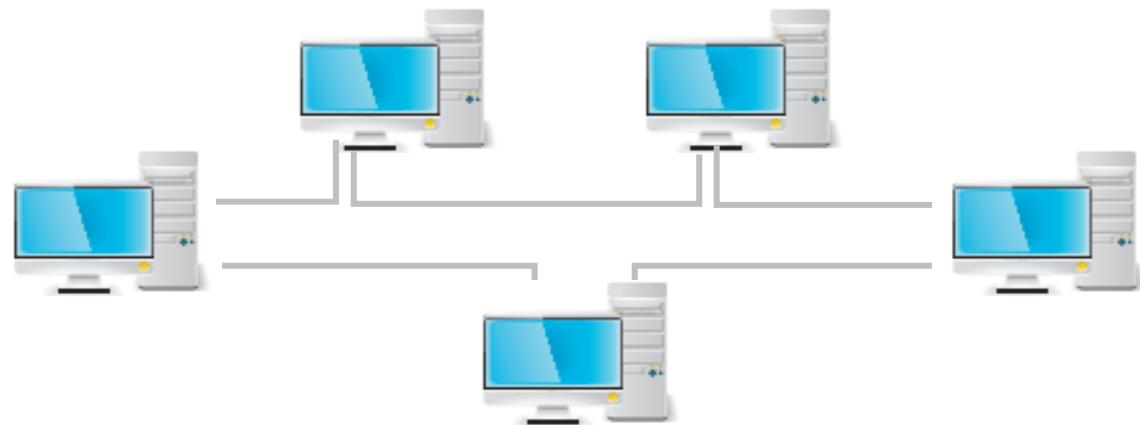


## Dạng vòng

| Trong liên kết vòng, các nút của mạng được kết nối với nhau trong một vòng tròn.

▶ Dạng vòng

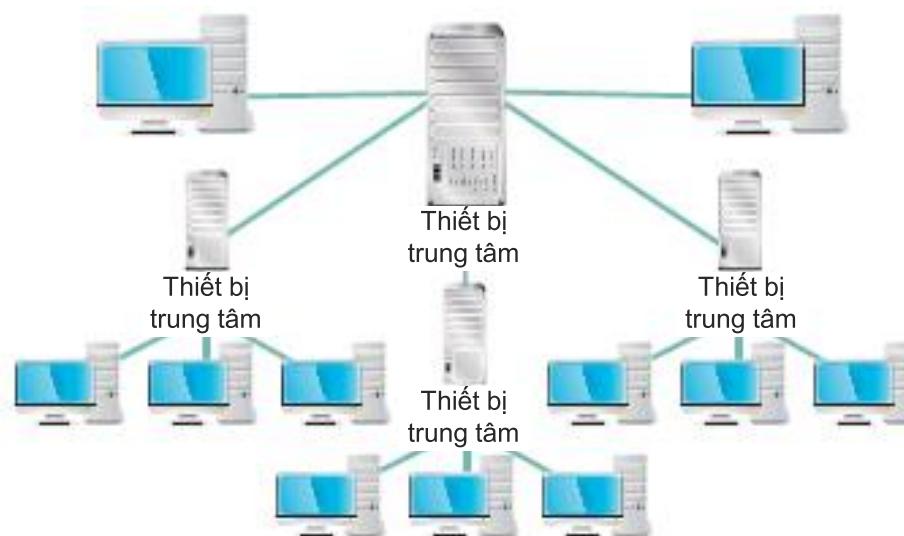
- Các dữ liệu được truyền từ một nút sẽ được tiếp tục truyền đi theo vòng tròn.
- Các nút, ngoài các nút nhận, luôn phát tín hiệu và truyền dữ liệu đến nút tiếp theo.
  - Quá trình phát lại có thể làm giảm lỗi trong quá trình truyền dữ liệu.
- Khi một lỗi xảy ra, việc tìm máy chủ bị lỗi sẽ rất dễ dàng.
- Việc số lượng các nút tăng lên không có ảnh hưởng lớn đến hiệu suất của mạng.
- Kiểu
  - Vòng đơn
  - Vòng đôi



## Dạng cây

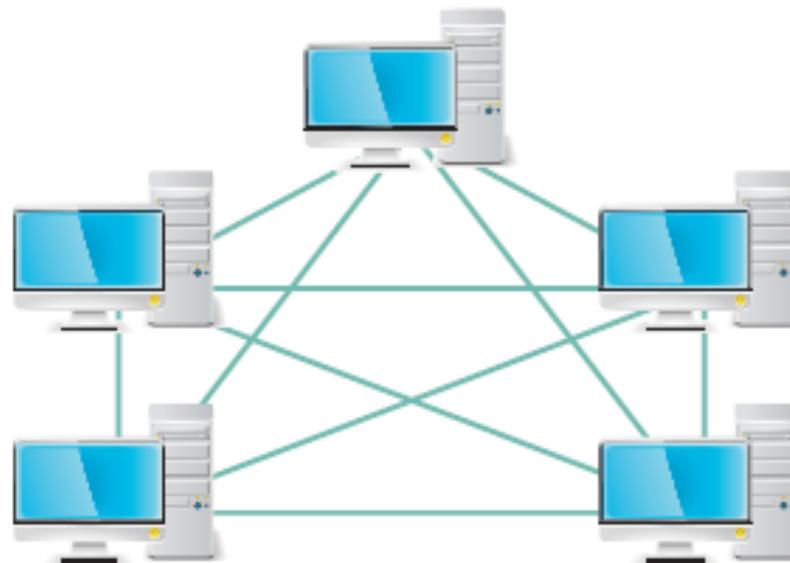
| Liên kết cây là một cấu trúc mà ở đó các nút được kết nối theo hình cây, và cấu trúc lưới là cấu trúc mà ở đó các nút đều được kết nối điểm - điểm với nhau trực tiếp mà không cần một nút điều khiển trung tâm.

- ▶ Dạng cây
  - Nút trên cùng của cấu trúc cây là nơi thiết bị trung tâm được đặt và kiểm soát các nút khác ở dưới.
    - Vì việc kiểm soát đơn giản nên kiểu mạng này rất dễ quản lý và mở rộng
  - ▶ Lưu lượng tập trung có thể gây tắc nghẽn, nếu điểm trung tâm bị lỗi, toàn bộ mạng sẽ bị lỗi theo



## Dạng lưới

- | Hiệu quả mạng tốt vì không cần chuyển tiếp thiết bị nào trong quá trình truyền dữ liệu.
  - ▶ thậm chí nếu đường truyền bị lỗi, dữ liệu có thể được truyền đi thông qua các con đường khác.
- | Được sử dụng trong các mạng có chi phí cao nhưng có độ tin cậy tốt vì mạng này khá phức tạp và cần nhiều đường truyền



Bài 1.

# Tìm hiểu về Mạng

- | 1.1. Tổng quan về Mạng
- | 1.2. Thiết bị Mạng
- | 1.3. Chuyển đổi Mạng
- | 1.4. Phương thức truyền dữ liệu của Mạng
- | 1.5. Phân loại mạng dựa trên kích cỡ
- | 1.6. Cấu trúc liên kết mạng
- | **1.7. Sử dụng Mạng**

## Địa chỉ IP

| Tất cả máy tính và thiết bị mạng kết nối mạng Internet phải có một địa chỉ IP riêng

▶ Địa chỉ IP

- Địa chỉ IP duy nhất

- Là địa chỉ IP của một máy tính mà không có máy tính nối mạng Internet nào khác trên thế giới có cùng địa chỉ IP đó.
- Tất cả bốn chữ số  $8 * 4 = 32$  bit

- Cấu trúc

- Trong bốn chữ số, hai hoặc ba chữ số đầu tiên là địa chỉ của mạng.
- Số còn lại là địa chỉ của máy tính trong mạng đó.
- Chẳng hạn với địa chỉ 147.46.X.X, thì địa chỉ của mạng là 147.46.0.0 và mạng này có thể chứa khoảng 216 máy tính.
- Nếu địa chỉ của một công ty là 203.246.245.X, thì địa chỉ mạng của công ty đó sẽ là 203.246.245.0 và mạng này có thể chứa khoảng 28 máy tính.

## Mạng con

| Mạng con là một đơn vị mạng nhỏ trong một mạng

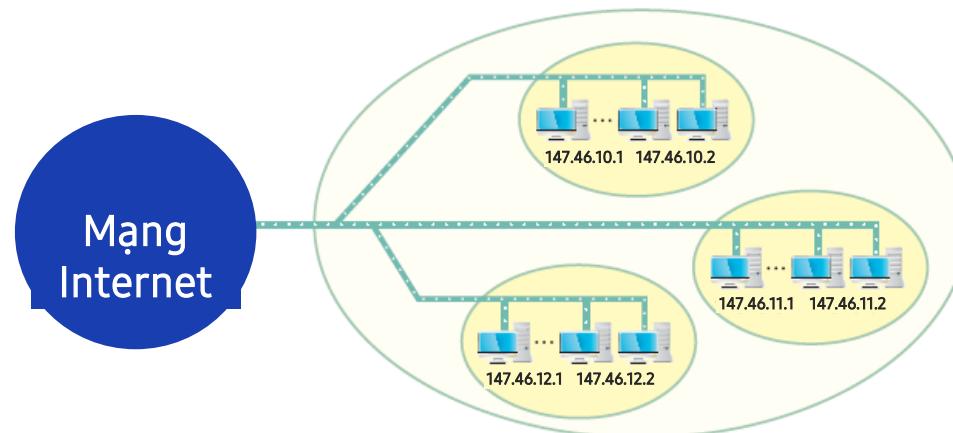
▶ Mạng con

- Mạng có tên 147.46.X.X

- 147.46.10.X, 147.46.11.X... Giống như địa chỉ ở bên trái, nó có thể được chia thành nhiều mạng nhỏ hơn, gọi là mạng con.

▶ Mặt nạ mạng con

- Nếu đặt mặt nạ mạng con là 255.255.255.0, khi một gói dữ liệu đi vào mạng là 11111111 11111111 11111111 00000000, nó sẽ đi vào mạng con và chỉ định tuyến phần được đặt là 1.



## Gateway mặc định và máy chủ DNS

| Các cài đặt Gateway mặc định và máy chủ DNS là các thông tin quan trọng khi cấu hình IP

▶ Gateway mặc định

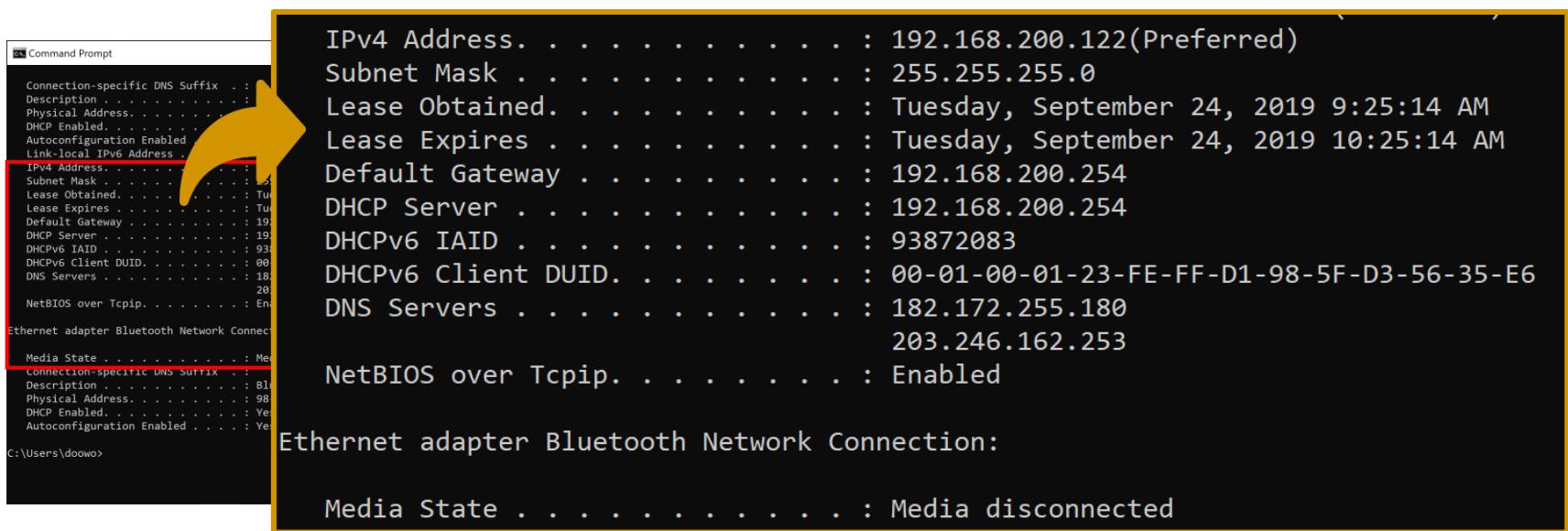
- Một điểm kết nối mà các gói dữ liệu cần đi qua khi một gói dữ liệu từ một máy tính riêng đi qua một mạng bên ngoài hoặc một gói dữ liệu từ mạng bên ngoài được gửi vào mạng riêng hoặc mạng con.
  - Tại văn phòng và trường học, địa chỉ gateway sẽ dựa trên vị trí hiện tại của máy tính.

▶ Máy chủ DNS

- Máy chủ này chuyển đổi tên miền thành các địa chỉ IP.
  - Máy chủ có chức năng chuyển tên miền ví dụ như www.google.com thành các địa chỉ IP ví dụ như 172.217.24.68
- Tên miền
  - Địa chỉ IP sử dụng các tên miền bằng các ký tự chữ cái, ví dụ như www.naver.com.
  - Lý do là vì các địa chỉ IP dạng số khá khó nhớ và khó sử dụng.

## Kiểm tra trạng thái mạng

- I Khi gõ “cmd” trong [Start]-[Run], một cửa sổ sẽ hiện ra để bạn có thể kiểm tra trạng thái và cài đặt với các lệnh ipconfig, nslookup, tracert, v.v.
  - ▶ ipconfig
  - ▶ Liệt kê trạng thái kết nối của mạng hiện tại
  - ▶ Địa chỉ vật lý, địa chỉ IP, máy chủ DHCP, v.v.



```
Administrator: Command Prompt
C:\> ipconfig /all

Connection-specific DNS Suffix . . . . . : 
Description . . . . . : Intel PRO/100 MT Desktop Adapter
Physical Address . . . . . : 00:0C:29:4D:9A:9B
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::4c29%10
    IPv4 Address . . . . . : 192.168.200.122(PREFERRED)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained . . . . . : Tuesday, September 24, 2019 9:25:14 AM
    Lease Expires . . . . . : Tuesday, September 24, 2019 10:25:14 AM
    Default Gateway . . . . . : 192.168.200.254
    DHCP Server . . . . . : 192.168.200.254
    DHCPv6 IAID . . . . . : 93872083
    DHCPv6 Client DUID . . . . . : 00-01-00-01-23-FE-FF-D1-98-5F-D3-56-35-E6
    DNS Servers . . . . . : 182.172.255.180
                                203.246.162.253
    NetBIOS over Tcpip. . . . . : Enabled

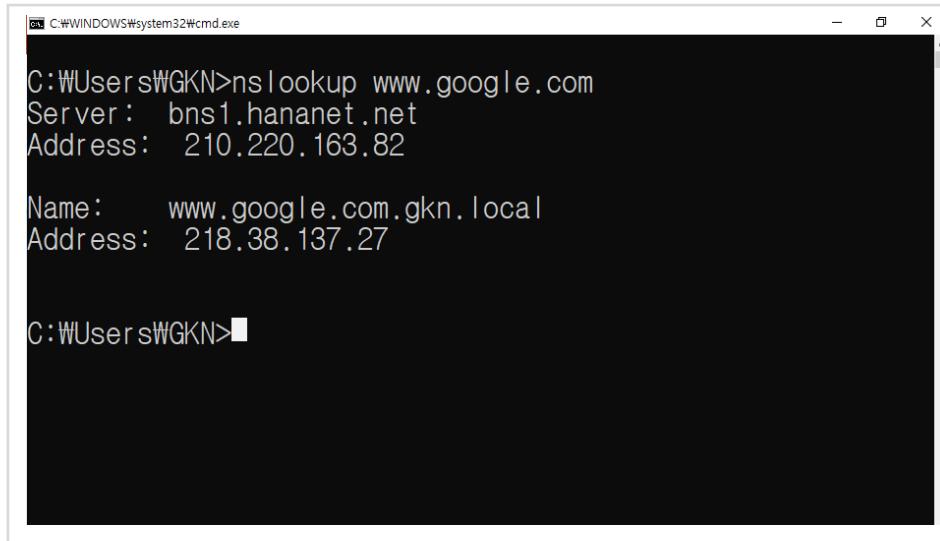
Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . : 
    Description . . . . . : Intel PRO/100 MT Desktop Adapter
    Physical Address . . . . . : 00:0C:29:4D:9A:9B
    DHCP Enabled . . . . . : Yes
    Autoconfiguration Enabled . . . . . : Yes

C:\Users\doowo>
```

### | Chuyển tên miền thành địa chỉ IP

- ▶ nslookup
  - Chuyển tên miền bằng chữ cái sang địa chỉ IP



A screenshot of a Windows Command Prompt window titled "cmd C:\WINDOWS\system32\cmd.exe". The window shows the output of the nslookup command for the domain "www.google.com". The output includes the server information ("Server: bns1.hananet.net") and the address ("Address: 210.220.163.82"). It also shows the fully qualified domain name ("Name: www.google.com.gkn.local") and its corresponding IP address ("Address: 218.38.137.27"). The prompt "C:\Users\WGKN>" is visible at the bottom.

```
C:\Users\WGKN>nslookup www.google.com
Server:  bns1.hananet.net
Address: 210.220.163.82

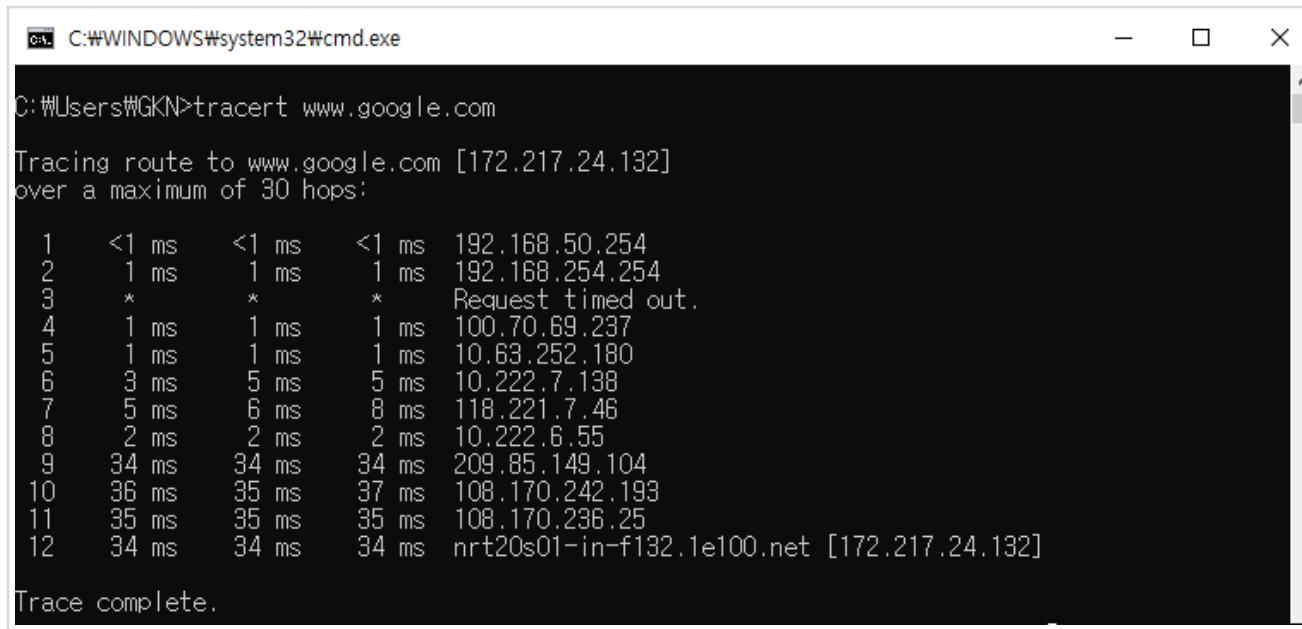
Name:   www.google.com.gkn.local
Address: 218.38.137.27

C:\Users\WGKN>
```

### | Lệnh hiển thị đường dẫn mạng

#### ▶ tracert

- Định tuyến đến một điểm đích cho trước (www.google.com)
- Không tác dụng với các đường dẫn thứ 3, 9 và 10, v.v.
- Ngày nay, vì lý do bảo mật, các dấu vết này thường bị chặn trên mạng của bạn, do đó, không phải lúc nào cũng có thể định tuyến được.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\GKN>tracert www.google.com

Tracing route to www.google.com [172.217.24.132]
over a maximum of 30 hops:
1 <1 ms <1 ms <1 ms 192.168.50.254
2 1 ms 1 ms 1 ms 192.168.254.254
3 * * * Request timed out.
4 1 ms 1 ms 1 ms 100.70.69.237
5 1 ms 1 ms 1 ms 10.63.252.180
6 3 ms 5 ms 5 ms 10.222.7.138
7 5 ms 6 ms 8 ms 118.221.7.46
8 2 ms 2 ms 2 ms 10.222.6.55
9 34 ms 34 ms 34 ms 209.85.149.104
10 36 ms 35 ms 37 ms 108.170.242.193
11 35 ms 35 ms 35 ms 108.170.236.25
12 34 ms 34 ms 34 ms nrt20s01-in-f132.1e100.net [172.217.24.132]

Trace complete.
```

Bài 2.

# Giao thức truyền thông

- | **2.1. Giao thức là gì?**
- | 2.2. Mô hình OSI
- | 2.3. Giao thức truyền thông TCP/IP
- | 2.4. HTTP
- | 2.5. MQTT
- | 2.6. CoAP
- | 2.7. WebSocket

## Giao thức

- | Vì có nhiều kiểu nền tảng và hệ thống trong một mạng, nên các nền tảng và hệ thống này sử dụng giao thức quy ước chung để giao tiếp với nhau.
- | Các giao thức mạng chung:
  - ▶ Giao thức điều khiển truyền vận (TCP), Giao thức Internet (IP), Giao thức biên dịch địa chỉ mạng (ARP), Giao thức cấu hình máy chủ (DCHP)
  - ▶ Một tập hợp logic của các giao thức hoạt động như một chồng giao thức.
- | Cách tốt nhất để hiểu về giao thức:
  - ▶ Hãy nghĩ nó giống như các quy định điều chỉnh ngôn ngữ của con người.
- | Ex (động từ, biểu thức kiểm toán, phương pháp định tuyến, phương pháp kết nối ban đầu, v.v.,)
- | Tùy theo chức năng, các giao thức có thể dễ hiểu hoặc khó hiểu.

| Hầu hết các giao thức đều xét đến các vấn đề sau:

- ▶ Khởi tạo kết nối
  - Máy khách hay máy chủ khởi tạo kết nối, cần trao đổi thông tin gì trước khi giao tiếp?
- ▶ Tư vấn các đặc điểm kết nối
  - Làm thế nào để mã hóa giao tiếp của giao thức, và làm thế nào để truyền các khóa mã khóa giữa các máy chủ giao tiếp?
- ▶ Định dạng dữ liệu
  - Làm thế nào để sắp xếp dữ liệu trong gói tin, và dữ liệu được xử lý bằng kiểu sắp xếp nào ở thiết bị nhận?
- ▶ Phát hiện lỗi và sửa lỗi
  - Chuyện gì sẽ xảy ra nếu một gói tin mất quá nhiều thời gian để đến được đích? Máy khách sẽ khôi phục như thế nào nếu không thiết lập được kết nối giữa máy khách và máy chủ trong một khoảng thời gian ngắn?
- ▶ Hủy bỏ kết nối
  - Làm thế nào để máy chủ thông báo máy chủ còn lại rằng kết nối đã bị hủy, và thông tin nào phải được gửi cuối cùng để hủy kết nối đúng cách?

Bài 2.

# Giao thức truyền thông

- | 2.1. Giao thức là gì
- | 2.2. Mô hình OSI
- | 2.3. Giao thức truyền thông TCP/IP
- | 2.4. HTTP
- | 2.5. MQTT
- | 2.6. CoAP
- | 2.7. WebSocket

## Mô hình OSI?

| Mô hình tiêu chuẩn quốc tế do Tổ chức Tiêu chuẩn Quốc tế (ISO) tạo ra

▶ Mô hình Liên kết Hệ thống mở (OSI)

- Năm 1978, Tổ chức Tiêu chuẩn Quốc tế (ISO) đã thiết lập mô hình tiêu chuẩn quốc tế OSI để cho phép hai hệ thống giao tiếp mà không cần quan tâm đến các cấu trúc ngầm.
  - Đây là một mô hình giúp tìm hiểu và thiết kế một cấu trúc mạng có thể tương tác an toàn.
- ▶ 7 tầng
- Vật lý (tầng 1), Liên kết dữ liệu (tầng 2), Mạng (tầng 3), Giao vận (tầng 4), Phiên (tầng 5), Trình diễn (tầng 6), Ứng dụng (tầng 7)
    - Hai tầng này độc lập với nhau.
    - Thay đổi trong một tầng không ảnh hưởng tới các tầng khác.
  - Các thiết bị mạng có thể giao tiếp bình thường nếu chúng đáp ứng được một số tầng trong hai tầng này theo chức năng và chuẩn hóa chúng.

Tầng 7	Ứng dụng
Tầng 6	Trình diễn
Tầng 5	Phiên
Tầng 4	Giao vận
Tầng 3	Mạng
Tầng 2	Liên kết dữ liệu
Tầng 1	Vật lý

### Mô hình OSI 7 Tầng

- ▶ Từ tầng ứng dụng đến tầng vật lý
  - Từ tầng ứng dụng đến tầng vật lý, nó đã đi qua bảy tầng của mô hình OSI để đến tầng dưới cùng, và hệ thống được nhận sẽ chuyển dữ liệu theo thứ tự từ tầng vật lý lên đến tầng ứng dụng.
- ▶ Tính độc lập của các Dịch vụ Giao thức
  - Nếu một tầng giao thức cung cấp một dịch vụ nào đó, các tầng khác sẽ không cung cấp dịch vụ đó.
- ▶ Giao tiếp giữa hai tầng liền kề nhau
  - Mỗi tầng giao tiếp với tầng ngay bên trên và ngay bên dưới nó.

(Truyền sau khi truyền dữ liệu từ Tầng 2, Tầng 1 và Tầng 3)



### Mô hình OSI 7 Tầng

- ▶ Các giao thức này dựa trên mô hình tham chiếu OSI đạt chuẩn công nghiệp và được phân loại theo chức năng của chúng.
- ▶ Tầng ứng dụng (tầng 7)
  - Tầng ứng dụng ở trên cùng mô hình OSI cung cấp cho người dùng quyền tiếp cận đến các tài nguyên mạng.
  - Cung cấp tất cả các giao diện cơ bản cho môi trường mạng.
  - Thường là tầng duy nhất mà người dùng có thể nhìn thấy.
- ▶ Tầng trình diễn (6 tầng)
  - Chuyển đổi dữ liệu đi qua tầng ứng dụng sang một định dạng có thể đọc được.
  - Mã hóa và giải mã các dữ liệu được gửi hoặc nhận từ tầng ứng dụng.
  - Tầng này có cách mã hóa và giải mã để sử dụng dữ liệu an toàn.



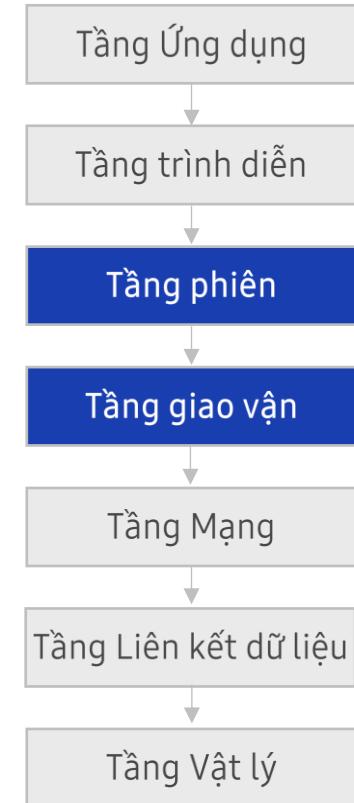
### I Mô hình OSI 7 Tầng

#### ▶ Tầng phiên (tầng 5)

- Quản lý các phiên làm việc hoặc đối thoại giữa hai máy tính.
- Kết nối, quản lý và ngắt tất cả các thiết bị giao tiếp.
- Có khả năng ngăn chặn tình trạng mất kết nối đột ngột và ngắt kết nối đúng cách giữa các máy chủ.
- Tầng này có nhiệm vụ kiểm tra xem các tác vụ giao tiếp là đơn phương hoặc song phương.

#### ▶ Tầng giao vận (tầng 4)

- Gửi dữ liệu tin cậy đến tầng dưới.
- Bao gồm định dạng, điều khiển lưu lượng, phân đoạn, hợp đoạn và quản lý lỗi.
- Tầng này đảm bảo không có lỗi trong kết nối điểm - điểm.
- Công việc truyền dữ liệu an toàn và tin cậy rất phức tạp nhưng cũng rất quan trọng.
- Là tầng nơi tường lửa và các máy chủ proxy hoạt động



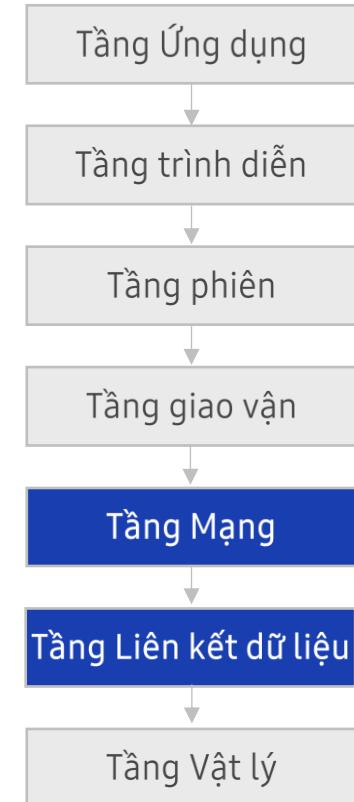
### I Mô hình OSI 7 Tầng

#### ▶ Tầng Mạng (tầng 3)

- Một trong những tầng OSI phức tạp nhất, chịu trách nhiệm định tuyến giữa các mạng vật lý.
- Tầng này chịu trách nhiệm quản lý các địa chỉ logic, phân đoạn gói dữ liệu, nhận diện giao thức, và phát hiện lỗi trong các máy chủ mạng.

#### ▶ Tầng Liên kết dữ liệu (tầng 2)

- Chịu trách nhiệm truyền dữ liệu giữa các mạng vật lý.
- Cung cấp lược đồ địa chỉ và kiểm tra lỗi
- Cầu nối và công tắc hoạt động ở tầng này

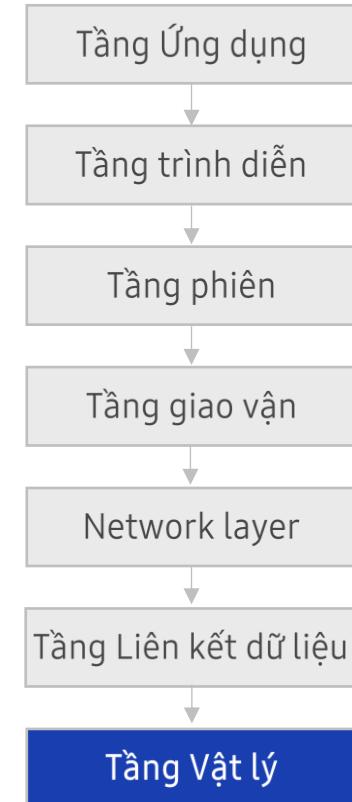


### Mô hình OSI 7 Tầng

#### Tầng Vật lý (tầng 1)

- Phương tiện vật lý được sử dụng khi truyền dữ liệu mạng.
- Xác định các tính chất vật lý và điện tử của tất cả phần cứng bao gồm điện áp, bộ chia mạng, bộ chuyển đổi mạng, bộ lặp, và các thông số kỹ thuật của dây cáp.
- Thiết lập và ngắt kết nối, cung cấp tài nguyên giao tiếp chung, chuyển đổi tín hiệu analog sang tín hiệu digital và ngược lại.

Tầng	Giao thức
Ứng dụng	HTTP, SMTP, FTP, Telnet
Trình diễn	ASCII, MPEG, JPEG, MIDI
Phiên	NetBIOS, SAP, SDP, NWLink
Giao vận	TCP, UDP, SPX
Mạng	IP, IPX
Liên kết dữ liệu	Ethernet, Token Ring, FDDI, AppleTalk



## Comparing wire communication and wireless communication (1/2)

Item	Wired	Wireless
Line Installation	<ul style="list-style-type: none"><li>Work of setting line is disadvantageous</li></ul>	<ul style="list-style-type: none"><li>No lines are needed, which is advantageous for installation.</li></ul>
Range of Communication	<ul style="list-style-type: none"><li>Long distance, Hundreds of meters</li></ul>	<ul style="list-style-type: none"><li>Limit of reach</li></ul>
Frequency of System Failures	<ul style="list-style-type: none"><li>Stable.</li></ul>	<ul style="list-style-type: none"><li>There are a lot of breakdowns</li></ul>
Maintenance cost	<ul style="list-style-type: none"><li>Low maintenance cost.</li></ul>	<ul style="list-style-type: none"><li>Maintenance costs are high.</li></ul>
Frequency of Communication error	<ul style="list-style-type: none"><li>Less Communication disturbance.</li></ul>	<ul style="list-style-type: none"><li>Many breakdowns</li></ul>
External Noise Effect	<ul style="list-style-type: none"><li>Less noise effect.</li></ul>	<ul style="list-style-type: none"><li>Noise problems and frequent communication disruptions.</li></ul>

## Comparing wire communication and wireless communication (2/2)

Item	Wired	Wireless
Weather Humidity Impact	<ul style="list-style-type: none"><li>Resistant to weather changes.</li></ul>	<ul style="list-style-type: none"><li>It is vulnerable to rain, wind, snow, frost, temperature, dew, etc.</li></ul>
Battery Utilization	<ul style="list-style-type: none"><li>Not required. Communication line and power line can be used.</li></ul>	<ul style="list-style-type: none"><li>Battery power is often needed.</li></ul>
Replace Power Battery	<ul style="list-style-type: none"><li>No need</li></ul>	<ul style="list-style-type: none"><li>Manual personnel are required for periodic replacement</li></ul>
Communication Modem Cost	<ul style="list-style-type: none"><li>Low cost with board included</li></ul>	<ul style="list-style-type: none"><li>It could be an additional cost.</li></ul>
Antenna Installation	<ul style="list-style-type: none"><li>No antenna needed</li></ul>	<ul style="list-style-type: none"><li>Various problems frequently occur at the site where the antenna is installed.</li></ul>
1:N Network Configuration	<ul style="list-style-type: none"><li>Multi-Drop is automatically built into the modem.</li></ul>	<ul style="list-style-type: none"><li>1:N configuration is possible, often with problems in the field.</li></ul>

Bài 2.

# Giao thức truyền thông

- | 2.1. Giao thức là gì
- | 2.2. Mô hình OSI
- | **2.3. Giao thức truyền thông TCP/IP**
- | 2.4. HTTP
- | 2.5. MQTT
- | 2.6. CoAP
- | 2.7. WebSocket

## Giao thức truyền thông TCP/IP

- | Giao thức điều khiển truyền vận/Giao thức Internet
- | Phát triển bởi DARPA
- | Không có tiêu chuẩn giao thức chính thức nào
- | Hệ thống phân cấp tương tác của các mô-đun cung cấp chức năng cụ thể
- | Xác định 5 tầng
  - ▶ Ứng dụng
  - ▶ Máy chủ - Máy chủ (vận chuyển)
  - ▶ Internet
  - ▶ Truy cập mạng
  - ▶ Vật lý

### Tầng Vật lý

| Giao diện vật lý giữa DTE (ví dụ máy tính hoặc thiết bị đầu cuối) và một phương tiện truyền dẫn

| Xác định:

- ▶ Đặc điểm của phương tiện
- ▶ Bản chất của tín hiệu
- ▶ Tốc độ truyền dữ liệu

## Tầng Truy cập mạng

- | Trao đổi dữ liệu giữa các hệ thống trong một mạng chung
- | Sử dụng địa chỉ của máy chủ và máy đích
- | Có thể ưu tiên truyền dữ liệu
- | Các phần mềm ở tầng này phụ thuộc vào mạng
  - ▶ e.g. X.25 vs. Ethernet
- | Độc lập nghĩa là không một phần mềm nào phải lo lắng về các thông số mạng

## Tầng Internet

- | Internet là sự kết nối của hai hoặc nhiều mạng
- | Tầng Internet xử lý các tác vụ giống như tầng truy cập mạng nhưng là giữa các mạng chứ không phải là giữa các nút trong mạng
- | Sử dụng IP để đánh địa chỉ và định tuyến trên toàn mạng
- | Được triển khai trong các máy trạm và các bộ định tuyến

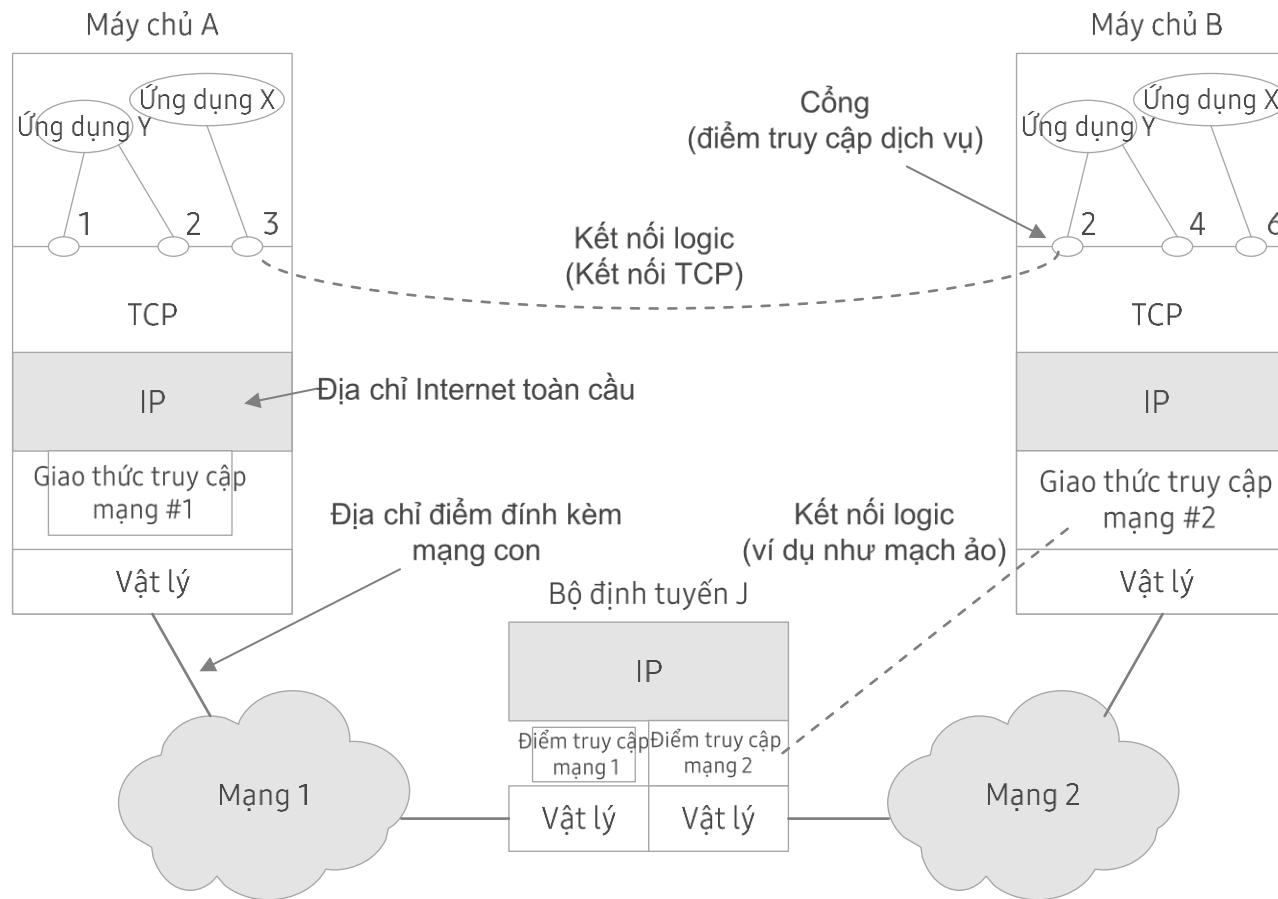
## Tầng giao vận

- | Còn được gọi là tầng máy chủ - máy chủ
- | Trao đổi dữ liệu một cách tin cậy giữa các ứng dụng
- | Sử dụng các giao thức TCP để truyền dữ liệu

### Tầng ứng dụng

- | Logic cần thiết để hỗ trợ nhiều ứng dụng
- | Các mô-đun độc lập hỗ trợ từng loại ứng dụng
  - ▶ e.g. truyền tệp tin

## Hoạt động của TCP/IP

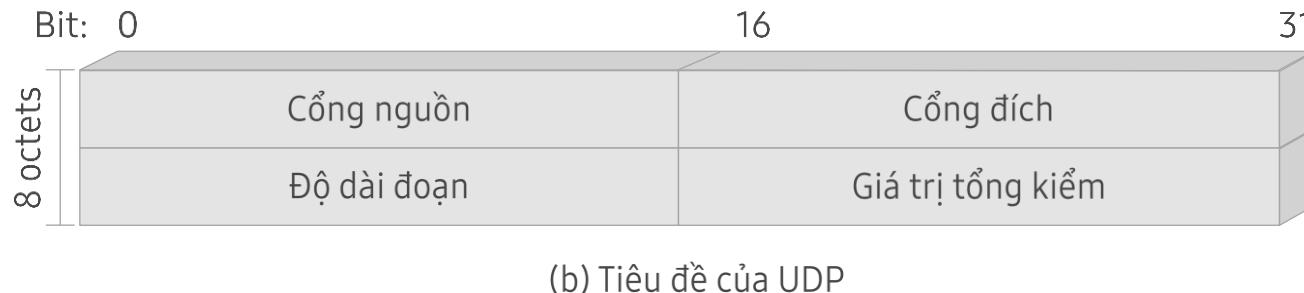
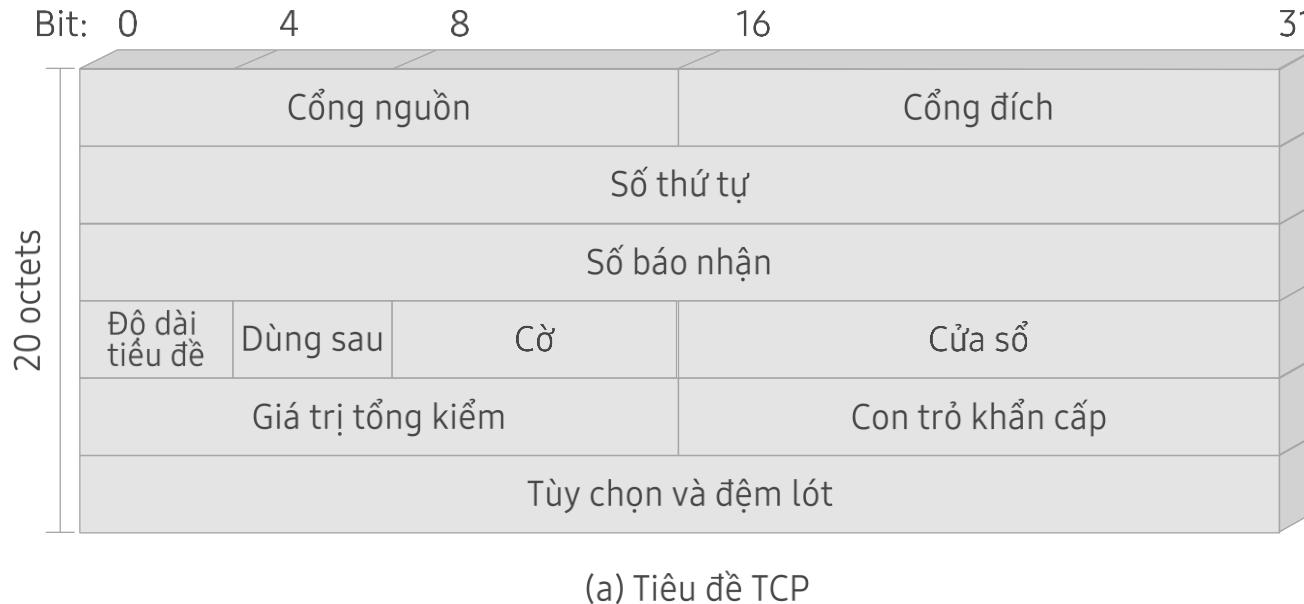


### TCP & UDP

- | Hầu hết các ứng dụng TCP/IP đều sử dụng giao thức TCP cho tầng giao vận
- | Giao thức TCP cung cấp kết nối (liên kết logic) giữa hai đối tượng để điều chỉnh lỗi kiểm tra luồng
- | UDP (Giao thức gói tin người dùng) không duy trì kết nối và do đó không đảm bảo việc truyền dữ liệu, bảo toàn trình tự hoặc chống trùng lặp

## 2.3. Giao thức truyền thông TCP/IP

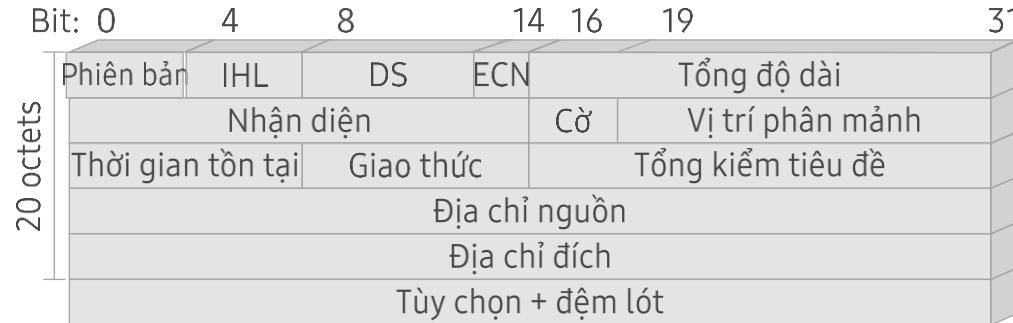
### Bài 02



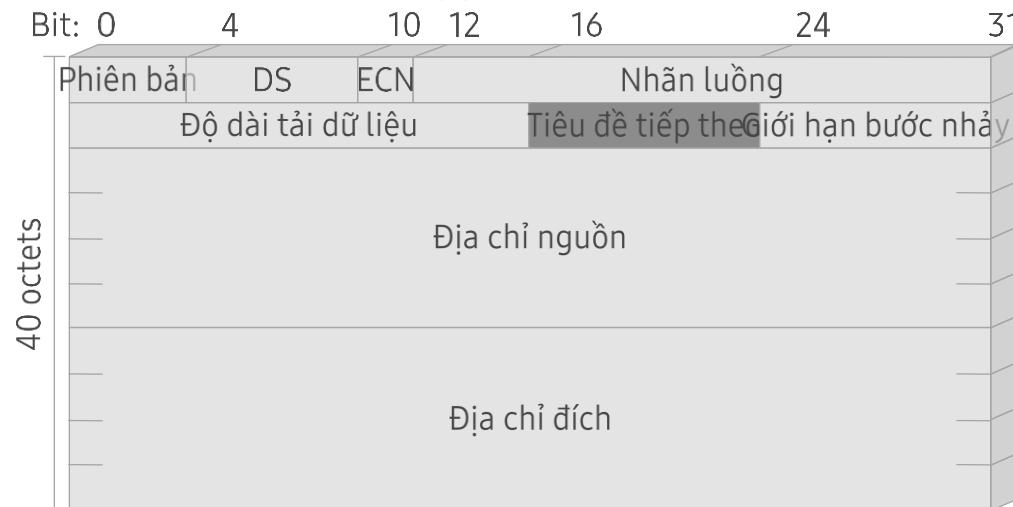
### IP và IPv6

- | IP hỗ trợ các địa chỉ nguồn và đích 32 bit
- | IPv6 (tiêu chuẩn 1996) hỗ trợ các địa chỉ 128 bit
- | Quá trình truyền dữ liệu đến IPv6 sẽ rất chậm

## Tiêu đề IP



(a) Tiêu đề IPv4



DS = Các dịch vụ khác biệt

ECN = Thông báo tắc nghẽn rõ ràng

## Ứng dụng TCP/IP

- | SMTP (Giao thức truyền tải thư tín đơn giản)
  - ▶ Thiết bị e-mail cơ bản, truyền dữ liệu giữa các máy chủ
- | FTP (Giao thức truyền tệp tin)
  - ▶ Gửi các tệp tin từ một hệ thống đến một hệ thống khác theo yêu cầu của người dùng
- | SSH (Shell bảo mật)
  - ▶ Bảo mật khả năng đăng nhập từ xa, cho phép người dùng đăng nhập từ xa vào máy tính

## Kết nối mạng Internet

- | Các mạng được kết nối với nhau thường sử dụng giao thức TCP/IP
- | Đối với người dùng, đó là một mạng lớn
- | Mạng Internet toàn cầu là ví dụ tiêu biểu nhất, nhưng mạng nội bộ và mạng ngoại bộ cũng là các ví dụ của kết nối mạng

## Bộ định tuyến

- | Thiết bị được sử dụng để kết nối các mạng độc lập
- | Có nhiều chức năng quan trọng
  - ▶ Cung cấp liên kết giữa các mạng
  - ▶ Cung cấp định tuyến và truyền dữ liệu giữa các quá trình trên hệ thống từ các mạng khác nhau
  - ▶ Cung cấp các chức năng mà không yêu cầu chỉnh sửa cấu trúc mạng

## So sánh giữa TCP/IP và OSI



## Đánh địa chỉ

| Mối liên hệ giữa Địa chỉ và tầng trong giao thức TCP/IP



### | Địa chỉ vật lý

- ▶ Địa chỉ liên kết
- ▶ Địa chỉ của nút được xác định trong mạng WAN hoặc mạng LAN
- ▶ Địa chỉ 6 byte (48 bit) của Card giao diện mạng (NIC) Ethernet
- ▶ Unicast, multicast, broadcast

07:01:02:01:2C:4B

địa chỉ vật lý gồm 6 byte (12 chữ số thập lục phân)

### | Địa chỉ logic (hoặc địa chỉ IP)

- ▶ Xác định các máy chủ đang kết nối mạng Internet: Hệ thống địa chỉ 32 bit
- ▶ Unicast (một người dùng), Multicast (nhiều người nhận), broadcast (tất cả hệ thống trong mạng)
- ▶ Sau đó được chuyển thành một địa chỉ IPv6 có độ dài địa chỉ gấp bốn lần

### | Địa chỉ cổng

- ▶ Cần phân biệt các quy trình gửi và nhận dữ liệu trong giao tiếp máy tính - máy tính
- ▶ Địa chỉ được gán với mỗi quá trình
- ▶ Dài 16 bit (từ 0 đến 65535)

Bài 2.

# Giao thức truyền thông

- | 2.1. Giao thức là gì
- | 2.2. Mô hình OSI
- | 2.3. Giao thức truyền thông TCP/IP
- | 2.4. **HTTP**
- | 2.5. MQTT
- | 2.6. CoAP
- | 2.7. WebSocket

## Khái niệm HTTP

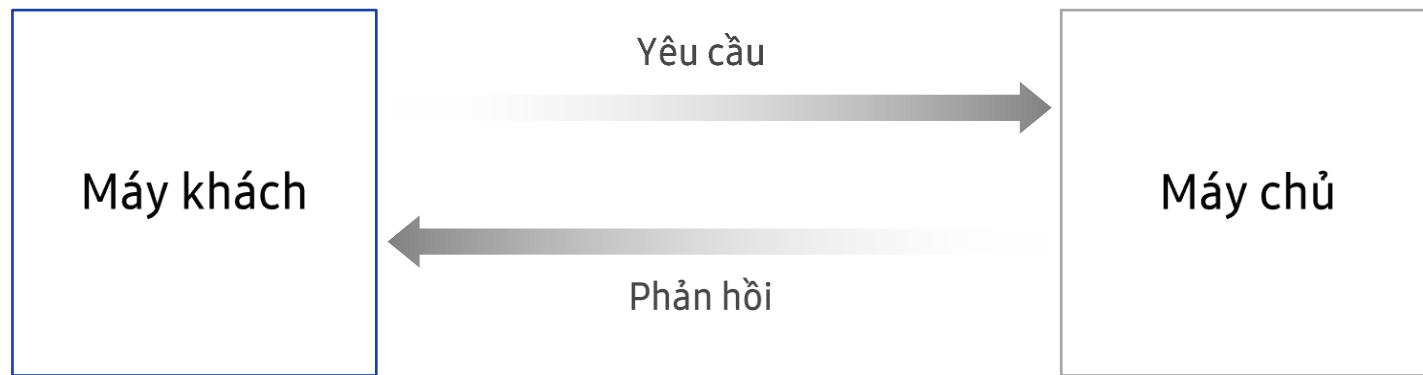
- | Giao thức truyền tải siêu văn bản (HTTP) là một giao thức sử dụng mô hình chủ/khách để gửi và nhận dữ liệu qua mạng Internet.
- | Giao thức yêu cầu/phản hồi là một giao thức truyền thông giữa trình duyệt web và máy chủ web

## Tính năng và lợi ích của HTTP

- | Sau khi kết nối đến máy chủ trong một giao tiếp không kết nối và nhận được phản hồi cho yêu cầu, kết nối sẽ bị hủy bỏ.
  - ▶ ưu điểm: Dịch vụ hiệu quả và không cần bảo dưỡng nhiều.
  - ▶ nhược điểm: Không xác định được trạng thái trước đó của máy khách sau khi hủy kết nối nên không thể lưu các thông tin về máy khách đó.
- | Cách để giải quyết vấn đề "không thể xác định trạng thái" của máy khách
  - ▶ Cookie: xử lý vấn đề bằng cách lưu các thông tin trạng thái cho máy khách

## Môi trường hoạt động của HTTP

- | Máy khách: Một máy tính có phần mềm khách được cài đặt trên máy chủ và kết nối đến máy chủ bằng một đường dẫn URL và dữ liệu yêu cầu (chrome, firefox, ie, v.v.).
- | Máy chủ: Một máy tính có phần mềm chấp nhận các yêu cầu từ máy khách, diễn giải và phản hồi các yêu cầu đó (Apache, nginx, IIS, lighttpd)



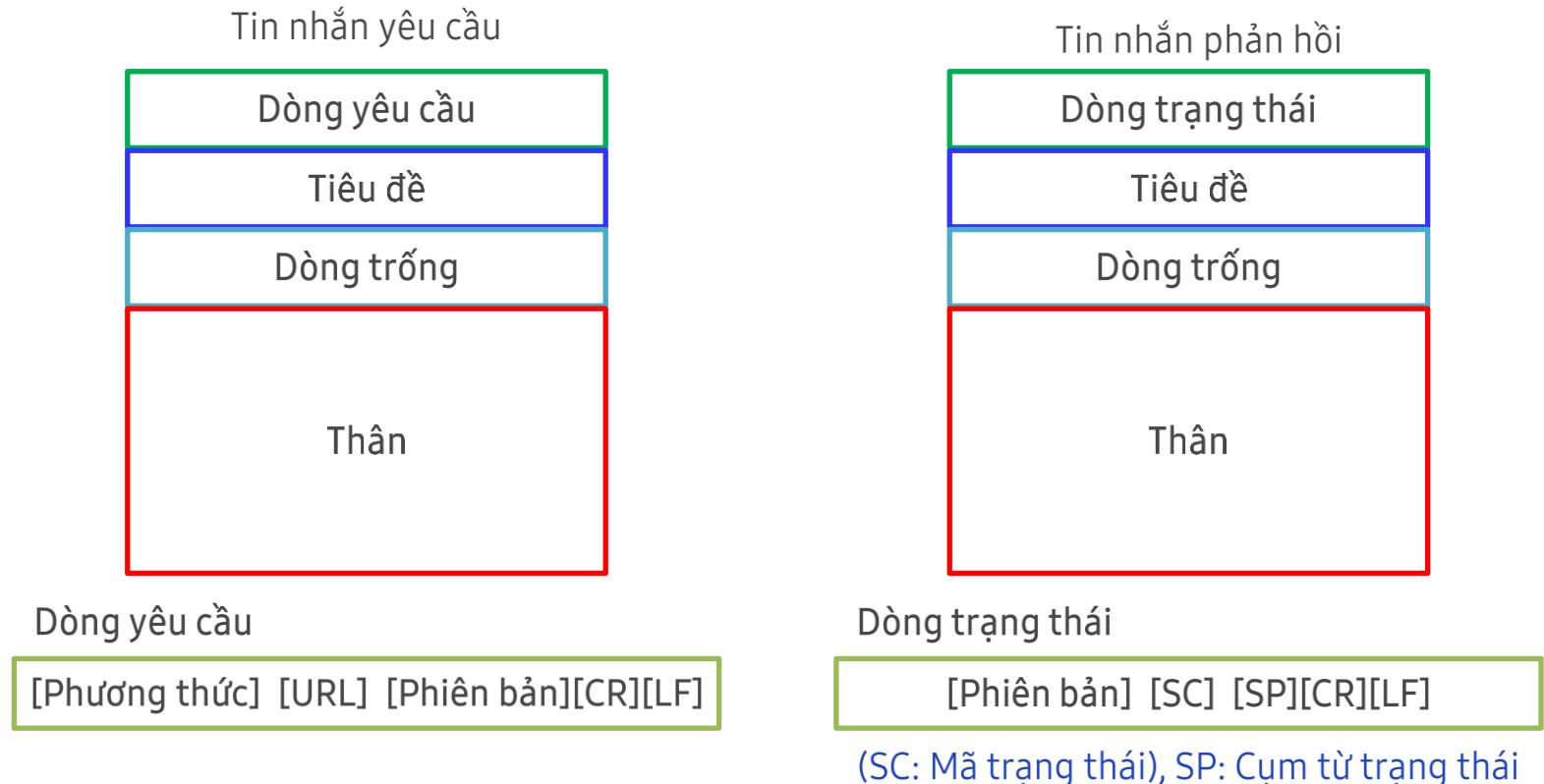
## Phương pháp HTTP

- | GET: Yêu cầu thông tin
- | POST: chèn thông tin
- | PUT: cập nhật thông tin
- | DELETE: xóa thông tin
- | HEAD: chỉ yêu cầu các thông tin tiêu đề
- | OPTIONS: Yêu cầu phương thức mà máy chủ web hỗ trợ
- | TRACE: Trả về yêu cầu từ máy khách

## Mã phản hồi HTTP

- | 1XX phản hồi có điều kiện
- | 2XX thành công
  - | 200: Thành công, 201: Đã tạo, 202: Đã cho phép, 204: Không có nội dung
  - | 205: đặt lại nội dung, 206: thành công một phần, 207: đa trạng thái
- | 3XX điều hướng lại
  - | 300: nhiều mục chọn, 301: chuyển động cố định, 302: chuyển động tạm thời
  - | 303: tầm nhìn ở các vị trí khác, 304: không được chỉnh sửa, 305: sử dụng proxy
- | 4XX Lỗi yêu cầu
  - | 400: Yêu cầu sai, 401: Lỗi ủy quyền, 403: Bị cấm 404: Không tìm thấy
  - | 405: Phương thức không được phép 406: Không được phép, 408: Hết thời gian yêu cầu
  - | 410 không tồn tại, 411: yêu cầu khai báo độ dài

## Định dạng dữ liệu yêu cầu và phản hồi HTTP



Bài 2.

# Giao thức truyền thông

- | 2.1. Giao thức là gì
- | 2.2. Mô hình OSI
- | 2.3. Giao thức truyền thông TCP/IP
- | 2.4. HTTP
- | 2.5. MQTT
- | 2.6. CoAP
- | 2.7. WebSocket

## Khái niệm MQTT

- | Message Queue Telemetry Transport (MQTT) là một giao thức gửi thông điệp được phát triển bởi IBM vào năm 1999 để gửi thông điệp tin cậy (giám sát từ xa) cho các thiết bị nhỏ như dụng cụ đo và cảm biến trong các môi trường mạng có độ trễ cao và thường bị mất tín hiệu.
- | MQTT là giao thức dạng Publish / Subscribe cấp thấp. Được sử dụng trong môi trường máy - máy và môi trường IoT

## Ưu nhược điểm của MQTT

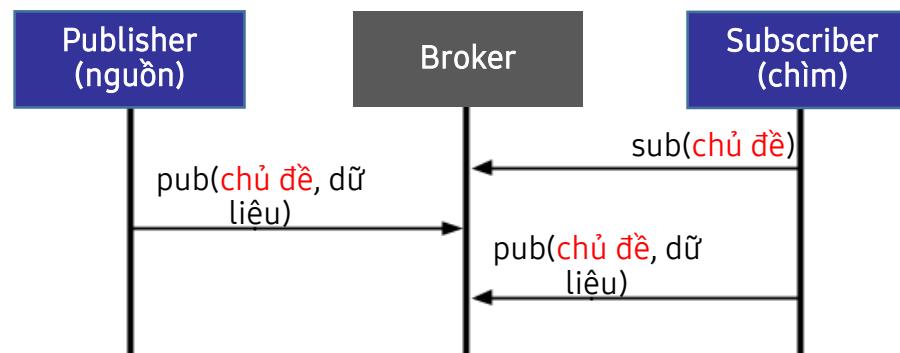
- | Định dạng thông điệp đơn giản, băng thông mạng hẹp và mức tiêu thụ pin thấp.
- | Được thiết kế cho các môi trường mạng hoạt động ở mức năng lượng thấp và có độ trễ hoặc mất dữ liệu cao khi truyền tải, tạo lợi thế kiểm soát các thiết bị nhỏ và thu thập thông tin cảm biến.
- | Giao thức này phù hợp để sử dụng trên các thiết bị nhỏ nhưng lại không hỗ trợ cụm thiết bị, do đó rất khó để có thể xây dựng một hệ thống thông điệp lớn.

## Các tính năng của MQTT

### | Publish/Subscribe

- ▶ Về nguyên lý, giao thức MQTT xuất bản một thông điệp và đăng ký theo dõi một chủ đề.
- ▶ Cả Publisher (Bên xuất bản) và Subscriber (Bên đăng ký theo dõi) đều là khách hàng của Broker (Máy chủ). Publisher kết nối đến Broker để xuất bản chủ đề, và kết nối đến Subscriber để đăng ký theo dõi chủ đề đó.
- ▶ Nhiều Publisher và Subscriber có thể kết nối đến một Broker để xuất bản hoặc đăng ký theo dõi các chủ đề. Nhiều máy khách có thể đăng ký theo dõi cùng một chủ đề.

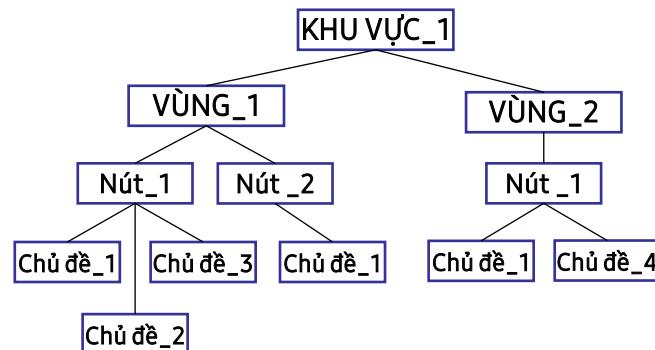
### | Môi trường hoạt động theo kiểu xuất bản/đăng ký theo dõi



- ▶ Các nhà môi giới đóng vai trò trung gian giữa Publisher và Subscriber. Vì nhiều Subscriber có thể đăng ký một chủ đề nên nó cũng rất hữu ích để xây dựng phương thức giao tiếp 1:N.

### I Chủ đề

- ▶ Publisher và Subscriber xuất bản/đăng ký theo dõi thông điệp trên từng kênh. Đây được gọi là một “chủ đề” trong MQTT.
- ▶ Các chủ đề có cấu trúc phân cấp được tách ra bằng các dấu gạch chéo (/) và hỗ trợ các ký tự đại diện để chỉ rõ nhiều chủ đề khi xuất bản/đăng ký theo dõi các thông điệp.
- ▶ Trong MQTT, các chủ đề được xây dựng bằng cách sử dụng /. Nếu có ý định cấu hình các lớp như trong hình bên dưới, sẽ rất dễ dàng để quản lý dữ liệu như cảm biến IoT.



- ▶ Kiểu đại diện
  - + (Cấp độ đơn): Xử lý ngẫu nhiên một chủ đề
  - # (Đa cấp độ): Liên quan đến nhiều chủ đề con, chỉ được đặt ở phía dưới cùng của chuỗi chủ đề
- ▶ Ví dụ về việc sử dụng Đại diện
  - sensors / NODE\_NAME / + / CPU
  - Sensors / NODE\_NAME / #

## I QoS (Chất lượng dịch vụ)

- ▶ MQTT có ba cấp độ chất lượng dịch vụ (QoS)
  - Cấp độ 0: Chỉ 1 thông điệp gửi, có thể mất
  - Cấp độ 1: Thông điệp được gửi ít nhất một lần, khả năng thành công cao gấp đôi
  - Cấp độ 2: Thông điệp được gửi đúng một lần, đảm bảo chất lượng nhưng đòi hỏi hệ thống phải có hiệu suất cao

※ Tài liệu tham khảo

**Cấp 0:** Một tin nhắn chỉ được gửi một lần. Sau khi giao hàng, quá trình tiếp nhận không được kiểm tra.

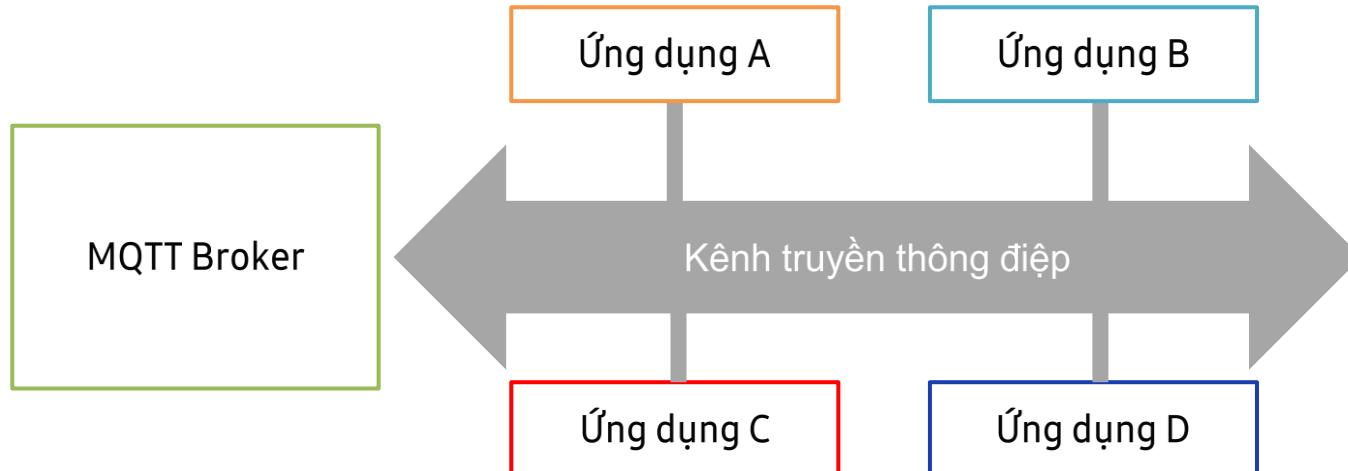
**Cấp 1:** Một tin nhắn được gửi nhiều hơn một lần. Quá trình handshaking được theo dõi, nhưng có khả năng nhận trùng lặp vì nó không được theo dõi chặt chẽ.

**Cấp 2:** Một tin nhắn chỉ được gửi một lần và tất cả các quy trình handshaking đều được kiểm tra.

- ▶ Mức QoS càng cao, chất lượng truyền thông càng tốt, nhưng có một khả năng làm giảm hiệu suất. Do đó, QoS của MQTT được quyết định tùy theo đặc điểm của dự án.

### I Kênh truyền thông điệp

- ▶ MQTT có hệ thống kênh truyền thông điệp. Khi máy chủ broker MQTT tạo ra một kênh truyền thông điệp và gửi một thông điệp từ publisher đến máy chủ, các ứng dụng đính kèm kênh truyền đó sẽ đọc thông điệp.
- ▶ Các thông điệp thuộc các chủ đề khác nhau có thể di chuyển trên kênh truyền thông điệp, tạo ra một kênh truyền thông điệp có tên là Chủ đề để phân biệt các thông điệp.



## Định dạng thông điệp MQTT

| Có thể tùy ý lựa chọn định dạng thông điệp MQTT, trừ tiêu đề cố định dài 2 byte

	0	1	2	3	4	5	6	7
Byte1	Kiểu thông điệp				Cờ DUP	Cấp độ QoS		RETAIN
Byte2	Độ dài còn lại							

### | Kiểu thông điệp

Kiểu	Đánh số	Mô tả
Dùng sau	0	Dùng sau
CONNECT	1	Máy khách yêu cầu kết nối đến máy chủ
CONNACK	2	Báo nhận kết nối
PUBLISH	3	Xuất bản thông điệp
PUBACK	4	Báo nhận xuất bản
PUBREC	5	Xuất bản đã nhận (xác nhận đã gửi phần 1)
PUBREL	6	Xuất bản đã nhận (xác nhận đã gửi phần 2)
PUBCOMP	7	Xuất bản hoàn thành (xác nhận đã gửi phần 3)
SUBSCRIBE	8	Máy khách yêu cầu đăng ký theo dõi
SUBACK	9	Báo nhận đăng ký theo dõi
UNSUBSCRIBE	10	Báo nhận hủy đăng ký theo dõi
UNSUBACK	11	Báo nhận hủy đăng ký theo dõi
PINGREQ	12	Yêu cầu PING
PINGRESP	13	Phản hồi PING
DISCONNECT	14	Máy khách ngắt kết nối
Dùng sau	15	Dùng sau

### | Cờ DUP & cấp độ QoS và RETAIN

- ▶ RETAIN chỉ được sử dụng trong các thông điệp Xuất bản
- ▶ (Dù các thông điệp được lưu trong hàng chờ)

Vị trí Bit	Tên	Mô tả
3	DUP	Phân phối trùng lặp
2-1	QoS	Chất lượng dịch vụ
0	RETAIN	Cờ RETAIN

## | Cấp độ QoS

Giá trị QoS	Bit 2	Bit 1	Mô tả
0	0	0	Phát và Quên (Tối đa một lần)
1	0	1	Xác nhận phân phối (Ít nhất một lần)
2	1	0	Phát và Quên (Chính xác một lần)
3	1	1	Phát và Quên (Tối đa một lần)
4	1	1	Dùng sau

### | Độ dài còn lại

- ▶ Được sử dụng để đo kích thước toàn bộ thông điệp, bao gồm tiêu đề thay đổi. Nó sử dụng từ 1 đến 4 byte tùy theo kích thước dữ liệu.

### | Tiêu đề thay đổi

- ▶ Một số lệnh MQTT có chứa một phần tử tiêu đề thay đổi. Các tiêu đề thay đổi được đặt giữa các tiêu đề cố định và tải dữ liệu

## Mục tiêu triển khai MQTT

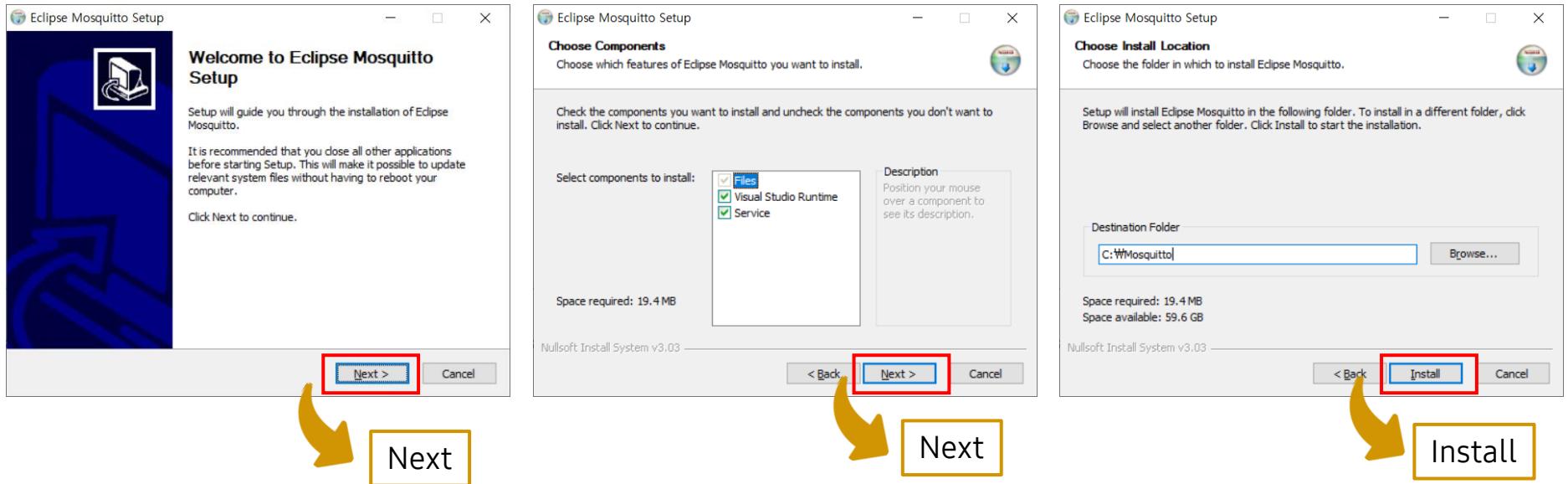
| Triển khai một chương trình trò chuyện đơn giản để giúp hiểu về giao thức MQTT

| Môi trường triển khai

- ▶ Windows 10 64bit
- ▶ Ruby 2.3.3
- ▶ Mosquitto 1.4.7
- ▶ (không bắt buộc) Raspberry pi

## Cấu hình môi trường MQTT

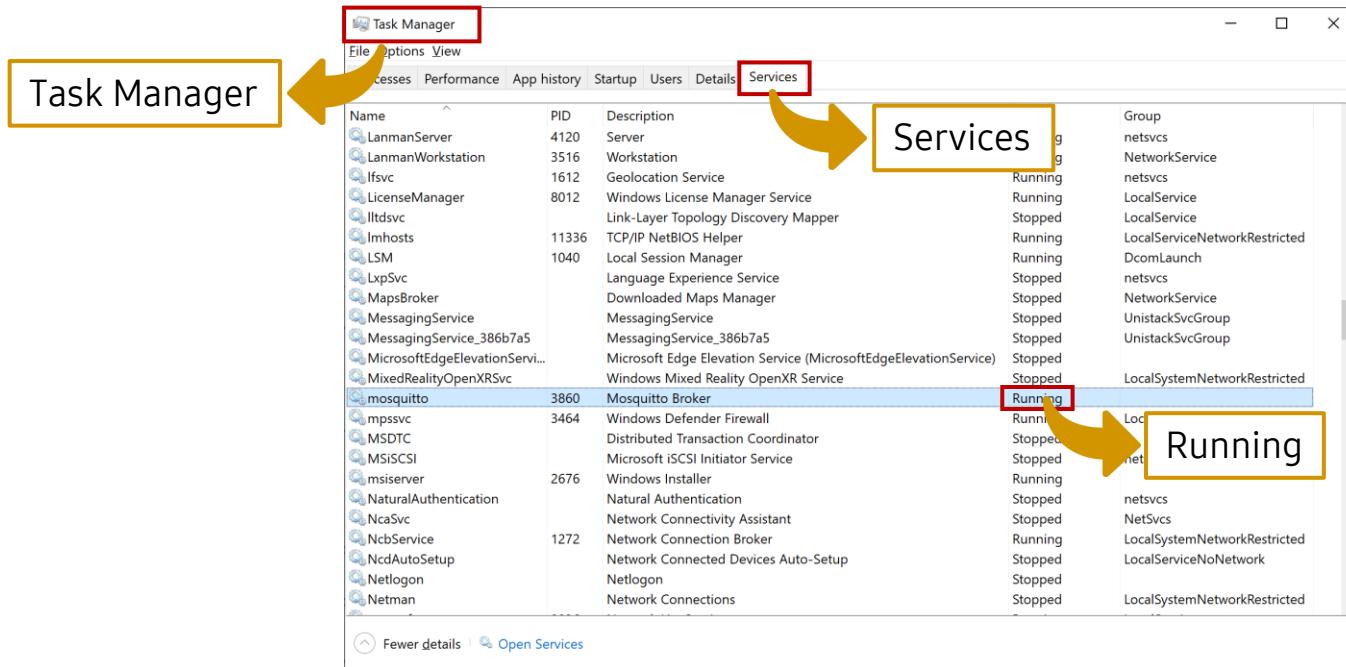
- I Truy cập trang web Mosquitto (<https://mosquitto.org/download/>) để tải xuống phiên bản Windows ‘mosquitto-2.0.12-install-windows-x64.exe’ và cài đặt trong C:\mosquitto.



## Running MQTT Broker Server (mosquitto)

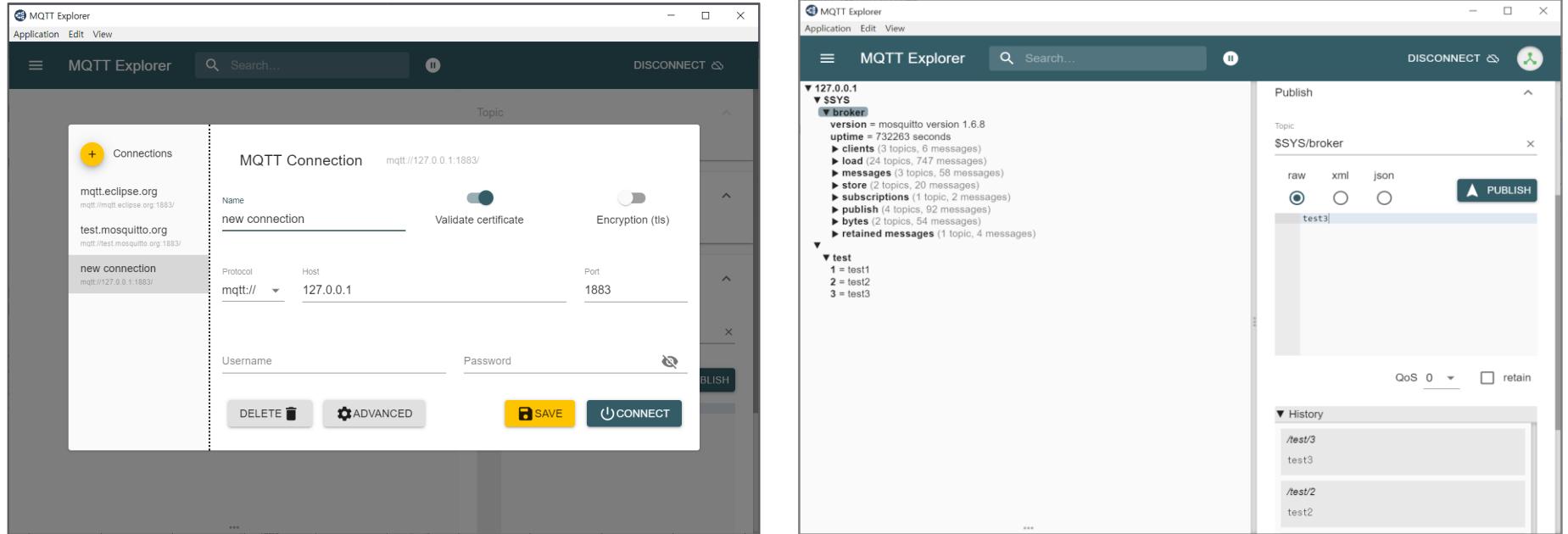
### I Running Mosquitto

- After installation is complete, Mosquitto is registered as a Windows service. You can check whether the Mosquitto service is running through the Windows task manager.
- Press Ctrl + Shift + Esc on your keyboard. When you press these keys, the task manager appears.



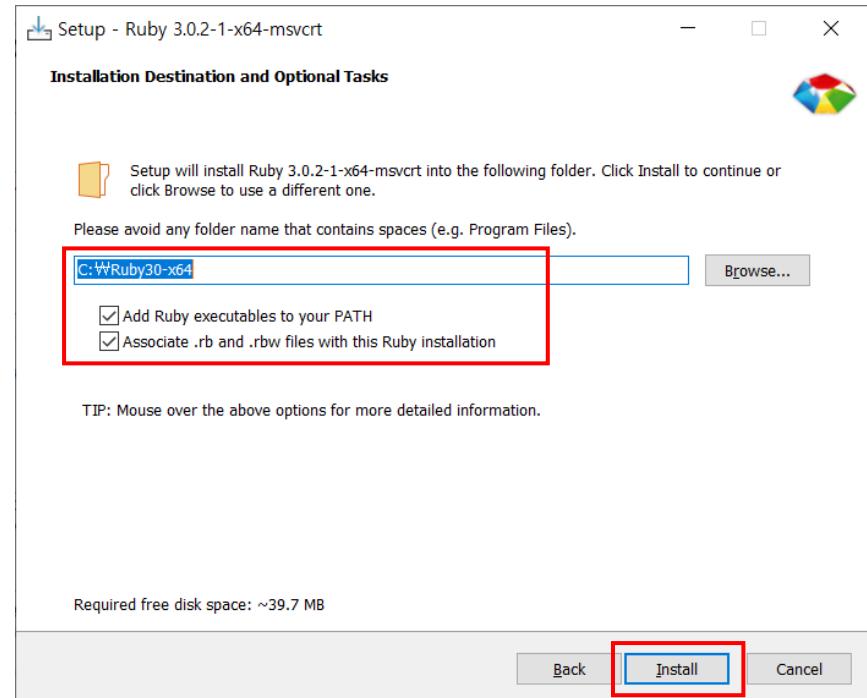
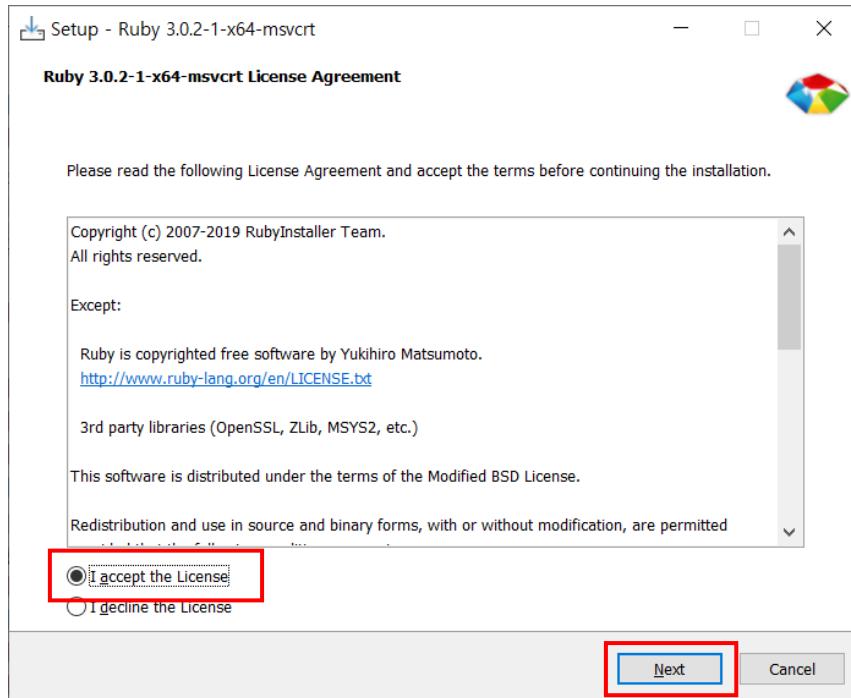
## Mosquitto Test

- Visit <https://mqtt-explorer.com/> to download MQTT Explorer and test the MQTT operation in a GUI environment.
- Create a new one in Connections in the figure on the left. Set the Host IP address to 127.0.0.1 (local host) and the Port number to 1883. You can see that the message is created and received as shown in the figure on the right.

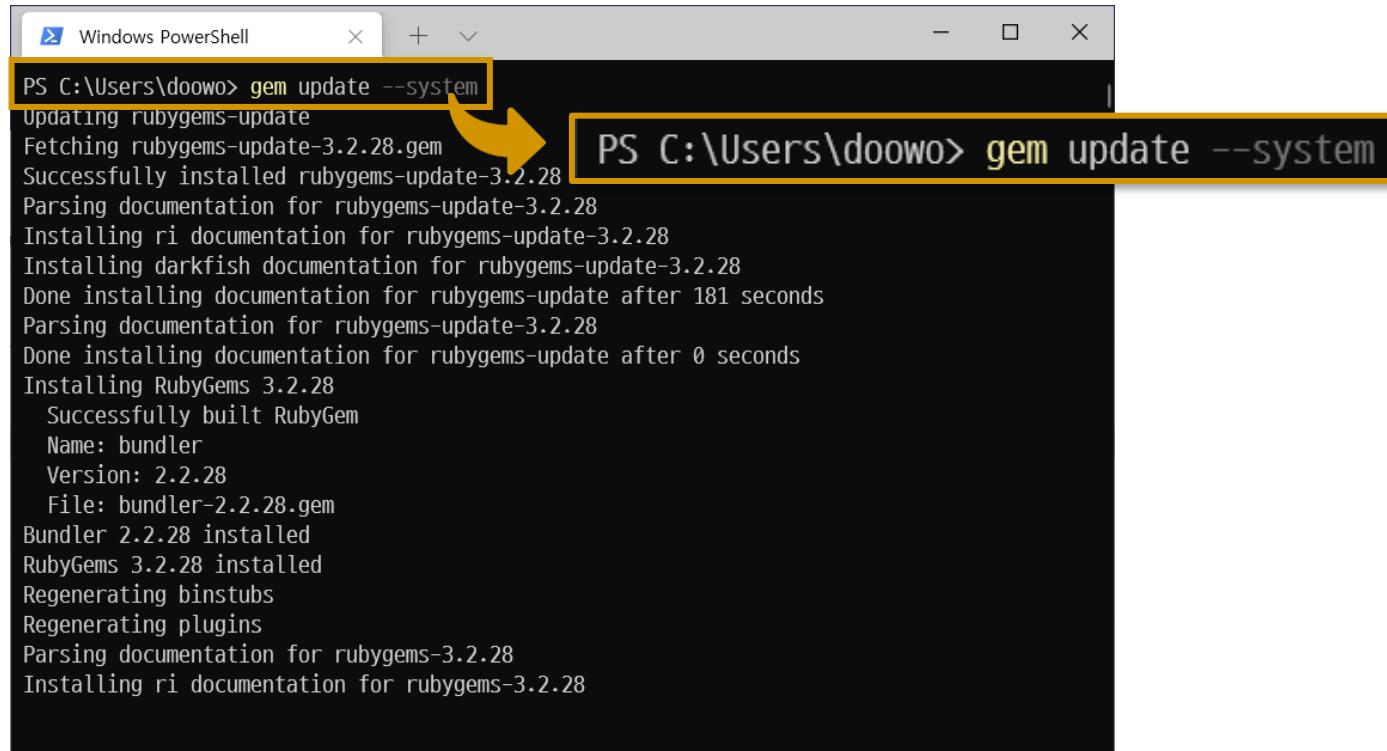


## Install Ruby to build a MQTT Client

- I Use Ruby to build a MQTT Client.
- I Install ruby by executing the ‘rubyinstaller-3.0.2-1-x64.EXE’ file.
  - ▶ Visit <https://rubyinstaller.org/downloads/> to download the file.



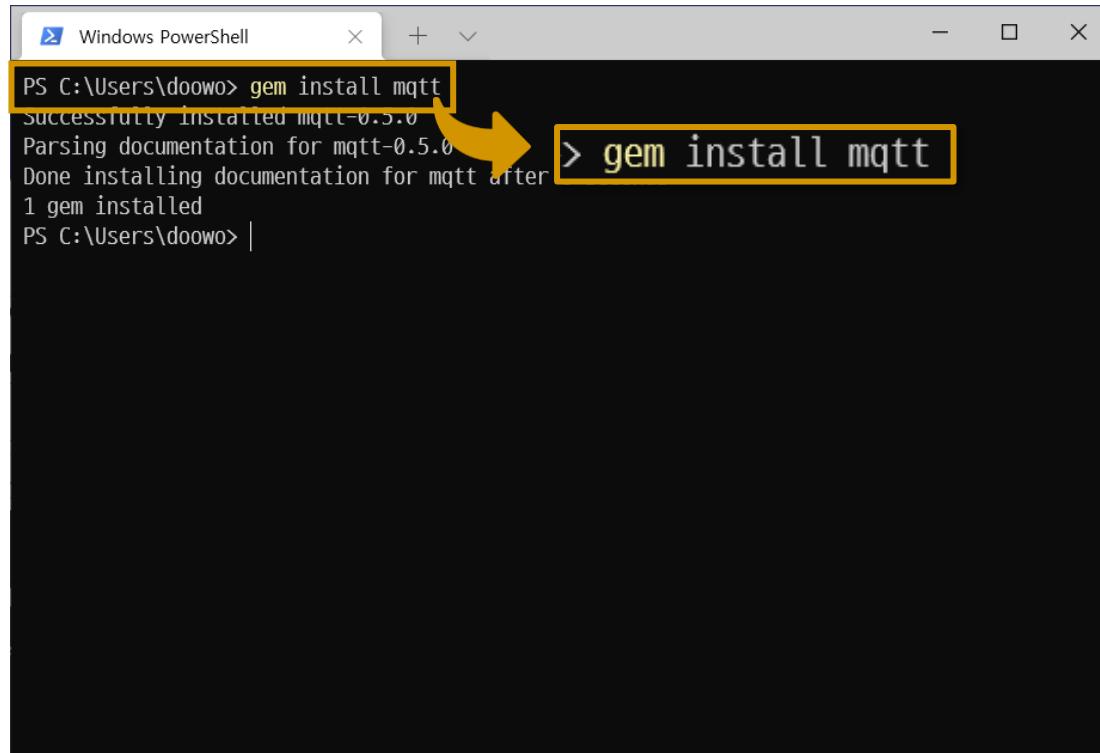
- | RubyGems is a package management framework for Ruby.
- | Pure Ruby gem implements the MQTT protocol which is a lightweight protocol for publish/subscribe messaging.
- | Visit <https://rubygems.org/pages/download> to download RubyGems.
  - ▶ <https://rubygems.org/>



```
Windows PowerShell
PS C:\Users\doowo> gem update --system
Updating rubygems-update
Fetching rubygems-update-3.2.28.gem
Successfully installed rubygems-update-3.2.28
Parsing documentation for rubygems-update-3.2.28
Installing ri documentation for rubygems-update-3.2.28
Installing darkfish documentation for rubygems-update-3.2.28
Done installing documentation for rubygems-update after 181 seconds
Parsing documentation for rubygems-update-3.2.28
Done installing documentation for rubygems-update after 0 seconds
Installing RubyGems 3.2.28
  Successfully built RubyGem
    Name: bundler
    Version: 2.2.28
    File: bundler-2.2.28.gem
Bundler 2.2.28 installed
RubyGems 3.2.28 installed
Regenerating binstubs
Regenerating plugins
Parsing documentation for rubygems-3.2.28
Installing ri documentation for rubygems-3.2.28
```

I Chạy Start/Command Prompt bằng phần mềm Ruby trên Windows, sau đó cài đặt thư viện MQTT theo hình minh họa và các lệnh dưới đây.

- ▶ <https://rubygems.org/>



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command "PS C:\Users\doowo> gem install mqtt" being typed and its execution. A yellow arrow points from the text "gem install mqtt" to the command line. The output shows the gem was successfully installed, documentation was parsed, and 1 gem was installed.

```
PS C:\Users\doowo> gem install mqtt
Successfully installed mqtt-0.5.0
Parsing documentation for mqtt-0.5.0
Done installing documentation for mqtt after
1 gem installed
PS C:\Users\doowo> |
```

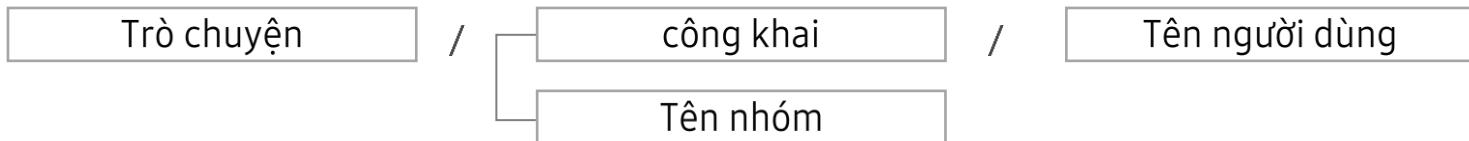
## Chạy Chương trình trò chuyện MQTT (Ruby)

| Chạy Command Prompt với Ruby và sử dụng lệnh dưới đây

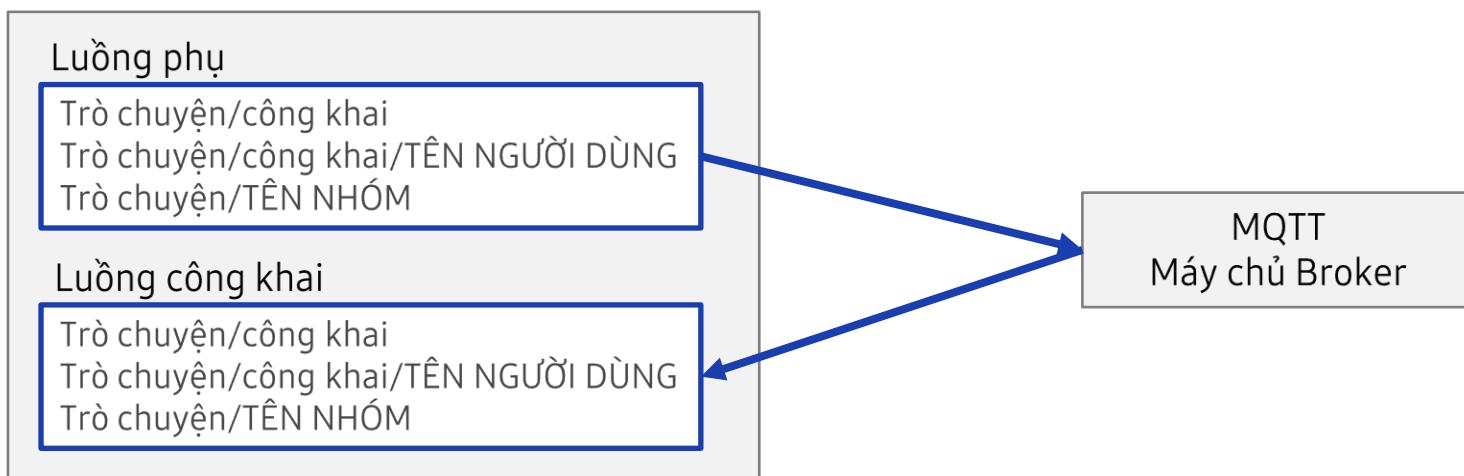
- ▶ cd C:\
- ▶ ruby filename.rb Argument value

## Tạo Chương trình trò chuyện MQTT bằng Ruby

- | Sơ đồ chủ đề của chương trình trò chuyện MQTT và sơ đồ cấu trúc hệ thống.



Máy khách



## Sử dụng thư viện API của MQTT bằng Ruby

### | Khai báo thư viện API

- ▶ require 'rubygems'
- ▶ require 'mqtt' # Load the mqtt library from the line of code

### | Tạo đối tượng máy khách MQTT.

- ▶ mqtt = MQTT::Client.new ('Địa chỉ IP của broker MQTT')

### | Kết nối bằng đối tượng khách MQTT

- ▶ mqtt.connect('test.mosquitto.org') do |client|
- ▶     # perform operations
- ▶ end

| Xuất bản API: Nhập chủ đề và thông điệp để xuất bản

- ▶ client.publish(topic, payload)

| Đăng ký theo dõi API: Nhập chủ đề để đăng ký theo dõi là giá trị đối số

- ▶ client.subscribe('topic1')
- ▶ client.subscribe('topic1', 'topic2')
- ▶ client.subscribe('foo/#')

| Nhận thông điệp: Nhận các chủ đề và thông điệp của chúng.

## Mẫu code triển khai MQTT

```
1  require 'rubygems'
2  require 'mqtt'
3  require 'readline'
4
5  # nickname
6  nickname=ARGV[0]
7
8  # localhost test
9  mqtt = MQTT::Client.new('localhost')
10
11 mqtt.connect do |client|
12   client.subscribe('chat/public')
13   client.subscribe("chat/private/#{nickname}")
14   enter_msg = "#{nickname} enter room!"
15   client.publish 'chat/public', enter_msg
16
17   # Publish Thread
18   Thread.new do
19     while message = Readline.readline("", true)
20       case message
21       when /^priv\s*(\w*)\s*(.*)/    # individual message
22         client.publish "chat/private/#{$1}", "<#{nickname}> : #{$2}"
23       when /^quit\s*(.*)/          # Exit
24         client.publish 'chat/public', "#{nickname} has quit (#{$1})"
25         exit 1
26       else                      # public message
27         client.publish 'chat/public', "#{nickname} : #{message}"
28       end
29     end
30   end
31
32   loop do
33     topic, message = client.get
34     print message, "\n"
35   end
36 end
```

- | Mã nguồn chương trình trò chuyện MQTT
- | Tin nhắn công khai: tin nhắn
- | Nói chuyện riêng: \ w usr\_name message
- | Thoát: \ q message
- | Chạy phương thức:  
ruby Chat\_Example.rb username

## Kết quả chạy Chương trình trò chuyện MQTT bằng Ruby

Mở hai cửa sổ như hình bên dưới và đổi tên thành chat. Bạn có thể thấy rằng cuộc trò chuyện đang diễn ra thông qua MQTT Broker.

```
PS C:\Mosquitto> ruby Chat_example.rb Daniel
Daniel enter room!!
Thomas enter room!!
Thomas : Hi ~~~
I'm Daniel.
Daniel : I'm Daniel.

PS C:\Mosquitto> ruby Chat_example.rb Daniel
```

```
PS C:\Mosquitto> ruby Chat_example.rb Thomas
Thomas enter room!!
Hi ~~~
Thomas : Hi ~~~
Daniel : I'm Daniel.

PS C:\Mosquitto> ruby Chat_example.rb Thomas
```

## Raspberry pi: Cấu hình chương trình trò chuyện MQTT

- | Chạy máy chủ mosquitto trên Windows
- | Thay đổi mã nguồn mẫu của chương trình trò chuyện MQTT
  - ▶ Trước khi thay đổi: mqtt = MQTT::Client.new('localhost')
  - ▶ Sau khi thay đổi: mqtt = MQTT::Client.new('mosquitto server ip')

```
# Username  
usr_name=ARGV[0]  
  
# connect to mqtt server in localhost  
#mqtt = MQTT::Client.new('localhost')  
mqtt = MQTT::Client.new('210.107.192.185')
```

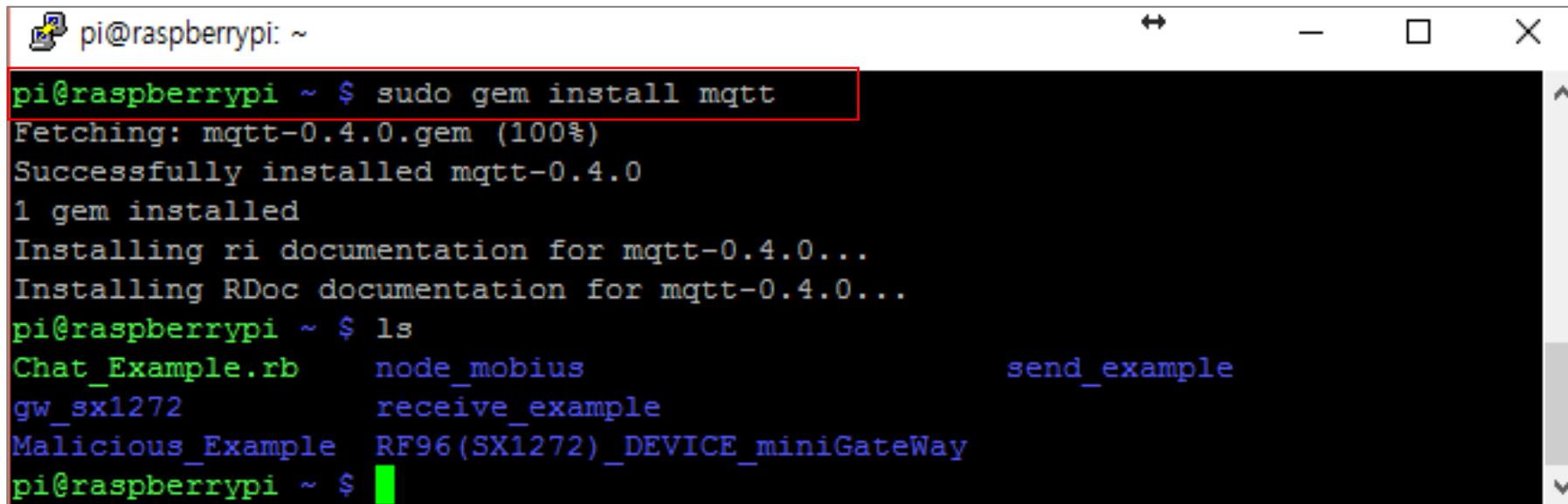
Lưu các tệp tin nguồn của Ruby bằng samba trong thư mục gốc của Pi

Network > 192.168.0.7 > pi		
	name	Date Modified
	gw_sx1272	2016-09-26 .
	Malicious_Example	2016-11-24 .
	node_mobius	2016-12-14 .
	receive_example	2017-01-08 .
	RF96(SX1272)_DEVICE_miniGateWay	2016-11-24 .
	send_example	2016-11-24 .
	Chat_Example	2017-01-18

| Kết nối Raspberry pi bằng putty hoặc màn hình LCD và sử dụng lệnh dưới đây

sudo gem install mqtt

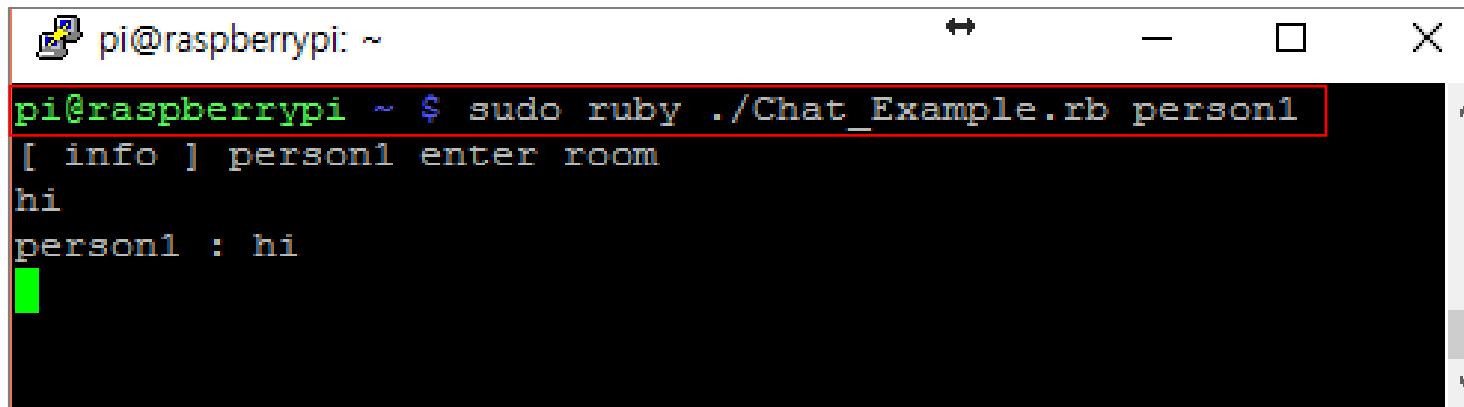
- ▶ Cài đặt thư viện MQTT từ gói cài đặt Rubygems



```
pi@raspberrypi: ~
pi@raspberrypi ~ $ sudo gem install mqtt
Fetching: mqtt-0.4.0.gem (100%)
Successfully installed mqtt-0.4.0
1 gem installed
Installing ri documentation for mqtt-0.4.0...
Installing RDoc documentation for mqtt-0.4.0...
pi@raspberrypi ~ $ ls
Chat_Example.rb      node_mobius          send_example
gw_sx1272           receive_example
Malicious_Example   RF96(SX1272)_DEVICE_miniGateWay
pi@raspberrypi ~ $
```

| Khởi động chương trình trò chuyện MQTT bằng lệnh dưới đây

- ▶ sudo ruby ./filename username



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The command entered is "sudo ruby ./Chat\_Example.rb person1". The output shows:

```
pi@raspberrypi ~ $ sudo ruby ./Chat_Example.rb person1
[ info ] person1 enter room
hi
person1 : hi
```

Bài 2.

# Giao thức truyền thông

- | 2.1. Giao thức là gì
- | 2.2. Mô hình OSI
- | 2.3. Giao thức truyền thông TCP/IP
- | 2.4. HTTP
- | 2.5. MQTT
- | **2.6. CoAP**
- | 2.7. WebSocket

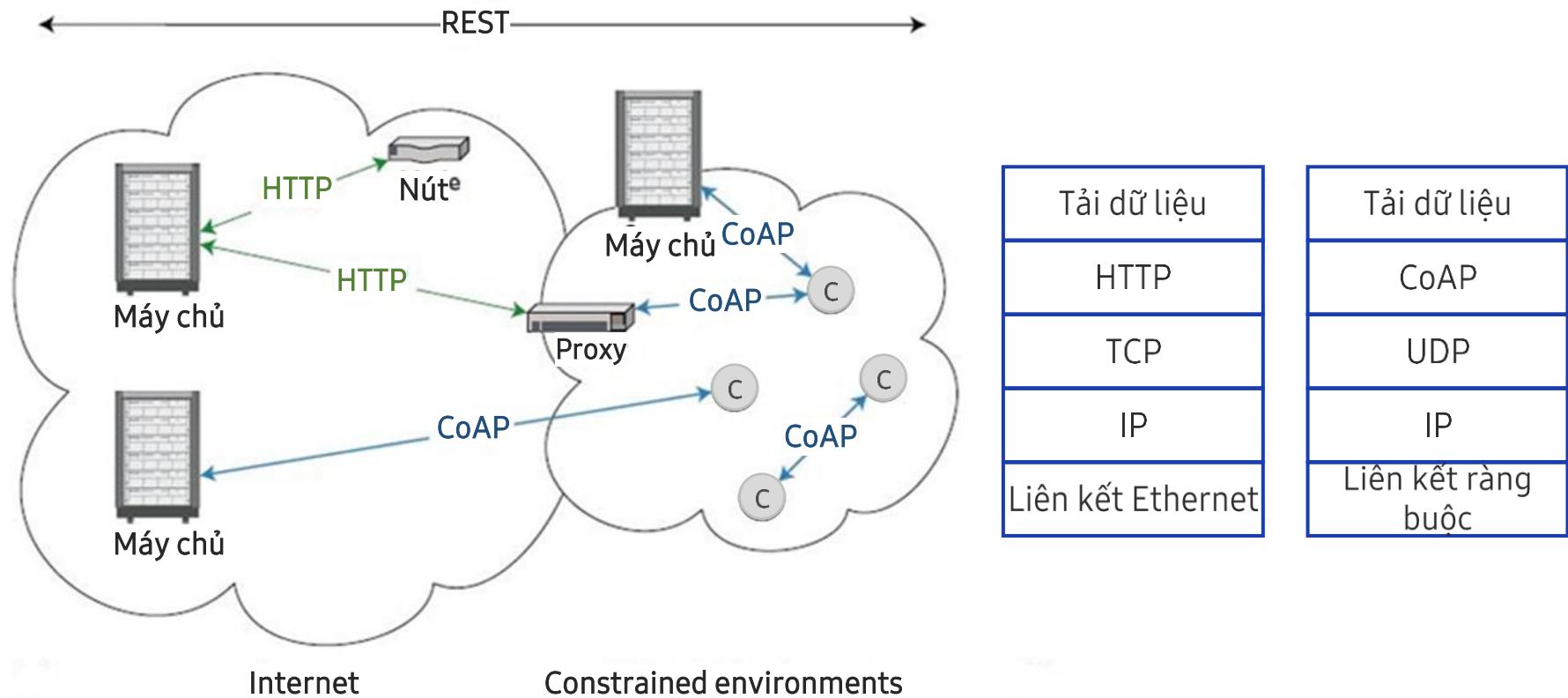
## Khái niệm CoAP

- | Giao thức ứng dụng ràng buộc (CoAP) là một giao thức phổ biến để hỗ trợ các dịch vụ web RESTful của các thiết bị cảm biến hoạt động trên các phần cứng cấp thấp trong môi trường mạng với độ trễ và tỉ lệ mất dữ liệu cao.
- | Đây là một ch่อง giao thức thấp hơn dựa trên tiêu chuẩn IEEE 802.15.4 và tầng mạng sử dụng IPv6.

## Tính năng của CoAP

- | Phương pháp tiếp cận sử dụng RESTful cho phép chuyển đổi và tích hợp dễ dàng với các giao thức HTTP hiện tại.
- | Khả năng thông điệp bị phân mảnh thấp
- | Hỗ trợ Unicast là Multicast trong môi trường UDP.
- | CoAP được định nghĩa bởi bốn kiểu thông điệp: có thể xác nhận, không thể xác nhận, báo nhận và đặt lại.
- | CoAP là một giao thức thay thế cho HTTP chứ không phải khả năng nén HTTP nói chung.

## Môi trường hoạt động của CoAP



## Thông điệp mà CoAP

### | Có thể xác nhận (CON)

- ▶ Được sử dụng để biết được một thông điệp hoặc phản hồi cho một thông điệp đã gửi thành công hay chưa.
- ▶ Nhận một thông điệp ACK/RST như một thông điệp phản hồi.

### | Báo nhận (ACK)

- ▶ It is divided into ACK message and message ID (mid) for CON message.
- ▶ Receives an ACK/RST message as a Phản hồi message.

### | Đặt lại (RST)

- ▶ Nhận một thông điệp CON từ máy khách nhưng thông báo lỗi quá trình.

**Ex** ▶ Ví dụ: Khởi động lại nút

### | Không thể xác nhận (NON)

- ▶ Kiểu thông điệp không yêu cầu ACK

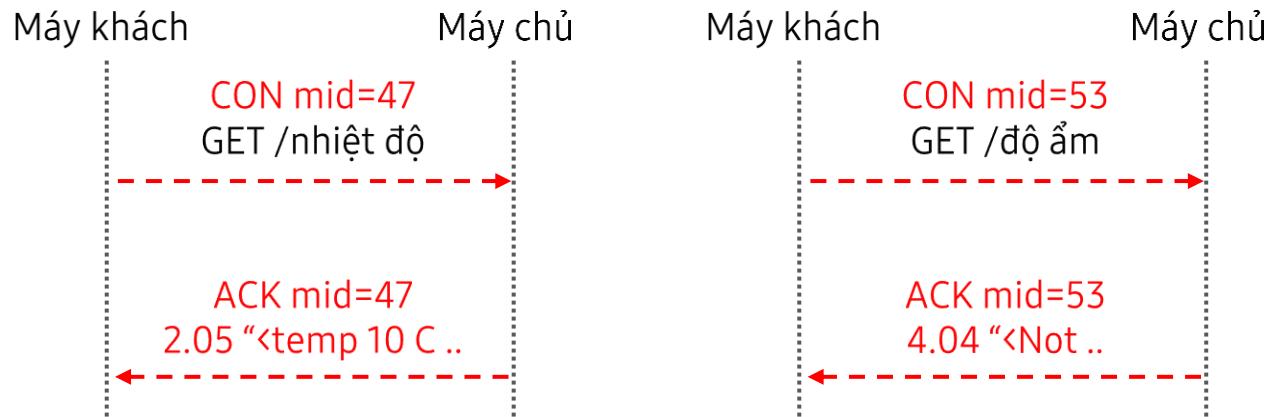
**Ex** ▶ Reading repetitive value from sensor

## Phương thức CoAP

- | GET: thu thập thông tin về mã định dạng tài nguyên bằng URI đã yêu cầu
- | POST: Yêu cầu cập nhật hoặc tạo tài nguyên cho máy chủ theo URI được yêu cầu
- | PUT: Yêu cầu cập nhật mã định dạng tài nguyên với phần nội dung thông điệp theo URI được yêu cầu
- | DELETE: Yêu cầu xóa một mã định dạng tài nguyên từ URI được yêu cầu

## Trao đổi thông điệp CoAP

- | Tương tác CoAP tương tự như mô hình khách/chủ
- | Hỗ trợ URI và Content-type giống như HTTP
- | Trao đổi thông điệp cũng tương tự như Kiểu thông điệp HTTP không cần ACK
  - ▶ Máy khách khởi tạo yêu cầu hành động
  - ▶ Máy chủ phản hồi đến một tài nguyên thông qua mã định dạng URI
  - ▶ Được gửi với một mã phản hồi trong thông điệp phản hồi



<https://tools.ietf.org/html/rfc7252>

## Mô hình giao dịch CoAP

### | UDP (Giao thức gói dữ liệu người dùng)

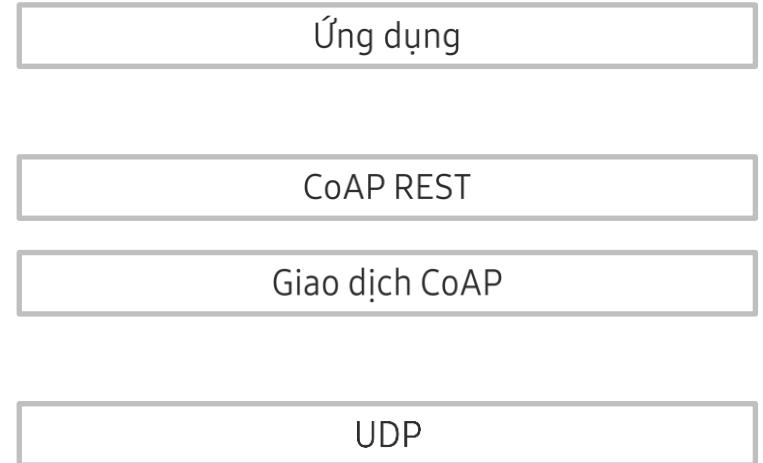
- ▶ CoAP được xác định sử dụng mô hình này với UDP

### | Giao dịch

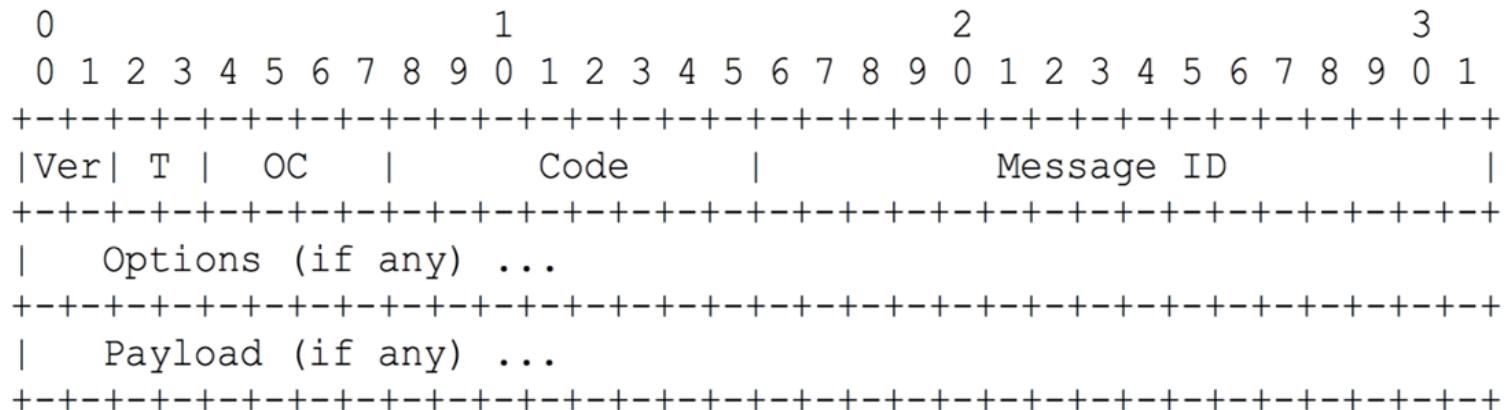
- ▶ Trao đổi một thông điệp với điểm cuối
- ▶ CON, NON, ACK, RST

### | REST

- ▶ Trao đổi một thông điệp với điểm cuối
- ▶ Mã phản hồi và các tùy chọn (URI, content-type)



## Định dạng thông điệp CoAP



Ver: Phiên bản,

T: Kiểu giao dịch (CON, NON, ACK...)

OC: Số lựa chọn

Code: Phương pháp hoặc mã phản hồi

Message ID: Một ID duy nhất được gắn bởi nguồn phát

## Mã trả lại thông điệp CoAP

	coap-03		coap-06	
	Code	Description	Code	Description
Request	1	GET	1	GET
	2	POST	2	POST
	3	PUT	3	PUT
	4	DELETE	4	DELETE
Response	40	100 Continue	-	-
	80	200 OK	69	2.05 Content
	81	201 Created	65	2.01 Created
	-	-	66	2.02 Deleted
	124	304 Not Modified	67	2.03 Valid
	-	-	68	2.04 Changed
	160	400 Bad Request	128	4.00 Bad Request
	-	-	129	4.01 Unauthorized
	-	-	130	4.02 Bad Option
	-	-	131	4.03 Forbidden
	164	404 Not Found	132	4.04 Not Found
	165	405 Method Not Allowed	133	4.05 Method Not Allowed
	-	-	141	4.13 Request Entity Too Large
	175	415 Unsupported Media Type	143	4.15 Unsupported Media Type
	200	500 Internal Server Error	160	5.00 Internal Server Error
	-	-	161	5.01 Not Implemented
	202	502 Bad Gateway	162	5.02 Bad Gateway
	203	503 Service Unavailable	163	5.03 Service Unavailable
	204	504 Gateway Timeout	164	5.04 Gateway Timeout
	-	-	165	5.05 Proxying Not Supported
	240	Token Option required	-	-
	241	Uri-Authority Option required	-	-
	242	Critical Option not supported	-	-

Source: Thomas Pötsch, ComNets, Master Thesis, 2011

Cấu trúc mã phản hồi  
8 bit

```
0 1 2 3 4 5 6 7
+-+-+-----+
| class | detail |
+-+-+-----+
```

3 cấp:

- 2 – Thành công
- 4 – Lỗi máy khách
- 5 – Lỗi máy chủ

```
0 1 2 3 4 5 6 7
+-+-+-----+
| class | detail |
+-+-+-----+
| 1 0 0 | 0 1 0 0 |
+-+-+-----+
```

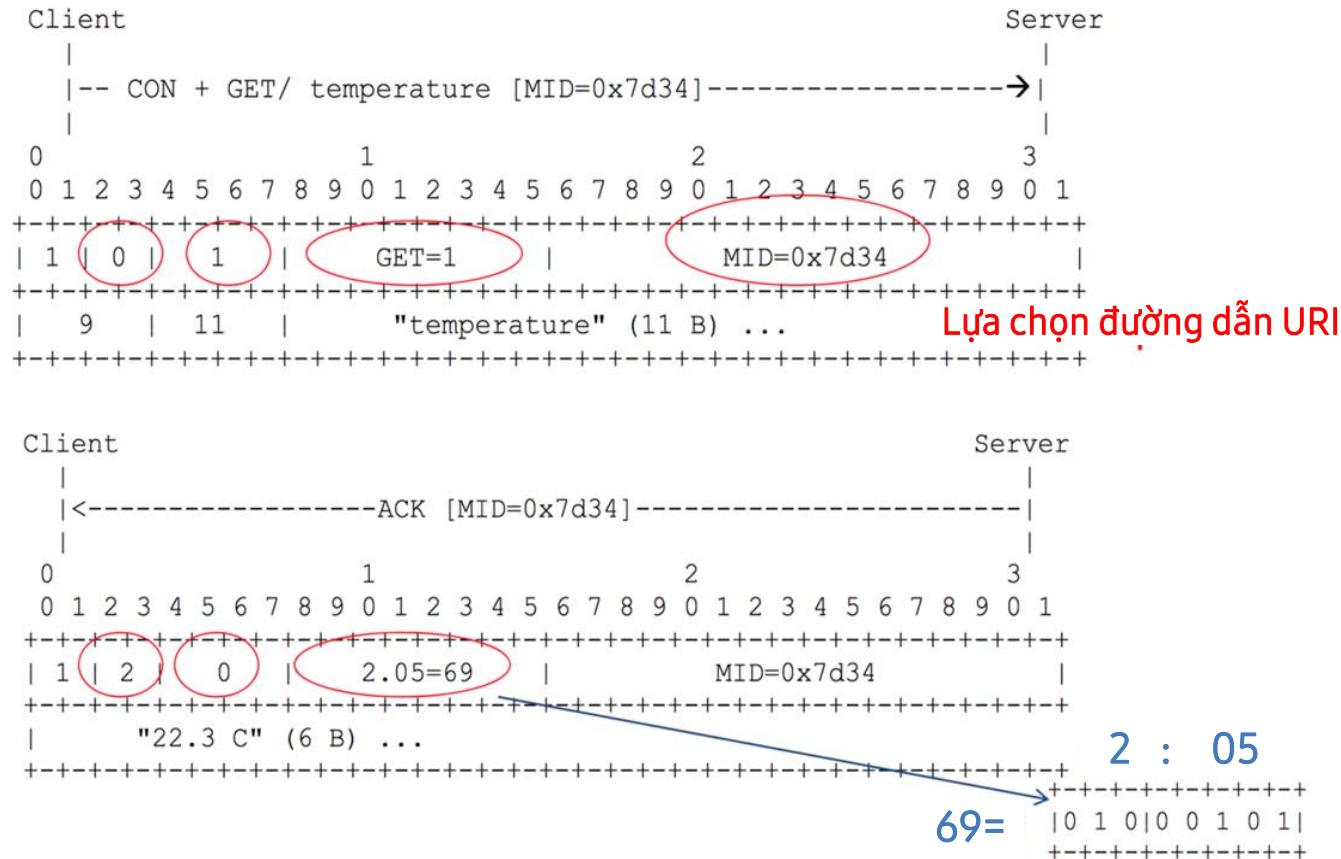
**4 : 04**

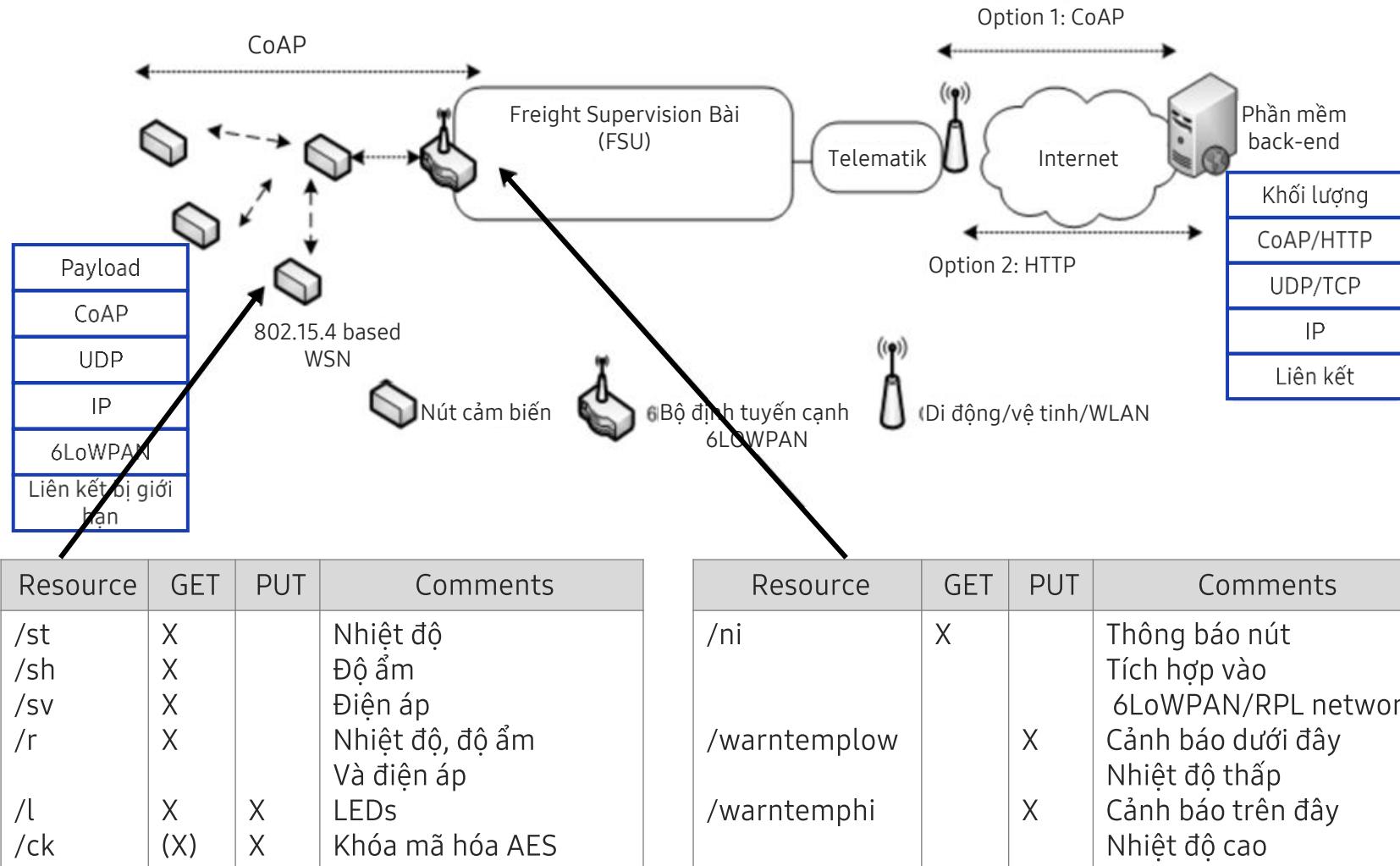
E.g., “Not Found” được viết là  
4.04 – chỉ giá trị thập phân  
132.

## Các tùy chọn CoAP

- | Content-Type: Chỉ ra định dạng của tải dữ liệu ứng dụng
- | Max-Age: Tuổi thọ tối đa để một thông điệp được lưu trong bộ nhớ đệm
- | Proxy-Uri: Xác định một absolute Uri cho một máy chủ proxy
- | Token: Xác định một absolute Uri cho một máy chủ proxy
- | Uri-Host: Chỉ ra máy chủ Internet của một tài nguyên (chỉ được bổ sung nếu không ghi địa chỉ IP đích của yêu cầu)
- | Uri-Path: Chỉ ra tài nguyên của máy chủ
- | Uri-Port: Chỉ ra cổng của máy chủ (chỉ bổ sung nếu khác với đích của yêu cầu cổng UDP)
- | Uri-Query: Chỉ ra các tùy chọn khác cho yêu cầu
- | v.v.

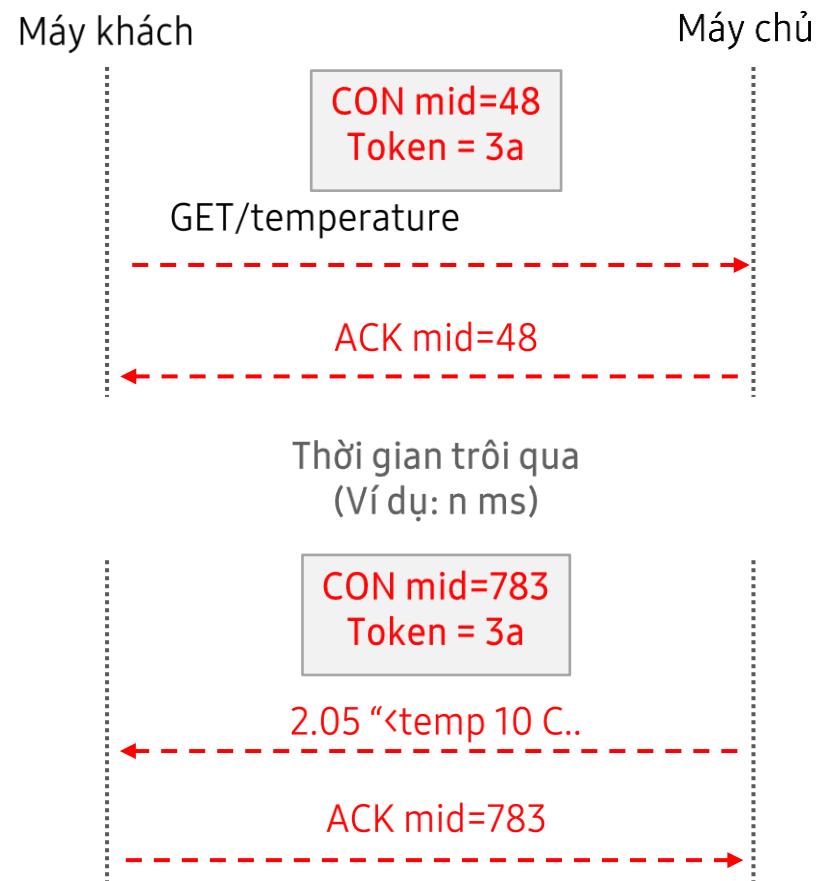
## Ví dụ về thông điệp và hành động trong CoAP





## CoAP và HTTP

- | Hỗ trợ truyền dữ liệu đơn hướng và đa hướng đáng tin
- | cậy qua giao thức UDP
  - ▶ Truyền lại, kiểm soát tắc nghẽn đơn giản
- | Trao đổi thông điệp bất đồng bộ
- | Hai tầng con giữa tầng ứng dụng và giao thức UDP
- | Máy chủ cung cấp một danh sách các liên kết thông qua URI tài nguyên phổ biến.  
(Khám phá tài nguyên)



### MQTT

- ▶ Kết nối với máy chủ broker trung tâm bằng việc xuất bản đăng ký (M: N)
- ▶ Máy chủ broker nhận thông điệp và sao chép thông điệp đó cho subscriber
- ▶ Hỗ trợ kết nối liên tục, truyền dữ liệu theo thời gian thực
- ▶ Giao thức sử dụng TCP/IP (sử dụng TCP)
- ▶ Không có chức năng khám phá dịch vụ, cần giúp đỡ như DNS - SD, SSDP

### CoAP

- ▶ Cách một máy chủ và một máy khác tham gia (1: 1)
- ▶ Không phù hợp cho giao tiếp dữ liệu dựa trên sự kiện
- ▶ Phù hợp cho các thông tin trạng thái trong một môi trường phân tán
- ▶ Chỉ sử dụng giao thức UDP
- ▶ Tính năng khám phá dịch vụ tích hợp

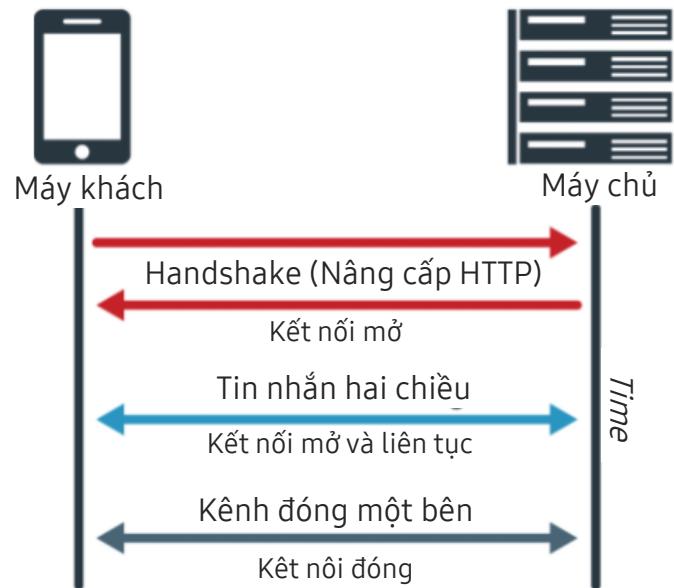
Bài 2.

# Giao thức truyền thông

- | 2.1. Giao thức là gì
- | 2.2. Mô hình OSI
- | 2.3. Giao thức truyền thông TCP/IP
- | 2.4. HTTP
- | 2.5. MQTT
- | 2.6. CoAP
- | **2.7. WebSocket**

## Khái niệm WebSocket

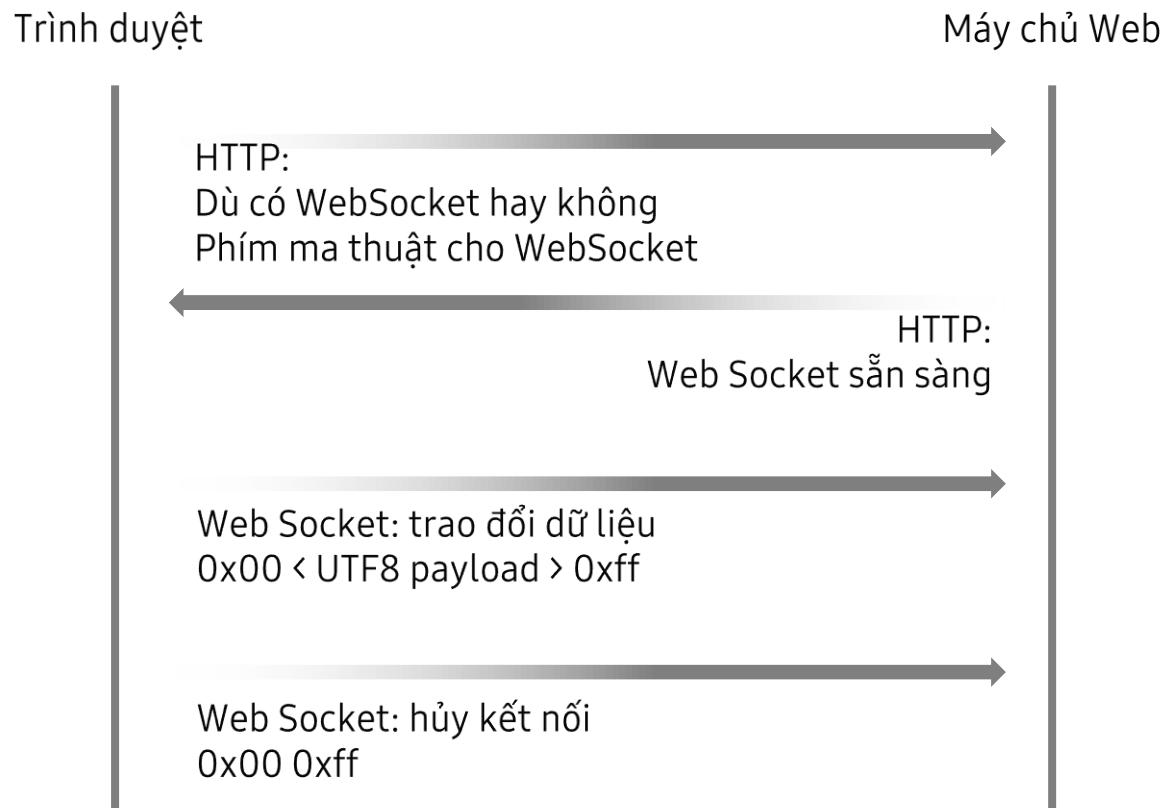
- | Giao thức \giải quyết các thiếu sót của HTTP
- | HTTP không duy trì kết nối. Khi có mươi yêu cầu được gửi đến, để tạo ra 10 kết nối, sẽ có 10 kết nối khác bị hủy bỏ. Trong mọi yêu cầu, tệp tiêu đề đều được sao chép và đi kèm. Giao thức này cũng không có tính tương tác theo thời gian thực.
- | Web Socket hỗ trợ giao tiếp hai chiều thời gian thực và chỉ chấp nhận các luồng thông điệp có định dạng UTF8.



## Các tính năng của Web Socket

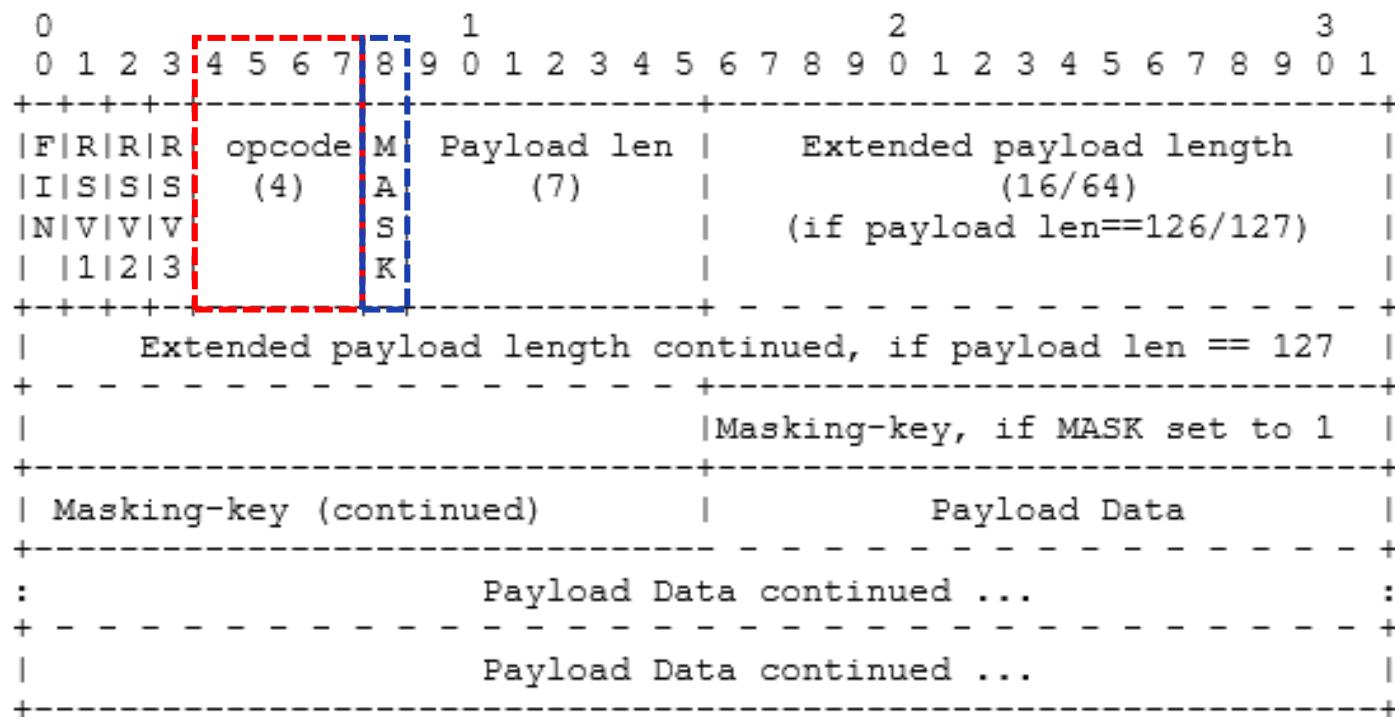
- | Giao tiếp web hai chiều tốc độ cao theo thời gian thực
- | Dễ dàng chứa được nhiều người dùng cùng một lúc
- | Tạo khung dữ liệu bằng HTML5, hỗ trợ giao tiếp bằng văn bản
- | Thay thế cho Comet, một phương pháp đẩy dữ liệu qua HTTP để xử lý dữ liệu theo thời gian thực

## Quy trình WebSocket

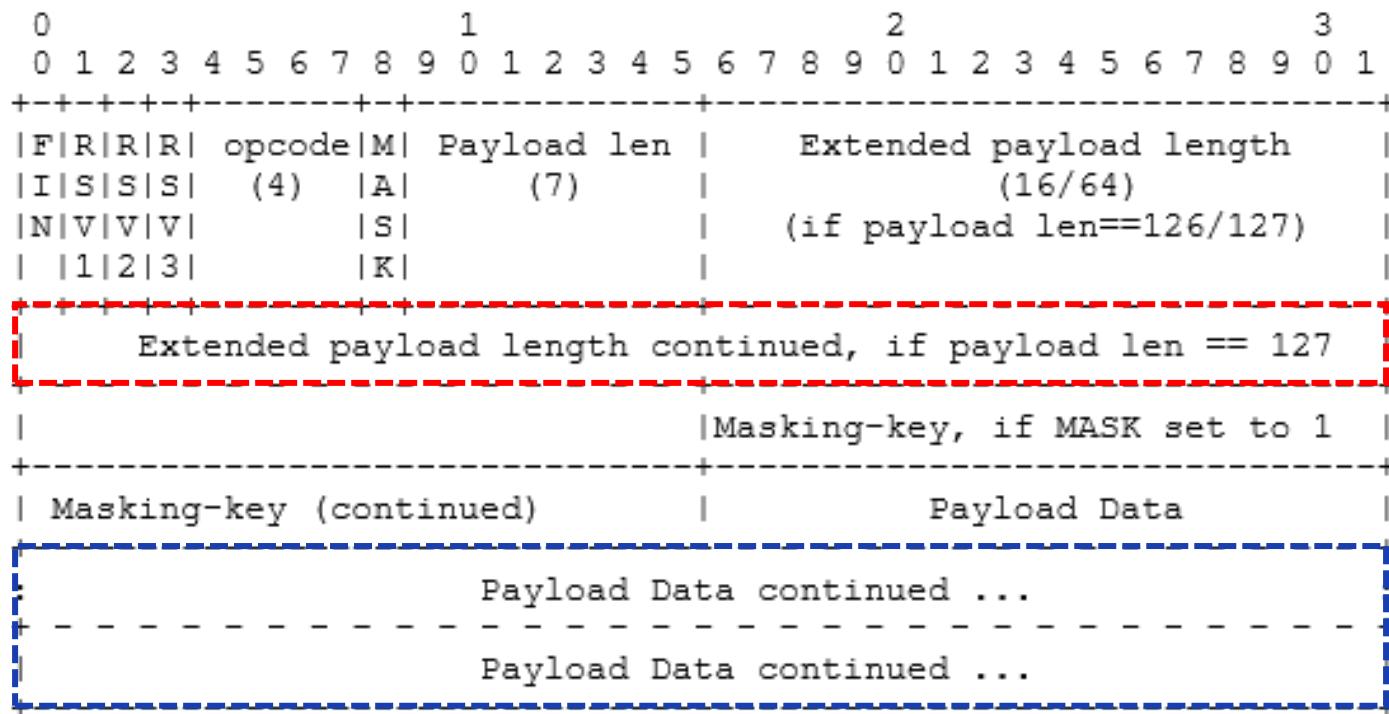


# Định dạng thông điệp WebSocket

- | Mã thực thi: Trạng thái của frame hiện tại
  - | Mật mã: Nếu giá trị của trường tương ứng là 1, trường tải dữ liệu sẽ được XOR thành trường khóa mật mã.



- | Độ dài tải dữ liệu: Độ dài của tải dữ liệu
- | Tải dữ liệu: Trường giá trị dữ liệu



Bài 3.

# Lập trình Socket

## | 3.1. Socket

- | 3.2. Lập trình TCP/IP Socket
- | 3.3. Lập trình UDP Socket
- | 3.4. Lập trình đa hướng

## Lập trình Socket là gì?

- | Lập trình ổ cắm là một điểm cuối của liên kết giao tiếp hai chiều giữa hai chương trình chạy trên (một nút trong) mạng máy tính. Một socket (máy chủ) lắng nghe trên một cổng và địa chỉ IP cụ thể, trong khi một ổ cắm khác (máy khách) kết nối với máy chủ lắng nghe để đạt được giao tiếp.
- | The cách socket gửi dữ liệu được kiểm soát bởi hai thuộc tính:
  - ▶ họ địa chỉ, xác định giao thức lớp mạng được sử dụng; và
  - ▶ Loại socket xác định giao thức lớp vận chuyển được sử dụng.

## Các loại Socket

- | Tùy thuộc vào giao thức tầng vận chuyển được sử dụng, ổ cắm có thể là SOCK\_DGRAM hoặc SOCK\_STREAM.
  - ▶ **SOCK\_DGRAM** để vận chuyển gói dữ liệu hướng thông báo: Các ổ cắm này thường được liên kết với Giao thức gói dữ liệu người dùng (UDP) cung cấp việc gửi các thông báo riêng lẻ không đáng tin cậy. Ổ cắm datagram thường được sử dụng khi thứ tự của các thông báo không quan trọng, chẳng hạn như khi gửi cùng một dữ liệu tới nhiều máy khách.
  - ▶ **SOCK\_STREAM** dành cho truyền tải theo hướng luồng thường được liên kết với Giao thức điều khiển truyền dẫn (TCP): TCP cung cấp khả năng phân phối byte đáng tin cậy và có trật tự giữa hai máy chủ, với việc xử lý và kiểm soát lỗi, giúp ích cho việc triển khai các ứng dụng liên quan đến việc truyền một lượng lớn dữ liệu .

## Mô-đun Socket

| Để sử dụng socket trong Python, cần nhập mô-đun socket.

### Socket đầu vào

- ▶ Cách chính để sử dụng mô-đun socket là thông qua hàm `sockets()` trả về một đối tượng socket m với các phương thức để thực hiện các lệnh gọi socket hệ thống khác nhau. Socket Python hỗ trợ một số họ địa chỉ theo giao thức lớp mạng.
  - `AF_INET`: đây là loại phổ biến nhất và sử dụng IPv4 để đánh địa chỉ mạng. Hầu hết các mạng internet hiện được thực hiện bằng IPv4
  - `AF_INET6`: đây là thế hệ tiếp theo của giao thức internet sử dụng IPv6 và cung cấp một số tính năng không có sẵn trong IPv4.
  - `AF_UNIX`: đây là họ địa chỉ cho Unix Domain Sockets (UDS), một giao thức liên lạc giữa các quá trình có sẵn trên các hệ thống tuân thủ POSIX. Việc triển khai này cho phép truyền dữ liệu giữa các quy trình trên một hệ điều hành mà không cần thông qua mạng.

## Các dịch vụ liên quan đến mạng

- | Các chức năng và phương thức của ổ cắm cung cấp quyền truy cập vào các tác vụ liên quan đến mạng
  - ▶ gethostname(): để lấy tên chính thức của máy chủ hiện tại

```
pi@raspberrypi:~ $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import socket
>>> print(socket.gethostname())
raspberrypi
>>> 
```

- ▶ gethostbyname(): để chuyển đổi tên của máy chủ thành địa chỉ số của nó bằng cách tham khảo cấu hình DNS của hệ điều hành

```
>>> print(socket.gethostbyname("google.com"))
142.250.199.110
>>> 
```

| Các chức năng và phương thức của ổ cắm cung cấp quyền truy cập vào các tác vụ liên quan đến mạng

- ▶ `gethostname_ex()`: để có thêm thông tin đặt tên về một máy chủ

```
>>> name, aliases, addresses = socket.gethostname_ex("google.com")
>>> print("Name : ", name)
('Name : ', 'google.com')
>>> print("Aliases : ", aliases)
('Aliases : ', [])
>>> print("Addresses : ", addresses)
('Addresses : ', ['172.217.31.174'])
>>> 
```

- ▶ `gethostbyaddr()`: để thực hiện tra cứu ngược tên miền

```
>>> socket.gethostbyaddr("172.217.31.174")
('nrt12s22-in-f14.1e100.net', [], ['172.217.31.174'])
>>> 
```

| Các chức năng và phương thức của ổ cắm cung cấp quyền truy cập vào các tác vụ liên quan đến mạng

- ▶ `getfqdn()`: để biết thêm thông tin đặt tên về máy chủ

```
>>> socket.getfqdn("www.google.com")
'nrt12s28-in-t4.1e100.net'
>>> socket.getfqdn()
'raspberrypi'
>>>
```

- ▶ Có thể truy cập nhiều chức năng liên quan đến mạng hơn từ tài liệu python được tham chiếu trong <https://docs.python.org/3/library/socket.html>

Bài 3.

# Lập trình Socket

- | 3.1. Socket
- | **3.2. Lập trình TCP/IP Socket**
- | 3.3. Lập trình UDP Socket
- | 3.4. Lập trình đa hướng

## Giao tiếp máy khách-máy chủ TCP/IP

- | Socket có thể được cấu hình để hoạt động như một máy chủ hoặc máy khách, để đạt được giao tiếp hai chiều qua TCP bằng cách sử dụng họ `SOCK_STREAM`.
- | Chúng tôi sẽ triển khai một ứng dụng echo đơn giản nhận tất cả dữ liệu đến và gửi chúng trở lại người gửi. Vì vậy, chúng tôi sẽ triển khai cả socket máy khách và máy chủ. Hơn nữa, chúng tôi sẽ sử dụng địa chỉ loopback cục bộ 127.0.0.1 hoặc localhost cho các kết nối của chúng tôi.

## Máy chủ Echo

- | Để thiết lập máy chủ, cần thực hiện trình tự các phương thức `socket()`, `bind()`, `listen()` và `accept()`.
  - ▶ `socket()`: tạo một socket mới với họ địa chỉ và loại socket.
  - ▶ `bind()`: liên kết đối tượng socket của chúng ta với một địa chỉ cụ thể bao gồm máy chủ và số cổng.
  - ▶ `listen()`: cho phép máy chủ bắt đầu chấp nhận các kết nối và nhận một đối số, tồn đọng, là số lượng kết nối không được chấp nhận mà hệ thống có thể cho phép trước khi từ chối tất cả các kết nối mới.
  - ▶ `accept()`: chấp nhận các kết nối đến và trả về một bộ (`conn`, địa chỉ) trong đó `conn` là một socket mới có thể được sử dụng để gửi và nhận tin nhắn từ kết nối và địa chỉ là địa chỉ được liên kết với socket ở đầu kia của kết nối.
  - ▶ `close()`: đánh dấu socket là đã đóng và không thể chấp nhận kết nối nữa.

| ~/rasp\_ex/socket/echo\_server.py (1)

- ▶ Khởi tạo một đối tượng socket, `sock` bằng cách chuyển họ địa chỉ (`socket.AF_INET`) và loại socket (`socket.SOCK_STREAM`) cho hàm `socket.socket()`.
- ▶ Liên kết đối tượng ổ cắm với địa chỉ của biểu mẫu ('192.168.0.3', 999). Nó liên kết đối tượng ổ cắm với 192.168.0.3 trên cổng 999, sử dụng phương thức `bind()`.  
(192.168.0.3 là địa chỉ IP của Raspberry Pi.)
- ▶ Lắng nghe các kết nối đến với hồ sơ tồn đọng là 1

```
1 import socket
2
3 # Create a TCP/IP socket
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 # Bind the socket to the port
7 server_address = ('192.168.0.3', 999)
8 print('Starting up on {} port {}'.format(*server_address))
9 sock.bind(server_address)
10
11 # Listen for incoming connections
12 sock.listen(1)
```

## | ~/rasp\_ex/socket/echo\_server.py (2)

- ▶ Gọi `recv(16)` trên kết nối được trả về để nhận dữ liệu trên các khối 16.
- ▶ Nếu dữ liệu đã được nhận, thì hãy truyền dữ liệu đã nhận trở lại người gửi bằng cách gọi phương thức `sendall(data)` trên kết nối, nếu không, chúng tôi sẽ in ra một câu lệnh cho biết không có dữ liệu nào được nhận.
- ▶ Cuối cùng, khi quá trình giao tiếp với máy khách hoàn tất (tất cả các đoạn tin nhắn đã được truyền đi), hãy gọi hàm `close()` trên đối tượng kết nối.
- ▶ Bạn có thể sử dụng khối `try/finally` để đảm bảo rằng `close()` được gọi ngay cả trong trường hợp có lỗi khi truyền tin nhắn.

```
13
14     while True:
15         # Wait for a connection
16         print('waiting for a connection')
17         connection, client_address = sock.accept()
18         try:
19             print('connection from', client_address)
20
21             # Receive the data in small chunks and retransmit it
22             while True:
23                 data = connection.recv(16)
24                 print('received {!r}'.format(data))
25                 if data:
26                     print('sending data back to the client')
27                     connection.sendall(data)
28                 else:
29                     print('no data from', client_address)
30                     break
31
32         finally:
33             # Clean up the connection
34             print("Closing current connection")
35             connection.close()
```

## Máy khách Echo

### I echo\_client.py (1)

- ▶ Khác với máy chủ, máy khách chỉ cần thực hiện chuỗi socket() và connect().
- ▶ Khởi tạo một đối tượng socket như bên dưới trong máy chủ.
- ▶ Sau đó, kết nối socket với cùng một địa chỉ mà máy chủ đang lắng nghe, trong trường hợp này là ('192.168.0.3', 999), sử dụng phương thức kết nối (địa chỉ).  
(192.168.0.3 là địa chỉ IP của Raspberry Pi.)

```
1 import socket
2
3 # Create a TCP/IP socket
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 # Connect the socket to the port where the server is listening
7 server_address = ('192.168.0.3', 999)
8 print('connecting to {} port {}'.format(*server_address))
9 sock.connect(server_address)
```

## Máy khách Echo

### I echo\_client.py (2)

- Trong khối `try/finally`, hãy soạn một tin nhắn dưới dạng chuỗi byte và sử dụng phương thức `sendall()` trên đối tượng ổ cắm với tin nhắn làm đối số.
- Sau đó, thiết lập các biến `số tiền_nhận được` với giá trị ban đầu là 0 và `số tiền_kỳ vọng` là độ dài của tin nhắn, để theo dõi các đoạn tin nhắn khi chúng được nhận.

```
11  try:  
12      # Send data  
13      message = b'It is very long message but will only be transmitted in chunks of 16 at a time'  
14      print('sending {!r}'.format(message))  
15      sock.sendall(message)  
16  
17      # Look for the response  
18      amount_received = 0  
19      amount_expected = len(message)  
20  
21      while amount_received < amount_expected:  
22          data = sock.recv(16)  
23          amount_received += len(data)  
24          print('received {!r}'.format(data))  
25  
26  finally:  
27      print('closing socket')  
28      sock.close()
```

## Máy khách Echo

### I echo\_client.py (3)

- ▶ Gọi `recv(16)` trên đối tượng socket cho phép nhận tin nhắn từ máy chủ theo khối 16 và nó tiếp tục được nhận cho đến khi `amount_received` bằng với `amount_expected`.
- ▶ Cuối cùng, đánh dấu socket là đã đóng.

```
11  try:  
12      # Send data  
13      message = b'It is very long message but will only be transmitted in chunks of 16 at a time'  
14      print('sending {!r}'.format(message))  
15      sock.sendall(message)  
16  
17      # Look for the response  
18      amount_received = 0  
19      amount_expected = len(message)  
20  
21      while amount_received < amount_expected:  
22          data = sock.recv(16)  
23          amount_received += len(data)  
24          print('received {!r}'.format(data))  
25  
26  finally:  
27      print('closing socket')  
28      sock.close()
```

## Chạy máy chủ

### Raspberry Pi

- ▶ sudo python3 ~/rasp\_ex/socket/echo\_server.py
- ▶ Đợi máy khách kết nối

```
> sudo python3 ~/rasp_ex/socket/echo_server_udp.py
```

```
pi@raspberrypi:~ $ sudo python3 rasp_ex/socket/echo_server.py
Starting up on 192.168.0.3 port 999
waiting for a connection
```

### | Windows (1)

- ▶ Bất kể vị trí nào, hãy chạy echo\_client.py

(Trong ví dụ này, echo\_client.py được thực thi bằng PyCharm trong Windows.)

```
C:\Users\dongh\anaconda3\python.exe C:/Users/dongh/Desktop/echo_client.py
connecting to 192.168.0.3 port 999
sending b'It is very long message but will only be transmitted in chunks of 16 at a time'
received b'It is very long '
received b'message but will'
received b' only be transmi'
received b'tted in chunks o'
received b'f 16 at a time'
closing socket

Process finished with exit code 0
```

## | Windows (2)

- ▶ Kiểm tra giao tiếp giữa Raspberry Pi Server và Windows Client.
- ▶ Máy chủ Raspberry Pi nhận được thông báo tương tự sau khi Giao tiếp socket TCP/IP với Máy khách Windows.

```
pi@raspberrypi:~ $ sudo python3 rasp_ex/socket/echo_server.py
Starting up on 192.168.0.3 port 999
waiting for a connection
connection from ('192.168.0.11', 56622)
received b'It is very long '
sending data back to the client
received b'message but will'
sending data back to the client
received b' only be transmi'
sending data back to the client
received b'tted in chunks o'
sending data back to the client
received b'f 16 at a time'
sending data back to the client
received b''
no data from ('192.168.0.11', 56622)
Closing current connection
waiting for a connection
```

Bài 3.

# Socket Programming

- | 3.1. Socket
- | 3.2. Lập trình TCP/IP Socket
- | **3.3. Lập trình UDP Socket**
- | 3.4. Lập trình đa hướng

## Giao tiếp máy khách-máy chủ UDP

- | Không giống như trường hợp truyền TCP truyền thông điệp theo cách có thứ tự, UDP hướng thông điệp và không yêu cầu kết nối lâu dài.
- | Một thông báo trong trường hợp này phải nắn gọn trong một gói dữ liệu và việc gửi không được đảm bảo.

## Máy chủ Echo

| Chỉ thực hiện chuỗi `socket()` và `bind()` vì thực sự không có kết nối nào để lắng nghe.

- ▶ Thay vào đó, chúng ta chỉ cần liên kết socket với một địa chỉ cụ thể và đợi tin nhắn đến.
- ▶ Sau đó, chúng ta sẽ đọc các tin nhắn đến bằng phương thức `recvfrom()` và gửi lại chúng bằng `sendto()`.
- ▶ () nhận dữ liệu từ một ổ cắm và trả về một bộ (`byte`, `địa chỉ`) trong đó `byte` là đối tượng byte chứa dữ liệu nhận được và `địa chỉ` là địa chỉ của người gửi.
- ▶ `sendto(byte, address)` gửi dữ liệu (được cung cấp bởi `byte`) đến một ổ cắm được liên kết với địa chỉ như được xác định bởi `địa chỉ`.

### 3.3. Lập trình UDP Socket

### Bài 03

| ~/rasp\_ex/socket/echo\_server\_udp.py

- ▶ Tạo một đối tượng ổ cắm bằng cách sử dụng `socket.socket(socket.AF_INET,socket.SOCK_DGRAM)`.
- ▶ Xin lưu ý rằng ở đây chúng tôi sử dụng `socket.socket_DGRAM` loại ổ cắm vì chúng tôi đang sử dụng UDP.
- ▶ Tiếp theo liên kết ổ cắm với ('192.168.0.3', 999) và đợi tin nhắn đến.  
(192.168.0.3 là địa chỉ IP của Raspberry)
- ▶ Khi một tin nhắn đến, hãy tiếp tục đọc nó bằng `recvfrom(4096)`, trong đó 4096 là số byte sẽ được đọc và giải nén giá trị trả về thành **dữ liệu và địa chỉ**. Tại thời điểm này, chúng ta có thể in ra độ dài của **dữ liệu**.
- ▶ Nếu một số dữ liệu được nhận, hãy gửi lại cho người gửi bằng phương thức `sendto()` và in ra độ dài của giá trị trả về — là dữ liệu đã gửi.

```
1 import socket
2
3 # Create a UDP socket
4 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
5
6 # Bind the socket to the port
7 server_address = ('192.168.0.3', 999)
8 print('starting up on {} port {}'.format(*server_address))
9 sock.bind(server_address)
10
11 while True:
12     print('\nwaiting to receive message')
13     data, address = sock.recvfrom(4096)
14
15     print('received {} bytes from {}'.format(
16         len(data), address))
17     print(data)
18
19     if data:
20         sent = sock.sendto(data, address)
21         print('sent {} bytes back to {}'.format(
22             sent, address))
```

## Máy khách Echo

### | echo\_client\_udp.py

- ▶ Máy khách này tương tự như máy chủ, chỉ khác là nó không liên kết socket với bất kỳ địa chỉ nào. Thay vào đó, nó sử dụng sendto() để gửi tin nhắn đến địa chỉ của máy chủ.
- ▶ Khởi tạo một đối tượng sock như bạn đã làm trong máy chủ.
- ▶ Sau đó, soạn tin nhắn dưới dạng chuỗi byte và xác định server\_address là một bộ của máy chủ lưu trữ và số cổng được liên kết với máy chủ mà chúng tôi muốn gửi tin nhắn đến.
- ▶ Bên trong khối try/finally, gửi tin nhắn và đợi phản hồi, in dữ liệu trong cả hai trường hợp.
- ▶ Cuối cùng, đánh dấu socket là đã đóng.

```
1 import socket
2
3 # Create a UDP socket
4 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
5
6 server_address = ('192.168.0.3', 999)
7 message = b'This is our message. It will be sent all at once'
8
9 try:
10     # Send data
11     print('sending {!r}'.format(message))
12     sent = sock.sendto(message, server_address)
13
14     # Receive response
15     print('waiting to receive')
16     data, server = sock.recvfrom(4096)
17     print('received {!r}'.format(data))
18
19 finally:
20     print('closing socket')
21     sock.close()
```

## Chạy máy chủ

### | Raspberry Pi

- ▶ sudo python3 ~/rasp\_ex/socket/echo\_server\_udp.py
- ▶ Đợi máy khách kết nối

```
pi@raspberrypi:~ $ sudo python3 rasp_ex/socket/echo_server_udp.py
starting up on 192.168.0.3 port 999
```

```
waiting to receive message
```

#### I Windows (1)

- ▶ Bất kể vị trí nào, hãy chạy echo\_client\_udp.py
- ▶ (Ví dụ sử dụng PyCharm để chạy echo\_client\_udp.py trên Windows.)

```
C:\Users\dongh\anaconda3\python.exe C:/Users/dongh/Desktop/echo_client_udp.py
sending b'This is our message. It will be sent all at once'
waiting to receive
received b'This is our message. It will be sent all at once'
closing socket

Process finished with exit code 0
```

## | Windows (2)

- ▶ Chúng ta có thể kiểm tra giao tiếp giữa máy chủ Raspberry Pi và máy khách Windows.
- ▶ Máy chủ Raspberry Pi nhận được thông báo tương tự sau khi Giao tiếp ổ cắm UDP giữa Máy khách Windows.

```
pi@raspberrypi:~ $ sudo python3 rasp_ex/socket/echo_server_udp.py
starting up on 192.168.0.3 port 999
```

```
waiting to receive message
received 48 bytes from ('192.168.0.11', 57179)
b'This is our message. It will be sent all at once'
sent 48 bytes back to ('192.168.0.11', 57179)
```

```
waiting to receive message
```

Bài 3.

# Socket Programming

- | 3.1. Socket
- | 3.2. Lập trình TCP/IP Socket
- | 3.3. Lập trình UDP Socket
- | **3.4. Lập trình đa hướng**

## Đa hướng

### I Xử lý đa kết nối

- ▶ Khi xử lý nhiều máy khách, việc duy trì một số kết nối điểm-điểm có thể gây khó khăn cho các ứng dụng do nhu cầu xử lý và băng thông tăng lên. Đây là nơi các tin nhắn phát đa hướng xuất hiện.
- ▶ Với phát đa hướng, các tin nhắn được gửi đến nhiều điểm cuối cùng một lúc.
- ▶ Phương pháp này đạt được hiệu quả cao hơn do việc gửi tin nhắn đến tất cả người nhận được ủy quyền cho cơ sở hạ tầng mạng.
- ▶ Tin nhắn multicast được gửi bằng UDP, vì TCP đảm nhận một cặp điểm cuối giao tiếp.
- ▶ Địa chỉ dành cho tin nhắn phát đa hướng, được xác định là nhóm phát đa hướng, là một tập hợp con của địa chỉ IPv4, thường nằm trong khoảng từ 224.0.0.0 đến 239.255.255.255.
- ▶ Các bộ định tuyến và chuyển mạch mạng coi các địa chỉ trong phạm vi này là đặc biệt vì chúng được dành riêng cho phát đa hướng, đảm bảo các thông báo gửi đến nhóm được phân phối tới tất cả các máy khách tham gia nhóm.

## Gửi tin nhắn đa hướng

| Để gửi tin nhắn, chúng tôi sử dụng phương thức `sendto()` thông thường với nhóm phát đa hướng làm địa chỉ

- ▶ Chúng tôi cũng cần chỉ định giá trị Thời gian tồn tại (TTL) để xác định khoảng cách mà tin nhắn sẽ được phát từ người gửi.
- ▶ TTL mặc định là 1, sẽ dẫn đến các tin nhắn chỉ được gửi đến các máy chủ trong mạng cục bộ.
- ▶ Chúng ta sẽ sử dụng `setsockopt()` với tùy chọn `IP_MULTICAST_TTL` để đặt TTL, sẽ được đóng gói thành một byte đơn.
- ▶ Chúng tôi cũng sẽ đặt giá trị thời gian chờ trên ổ cắm, để ngăn nó chờ phản hồi vô thời hạn, vì chúng tôi không biết có bao nhiêu phản hồi mà chúng tôi mong đợi nhận được từ mạng.

## I multicast\_sender.py (1)

- Tạo một socket loại `socket.SOCK_DGRAM`, soạn hoặc nhắn tin dưới dạng chuỗi byte và liên kết socket của chúng tôi với địa chỉ nhóm phát đa hướng ('224.10.10.10', 10000).
- Sau đó đặt thời gian chờ là 0,2 bằng cách sử dụng `sock.settimeout(0.2)`
- Sử dụng mô-đun `struct`, đóng gói số 1 thành một byte và gán byte cho `ttl`.
- Sử dụng `setsockopt()`, đặt tùy chọn `IP_MULTICAST_TTL` của ổ cắm thành ttl mà bạn vừa tạo ở trên và gửi tin nhắn bằng `sendto()`.

```
1 import socket
2 import struct
3
4 message = b'very important data'
5 multicast_group = ('224.10.10.10', 10000)
6
7 # Create the datagram socket
8 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9
10 # Set a timeout so the socket does not block
11 # indefinitely when trying to receive data.
12 sock.settimeout(0.2)
13
14 # Set the time-to-live for messages to 1 so they do not
15 # go past the local network segment.
16 ttl = struct.pack('b', 1)
17 sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
```

## | multicast\_sender.py (2)

```
19     try:
20
21         # Send data to the multicast group
22         print('sending {!r}'.format(message))
23         sent = sock.sendto(message, multicast_group)
24
25     # Look for responses from all recipients
26     while True:
27         print('waiting to receive')
28         try:
29             data, server = sock.recvfrom(16)
30         except socket.timeout:
31             print('timed out, no more responses')
32             break
33         else:
34             print('received {!r} from {}'.format(
35                 data, server))
36
37     finally:
38         print('closing socket')
39         sock.close()
```

- ▶ Sau đó, chờ phản hồi từ các máy chủ khác trên mạng, miễn là thời gian chờ chưa hết và in ra các phản hồi.

## Nhận tin nhắn đa hướng

- | Để nhận tin nhắn, sau khi tạo ổ cắm thông thường của chúng tôi và liên kết nó với một cổng, bạn cần thêm nó vào nhóm phát đa hướng.
- | Điều này có thể được thực hiện bằng cách sử dụng `setsockopt()` để đặt tùy chọn `IP_ADD_MEMBERSHIP`, tùy chọn này sẽ được đóng gói thành một biểu diễn 8 byte của nhóm phát đa hướng và giao diện mạng mà máy chủ sẽ lắng nghe các kết nối.
- | `socket.inet_aton()` nên được sử dụng để chuyển đổi địa chỉ IPv4 của nhóm phát đa hướng từ định dạng chuỗi bốn chấm ('224.10.10.10') sang định dạng nhị phân đóng gói 32 bit.

## | multicast\_receiver.py (1)

```
1 import socket
2 import struct
3 import sys
4
5 multicast_group = '224.10.10.10'
6 server_address = ('', 10000)
7
8 # Create the socket
9 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10
11 # Bind to the server address
12 sock.bind(server_address)
13
14 # Tell the operating system to add the socket to
15 # the multicast group on all interfaces.
16 group = socket.inet_aton(multicast_group)
17 mreq = struct.pack('4sL', group, socket.INADDR_ANY)
18 sock.setsockopt(
19     socket.IPPROTO_IP,
20     socket.IP_ADD_MEMBERSHIP,
21     mreq)
```

- ▶ Để thêm socket vào nhóm phát đa hướng, hãy sử dụng `struct` để đóng gói địa chỉ nhóm do `socket.inet_aton(multicast_group)` cung cấp và giao diện mạng do `socket.INADDR_ANY` cung cấp thành một biểu diễn 8 byte, sau đó chúng tôi đặt thành `IP_ADD_MEMBERSHIP` của socket.

## | multicast\_receiver.py (2)

- ▶ Khi socket nhận tin nhắn, hãy sử dụng `recvfrom(1024)` để giải nén phản hồi thành dữ liệu và địa chỉ, đồng thời gửi xác nhận tới địa chỉ bằng `sendto()`.

```
23     # Receive/respond loop
24     while True:
25         print('\nwaiting to receive message')
26         data, address = sock.recvfrom(1024)
27
28         print('received {} bytes from {}'.format(
29             len(data), address))
30         print(data)
31
32         print('sending acknowledgement to', address)
33         sock.sendto(b'ack', address)
```

#### I Sử dụng 2 dấu nhắc cmd Windows khác nhau (1)

- ▶ 1. chạy multicast\_receiver.py (A cmd)
- ▶ 2. chạy multicast\_sender.py (B cmd)
  - Tại thời điểm này, người gửi sẽ gửi tin nhắn theo phương thức phát đa hướng.
  - Đồng thời, người nhận đã nhận được tin nhắn.

The image shows two separate Command Prompt windows running on Microsoft Windows 10. Both windows have a title bar 'Command Prompt' and a status bar indicating the version is 10.0.19041.1237 and the copyright is (c) Microsoft Corporation. All rights reserved.

**Left Window (Receiver):**

```
Microsoft Windows [Version 10.0.19041.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dongh>cd Desktop
C:\Users\dongh\Desktop>python multicast_receiver.py
waiting to receive message
```

**Right Window (Sender):**

```
Microsoft Windows [Version 10.0.19041.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dongh>cd Desktop
C:\Users\dongh\Desktop>python multicast_sender.py
sending b'very important data'
waiting to receive
received b'ack' from ('192.168.208.1', 10000)
waiting to receive
timed out, no more responses
closing socket
```

In both windows, the command being run is highlighted with a red box. In the receiver window, the command is 'python multicast\_receiver.py'. In the sender window, the command is 'python multicast\_sender.py'.

### | Sử dụng 2 dấu nhắc cmd Windows khác nhau (2)

- ▶ Người nhận sẽ tiếp tục chờ nhận tin nhắn.
- ▶ Người gửi có thể gửi lại tin nhắn.

The image shows two Microsoft Windows Command Prompt windows side-by-side. Both windows have a title bar 'Command Prompt' and are running on 'Microsoft Windows [Version 10.0.19041.1237]'. The left window, titled 'python multicast\_receiver.py', shows the receiver script running. It outputs:

```
Microsoft Windows [Version 10.0.19041.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dongh>cd Desktop
C:\Users\dongh\Desktop>python multicast_receiver.py
waiting to receive message
received 19 bytes from ('192.168.208.1', 50866)
b'very important data'
sending acknowledgement to ('192.168.208.1', 50866)

waiting to receive message
```

The right window, also titled 'Command Prompt', shows the sender script running. It outputs:

```
Microsoft Windows [Version 10.0.19041.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dongh\Desktop>python multicast_sender.py
sending b'very important data'
waiting to receive
received b'ack' from ('192.168.208.1', 10000)
waiting to receive
timed out, no more responses
closing socket

C:\Users\dongh\Desktop>
```

Both windows have their scroll bars visible at the bottom.

Bài 4.

# Hướng dẫn giao tiếp Socket với Raspberry Pi

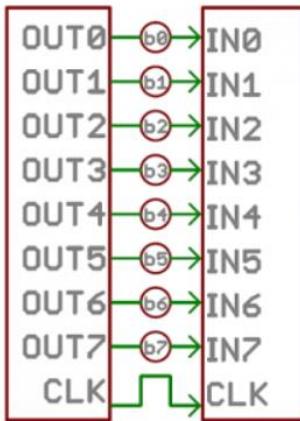
- | 4.1. Truyền thông nối tiếp
- | 4.2. Truyền thông SPI
- | 4.3. Truyền thông Bluetooth
- | 4.4. Máy chủ web Flask

# Truyền thông nối tiếp là gì?

## | Giới thiệu truyền thông nối tiếp

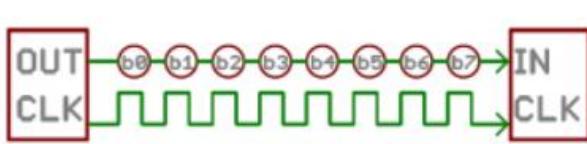
- ▶ Vì cốt lõi của một hệ thống nhúng là giao tiếp và vận hành giữa các bộ xử lý và mạch khác nhau nên nó yêu cầu một giao thức giao tiếp chung để trao đổi dữ liệu.
- ▶ Có nhiều giao thức truyền thông, nhưng nhìn chung, chúng có thể được chia thành hai loại: song song và nối tiếp.

### I Giao thức song song



- ▶ Giao thức song song truyền nhiều bit cùng một lúc.
- ▶ Giao thức song song gửi tất cả dữ liệu cùng một lúc qua 8, 16 hoặc nhiều dây.
- ▶ Do cần có đường đồng hồ (CLK) để tất cả các đường có thể hoạt động cùng nhau theo thời gian (đồng hồ), nên 9 đường được sử dụng cho bus dữ liệu 8 bit.
- ▶ Mặc dù song song dễ thực hiện hơn và nhanh hơn so với nối tiếp, nhưng nó đòi hỏi nhiều tài nguyên và chi phí hơn, đồng thời tiêu tốn nhiều đường I/O (Đầu vào/Đầu ra) hơn.

### I Giao thức nối tiếp



- ▶ Giao thức nối tiếp biến dữ liệu thành một luồng (tuần tự hóa) để gửi từng bit một.
- ▶ Như hình bên trái, giao diện nối tiếp truyền dữ liệu theo 2 dòng bao gồm theo chiều kim đồng hồ đồng hồ.

## Thiết bị nối tiếp

### I Một giao diện truyền thông quan trọng trong hệ thống nhúng

- ▶ Giao tiếp RS232, RS485 và UART đều là các giao thức Truyền thông kết nối không đồng bộ..
- ▶ Trong giao tiếp không đồng bộ, không có đồng hồ giữa người gửi và người nhận. Dữ liệu được nhận biết bởi máy thu thông qua tín hiệu chỉ điểm bắt đầu và điểm kết thúc của dữ liệu.
- ▶ Vì Truyền thông kết nối tiếp tục gửi dữ liệu theo đơn vị bit, BPS được sử dụng làm đơn vị tốc độ truyền của Truyền thông kết nối.

### | UART: Truyền kết nối không đồng bộ Cấu trúc dữ liệu

- ▶ Trước khi gửi dữ liệu, luôn gửi tín hiệu để gửi dữ liệu bằng Start Bit trước.
- ▶ Dữ liệu được gửi khi đường truyền dữ liệu thay đổi từ THẤP sang CAO.
- ▶ Nó chỉ ra điểm bắt đầu của dữ liệu và gửi dữ liệu từng bit theo trình tự thời gian. Khi tất cả 1 byte dữ liệu đã được gửi, nó sẽ gửi Bit chẵn lẻ (có thể không được sử dụng) và Bit dừng.

Bit	1	2	3	4	5	6	7	8	9	10	11
	Start Bit	Dữ liệu Bit (5 ~ 8 bit)							Parity Bit	Sytop Bit	
Dữ liệu	Start	D0	D1	D2	D3	D4	D5	D6	D7	Parity	Stop

### | UART: Truyền kết nối không đồng bộ Cấu trúc dữ liệu

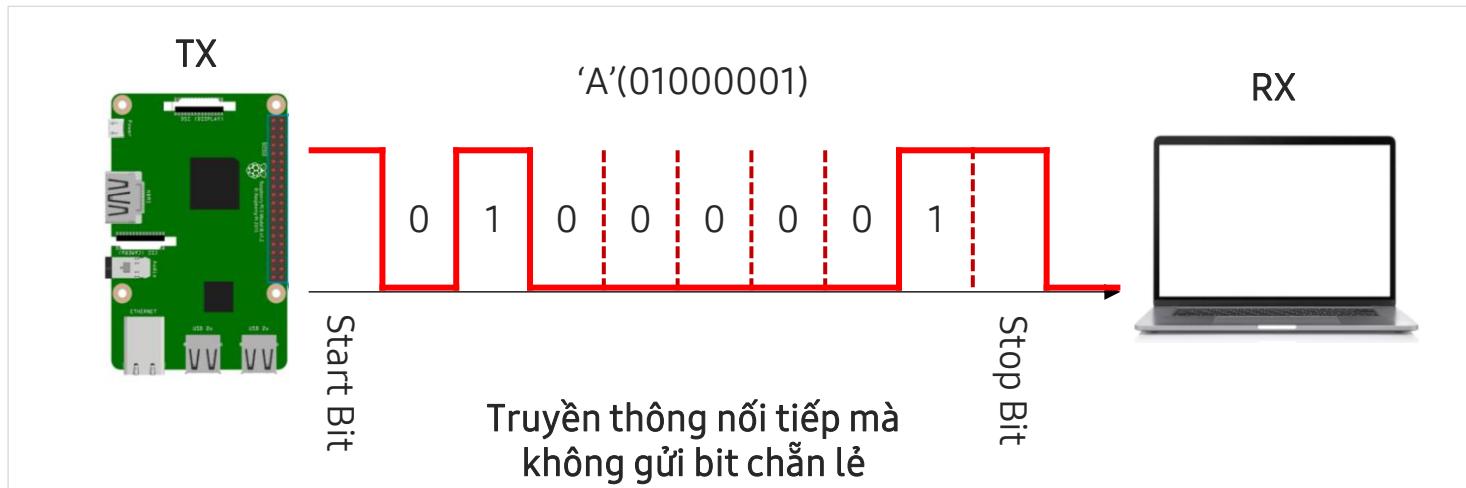
#### Truyền thông nối tiếp Thuật ngữ

- 1) Start Bit: Nó có nghĩa là bắt đầu giao tiếp. Nó duy trì bao nhiêu thời gian để truyền 1 bit. Thời gian truyền 1 bit phụ thuộc vào tốc độ Baud. Tốc độ giao tiếp càng nhanh thì thời gian càng ngắn.
- 2) Bit dữ liệu: 5~8 Bit được chọn và sử dụng, nhưng thông thường 8 Bit được sử dụng nhiều nhất.
- 3) Bit chẵn lẻ: Có thể tạo và gửi một giá trị chẵn lẻ để phát hiện lỗi. Khi bị tắt, bản thân Bit chẵn lẻ không được truyền đi.
- 4) Stop Bit: Thông báo kết thúc truyền dữ liệu

Bit	1	2	3	4	5	6	7	8	9	10	11
	Start Bit	Data Bit (5 ~ 8 bit)								Parity Bit	Stop Bit
Data	Start	D0	D1	D2	D3	D4	D5	D6	D7	Parity	Stop

### I UART: Truyền kết nối không đồng bộ Cấu trúc dữ liệu

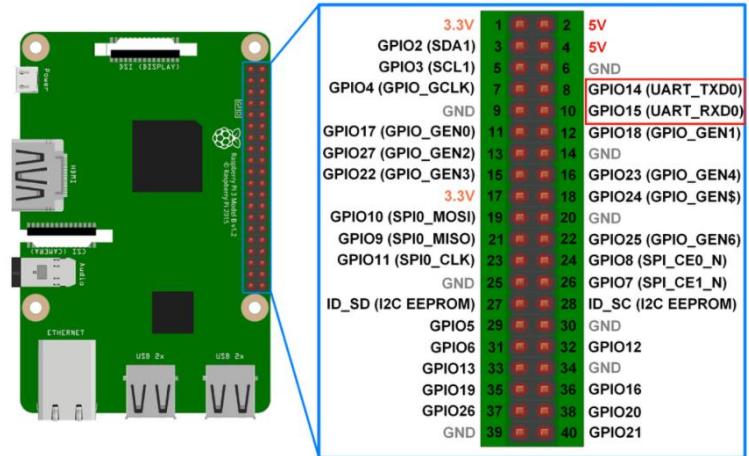
- ▶ Khi gửi ký tự 'A' thông qua Truyền kết nối giữa Raspberry Pi và PC, ký tự 'A' được biểu thị bằng 65 ở dạng thập phân và 10000001 ở dạng nhị phân theo bảng mã ASCII.
- ▶ 1 là CAO, 0 là THẤP, và truyền tín hiệu điện theo thứ tự thời gian.
- ▶ Hình bên dưới khái niệm hóa quá trình truyền không sử dụng bit chẵn lẻ.
  - Người gửi: TX, Người nhận: RX, Tốc độ truyền tin: tỷ lệ Baud
  - Cả bộ phát và bộ thu phải hoạt động trên cùng tốc độ truyền.
  - RS232 là một giao thức truyền thông không đồng bộ không sử dụng đồng hồ và đường dữ liệu.



## Thiết bị nối tiếp Raspberry Pi

Raspberry Pi có 4 kênh USB và 2 kênh UART.

Kênh UART	Cổng nối tiếp	Tên thiết bị	Sử dụng
UART0	Serial0	/dev/ttyS0 Or /dev/serial0	Console or Serial GPIO14(TXD) GPIO15(RXD)
UART1	Serial1	/dev/ttyAMA0 Or /dev/serial1	Bluetooth



Bài 4.

# Hướng dẫn giao tiếp Socket with Raspberry Pi

- | 4.1. Truyền thông nối tiếp
- | **4.2. Truyền thông SPI**
- | 4.3. Truyền thông Bluetooth
- | 4.4. Máy chủ web Flask

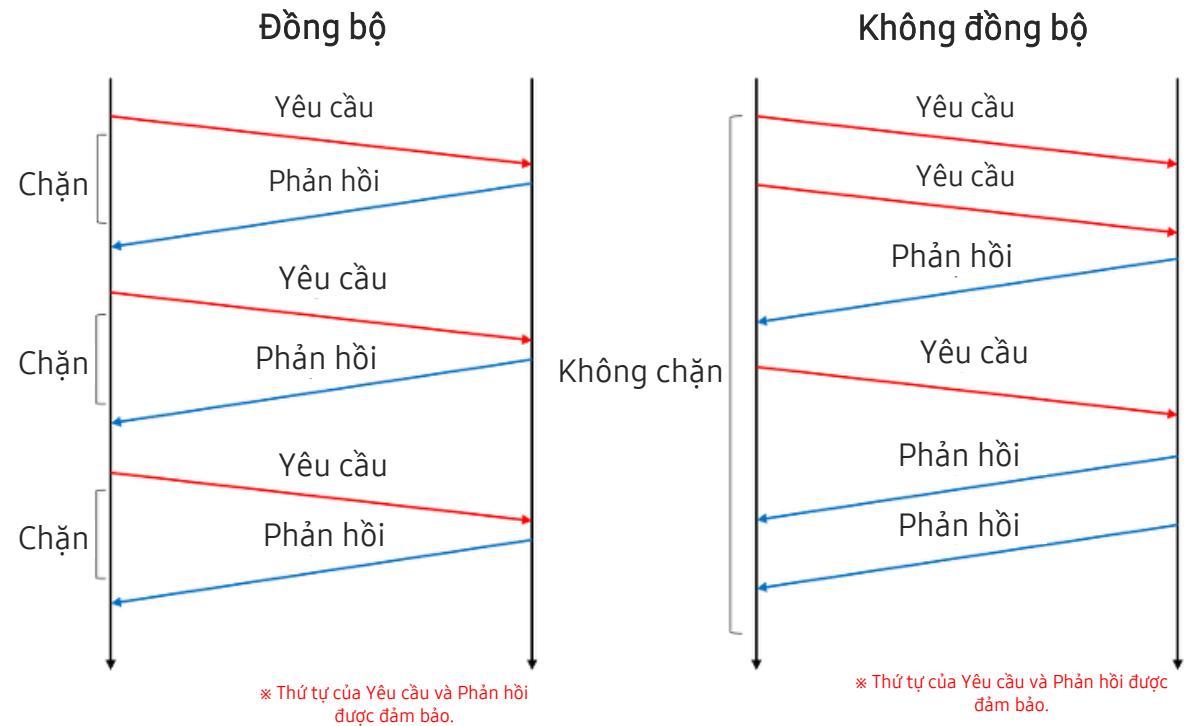
## What is Truyền thông SPI ?

### I SPI (Serial Peripheral Interface) Giao diện ngoại vi nối tiếp

- ▶ SPI là một trong những giao thức Truyền thông kết nối thường được sử dụng trong các bộ vi xử lý như I2C và UART.
- ▶ SPI hỗ trợ giao tiếp một-nhiều.
- ▶ SPI chủ yếu được sử dụng để trao đổi dữ liệu nhanh với thẻ nhớ SD, bộ nhớ nối tiếp, mô-đun giao tiếp mạng.
- ▶ RS232 và UART là các giao thức Truyền thông nối tiếp không đồng bộ.
- ▶ SPI và I2C sử dụng các giao thức Truyền thông nối tiếp đồng bộ.

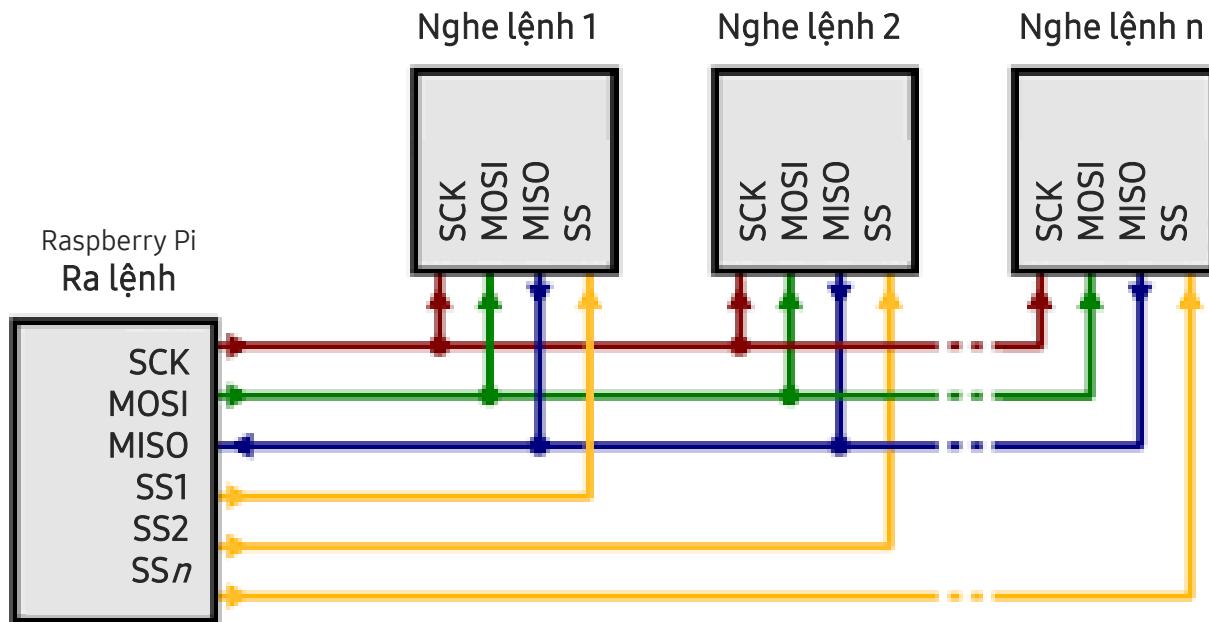
### | So sánh giao thức truyền thông đồng bộ và không đồng bộ

- ▶ Cách dễ nhất để phân biệt giữa truyền thông đồng bộ và không đồng bộ là kiểm tra xem đồng hồ có được sử dụng hay không trong các đường truyền thông.
- ▶ Giao tiếp đồng bộ truyền và nhận dữ liệu bằng cách đồng bộ hóa việc truyền dữ liệu tới đồng hồ.
- ▶ Giao tiếp không đồng bộ không có đồng hồ và chỉ phân biệt dữ liệu dưới dạng tín hiệu cho biết bắt đầu và kết thúc giao tiếp



### I Truyền thông Giao thức SPI

- ▶ SPI là một giao thức Truyền thông nối tiếp đồng bộ với một đồng hồ cho phép một Thiết bị Chính SPI và nhiều Thiết bị Phụ SPI giao tiếp với nhau. Tín hiệu đồng hồ do chính tạo ra và được kết nối với tất cả các phụ.
- ▶ Trong hình bên dưới có 3 Slave Device. Chúng không giao tiếp với Thiết bị Chính cùng một lúc. Trong khi SPI Master (Raspberry Pi) cung cấp đồng hồ, Thiết bị chính chọn Thiết bị phụ muốn Truyền thông SPI bằng tín hiệu Chọn chip.

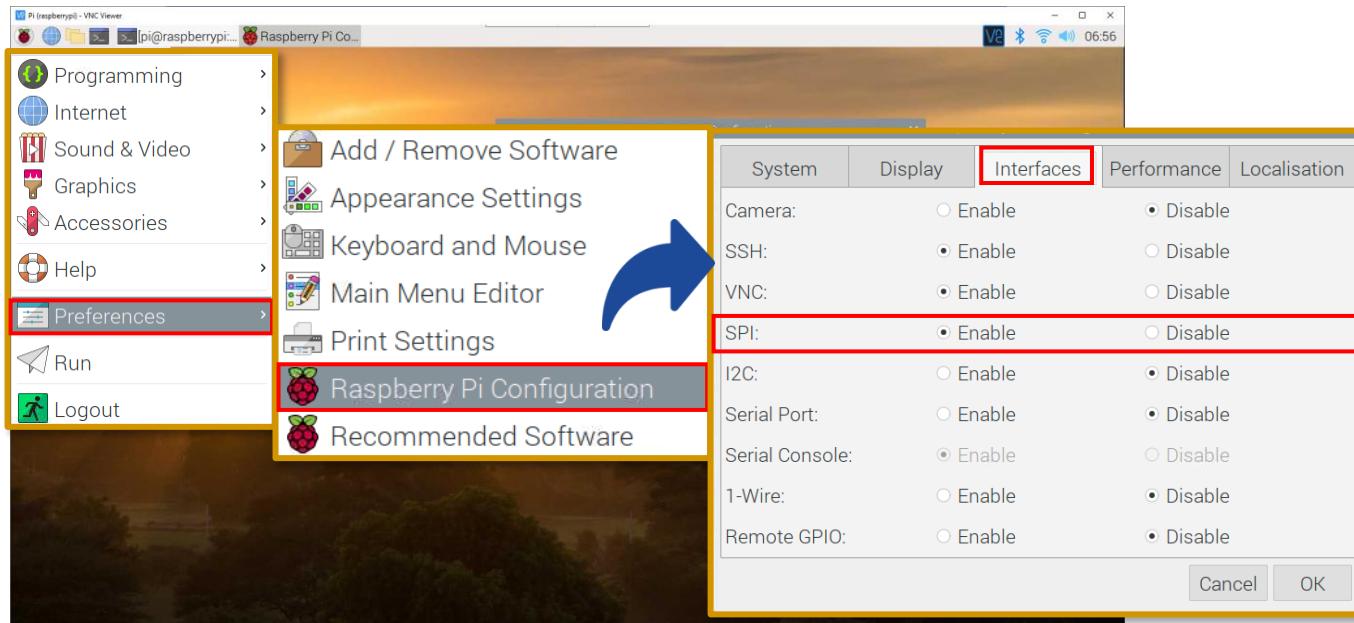


### | Đồng hồ và tín hiệu dùng cho Truyền thông SPI

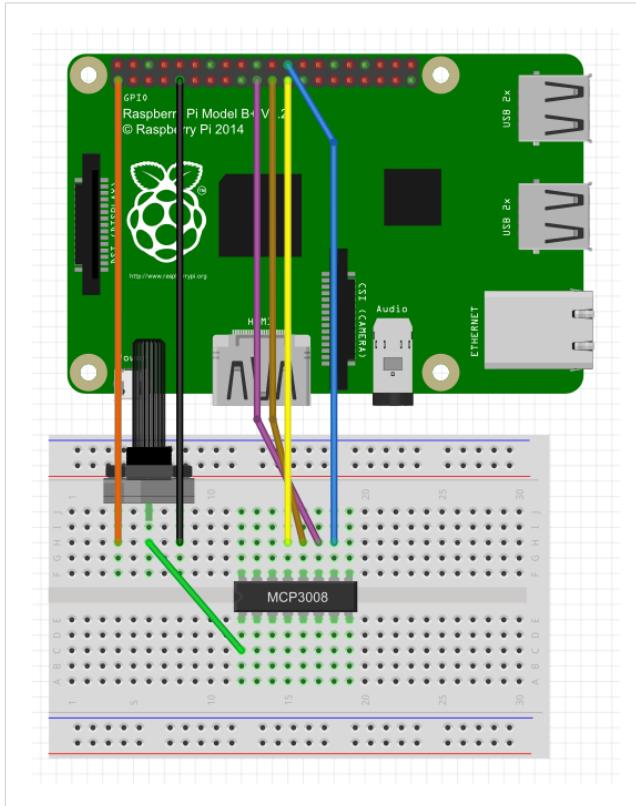
SCK (đồng hồ)	Đồng hồ cho giao tiếp đồng bộ – được cung cấp từ Master
MOSI (Master Out Slave In)	Tín hiệu truyền dữ liệu từ Master đến Slave
MISO (Master In Slave Out)	Là tín hiệu truyền dữ liệu từ Slave đến Master.
	Tín hiệu chọn nô lệ để giao tiếp với chủ khi nhiều nô lệ được sử dụng.
CS (Chip Select)	Nó yêu cầu một tín hiệu CS cho mỗi thiết bị được kết nối. Nó gửi tín hiệu CAO đến thiết bị muốn liên lạc và duy trì tín hiệu THẤP cho các thiết bị còn lại trong quá trình liên lạc.

## Xuất ADC và giá trị điện áp được chuyển đổi bằng SPI

- | Sử dụng Raspberry Pi để đọc ADC (bộ chuyển đổi tương tự sang kỹ thuật số) và giá trị điện áp được chuyển đổi
  - ▶ Có tổng cộng 2 thiết bị, 0 và 1 được kết nối với BUS 0 trong Raspberry Pi.
  - ▶ Sau khi đọc kênh SPI 2 được kết nối với biến trở để nhận giá trị ADC, tính toán giá trị tương tự dưới dạng điện áp và xuất giá trị ADC và giá trị điện áp được chuyển đổi sang bàn điều khiển.
  - ▶ Đối với Truyền thông SPI , hãy thay đổi Cấu hình Raspberry Pi - Giao diện - Truyền thông SPI thành Bật.



### Mạch ví dụ thực hành



- ▶ Khi gắn lên breadboard sau khi vặn nút biến trở lên tương ứng với 3.3v, Out, GND từ trái sang.Variable resistance 3.3v – Raspberry Pi 3.3v
  - kháng biến GND – Raspberry Pi GND
  - kháng biến Out – MCP 3008 Ch0
  - MCP 3008 CLK – Raspberry Pi GPIO11(CLK)
  - MCP 3008 Dout – Raspberry Pi GPIO9(MISO)
  - MCP 3008 Din – Raspberry Pi GPIO10(MOSI)
  - MCP 3008 CS/SHDN – Raspberry Pi GPIO8(CS)

| ~~~~/~~~/adc.py

```
1 import spidev
2 import time
3
4 def analog_read(channel):
5     r = spi.xfer2([1, (0x08+channel)<<4, 0])
6     adc_out = ((r[1]&0x03)<<8) + r[2]
7     return adc_out
8
9 spi = spidev.SpiDev()
10 spi.open(0,0)
11 spi.max_speed_hz = 1000000
12
13 try:
14     while True:
15         adc = analog_read(2)
16         voltage = adc*3.3/1023
17         print("ADC = %d  Voltage = %.3fV" % (adc, voltage))
18         time.sleep(0.5)
19 except KeyboardInterrupt:
20     pass
21 finally:
22     spi.close()
```

- ▶ dòng 1, 2: Sử dụng gói thiết bị SPI và gói hẹn giờ
- ▶ dòng 4: Hàm xác định đọc SPI
- ▶ dòng 5: Đọc kênh SPI được kết nối với Bus 0
- ▶ dòng 9: Khai báo phiên bản thiết bị SPI
- ▶ dòng 10: spi.open(xe buýt, thiết bị)
- ▶ dòng 11: đặt tốc độ SPI
- ▶ dòng 15: Đọc SPI kênh 2(A2)
- ▶ dòng 16: Tính toán giá trị Điện áp sử dụng giá trị Analog
- ▶ dòng 17: Xuất giá trị ADC và Điện áp theo Format
- ▶ dòng 22: đóng thiết bị SPI
- ▶ Lưu tập lệnh python dưới dạng  
~/rasp\_ex/gpio/vareint\_res.py

## | Ví dụ thực thi

- ▶ Thư viện cho Truyền thông SPI được cài đặt theo mặc định, nhưng trong trường hợp nó chưa được cài đặt, hãy chạy lệnh `sudo pip install spidev`.  
▶ > `sudo pip install spidev`
  - ▶ Thực thi lệnh `python3 ~/rasp_ex/gpio/varient_res.py` để chạy tập lệnh Python đã tạo.
  - ▶ Xoay núm biến tròn và kiểm tra đầu ra. Bạn có thể thay giá trị thay đổi tùy thuộc vào núm.
  - ▶ Nếu bạn muốn dừng thực thi, hãy nhấn `Ctrl + C` để ngắt nó.

Bài 4.

## Hướng dẫn giao tiếp Socket with Raspberry Pi

- | 4.1. Truyền thông nối tiếp
- | 4.2. Truyền thông SPI
- | **4.3. Truyền thông Bluetooth**
- | 4.4. Máy chủ web Flask

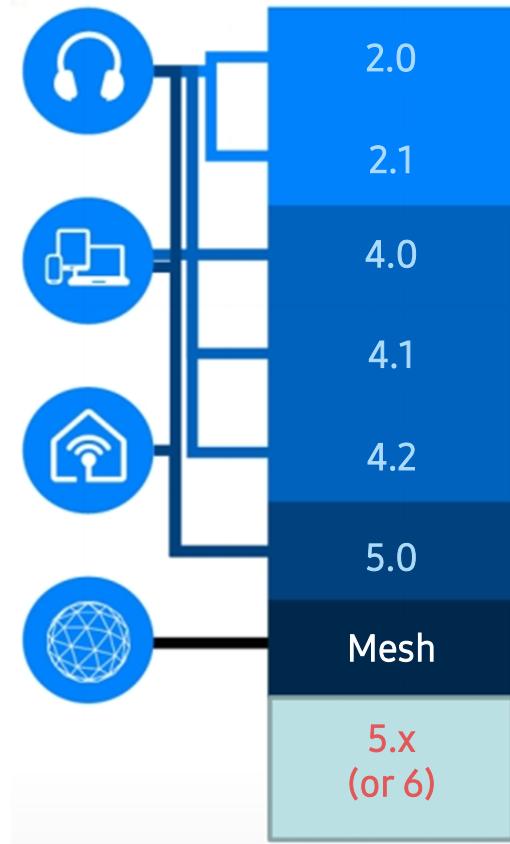
## Bluetooth là gì?

| Công nghệ không dây tầm ngắn kết nối các thiết bị di động để gửi và nhận thông tin

- ▶ Hệ thống không dây của Bluetooth sử dụng dải tần ISM (Industrial Science Medical) 2400 ~ 2483,5 MHz.
- ▶ Raspberry Pi 4 hỗ trợ Bluetooth 5
- ▶ Bluetooth 5 tăng phạm vi, tốc độ và dung lượng dữ liệu của các ứng dụng Bluetooth.
  - Nó cung cấp một giao diện mới giúp hy sinh băng thông cho phạm vi mở rộng đáng kể hoặc sử dụng nhiều băng thông hơn cho phạm vi ngắn.
  - Nó truyền ít dữ liệu hơn trong khoảng cách xa hơn và nhiều dữ liệu hơn trong khoảng cách ngắn.
  - Điều đó cho phép hoạt động năng lượng thấp, và hoạt động linh hoạt dựa trên nhu cầu ứng dụng.

## sự phát triển của công nghệ Bluetooth

| Bluetooth có lợi thế về chi phí thấp, năng lượng thấp và âm thanh về giải pháp không dây IoT cấp thấp.



### Tốc độ cơ bản/Tốc độ dữ liệu nâng cao

- Hầu hết các trường hợp sử dụng liên quan đến kết nối và phát trực tuyến

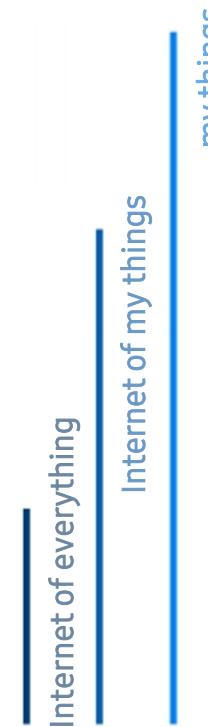
### Bluetooth tiết kiệm năng lượng (4.0, 5.0)

- Các trường hợp sử dụng liên quan đến việc kết nối với mọi thứ để truyền dữ liệu (ví dụ: cảm biến và những thứ tương tự)
- Các trường hợp sử dụng liên quan đến việc phát dữ liệu (quảng cáo bắt đầu cảnh báo)

### Tầm xa 1Km / Tiện ích mở rộng quảng cáo Le 2Mbps

#### Mesh

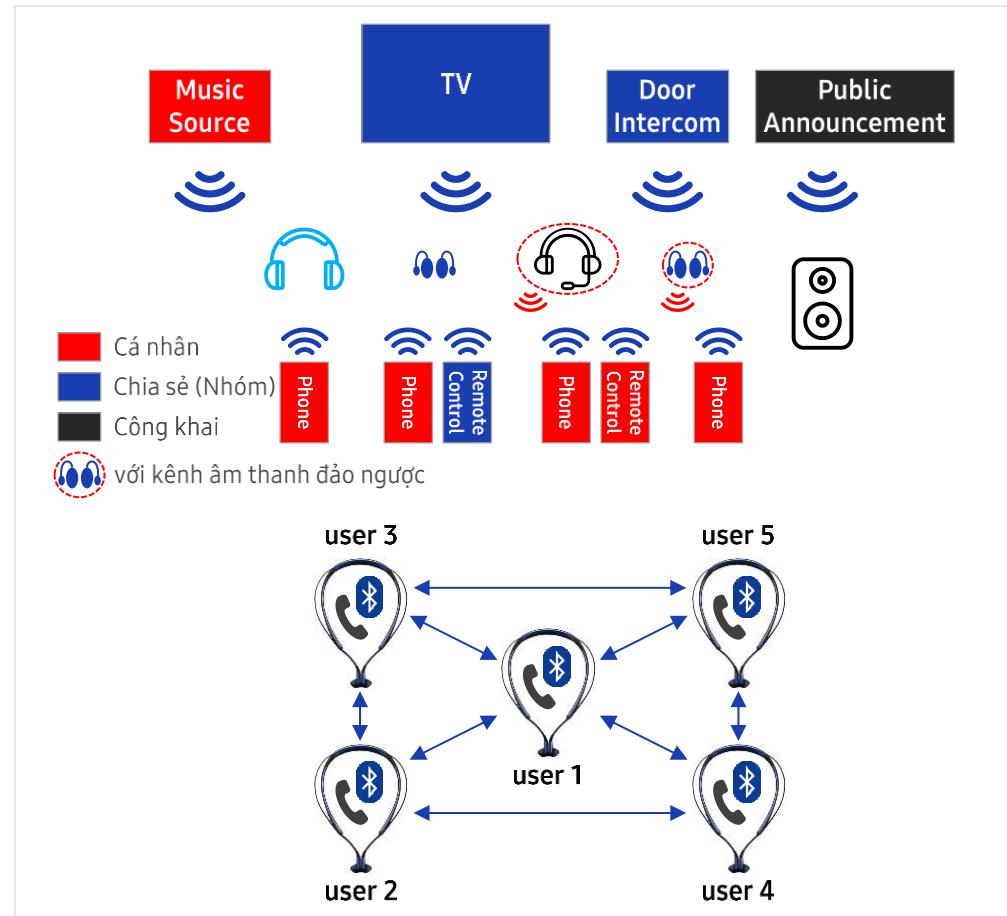
- Cấu trúc liên kết mạng thực sự (dựa trên lưới)
- Lệnh và điều khiển + dữ liệu, KHÔNG băng thông cao
- Âm thanh BLE chất lượng cao công suất thấp (Phát sóng/Multicast, kết nối M:N, nhận dạng giọng nói BLE, Chất lượng cao)
- Đa ăng-ten: Định vị chính xác cao



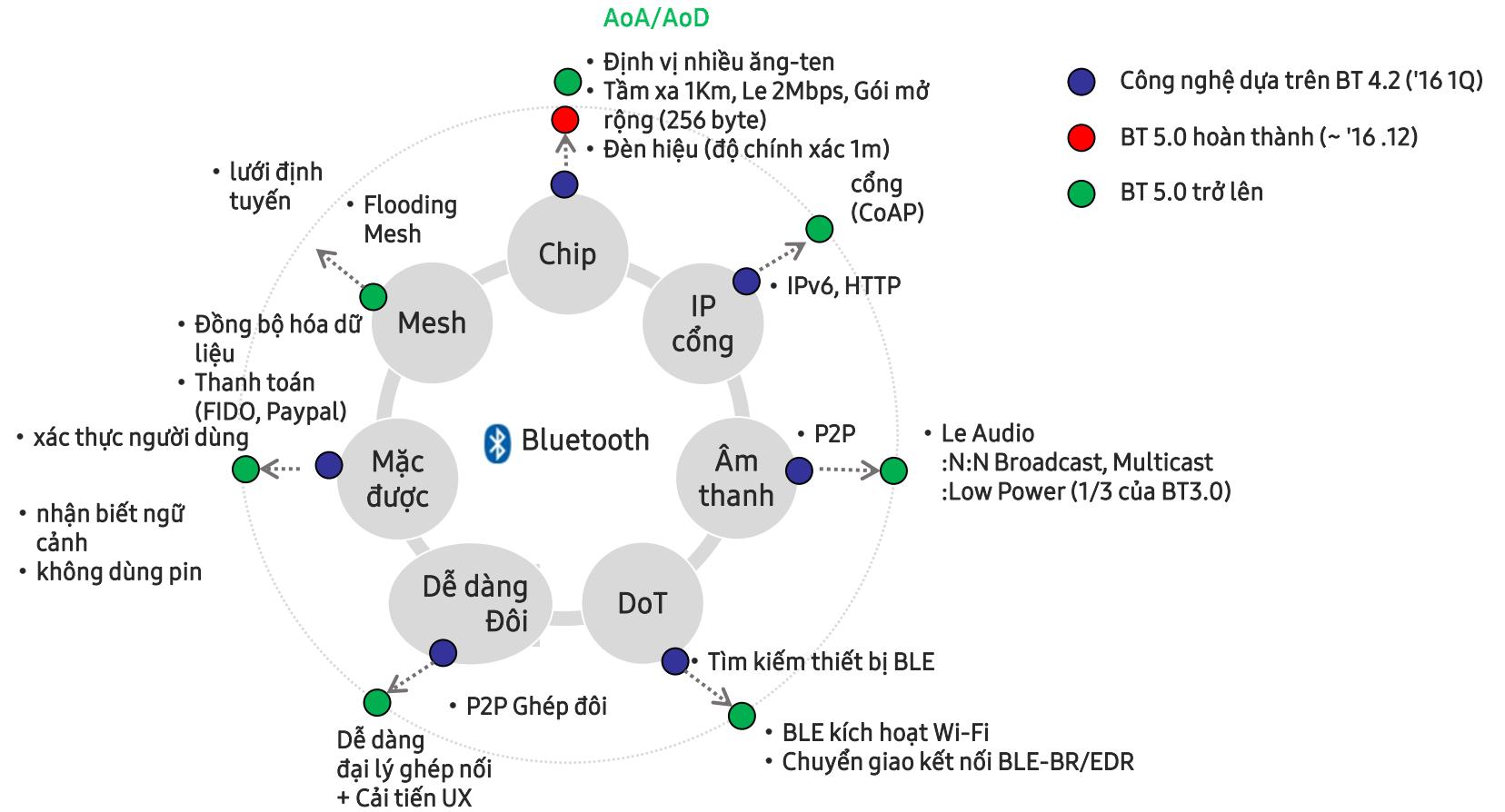
## Âm thanh Bluetooth LE

### I Âm thanh LE

- ▶ Chế độ nhóm: Đa tuyến 1: N, Đa kênh, PTT
- ▶ Điều khiển dễ dàng: Kết nối Easy 1:N, Chuyển đổi Src / Sink, Điều khiển âm lượng
- ▶ Chế độ chia sẻ: Chia sẻ âm nhạc, Chuyển tiếp âm nhạc
- ▶ Hỗ trợ giọng nói: Nhận dạng giọng nói, Nhận dạng giọng nói



## Sự phát triển và ứng dụng của công nghệ Bluetooth



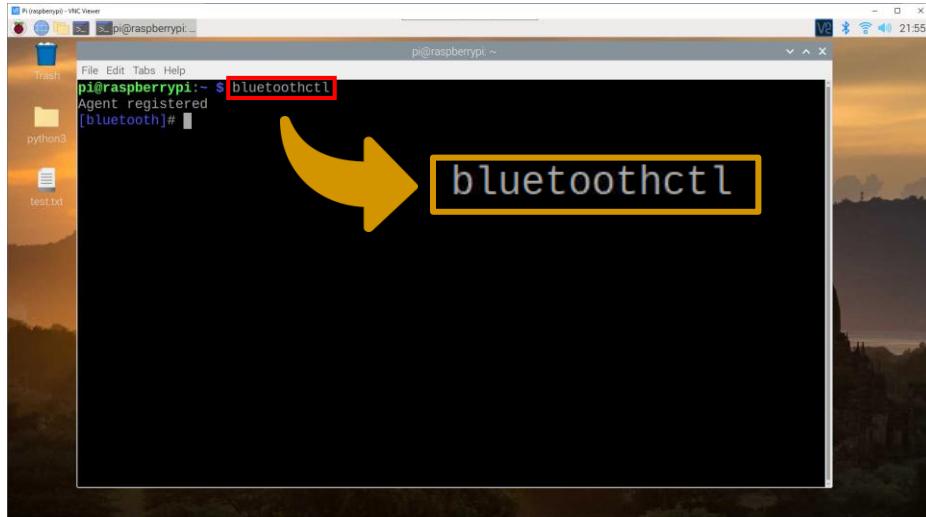
## Thiết bị Bluetooth Raspberry Pi

- I Kết nối Raspberry Pi với điện thoại thông minh dựa trên Android thông qua Bluetooth cho Truyền thông Bluetooth hai chiều
  - ▶ Thiết bị Bluetooth trên Raspberry Pi 4 được kết nối bên trong với CPU để giao tiếp với UART.
  - ▶ Nó có thể được sử dụng theo cách gần giống như sử dụng thiết bị nối tiếp, nhưng điểm khác biệt là nó yêu cầu quá trình ghép nối với thiết bị được kết nối không dây.
  - ▶ Một điểm khác biệt nữa là tên thiết bị nối tiếp không thể được sử dụng là “/dev/serial1”, mà là “/dev/rfcomm0”.
  - ▶ Another difference is that the serial device name cannot be used as “/dev/serial1”, but as “/dev/rfcomm0”.

Kênh UART	Cổng nối tiếp	Tên thiết bị	Sử dụng
UART1	Serial1	/dev/ttyAMA0 Or /dev/serial1	Bluetooth

## Ghép nối thiết bị Bluetooth

### | Quá trình ghép nối

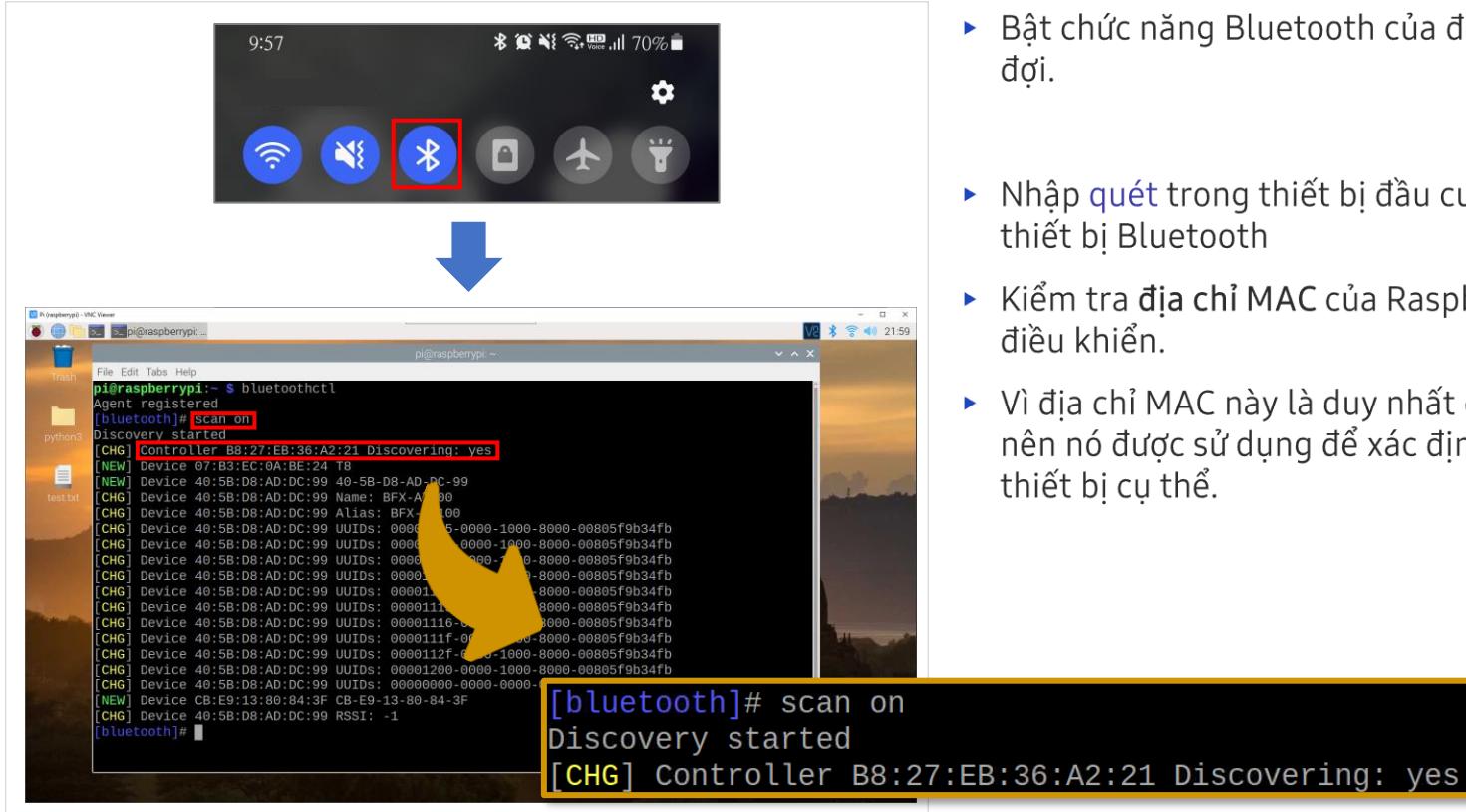


- ▶ Để Raspberry Pi và điện thoại thông minh giao tiếp không dây, trước tiên cần phải ghép nối.
- ▶ Chọn thiết bị bạn muốn kết nối vì có thể có nhiều thiết bị xung quanh thiết bị không dây. Nếu bạn cần số PIN, hãy nhập nó và kết nối.
- ▶ Quá trình ghép nối chỉ được yêu cầu một lần. Sau khi bạn ghép nối thiết bị Bluetooth lần đầu tiên, thiết bị của bạn có thể tự động ghép nối.
- ▶ Đầu tiên hãy chạy chương trình điều khiển Bluetooth để kiểm tra chi tiết quá trình ghép nối và kết nối trên Raspberry Pi.
- ▶ Nhập lệnh [bluetoothctl](#) trong thiết bị đầu cuối Raspberry Pi và nhấn Enter.

#### 4.3. Truyền thông Bluetooth

Bài 04

## Quá trình ghép nối



- ▶ Bật chức năng Bluetooth của điện thoại thông minh và đợi.
  - ▶ Nhập **quét** trong thiết bị đầu cuối để bắt đầu tìm kiếm thiết bị Bluetooth
  - ▶ Kiểm tra **địa chỉ MAC** của Raspberry Pi bên cạnh Bộ điều khiển.
  - ▶ Vì địa chỉ MAC này là duy nhất cho một thiết bị cụ thể nên nó được sử dụng để xác định, kết nối và quản lý thiết bị cụ thể.

### Quá trình ghép đôi

```

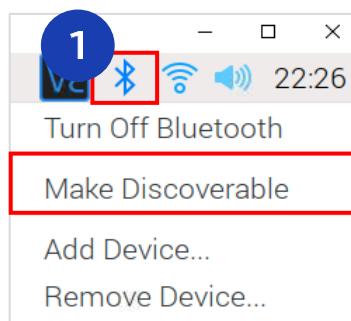
pi@raspberrypi: ~
[CHG] Device 40:5B:D8:AD:DC:99 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 40:5B:D8:AD:DC:99 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 40:5B:D8:AD:DC:99 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 40:5B:D8:AD:DC:99 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 40:5B:D8:AD:DC:99 UUIDs: 00000000-0000-0000-0000-000000000000
[NEW] Device CB:E9:13:80:84:3F CB-E9-13-80-84-3F
[CHG] Device 40:5B:D8:AD:DC:99 RSSI: -1
[CHG] Device CB:E9:13:80:84:3F Name: Mi Smart Band 4
[CHG] Device CB:E9:13:80:84:3F Alias: Mi Smart Band 4
[CHG] Device CB:E9:13:80:84:3F HUUD: 0000fca0_0000_0000_0000_feb34fb
[CHG] Device 07:B3:EC:0A:BE:24
[CHG] Device 40:5B:D8:AD:DC:99
[CHG] Device 07:B3:EC:0A:BE:24
[CHG] Device CB:E9:13:80:84:3F
[blueuetooth]# scan off
Discovery stopped
[CHG] Controller B8:27:F5
[CHG] Device CB:E9:13:80:84:3F
[CHG] Device 40:5B:D8:AD:DC:99
[CHG] Device 07:B3:EC:0A:BE:24
[DEL] Device 07:B3:EC:0A:BE:24
[DEL] Device 40:5B:D8:AD:DC:99
[DEL] Device CB:E9:13:80:84:3F
[blueuetooth]# quit
pi@raspberrypi: ~ $ 

```

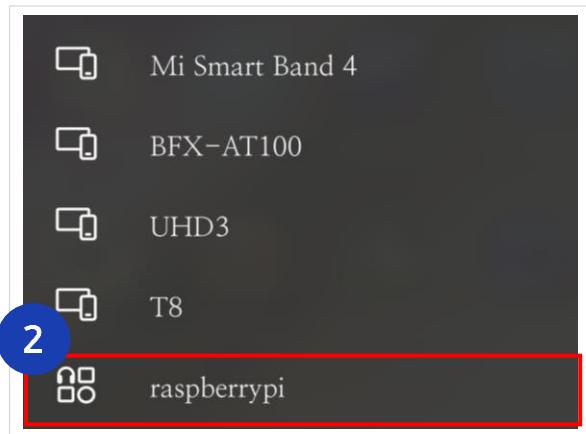
- Nếu bạn có thể thấy nhiều thiết bị trong lệnh quét, thì không có vấn đề gì với thiết bị Bluetooth của Raspberry Pi.
- Ngừng tìm kiếm thiết bị Bluetooth bằng [lệnh quét tắt](#) và kết thúc lệnh bluetoothctl bằng [lệnh thoát](#).

## Kết nối Raspberry Pi và Bluetooth

### I Kết nối Raspberry Pi với điện thoại thông minh

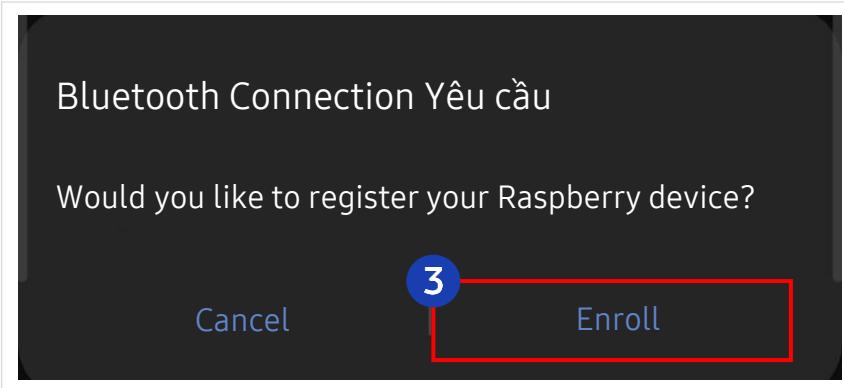


- Nhấp vào biểu tượng Bluetooth ở trên cùng bên phải của màn hình Raspberry Pi và nhấp vào **Make Discoverable** để làm cho Raspberry Pi có thể phát hiện được với các thiết bị khác.



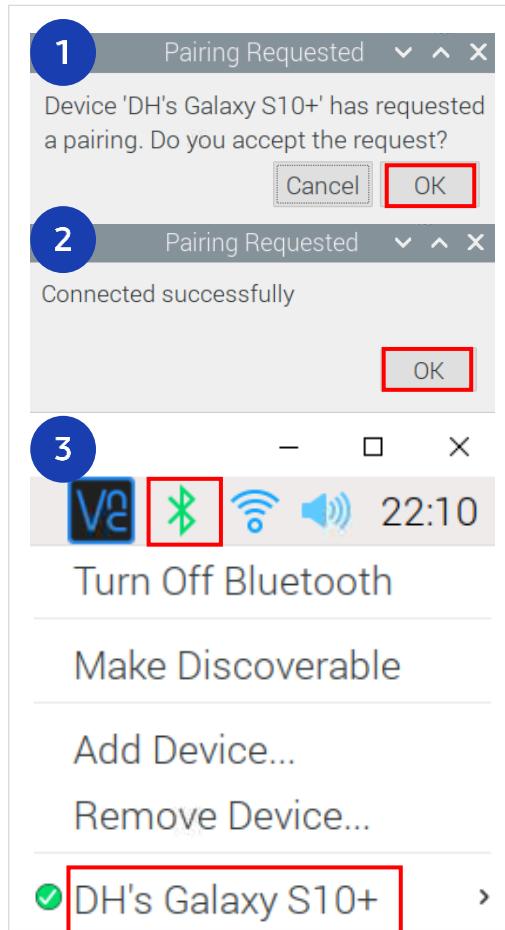
- Cuộn qua danh sách Bluetooth trên điện thoại thông minh của bạn để tìm raspberry pi và nhấp vào nó để thử kết nối

### I Kết nối Raspberry Pi với điện thoại thông minh



- Nhấp vào Đăng ký trong cửa sổ bật lên xác nhận có đăng ký thiết bị raspberry pi hay không.

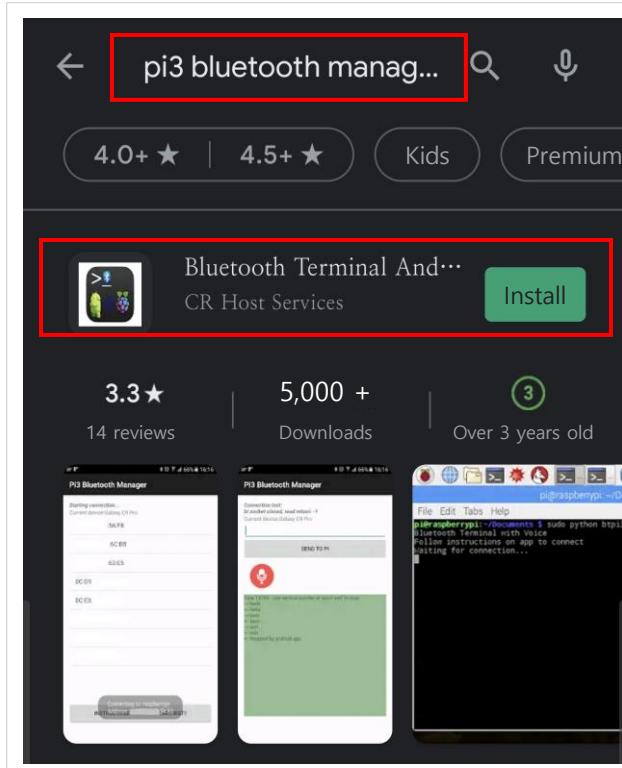
### Kết nối Raspberry Pi với điện thoại thông minh



- ▶ Khi cửa sổ bật lên **Yêu cầu ghép nối** xuất hiện trong thiết bị đầu cuối Raspberry Pi, hãy nhấp vào OK để chấp nhận Yêu cầu.
- ▶ Nếu xuất hiện thông báo **Connected Successfully** thì kết nối thành công.
- ▶ Nhấp vào biểu tượng ở phía trên bên phải để xác nhận, bạn sẽ thấy tên của thiết bị được kết nối.

## Cài đặt ứng dụng Bluetooth

### I Cài đặt ứng dụng Bluetooth cho thiết bị Android

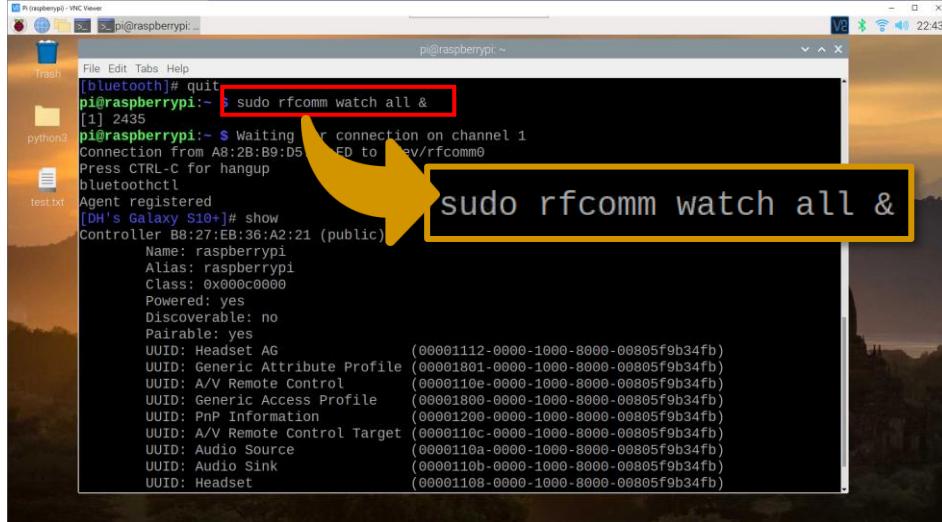


- ▶ Mặc dù điện thoại thông minh Android và Raspberry Pi được kết nối thông qua quá trình ghép nối, một ứng dụng bổ sung phải được cài đặt trên thiết bị Android để gửi và nhận dữ liệu qua Bluetooth.
- ▶ Tìm kiếm và tải xuống [trình quản lý bluetooth pi3](#) từ Cửa hàng Google Play

## 4.3. Truyền thông Bluetooth

## Bài 04

### Cài đặt ứng dụng Bluetooth cho thiết bị Android



```
pi@raspberrypi: ~
[bluetooth]# quit
pi@raspberrypi: ~ [1] 2435
pi@raspberrypi: $ Waiting for connection on channel 1
Connection from A8:2B:B9:D5:ED to /dev/rfcomm0
Press CTRL-C to hangup
bluetoothctl
Agent registered
[DH's Galaxy S10+]# show
Controller B8:27:EB:36:A2:21 (public)
  Name: raspberrypi
  Alias: raspberrypi
  Class: 0x00c0000
  Powered: yes
  Discoverable: no
  Pairable: yes
  UUID: Headset AG          (00001112-0000-1000-8000-00005f9b34fb)
  UUID: Generic Attribute Profile (00001801-0000-1000-8000-00005f9b34fb)
  UUID: A/V Remote Control    (0000110e-0000-1000-8000-00005f9b34fb)
  UUID: Generic Access Profile (00001800-0000-1000-8000-00005f9b34fb)
  UUID: PnP Information       (00001200-0000-1000-8000-00005f9b34fb)
  UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00005f9b34fb)
  UUID: Audio Source          (0000110a-0000-1000-8000-00005f9b34fb)
  UUID: Audio Sink             (0000110b-0000-1000-8000-00005f9b34fb)
  UUID: Headset                (00001108-0000-1000-8000-00005f9b34fb)
```

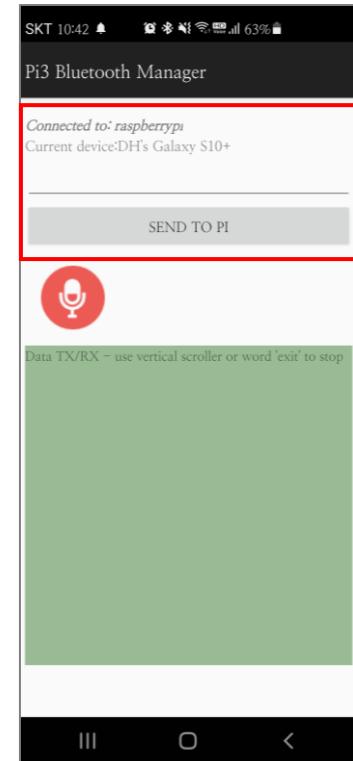
- ▶ Trước khi chạy ứng dụng đã tải xuống, trước tiên bạn phải thực thi `rfcomm watch all` & trên Raspberry Pi.
- ▶ `rfcomm watch all` là một lệnh kết nối với thiết bị đầu cuối này khi được kết nối với Pi từ một thiết bị khác qua Bluetooth. & là để thực thi lệnh dưới dạng nền.

### | Cài đặt ứng dụng Bluetooth cho thiết bị Android

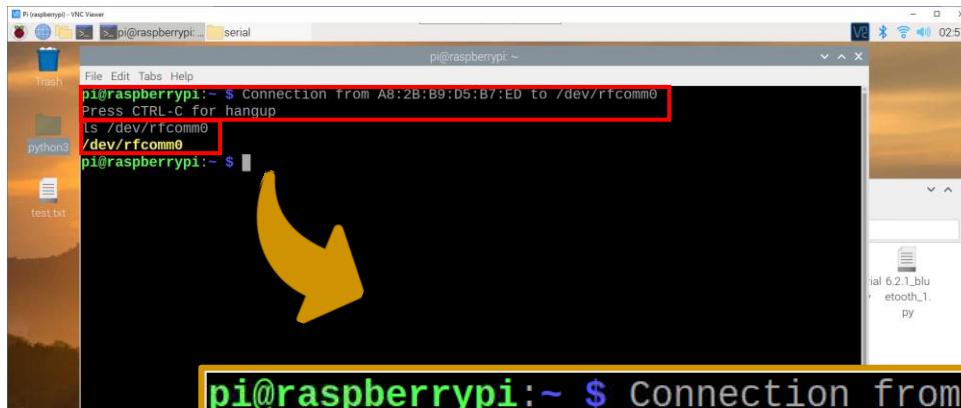
- ▶ Quay lại ứng dụng trên điện thoại thông minh của bạn, tìm địa chỉ MAC của Raspberry Pi và nhấp vào nó.



- ▶ Nếu kết nối thành công, bạn sẽ thấy màn hình như hình bên dưới.
- ▶ Văn bản đã nhập được gửi đến Raspberry Pi.



### Cài đặt ứng dụng Bluetooth cho thiết bị Android



A screenshot of an SSH terminal window titled "Pi (raspberrypi) - VNC Viewer". The terminal shows a command-line session with the user "pi" on "raspberrypi". The terminal window has a red border around the command output. A large yellow arrow points from the terminal window down to a larger, highlighted area below it.

```
pi@raspberrypi:~ $ Connection from A8:2B:B9:D5:B7:ED to /dev/rfcomm0
Press CTRL-C for hangup
ls /dev/rfcomm0
/dev/rfcomm0
pi@raspberrypi:~ $
```

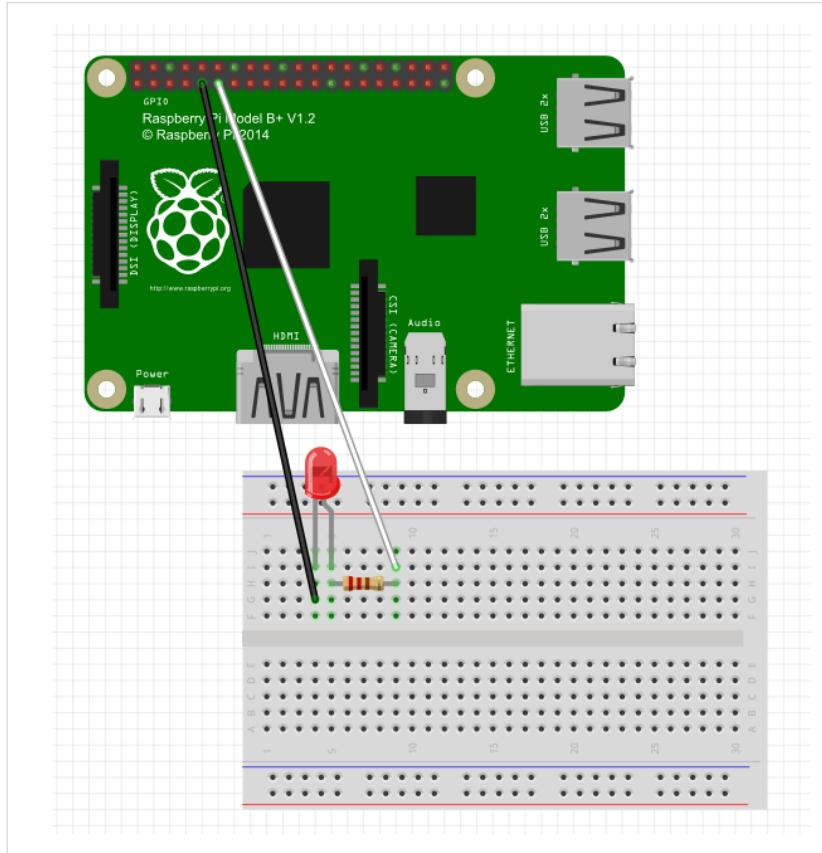
The highlighted area contains the following text:

```
pi@raspberrypi:~ $ Connection from A8:2B:B9:D5:B7:ED to /dev/rfcomm0
Press CTRL-C for hangup
ls /dev/rfcomm0
/dev/rfcomm0
```

- ▶ Bây giờ bạn đã sẵn sàng để gửi đến Raspberry Pi. Bạn cần có khả năng xử lý chuỗi nhận được trên Raspberry Pi
- ▶ Nếu kết nối thành công, bạn sẽ thấy thông báo bắt đầu bằng **Kết nối từ** như trong hình. Kết nối Từ cho biết rằng nó được kết nối từ điện thoại thông minh.
  
- ▶ Nhập [ls /dev/rfcomm0](#). Nếu thư mục xuất hiện như trong hình, thiết bị Bluetooth đã được kết nối thành công.
- ▶ Thư mục này sẽ biến mất khi thiết bị Bluetooth bị ngắt kết nối.

## Sơ đồ mạch

### I Tạo sơ đồ mạch



- Để nhấp nháy đèn LED bằng Raspberry Pi, trước tiên chúng ta cần kết nối đèn LED với Raspberry Pi.
- Chúng tôi sẽ gửi tín hiệu đến đèn LED thông qua GPIO17. Kết nối đèn LED theo thứ tự **GPIO17 → Điện trở → LED+ → LED- → GND**.

## Tạo bluetooth\_led.py

### I Tạo tệp script Python trên Raspberry Pi và lưu nó vào vị trí mong muốn

- ▶ Mã này sẽ bật đèn LED khi nhận được chuỗi “ON\n” được gửi từ điện thoại thông minh trong Raspberry Pi và tắt đèn LED khi nhận được chuỗi “OFF\n”.

```
1 import RPi.GPIO as GPIO
2 import serial
3
4 LED = 17
5
6 GPIO.setmode(GPIO.BCM)
7 GPIO.setup(LED, GPIO.OUT)
8
9 ser = serial.Serial('/dev/rfcomm0', 115200)
10 ser.close()
11 ser.open()
12
13 str = b'Bluetooth LED Control\r\n'
14 n = ser.write(str)
```

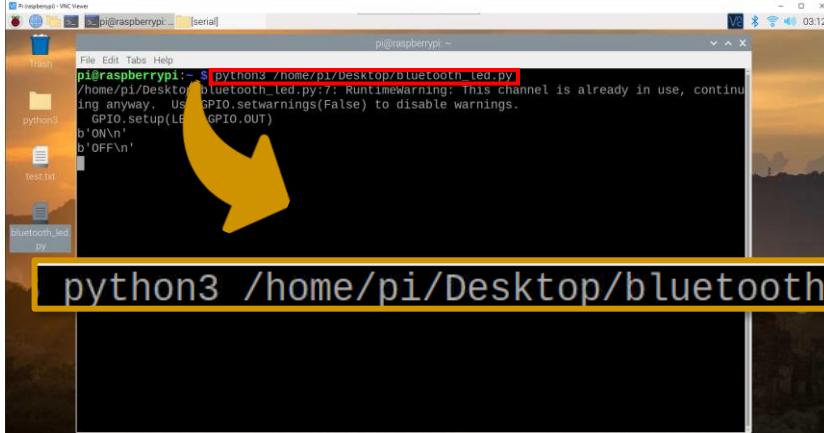
- dòng 1: Sử dụng Gói Rpi.GPIO
- dòng 2: Sử dụng Gói nối tiếp py
- dòng 4: Số GPIO với đèn LED được chọn
- dòng 6: Sử dụng chế độ BCM
- dòng 7: Đặt đèn LED làm đầu ra
- dòng 9: Đặt 115200bps làm tốc độ giao tiếp
- dòng 10, 11: Đóng cổng, Mở cổng nối tiếp
- dòng 13: Byte dữ liệu để gửi đến PC
- dòng 14: Truyền dữ liệu Byte

### Tạo tệp script Python trên Raspberry Pi và lưu nó vào vị trí mong muốn

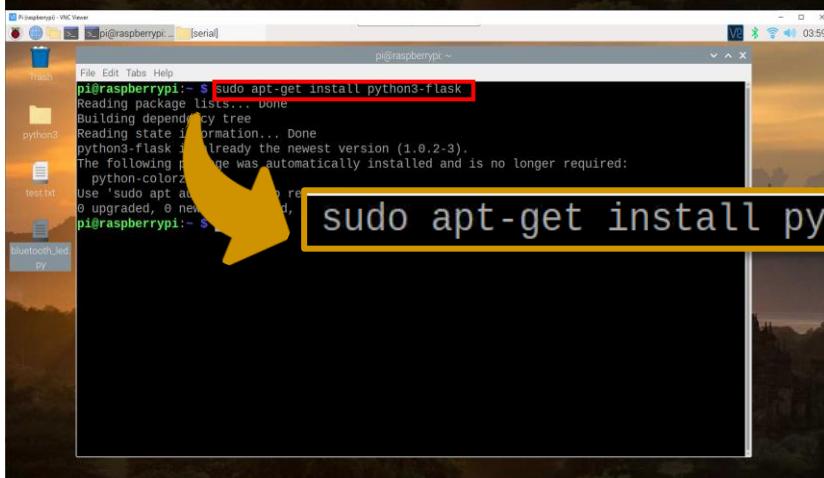
```
15
16     try:
17         while True:
18             if ser.readable():
19                 response = ser.readline()
20                 if response == b'ON\n':
21                     GPIO.output(LED, True)
22                 elif response == b'OFF\n':
23                     GPIO.output(LED, False)
24
25                 print(response)
26
27     except KeyboardInterrupt:
28         pass
29
finally:
    ser.close()
```

- dòng 18: Nếu có dữ liệu nhận được từ điện thoại thông minh
- dòng 19: Đọc tối đa \r\n ký tự
- dòng 20, 21: LED sáng khi nhận được “ON\r\n”
- dòng 22, 23: LED tắt khi nhận được “OFF\r\n”

### Lưu tập lệnh ví dụ vào vị trí mong muốn và chạy nó



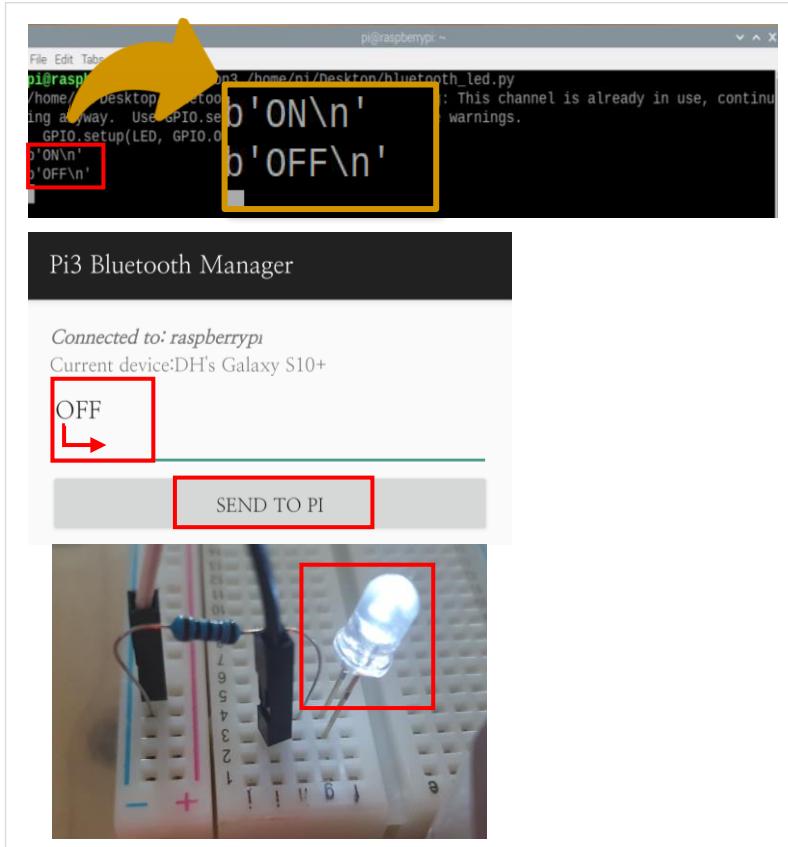
```
pi@raspberrypi:~ $ python3 /home/pi/Desktop/bluetooth_led.py
/home/pi/Desktop/bluetooth_led.py:7: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(LED, GPIO.OUT)
b'ON\n'
b'OFF\n'
```



```
pi@raspberrypi:~ $ sudo apt-get install python3-flask
Reading package lists... Done
Building dependency tree...
Reading state information... Done
python3-flask is already the newest version (1.0.2-3).
The following package was automatically installed and is no longer required:
  python-colorama
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $
```

- ▶ Lưu tập lệnh dưới dạng bluetooth\_led.py thành /home/pi/Desktop/bluetooth\_led.py
- ▶ Nếu bạn chạy tập lệnh bằng python3 /home/pi/Desktop/bluetooth\_led.py, tập sẽ chuyển sang trạng thái chờ đầu vào.

### | Lưu tập lệnh ví dụ vào vị trí mong muốn và chạy nó



- ▶ Nhập **BẬT\n** hoặc **TẮT\n** vào trường văn bản và nhấn **GỬI ĐẾN PI** trong điện thoại thông minh. (Đối với \n, nhấn Enter trên bàn phím. Nhập BẬT (hoặc TẮT) rồi nhấn Enter.)
- ▶ Bạn sẽ thấy đèn LED bật và tắt và các giá trị đầu ra thay đổi trên màn hình đầu cuối bằng cách nhấn **GỬI ĐẾN PI**.

Bài 4.

# Hướng dẫn giao tiếp Socket with Raspberry Pi

- | 4.1. Truyền thông nối tiếp
- | 4.2. Truyền thông SPI
- | 4.3. Truyền thông Bluetooth
- | **4.4. Máy chủ web Flask**

## Máy chủ web

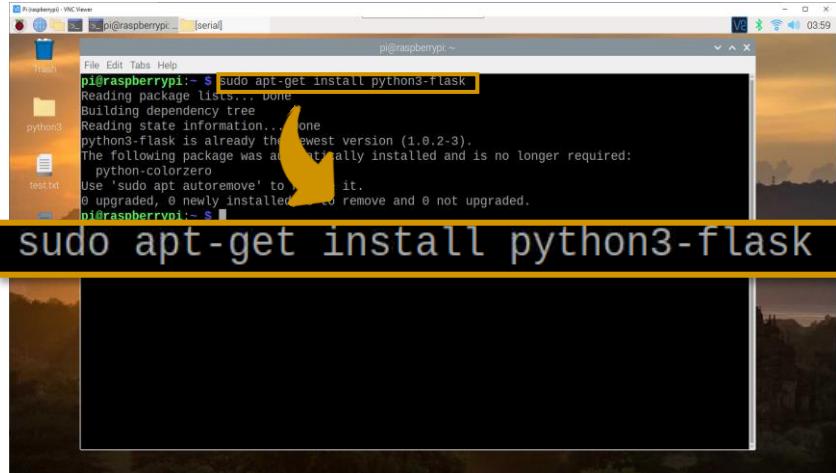
### I Máy chủ Web là gì?



- ▶ Máy chủ Web là một chương trình chạy trên máy chủ lưu trữ trang web.
- ▶ Máy chủ Web đáp ứng các giao thức HTTP mà trình duyệt Web yêu cầu.
- ▶ Mặc dù có nhiều phần mềm máy chủ web, chúng tôi sẽ sử dụng Flask vì nó nhẹ và dễ cài đặt. Flask phù hợp với Raspberry Pi do dung lượng bộ nhớ hạn chế của Pi.
- ▶ Flask giống một khung web hơn là hoạt động như một máy chủ web hoàn chỉnh và triển khai các chức năng của máy chủ web bằng Python.
- ▶ ※ Xem thêm thông tin về Flask:  
<https://flask.palletsprojects.com/en/2.0.x/>

## Máy chủ web Flask

### I Cài đặt Flask web của máy chủ



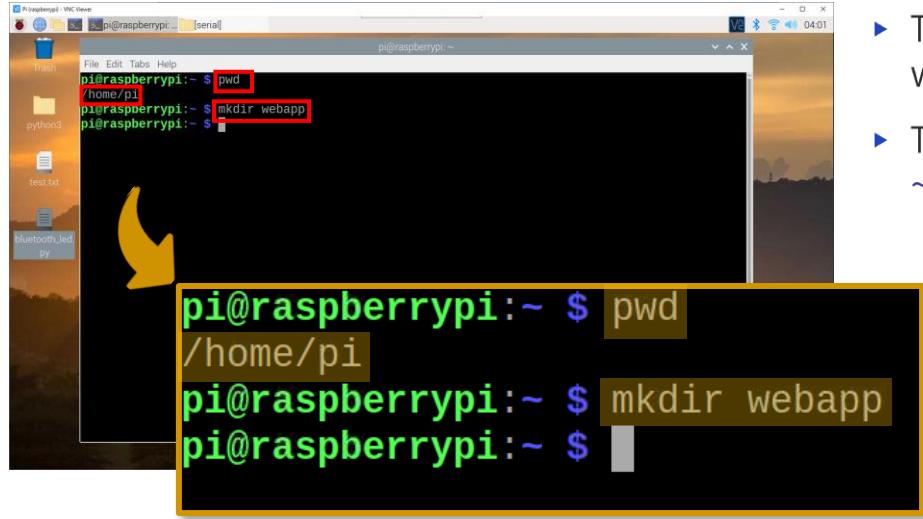
```
pi@raspberrypi: ~$ sudo apt-get install python3-flask
Reading package lists... done
Building dependency tree
Reading state information... done
python3-flask is already the newest version (1.0.2-3).
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi: ~$
```

**sudo apt-get install python3-flask**

- ▶ Xem thêm thông tin về Flask:  
<https://flask.palletsprojects.com/en/2.0.x/>
- ▶ Để cài đặt Flask, hãy thực hiện lệnh **sudo apt-get install python3-flask** trong terminal và cài đặt nó.

> **sudo apt-get install python3-flask**

### Thực hành Flask



```
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $ mkdir webapp  
pi@raspberrypi:~ $
```

```
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $ mkdir webapp  
pi@raspberrypi:~ $
```

- ▶ Thu thập các tệp sẽ được sử dụng để triển khai Flask web của Máy chủ vào một thư mục.
- ▶ Tạo một thư mục có tên **webapp** tại vị trí **~(/home/pi/rasp\_ex)**.

## ~/webapp/index.py

### I Thực hành Flask

- Thực hành một chương trình thử nghiệm đơn giản xuất ra cụm từ Hello Flask cho máy khách khi kết nối với cổng 80 của Raspberry Pi thông qua máy khách (trình duyệt Internet trên PC).

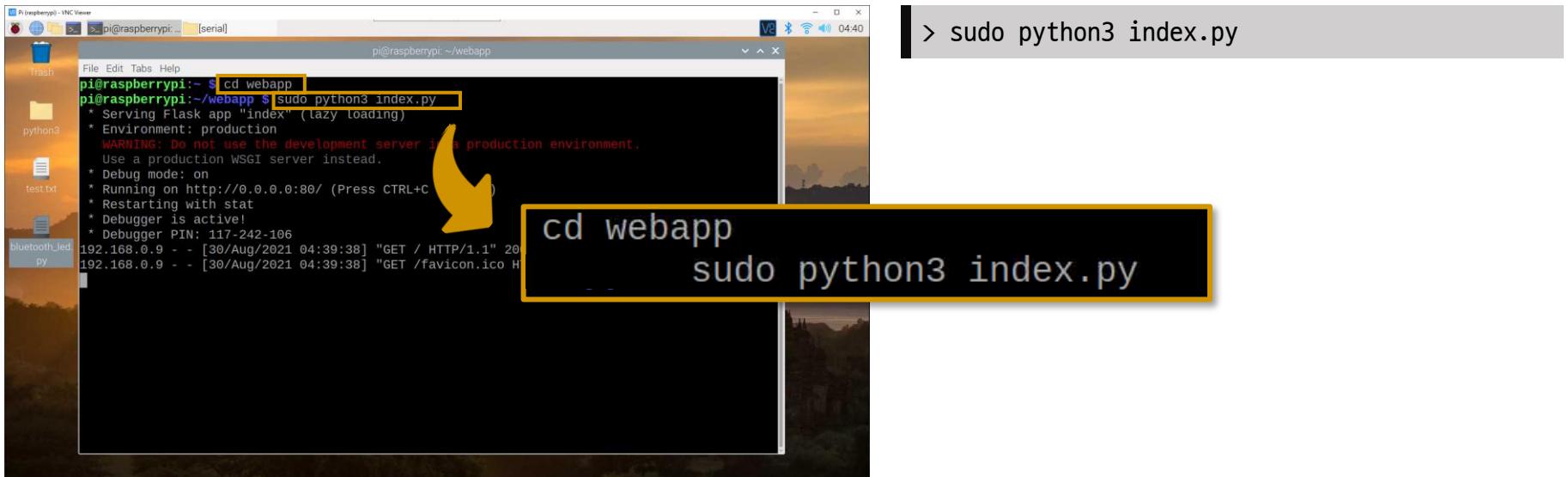
```
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route('/')  
6  def hello():  
7      return 'Hello Flask'  
8  
9  if __name__ == '__main__':  
10     app.run(debug=True, port=80, host='0.0.0.0')
```

- dòng 1: Sử dụng gói Flask
- dòng 3: Tạo phiên bản Flask
- dòng 5: Khi một Yêu cầu xuất phát từ URL '/'
- dòng 6: Thực hiện chức năng xin chào
- dòng 9: Dịch vụ với 80 cổng

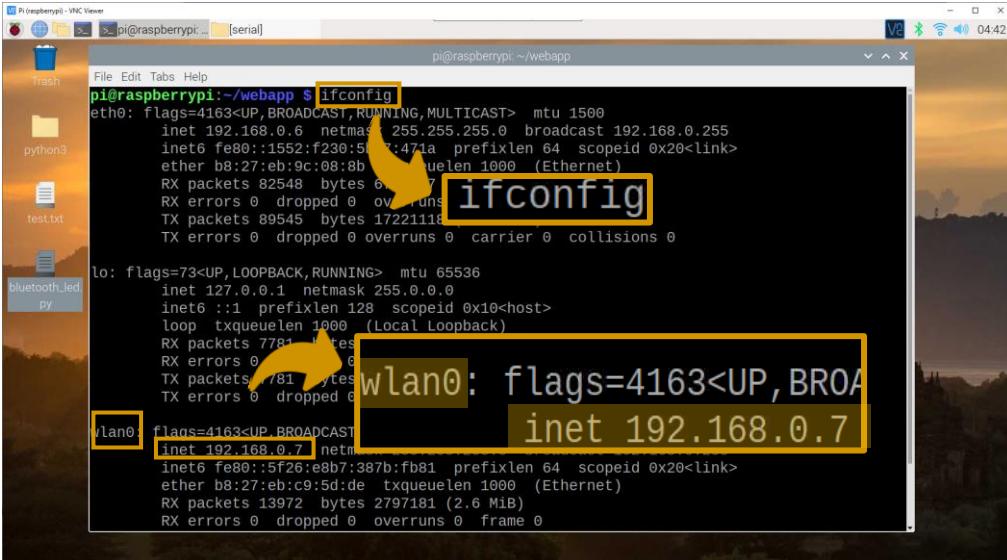
### I Thực hành Flask

```
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route('/')  
6  def hello():  
7      return 'Hello Flask'  
8  
9  if __name__ == '__main__':  
10     app.run(debug=True, port=80, host='0.0.0.0')
```

- ▶ Chỉ định URL '/' có nghĩa là chỉ định vị trí mặc định để truy cập khi không có gì khác ngoài tên miền hoặc địa chỉ IP được chỉ định trong trình duyệt web.
- ▶ Bạn cần có quyền root để triển khai dịch vụ Flask.
- ▶ Di chuyển đến thư mục ứng dụng web và nhập **sudo python3 index.py** để thực thi tệp Python được tạo với quyền root.



## Thực hành Flask



```
pi@raspberrypi:~/webapp $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.6 netmask 255.255.255.0 broadcast 192.168.0.255
                inet6 fe80::1552:f230:5177:471a prefixlen 64 scopeid 0x20<link>
                    ether b8:27:eb:9c:08:b8 txqueuelen 1000 (Ethernet)
                    RX packets 82548 bytes 6977476
                    RX errors 0 dropped 0 overruns 0
                    TX packets 89545 bytes 17221118
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 7781 bytes 6066
            RX errors 0 dropped 0
            TX packets 781 bytes 6066
            TX errors 0 dropped 0

wlan0: flags=4163<UP,BROADCAST,NOARP> mtu 1500
        inet 192.168.0.7 netmask 255.255.255.0 broadcast 192.168.0.255
                inet6 fe80::5f26:e8b7:387b:fb81 prefixlen 64 scopeid 0x20<link>
                    ether b8:27:eb:c9:5d:de txqueuelen 1000 (Ethernet)
                    RX packets 13972 bytes 2797181 (2.6 MiB)
                    RX errors 0 dropped 0 overruns 0 frame 0
```



- ▶ Địa chỉ IP phải được nhập khi kết nối từ trình duyệt web của PC. Khi nhập lệnh `ifconfig` thì nó hiện ra địa chỉ IP cạnh `inet` bên dưới `wlan0`. Đó là địa chỉ IP của Raspberry Pi.
- ▶ Nếu bạn mở trình duyệt web đến địa chỉ IP, bạn sẽ thấy Hello Flask.

## Thêm Trang Web

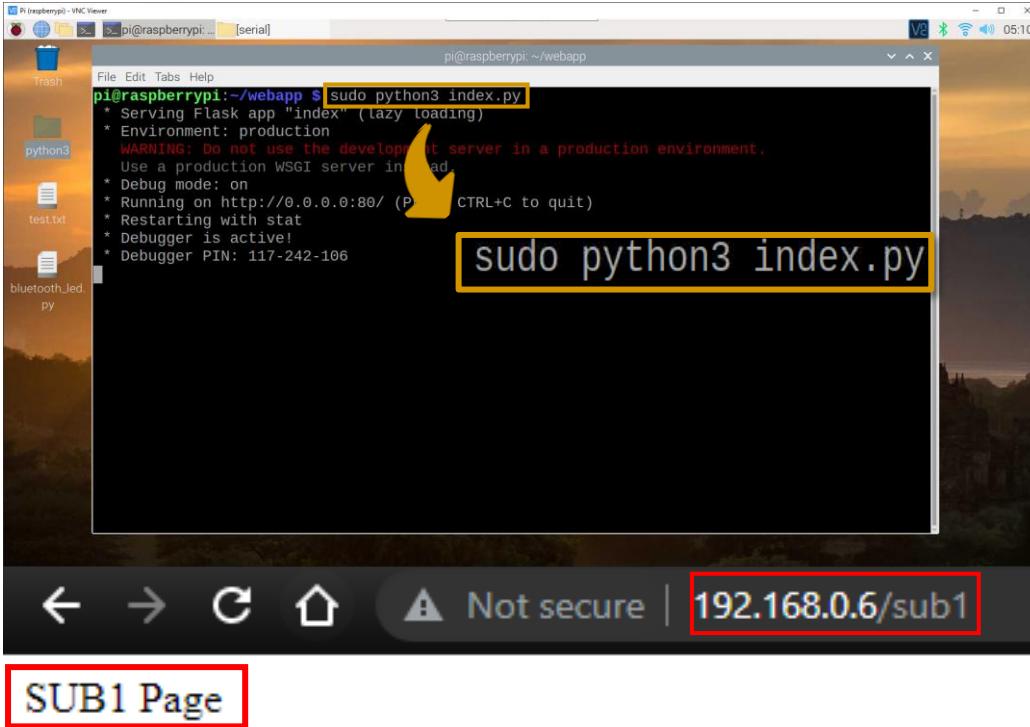
### | @app.route()

- ▶ Có thể thêm nhiều trang dịch vụ bằng hàm @app.route().
- ▶ Hãy thêm **trang phụ1**.
- ▶ Sửa đổi ~/rasp\_ex/webapp/index.py bạn đã viết để thêm nó.

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello():
7      return 'Hello Flask'
8
9  @app.route('/sub1')
10 def sub1():
11     return 'SUB1 Page'
12
13 if __name__ == '__main__':
14     app.run(debug=True, port=80, host='0.0.0.0')
```

- dòng 9 : Khi có Request /sub1
- dòng 10 : hàm sub1() thực thi.

## @app.route()

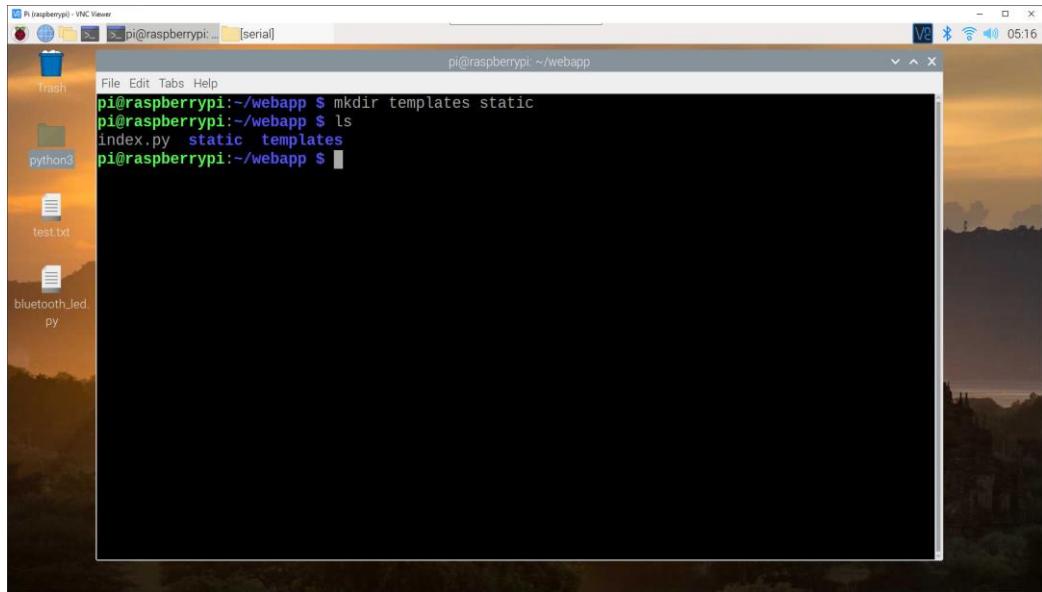


- ▶ Thực thi index.py đã sửa đổi bằng cách nhập lệnh `sudo python3 index.py`.
- ▶ Sau khi thực hiện, hãy mở trình duyệt web và nhập địa chỉ IP bạn đã sử dụng và thêm `/sub1` sau đó và chạy nó. Bạn có thể xem cụm từ SUB1 Page.

> `sudo python3 index.py`

## Tạo trang HTML

### I Hiển thị nhiều hình ảnh, siêu văn bản, bảng, v.v. trên một trang web

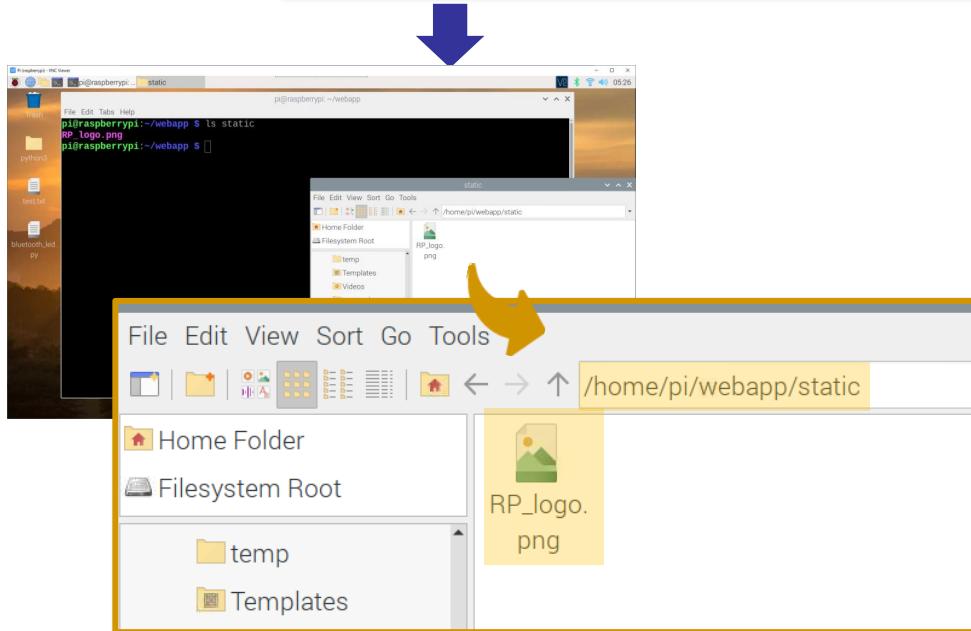
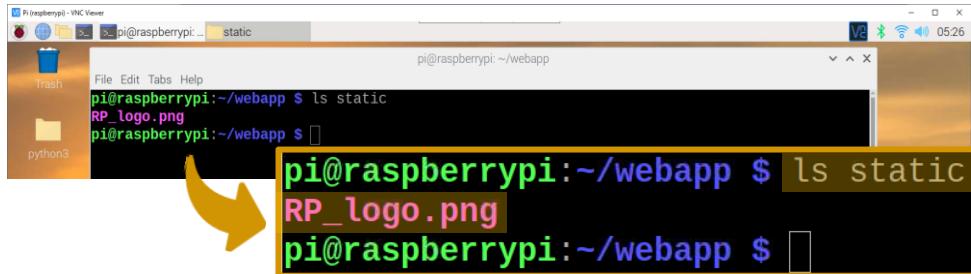


- ▶ Để hiển thị nhiều thông tin khác nhau trên trang web không chỉ là thông báo đơn giản, bạn có thể tạo một trang HTML.
- ▶ Để xuất trang HTML, hãy tạo một tệp HTML và thêm nó vào thư mục sẽ được Flask phục vụ.
- ▶ Nếu bạn muốn hiển thị hình ảnh hoặc video trong HTML, tệp hình ảnh và tệp video cần được chuẩn bị.
- ▶ Viết mã Python hiển thị văn bản và hình ảnh bằng các thẻ HTML đơn giản bằng hàm `render_template()` trong Flask.
- ▶ Để hiển thị các tệp HTML, các mẫu và thư mục tĩnh cũng cần có trong thư mục nơi mã Python thực thi. Vì vậy, trước tiên hãy tạo chúng.

## 4.4. Máy chủ web Flask

## Bài 04

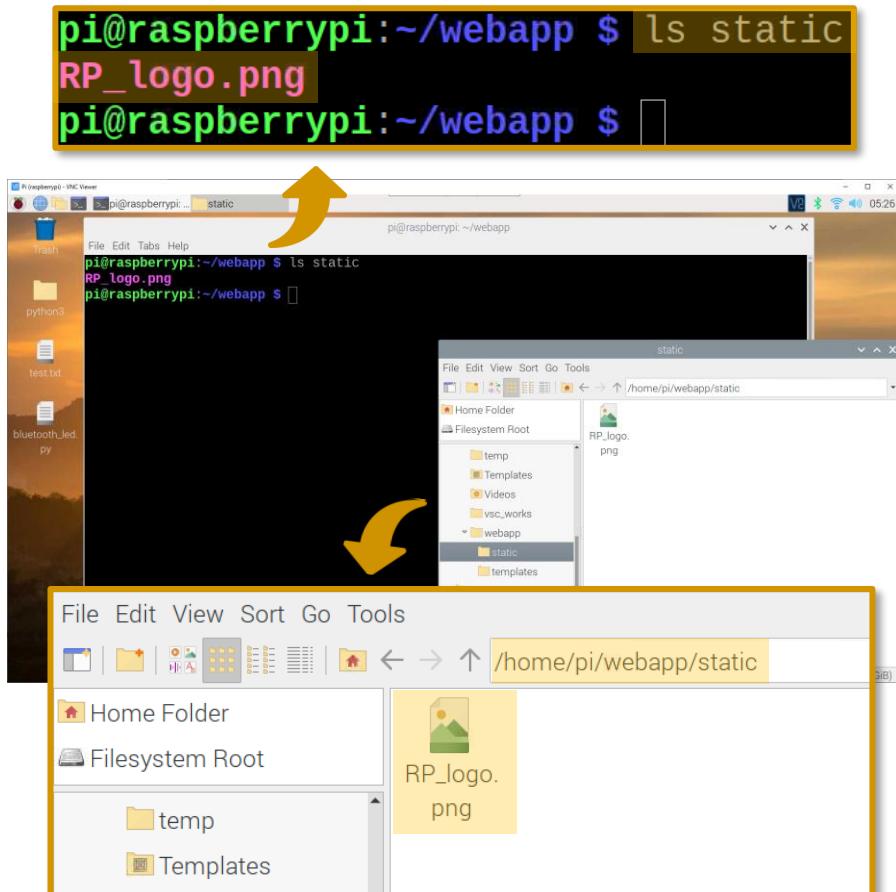
| Hiển thị nhiều hình ảnh, siêu văn bản, bảng, v.v. trên một trang web



- ▶ [https://media.vlpt.us/images/mythos/post/5584cec3-97f8-45b4-a8c8-003c574f732c/Raspberry\\_Pi\\_OS\\_Logo.png](https://media.vlpt.us/images/mythos/post/5584cec3-97f8-45b4-a8c8-003c574f732c/Raspberry_Pi_OS_Logo.png)

- ▶ Ví dụ, tải xuống tệp img để sử dụng.
- ▶ Tải xuống tệp png hình ảnh, hiển thị và đặt nó vào thư mục tĩnh.

| Hiển thị nhiều hình ảnh, siêu văn bản, bảng, v.v. trên một trang web



- ▶ Tải xuống tệp png hình ảnh bạn muốn hiển thị và lưu nó trong thư mục tĩnh.
- ▶ Trong ví dụ này, tệp hình ảnh được sử dụng từ liên kết bên dưới.  
[https://media.vlpt.us/images/mythos/post/5584cec3-97f8-45b4-a8c8-003c574f732c/Raspberry\\_Pi\\_OS\\_Logo.png](https://media.vlpt.us/images/mythos/post/5584cec3-97f8-45b4-a8c8-003c574f732c/Raspberry_Pi_OS_Logo.png)

| ~/webapp/templates/test\_html.html

```
1 <html>
2   <head>
3     <title>Flask HTML test page</title>
4   </head>
5
6
7   <body>
8     <center>
9       <br>
10      <strong>Raspberry Pi 4 Flask HTML Test Page</strong>
11      <br>
12      
13    </center>
14  </body>
15
16 </html>
```

- ▶ Đầu tiên, hãy tạo một trang HTML đơn giản. Chúng ta sẽ tìm hiểu ngắn gọn cách tạo một trang web thay vì bao gồm tất cả các thẻ HTML.
- ▶ Tất cả các thẻ HTML phải được đặt trong <> các dấu ngoặc này.
- ▶ Thẻ mở <tag> bắt đầu một phần của nội dung trang và thẻ đóng </tag> kết thúc phần đó.  
`<tag> nội dung </tag>`
- ▶ <title> tag: tiêu đề của trang web
- ▶ <body>: nội dung của trang web
- ▶ <strong>: hiệu ứng nổi bật
- ▶ <img src>: Viết tắt của nguồn hình ảnh. Nó được sử dụng để tải một tập tin hình ảnh. Nhập đường dẫn hình ảnh như thế này .
- ▶ Sau trang HTML, hãy lưu nó dưới dạng test\_html.html trong các mẫu.

| Write ~/webapp/html\_test.py

```
1  from flask import Flask, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      return render_template('html_test.html')
8
9  if __name__ == '__main__':
10     app.run(debug=True, port=80, host='0.0.0.0')
```

```
pi@raspberrypi:~/webapp $ tree
.
├── html_test.py
├── index.py
└── static
    └── RP_logo.png
└── templates
    └── html_test.html

2 directories, 4 files
pi@raspberrypi:~/webapp $
```

- ▶ Kết xuất (tạo và xem) tài liệu HTML bằng Python..
  - dòng 7: Kết xuất tệp html\_test.html

- ▶ Khi bạn tạo và lưu html\_test.py, cấu trúc được tạo như bạn thấy ở bên trái.

### Xem trang đã tạo trên web

```
pi@raspberrypi:~/webapp $ sudo python3 html_test.py
 * Serving Flask app "html_test" (lazy loading)
 * Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting
 * Debugger is active
 * Debugger PIN: 12345
```

A yellow arrow points from the terminal window to the command `sudo python3 html_test.py`.

- ▶ Nhập `sudo python3 html_test.py` để thực thi tập lệnh Python đã tạo.
- ▶ > `sudo python3 html_test.py`
- ▶ Nhập địa chỉ IP vào trình duyệt web.
- ▶ Hình ảnh dưới đây là đầu ra.



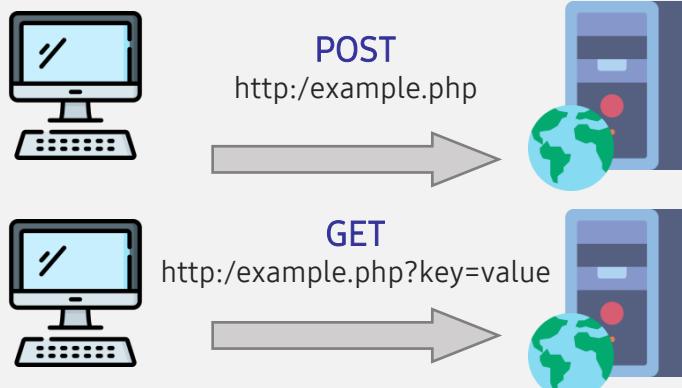
## GET, POST

### I Các trang HTML một mình không thể phục vụ nội dung động

- Trong Flask, đầu ra động có thể thực hiện được bằng cách nhận các tham số được truyền từ máy khách web.
- Có hai phương thức để chuyển động các tham số từ máy khách web sang máy chủ web (Flask): GET và POST.

#### GET

- Yêu cầu bằng cách bao gồm biến (Dữ liệu) trong URL
- Truyền bao gồm cả dữ liệu trong tiêu đề
- Dễ bị bảo mật vì dữ liệu bị lộ trong URL
- Bộ nhớ đệm có thể



#### POST

- Không để lộ các biến (Dữ liệu) trong URL
- Truyền bao gồm cả dữ liệu trong cơ thể
- Dữ liệu không bị lộ trong URL, vì vậy bảo mật cơ bản được cung cấp
- Bộ nhớ đệm

# GET

## | Tham số phương thức GET

- ▶ Trong phương thức GET, bạn có thể đặt tham số trong URL của trình duyệt web và chuyển nó đến máy chủ web hoặc Yêu cầu bằng cách sử dụng phương thức Gửi dưới dạng GET trong thẻ đầu vào HTML.
- ▶ Hãy cùng thực hành với ví dụ về truyền và nhận tham số.
- ▶ Để Yêu cầu một tham số trong phương thức GET, hãy bắt đầu bằng "?" trong URL của trang máy chủ web và viết nó ở dạng "tên=giá trị".
- ▶ Để gửi nhiều tham số, hãy tách chúng bằng dấu "&".

### | Phương thức GET cơ bản

- ▶ Đó là một phương thức để gửi yêu cầu bằng GET bằng cách nhập tham số vào URL ở dạng cơ bản nhất.

```
1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/method', methods=['GET'])
6  def method():
7      if request.method == 'GET':
8          id = request.args["id"]
9          password = request.args.get("password")
10         return "Transferred by GET method ({}, {})".format(id, password)
11
12  if __name__ == '__main__':
13      app.run(debug=True, port=80, host='0.0.0.0')
14
```

- dòng 5 : Khi một Yêu cầu xuất hiện trong phương thức GET
- dòng 7 : Nếu phương thức Request là GET
- dòng 8: “id” trong URL
- dòng 9: “mật khẩu” trong URL

- ▶ Đây là một tập lệnh python xuất các tham số đầu vào theo mẫu trên trang.

### | Tham số phương thức GET

- ▶ Phương thức nhập tham số vào URL và gửi chúng là ví dụ cơ bản nhất của phương thức GET.
- ▶ Tuy nhiên, phương pháp này có nhược điểm là mỗi lần gõ tham số vào thanh địa chỉ, chúng ta sẽ viết một tệp HTML và thực hiện yêu cầu trong cùng một phương thức GET.
- ▶ Đầu tiên, hãy viết `method_get.html` trước khi viết tập lệnh Python. File html này là file để thực hiện các chức năng và form tương tự như khi chúng ta đăng nhập từ một trang web.

| ~/rasp\_ex/webapp/templates/method\_get.html

```
1  <html>
2  <head>
3  <title>GET Method Request Test</title>
4  </head>
5
6  <body>
7  <h2>ID:{{ id }}, PASSWORD:{{ password }}</h2>
8  <form method="get" action="/method_get_act">
9      <label id="Label1">id</label>
10     <input name="id" type="text" />
11     <label id="Label1"><br />password </label>
12     <input name="password" type="text" /><br /><br />
13     <input name="Submit1" type="submit" value="submit" />
14 </form>
15 </body>
16 </html>
```

- dòng 7: Nhận giá trị truyền từ Flask Python Script và xuất ra theo định dạng
- dòng 8-14: Tạo khung để nhận id và mật khẩu nhập vào và gửi thông qua nút gửi
- dòng 8: Nếu phương thức trong thẻ biểu mẫu là get, Yêu cầu trong phương thức GET. Mã sau hành động = quyết định chức năng nào của Flask Server sẽ gọi khi nhấn nút gửi

- ▶ Khi trình duyệt ban đầu Yêu cầu URL `method_get.html`, không có gì được hiển thị trong `{{id}}` và `{{password}}` ở dòng 7 vì không có dữ liệu nào được truyền từ Flask.
- ▶ Lưu tệp này ở cùng dạng với `~/rasp_ex/webapp/templates/method_get.html`.

| ~/webapp/method\_get.py

- ▶ Viết tập lệnh python sẽ chuyển tham số tới method\_get.html mà bạn đã viết.

```

1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/method_get', methods=['GET'])
6  def method_get():
7      return render_template('method_get.html')
8
9  @app.route('/method_get_act', methods=['GET'])
10 def method_get_act():
11     if request.method == 'GET':
12         id = request.args["id"]
13         password = request.args.get("password")
14         return render_template('method_get.html', id=id, password=password)
15
16 if __name__ == '__main__':
17     app.run(debug=True, port=80, host='0.0.0.0')
```

- dòng 5-7: Phương thức kết xuất\_get.html
- dòng 9: Gọi bằng thẻ <form method="get" action="/method\_get\_act"> trong html
- dòng 12-13 : Đọc “id” và “password” Yêu cầu bởi method\_get.html
- dòng 14: Truyền tham số id và mật khẩu cho method\_get.html

| > sudo python3 method\_get.py

- ▶ Sau khi lưu tệp này vào ~/rasp\_ex/webapp/method\_get.py, hãy thực thi method\_get.py bằng lệnh python3.

```

pi@raspberrypi:~/rasp_ex/webapp $ ls
html_test.py index.py method_get.py static templates
pi@raspberrypi:~/rasp_ex/webapp $ sudo python3 method_get.py
* Serving Flask app "method_get" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in
  Production. Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-242-106
```



```

pi@raspberrypi:~/webapp $ ls
html_test.py index.py method_get.py static templates
pi@raspberrypi:~/webapp $ sudo python3 method_get.py
```

## I màn hình kết nối

1 192.168.0.5/method\_get

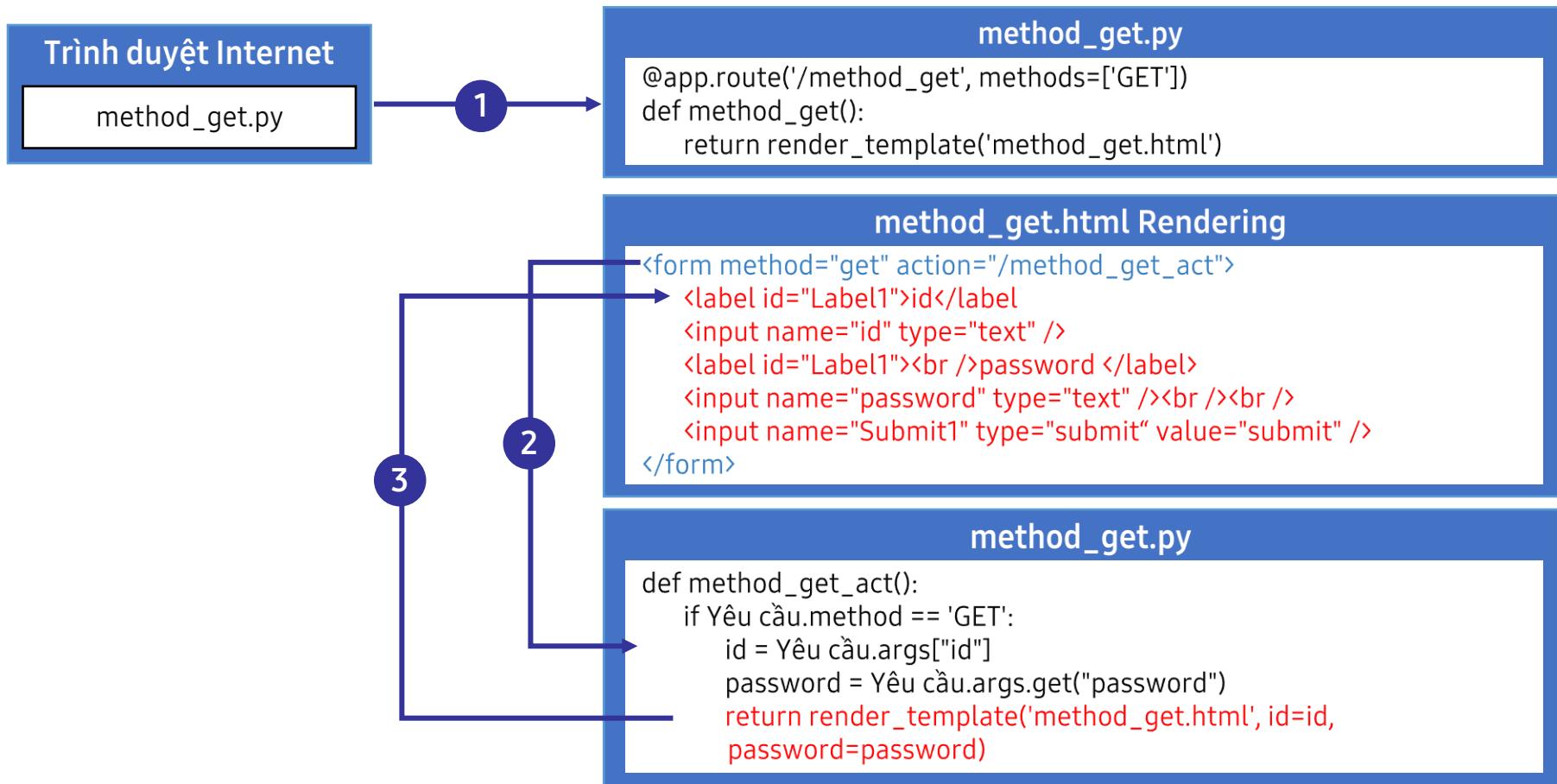
2 192.168.0.5/method\_get

3 192.168.0.5/method\_get\_act?id=Daniel&password=1234&Submit1=submit

ID:Daniel, PASSWORD:1234

- ▶ Khi kết nối qua trình duyệt web với địa chỉ IP [/method\\_get](http://192.168.0.5/method_get) của Raspberry Pi, bạn có thể thấy màn hình hiển thị bên trái.
- ▶ Hiện tại chưa nhận được ID hoặc Password, id và password trống như hình bên trái.
- ▶ Nhập id và mật khẩu mong muốn của bạn và nhấp vào gửi.
- ▶ Bạn có thể thấy nội dung ID và MẬT KHẨU đã được thay đổi trên màn hình của trang web.

## Thứ tự các cuộc gọi giữa trình duyệt web và máy chủ bình



## POST

### | Tham số phương thức POST

- ▶ Không giống như phương thức GET, phương thức POST không thực hiện Yêu cầu thông qua các tham số tới URL.
- ▶ Phương thức POST phải được gọi với thuộc tính method="post" trong thẻ HTML "<form method="post" action="/method\_get\_act">"
- ▶ Hãy lấy id và mật khẩu đã nhập bằng phương thức POST Yêu cầu.

## | ~/webapp/templates/method\_post.html

- ▶ Mã HTML để kiểm tra phương thức đăng bài

```
1  <html>
2    <head>
3      <title>POST Method Request Test</title>
4    </head>
5
6    <body>
7      <h2>ID:{{ id }}, PASSWORD:{{ password }}</h2>
8      <form method="post" action="/method_post_act">
9        <label id="Label1">id</label>
10       <input name="id" type="text" />
11       <label id="Label1"><br />password </label>
12       <input name="password" type="text" /><br /><br />
13       <input name="Submit1" type="submit" value="submit" />
14     </form>
15   </body>
16 </html>
```

- dòng 7: Nhận giá trị truyền từ Flask Python Script và xuất ra theo định dạng
- dòng 8-14: Tạo khung nhận id và mật khẩu nhập vào và gửi thông qua nút gửi
- dòng 8: Nếu phương thức là post trong thẻ form thì Yêu cầu được thực hiện trong phương thức POST. Mã sau hành động = quyết định chức năng nào của Flask Server sẽ gọi khi nhấn nút gửi.

- ▶ Lưu tệp này ở cùng dạng với ~/rasp\_ex/webapp/templates/method\_post.html.

| ~/rasp\_ex/webapp/method\_post.py

- ▶ Tập lệnh Python để kiểm tra phương thức POST

```
1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/method_post', methods=['GET', 'POST'])
6  def method_post():
7      return render_template('method_post.html')
8
9  @app.route('/method_post_act', methods=['GET', 'POST'])
10 def method_post_act():
11     if request.method == 'POST':
12         id = request.form["id"]
13         password = request.form["password"]
14         return render_template('method_post.html', id=id, password=password)
15
16 ▶ if __name__ == '__main__':
17     app.run(debug=True, port=80, host='0.0.0.0')
18
```

- dòng 1: Truyền tham số id và mật khẩu cho method\_post.html
- dòng 5-7: Phương thức kết xuất\_post.html
- dòng 9: Được gọi bởi thẻ <form method="get" action="/method\_post\_act"> trong html
- dòng 12-13: Đọc "id" và "password" Yêu cầu trong method\_post.html. Không giống như phương thức GET, bạn có thể đọc các tham số bằng cách đọc giá trị thẻ HTML FORM bằng cách sử dụng biểu mẫu tương tự như Yêu cầu.form[""]

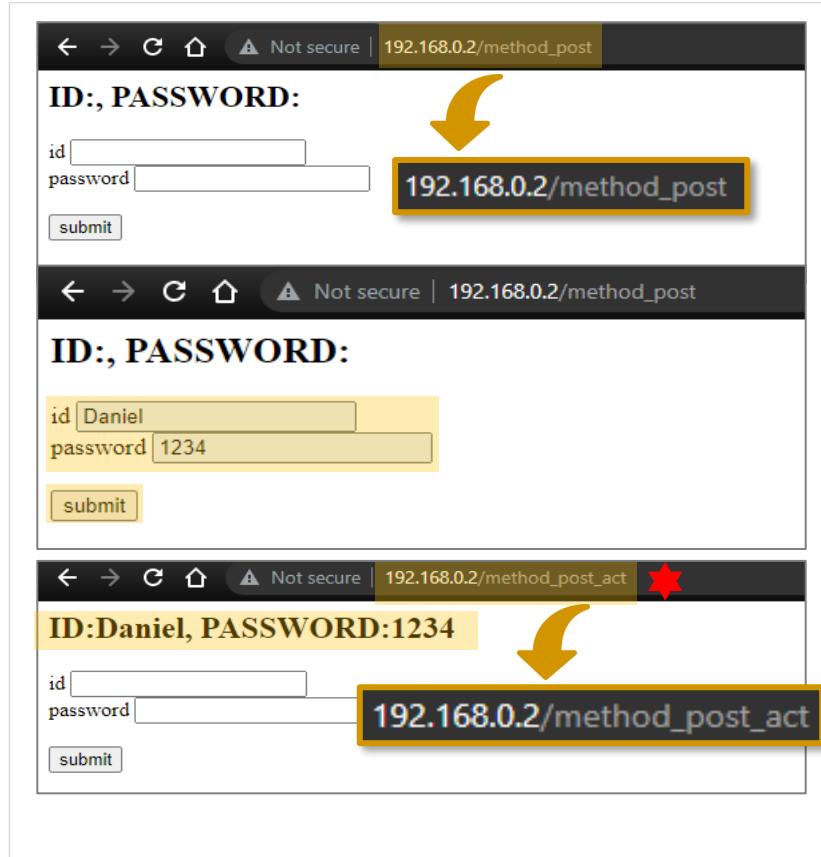
| ~/rasp\_ex/webapp/method\_post.py

- ▶ Sau khi lưu tệp này vào vị trí ~/rasp\_ex/webapp/method\_post.py, hãy thực thi method\_get.py bằng lệnh python3.

```
| > sudo python3 method_post.py
```

```
pi@raspberrypi:~/webapp $ pwd
/home/pi/webapp
pi@raspberrypi:~/webapp $ ls
html_test.py  index.py  method_get.py  method_post.py  static  templates
pi@raspberrypi:~/webapp $ ls templates/
html_test.html  method_get.html  method_post.html
pi@raspberrypi:~/webapp $ sudo python3 method_post.py
* Serving Flask app "method_post" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-242-106
```

## I màn hình kết nối



- ▶ Khi kết nối thông qua trình duyệt web với địa chỉ IP/method\_post của Raspberry Pi, bạn có thể thấy màn hình hiển thị bên trái.
- ▶ lúc đầu id và mật khẩu trống vì không nhận được id hoặc mật khẩu.
- ▶ Nhập ID và mật khẩu mong muốn của bạn vào đây và nhấn vào gửi.
- ▶ Bạn có thể thấy rằng ID và MẬT KHẨU được chỉ định khi bạn nhập.
- ▶ Ở đây, **điểm khác biệt** so với GET là trong POST, ID và mật khẩu đã gửi không được hiển thị dưới dạng tham số Yêu cầu trong URL.
- ▶ Bạn có thể tạo Yêu cầu bằng phương thức GET để truyền tham số đơn giản và sử dụng phương thức POST để truyền mật khẩu cần được bảo mật hoặc lượng dữ liệu lớn đến máy chủ web.

## Web GPIO Control

### Điều khiển BẬT/TẮT đèn LED thông qua Web

```

1  from flask import Flask, request, render_template
2  import RPi.GPIO as GPIO
3
4  LED = 17
5  GPIO.setmode(GPIO.BCM)
6  GPIO.setup(LED, GPIO.OUT)
7
8  app = Flask(__name__)
9
10 @app.route('/led_control')
11 def led_control():
12     return render_template('led_control.html')
13
14 @app.route('/led_control_act', methods=['GET'])
15 def led_control_act():
16     if request.method == 'GET':
17         status = ''
18         led = request.args["led"]
19         if led == '1':
20             GPIO.output(LED, True)
21             status = 'ON'
22         else:
23             GPIO.output(LED, False)
24             status = 'OFF'
25     return render_template('led_control.html', ret=status)
26
27 if __name__ == '__main__':
28     app.run(debug=True, port=80, host='0.0.0.0')

```

- ▶ Hãy tạo các nút LED ON và LED OFF trong trang HTML để điều khiển đèn LED được kết nối với Raspberry Pi.
- ▶ Đầu tiên, tạo và lưu tệp ~/rasp\_ex/webapp/led\_control.py
- ▶ dòng 15-25: led\_control\_act hàm điều khiển led. Bật/tắt led theo biến led với Yêu cầu
- ▶ 1 bật đèn LED, nếu không thì tắt.



```

15 def led_control_act():
16     if request.method == 'GET':
17         status = ''
18         led = request.args["led"]
19         if led == '1':
20             GPIO.output(LED, True)
21             status = 'ON'
22         else:
23             GPIO.output(LED, False)
24             status = 'OFF'
25     return render_template('led_control.html', ret=status)

```

- ▶ Truyền trạng thái cho một biến có tên là **ret** trong html (trình duyệt web)

## Điều khiển BẬT/TẮT đèn LED thông qua Web

```

1  <html>
2  <head>
3  <title>WEB LED Control</title>
4  <style type="text/css">
5      .auto-style1 {
6          text-align: center;
7      }
8      .auto-style3 {
9          background-color: #008000;
10     }
11     .auto-style6 {
12         border-style: solid;
13         border-color: #000000;
14         text-align: center;
15         color: #FFFFFF;
16         background-color: #FF9900;
17     }
18 </style>
19 </head>
20 <body>
21 <center>
22 <strong><br>HOME IOT Service<br><br></strong>
23 <table style="width: 50%">
24 <tr>
25     <td class="auto-style6" style="height: 81; width: 30%"><strong>
26         <a href="led_control_act?led=1">LED ON</a></strong></td>
27     <td class="auto-style1" style="height: 81; width: 50%"> </td>
28     <td class="auto-style6" style="height: 81; width: 30%;"><strong>
29         <a href="led_control_act?led=2">LED OFF</a></strong></td>
30     </td>
31 </tr>
32 </table>
33 <strong><br>LED is {{ ret }}</strong>
34 </center>
35 </body>
36 </html>

```

- ▶ Viết và lưu tệp ~/rasp\_ex/webapp/template/led\_control.html
  - dòng 24-30: Định nghĩa của nút để vận hành đèn LED
  - dòng 26, 29: Truyền số nguyên tương ứng dưới dạng tham số Yêu cầu là Yêu cầu.arg["led"] trong hàm led\_control\_act() của led\_control.py
  - dòng 32: Hiển thị trạng thái của đèn LED bằng ret khi trạng thái của led\_control.py được phân phối dưới dạng ret



```

24 <tr>
25     <td class="auto-style6" style="height: 81; width: 30%"><strong>
26         <a href="led_control_act?led=1">LED ON</a></strong></td>
27     <td class="auto-style1" style="height: 81; width: 50%"> </td>
28     <td class="auto-style6" style="height: 81; width: 30%;"><strong>
29         <a href="led_control_act?led=2">LED OFF</a></strong></td>
30     </td>
31 </tr>
32 </table>
33 <strong><br>LED is {{ ret }}</strong>

```

### | Điều khiển BẬT/TẮT đèn LED thông qua Web

- ▶ Sau khi lưu led\_control.py và led\_control.html, hãy thực hiện lệnh sudo python3 ~/webapp/led\_control.py.
- ▶ Khi kết nối thông qua trình duyệt web với địa chỉ IP/led\_control của Raspberry Pi, bạn có thể thấy màn hình bên dưới.
- ▶ Bằng cách nhấn từng nút, bạn có thể kiểm tra văn bản của trang web thay đổi như thế nào và đèn LED thay đổi như mong muốn.

```
> sudo python3 rasp_ex/webapp/led_control.py
```

```
pi@raspberrypi:~ $ sudo python3 rasp_ex/webapp/led_control.py
* Serving Flask app "led_control" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
/home/pi/webapp/led_control.py:6: RuntimeWarning: This channel is already in use, continuing anyway.  Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(LED, GPIO.OUT)
* Debugger is active!
* Debugger PIN: 117-242-106
```

LED ON

HOME IOT ứng  
dụng

LED is OFF

LED OFF

Bài 5.

# Sử dụng Raspberry Pi làm Máy chủ DB

- | 5.1. Sử dụng MariaDB
- | 5.2. Sử dụng HeidiSQL
- | 5.3. Ứng dụng của MariaDB

## MariaDB

### | Một hệ thống cơ sở dữ liệu quan hệ mã nguồn mở

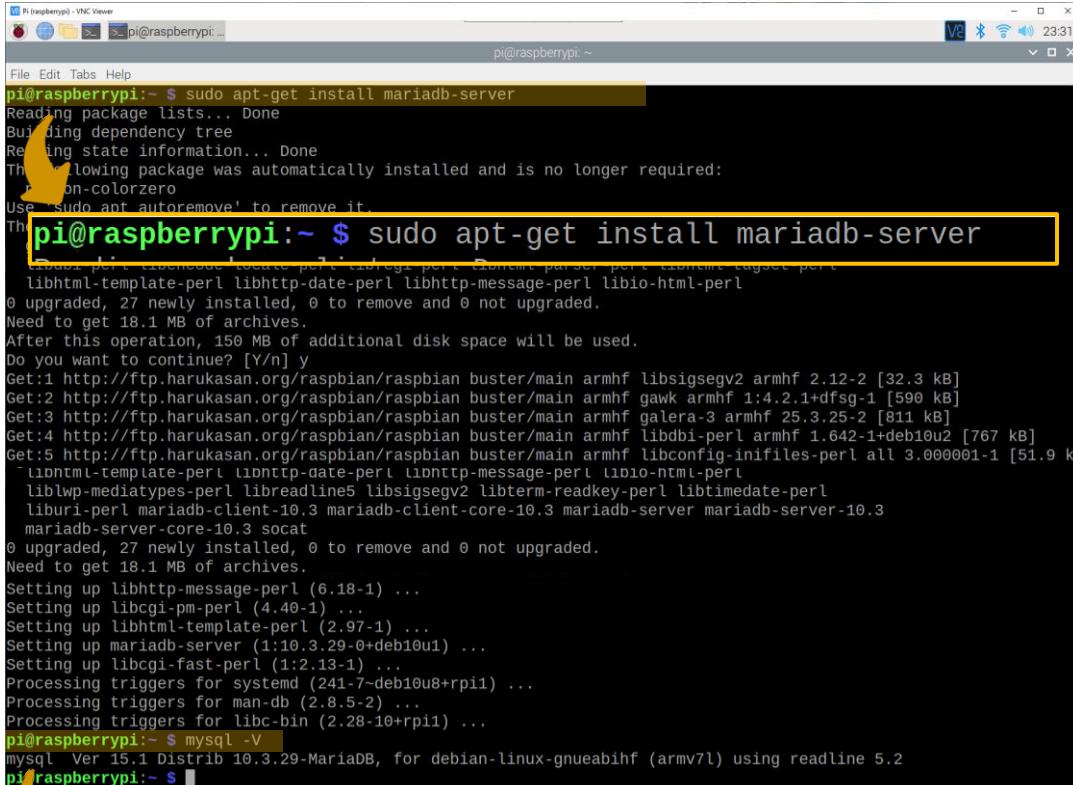
- ▶ MariaDB là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS).
- ▶ MariaDB dựa trên cùng một mã nguồn với MySQL, vì vậy nó có giấy phép GPL v2
- ▶ Nhà phân phối phải chia sẻ bản quyền với Monty Program AB để duy trì khả năng tương thích cao với MySQL
- ▶ MariaDB dùng chung mã nguồn với MySQL nên cách sử dụng và cấu trúc cũng giống như của MySQL.



## 5.1. Sử dụng MariaDB

## Bài 05

### Cài đặt MariaDB trên Raspberry Pi

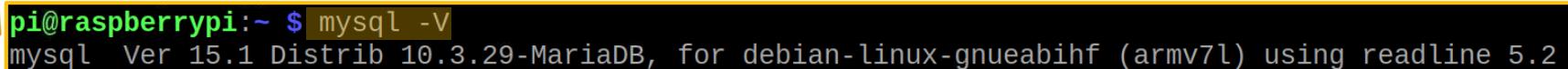


```
pi@raspberrypi:~ $ sudo apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
pi@raspberrypi:~ $ sudo apt-get install mariadb-server
libdbi-perl libdigest_md5-perl libdigest_stem-perl libdigest_tiny-perl libdigest_tiny-perl
libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
0 upgraded, 27 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.1 MB of archives.
After this operation, 150 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf libsigsegv2 armhf 2.12-2 [32.3 kB]
Get:2 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf gawk armhf 1:4.2.1+dfsg-1 [590 kB]
Get:3 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf galera-3 armhf 25.3.25-2 [811 kB]
Get:4 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf libdbi-perl armhf 1.642-1+deb10u2 [767 kB]
Get:5 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf libconfig-inifiles-perl all 3.000001-1 [51.9 kB]
libhtml-template-perl libhttp-date-perl libntp-message-perl libio-html-perl
liblwp-mediatypes-perl libreadline5 libsigsegv2 libterm-readkey-perl libtimeinterval-perl
liburi-perl mariadb-client-10.3 mariadb-client-core-10.3 mariadb-server mariadb-server-10.3
mariadb-server-core-10.3 socat
0 upgraded, 27 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.1 MB of archives.
Setting up libhttp-message-perl (6.18-1) ...
Setting up libcgi-pm-perl (4.40-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mariadb-server (1:10.3.29-0+deb10u1) ...
Setting up libcgi-fast-perl (1:2.13-1) ...
Processing triggers for systemd (241-7-deb10u8+rp1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rp1) ...
pi@raspberrypi:~ $ mysql -V
mysql Ver 15.1 Distrib 10.3.29-MariaDB, for debian-linux-gnueabihf (armv7l) using readline 5.2
pi@raspberrypi:~ $
```

- Thực thi lệnh `sudo apt-get install mariadb-server` để cài đặt MariaDB trên Raspberry Pi và trả lời bằng `y` cho thông báo `[Y/n]` xuất hiện ở giữa.

> `sudo apt-get install mariadb-server`

- Sau khi cài đặt xong, dùng lệnh `mysql -V` để kiểm tra phiên bản DB.



```
pi@raspberrypi:~ $ mysql -V
mysql Ver 15.1 Distrib 10.3.29-MariaDB, for debian-linux-gnueabihf (armv7l) using readline 5.2
```

## thiết lập tài khoản root

### I Setting up the root account

- Thực hiện lệnh `sudo mysql -u root` để vào dấu nhắc lệnh MariaDB.
- Thực hiện lệnh sử dụng `mysql`; để chọn mysql DB.
- Bạn có thể thấy rằng dấu nhắc lệnh đã được thay đổi từ MaraiDB [(None)]> thành MariaDB[mysql]>.

```
> sudo mysql -u root
```

The screenshot shows a terminal window titled "Pi (raspberrypi) - VNC Viewer". The window title bar says "pi@raspberrypi: ~". The status bar at the top right shows "23:48". The terminal content is as follows:

```
File Edit Tabs Help
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service
.
Setting up libhttp-message-perl (6.18-1) ...
Setting up libcgi-pm-perl (4.40-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mariadb-server (1:10.3.29-0+deb10u1) ...
Setting up libcgi-fast-perl (1:2.13-1) ...
Processing triggers for systemd (241-7-deb10u8+rpi1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rpi1) ...
pi@raspberrypi:~ $ mysql -v
mysql Ver 15.1 Distrib 10.3.29-MariaDB, for debian-linux-gnueabihf (armv7l) using readline 5.2
pi@raspberrypi:~ $ sudo mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 1
Server version: 10.3.29-MariaDB MariaDB Server
pi@raspberrypi:~ $ sudo mysql -u root
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current query.
MariaDB [(none)]> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [mysql]>
```

A yellow arrow points from the command `pi@raspberrypi:~ $ sudo mysql -u root` to the resulting prompt `MariaDB [(none)]>`. Another yellow arrow points from the command `MariaDB [(none)]> use mysql;` to the confirmation message `Database changed`.

### I Thiết lập tài khoản root

```
MariaDB [(none)]> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]> select user, host, password from user;
+-----+-----+-----+
| user | host   | password |
+-----+-----+-----+
| root | localhost |          |
+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [mysql]> update user set password=password('1234') where user='root';
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [mysql]>
```

- ▶ Chọn lệnh để chọn kiểm tra các giá trị user, host, password của user tabledml của mysql DB
- ▶ **chọn người dùng, máy chủ, mật khẩu từ người dùng;** Đây là một lệnh cú pháp SQL để lấy các thuộc tính người dùng, máy chủ và mật khẩu từ bảng người dùng.
- ▶ Bạn có thể thấy rằng mật khẩu cho người dùng root trống. Chạy lệnh cập nhật để cập nhật mật khẩu cho người dùng root.
- ▶ Lệnh cập nhật mật khẩu của người dùng root thành 1234 là **cập nhật bộ mật khẩu người dùng = mật khẩu ('1234')** trong đó **người dùng = 'root'**.
- ▶ Các đặc quyền **tạo ra lệnh**; phải được thực thi để cập nhật được phản ánh chính xác.

### | Thiết lập tài khoản root

```
MariaDB [mysql]> select user, host, password from user;
+-----+-----+-----+
| user | host   | password          |
+-----+-----+-----+
| root | localhost | *A4B6157319038724E3560894F7F932C8886EBFCF |
+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [mysql]> exit
Bye
pi@raspberrypi:~ $
```



- ▶ Nếu bạn kiểm tra lại các giá trị thuộc tính của người dùng, máy chủ và mật khẩu trong bảng người dùng, bạn có thể xác nhận rằng mật khẩu đã được thay đổi.
- ▶ Bởi vì mật khẩu dường như ở trạng thái được băm, nên nó không giống như mật khẩu đã đặt, 1234, mà ở **dạng được băm**.
- ▶ Thoát khỏi dấu nhắc lệnh MariaDB bằng **lệnh thoát**.

## Cài đặt quyền của người dùng DB

### I Setting all permissions for root user of DB mysql

The screenshot shows a terminal window titled 'Pi (raspberrypi) - VNC Viewer' with the command line interface. A yellow arrow points to the first command:

```
pi@raspberrypi:~ $ sudo mysql -u root
```

The terminal then displays the MySQL monitor prompt. A yellow box highlights the 'use mysql' command, and another yellow box highlights the 'grant' command in the subsequent 'grant all' command:

```
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 38  
Server version: 10.3.29-MariaDB-0+deb10u1 Raspbian 10  
Copyright (c) 2023 MariaDB Corporation and its affiliates. All rights reserved.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> use mysql  
Reading table information for this schema  
You can turn off this feature to get a quicker startup with -A  
Database changed  
MariaDB [mysql]> grant all on *.* to 'root'@'%' identified by '1234';  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [mysql]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [mysql]> exit  
exit
```

- ▶ Thực hiện lệnh `sudo mysql -u root` để vào dấu nhắc lệnh MariaDB.
- ▶ > `sudo mysql -u root`
- ▶ Thực hiện lệnh sử dụng `mysql`; để chọn mysql DB
- ▶ Ý nghĩa của lệnh cấp là **cấp quyền** cho tất cả các bảng và DB, đồng thời đặt nó ở chế độ có thể truy cập được từ cả vị trí cục bộ và từ xa. Sau khi được xác định bởi, hãy nhập mật khẩu của người dùng root.
- ▶ Thực hiện lệnh tạo ra **đặc quyền** để áp dụng các sửa đổi
- ▶ Thực hiện lệnh Thoát để điều hướng đến thiết bị đầu cuối Raspberry Pi

Bài 5.

# Sử dụng Raspberry Pi làm Máy chủ DB

- | 5.1. Sử dụng MariaDB
- | **5.2. Sử dụng HeidiSQL**
- | 5.3 Ứng dụng của MariaDB

## HeidiSQL

- | Môi trường GUI thuận tiện cho phép bạn truy cập MariaDB từ xa từ PC, tạo bảng và quản lý chúng
  - ▶ HeidiSQL là một ứng dụng khách mã nguồn mở dành cho MySQL. Nó là công cụ tốt nhất để quản lý cơ sở dữ liệu MySQL.
  - ▶ Nó cung cấp một giao diện đồ họa mạnh mẽ để quản lý bảng, nhật ký và người dùng cơ sở dữ liệu MySQL.

- Những đặc điểm chính

- Tạo báo cáo SQL

- Đồng bộ hóa thông tin từ hai bảng khác nhau

- Nhập tệp văn bản và trích xuất bảng ở các định dạng khác nhau

- Chỉnh sửa cơ sở dữ liệu bằng cách tạo và xóa bảng, chèn và chỉnh sửa nhật ký cũng như xóa nhật ký, bao gồm khu vực MEMO hỗ trợ nhiều thứ như Bitmap, GIF và JPEG

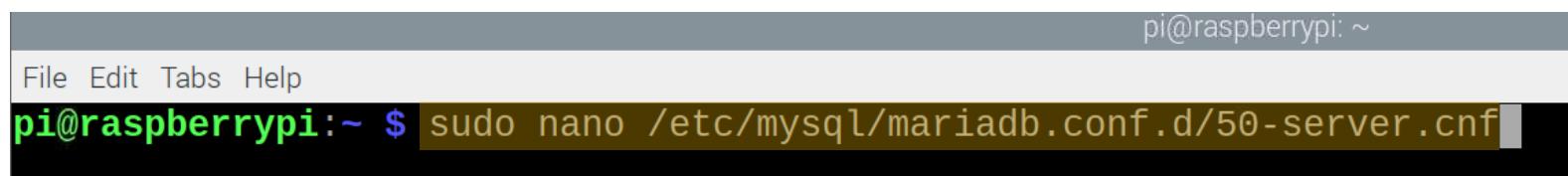


## Cài đặt truy cập bên ngoài HeidiSQL

### I Cài đặt để cho phép truy cập bên ngoài

- ▶ Sẽ không thành vấn đề nếu bạn chỉ kết nối từ bên trong Raspberry Pi, nhưng bạn nên cho phép truy cập bên ngoài truy cập MariaDB từ xa vào từ PC để tạo và quản lý bảng.
- ▶ Thực hiện lệnh sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf để chỉnh sửa tệp này

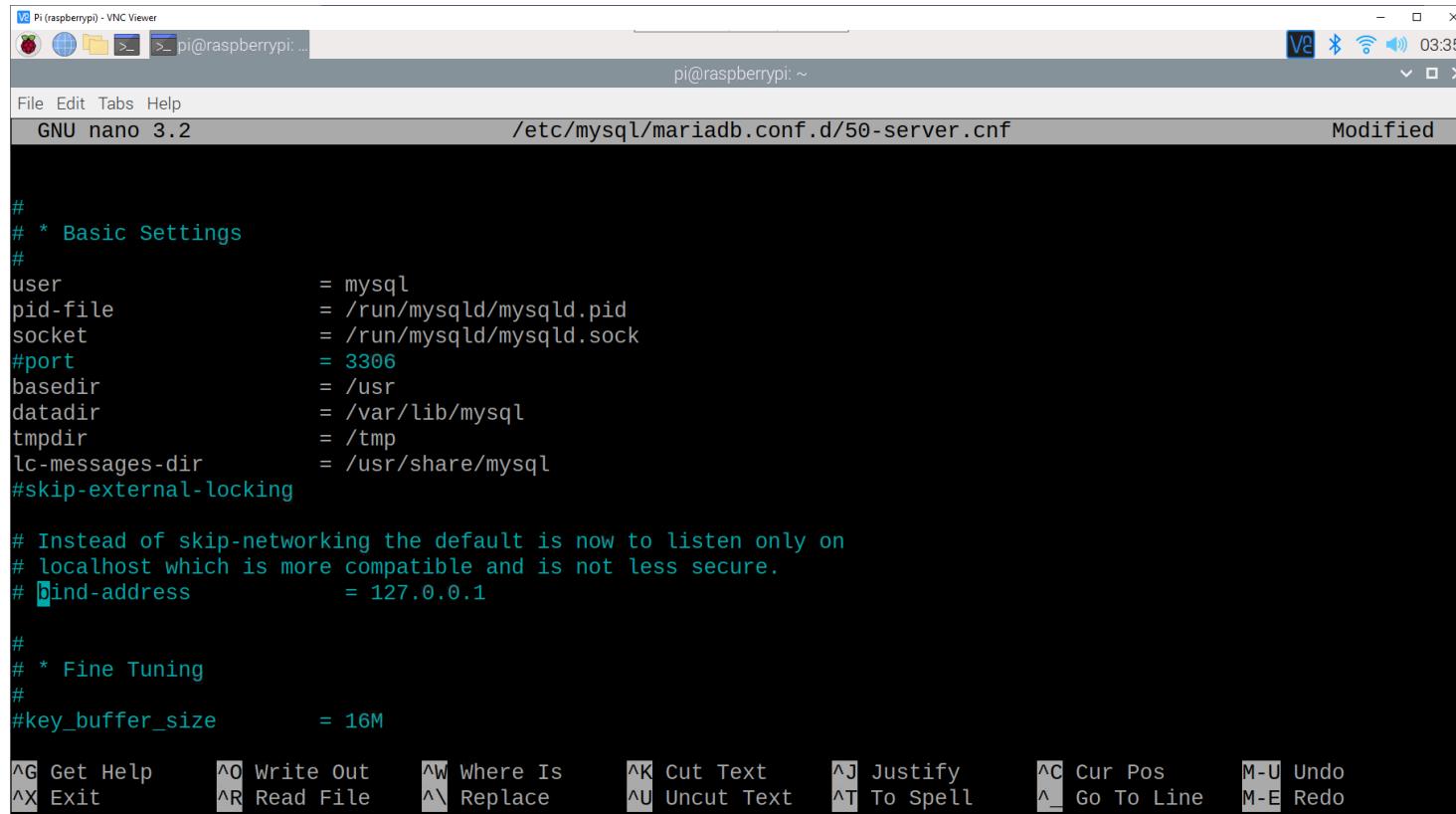
```
> sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```



A screenshot of a terminal window on a Raspberry Pi. The window has a dark grey header bar with the text "pi@raspberrypi: ~". Below the header is a light grey menu bar with the options "File", "Edit", "Tabs", and "Help". The main area of the terminal is black and contains the command "pi@raspberrypi:~ \$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf" in white text. The cursor is visible at the end of the command line.

### Cài đặt để cho phép truy cập bên ngoài

- ▶ Nhận xét cài đặt liên kết địa chỉ mạng của riêng bạn (địa chỉ liên kết)
- ▶ Sử dụng Ctrl + o → Enter → Ctrl + x để lưu và thoát các chỉnh sửa nano.



The screenshot shows a VNC session titled "Pi (raspberrypi) - VNC Viewer". The terminal window displays the contents of the file "/etc/mysql/mariadb.conf.d/50-server.cnf". The file contains MySQL configuration settings, including port 3306 and bind-address 127.0.0.1. The nano editor interface is visible at the bottom, showing keyboard shortcuts for various functions.

```
# * Basic Settings
#
user          = mysql
pid-file      = /run/mysqld/mysqld.pid
socket        = /run/mysqld/mysqld.sock
#port         = 3306
basedir       = /usr
datadir       = /var/lib/mysql
tmpdir        = /tmp
lc-messages-dir = /usr/share/mysql
#skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address    = 127.0.0.1

#
# * Fine Tuning
#
#key_buffer_size     = 16M
```

### I Cài đặt để cho phép truy cập bên ngoài

- ▶ Đặt bộ lọc TCP để cho phép INPUT và OUTPUT trên cổng 3306 được sử dụng bởi MariaDB

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
pi@raspberrypi:~ $ sudo iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
pi@raspberrypi:~ $ sudo iptables -A OUTPUT -p tcp --dport 3306 -j ACCEPT
pi@raspberrypi:~ $ sudo iptables-save
# Generated by xtables-save v1.8.2 on Tue Sep 21 03:40:07 2021
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 3306 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 3306 -j ACCEPT
COMMIT
# Completed on Tue Sep 21 03:40:07 2021
pi@raspberrypi:~ $
```

- sudo iptables -A INPUT -p tcp --dport 3306 -j CHẤP NHẬN
- sudo iptables -A ĐẦU RA -p tcp -dport 3306 -j CHẤP NHẬN
- sudo iptables -save

```
> sudo iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
> sudo iptables -A OUTPUT -p tcp -dport 3306 -j ACCEPT
> sudo iptables -save
```

- ▶ Sau khi thực hiện các lệnh này, hãy khởi động lại Raspberry Pi để phản ánh các cài đặt và hoàn tất quá trình chuẩn bị cho truy cập bên ngoài.

## Cài đặt HeidiSQL

### I Truy cập trang web HeidiSQL và tải xuống HeidiSQL cho Windows

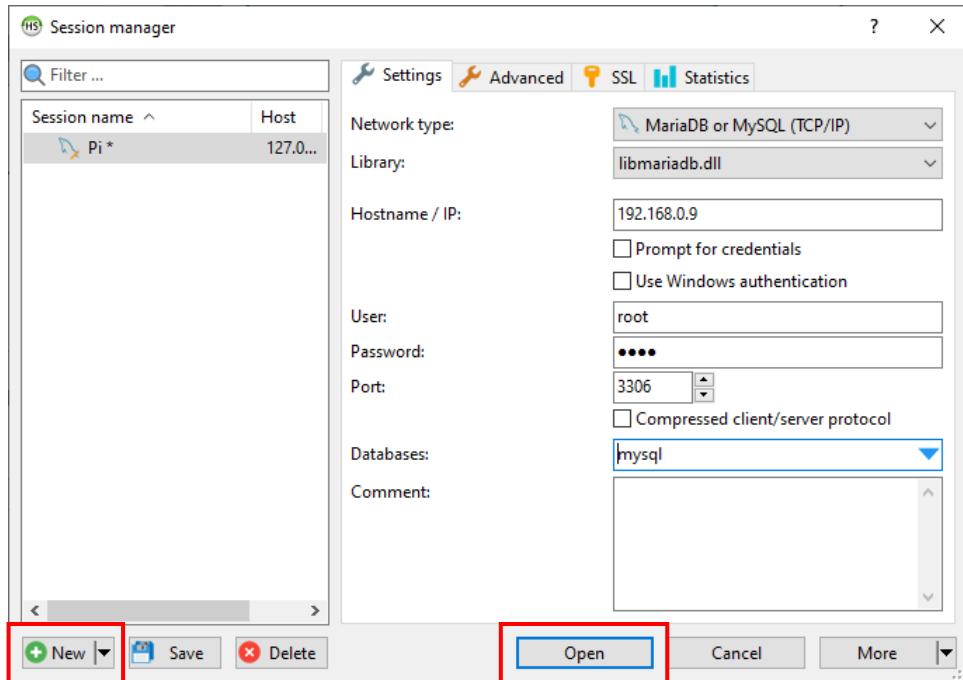


- ▶ <https://www.heidisql.com/download.php>
- ▶ Trình cài đặt, Nhấp vào 32/64 bit để tải xuống Tệp cài đặt
  
- ▶ Khi chạy tệp Cài đặt, hãy giữ nguyên tất cả các cài đặt cơ bản và chạy HeidiSQL sau khi hoàn thành

# HeidiSQL

## | Thực thi HeidiSQL

- ▶ Tạo Phiên mới để kết nối với Raspberry Pi trong Trình quản lý phiên
- ▶ Nhấp vào nút Mới ở dưới cùng bên trái để tạo Phiên và nhập thông tin như hình bên phải.



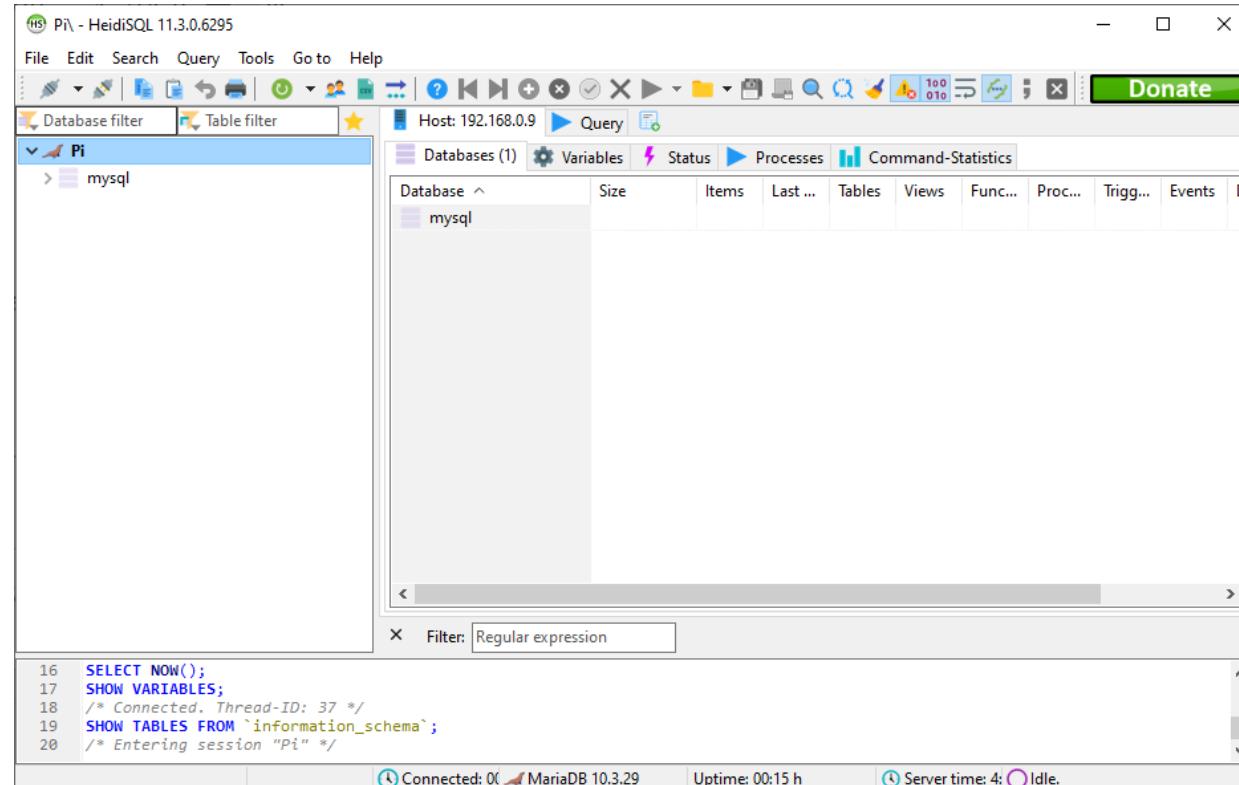
- Tên phiên: Pi
- Hostname/IP: Địa chỉ IP của Raspberry Pi
- Người dùng: gốc
- Mật khẩu: Mật khẩu của người dùng root đã đặt
- Cơ sở dữ liệu: mysql



Sau khi nhập thông tin,  
nhấp vào Mở ở dưới cùng.

### Kiểm tra phiên Pi

- Nếu kết nối thành công, bạn có thể kiểm tra trạng thái và dữ liệu của mysql DB trên Raspberry Pi như hình bên dưới.
- Bây giờ, hãy tạo một bảng.



### | Lược đồ bảng

- ▶ Hãy thực hành với một ví dụ lưu trữ các giá trị đo được của cảm biến nhiệt độ và độ ẩm trong bảng DB.
- ▶ Cấu trúc bảng như dưới đây.

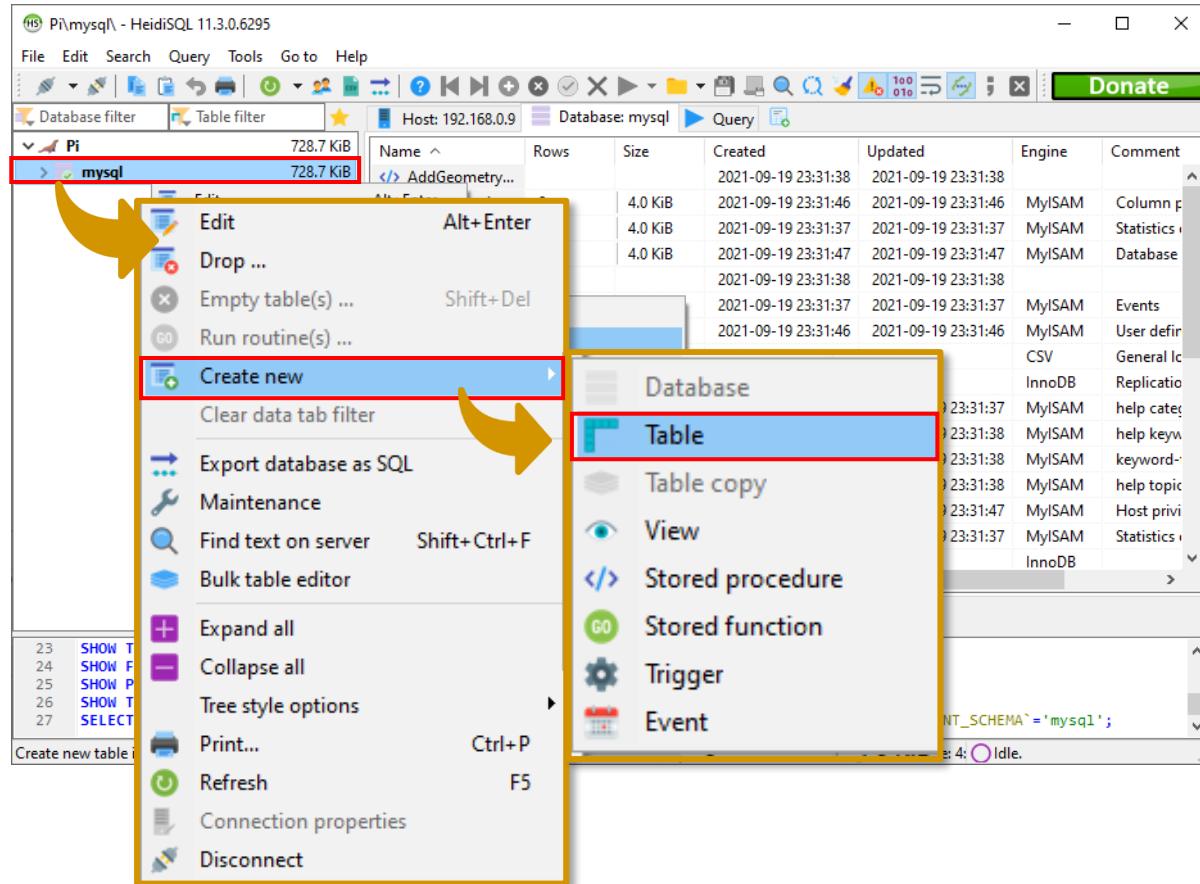
temperature
datatime (DATETIME)
temp (FLOAT)
humid (FLOAT)

## 5.2. Sử dụng HeidiSQL

## Bài 05

### Tạo bảng

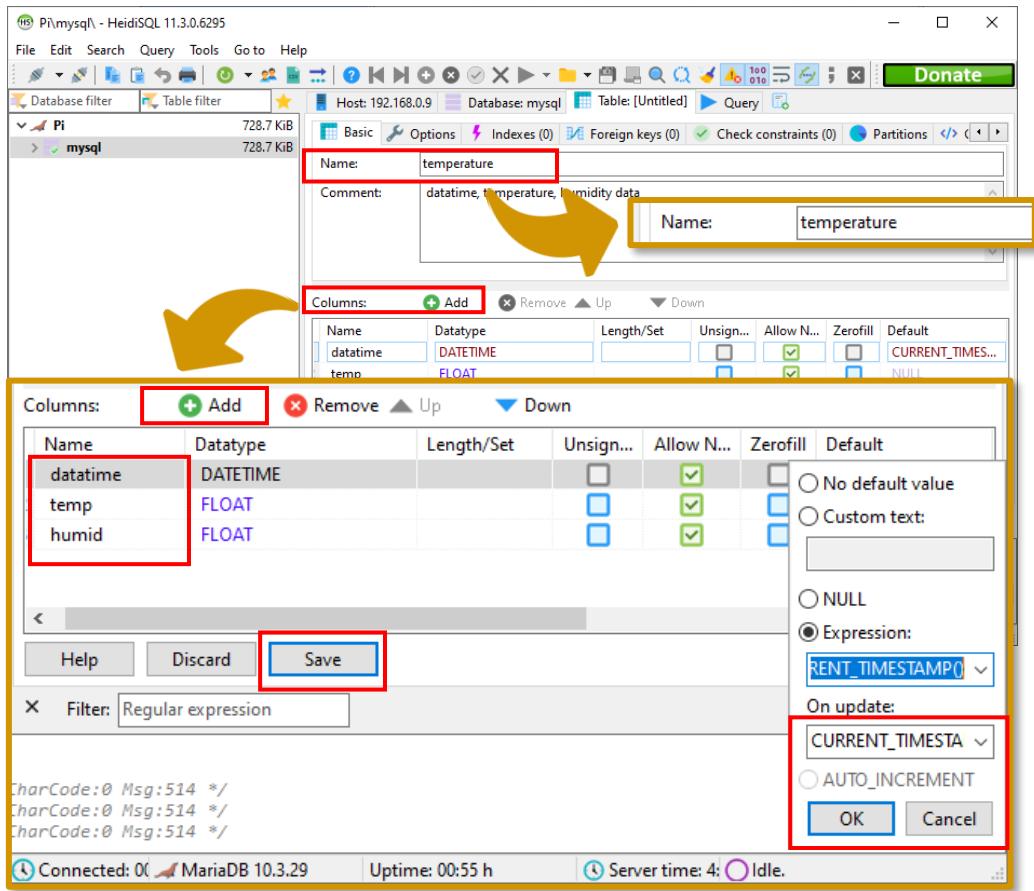
- Nhấp chuột phải vào mysql trên Pi → Nhấp vào Tạo mới → Nhấp vào Bảng để tạo bảng mới.



## 5.2. Sử dụng HeidiSQL

## Bài 05

### thiết lập bảng



- Nhập nhiệt độ làm tên của bảng và nhập mô tả trong Nhận xét.
- Sử dụng nút Thêm ở bên phải Cột: để thêm cột. Thêm ba cột: thời gian dữ liệu, nhiệt độ và độ ẩm.
- Đặt kiểu dữ liệu của cột theo lược đồ bảng và thay đổi giá trị mặc định của thời gian dữ liệu thành CURRENT\_TIMESTAMP().
- Sau khi thiết lập mọi thứ, nhấp vào nút Lưu để tạo bảng.

## DB Query

### I Query syntax

- ▶ Truy vấn là một cú pháp để thêm, sửa, xóa và đọc dữ liệu từ DB.
- ▶ Có nhiều cú pháp Truy vấn SQL khác nhau, nhưng chúng ta sẽ xem xét ngắn gọn các câu lệnh Truy vấn cơ bản và thường được sử dụng nhất.
  - Câu lệnh INSERT để thêm dữ liệu vào bảng

```
INSERT INTO TABLE_NAME([COL_NAME],[COL_NAME]) VALUES ([DATA], [DATA]);
```

- Câu lệnh SELECT để đọc dữ liệu từ bảng

```
SELECT [COL_NAME],[COL_NAME] FROM TABLE_NAME [WHERE CONDITION];
```

- Câu lệnh CẬP NHẬT để sửa đổi dữ liệu được lưu trữ trong bảng

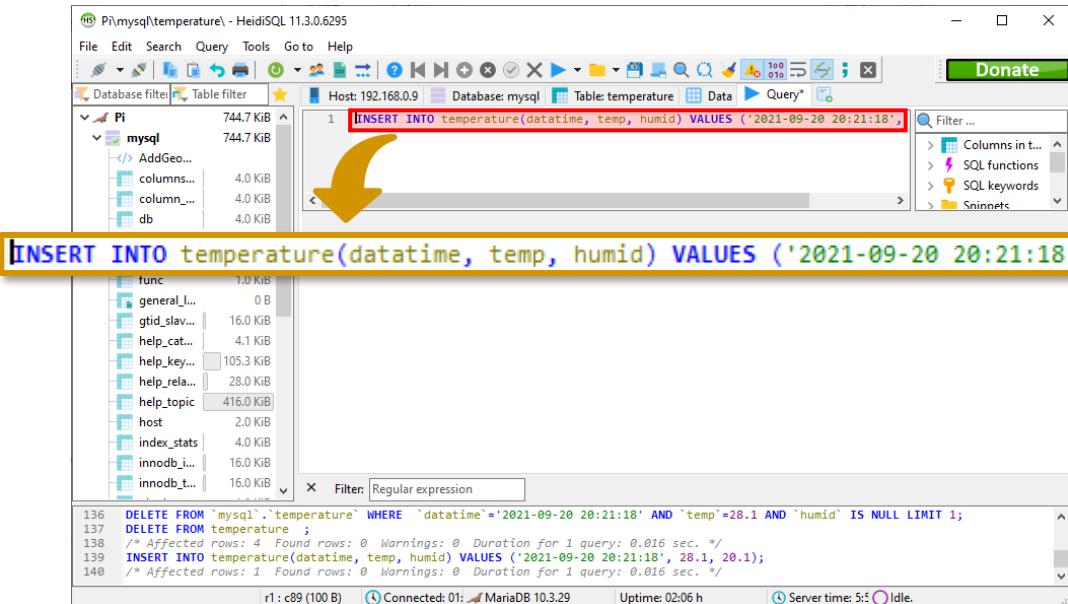
```
UPDATE TABLE_NAME SET COL_NAME = 'VALUE' [WHERE CONDITION];
```

- Câu lệnh DELETE để xóa dữ liệu được lưu trữ trong một bảng

```
DELETE FROM TABLE_NAME [WHERE CONDITION];
```

### | Áp dụng các câu lệnh Truy vấn đơn giản cho DB mẫu

- Thực hiện các câu lệnh và kiểm tra kết quả



The screenshot shows the HeidiSQL interface. In the main query window, the following SQL command is highlighted with a yellow arrow pointing to it:

```
INSERT INTO temperature(datatime, temp, humid) VALUES ('2021-09-20 20:21:18', 28.1, 20.1);
```

The status bar at the bottom indicates the connection details: r1 : c89 (100 B), Connected: 01: MariaDB 10.3.29, Uptime: 02:06 h, Server time: 5:5 Idle.

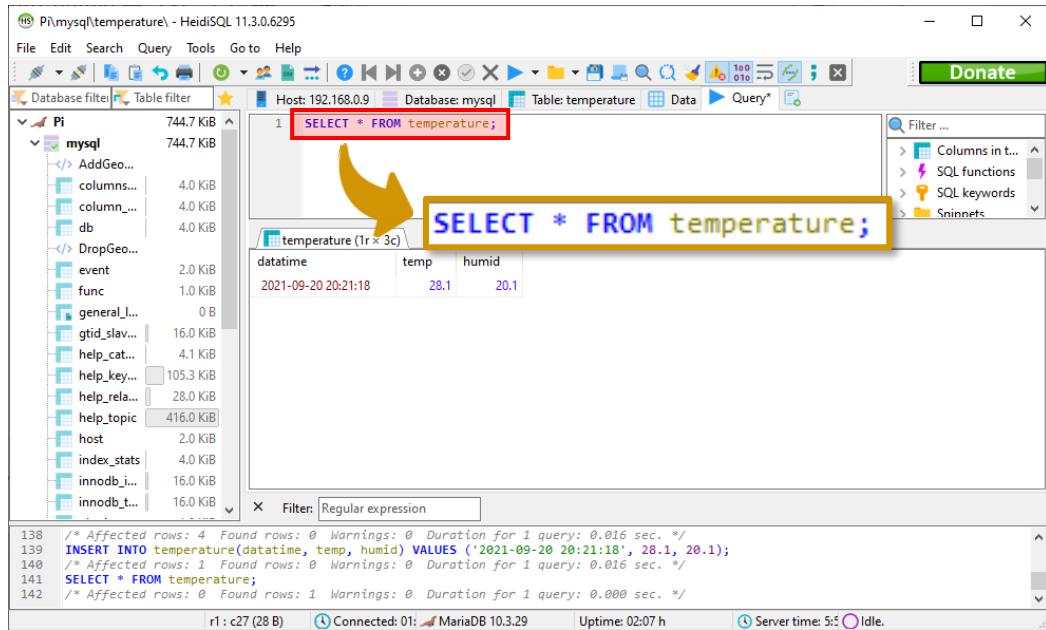
- Thêm '2021-09-20 20:21:18' vào cột ngày giờ trong bảng nhiệt độ, 28.1 vào cột tạm thời và 20.1 vào cột độ ẩm

```
INSERT INTO temperature(datatime, temp, humid)
VALUES ('2021-09-20 20:21:18', 28.1, 20.1);
```

## 5.2. Sử dụng HeidiSQL

## Bài 05

### Áp dụng các câu lệnh Truy vấn đơn giản cho DB mẫu



The screenshot shows the HeidiSQL interface connected to a MySQL database named 'Pi'. The left sidebar lists various databases and tables. In the main pane, a query window displays the following:

```
Host: 192.168.0.9 Database: mysql Table: temperature Data Query*  
1 SELECT * FROM temperature;
```

The results pane shows a single row from the 'temperature' table:

datatime	temp	humid
2021-09-20 20:21:18	28.1	20.1

The status bar at the bottom indicates the connection details: r1 : c27 (28 B) Connected: 01: MariaDB 10.3.29 Uptime: 02:07 h Server time: 5:5 Idle.

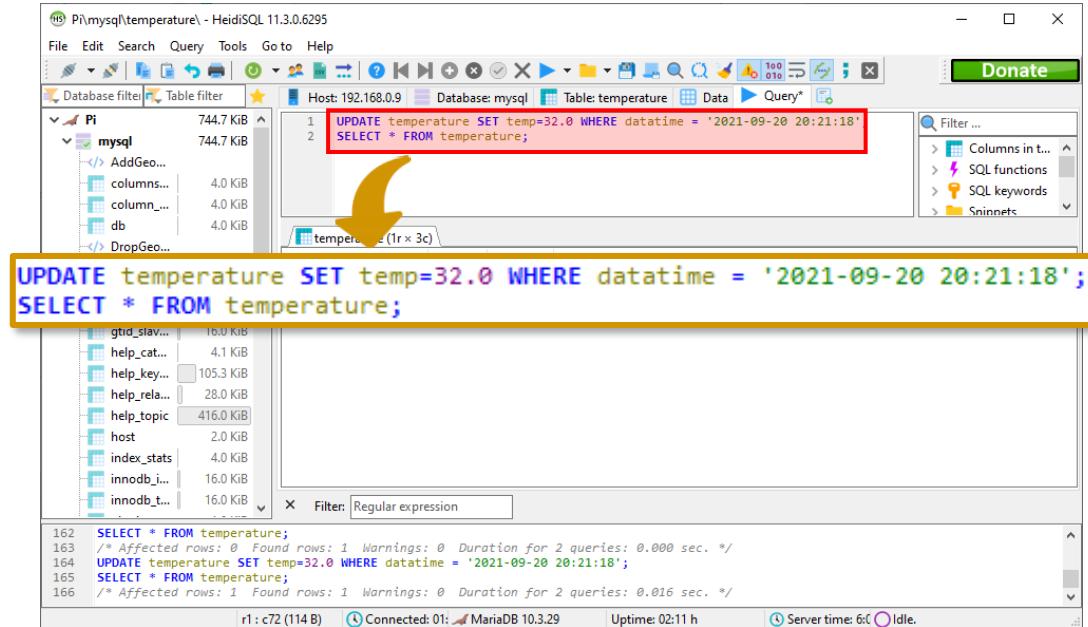
- Đọc tất cả các giá trị thuộc tính (\*) trong bảng nhiệt độ. Nghĩa là, nó đọc các giá trị của tất cả các thuộc tính thời gian, nhiệt độ và độ ẩm.

```
SELECT * FROM temperature;
```

## 5.2. Sử dụng HeidiSQL

## Bài 05

### Áp dụng các câu lệnh Truy vấn đơn giản cho DB mẫu



```
P:\mysql\temperature\ - HeidiSQL 11.3.0.6295
File Edit Search Query Tools Go to Help
Database filter Table filter
Host: 192.168.0.9 Database: mysql Table: temperature Data Query*
Donate
Database: mysql Table: temperature
1 UPDATE temperature SET temp=32.0 WHERE datatime = '2021-09-20 20:21:18';
2 SELECT * FROM temperature;
temp (1r x 3c)
gtid_slave_pos 16.0 KiB
help_category 4.1 KiB
help_keyword 105.3 KiB
help_relation 28.0 KiB
help_topic 416.0 KiB
host 2.0 KiB
index_stats 4.0 KiB
innodb_index_size 16.0 KiB
innodb_table_size 16.0 KiB
Filter: Regular expression
162 SELECT * FROM temperature;
163 /* Affected rows: 1 Found rows: 1 Warnings: 0 Duration for 2 queries: 0.000 sec. */
164 UPDATE temperature SET temp=32.0 WHERE datatime = '2021-09-20 20:21:18';
165 SELECT * FROM temperature;
166 /* Affected rows: 1 Found rows: 1 Warnings: 0 Duration for 2 queries: 0.016 sec. */
r1: c72 (114 B) Connected: 01: MariaDB 10.3.29 Uptime: 02:11 h Server time: 6:0 Idle.
```

- Thay đổi giá trị thuộc tính tạm thời của dữ liệu có thời gian dữ liệu là '2021-09-20 20:21:18' thành 32,0 trong bảng nhiệt độ.

**CẬP NHẬT** nhiệt độ SET temp=32.0 WHERE datatime = '2021-09-20 20:21:18';

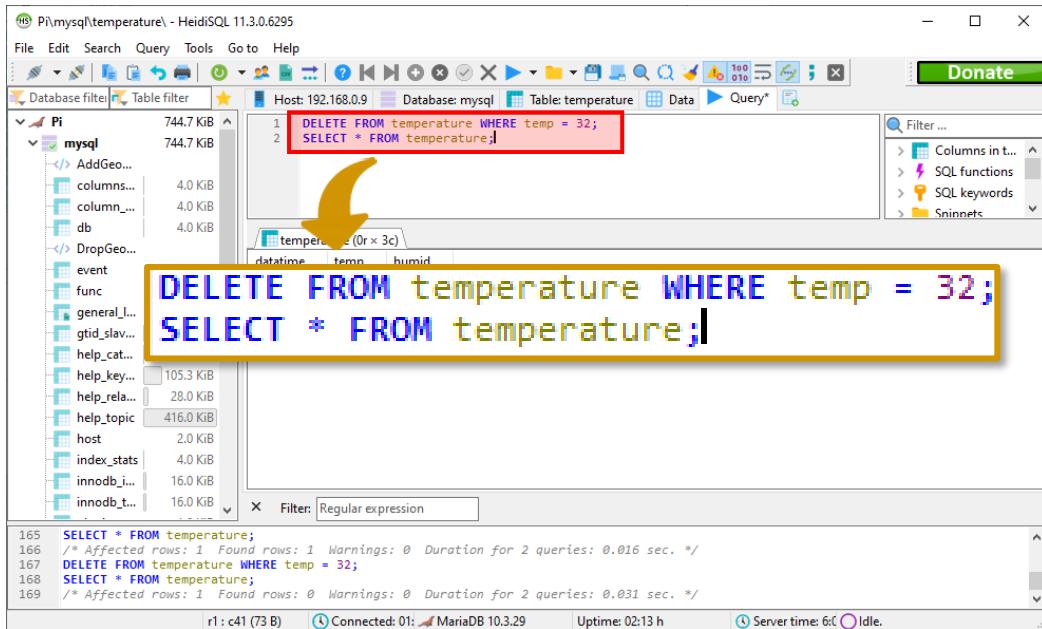
- Đọc tất cả dữ liệu trong bảng để xem các thay đổi

**CHỌN \* TỪ** nhiệt độ;

## 5.2. Sử dụng HeidiSQL

## Bài 05

### Áp dụng các câu lệnh Truy vấn đơn giản cho DB mẫu



```
DELETE FROM temperature WHERE temp = 32;
SELECT * FROM temperature;
```

```
165 SELECT * FROM temperature;
166 /* Affected rows: 1 Found rows: 1 Warnings: 0 Duration for 2 queries: 0.016 sec. */
167 DELETE FROM temperature WHERE temp = 32;
168 SELECT * FROM temperature;
169 /* Affected rows: 1 Found rows: 0 Warnings: 0 Duration for 2 queries: 0.031 sec. */
```

- Xóa dữ liệu có giá trị thuộc tính tạm thời là 32,0 khỏi bảng nhiệt độ

```
DELETE FROM temperature WHERE temp = 32;
```

- Đọc tất cả dữ liệu trong bảng để xem các thay đổi

```
SELECT * FROM temperature;
```

Bài 5.

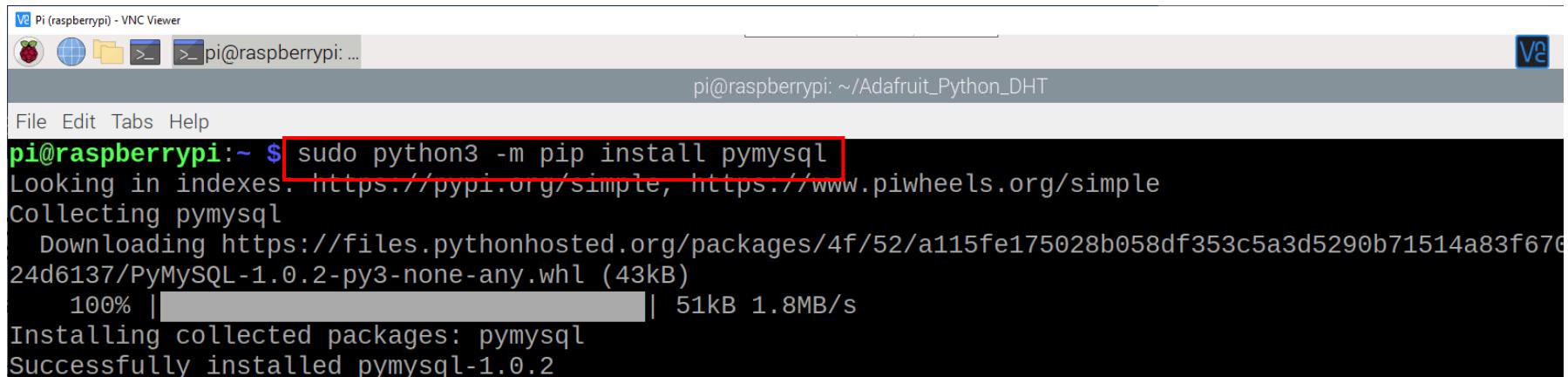
# Sử dụng Raspberry Pi làm Máy chủ DB

- | 5.1. Sử dụng MariaDB
- | 5.2. Sử dụng HeidiSQL
- | **5.3 Ứng dụng của MariaDB**

## Sử dụng cảm biến nhiệt độ và độ ẩm

### I Sử dụng Adafruit để sử dụng cảm biến nhiệt độ và độ ẩm

- ▶ sudo python3 -m pip install pymysql
  - Cài đặt thư viện pymysql cho phép python3 hoạt động với MariaDB



```
pi@raspberrypi:~ $ sudo python3 -m pip install pymysql
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pymysql
  Downloading https://files.pythonhosted.org/packages/4f/52/a115fe175028b058df353c5a3d5290b71514a83f670
24d6137/PyMySQL-1.0.2-py3-none-any.whl (43kB)
    100% |██████████| 51kB 1.8MB/s
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

### I Sử dụng Adafruit để sử dụng cảm biến nhiệt độ và độ ẩm

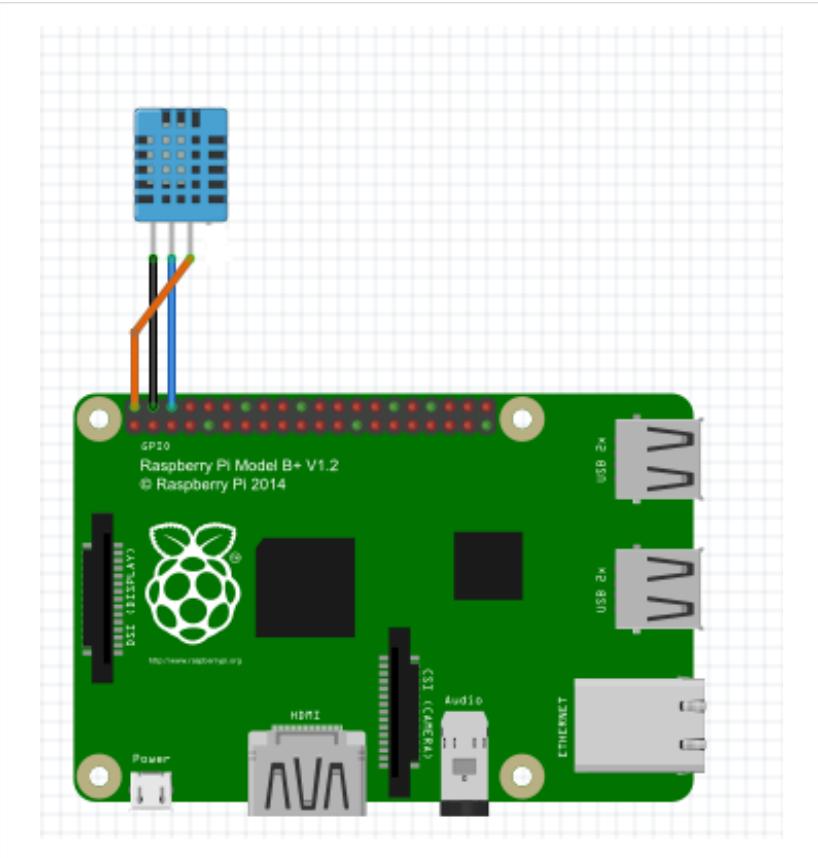
- ▶ `git clone https://github.com/adafruit/Adafruit_Python_DHT.git`
  - Sao chép git để sử dụng dữ liệu về cảm biến nhiệt độ và độ ẩm

```
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
Cloning into 'Adafruit_Python_DHT'...
remote: Enumerating objects: 325, done.
remote: Total 325 (delta 0), reused 0 (delta 0), pack-reused 325
Receiving objects: 100% (325/325), 98.35 KiB | 2.23 MiB/s, done.
Resolving deltas: 100% (176/176), done.
```

- ▶ `cd Adafruit_Python_DHT`
- ▶ Cài đặt sudo python setup.py
  - Chạy setup.py để hoàn tất cài đặt

```
pi@raspberrypi:~ $ cd Adafruit_Python_DHT/
pi@raspberrypi:~/Adafruit_Python_DHT $ sudo python setup.py install
running install
running bdist_egg
running egg_info
creating Adafruit_DHT.egg-info
writing Adafruit_DHT.egg-info/PKG-INFO
writing top-level names to Adafruit_DHT.egg-info/top_level.txt
writing dependency_links to Adafruit_DHT.egg-info/dependency_links.txt
writing manifest file Adafruit_DHT.egg-info/SOURCES.txt
```

### | Cấu hình sơ đồ mạch thực hành đo nhiệt độ, độ ẩm



- ▶ Cảm biến nhiệt độ và độ ẩm DOUT: Raspberry Pi GPIO2
- ▶ Cảm biến nhiệt độ và độ ẩm GND: Raspberry Pi GND
- ▶ Cảm biến nhiệt độ và độ ẩm VCC: Raspberry Pi 3.3V

### | Thực thi AdafruitDHT.py

- ▶ Thực hiện lệnh python AdafruitDHT.py 11 2
  - 11 nghĩa là loại cảm biến và 2 nghĩa là GPIO2.
- ▶ Nhiệt độ và độ ẩm được đo chính xác trong khoảng thời gian 5 giây trong khi thực hiện.
- ▶ Nếu muốn thoát nhấn Ctrl + c để tạo ngắt.

```
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ python AdafruitDHT.py 11 2
Temp=27.0*  Humidity=42.0%
Temp=27.0*  Humidity=56.0%
^CTraceback (most recent call last):
  File "AdafruitDHT.py", line 55, in <module>
    time.sleep(5)
KeyboardInterrupt
pi@raspberrypi:~/Adafruit_Python_DHT/examples $
```

## Lưu dữ liệu đo được bởi cảm biến nhiệt độ và độ ẩm trong DB

### | Thay đổi AdafruitDHT.py

- Thực hiện chức năng lưu mã ví dụ đã xác nhận rằng nhiệt độ và độ ẩm được đo tốt trong bảng nhiệt độ bằng cách sử dụng câu lệnh INSERT.

```
1  import sys
2  import time
3  import Adafruit_DHT
4  import pymysql
5
6  sensor = Adafruit_DHT.DHT11
7  pin = 2
8
9  db, cur = None, None
10
11 db = pymysql.connect(host='192.168.0.4', user='root', password='1234', db='mysql', charset='utf8')
12
13 try:
14     cur = db.cursor()
15     while True:
16         humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
17         data = (humidity, temperature)
18         if humidity is not None and temperature is not None:
19             sql = "INSERT INTO temperature(temp, humid) VALUES (%4.1f, %4.1f)" % data
20             print(sql)
21             cur.execute(sql)
22             db.commit()
23         else:
24             print('Failed to get reading. Try again!')
25             time.sleep(5)
26
27 except KeyboardInterrupt:
28     pass
29 finally:
30     db.close()
```

- dòng 6: Cố định với cảm biến DHT11
- dòng 9: Khai báo biến toàn cục
- dòng 11: máy chủ: địa chỉ ip của raspberry pi, người dùng: root, mật khẩu: đặt mật khẩu, loại db là mysql
- dòng 14: Tạo con trỏ DB
- dòng 16-17: Lưu nhiệt độ và độ ẩm trong dữ liệu
- dòng 19-22: Thực thi và cam kết (lưu) câu lệnh INSERT đặt nó phù hợp với các đặc tính nhiệt độ và độ ẩm của nhiệt độ

## 5.3 Ứng dụng của MariaDB

## Bài 05

### | Thực thi AdafruitDHT.py

The terminal window shows the command `python3 AdafruitDHT.py` being run, which inserts two rows of data into the `temperature` table:

```
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ python3 AdafruitDHT.py
INSERT INTO temperature(temp, humid) VALUES (61.0, 26.0)
INSERT INTO temperature(temp, humid) VALUES (60.0, 26.0)
```

The HeidiSQL interface shows the `temperature` table with the following data:

	temp	humid
2021-09-22 06:23:48	60	26
2021-09-22 06:23:54	60	26
2021-09-22 06:23:59	60	26
2021-09-22 06:29:38	60	26
2021-09-22 06:29:43	59	27
2021-09-22 06:29:49	61	26
2021-09-22 06:32:14	61	26
2021-09-22 06:32:20	60	26
2021-09-22 06:32:25	59	27

The bottom of the HeidiSQL window shows the SQL history:

```
43 SHOW COLLATION;
44 SHOW CREATE TABLE `mysql`.`temperature`;
45 SHOW TABLES FROM `information_schema`;
46 SELECT * FROM temperature;
47 /* Affected rows: 0  Found rows: 10  Warnings: 0  Duration for 1 query: 0.016 sec. */
```

- Sử dụng lệnh `python3` để thực thi tệp ví dụ/Adaf nhungDHT.py. Bạn có thể kiểm tra các giá trị đo được như minh họa ở bên trái và bạn có thể kiểm tra câu lệnh SQL đã thực thi.
- Quay trở lại heidiDB và vào bảng nhiệt độ để kiểm tra xem bảng nhiệt độ đã được lưu tốt chưa.
- Như trước đây, câu lệnh Truy vấn là `SELECT * FROM nhiệt độ`; Bạn ghi và thực hiện như hình bên trái, bạn có thể kiểm tra dữ liệu mới lưu thành công như hình bên dưới.

## Cung cấp dữ liệu được lưu trữ trong DB tới trang web

### | Phục vụ Trực quan hóa với Trang Web bằng cách lưu trữ dữ liệu trong DB

- ▶ Viết tập lệnh python để chuyển dữ liệu được lưu trữ trong MariaDB sang html

```
1  from flask import Flask, request, render_template
2  import pymysql
3
4  db = None
5  cur = None
6  app = Flask(__name__)
7
8  def select(query):
9      db = pymysql.connect(host='192.168.0.3', user='root', password='1234', db='mysql', charset='utf8')
10     cur = db.cursor()
11     cur.execute(query)
12     result = cur.fetchall()
13     db.close()
14     return result
15
16 @app.route('/temp_humid_chart')
17 def lm35_chart():
18     sql = "SELECT DATETIME, TEMP FROM temperature ORDER BY DATETIME ASC LIMIT 100"
19     result = select(sql)
20     return render_template("temp_humid_chart.html", result=result)
21
22 if __name__ == '__main__':
23     app.run(debug=True, port=80, host='0.0.0.0')
```

- dòng 8-14: Chọn hàm trả về kết quả thực hiện câu lệnh truy vấn bằng cách kết nối với MariaDB và tạo con trỏ
- dòng 16-20: Một hàm lấy 100 nhiệt độ và thời gian được sắp xếp theo thứ tự thời gian và đưa kết quả vào tệp temp\_humid\_chart.html

- ▶ Save as ~/rasp\_ex/webapp/temp\_humid\_chart.py

| ~/rasp\_ex/webapp/temp\_humid\_chart.html (1)

- ▶ tệp html cung cấp trực quan hóa bằng cách sử dụng kết quả được chuyển bởi temp\_humid\_chart.py

```
1  <html>
2  <head>
3      <title>RaspberryPi Visualization</title>
4      <style type="text/css">
5          .auto-style1 {
6              border: 1px solid #808080;
7          }
8          .auto-style2 {
9              border: 1px solid #008080;
10         }
11     </style>
12     </head>
13     <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.4.0/Chart.min.js"></script>
14     <body>
15         <center>
16             <br>
17             <strong>Temperature, Humidity Data Visualization</strong>
18             <br>
19             <br>
20             <br>
```

- dòng 3: Tiêu đề của trang web
- dòng 4-11: Cài đặt kiểu
- dòng 13: Bao gồm chart.js bằng CDN
- dòng 17: Cụm văn bản cần bôi đen trên trang web

| ~/rasp\_ex/webapp/temp\_humid\_chart.html (2)

- ▶ tệp html cung cấp trực quan hóa bằng cách sử dụng kết quả được chuyển bởi temp\_humid\_chart.py

```
21      <table style="width: 100%">
22        <tr>
23          <td width="50%">
24            <table cellspacing="1" class="auto-style1" style="width: 100%">
25              {% for i in result%}
26                <tr>
27                  <td class="auto-style2" style="width: 50%">Time: {{ i[0] }}</td>
28                  <td class="auto-style2" style="width: 50%">Temperature: {{ i[1] }}</td>
29                </tr>
30              {% endfor %}
31            </table>
32          </td>
33
34          <td width="50%">
35            <div>
36              <canvas id="Chart"></canvas>
37            </div>
38
39          </td>
40        </tr>
41      </table>
42      <br>
43    </center>
44  </body>
45
46</html>
```

- dòng 26-31 : Xuất thời gian và nhiệt độ tuần tự theo kích thước được chỉ định trong kiểu bằng cách sử dụng biến kết quả được gửi đến trang web

| ~/rasp\_ex/webapp/temp\_humid\_chart.html (3)

- ▶ Tập lệnh vẽ biểu đồ bằng chart.js

```
50 <script>
51     var ctx = document.getElementById('Chart').getContext('2d');
52     var data = {
53         // The type of chart we want to create
54         type: 'line',
55         // The data for our dataset
56         data: {
57             labels: [ // Date
58                 {% for i in result%}
59                 '{{ i[0] }}',
60                 {% endfor %}
61             ],
62             datasets: [{{
63                 label: "Temperature",
64                 backgroundColor: 'rgb(255, 120, 132)',
65                 fill:false,
66                 borderColor: 'rgb(255, 128, 132)',
67                 lineTension : 0.8,
68                 data: [ // Temperature
69                     {% for i in result%}
70                     {{ i[1] }},
71                     {% endfor %}
72                 ],
73             }]
74         },
75     },
```

- dòng 54: Biểu đồ đường
- dòng 56-59: Định nghĩa nhãn dữ liệu. Nó có nghĩa là ngày là đối số đầu tiên của kết quả nhận được.
- dòng 62-71: Xuất các giá trị Nhiệt độ ở định dạng đã chỉ định. Nếu giá trị lineTension tăng lên, đầu ra sẽ cong hơn.

| ~/rasp\_ex/webapp/temp\_humid\_chart.html (4)

- ▶ Tập lệnh vẽ biểu đồ bằng chart.js

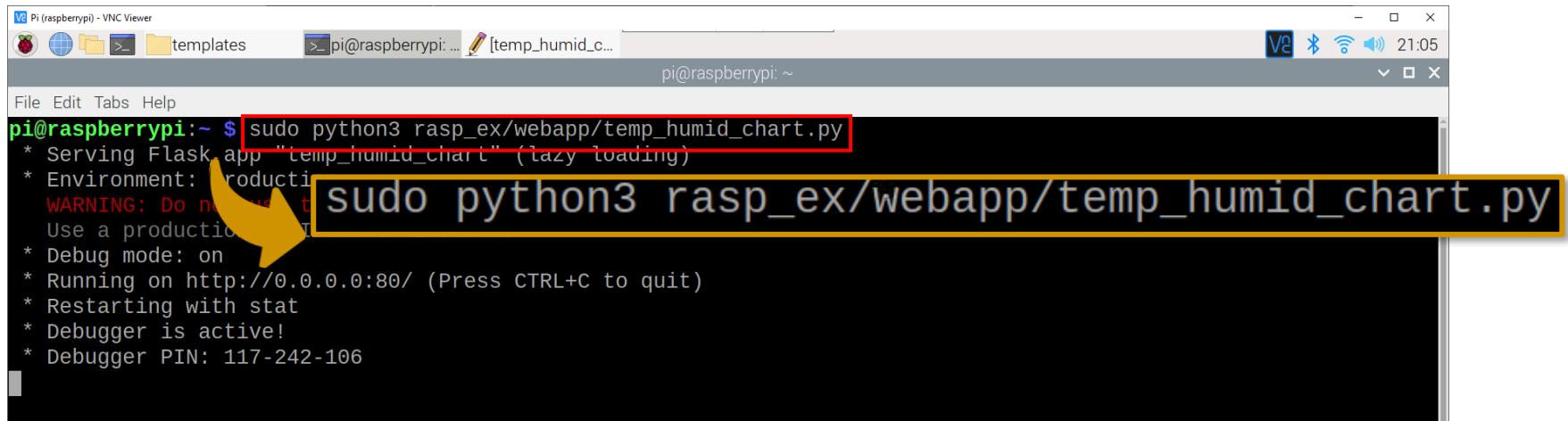
```
75     // Configuration options go here
76     options: {
77         scales: [
78             xAxes: [{  
79                 ticks: {  
80                     suggestedMin: 0,  
81                     suggestedMax: 100  
82                 }  
83             }],  
84             yAxes: [{  
85                 ticks: {  
86                     suggestedMin: 10, // Y axis min  
87                     suggestedMax: 50 // Y axis max  
88                 }  
89             }]  
90         }  
91     }  
92 }  
93 var chart = new Chart(ctx, data);  
94 </script>
```

- dòng 79-81: Đặt phạm vi cho trục x
- dòng 84-89: Đặt phạm vi cho trục y
- dòng 93: Tạo biểu đồ và lưu thành đối tượng trong biểu đồ

- ▶ Save the html file as ~/rasp\_ex/webapp/temp\_humid\_chart.html

### | Thực thi tập lệnh python đã tạo (1)

- ▶ Since you are using Flask, you need sudo authenticate.
  - Thực thi lệnh `sudo python3 rasp_ex/webapp/temp_humid_chart.py` để thực thi tệp lệnh.

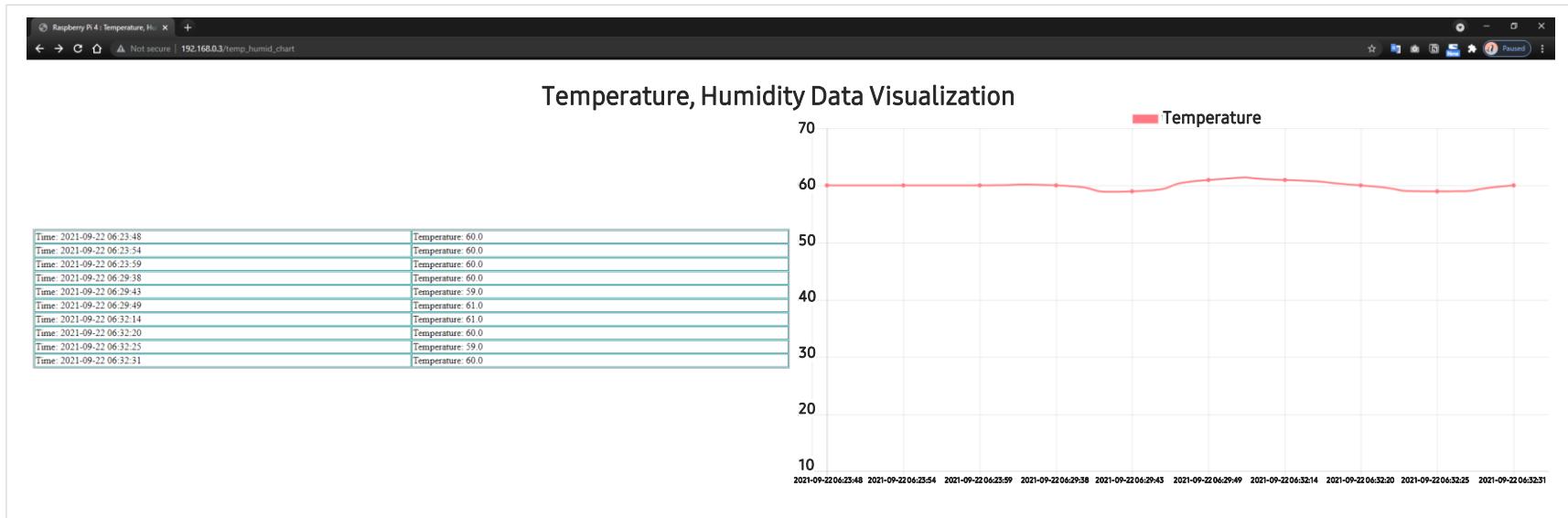


The screenshot shows a terminal window titled "Pi (raspberrypi) - VNC Viewer". The window has a toolbar with icons for file operations and a status bar showing "21:05". The terminal prompt is "pi@raspberrypi:~". The user has run the command `sudo python3 rasp_ex/webapp/temp_humid_chart.py`. The output shows the application is serving a Flask app named "temp\_humid\_chart" in production mode. A yellow arrow points from the bottom of the question text to the command in the terminal window.

```
pi@raspberrypi:~ $ sudo python3 rasp_ex/webapp/temp_humid_chart.py
* Serving Flask app "temp_humid_chart" (lazy loading)
* Environment: production
  WARNING: Do not use a production secret key if you're sharing this machine
  Use a production secret key if you're deploying to a public endpoint
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-242-106
```

### I Executing the created python script (2)

- ▶ Nhập địa chỉ IP Raspberry Pi của bạn/temp\_humid\_chart trong trình duyệt web
  - Trong trang web, bạn có thể kiểm tra dữ liệu thời gian và nhiệt độ trong bảng DB và bạn cũng có thể kiểm tra trực quan hóa dữ liệu thông qua biểu đồ.



# Kết thúc Tài liệu



# Together for Tomorrow! Enabling People

Education for Future Generations

©2022 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of book.

This book is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this book other than the curriculum of Samsung Innovation Campus or to use the entire or part of this book, you must receive written consent from copyright holder.