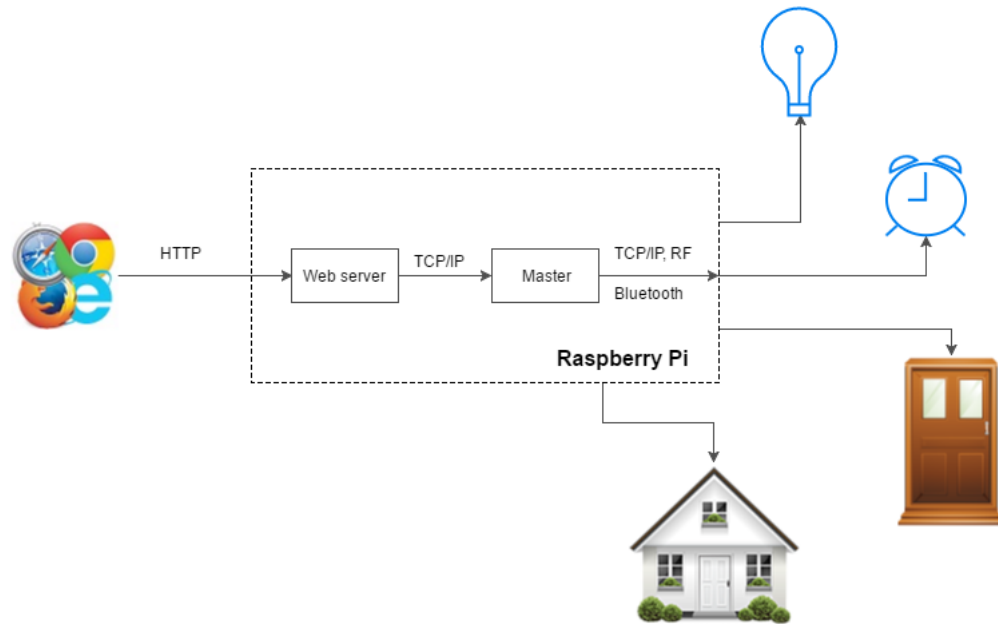


## Bài 11 : Lập trình web-server trên Raspberry pi – phần 2

Trong phần trước của tutorial về lập trình web-server mình đã chia sẻ với các bạn những nền tảng kiến thức lập trình và công cụ các bạn cần cài đặt nếu muốn tự tạo cho mình một web-server trên Raspberry. Phần tiếp theo này mình sẽ tiếp tục chia sẻ một "chương trình" web-server nhỏ do mình tự viết để các bạn có thể nhìn tổng quan hơn về hệ thống. Ngoài ra, chương trình của mình làm nền để xây dựng dự án của riêng mọi người.

Để xây dựng được một web-server thực sự tốt có lẽ sẽ là cả một dãy núi kiến thức các bạn cần phải vượt qua. Kiến thức về lập trình web như HTML, CSS, Javascript, PHP, cơ sở dữ liệu sql. Thật sự rất khó khăn và tốn kém thời gian khi muốn thực hiện nếu bạn không phải là dân công nghệ thông tin. Tuy nhiên, mong muốn ứng dụng của các bạn không phức tạp hoặc bạn có thể dựa trên những chương trình có sẵn thì khối lượng công việc sẽ được giảm xuống nhiều. Những người làm ứng dụng thường nói câu : "khó khăn của người khác đã làm, hãy sử dụng nó để gây dựng chiếc xe cho riêng mình.". Câu này rất phù hợp với người đã có nền tảng căn bản hoặc không chuyên và có lẽ đang mong muốn làm nhanh ứng dụng này.

Dưới đây là mô hình web-controller mình chia sẻ cho mọi người. Có những phần của mô hình đã hoàn thiện, có những phần mọi người phải tự thực hiện tùy theo yêu cầu.



Hình 1 : mô hình web-controller

Kết nối giữa trình duyệt và Web-server.

Web-server sau khi cài đặt sẽ được người dùng truy cập thông qua trình duyệt trên máy tính hoặc điện thoại. Người dùng mở trình duyệt và nhập địa chỉ "địa chỉ của Pi/thư mục chứa web-server là lazy-man". Chúng ta không cần chỉ đích xác chương trình web-server (ví dụ Apache sẽ tự động thay ta tìm tới file chương trình và thực thi file đó).

Ngay khi có tín hiệu yêu cầu từ trình duyệt gửi tới server. Server ngay lập tức đọc dữ liệu từ cơ sở dữ liệu sau đó trả dữ liệu đó dưới dạng HTML tới trình duyệt. Trình duyệt hiển thị lên màn hình. (Những thứ được hiển thị trên trình duyệt đều là HTML).

Kết nối giữa Web-server và Master.

Khi người dùng thực hiện một thao tác điều khiển nào đó trên trình duyệt ví như bật tắt một bóng đèn thì thông tin đó lại được gửi lên server. Server sẽ chuyển tiếp thông tin từ phương thức TCP/IP. Web-server lúc này sẽ là client của Master. Mọi người có thể xem thêm thông tin về phương thức này trong tutorial trước "mô hình Client-Server". Sau Server sẽ cập nhật thông tin vào cơ sở dữ liệu. Thông tin điều khiển cũng được hiển thị ngay trên trình duyệt.

Việc sử dụng cơ sở dữ liệu là điều cần thiết vì các thông tin trên trình duyệt chỉ là tạm thời, nếu không lưu trữ thì khi tắt bật lại trình duyệt thì thông tin đã không chính xác rồi được lưu cứng trong cơ sở dữ liệu sẽ phòng tránh các trường hợp sự cố phần cứng như mất nguồn.

Kết nối giữa Master và các thiết bị.

Phần này mình không thực hiện. Mọi người có thể tự do lựa chọn phương thức kết nối không nhất thiết phải là TCP/IP mà có thể là RF, Zigbee hay bluetooth. Vì Master đang quen thuộc với lập trình viên nên mọi người có thể trực tiếp vào chương trình Master. Có một đặc điểm chú ý là Master đang là server đối với client là web-server. Địa chỉ trong Master luôn là "127.0.0.1" vì cùng host (trên cùng một thiết bị). Đây là đặc điểm dùng làm nhận định thông tin điều khiển từ người dùng. Master sẽ dùng thông tin này để điều khiển.

Mọi người nên làm thông tin phản hồi từ thiết bị. Ví như nếu bật đèn thành công, thiết bị sẽ gửi lại thông tin thông báo cho Master là đã thực hiện thành công, ngược lại sẽ thông báo lỗi.

### 1. Cài đặt web-server

Chương trình web-server mình chia sẻ miễn phí với mọi người, mọi người có thể tự do chỉnh sửa, sử dụng.

Bước 1 - Tải chương trình web-server :

Mọi người truy cập vào [github](#) và tải chương trình về. Thư mục chứa chương trình là web-server nhé. Bỏ thư mục "lazy" vào thư mục chỉ "var/www/html".

Chú ý và thư mục "www" thuộc quyền của root nên các bạn phải sử dụng quyền root để copy hoặc chuyển thư mục này sang quyền user thông thường :

```
cd '/var'
sudo chown -R pi:pi www
```

Trong đó owner:goup là pi: pi. Có thể raspberry của bạn đang dùng owner và group khác. Hãy kiểm tra nhóm của mình bằng lệnh : groups.

Bước 2 - Cài đặt cơ sở dữ liệu :

Mở thư mục sql/sql-function.php lên và sửa 2 dòng lệnh sau :

- define("userName", "root") thành define("userName", "tên khi bạn cài đặt mysql, mặc định là root").
- define("password", "password") thành define("password", "mật khẩu khi cài đặt mysql ứng với tên bên trên").

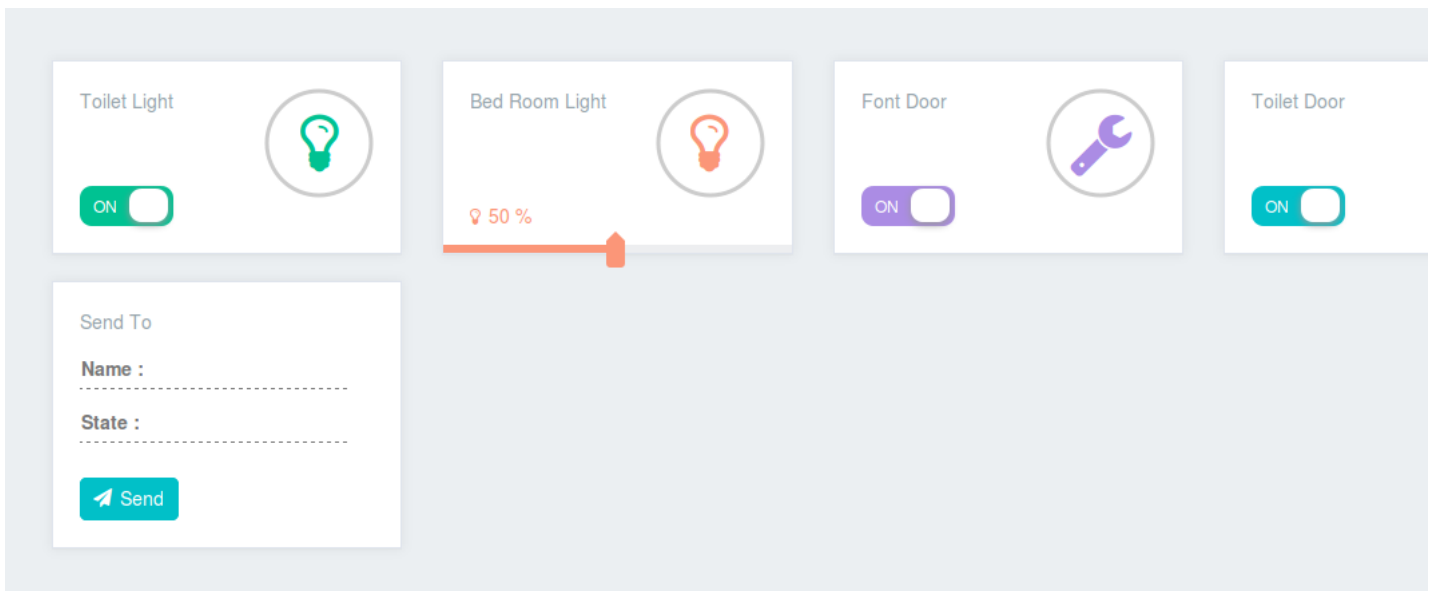
Trở con trở tới thư mục sql và thực hiện tạo cơ sở dữ liệu bằng lệnh :

```
php sql.php
```

Ngay sau lệnh này các bạn hãy **restart lại raspberry**.

Chương trình sẽ tạo ra những ví dụ về những thiết bị điều khiển đơn giản mà chương trình hỗ trợ. Hãy thử điều khiển với những thiết bị đó trước khi tạo thiết bị của riêng mình trong việc xem các project trong thư mục html, bạn có thể xóa tệp "index.html" đi. Khi nhập vào "địa chỉ của pi" trên thanh nhập địa chỉ sẽ xuất hiện ra thông tin về project của bạn.

Mở trình duyệt lên và xem gõ "địa chỉ Pi/lazy". Nếu trình duyệt hiển thị các thông tin về thiết bị thì các bạn đã cài đặt thành công. Ngoài ra các bạn cũng có thể truy cập trực tiếp Database được tạo là "home" và table là "device".



Hình 2 : kết quả sau khi cài đặt database

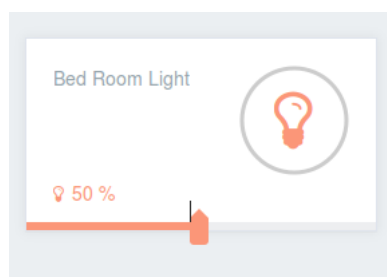
Bước 3 – Tạo lập Master :

Truy cập vào thư mục master và gõ lệnh sau :

```
gcc -std=c11 -Wall -g -D_BSD_SOURCE server-thread.c -o server-thread -lpthread
./server-thread
```

Bạn hãy thử điều khiển một vài thiết bị trên trình duyệt. Mọi thông tin điều khiển sẽ được hiển thị trên terminal.

Các thiết bị được hỗ trợ là các thiết bị bật tắt đơn giản. Có thể điều chỉnh thông tin cường độ.



Ngoài ra còn có thêm một bảng gửi thông tin riêng biệt. Với bảng thông tin này mọi người có thể gửi mọi loại thông tin mình mong muốn.

Send To

Name : \_\_\_\_\_

State : \_\_\_\_\_

Send

Nếu các bạn muốn làm các thiết bị khác mà không được hỗ trợ có thể dựa vào chương trình có sẵn để tạo thêm.

## 2. Các hàm chức năng

Các bạn có thể tạo thêm thiết bị để điều khiển hoặc xóa các thiết bị các bạn không mong muốn trong cơ sở dữ liệu. Dưới đây là các hàm chức năng trong cơ sở dữ liệu mà dùng.

1) Tạo một thiết bị điều khiển mới :

```
InsertObject($conn,object-type,object-name,initial-state,color,amplitude,icon)
```

Trong đó :

- \$conn : là biến liên kết tới mysql
- object-type (string) : loại thiết bị, loại thiết bị khác nhau có cách điều khiển khác nhau. Hỗ trợ 2 loại là "obj-slider" và "obj-button".
- object-name (string) : tên thiết bị điều khiển.
- initial-state : trạng thái thiết lập. 1 tương ứng với bật và 0 tương ứng với tắt.
- color (string) : màu sắc cho giao diện điều khiển. Có 4 màu là blue (mặc định : NULL), green (flavor-green), orange (flavor-orange), violet (flavor-violet).
- amplitude (number) : Mức hoạt động của thiết bị. Ví dụ như đèn sẽ sáng bao nhiêu % so với định mức.
- icon (string) : icon đại diện cho thiết bị. Mặc định là "fa-wrench". Xem tên các icon ở đây [Font Awesome](#).

1) Xóa một thiết bị điều khiển :

```
DeleteObject($conn, object-name)
```

Trong đó :

- object-name (string) : tên thiết bị muốn xóa

2) Update thông tin cho thiết bị

```
UpdateObject($conn, object-name, "name='value'")
```

Trong đó :

- object-name (string) : tên thiết bị muốn update
- name (string) : tên thuộc tính muốn thay đổi
- value : giá trị muốn thay đổi.

3) Thực thi file "sql.php"

```
php sql.php
```

Các bạn có thể xem hướng dẫn chi tiết hơn với video dưới đây. (Rất tiếc mình không có micro tốt - nghe rất nhiều nên không thể nói trong video. Mọi người xem tạm nhé. )

Hướng dẫn tạo lập một web-controler đơn giản trên Raspberry Pi



3. Những hạn chế cần cải tiến

Vì mục đích chỉ để làm quen với nền tảng căn bản và mình cũng không phải dân IT nên chương trình cũng rất sơ sài và có thể sẽ bị mắc lỗi. Mọi thông tin đóng góp và hỗ trợ đều đón nhận nhiệt tình.

Hạn chế lớn nhất của chương trình trên là các thiết bị không được hỗ trợ đa dạng. Chỉ hỗ trợ loại thiết bị cơ bản là bật tắt. Chức năng cũng bị hạn chế theo. Do cơ sở dữ liệu chung một bảng nên tính năng giống nhau – cũng là đặc điểm gây hạn chế. Để khắc phục có thể tạo ra các bảng riêng biệt. Tuy nhiên mọi người phải tự code phần này. Nếu bị thì mọi người cũng phải tự lập trình html/css để hiển thị trên trình duyệt.

Hạn chế trên là lý do có bảng gửi thông tin đặc biệt để gửi bất kỳ thông tin mong muốn thay cho việc tạo dựng mới thiết bị.

Bài tiếp theo :

Bài 3 : Bản demo đầu cuối cho web-server

Xem list bài lập trình của Raspberry Pi .

Ngo Van Viet |  11/19/2018

chắc do bản PHP của tác giả đã cũ.. mình đang dùng PHP 7 nên đến bước " php sql.php " thì báo lỗi HP Fatal error: Uncaught Error: Call to undefined function mysqli\_connect() in /var/www/html/lazy/sql/sql-function.php:10 Stack trace: #0 /var/www/html/lazy/sql/sql.php(7): ConnectToSql() #1 {main} thrown in /var/www/html/lazy/sql/sql-function.php on line 10 vậy giờ mình phải làm sao.. chỉnh lại như thế nào cho phù hợp với bản PHP mới khi mình không chuyên về code

hung |  10/31/2018

lỗi này thì phải làm sao khi gõ lệnh php sql.php PHP Fatal error: Uncaught Error: Call to undefined function mysqli\_connect() in /var/www/html/Raspberry-Pi/Tutorial/web-server/lazy/sql/sql-function.php:10 Stack trace: #0 /var/www/html/Raspberry-Pi/Tutorial/web-server/lazy/sql/sql.php(6): ConnectToSql() #1 {main} thrown in /var/www/html/Raspberry-Pi/Tutorial/web-server/lazy/sql/sql-function.php on line 10

ngọc hoàng |  06/11/2018

anh ơi cho em hỏi ở trong file query.js cái thuộc tính \$(".switch-button:not('.type-turn')") cái .type-turn đc giải thích sao vậy anh.

Phan Hữu Đan |  09/22/2017

Chào admin, tại bước cài đặt chương trình, tại sao đổi mật khẩu của mysql thành "nightsky" rồi mà nó vẫn báo lỗi ạ? Nhờ admin xem lại giúp với ạ!

Tiến Lành |  07/08/2017

Bài viết rất hay. Ad cho mình hỏi mình làm như vậy đến khi thực thi file php sql.php thì không được lỗi sao vậy ! :'(

Viết đánh giá

Họ và tên:

Đánh giá của bạn:

Lưu ý: Không hỗ trợ HTML!

Bình chọn:  Dở Hay

Nhập mã bảo vệ:

9cfe26

Tiếp tục