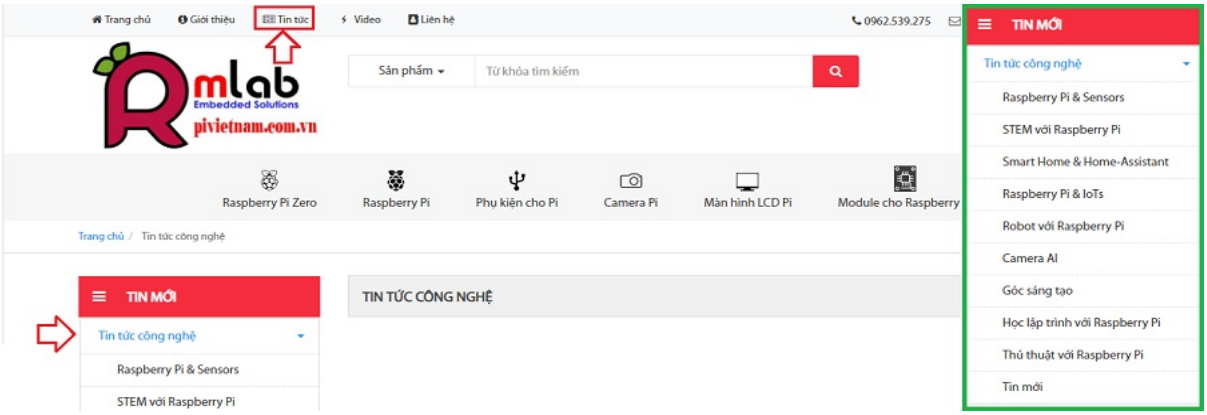
Bài 12 : Lập trình web-server trên Raspberry pi – phần 3

CHÚ Ý : ***Từ 2019 MLAB có thêm một website cho riêng Raspberry Pi và trở thành website chính về Raspberry Pi tại MLAB, các thông tin về sản phẩm - tin tức cập nhật về Raspberry Pi - Bài viết kỹ thuật hỗ trợ cho Raspberry Pi, ... MLAB cập nhật tại website :*** *pivietnam.com.vn*

MLAB trân trọng thông báo tới quý khách hàng!!!

Các bạn có thể tham khảo các bài viết hỗ trợ kỹ thuật và các tin tức mới nhất tại phần "tin tức"trên website PVIETNAM.COM.VN



Bài viết hỗ trợ kỹ thuật tại website PIVIETNAM.COM.VN - Bài 12: Lập trình web-server trên Raspberry Pi - Phần 3 (Link here)

1. Giới thiệu

Trong phần 3 này mình sẽ làm bản demo cuối cùng cho loạt bài web-server. Mình sẽ sử dụng trình duyệt trên máy tính/ điện thoại truy cập vào server trên Raspberry của mình để điều khiển bật tắt bóng đèn. Tuy demo đơn giản nhưng qua đó các bạn sẽ nhìn được hệ thống một cách tiệm cận trực quan hơn.

2. Chuẩn bị

Chắc chắn rằng các bạn đã đọc và thực hiện 2 bài trước : phần 1 và phần 2.

Phần cứng gồm có:

- Raspberry Pi
- ESP8266 Evaluation Board
- Đèn 12V
- Nguồn 12V

Kết nối :

ESP8266 evaluation board gồm có một module wifi esp8266 version 7 và được tích hợp thêm Relay 5V. Chân điều khiển relay kết nối với chân số 5 của esp. Relay sẽ được dùng để điều khiển bóng đèn. Các bạn cũng có thể lấy ngay bóng đèn điện phòng của mình để điều khiển.

Raspberry Pi phải được kết nối mạng. Máy tính/ điện thoại cùng mạng LAN với Pi.

Thông tin thêm :

ESP8266 v7 được tích hợp vi xử lý bên trong có thể lập trình được. Có thể dùng nó như một vi điều khiển bình thường ngoại trừ hạn chế về mặt bộ nhớ hay các cổng ngoại vi. Thư viện lập trình cho ESP8266 được viết trên nền tảng arduino – dùng phần mềm arduino có thể lập trình được. Các bạn có thể xem hướng dẫn cài đặt và sử dụng thư viện ở đây.

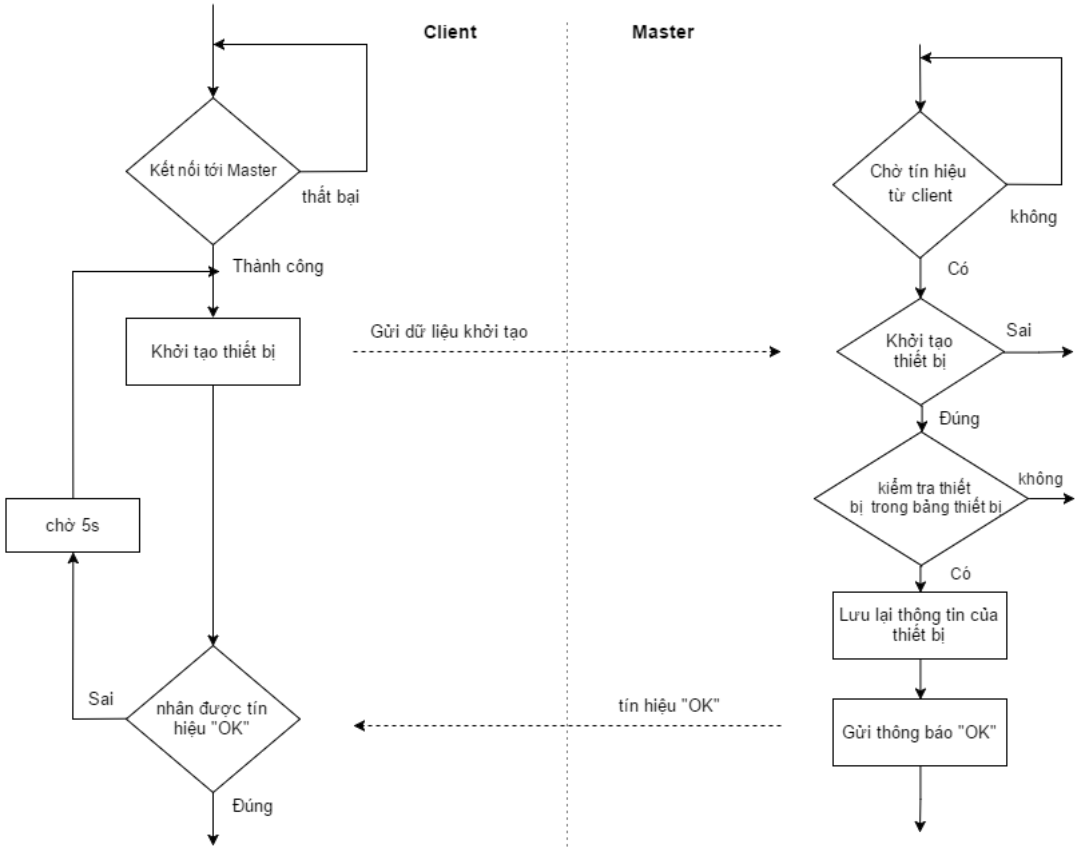
3. Thực hiện

Để thực hiện một cơ chế quản lý tốt các thiết bị cần phải giải quyết nhiều vấn đề khác nhau. Điều đầu tiên phải quan tâm tới là phân biệt các thiết bị để điều khiển. Vì các thiết bị và web-server đều kết nối tới trung tâm qua phương thức TCP-IP nên cũng cần phải phân biệt được web-server. Cách bóc tách dữ liệu từ server và gửi dữ liệu đó tới thiết bị.

a) Khởi tạo thiết bị

Mỗi thiết bị sẽ có một mã ID riêng; khi thiết bị kết nối tới trung tâm (Master) nó sẽ gửi thông tin khởi tạo thiết bị gồm có mã ID để Master xác nhận. Chú ý rằng Master sẽ biết trước rằng có những thiết bị nào. Nếu thiết bị nằm trong danh sách quản lý, nó sẽ chấp nhận việc khởi tạo thiết bị và lưu lại thông tin. Chính thông tin này được dùng làm để phân biệt thiết bị.

Hãy cùng xem thuật toán khởi tạo thiết bị dưới đây. Gồm các 2 phía thiết bị (client) và trung tâm (Master)



Hình 1: khởi tạo thiết bị

Thuật toán bên Master :

B1. Chờ tín hiệu kết nối từ client.

B2. Kiểm tra thông tin đó có phải thông tin khởi tạo hay không.

B3. Nếu đúng là thông tin khởi tạo thì kiểm tra xem trong danh sách thiết bị có thiết bị đó không.

B4. Nếu có thì lưu lại thông tin thiết bị (có thể là địa chỉ IP). Thực chất trong chương trình này lưu lại file-descriptor tương ứng với thiết bị.

B5. Gửi thông báo phản hồi đã khởi tạo thành công.

Thuật toán bên Client :

B1. Kết nối tới Master.

B2. Gửi thông tin khởi tạo. Thông tin khởi tạo có thể là chuỗi ký tự trong đó có chuỗi ký tự đặc biệt thể hiện đó là khởi tạo. Ví dụ dữ liệu khởi tạo : “INIT:L2”. Trong đó “INIT” thông báo cho Master biết đây là thông tin khởi tạo và mã thiết bị là “L2”.

B3. Chờ tín hiệu phản hồi từ Master. Nếu không có tín hiệu trả về là “OK” thì sẽ thực hiện việc gửi lại thông tin khởi tạo (sau một khoảng thời gian : 5s).

Chương trình bên Master :

Danh sách quản lý thiết bị là một mảng bao gồm 3 thông tin là tên thiết bị, mã ID thiết bị, và file-desciptor tương ứng của thiết bị.

```
char *IdDevice[4][3] = { {"Living Room Light", "L1", NULL},
                         {"Toilet Light", "L2", NULL},
                         {"Bed Room Light", "L3", NULL},
```

```
                {"Font Door", "D1", NULL},
};
```

Chú ý tên thiết bị như là "Living Room Light" phải giống hệt như trong cơ sở dữ liệu. Vì tên này sẽ nằm trong dữ liệu từ web-server gửi sang, nó dùng làm tham chiếu tới thiết bị. Mã ID của thiết bị là “L1”, “L2” v.vv . Và cuối cùng file-descriptor mặc định là NULL. “file-descriptor” thực chất là số kiểu int. Khi sử dụng nó ta phải chuyển lại kiểu int để sư dụng còn lưu trữ ta phải đổi sang kiểu xâu ký tự.

Trong bảng có thể sự trùng lặp giữa tên thiết bị nhưng mã thiết bị và file-descriptor phải khác nhau. Lý do là vì có thể trong “Living Room Light” có thể có nhiều thiết bị đèn khác nhau. Vì thế ta có thể điều khiển chung một nhóm thiết bị.

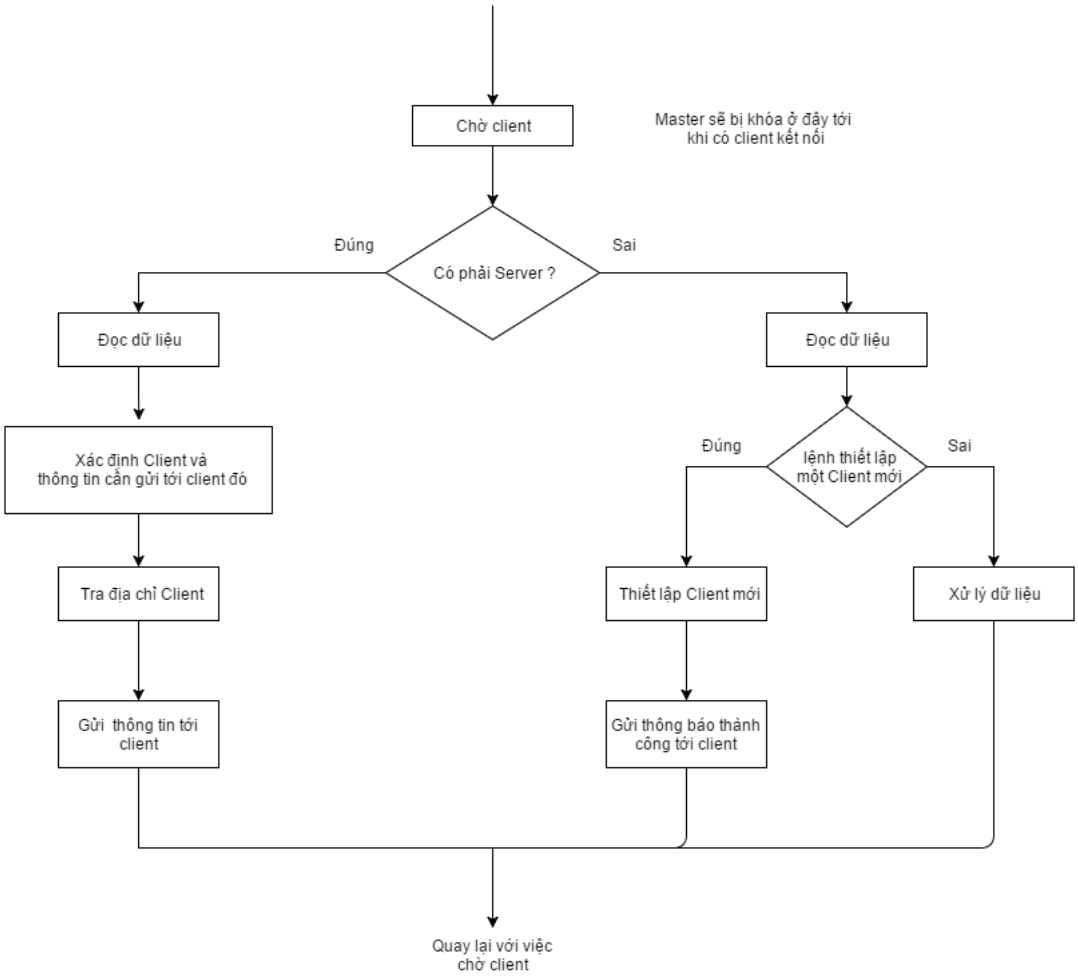
```
bool IdentifyDevice(int clntSock, char *str) {

    for (int i = 0; i < SIZE_OF_ARRY2(IdDevice); ++i) {
        if(IdDevice[i][0]) // check if IdDevice[i][0] is not NULL
            if(strcmp(str, IdDevice[i][1])==0) {
                printf("+ Detecting a new device: %s, ID:%s\n\n", IdDevice[i][0], IdDevice[i][1]);
                if ( send(clntSock, SET,strlen(SET),0) < 0) { // send "OK"
                    printf("- Sending \"OK\" to device: \"%s\" false\n", IdDevice[i][0]);
                    return false;
                }
                char clientAddr[4]; // save client socket as string, max is "9999"
                sprintf(clientAddr, "%d", clntSock);    // convert to string
                IdDevice[i][2] = strdup(clientAddr);    // save to IdDevice
                return true;
            }
    }
    return false;
}
```

Hàm khởi tạo thiết bị sẽ so sánh mã ID (biến str) từ thiết bị gửi tới và mã ID trong bảng IdDevice. Nếu có trong bảng thì sẽ gửi thông báo “OK” và chuyển đổi giá trị clntSock (chính là file-descriptor) sang kiểu ký tự và lưu lại vào bảng IdDevice.

b) Đọc dữ liệu từ Web-server

Web-server trong mạng cũng là một client đối với Master. Vì nó cũng được chay song song với Master trong một máy tính nên client này luôn có địa chỉ cố định là “127.0.0.1” (localhost). Chúng ta sẽ dựa vào đặc điểm này để nhận dạng web-server

Thuật toán hoàn thiện của Master sẽ như sau :



Hình 2: thuật toán chương trình Master

Chương trình cho Master :

```
// kiem tra xem co phai la web-server hay khong. Nếu có gọi hàm
// ServerCommand()
if( !strcmp(((struct ThreadArgs *) threadArgs) -> addr, localhost) ) {
    printf(". Web server: %s\n", buffer);
    ServerCommand(buffer);
}


void ServerCommand(char *str) {

    char *header = strtok(str, ":");
    char *content = strtok(NULL, ":");
    for (int i = 0; i < SIZE_OF_ARRY2(IdDevice); ++i) {
        if(IdDevice[i][0]) // check if IdDevice[i][0] is not NULL
            if(strcmp(header, IdDevice[i][0])==0) {
                // printf("header: %s, IdDevice: %s, content: %s\n", header, IdDevice[i][1], content);
                if(IdDevice[i][2]!=NULL)    // check if has device
                    SendCommandToDevice(i, content);
                else
                    printf("- There is no device: \"%s\"\n", IdDevice[i][0]);
                // break;       // break when u just wanna send to one device
            }
    }
}
```

Hàm ServerCommand() sẽ bóc tách dữ liệu từ server thành 2 phần. Một phần là header sẽ chứa tên thiết bị, phần còn lại là content chứa nội dung điều khiển. Tiếp đó so sánh header để tìm ra thiết bị tương ứng và gửi content tới thiết bị đó. Nếu không muốn điều khiển nhiều thiết bị trùng tên (khác ID) cùng lúc thì sử dụng lệnh break ở cuối.

c) Chương trình ESP8266

Chương trình chính của ESP8266.

=

```
while(!SentID()); // wait until set successful

while(1){
    // Read all the lines of the reply from server and print them to Serial
    while(client.available()) {
        String line = client.readStringUntil('\r');
        Serial.println(line);

        if(line.equals("state=1"))
          Switch(true);
        else if(line.equals("state=0"))
          Switch(false);
    }
}
```

Chương trình chính sẽ gửi thông tin khởi tạo thiết bị liên tục tới khi thành công. Tiếp đó chuyển sang chế độ chờ thông tin điều khiển và so sánh dữ liệu để tiến hành thực hiện công việc tương ứng.

Hàm SentID() thực hiện gửi thông tin khởi tạo (“INIT:ID”) tới Master. Sau khi gửi sẽ delay một khoảng thời gian trước khi đọc dữ liệu phản hồi. Vì phải mất một khoảng thời gian dữ liệu mới phản hồi lại được (mất thời gian xử lý và truyền qua mạng). Bên dưới là delay(500), các bạn có thể điều chỉnh lại cho phù hợp hơn.

```
bool SentID() {
  Serial.print("Send identify: "); Serial.println(DEVICE_NAME);
  client.print(INIT_IdDevice);
  delay(500);
  while(client.available()) {
      String line = client.readStringUntil('\r');
      if(line.equals("OK")) {
          Serial.println("the device was successful setted");
          return true;
      }
  }
  return false;
}
```

Hàm thực hiện bật tắt đèn :

```
bool Switch(bool set) {
  Serial.print("Switch is ");
  if(set) Serial.println("on !");
  else  Serial.println("off !");
  digitalWrite(LIGHT_PIN, set);
  return true;
}
```

d) Kết quả

Chương trình hoàn thiện Master và Esp8266 trên github.

3. Những hạn chế cần cải tiến.

Vì mục đích của chương trình đê thực hiện việc demo hệ thống nên còn khá đơn giản. Không hỗ trợ việc kiểm tra lỗi phát sinh như khi mất mạng thiết bị sẽ có không phát hiện ra cũng như không tự động kết nối lại.

Không có cơ chết phản hồi từ thiết bị. Ví dụ như khi thiết bị bật đèn thành công thì thiết bị cần báo cho Master biết để cập nhật thông tin vào thiết bị.

Van tuan |
04/19/2021

Nếu điều khiển nhiều đèn thì chương trình chính của esp viết như thế nào ạ

NGUYỄN MINH CHÍ |
06/20/2017

em dùng server trên windown 7, em kết nối với server mà chỉ in liên tục Send identify: Toilet Light, không kết nối được, em sửa port trên chương trình nạp 80, monng anh chỉ giúp em làm sao kết nối với được ạ, em cám ơn

Hoang Trong |
06/14/2017

Cho mình hỏi, nếu mình muốn truy cập bằng điện thoại thì được k ạ ? tại mình kết nối raspi bằng remote desktop, thì chỉ máy tính mình mới lên đc thôi, còn lên bằng điện thoại thì k thấy gì cả ? :(

NGUYỄN MINH CHÍ |
05/20/2017

em cấu hình lai làm chạy được trên window 7 thu muc xampp\htdocs, nhưng gặp sự cố lỗi như thế này và em chưa kết nối với esp8266, mong anh chỉ thêm Notice: Undefined variable: stateName in C:\xampp\htdocs\lazy\function\print-HTML.php on line 27, Anh em nao giup do toi khac phuc loi nay duoc khong

le dai |
04/06/2017

https://github.com/quydai94/home-center

1 2 3 4 > >|

Hiển thị từ 1 đến 5 trong tổng số 16 (4 trang)

Viết đánh giá

Họ và tên:

Đánh giá của bạn:

Lưu ý: Không hỗ trợ HTML!

Bình chọn: Dở ○ ○ ○ ○ ○ Hay

Nhập mã bảo vệ:

Tiếp tục

Công ty TNHH MLAB
Số chứng nhận kinh doanh: 0106356768. Nơi cấp: Sở kế hoạch và đầu tư Thành Phố Hà Nội. Ngày cấp: 07/11/2013
Trụ sở : Số 30F9 - Ngõ 104 Lê Thanh Nghị - Hai Bà Trưng - Hà Nội
Email mua bán hàng: smarttechvn.group@gmail.com
Email hỗ trợ kỹ thuật : mlab.services.tech@gmail.com
website:https://mlab.vn
Số điện thoại: 02436231170 hoặc 0984058846 hoặc 0866828846