



(<https://pivietaam.com.vn/>)



Trang chủ (<https://pivietaam.com.vn/>) / Tin tức công nghệ (<https://pivietaam.com.vn/tin-tuc-cong-nghe>)

/ Học lập trình với Raspberry Pi (<https://pivietaam.com.vn/tin-tuc-cong-nghe/hoc-lap-trinh-voi-raspberry-pi-pivietaam-com-vn>)

/ Bài 14 : Tập tành viết thư viện cho Raspbery Pi – Arduino Pi (test thư viện I2C cho module BH1750)

TIN MỚI



VIDEO



Bài 14 : Tập tành viết thư viện cho Raspbery Pi – Arduino Pi (test thư viện I2C cho module BH1750)

🕒 16:00 - 15/01/2019

Thư viện I2C cho raspbery pi. Thư viện cho module BH1750

- » Hướng cài đặt Hệ điều hành và Remote Desktop cho Raspberry Pi nhanh chóng và cực kỳ đơn giản (<https://pivietaam.com.vn/huong-cai-dat-he-dieu-hanh-va-remote-desktop-cho-raspberry-pi-nhanh-chong-va-cuc-ky-don-gian-pivietaam-com-vn.html>)
- » Remote Desktop Raspberry Pi không cần Wifi, mạng LAN và IP (<https://pivietaam.com.vn/remote-desktop-raspberry-pi-without-wifi-lan-and-ip-pivietaam-com-vn.html>)
- » Camera nhiệt giải pháp tuyệt vời cho mùa Covid-19 (<https://pivietaam.com.vn/camera-nhiet-giai-phap-tuyet-voi-cho-mua-covid-19-pivietaam-com-vn.html>)
- » Lập trình cơ bản với OpenPLC trên Raspberry Pi (<https://pivietaam.com.vn/lap-trinh-co-ban-voi-openplc-tren-raspberry-pi-pivietaam-com-vn.html>)
- » Hướng dẫn cài đặt OpenPLC trên Raspberry Pi (<https://pivietaam.com.vn/huong-dan-cai-dat-openplc-tren-raspberry-pi-pivietaam-com-vn.html>)

Message us

(<https://m.me/599911456>)



1. Chém gió

Mục tiêu của thư viện là nó có tính phổ quát, có thể sử dụng đơn giản và dễ dàng tái sử dụng hay gắn vào thư viện khác. Trong cộng đồng mạng có rất nhiều thư viện cho các module khác nhau sử dụng giao tiếp I2C nhưng thư viện cho Pi thì rất hiếm. Đa phần là được viết cho framework arduino. Lý do cũng đơn giản vì độ phổ biến của Arduino và cũng vì Pi phù hợp với những công việc cao cấp hơn. Tuy nhiên hoàn toàn hoàn toàn có thể viết thư viện cho các thiết bị đó trên Pi. Vậy tại sao không thiết kế một nền tảng để tái sử dụng thư viện cho Arduino trên Pi, như thế Pi có thể có ngay hàng ngàn thư viện có sẵn. WiringPi là một ví dụ, trên nền tảng framework wiring (giống như arduino), có thể tái chế code phần nào từ arduino chuyển sang. WiringPi có cách viết I2C không giống với arduino nên không thể tái chế. Mục đích của viết này mình sẽ giải quyết vấn đề đó bằng cách viết một thư viện đơn giản với I2C, có hầu hết các hàm tương đồng với arduino,; như vậy mọi người có thể dựa trên nó để sử dụng các thư viện module dựa trên thư viện Wire của arduino. Mình gọi nó là “Arduino-Pi”.

Ở cuối bài sẽ có bài test với module ánh sáng BH1750. Module này dùng chuẩn I2C để giao tiếp, và trên mạng đang có thư viện cho arduino. Mình sẽ dùng thư viện này để test với Pi.

2. Vẽ vờ

Nếu bạn chưa biết thiết lập I2C trên pi thì hãy xem bài viết “Lập trình Raspberry giao tiếp I2C”. Trong đó có ví dụ về sử dụng I2C với thư viện WiringPi. Ở đây mình sẽ trình bày cách sử dụng trực tiếp từ các hàm có sẵn trong hệ thống để giao tiếp I2C. Cũn rất đơn giản thôi.

Các bước thực hiện :

1. Mở cổng I2C bằng hàm open(). (<https://linux.die.net/man/2/open>) Hàm này trả về file descriptor (https://en.wikipedia.org/wiki/File_descriptor).
2. Gắn địa chỉ I2C slave vào file descriptor.
3. Gửi hoặc nhận thông qua file descriptor. Sử dụng các hàm căn bản là write() (<https://linux.die.net/man/2/write>) hoặc read(). (<https://linux.die.net/man/2/read>)

Trong hệ thống linux, các xuất nhập vào ra (truy xuất I/O) hay xử lý với file đều được xử lý thông qua “file descriptor”.

Chương trình gửi và nhận I2C trên Pi

Message us

(<https://m.me/59991145f>)

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/i2c-dev.h>

int main() {

    int file_i2c;
    int length;
    unsigned char buffer_tx[2] = {0};
    unsigned char buffer_rx[5] = {0};

    //----- OPEN THE I2C BUS -----
    char *filename = (char*)"/dev/i2c-1";
    if ((file_i2c = open(filename, O_RDWR)) < 0)
    {
        printf("Failed to open the i2c bus");
        return -1;
    }

    // Add address if I2C slave
    int addr = 0x13; // address of i2c slave
    if (ioctl(file_i2c, I2C_SLAVE, addr) < 0)
    {
        printf("Failed to acquire bus access and/or talk to slave.\n");
        return -1;
    }

    buffer_tx[0] = 0x01;
    buffer_tx[1] = 0x02;

    while(1){

        //----- WRITE BYTES -----
        length = 2; // Number of bytes to write
        if (write(file_i2c, buffer_tx, length) != length)
        {
            printf("Failed to write to the i2c bus.\n");
        } else {
            printf("Write %d bytes to arduino.\n", length);
        }
        usleep(2000000);
    }
}
```

[Message us](https://m.me/59991145f)<https://m.me/59991145f>

```

//----- READ BYTES -----
length = 5;          // Number of bytes to read
if (read(file_i2c, buffer_rx, length) != length)
{
    printf("Failed to read from the i2c bus.\n");
} else {
    printf("Data read from arduino: \n");
    for(int i=0; i<length; i++) {
        printf("-\tbuffer_rx[%d]: %#x\n", i, buffer_rx[i]);
    }
}

usleep(2000000);

}

return 0;
}

```

Để biên dịch chương trình

```
gcc -std=c11 simple.c -o simple
```

Hàm `open(filename, O_RDWR)` sẽ mở cổng I2C để đọc và ghi (`O_RDWR`). Chú ý hãy sử dụng câu lệnh

Để xem filename sẽ tương ứng với `/dev/i2c-0` hay `/dev/i2c-1`.

```
ls /dev/i2c*
```

Thiết lập địa chỉ I2C slave addr = 0x13.

Sử dụng hàm ghi (`write(file_i2c, buffer_tx, length)`) để viết vào file_i2c chuỗi buffer_tx có độ dài là length. Kết quả trả về của hàm này là số bytes ghi được.

Hàm đọc `read(file_i2c, buffer_rx, length)` yêu cầu slave gửi length bytes và ghi vào buffer_rx. Kết quả của hàm trả về số bytes nhận được.

OK. Như thế đã hoàn thiện chương trình cho Pi. Giờ chuyển sang với Arduino, mình sẽ không giải thích gì về chương trình vì nó có lẽ đã rất quen thuộc với các bạn.

Message us

(<https://m.me/599911456>)

```
#include <Wire.h>

void setup() {
  Wire.begin(0x13);          // join i2c bus with address #8
  Wire.onReceive(receiveEvent); // register event
  Wire.onRequest(requestHandle);
  Serial.begin(9600);        // start serial for output
}

void loop() {
  delay(100);
}

// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void receiveEvent(int howMany) {
  Serial.print("receive: ");
  while(Wire.available()) // slave may send less than requested
  {
    byte c = Wire.read(); // receive a byte as character
    Serial.print(c);      // print the character
    Serial.print(',');
  }
  Serial.println();
}

void requestHandle() {
  Serial.println("Send 5 bytes");
  for(int i=0; i<5; i++){
    Wire.write(i);
  }
}
```

Kết nối I2C giữa arduino và Pi. Sử dụng công cụ I2C tool để kiểm tra xem kết nối tới arduino đã chính xác chưa.

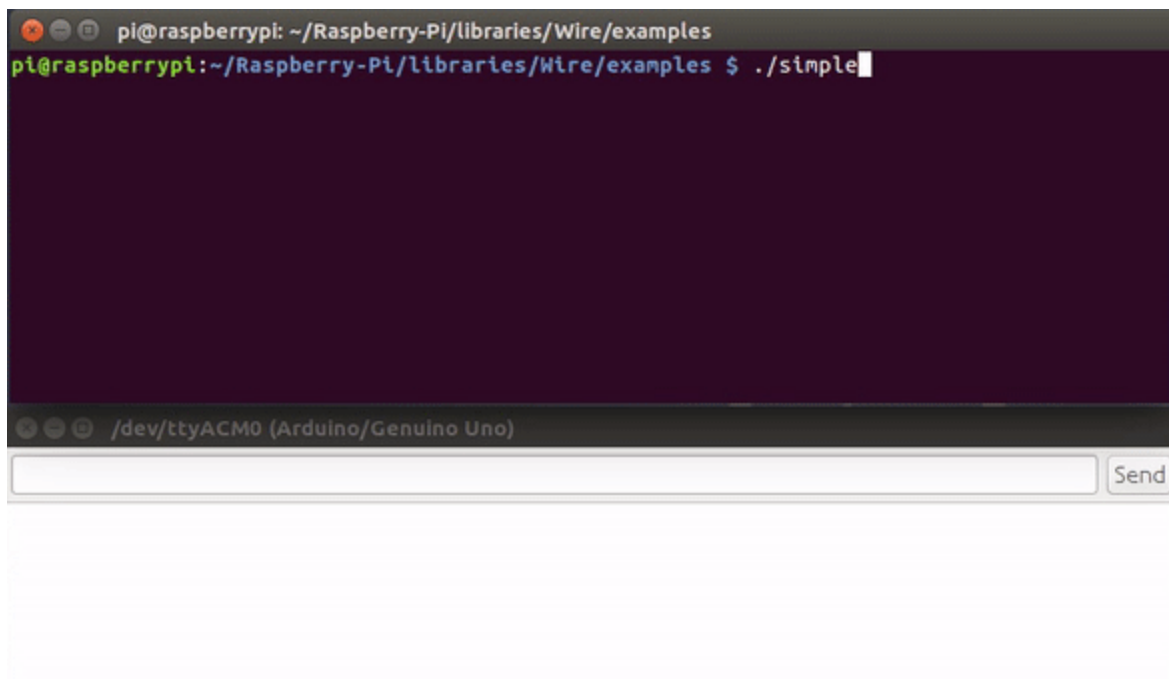
```
sudo apt-get install i2c-tools
sudo apt-get update
sudo i2cdetect -y 0
//or
sudo i2cdetect -y 1
```

Message us

(<https://m.me/599911456>)

Kết quả khi chạy hai chương trình :





Bạn thấy chưa, ăn bánh còn lằng ngoằng hơn phải không ? Chỉ với mấy hàm có sẵn trên linux, là có thể kết nối trực tiếp tới arduino thông qua I2C rồi. Nếu muốn chương trình hiệu quả về mặt tốc độ và tiết kiệm bộ nhớ thì chỉ cần dùng chương trình trên là ổn. Nhưng không, Raspberry rất tuyệt, rất nhanh, rất khỏe nên mấy thứ tiết kiệm đó không đáng so với mặt tiện dụng của thư viện, và mục đích của bài viết này cũng là xây dựng một thư viện có cách viết giống với arduino. Vậy, hãy cùng tiếp tục tọc mạch tới phần tiếp theo.

3. Tọc mạch Arduino Pi

Chú ý rằng Pi không hỗ trợ slave mode nên không thể thiết lập slave giống như arduino, do đó sẽ không có hàm `onReceive()`, `onRequest()`. Hàm `setClock()` cũng sẽ không được hỗ trợ. Mặc định tốc độ của I2C là standard mode 100kbit/s và có thể điều chỉnh bằng cách :

```
sudo modprobe -r i2c_bcm2708 && sudo modprobe i2c_bcm2708 baudrate=400000
```

Ở đây thiết lập tốc độ là fast mode 400kbit/s. Mình chưa có điều kiện test với các tốc độ khác nhau. Nếu ai đó đã test rồi hãy vui lòng chia sẻ lại thông tin với mình và mọi người.

Như vậy các hàm trong thư viện I2C cho Pi sẽ bao gồm :

- `bool begin(char *path = "/dev/i2c-1");`
- `int8_t beginTransmission(uint16_t addr);`
- `int8_t endTransmission(void);`
- `int8_t requestFrom(uint16_t addr, uint8_t length);`
 - `uint8_t available(void);`
 - `uint8_t write(int8_t dt);`
- `uint8_t write(const uint8_t *dt, uint8_t length);`

Message us

(<https://m.me/59991145f>)

- int16_t read();
- uint8_t read(uint8_t *dt, uint8_t length);

Cách sử dụng các hàm giống hệt như trên Arduino. Ngoài ra có thêm hàm read(uint8_t *dt, uint8_t length) có thể đọc nhiều bytes và ghi vào mảng dt[].

Thư viện được public trên github (<https://github.com/DuongNguyenHai/Raspberry-Pi/tree/master/libraries>) của mình. Đoạn code dưới đây sẽ viết lại chương trình đầu bài với phong cách của arduino.

```
// g++ send-receive.c ../src/Wire.cpp -o send-receive

#include "../src/Wire.h"
#include <unistd.h>

#define addrSlave 0x13

int main(int argc, char const *argv[])
{
    Wire.begin();
    unsigned char buffer_tx[2] = {0x01,0x02};
    unsigned char buffer_rx[5] = {0};
    while(1) {
        printf("Write 2 bytes to arduino.\n");
        Wire.beginTransmission(addrSlave);
        Wire.write(buffer_tx, 2);
        Wire.endTransmission();
        usleep(2000000);
        Wire.requestFrom(addrSlave, 5);
        if(Wire.available()) {
            Wire.read(buffer_rx, 5);
            printf("Data read from arduino: \n");
            for(int i=0; i<5; i++) {
                printf("-\tbuffer_rx[%d]: %#x\n", i, buffer_rx[i]);
            }
        }
        usleep(2000000);
    }
    return 0;
}
```

Để biên dịch chương trình.

g++ send-receive.c ../src/Wire.cpp -o send-receive

Message us

(<https://m.me/599911456>)

Có một điều thú vị là các bạn không dùng được hàm write() với read() như trên đầu bài, vì nó trùng tên với hàm write() và read() mà mình muốn sử dụng cho thư viện. Do đó mình chuyển qua dùng hàm pwrite() và pread(). Sử dụng 2 hàm này cũng không có khác biệt nhiều.

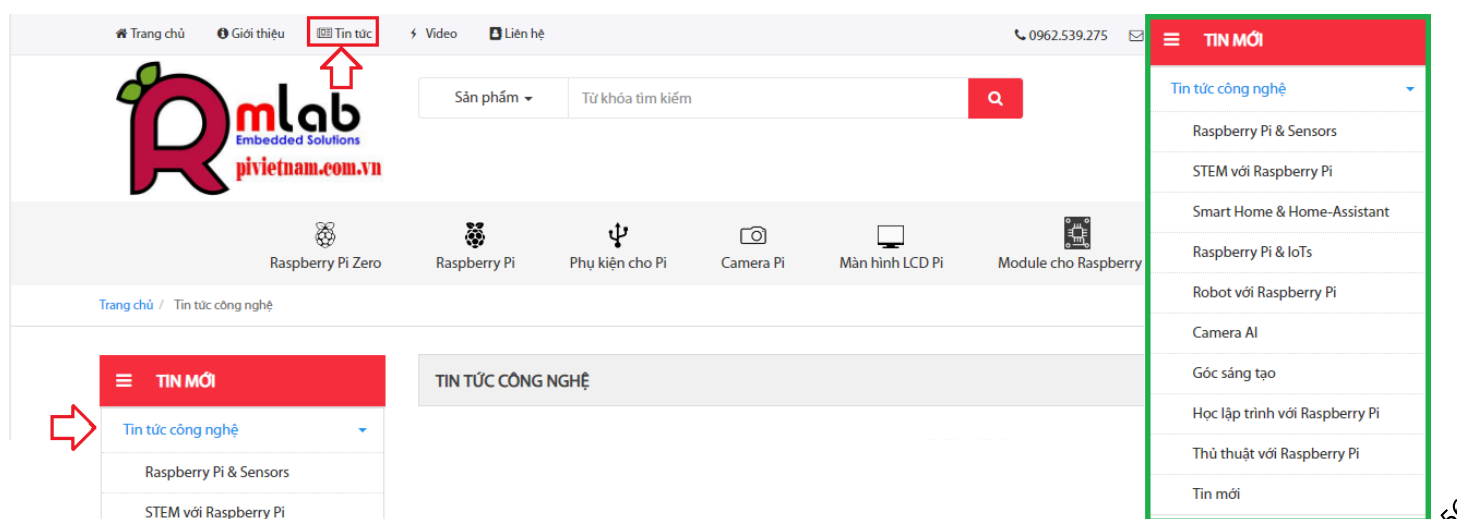
Một bài test khác với thư viện cho arduino là module đo cường độ ánh sáng BH1750 (<https://github.com/DuongNguyenHai/Raspberry-Pi/tree/master/libraries/BH1750>). Các bạn có thể dùng nguyên thư viện của BH1750 cho Pi mà không cần chỉnh sửa gì cả. Chú ý hãy xóa các define dành riêng cho compile của arduino đi.

4. Hết nước

Thư viện được mình cho phép sử dụng tự do trong cộng đồng với bất kỳ mục đích nào. Các bạn có thể sửa sao chép, chỉnh sửa hay sử dụng cá nhân đều được. Lưu ý rằng thư viện vừa mới được viết nên nó rất cần nhiều bài kiểm tra khác nhau để chỉnh sửa và cập nhật thêm. Nếu mọi người có khúc mắc hay muốn bổ sung thêm thì hãy bình luận bên dưới hoặc trên github. Mình luôn sẵn sàng đón nhận thông tin từ mọi người.

Để cập nhật các tin tức công nghệ mới các bạn làm theo hướng dẫn sau đây :

Các bạn vào Trang chủ >> Tin tức. ở mục này có các bài viết kỹ thuật thuộc các lĩnh vực khác nhau các bạn có thể lựa chọn lĩnh vực mà mình quan tâm để đọc nhé !!!



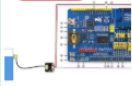
Các bạn cũng có thể kéo xuống cuối trang để xem những tin tức công nghệ mới nhất.

Message us

(<https://m.me/599911456>)

TIN TỨC

VIDEO




🕒 11:34 - 21/02/2019

Phát hiện mức chất

Chúng ta có thể phát hiện mức chất lỏng mà cảm biến

XEM THÊM >>




🕒 18:13 - 16/02/2019

Trở về tuổi thơ cùng

Xa rồi thời trốn bố mẹ sang nhà hàng xóm chơi game tay

XEM THÊM >>




🕒 14:33 - 01/02/2019

Streaming video với

Streaming video với Raspberry Pi Camera lên web

XEM THÊM >>




🕒 13:47 - 01/02/2019

Streaming video với

Streaming video với Raspberry Pi sử dụng VLC


XEM THÊM >>



Sản phẩm chính hãng


Chúng tôi cam kết 100% sản phẩm là hàng chính hãng, chất lượng cao

0962.539.275 (Mr Thùy)





Bảo hành chuyên nghiệp

Dịch vụ bảo hành chuyên nghiệp, tận tình và chu đáo cho khách hàng



Giá tốt nhất tại Việt Nam

Tự tin là nhà cung cấp sản phẩm, phụ kiện cho Raspberry Pi với giá cả tốt nhất tại Việt Nam

VIDEO

VỀ CHÚNG TÔI

Website uy tín cung cấp Raspberry Pi chính hãng , và các phụ kiện , board mạch mở rộng cho Raspberry Pi tại Việt Nam.

📍 Số 30F9 - Ngõ 104 Lê Thanh Nghị - Hai Bà Trưng - Hà Nội

☎ 02436.231.170

✉ smarttechnvn.group@gmail.com

HOTLINE TƯ VẤN TRỰC TIẾP

086.262.8846 (Mr Thùy) (tel:0962539275)

(Thời gian làm việc 8h - 17h30, thứ 2 tới thứ 7. Hỗ trợ Online ngoài giờ hành chính và chủ nhật.)

VỀ CHÚNG TÔI

Giới thiệu (<https://pivietnam.com.vn/ve-chung-toi>)

Lịch sử hình thành (<https://pivietnam.com.vn/lich-su-hinh-thanh>)

Đội ngũ lãnh đạo (<https://pivietnam.com.vn/doi-ngu-lanh-dao>)

Tuyển dụng (<https://pivietnam.com.vn/tuyen-dung-quy-i>)

Liên hệ (<https://pivietnam.com.vn/lien-he>)

Message us

(<https://m.me/599911456>)

<https://pivietnam.com.vn/bai-14-tap-tanh-viet-thu-vien-cho-raspberry-pi-arduino-pi-test-thu-vien-i2c-cho-module-bh1750-pivietnam-com-vn.html>

9/11

**ĐÃ THÔNG BÁO**
BỘ CÔNG THƯƠNG<http://online.gov.vn/Home/WebDetails/101224>

CHÍNH SÁCH

Hướng dẫn mua hàng online (<https://pivietnam.com.vn/huong-dn-mua-hang-online-mlab-vn>)

Chính sách vận chuyển và giao nhận (<https://pivietnam.com.vn/chinh-sach-van-chuyen-va-giao-nhan-mlab-vn>)

Chính sách kiểm hàng (<https://pivietnam.com.vn/chinh-sach-kiem-hang>)

Thông tin chuyển khoản (<https://pivietnam.com.vn/thong-tin-chuyen-khoan-mlab-vn>)

Hỗ trợ sau bán hàng (<https://pivietnam.com.vn/ho-tro-sau-ban-hang-mlab-vn>)

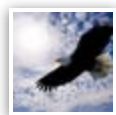
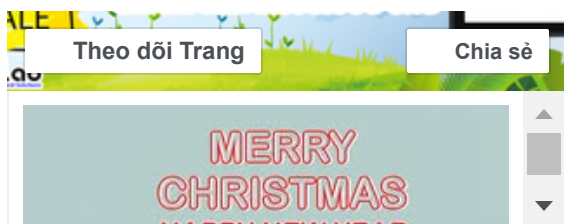
Chính sách bảo hành (<https://pivietnam.com.vn/chinh-sach-bao-hanh-mlab-vn>)

Chính sách đổi trả, hoàn tiền (<https://pivietnam.com.vn/chinh-sach-doi-tra-hoan-tien-mlab-vn>)

Chính sách bảo mật thông tin (<https://pivietnam.com.vn/chinh-sach-bao-mat-thong-tin-mlab-vn>)

ĐĂNG KÝ NHẬN BẢN TIN

FACEBOOK FANPAGE

**MLAB**
5.572 người theo dõi

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://pivietnam.com.vn/>

Số điện thoại : 02436.231.170 or 086.262.8846

Message us

<https://m.me/599911456>



(tel:0962539275)

Message us

(<https://m.me/599911456>)

