



☎ 086.262.8846



(https://pivietnam.com.vn/)

Từ khóa tìm kiếm



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

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

/ Bài 4 : Lập trình Raspberry Pi sử dụng cổng truyền thông UART

TIN MỚI



VIDEO



Bài 4 : Lập trình Raspberry Pi sử dụng cổng truyền thông UART

🕒 14:53 - 15/01/2019

Bài 4 : Lập trình Raspberry Pi sử dụng cổng truyền thông UART

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

Message us

(https://m.me/599911456)



Giao tiếp qua cổng UART với Raspberry Pi

1. Giới thiệu UART trên Raspberry Pi

Raspberry Pi hỗ trợ chuẩn Uart trên chân BCM14 (TX) và BCM15 (RX). Mặc định uart được sử dụng làm serial console; nó sẽ gửi toàn bộ thông tin của kernel trong quá trình Pi khởi động. Bạn có thể kết nối trực tiếp uart của pi thông qua mạch USB-TTL để xem thêm. Để sử dụng Uart với mục đích riêng bạn phải giải phóng chân Uart. Lưu ý với Raspberry Pi 3 Uart còn được sử dụng để kết nối bluetooth, và chân BCM14 và BCM15 trở thành mini uart port.

Để Setup UART cho Pi dùng trong kết nối với các MCU khác, các bạn có thể tham khảo hướng dẫn trong video sau :

[Học Raspberry Pi] Video 6 : Setup UART trên Raspberry Pi, demo kết n...

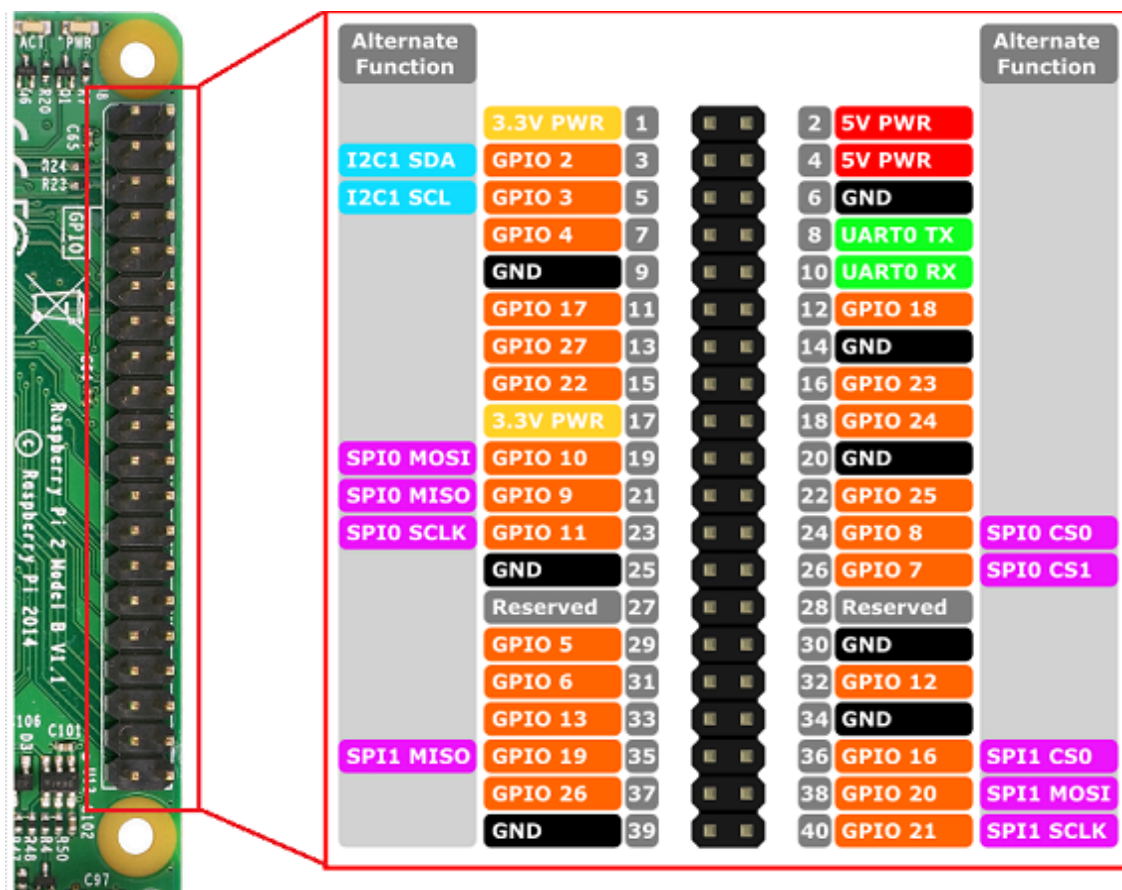


2. Giao tiếp UART với Raspberry Pi

Message us

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





Hình 1 : chân GPIO của Raspberry Pi 2 model B

Dưới đây là hai ví dụ đơn giản giao tiếp giữa Raspberry và Arduino Vn01:

- Ví dụ 1 : Gửi chuỗi ký tự từ Raspberry tới Arduino.
- Ví dụ 2 : Gửi chuỗi ký tự từ Arduino tới Raspberry.

Thông tin : raspberry Pi 2 model B (http://mlab.vn/index.php?route=product/product&filter_name=raspberry&product_id=1060) và Arduino Vn01 (http://mlab.vn/1697809-uno-vn01-arduino-uno-phien-ban-viet-nam.html?filter_name=arduino)

2.1 Kết nối phần cứng

Kết nối giữa Arduino và Pi như sau : RX của arduino <-> TX của Pi và ngược lại TX của arduino <-> RX của Pi, kết nối chung 2 chân GND với nhau.

2.2.1 Lập trình giao tiếp UART

Chuẩn bị thư viện cho C là wiring-pi (<http://wiringpi.com/reference/serial-library/>) như trong bài 1 đã đề cập tới.

Thư viện cho python là pySerial (http://pyserial.readthedocs.io/en/latest/pyserial_api.html)

```
sudo apt-get install python-serial
```

Khi cần dùng thư viện chỉ cần thực hiện "import serial"

Message us

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

a. Lập trình bằng C

Ví dụ 1

Chương trình “uart-send.c” trên Raspberry

```
// Compile : gcc -Wall uart-send.c -o uart-send -lwiringPi

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringSerial.h>

int main() {

    int fd;

    printf("Raspberry's sending : \n");

    while(1) {
        if((fd = serialOpen ("/dev/ttyAMA0", 9600)) < 0 ){
            fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno));
        }
        serialPuts(fd, "hello");
        serialFlush(fd);
        printf("%s\n", "hello");
        fflush(stdout);
        delay(1000);
    }
    return 0;
}
```

Chương trình trên Pi sẽ gửi chuỗi ký tự “hello” thông qua uart tới arduino. Nó sẽ gửi liên tục sau 1s.

Thư viện sử dụng : #include <wiringSerial.h>

- Mở cổng kết nối Serial.

int serialOpen (char *device, int baud);

Luôn nhớ rằng cổng uart có tên là ttyAMA0. Hàm này sẽ trả về file-descriptor. Nếu file-descriptor lỗi sẽ có giá trị là -1.

- Hàm gửi dữ liệu

Void serialPutchar (int fd, unsigned char c) ;

Hàm này gửi sẽ gửi 1 byte. Nếu muốn gửi một mảng (hoặc chuỗi ký tự) có thể sử dụng hàm sau.

void serialPuts (int fd, char *s) ;

- Hàm void serialFlush (int fd) : chờ đến khi dữ liệu được gửi xong hoặc bỏ qua tất cả dữ liệu được gửi tới. Hàm này rất cần với uart, nếu không có sẽ không thể gửi chính xác được. Bạn nên bỏ hàm này đi và xem thử. Có lẽ đây là một bug nhỏ đối với chương trình này vì đáng lý ra chương trình sau khi thực hiện lệnh gửi dữ liệu có thể thực hiện ngay lập tức lệnh tiếp theo, dữ liệu sẽ được uart tự động ngầm gửi đi. Chương trình không cần phải chờ.

Các hàm được hỗ trợ : wiringPi (<https://projects.drogon.net/raspberry-pi/wiringpi/serial-library/>)

Để compile được chương trình thực hiện lệnh

```
gcc -Wall uart-send.c -o uart-send -lwiringPi
```

Chương trình “uart-receive.ino” trên Arduino

```
void setup() {
    Serial.begin(9600);    // opens serial port, baudrate : 9600 bps
}

void loop() {

    if (Serial.available() > 0) {
        String str = Serial.readString();
        Serial.print("Received: ");
        Serial.println(str);
    }
}
```

- Khởi tạo Uart :

```
Serial.begin(9600);
```

Lưu ý rằng baudrate bằng 9600 giống như trên Pi.

- Hàm kiểm tra xem có dữ liệu tới uart không :

```
Serial.available()
```

Hàm này trả về số byte nhận được.

- Đọc dữ liệu chuỗi ký tự :

```
Serial.readString();
```

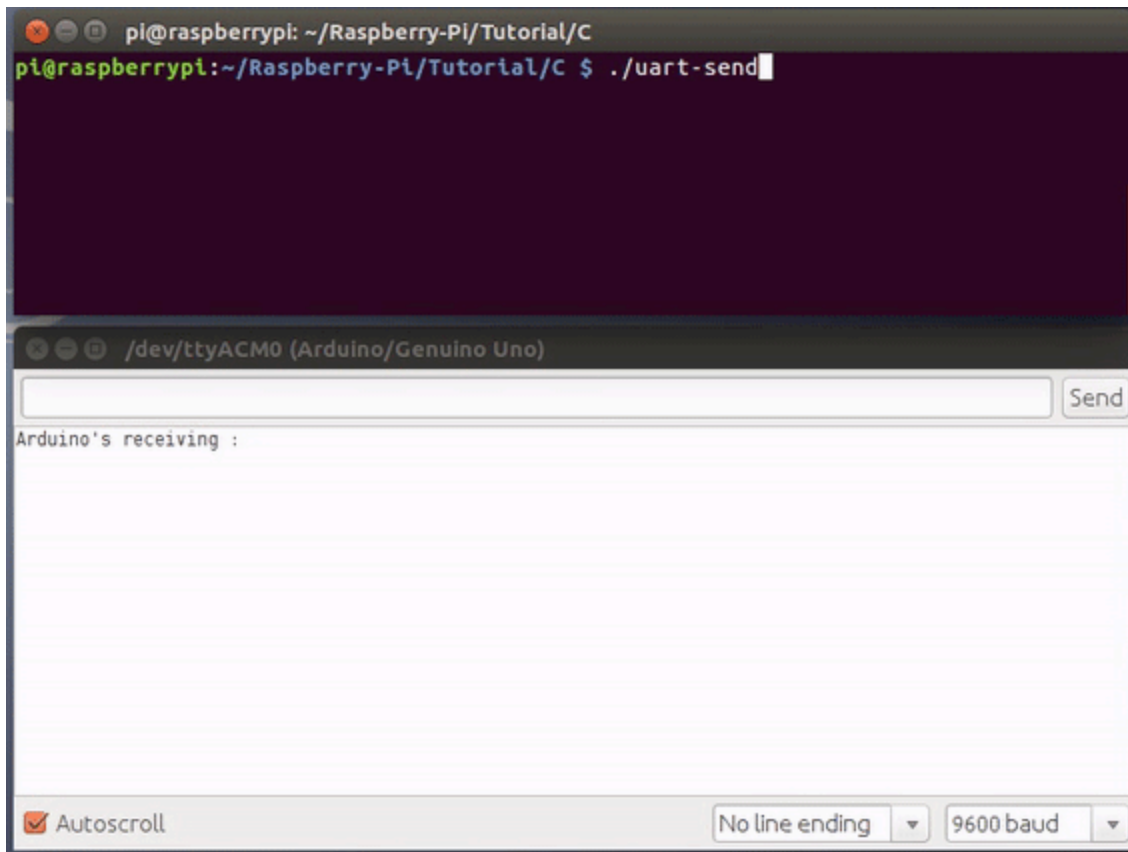
Trả về dữ liệu kiểu String.

- Lưu ý:

```
Serial.print();
```

Mục đích để hiển thị lên màn hình kết quả, nhưng nó gửi qua Uart nên cũng đồng thời nó gửi dữ liệu tới Pi.

Kết quả :



Ví dụ 2

Chương trình “uart-receive.c” trên Raspberry

Message us

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

```
// Compile : gcc -Wall uart-receive.c -o uart-receive -lwiringPi

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringSerial.h>

int main() {

    int fd;
    char c;
    printf("Raspberry's receiving : \n");

    while(1) {
        if((fd = serialOpen ("/dev/ttyAMA0", 9600)) < 0 ){
            fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno)) ;
        }else{
            do{
                c = serialGetchar(fd);
                printf("%c",c);
                fflush (stdout);
            }while(serialDataAvail(fd));
        }
    }
    return 0;
}
```

- Kiểm tra xem có dữ liệu tới không :

int serialDataAvail (int fd);

Hàm này trả về số byte dữ liệu xuất hiện. Trả về -1 nếu bị lỗi.

- Đọc dữ liệu

int serialGetchar (int fd);

Trả về một byte dữ liệu tới. Con trỏ của nó sẽ tự tăng sau mỗi lần gọi. Hàm sẽ chờ tới tận 10s nếu không có dữ liệu. Trả về -1 nếu bị lỗi.

Chương trình “uart-send.ino” trên Arduino

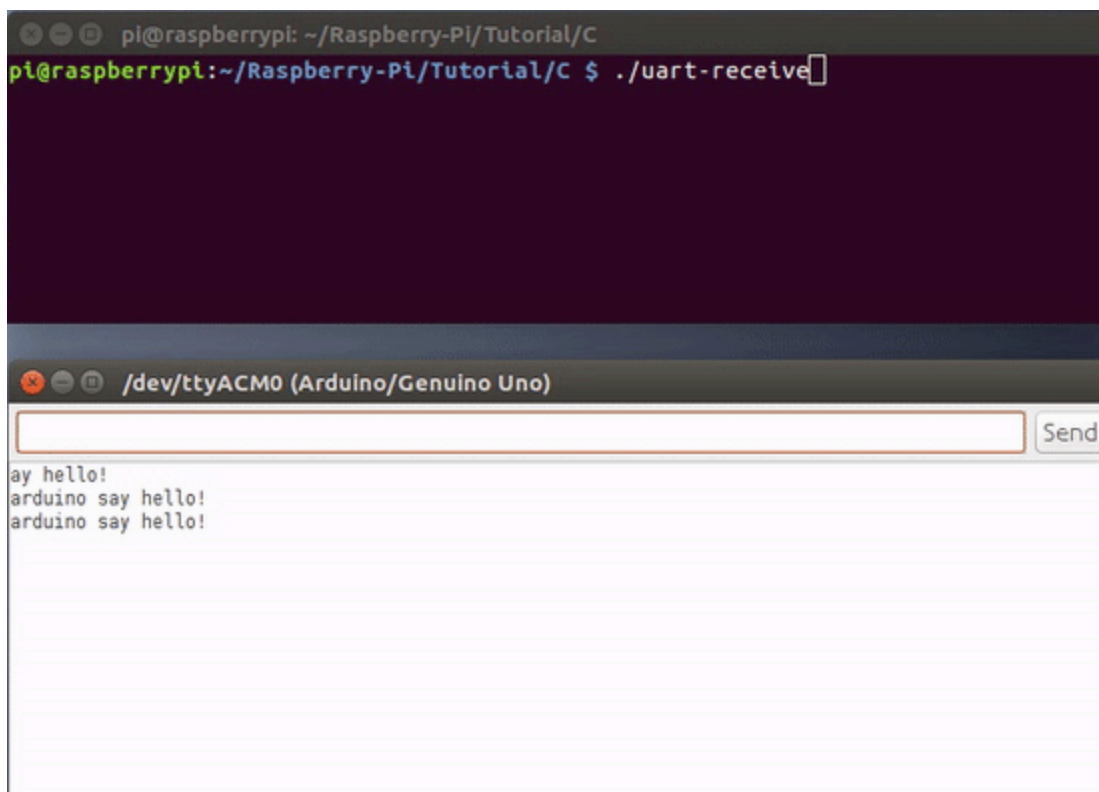
Message us

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



```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    Serial.println("arduino say hello!");  
    delay(1000);  
}
```

Kết quả :



b. Lập trình Python

Xem một số ví dụ về thư viện serial (<http://pyserial.readthedocs.io/en/latest/shortintro.html#readline>)

Ví dụ 1

Chương trình “uart-send.py” trên Raspberry

Message us

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


```
#!/usr/bin/python3

import time
import serial

ser = serial.Serial(
    port = '/dev/ttyAMA0',
    baudrate = 9600,
    parity = serial.PARITY_NONE,
    stopbits = serial.STOPBITS_ONE,
    bytesize = serial.EIGHTBITS,
    timeout = 1
)

print("Raspberry's sending : ")

try:
    while True:
        ser.write(b'hehe')
        ser.flush()
        print("hehe")
        time.sleep(1)
except KeyboardInterrupt:
    ser.close()
```

- Thiết lập serial

```
serial.Serial()
```

Nếu không thể khởi tạo Serial được sau timeout thì sẽ thông báo lỗi. Ở đây đặt timeout = 1s.

- Gửi dữ liệu :

```
int write(data);
```

Hàm này có thể gửi nhiều byte dữ liệu cùng lúc. Hàm trả về là số byte đã gửi.

- Hàm flush()

Cũng như với chương trình C, cần flush nếu không serial sẽ không gửi đúng.

- Đóng Serial

```
close();
```

Nếu có interrupt xảy ra, ví dụ như CTR+C thì chương trình sẽ bắt được sự kiện đó và sẽ đóng serial. Lí do vì Serial sẽ không tự đóng lại khi chương trình bị ngắt không kiểm soát. Như vậy nó sẽ tiếp tục được mở và không cho phép chương trình nào khác sử dụng nó → chương trình chạy sẽ bị lỗi ở lần tiếp theo.

Đây là python version 3 nên khi chạy bạn thực hiện lệnh

python3 uart-send.py

Kết quả tương tự như chương trình C**Ví dụ 2****Chương trình “uart-receive.py” trên Raspberry**

```
#!/usr/bin/python3

import time
import serial

ser = serial.Serial(
    port = '/dev/ttyAMA0',
    baudrate = 9600,
    parity = serial.PARITY_NONE,
    stopbits = serial.STOPBITS_ONE,
    bytesize = serial.EIGHTBITS,
    timeout = 1
)

print("Raspberry's receiving : ")

try:
    while True:
        s = ser.readline()
        data = s.decode()           # decode s
        data = data.rstrip()       # cut "\r\n" at last of string
        print(data)               # print string

except KeyboardInterrupt:
    ser.close()
```

- Đọc dữ liệu

readline()

Hàm trả về là những byte đọc được. Hàm đọc kết thúc với timeout=1 được thiết lập bên trên.

- Lưu ý :

Ở trên có sử dụng 2 hàm dưới đây trước khi hiển thị kết quả

data=s.decode()

data=data.rstrip()

Message us

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

Mọi người nên kiểm tra kết quả khi không có 2 hàm trên để hiểu thêm về kết quả nhận được.

Kết quả tương tự với chương trình C

3. Sử dụng cổng USB để kết nối

Uart được Raspberry Pi với mục đích log cho kernel. Nếu không thiếu thốn hay bắt buộc phải sử dụng cổng uart các bạn có thể chuyển sang cổng usb để thay thế. Với hệ điều hành linux các kênh xuất nhập đều có thể kết nối thông qua file-descriptor, dù uart hay usb thì chỉ cần thay đổi file-descriptor thì có thể kết nối tới nó. Toàn bộ code bên trên có thể dùng kết nối qua cổng USB, chỉ cần thay "ttyAMA0" bằng tên cổng USB có dạng "ttyUSB*". Các bạn có thể kết nối arduino với thiết bị USB-TTL như PL2303 và sau đó cắm thiết bị vào Pi qua cổng USB là có thể kết nối. Lưu ý đây là sử dụng cổng USB để kết nối chứ không phải kết nối hoàn toàn bằng giao thức USB vì phía bên arduino sử dụng phương thức uart.

Khi PL2303 kết nối tới Raspberry, cần kiểm tra thiết bị đang nằm trên cổng USB nào

```
dmesg | grep tty
```

Kết quả sẽ trả về các Serial trên Pi. Trong đó sẽ có kết quả dạng như bên dưới:

```
pl2303 converter now attached to ttyUSB0
```

PL2303 đang kết nối tới cổng ttyUSB0.

Kết quả chương trình sẽ giống hệt như với kết nối trực tiếp với Uart.

Thông tin tham khảo :

[1] <https://electrosome.com/uart-raspberry-pi-python/> (<https://electrosome.com/uart-raspberry-pi-python/>)

[2] http://elinux.org/RPi_Serial_Connection (http://elinux.org/RPi_Serial_Connection)

[3] <http://www.briandorey.com/post/Raspberry-Pi-3-UART-Overlay-Workaround>
(<http://www.briandorey.com/post/Raspberry-Pi-3-UART-Overlay-Workaround>)

[4] http://pyserial.readthedocs.io/en/latest/pyserial_api.html
(http://pyserial.readthedocs.io/en/latest/pyserial_api.html)

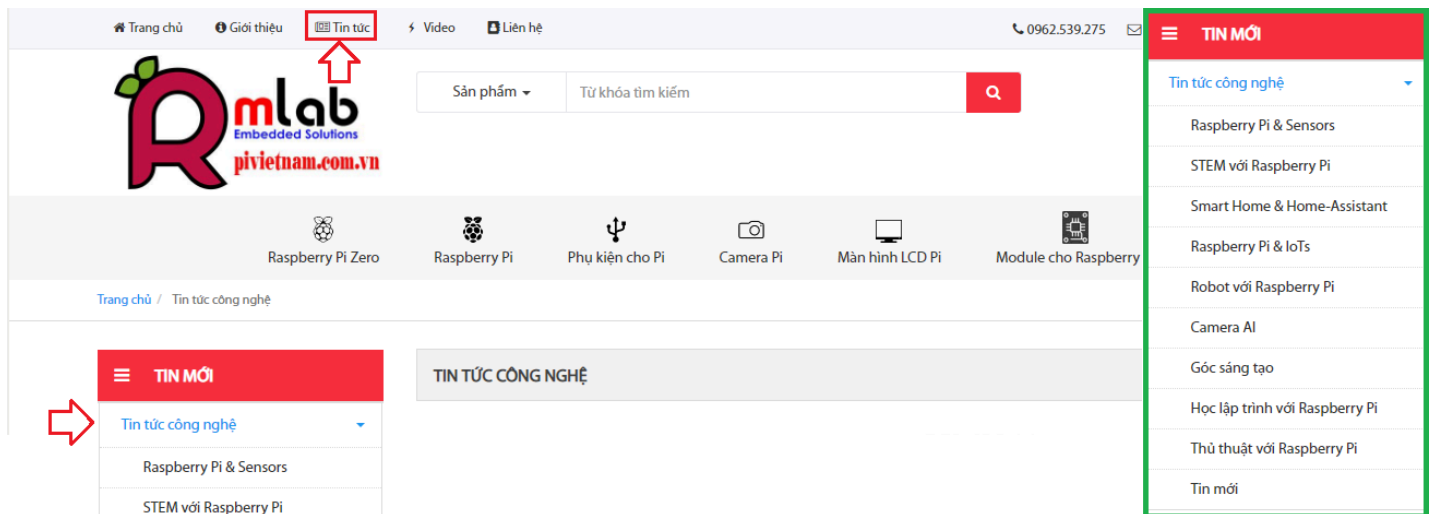
Để cập nhật các tin tức công nghệ mới các bạn làm theo hướng dẫn sau đây :

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 !!!

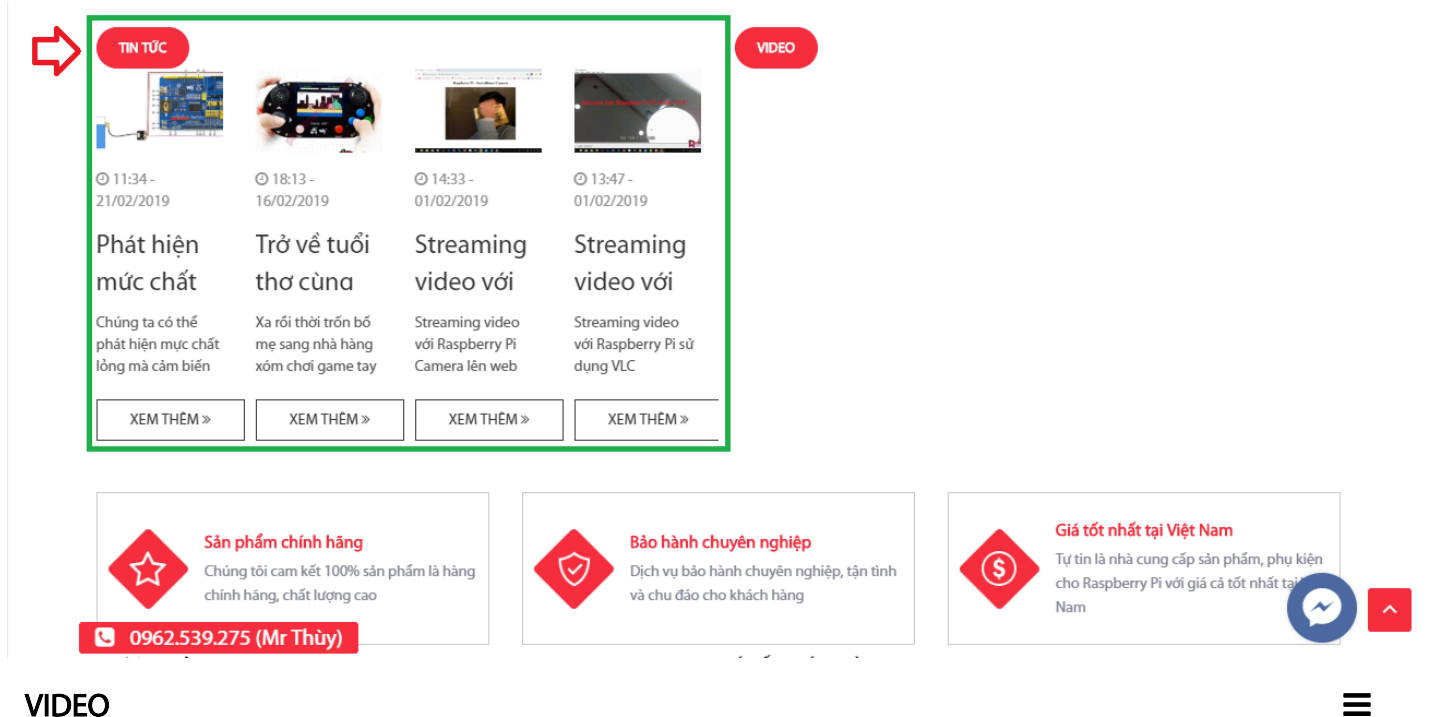
Message us

<https://t.me/599911456>





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



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

✉ smarttechvn.group@gmail.com

Message us

(https://m.me/599911456)

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>)

**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

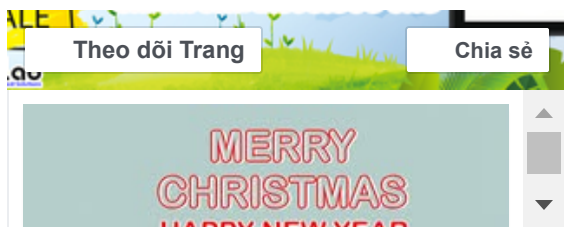
[Đăng ký](#)

FACEBOOK FANPAGE

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



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



(tel:0962539275)

Message us

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

