

Mục lục

1	Cài đặt hệ điều hành cho Raspberry Pi	2
1.1	Chuẩn bị	2
1.1.1	Phần cứng	2
1.1.2	Phần mềm	3
1.2	Cài đặt hệ điều hành	3
1.2.1	Cài đặt hệ điều hành Raspbian	3
1.2.2	Cài đặt hệ điều hành Raspbian bằng NOOBS	4
1.3	Thiết lập ban đầu cho Raspberry Pi	5
1.3.1	Thiết lập ngôn ngữ, múi giờ và kiểu bàn phím	6
1.3.2	Cập nhật Raspberry Pi	8
1.3.3	Thiết lập dòng cung cấp cho ngõ ra USB	8
1.3.4	Bật SSH	9
2	Điều khiển và sao chép dữ liệu với Raspberry Pi	10
2.1	Điều khiển bằng cách kết nối trực tiếp với màn hình, bàn phím và chuột	10
2.1.1	Màn hình hỗ trợ cổng HDMI	10
2.1.2	Màn hình hỗ trợ cổng VGA	10
2.2	Điều khiển bằng giao tiếp nối tiếp thông qua cổng RS232	11
2.2.1	Cách kết nối	11
2.2.2	Hệ điều hành Windows	11
2.2.3	Hệ điều hành Ubuntu	12
2.3	Điều khiển bằng giao tiếp nối tiếp thông qua cáp USB to COM PL2303	13
2.4	Điều khiển từ xa khi Raspberry Pi có kết nối mạng	14
2.5	Điều khiển Raspberry Pi từ xa với giao diện đồ họa khi có kết nối Internet	15
2.5.1	Sử dụng VNC trên Window	15
2.5.2	Sử dụng VNC trên Ubuntu	15
3	Mạng Internet với Raspberry Pi	17
3.1	Kết nối internet với dây mạng LAN	17
3.2	Kết nối internet với USB Wifi	17
3.2.1	Cài đặt Driver	18
3.2.2	Cài đặt địa chỉ IP tĩnh	18
3.3	Sử dụng chung Wifi với Laptop	20
3.3.1	Hệ điều hành Ubuntu	20
4	Tự động đăng nhập Raspberry Pi sau khi khởi động	23

5	Chạy chương trình Python trên Raspberry Pi	24
5.1	Python trên Raspberry Pi	24
5.2	Tự động chạy một chương trình python sau khi Reboot	24
5.2.1	Sử dụng cron	24
5.2.2	Sử dụng profile	26
5.2.3	Sử dụng cron kết hợp với tạo file .sh	27
6	Điều khiển phần cứng qua các chân GPIO	28
6.1	Giới thiệu	28
6.2	Mô tả chức năng của các nhóm chân	29
6.3	Các cài đặt và cấu hình cần thiết để sử dụng được các chân GPIO	29
6.3.1	Cài đặt thư viện RPi.GPIO	29
6.3.2	Xác nhận cấu hình I2C	29
6.3.3	Xác nhận cấu hình SPI	32
6.4	Sử dụng các chân GPIO để điều khiển phần cứng cùng với ngôn ngữ lập trình Python	32
6.4.1	Kiểm tra chức năng của các chân GPIO	32
6.4.2	Các lệnh cơ bản trong thư viện RPi.GPIO	33
6.4.3	Ngắt và phát hiện tín hiệu cạnh	35
6.4.4	Sử dụng chân GPIO với chức năng PWM	36

Chủ đề 1

Cài đặt hệ điều hành cho Raspberry Pi

1.1 Chuẩn bị

1.1.1 Phần cứng

- Máy tính Raspberry Pi: bạn có thể sử dụng Model *A*, *B*, *A+*, *B+*, *Pi 2*, *Pi 3*, *PiZero*.



Hình 1.1: Raspberry Pi model B+

- Thẻ nhớ SD: Dung lượng 4GB trở lên, tốc độ cao. Nên sử dụng thẻ nhớ có dung lượng 8GB. Có thể sử dụng thêm Adapter (tùy theo phiên bản Pi mà bạn sử dụng).



Hình 1.2: Thẻ nhớ SD và Adapter

- Nguồn điện cấp cho Pi hoạt động: nguồn DC 5V – 1A

- Các ngoại vi khác: dây kết nối HDMI (hoặc cáp VGA và cáp chuyển đổi HDMI to VGA), màn hình có cổng HDMI (hoặc hoặc màn hình có cổng VGA), chuột, bàn phím, dây kết nối mạng.
- * Nếu không có các thiết bị ngoại vi trên thì sử dụng các loại cáp truyền nhận dữ liệu giữa Pi và máy tính cũng có thể thực hiện được, ví dụ: RS232, PL2303.

1.1.2 Phần mềm

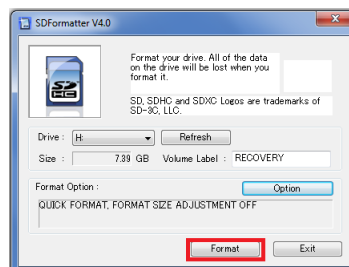
- Phần mềm Format thẻ nhớ, ví dụ: SDFormatter.
Địa chỉ: https://www.sdcard.org/downloads/formatter_4/
- Phần mềm ghi hệ điều hành vào thẻ nhớ, ví dụ: Win32DiskImage.
Địa chỉ: <https://sourceforge.net/projects/win32diskimager/>
- Hệ điều hành: Chọn một trong hai hệ điều hành sau để tải về máy.
 - + Trong phần này mình sử dụng hệ điều hành Raspbian.
Địa chỉ: <https://www.raspberrypi.org/downloads/raspbian/>
 - + Cài đặt bằng NOOBS: tích hợp nhiều hệ điều hành vào một gói, bạn thích sử dụng hệ điều hành nào thì chọn hệ điều hành đó.
Địa chỉ: <https://www.raspberrypi.org/downloads/noobs/>

1.2 Cài đặt hệ điều hành

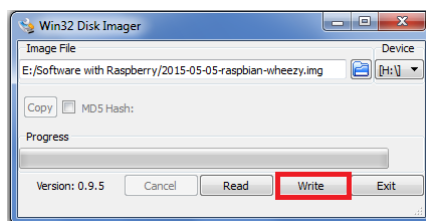
Bạn chọn một trong các cách sau để cài đặt.

1.2.1 Cài đặt hệ điều hành Raspbian

- Cài đặt các phần mềm ở phần chuẩn bị, ví dụ: SDFormatter và Win32DiskImage.
- Giải nén thư mục chứa hệ điều hành Raspbian (file .zip) được file .img.
- Mở phần mềm SDFormatter, định dạng lại thẻ nhớ chuẩn bị ghi hệ điều hành lên. Chọn **Format**. Thông báo xuất hiện chọn YES.



Hình 1.3: Format lại thẻ nhớ để cài đặt hệ điều hành



Hình 1.4: Ghi hệ điều hành vào thẻ nhớ

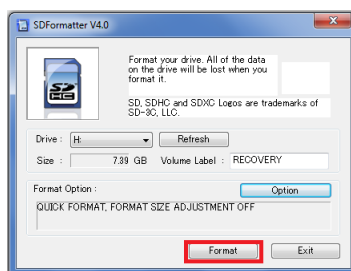
- Mở phần mềm Win32DiskImage để ghi hệ điều hành vào thẻ nhớ, chọn đường dẫn đến file .img đã giải nén ở bước trên. Chọn **Write**. Thông báo xuất hiện chọn YES.
- Đợi cho quá trình ghi hệ điều hành hoàn thành là được.
- Gắn thẻ nhớ vào Pi, kết nối màn hình, chuột, bàn phím, cấp nguồn khởi động và tiến hành một số cài đặt cần thiết khác.

1.2.2 Cài đặt hệ điều hành Raspbian bằng NOOBS

Khi cài đặt hệ điều hành bằng bản NOOBS, nó cũng có chứa hệ điều hành Raspbian, bạn có thể cài đặt hệ điều hành Raspbian bằng cách này.

Muốn có nhiều hệ điều hành khác được chọn thì bạn cần có kết nối Internet (sử dụng dây LAN).

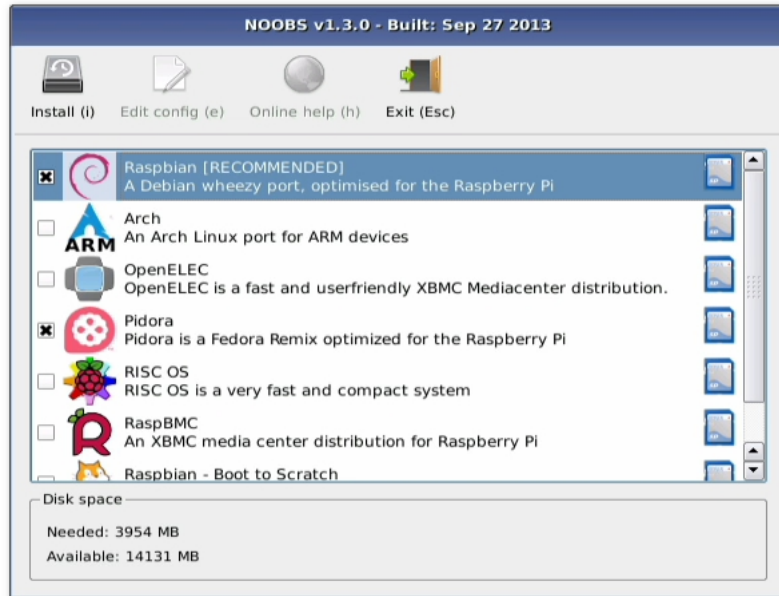
- Cài đặt các phần mềm ở phần chuẩn bị, ví dụ: **SDFormatter** (không cần sử dụng phần mềm ghi hệ điều hành).
- Giải nén thư mục chứa hệ điều hành trong bản NOOBS (file .zip).
- Mở phần mềm **SDFormatter**, định dạng lại thẻ nhớ chuẩn bị chép hệ điều hành lên. Chọn **Format**. Thông báo xuất hiện chọn YES.



Hình 1.5: Format lại thẻ nhớ để cài đặt hệ điều hành

- Copy toàn bộ nội dung trong thư mục đã giải nén ở trên (của bản NOOBS) vào thẻ nhớ.

- Gắn thẻ nhớ vào Pi, kết nối màn hình, chuột và bàn phím (hoặc các phương pháp tương đương khác), sẽ có một danh sách các hệ điều hành được đưa ra, bạn chọn hệ điều hành muốn cài đặt rồi nhấn **Install** sau đó đợi quá trình cài đặt xong, khởi động lại Pi là hoàn thành.



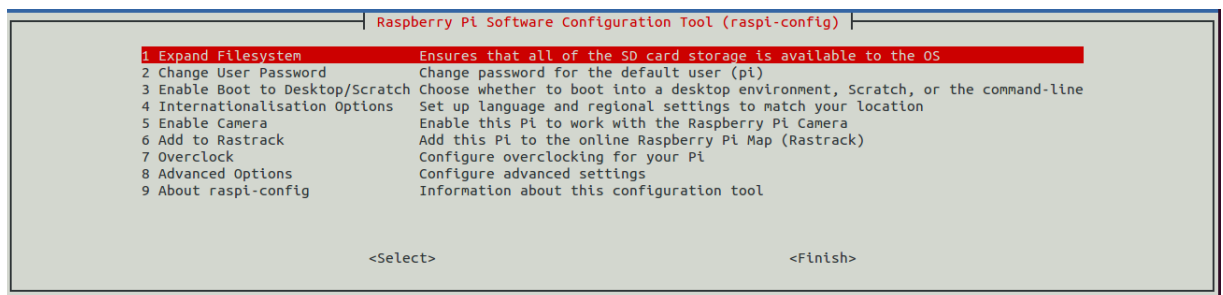
Hình 1.6: Các hệ điều hành trong bản NOOBS

- Sau khi cài đặt hệ điều hành xong, tiến hành một số cài đặt cần thiết khác.

1.3 Thiết lập ban đầu cho Raspberry Pi

Sau khi cài đặt và khởi động, bạn có thể muốn thiết lập lại các cài đặt mặc định của hệ điều hành sao cho phù hợp với bạn. Sau khi thiết lập xong, reset lại Pi: `sudo reboot`

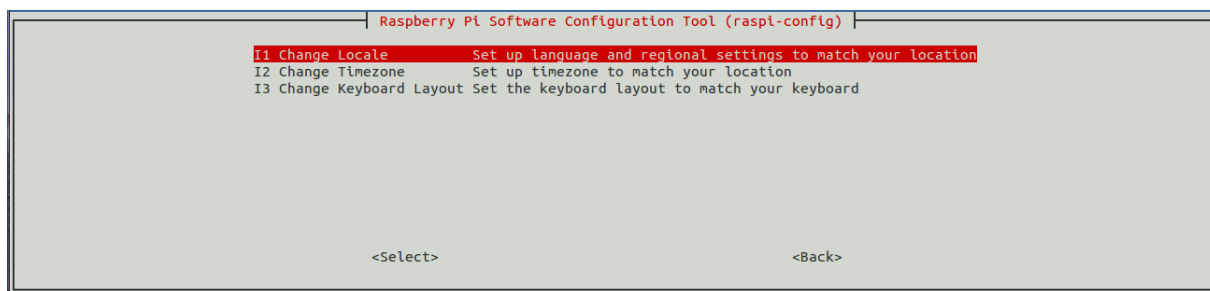
Thực hiện lệnh: `sudo raspi-config` để tiến hành cài đặt. Các tùy chọn thiết lập như hình dưới:



Hình 1.7: Các tùy chọn thiết lập cho Pi

1.3.1 Thiết lập ngôn ngữ, múi giờ và kiểu bàn phím

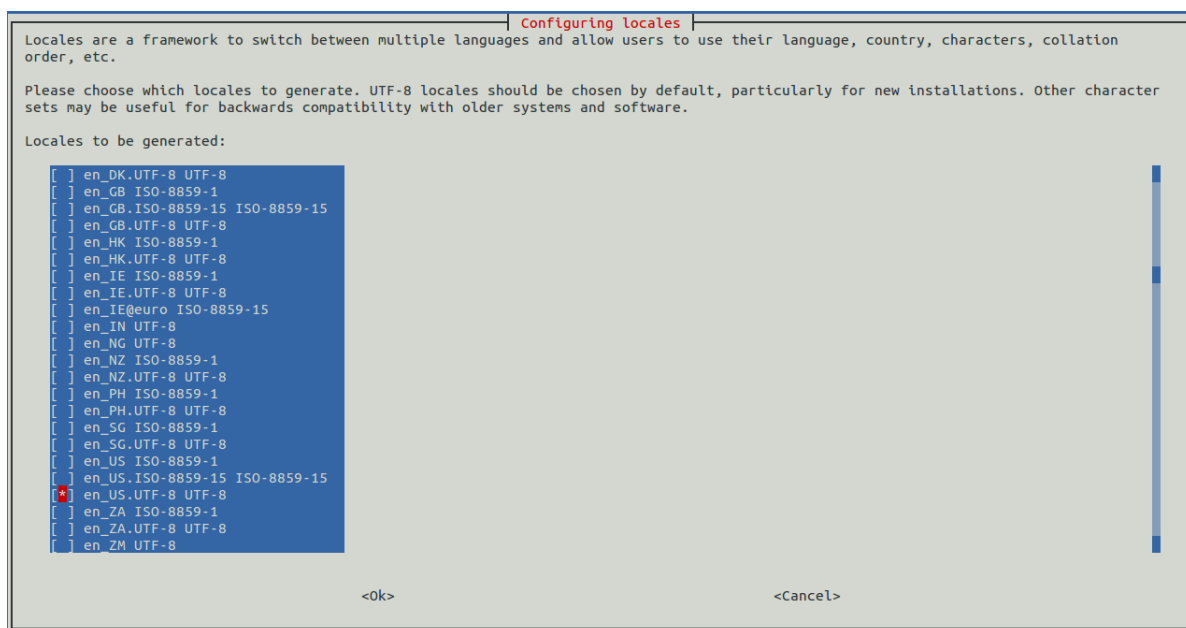
Thay đổi ngôn ngữ, múi giờ, kiểu bàn phím: Internationalisation Options, được các lựa chọn sau:



Hình 1.8: Các tùy chọn thiết lập ngôn ngữ, múi giờ, kiểu bàn phím

- Chọn **Change Locale**: để thay đổi ngôn ngữ.

Đổi `en_GB.UTF-8` UTF-8 thành `en_US.UTF-8` UTF-8 (sử dụng dấu cách space để chọn hoặc bỏ chọn).



Hình 1.9: Thay đổi ngôn ngữ thành `en_US.UTF-8` UTF-8

- Chọn **Change TimeZone**: để thay đổi múi giờ.
Chọn **Asia** rồi chọn tiếp **Ho_Chi_Minh**.



(a) Chọn Asia



(b) Chọn Ho_Chi_Minh

Hình 1.10: Thay đổi múi giờ

- Thay đổi kiểu bàn phím, dùng lệnh:

```
1 $ sudo nano /etc/default/keyboard
```

Nội dung file như sau:

```
1 # KEYBOARD CONFIGURATION FILE
2
3 # Consult the keyboard(5) manual page.
4
5 XKBMODEL="pc105"
6 XKBLAYOUT="gb"
7 XKBVARIANT=""
8 XKBOPTIONS=""
9
10 BACKSPACE="guess"
```

Đổi các khai báo sau: `pc105` thành `pc104` và `gb` thành `us`, sau khi thay đổi nội dung như sau:

```
1 # KEYBOARD CONFIGURATION FILE
2
3 # Consult the keyboard(5) manual page.
4
5 XKBMODEL="pc104"
6 XKBLAYOUT="us"
7 XKBVARIANT=""
8 XKBOPTIONS=""
9
10 BACKSPACE="guess"
```

Nhấn `Ctrl + X + Y` và `Enter` để lưu nội dung thay đổi.

* Để thoát khỏi `raspi-config` nhấn phím `Esc`.

1.3.2 Cập nhật Raspberry Pi

– Kiểm tra cập nhật:

```
1 $ sudo apt-get update
```

– Tiến hành cập nhật:

```
1 $ sudo apt-get upgrade
```

Nếu có các thay đổi bạn cần chọn `Y` để bắt đầu cập nhật.

– Cập nhật firmware:

```
1 $ sudo rpi-update
```

1.3.3 Thiết lập dòng cung cấp cho ngõ ra USB

Dòng mặc định là $600mA$, cần tăng dòng lên $1200mA$ để sử dụng các USB.

– Mở file `config.txt` để thay đổi nội dung:

```
1 $ sudo nano /boot/config.txt
```

– Thêm nội dung sau vào cuối file `config.txt` (dùng `Ctrl + V` để di chuyển nhanh):

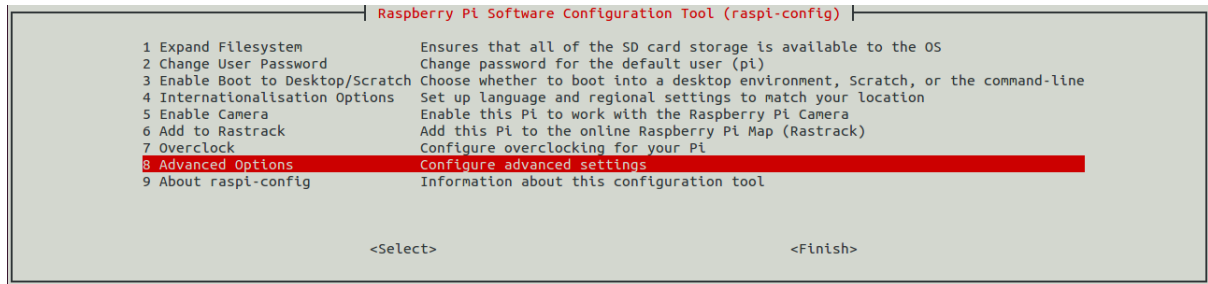
```
1 max_usb_current=1
```

Nhấn `Ctrl + X + Y` để lưu thay đổi.

1.3.4 Bật SSH

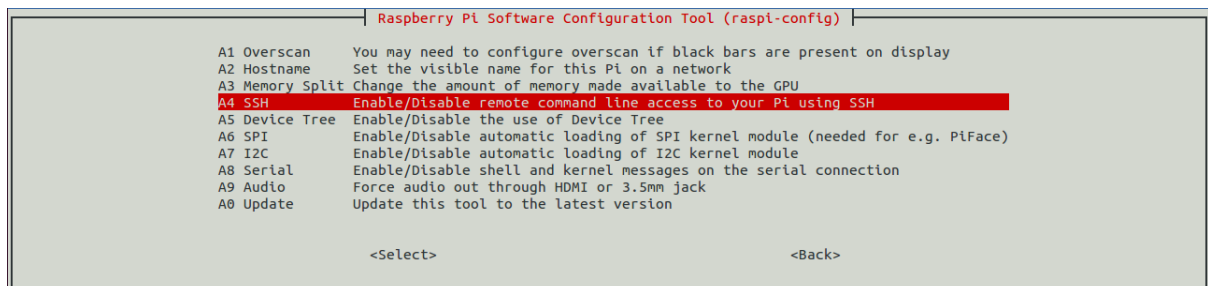
SSH cho phép điều khiển Pi từ xa thông qua mạng Internet.

- Chọn Advanced Options: hình 1.11.



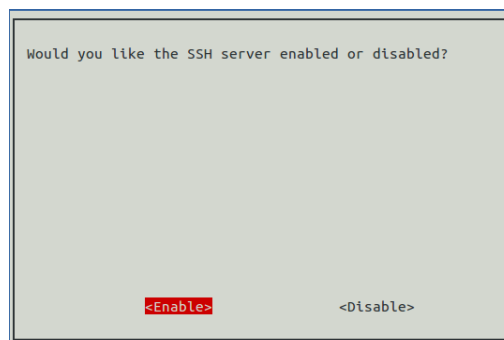
Hình 1.11: Mở các tùy chọn nâng cao Advanced Options

- Chọn A4 SSH: hình 1.12.



Hình 1.12: Chọn A4 SSH

- Chọn Enable: hình 1.13.



Hình 1.13: Chọn Enable

Chủ đề 2

Điều khiển và sao chép dữ liệu với Raspberry Pi

2.1 Điều khiển bằng cách kết nối trực tiếp với màn hình, bàn phím và chuột

- *Điều khiển*: Kết nối chuột và bàn phím qua các cổng USB. Với màn hình thông thường có 2 loại: màn hình hỗ trợ cổng HDMI và màn hình hỗ trợ cổng VGA.
- *Sao chép dữ liệu*: Sử dụng USB.

2.1.1 Màn hình hỗ trợ cổng HDMI

Ta kết nối màn hình qua cable HDMI. Có thể bạn sẽ cần tùy chỉnh một số thông số cho phù hợp:

2.1.2 Màn hình hỗ trợ cổng VGA

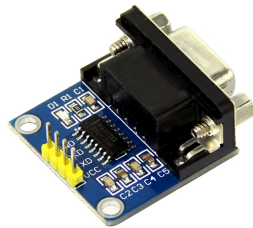
Để hiển thị được, ta cần có cable chuyển đổi từ VGA sang HDMI.



Hình 2.1: Cáp chuyển đổi từ cổng HDMI sang cổng VGA

2.2 Điều khiển bằng giao tiếp nối tiếp thông qua cổng RS232

Khi kết nối bằng module RS232, cần cấp nguồn cho Pi hoạt động.



(a) Module RS232 to TTL



(b) Cáp USB to COM

Hình 2.2: Module RS232 to TTL và Cáp USB to COM

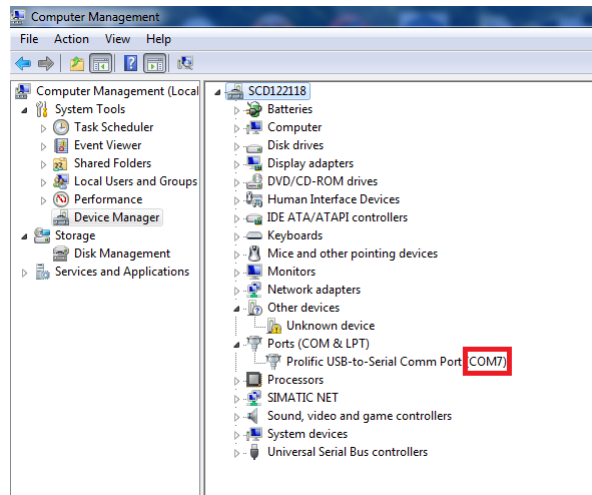
2.2.1 Cách kết nối

Thực hiện kết nối Pi và module RS232 như sau:

Pi	RS232
3.3V	VCC
TX	TX
RX	RX
GND	GND

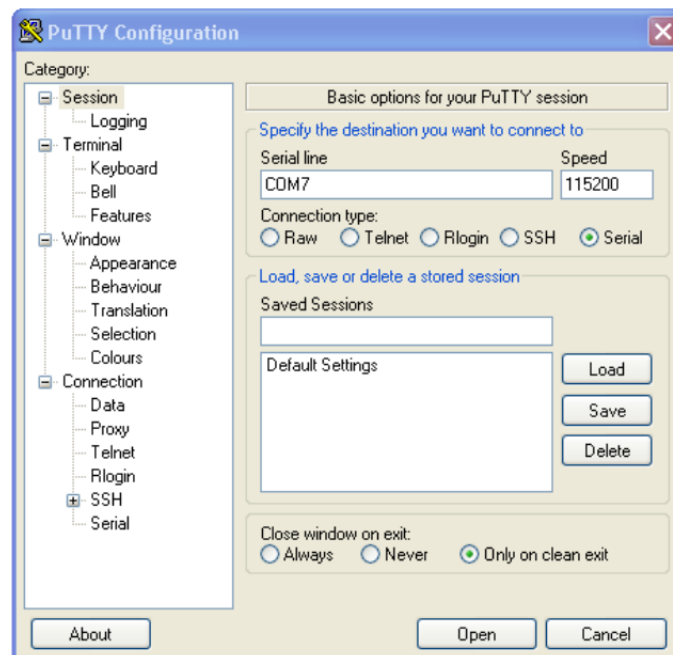
2.2.2 Hệ điều hành Windows

- Cài đặt Driver cho cáp, địa chỉ tải về:
https://drive.google.com/file/d/0B6_x_4VdySxqVUJVVV1TTXZfaEE/view?usp=sharing
- Phần mềm Putty, địa chỉ tải về:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- Cài đặt các phần mềm trên.
- Kết nối cáp với máy tính, kiểm tra xem cổng COM mấy, để sử dụng giao tiếp: Vào Device Manager, để kiểm tra như hình 2.3.



Hình 2.3: Địa chỉ COM của cáp trên máy tính

- Mở chương trình Putty lên để kết nối: Chọn **Serial** và tên cổng **COM** trong ô **Serial line**, tốc độ truyền trong ô **Speed** là 115200, như hình 2.4.



Hình 2.4: Thiết lập các thông số cho Putty để kết nối

- Nhấn **Open**, vào màn hình đen, Enter thêm một lần nữa.
- Nhập username và password để đăng nhập.

2.2.3 Hệ điều hành Ubuntu

- Cài đặt gói phần mềm **screen**: `sudo apt-get install screen` trên máy tính Ubuntu.

- Chạy lệnh sau: `sudo screen /dev/ttyUSB0 115200`
- Thực hiện xong lệnh trên, ta nhấn Enter một lần nữa để kết nối với Pi.
- Nhập username và password để đăng nhập.
- Sao chép dữ liệu: dùng USB.

2.3 Điều khiển bằng giao tiếp nối tiếp thông qua cáp USB to COM PL2303

Khi kết nối bằng cáp USB to COM PL2303 thì không cần cấp nguồn ngoài cho Pi hoạt động (do Pi sẽ lấy nguồn từ cổng USB thông qua cáp).



Hình 2.5: Cáp USB to COM PL2303

- Thực hiện kết nối Pi và cáp USB to COM PL2303 như sau:

Pi	PL2303	Màu dây
5V	VCC	Đỏ
TX	RX	Trắng
RX	TX	Xanh
GND	GND	Đen

- Phần cài đặt và điều khiển tương tự như cổng RS232 (xem *mục 2.2 trang 11*).
- Khi sử dụng cáp PL2303 ta sử dụng Driver khác mới kết nối được, địa chỉ tải về:
http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41
- Giải nén file `PL2303_Prolific_DriverInstaller_v1_12_0.zip` vừa tải về và chạy file `PL2303_Prolific_DriverInstaller_v1.7.0.exe` để cài đặt.
- Phần còn lại thực hiện giống như mục 2.2 trên cả hệ điều hành Windows và hệ điều hành Ubuntu.

2.4 Điều khiển từ xa khi Raspberry Pi có kết nối mạng

Khi Raspberry Pi có kết nối mạng Internet, ta có thể dùng các phần mềm: SSH, Remote Desktop, VNC,... để điều khiển.

- Kiểm tra địa chỉ IP của Pi bằng phần mềm: ipscan (trên Windows) hoặc nmap (trên Ubuntu).

- Phần mềm **ipscan**: giao diện trực quan, dễ sử dụng.

- Chương trình **nmap** trên Ubuntu:

- + Cài đặt chương trình:

```
1 sudo apt-get install nmap
```

- + Tìm các địa chỉ IP trong mạng nội bộ:

```
1 sudo nmap -sP 192.168.1.1-254
```

THay đổi địa chỉ IP 192.168.1.1 cho phù hợp với địa chỉ IP của bạn.

- Chọn chương trình phù hợp để điều khiển Raspberry Pi:

- Với **SSH**: không hỗ trợ giao diện đồ họa.

```
1 ssh pi@192.168.0.100
```

Trong đó: **pi** là tên user bạn cần đăng nhập, kế tiếp là địa chỉ IP của Raspberry Pi.

- Với **VNC** hoặc **Remote Desktop**: có hỗ trợ giao diện đồ họa.

- * Với **Remote Desktop** Pi cần cài đặt **xrdp**:

```
1 sudo apt-get install xrdp
```

- Tùy theo chương trình bạn chọn: ta cần phải nhập địa chỉ IP, username và password (nếu có yêu cầu điền số **port**: ta điền 22).

- Sao chép dữ liệu:

- Trên Window: dùng **Winscp**.

- Trên Ubuntu: dùng **FileZilla**.

- + **Host**: điền địa chỉ IP.

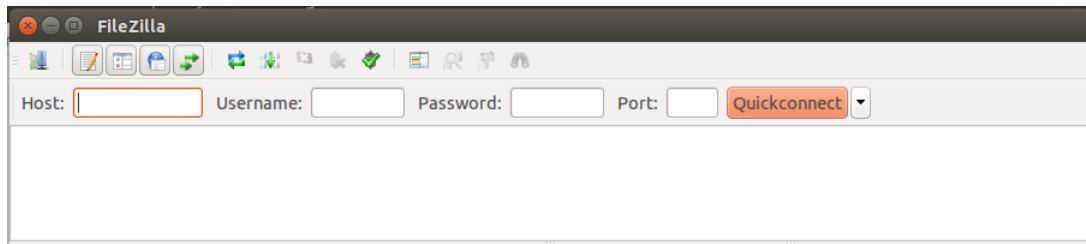
- + **Username**: tên cần đăng nhập.

- + **Password**: mật khẩu đăng nhập cho tài khoản ở ô Username.

- + **Port**: điền số 22.

- * Ta cũng cần nhập vào thông tin như trên để truy cập được Pi.

- *Lưu ý*: Phần trình bày trên áp ngay cho mạng nội bộ, khi không phải mạng nội bộ ta cần cấu hình mạng rồi mới áp dụng được hướng dẫn ở phần này.



Hình 2.6: Sao chép dữ liệu từ Raspberry Pi với FileZilla

2.5 Điều khiển Raspberry Pi từ xa với giao diện đồ họa khi có kết nối Internet

2.5.1 Sử dụng VNC trên Window

2.5.2 Sử dụng VNC trên Ubuntu

- Cài đặt phần mềm Remote Desktop Viewer (trong Ubuntu Software Center).
- Cài đặt VNC trên Raspberry Pi:

- + Cài đặt TightVNC Server trên Pi:

```
1 $ sudo apt-get install tightvncserver
```

- + Cài đặt password để VNC Client kết nối với VNC Server trên Pi:

```
1 $ tightvncserver
```

Nhập password 2 lần (tối đa 8 ký tự). Và ghi nhớ password này.
Xuất hiện dòng bên dưới:

```
1 Would you like to enter a view-only password (y/n)?
```

Nhấn n rồi Enter.

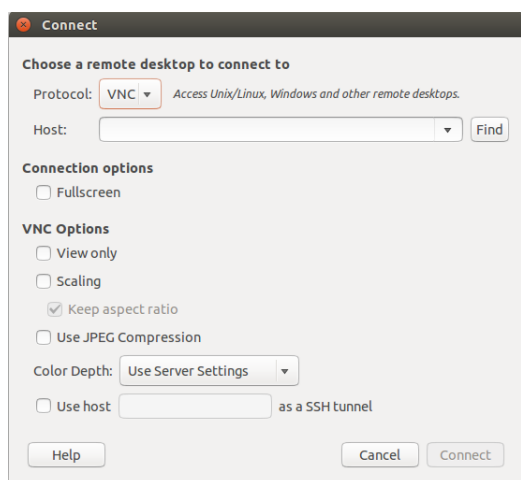
- Sử dụng VNC giao tiếp giữa máy tính Ubuntu và Pi:

- + Sử dụng ssh kết nối với Pi.

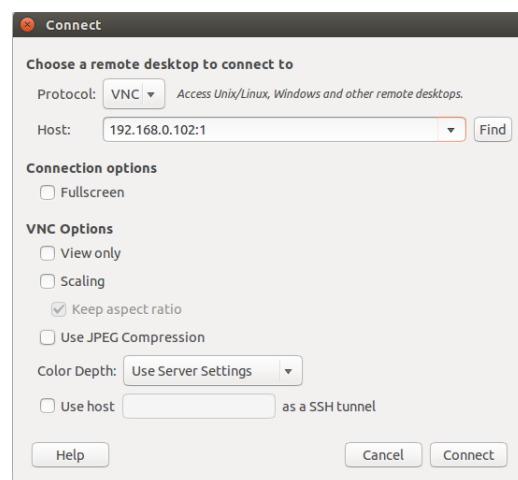
- + Gõ lệnh sau:

```
1 $ vncserver :1
```

- + Mở phần mềm Remote Desktop Viewer, click vào biểu tượng có chữ Connect trên thanh công cụ. Xuất hiện hộp thoại như 2.7a.
- + Trong ô Protocol: chọn VNC. Trong ô Host điền vào địa chỉ IP kèm với số :1, ví dụ: 192.168.0.102:1, như hình 2.7b.
- + Nhấn vào Connect để kết nối.
- + Hộp thoại như hình 2.8 xuất hiện, yêu cầu nhập password đã cài đặt ở VNC Client. Nhập password xong, nhấn vào Authenticate.

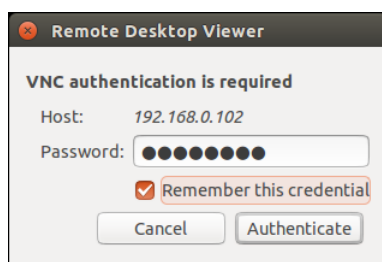


(a) Mở phần mềm Remote Desktop Viewer



(b) Điền thông tin để kết nối qua VNC

Hình 2.7: Sử dụng Remote Desktop Viewer



Hình 2.8: Nhập password xác nhận kết nối VNC

Chủ đề 3

Mạng Internet với Raspberry Pi

3.1 Kết nối internet với dây mạng LAN

Trên Raspberry Pi có hỗ trợ cổng Ethernet, chúng ta có thể kết nối dây mạng trực tiếp vào đây.



Hình 3.1: Dây LAN

3.2 Kết nối internet với USB Wifi

Trong phần này, mình sử dụng USB Wifi TP Link 725N



Hình 3.2: USB Wifi TL – WN725N

3.2.1 Cài đặt Driver

Tham khảo tại: <https://www.raspberrypi.org/forums/viewtopic.php?p=462982>
Thực hiện theo các bước sau:

- Xác định phiên bản hệ điều hành Raspbian:

```
1 $ uname -a
2 Linux raspberrypi 4.1.13+ #826 PREEMPT Fri Nov 13 20:13:22 GMT 2015
   armv6l GNU/Linux
```

Trong ví dụ trên, phiên bản hệ điều hành là 4.1.13+ #826.

- Vào địa chỉ bên dưới để tải driver:

<https://www.raspberrypi.org/forums/viewtopic.php?p=462982>

- Cài đặt driver, thực hiện các lệnh bên dưới:

```
1 wget
   https://dl.dropboxusercontent.com/u/80256631/8188eu-201xyyzz.tar.gz
2 tar -zxvf 8188eu-201xyyzz.tar.gz
3 sudo install -p -m 644 8188eu.ko /lib/modules/$(uname
   -r)/kernel/drivers/net/wireless
4 sudo insmod /lib/modules/$(uname
   -r)/kernel/drivers/net/wireless/8188eu.ko
5 sudo depmod -a
```

Đối với Raspberry Pi 2, chúng ta chỉ cần thực hiện 2 lệnh sau:

```
1 tar xzf 8188eu-2015yyzz.tar.gz
2 ./install.sh
```

Với 8188eu-201xyyzz.tar.gz là driver phù hợp với phiên bản hệ điều hành của bạn.

Bạn có thể tải driver từ máy tính rồi chép vào Raspberry Pi để cài đặt (cách này dùng cho Pi chưa được kết nối với Internet).

- * Trong quá trình cập nhật các phiên bản mới của hệ điều hành, khi đó Raspberry Pi không còn nhận USB Wifi nữa, lúc đó ta cần cài đặt Driver mới cho USB Wifi.

3.2.2 Cài đặt địa chỉ IP tĩnh

Tham khảo tại địa chỉ:

<http://weworkweplay.com/play/automatically-connect-a-raspberry-pi-to-a-wifi-network/>

Trong trường hợp, bạn cài IP tĩnh khi chuyển sang một địa chỉ IP khác (ví dụ: địa chỉ IP tĩnh 192.168.0.100 sang nơi có địa chỉ IP là 192.168.1.88) thì sẽ không kết nối được Internet, bạn cần đổi lại địa chỉ IP cho phù hợp hoặc chuyển sang địa chỉ IP động (đổi static thành dhcp và chú thích các dòng khai báo IP tĩnh – xem phần hướng dẫn bên dưới). Ta sửa đổi nội dung của 2 tập tin dưới đây:

- Tập tin: `interfaces`, mở tập tin:

```
1 $ sudo nano /etc/network/interfaces
```

và thay đổi nội dung như sau:

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 allow-hotplug eth0
6 iface eth0 inet manual
7
8 auto wlan0
9 allow-hotplug wlan0
10 iface wlan0 inet manual
11 wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
12
13 auto wlan1
14 allow-hotplug wlan1
15 iface wlan1 inet manual
16 wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
17
18 iface wlan0 inet static
19 address 192.168.0.105 #Dia chi IP cua ban
20 netmask 255.255.255.0 #Thay doi
21 gateway 192.168.0.1 #Thay doi
22 #De xem cac thong so, dung lenh: ifconfig va netstat -rn
```

- Tập tin: `wpa_supplicant.conf`, mở tập tin:

```
1 $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

và thay đổi nội dung như sau:

```
1 ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
2 update_config=1
3
4 network={
5     ssid="Tam3" # Thay doi: ten Wifi, vi du: Tam3
6     psk="21019400" #Thay doi: mat khau wfii: 21019400
7     proto=WPA
8     key_mgmt=WPA-PSK
9     pairwise=TKIP
10    auth_alg=OPEN
11 }
```

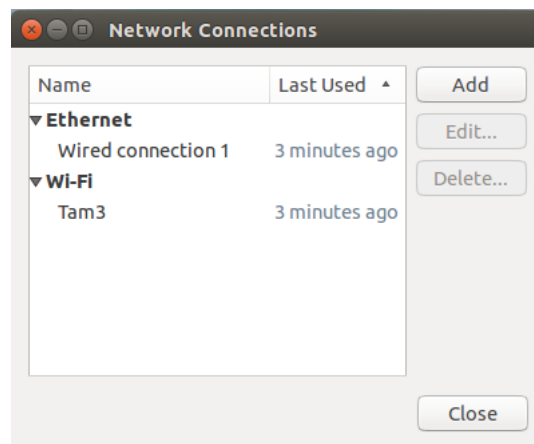
3.3 Sử dụng chung Wifi với Laptop

Ta kết nối cổng Ethernet của Pi và Laptop với nhau. Sử dụng tính năng Share Wifi trên Laptop:

3.3.1 Hệ điều hành Ubuntu

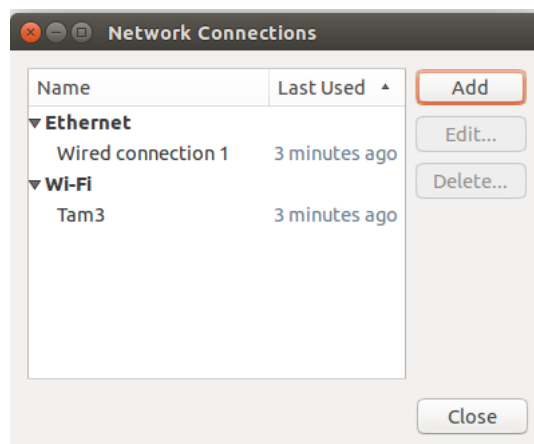
Thực hiện theo các bước sau:

- Trong thanh tìm kiếm Dash: gõ vào Network Connections, chọn Network Connections để mở lên.



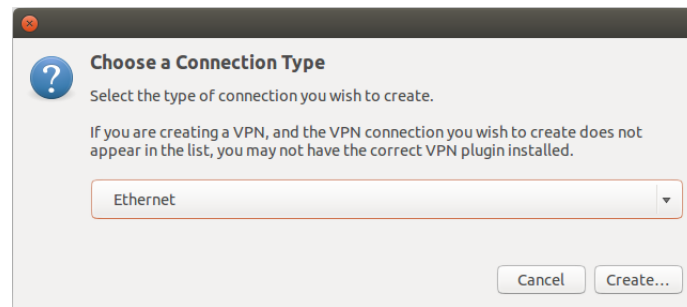
Hình 3.3: Mở Network Connections

- Chọn Add:



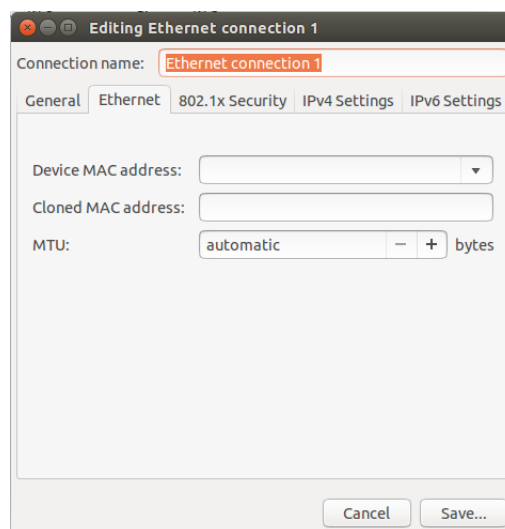
Hình 3.4: Chọn Add

- Chọn Create:



Hình 3.5: Chọn Create...

- Điền tên trong ô Connection name:

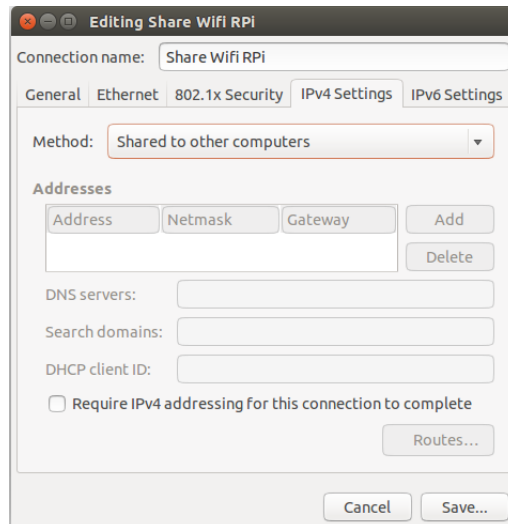


Hình 3.6: Điền tên trong ô Connection name

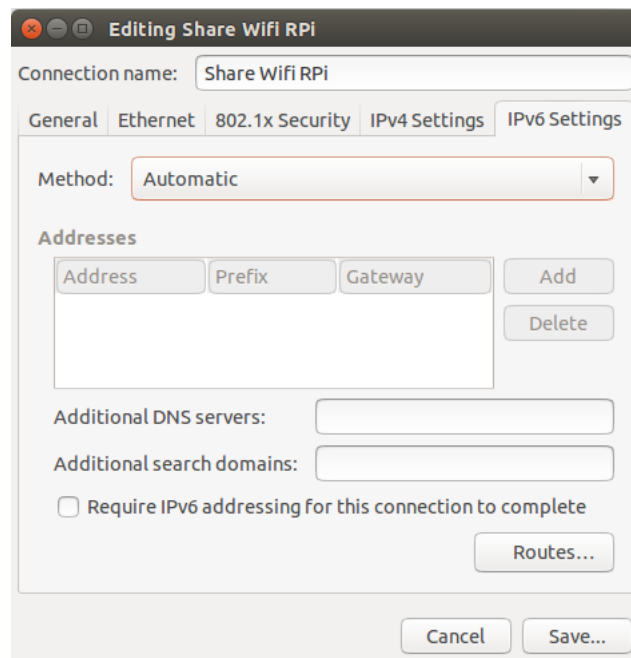
- Chọn tab IPv6 Settings, chọn Method là Automatic: hình 3.7.
- Chọn tab IPv4 Settings, chọn Method là Share to other computers: hình 3.8.
- Chọn Save rồi chọn Close.
- Mở cửa sổ lệnh Terminal gõ lệnh:

```
1 $ sudo cat /var/lib/misc/dnsmasq.leases
2 1461927547 b8:27:eb:6a:bf:9a 10.42.0.31 raspberrypi
   ff:eb:6a:bf:9a:00:01:00:01:1c:dd:60:6b:b8:27:eb:6a:bf:9a
```

- Địa chỉ của Pi lúc này là 10.42.0.31
- Truy cập qua SSH bằng lệnh:



Hình 3.7: Chọn tab IPv6 Settings, chọn Method là Automatic



Hình 3.8: Chọn tab IPv6 Settings, chọn Method là Share to other computers

```
1 $ ssh pi@10.42.0.31
```

- Nhập password đăng nhập tài khoản username là pi.

* Nếu chạy lệnh trên không được thì bạn chạy với quyền root:

```
1 $ sudo ssh pi@10.42.0.31
```

Chủ đề 4

Tự động đăng nhập Raspberry Pi sau khi khởi động

Tham khảo tại:

<http://www.raspberrypi-spy.co.uk/2015/02/how-to-autorun-a-python-script-on-raspberry-pi-boot/>

Mặc định khi Raspberry Pi khởi động, cần phải đăng nhập username và password mới sử dụng được. Với một số ứng dụng thực tế, cần tự động đăng nhập mới có thể hoạt động được.

Thực hiện theo các bước sau:

- Mở file `inittab`, dùng lệnh:

```
1 $ sudo nano /etc/inittab
```

- Tìm đến dòng dưới và thêm dấu `#` vào trước nó:

```
1 1:2345:respawn:/sbin/getty 115200 tty1
```

(tìm đến dòng có `1:2345:respawn:/sbin/getty` là được, còn các tham số phía sau tùy thuộc vào phiên bản hệ điều hành).

- Thay thế dòng trên bằng dòng sau (với `pi` là tên username):

```
1 1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1
```

- Nhấn `Ctrl - X - Y` để lưu lại nội dung thay đổi và thoát.
- Khởi động lại Pi: `sudo reboot`.

Chủ đề 5

Chạy chương trình Python trên Raspberry Pi

5.1 Python trên Raspberry Pi

Ta phân biệt theo 2 trường hợp sau:

- Chạy các lệnh không liên quan đến phần cứng là các chân GPIO, ta sẽ gõ các lệnh sau:

- Chạy ở chế độ dòng lệnh:

```
1 $ python
```

- Khi đã có sẵn một file `.py` (ví dụ: `file.py`):

```
1 $ python file.py
```

- Các lệnh liên quan đến phần cứng can thiệp vào các chân GPIO hoặc cần quyền root, ta phải chạy với quyền root: dùng `sudo python` hoặc `sudo python file.py` (cách dùng 2 lệnh này giống như trên).

5.2 Tự động chạy một chương trình python sau khi Reboot

Ở phần 5.1, ta phải thực hiện đánh lệnh thì file python mới được gọi, với nhiều ứng dụng tự động, cần tự động chạy chương trình python sau khi reboot. Ta có một số cách sau:

Ví dụ, ta cần chạy file python có tên là `myfile.py`.

5.2.1 Sử dụng cron

Tham khảo tại: <https://www.youtube.com/watch?v=8iU9TnYF0V0>
Thực hiện theo các bước sau:

- Không cần tự động đăng nhập.

- Mở file `crontab` gõ lệnh:

```
1 $ sudo crontab -e
```

- Thêm dòng sau vào cuối file:

```
1 @reboot sudo python /home/pi/myfile.py &
```

- Để chạy nhiều file cùng một lúc:

```
1 @reboot sudo python /home/pi/myfile1.py & sudo python
    /home/pi/myfile2.py &
```

Ký hiệu `&` có nghĩa là file `myfile.py` sẽ chạy nền.

Đường dẫn `/home/pi` là địa chỉ của thư mục lưu tập tin Python (Thay đổi địa chỉ này cho phù hợp với nơi tập tin Python của bạn).

```

pi@raspberrypi: ~
GNU nano 2.2.6      File: /tmp/crontab.oeI7Q0/crontab      Modified

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
@reboot sudo python myfile.py &

```

- Nhấn `Ctrl - X - Y` để lưu lại nội dung thay đổi và thoát.
- Khởi động lại Pi: `sudo reboot`.

Nói thêm về `cron`¹ `cron` là một lịch trình, được khai báo với cú pháp sau:

```
1 1 2 3 4 5 /path/to/command arg1 arg2
```

¹<https://embeddedday.com/projects/raspberry-pi/a-step-further/running-python-script-at-boot/>

Ý nghĩa của các tham số:

- 1 = Minutes (0 - 59)
- 2 = Hours (0 - 23)
- 3 = Days (0 - 31)
- 4 = Month (0 - 12)
- 5 = Day of the week (0 - 7) (Sunday is the 0 day)|

Ta có thể thay thế 1 trong 5 tham số trên bằng các tham số dưới đây:

- @reboot = Run once, at startup.
- @yearly = Run once a year
- @monthly = Run once a month
- @weekly = Run once a week
- @daily = Run once a day
- @midnight = Pretty much the same as @daily
- @hourly = Run once an hour

5.2.2 Sử dụng profile

Tham khảo tại:

<http://www.raspberrypi-spy.co.uk/2015/02/how-to-autorun-a-python-script-on-raspberry-pi-boot/>

Thực hiện theo các bước sau:

- Làm cho Pi có thể tự động đăng nhập được (xem *chủ đề 4 trang 23*).
- Mở file profile, dùng lệnh:

```
1 $ sudo nano /etc/profile
```

- Kéo xuống dòng cuối dòng, thêm nội dung sau vào file:

```
1 sudo python /home/pi/myfile.py &
```

Ký hiệu & có nghĩa là file myfile.py sẽ chạy nền.

- Nhấn Ctrl - X - Y để lưu lại nội dung thay đổi và thoát.
- Khởi động lại Pi: `sudo reboot`.

5.2.3 Sử dụng cron kết hợp với tạo file .sh

Tham khảo tại:

<http://www.instructables.com/id/Raspberry-Pi-Launch-Python-script-on-startup/?ALLSTEPS>

Thực hiện theo các bước sau:

- Tạo một .sh (ví dụ: launcher.sh):

```
1 $ nano launcher.sh
```

- Nội dung file như sau (thay đổi nội dung của ví dụ cho phù hợp):

```
1 #!/bin/sh
2 # launcher.sh
3 # navigate to home directory, then to this directory, then execute
  python script, then back home
4
5 cd /
6 cd home/pi/bbt #thư mục chứa file .py
7 sudo python bbt.py #Lệnh chạy file python
8 cd /
```

Nhấn Ctrl - X - Y để lưu và thoát.

- Làm cho file .sh trở thành file thực thi (executable):

```
1 $ chmod 755 launcher.sh
```

- Kiểm tra file .sh ta vừa tạo có thực thi được không:

```
1 $ sh launcher.sh
```

- Tạo thư mục logs trong thư mục \home\pi:

```
1 $ cd ~
2 $ mkdir logs
```

- Mở cron:

```
1 $ sudo crontab -e
```

- Thêm file .sh vào cron:

```
1 @reboot sh /home/pi/bbt/launcher.sh >/home/pi/logs/cronlog 2>&1
```

Nhấn Ctrl - X - Y để lưu và thoát.

- Khởi động lại Pi: `sudo reboot`

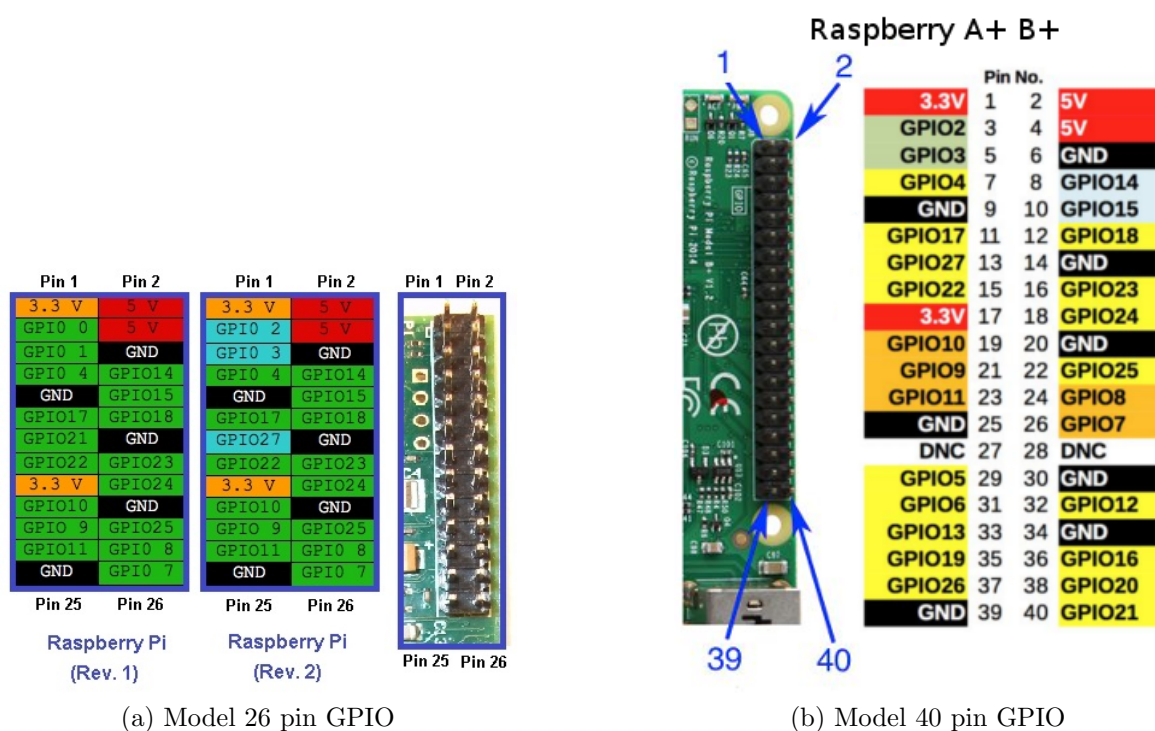
Chủ đề 6

Điều khiển phần cứng qua các chân GPIO

6.1 Giới thiệu

Ngoài chức năng như một máy tính mini (dùng để học tập, giải trí,...), Raspberry Pi còn có khả năng giao tiếp và điều khiển các phần cứng khác (cảm biến, các module mở rộng khác, động cơ, giao tiếp với các IC khác,...) thông qua các chân GPIO.

Số chân GPIO tùy thuộc vào từng phiên bản Raspberry Pi: có hai loại là 26 chân GPIO (model A, B) và 40 chân GPIO (model A+, B+, Pi 2, Pi 3, PiZero).



Hình 6.1: Số chân GPIO của các phiên bản Raspberry Pi

6.2 Mô tả chức năng của các nhóm chân

Trên Raspberry Pi sẽ có các nhóm chân sau đây (ta sẽ sử dụng cách đánh số từ 1 → 40 như trên hình 6.1):

- *Nhóm chân cấp nguồn:*
 - Nguồn 5V: gồm 2 chân – số 2 và 4.
 - Nguồn 3.3V: gồm 2 chân – số 1 và 17.
 - Chân GND: gồm 5 chân – số 6, 9, 14, 25 (Model 26 chân) hoặc gồm 8 chân – số 6, 9, 14, 25, 30, 34, 39 (Model 40 chân).
- *Nhóm chân GPIO (I/O):*
 - Model 26 chân: gồm 8 chân – số 7, 11, 12, 13, 15, 16, 18, 22.
 - Model 40 chân: gồm 17 chân – số 7, 11, 12, 13, 15, 16, 18, 22, 29, 31, 32, 33, 35, 36, 37, 38, 40.
- *Chân PWM – Pulse Width Modulation:* gồm 2 chân – chân số 12 và 13.
- *Nhóm chân giao tiếp I2C:* gồm 2 chân – số 3 và 5.
Ta có thể điều khiển nhiều thiết bị I2C chỉ với 2 chân này, mỗi thiết bị sẽ được phân biệt bởi một địa chỉ riêng của nó. Cần khai báo đúng địa chỉ của thiết bị cần điều khiển.
- *Nhóm chân giao tiếp SPI:* gồm 5 chân – số 19, 21, 23, 24, 26.
- *Nhóm chân giao tiếp UART:* gồm 2 chân – số 8 và số 10.
- *Nhóm chân giao tiếp EEPROM:* gồm 2 chân – số 27 và 28 (chỉ có ở model Pi có 40 chân GPIO).

6.3 Các cài đặt và cấu hình cần thiết để sử dụng được các chân GPIO

Trong bài viết mình sẽ chọn ngôn ngữ lập trình Python (Python 2) để điều khiển các chân GPIO của Raspberry Pi. Những dòng lệnh bên dưới, sẽ được thực hiện trên cửa sổ dòng lệnh.

6.3.1 Cài đặt thư viện RPi.GPIO

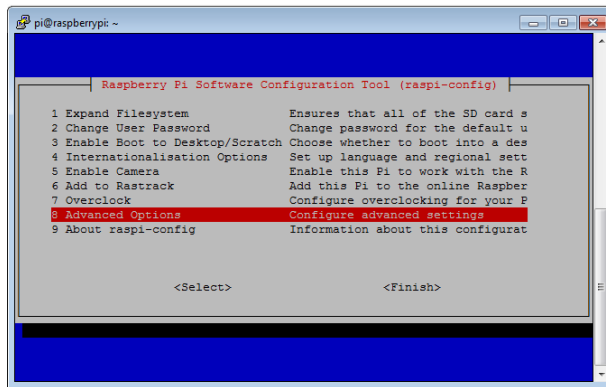
Gõ lệnh sau: `sudo apt-get install python-dev python-rpi.gpio`

6.3.2 Xác nhận cấu hình I2C

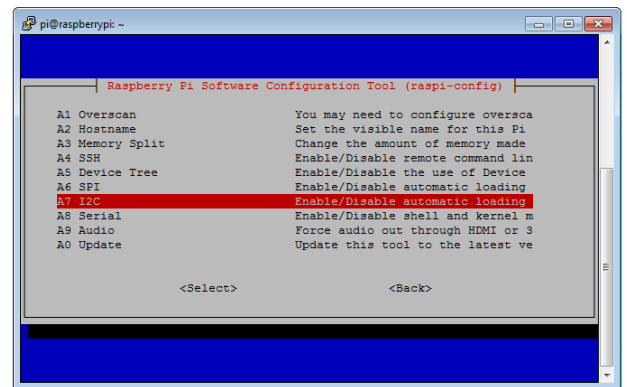
Chỉ khi nào giao tiếp I2C thì mới thực hiện *mục 6.3.2*.

Gõ lệnh sau: `sudo apt-get install python-smbus i2c-tools`

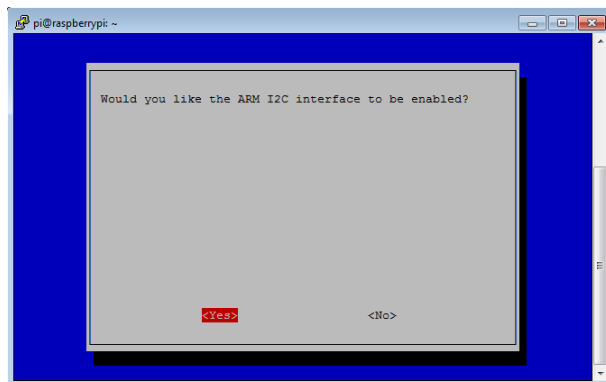
Chạy `sudo raspi-config` để Enable I2C (thường khi cài đặt hệ điều hành thì I2C ở chế độ Disable).



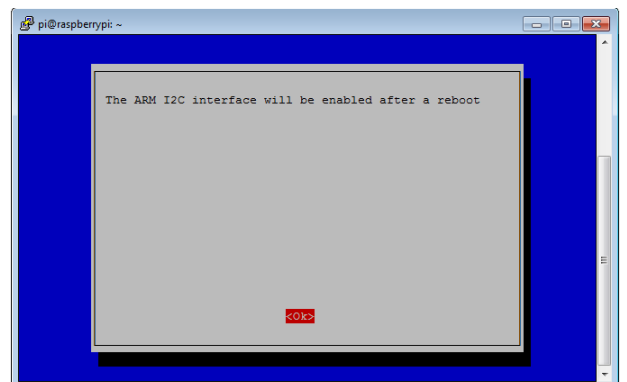
(a) Chọn 8 Advance Options rồi Enter



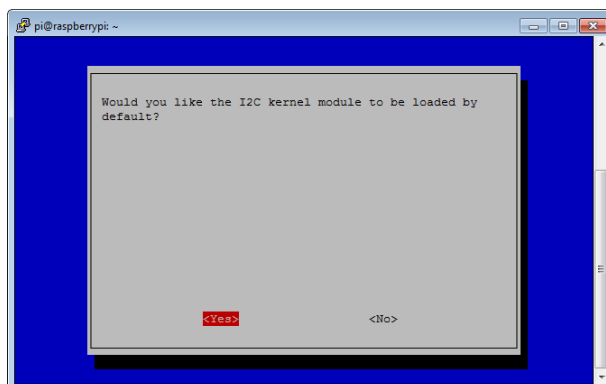
(b) Chọn A7 I2C rồi Enter



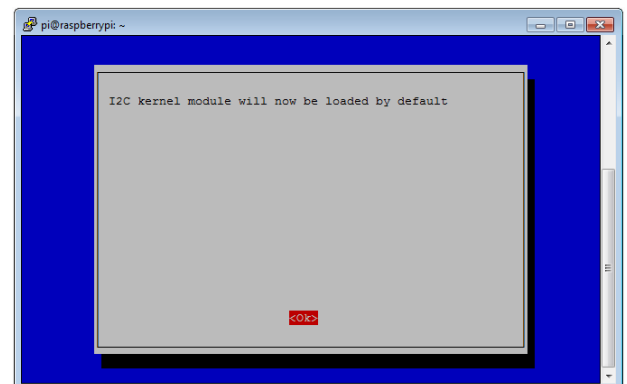
(c) Chọn Yes rồi Enter



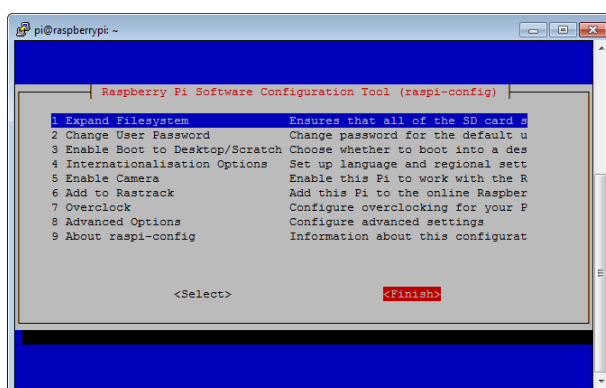
(d) Chọn OK rồi Enter



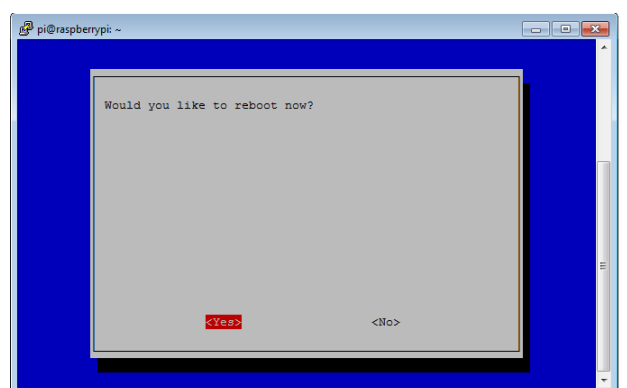
(e) Chọn Yes rồi Enter



(f) Chọn OK rồi Enter



(g) Chọn Finish rồi Enter

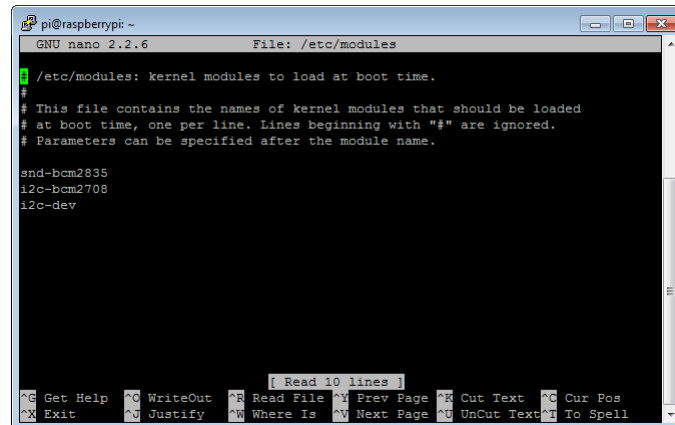


(h) Chọn Yes rồi Enter

Hình 6.2: Cách cấu hình I2C trên Raspberry Pi

Thực hiện theo các bước từ *hình 6.2a* đến *hình 6.2h* trong *hình 6.2*. Sau khi thực hiện bước cuối cùng trong *hình 6.2h*, Raspberry Pi sẽ reset lại. Khi Raspberry Pi khởi động trở lại, ta thực hiện, tiếp:

- Mở file `modules`, dùng lệnh: `sudo nano /etc/modules`
- Thêm vào file `modules` với nội dung giống như hình bên dưới: thêm vào 2 dòng `i2c-bcm2708` và `i2c-dev` (Nhấn `Ctrl + X + Y` để lưu file và thoát khỏi).



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/modules

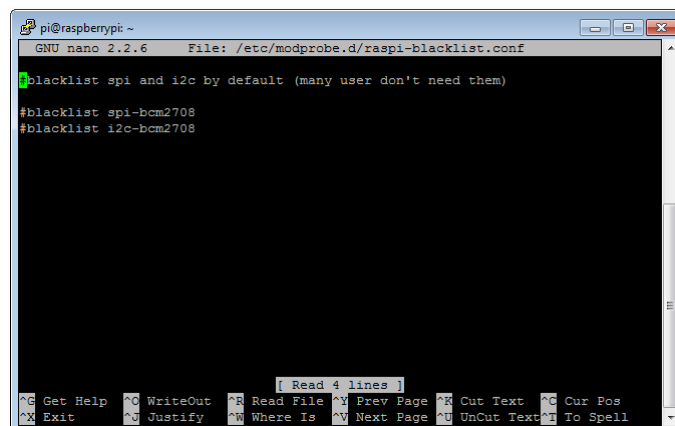
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-bcm2708
i2c-dev
```

- Mở file `raspi-blacklist.conf`, dùng lệnh:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

- Thêm dấu `#` vào trước 2 dòng: `blacklist spi-bcm2708` và `blacklist i2c-bcm2708` (Nhấn `Ctrl + X + Y` để lưu file và thoát khỏi).



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/modprobe.d/raspi-blacklist.conf

#blacklist spi and i2c by default (many user don't need them)
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

- Sau khi thực xong, chúng ta reset lại Raspberry Pi: `sudo reboot`

6.3.3 Xác nhận cấu hình SPI

Chỉ khi nào giao tiếp SPI thì mới thực hiện *mục 6.3.3*.

Thực hiện Enable SPI với lệnh `sudo raspi-config`, tương tự như Enable I2C (đến *hình 6.2b* ta chọn A6 SPI).

Khi Raspberry Pi khởi động trở lại, ta thực hiện:

- Mở file `raspi-blacklist.conf`, dùng lệnh:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

- Thêm dấu `#` vào trước 2 dòng: `blacklist spi-bcm2708` và `blacklist i2c-bcm2708` (Nhấn Ctrl + X + Y để lưu file và thoát khỏi). Nếu không có file này thì hãy tạo lại với nội dung như bên dưới.

6.4 Sử dụng các chân GPIO để điều khiển phần cứng cùng với ngôn ngữ lập trình Python

Phần này dựa vào bài viết của tác giả: Anonymous với chủ đề: *raspberrypi-gpio-python – A Python module to control the GPIO on a Raspberry Pi*¹

Hiện tại thì thư viện này chưa hỗ trợ các giao tiếp SPI, I2C, 1-wire, chúng ta chỉ sử dụng nó với các lệnh cơ bản bên dưới.

Tránh đặt điện áp lớn hơn 3.3V vào trực tiếp các chân GPIO vì sẽ làm hỏng chúng.

6.4.1 Kiểm tra chức năng của các chân GPIO

Sử dụng các chân GPIO trên Raspberry Pi phải đúng với chức năng của nó thì mới đem lại hiệu quả.

- Kiểm tra chức năng của từng chân:

```
1 import RPi.GPIO as GPIO
2
3 GPIO.setmode(GPIO.BOARD)
4 func = GPIO.gpio_function(pin)
```

- Kết quả trả về là: `GPIO.IN`, `GPIO.OUT`, `GPIO.SPI`, `GPIO.I2C`, `GPIO.HARD_PWM`, `GPIO.SERIAL`, `GPIO.UNKNOWN`

¹<https://sourceforge.net/p/raspberrypi-gpio-python/wiki/Examples/>

6.4.2 Các lệnh cơ bản trong thư viện RPi.GPIO

- Khai báo thư viện RPi.GPIO:

```
1 import RPi.GPIO as GPIO #Dùng ten GPIO thay cho RPi.GPIO
```

Kiểm tra thông tin về Pi và chân GPIO:

```
1 GPIO.RPI_INFO
2 #Ket qua tra ve: P1_REVISION; RAM; REVISION; TYPE; PROCESSOR;
  MANUFACTURER
3 #Vi du: {'P1_REVISION': 3, 'RAM': '512M', 'REVISION': '0010',
  'TYPE': 'Model B+', 'PROCESSOR': 'BCM2835', 'MANUFACTURER':
  'Unknown'}
4 #Ta co the dung lenh: GPIO.RPI_INFO[Tham so] de xem mot thong so
  quan tam
5
6 #Dung lenh: GPIO.RPI_REVISION de xem phien ban chan GPIO
7 GPIO.VERSION #Kiem tra phien ban GPIO
8 #Vi du: '0.6.2'
```

- Xác định cách khai báo số chân là GPIO.BOARD hay GPIO.BCM:

```
1 GPIO.setmode(GPIO.BOARD) #Danh so chan theo so tren Board tu so 1
  den so 40
2 GPIO.setmode(GPIO.BCM) #Danh so chan theo ten GPIO, vi du GPIO27,
  GIP014,...
```

Để kiểm tra xem bạn đang sử dụng cách khai báo nào trong 2 cách khai báo BOARD hoặc BCM, dùng lệnh sau:

```
1 mode = GPIO.getmode() #Xem khai bao BOARD hoac BCM
2 print mode
3 #mode = -1: chua khai bao
4 #mode = 10: khai bao la BOARD
5 #mode = 11: khai bao la BCM
```

- Khi các chân GPIO đã được sử dụng trước đó (chưa được cleanup) thì sử dụng lại chương trình sẽ thông báo các cảnh báo, ta sử dụng lệnh sau để vô hiệu cảnh báo:

```
1 GPIO.setwarnings(False) #Bo qua cac canh bao ve GIPO
```

- Khai báo chân GPIO cần điều khiển là chân input hay chân output:

```
1 #pin la chan GPIO can dieu khien
2 GPIO.setup(pin, GPIO.IN) #Khai bao pin chan INPUT
3 GPIO.setup(pin, GPIO.OUT) #Khai bao pin chan OUTPUT
4
5
6 #Khi can khai bao nhieu chan INPUT va OUTPUT
```

```

7 pin_input = [pin1, pin2, pin2] #Danh sach cac chan INPUT
8 pin_output = [pin4, pin4, pin6] #Danh sach cac chan OUTPUT
9
10 GPIO.setup(pin_input, GPIO.OUT) #Khai bao nhieu chan la INPUT
11 GPIO.setup(pin_output, GPIO.OUT) #Khai bao nhieu chan la OUTPUT

```

- Với một số ứng dụng ta cần *mắc điện trở treo (lên nguồn hoặc nối mass)*: có thể làm việc này bằng phần cứng hoặc bằng phần mềm. Ở đây ta sử dụng phần mềm.

```

1 #pin1, pin2 la chan GPIO can dieu khien
2
3 #Khai bao pin chan INPUT, co dien tro mac len nguon - 3.3V
4 GPIO.setup(pin1, GPIO.IN, pull_up_down = GPIO.PUD_UP)
5
6 #Khai bao pin chan INPUP, co dien tro mac xuong mass - 0V
7 GPIO.setup(pin2, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

```

- Đọc tín hiệu từ chân INPUT:

```

1 #pin la chan GPIO can dieu khien
2 read_input = GPIO.input(pin) #Doc tin hieu cua chan INPUT
3 #read_input la 0 hoac 1 hoac GPIO.LOW hoac GPIO.HIGH hoac False hoac True

```

- Xuất tín hiệu ra chân GPIO là chân OUTPUT:

```

1 #pin la chan OUTPUT
2
3 #Co 3 cach xuat chan GPIO pin ra muc cao
4 GPIO.output(pin,1)
5 GPIO.output(pin,True)
6 GPIO.output(pin,GPIO.HIGH)
7
8 #Co 3 cach xuat chan GPIO pin ra muc thap
9 GPIO.output(pin,0)
10 GPIO.output(pin,False)
11 GPIO.output(pin,GPIO.LOW)
12
13 #Khi can xuat tin hieu ra nhieu chan
14 pin_output = [pin1, pin2, pin3] #Danh sach cac chan OUTPUT
15 GPIO.output(pin_output,0) #Cac pin o muc thap,
16 GPIO.output(pin_output,0) #Cac pin o cao
17 #Co the su dung tham so: 0, 1 hoac False, True hoac GPIO.LOW,
    GPIO.HIGH
18
19 #Ham Output va Input ket hop voi nhau
20 #Doc gia tin hieu tu 1 chan va xuat no ra chinh no pin
21 GPIO.output(pin, not GPIO.input(pin))

```

- Kết thúc quá trình làm việc với các chân GPIO, có 2 tùy chọn:

```

1 GPIO.cleanup() #clear tat ca cac chan GPIO
2 GPIO.cleanup([pin_1, pin_2]) #Chi clear mot so chan GPIO

```

6.4.3 Ngắt và phát hiện tín hiệu cạnh

Ngắt là bất chương trình dừng công việc đang thực hiện để thực hiện một công việc khác.

Tín hiệu cạnh là tín hiệu số biến đổi từ mức thấp lên mức cao ($0 \rightarrow 1$) hoặc từ mức cao xuống mức thấp ($1 \rightarrow 0$).

- Hàm `wait_for_edge()`: thực hiện chương trình khi phát hiện được tín hiệu cạnh.

```

1 #Phat hien tin hieu canh o chan GPIO pin
2 GPIO.wait_for_edge(pin, GPIO.RISING) #RISING chuyen tu 0 len 1
3 GPIO.wait_for_edge(pin, GPIO.FALLING) #FALLING chuyen tu 1 xuong 0
4 GPIO.wait_for_edge(pin, GPIO.BOTH) #BOTH chi can co tin hieu canh
5 #GPIO.BOTH gom GPIO.RISING va GPIO.FALLING
6
7 #Khi can doi tin hieu canh trong thoi gian nhap dinh
8 #Doi tin hieu canh (RISING,FALLING, BOTH) trong thoi gian timeout = t
9 #Don vi cua t la ms, vi du: timeout = 5000 (t = 5s)
10 #Neu qua thoi gian timeout ma khong phat hien canh thi ket qua la
    None
11
12 #Doi canh chuyen tu 0 len 1 voi thoi gian t (ms)
13 signal_edge = GPIO.wait_for_edge(pin, GPIO.RISING, timeout=t)
14
15 #Doi canh chuyen tu 1 len 0 voi thoi gian t (ms)
16 signal_edge = GPIO.wait_for_edge(pin, GPIO.FALLING, timeout=t)
17
18 #Doi tin hieu canh voi thoi gian t (ms)
19 signal_edge = GPIO.wait_for_edge(pin, GPIO.BOTH, timeout=t)

```

- Hàm `event_detected()`: khi chương trình đang thực hiện một công việc nào đó, nếu có tín hiệu cạnh thì nó sẽ nhảy sang một việc khác.

```

1 GPIO.add_event_detect(pin, GPIO.RISING) #FALLING, BOTH
2 #Phat hien tin hieu canh tu 0 len 1
3 #
4 #Cong viec dan thuc hien vi du: trong vong lap for, while,...
5 #
6 if GPIO.event_detected(channel):
7     print "Thuc hien chuong trinh o phan nay"

```

- Dùng chức năng phát hiện cạnh thì nhảy sang *chương trình ngắt* được định nghĩa trước để thực hiện câu lệnh.

```

1 def chuong_trinh_ngat_1():
2     #
3     #Noi dung can thuc hien khi co ngat
4     #
5
6 def chuong_trinh_ngat_2():
7     #
8     #Noi dung can thuc hien khi co ngat
9     #
10 GPIO.add_event_detect(pin, GPIO.RISING) #FALLING, BOTH
11 GPIO.add_event_callback(pin, chuong_trinh_ngat_1())
12 GPIO.add_event_callback(pin, chuong_trinh_ngat_2())
13
14 #Them vao tham so bouncetime = t(ms) de chong nhieu
15 GPIO.add_event_callback(pin, chuong_trinh_ngat_1, bouncetime=t)
16 GPIO.add_event_callback(pin, chuong_trinh_ngat_2, bouncetime=t)

```

- Hàm `remove_event_detect()`: không sử dụng chức của hàm `event_detected()` nữa.

```

1 GPIO.remove_event_detect(pin) #pin GPIO duoc su dung voi ngat truoc
    do

```

6.4.4 Sử dụng chân GPIO với chức năng PWM

Ta thực hiện theo các bước:

- Khởi tạo một khối PWM:

```

1 p = GPIO.PWM(channel, freq)
2 #channel: chan so 12 va so 13 hoac GPIO18 va GPIO27
3 #freq: tan so xung vuong

```

- Bắt đầu với PWM:

```

1 p.start(duty_cycle)
2 #duty_cycle = T_on/T = T_on/(T_on T_off) = 0 - 100%

```

- Lệnh thay đổi tần số và chu kỳ:

```

1 #Thay doi tan so
2 p.ChangeFrequency(freq) #freq: la tan so moi
3
4 #Thay doi phan tram T_on
5 p.ChangeDutyCycle(duty_cycle) #duty_cycle: phan tram T_on moi

```

- Dừng khối PWM:

```

1 p.stop() #Dung khoi PWM voi ten bien la p

```
