

Mục lục

1	Điều khiển và sao chép dữ liệu với Raspberry Pi	3
1.1	Điều khiển bằng cách kết nối trực tiếp với màn hình, bàn phím và chuột	3
1.1.1	Màn hình hỗ trợ cổng HDMI	3
1.1.2	Màn hình hỗ trợ cổng VGA	3
1.2	Điều khiển bằng giao tiếp nối tiếp thông qua cổng RS232	4
1.3	Điều khiển bằng giao tiếp nối tiếp thông qua cáp USB to COM PL2303	4
1.4	Điều khiển từ xa khi Raspberry Pi có kết nối mạng	5
2	Mạng Internet với Raspberry Pi	7
2.1	Kết nối internet với dây mạng LAN	7
2.2	Kết nối internet với USB Wifi	7
2.2.1	Cài đặt Drive	7
2.2.2	Cài đặt địa chỉ IP tĩnh	8
2.3	Sử dụng chung Wifi với Laptop	9
2.3.1	Hệ điều hành Ubuntu	9
3	Tự động đăng nhập Raspberry Pi sau khi khởi động	13
4	Chạy chương trình Python trên Raspberry Pi	14
4.1	Python trên Raspberry Pi	14
4.2	Tự động chạy một chương trình python sau khi Reboot	14
4.2.1	Sử dụng cron	14
4.2.2	Sử dụng profile	16
4.2.3	Sử dụng cron kết hợp với tạo file .sh	17
5	Điều khiển phần cứng qua các chân GPIO	18
5.1	Giới thiệu	18
5.2	Mô tả chức năng của các nhóm chân	18
5.3	Các cài đặt và cấu hình cần thiết để sử dụng được các chân GPIO	19
5.3.1	Cài đặt thư viện RPi.GPIO	19
5.3.2	Xác nhận cấu hình I2C	20
5.3.3	Xác nhận cấu hình SPI	20
5.4	Sử dụng các chân GPIO để điều khiển phần cứng cùng với ngôn ngữ lập trình Python	20
5.4.1	Kiểm tra chức năng của các chân GPIO	22
5.4.2	Các lệnh cơ bản trong thư viện RPi.GPIO	23

5.4.3	Ngắt và phát hiện tín hiệu cạnh	25
5.4.4	Sử dụng chân GPIO với chức năng PWM	26

Chủ đề 1

Điều khiển và sao chép dữ liệu với Raspberry Pi

1.1 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.

1.1.1 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:

1.1.2 Màn hình hỗ trợ cổng VGA

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



Hình 1.1: Cáp chuyển đổi từ cổng HDMI sang cổng VGA

1.2 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 1.2: Module RS232 to TTL và Cáp USB to COM

- Thực hiện kết nối Pi và module RS232 như sau:

Pi	RS232
3.3V	VCC
TX	TX
RX	RX
GND	GND

- 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.
- * Ta có thể dùng Putty (trên hệ điều hành Window) để điều khiển: chọn **Serial**, điền vào khung **Serial line** tên cổng (ví dụ: COM1, COM2,...), trong khung **Speed** điền tốc độ là 115200. Nhập username và password để đăng nhập.

1.3 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 1.3: 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 1.2 trang 4).

1.4 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).
- Chọn chương trình phù hợp để điều khiển Raspberry Pi:
 - Với SSH: không hỗ trợ giao diện đồ họa.
 - Với Remote Desktop (Pi cần cài đặt: xrdp, dùng lệnh: `sudo apt-get install xrdp`), VNC: có hỗ trợ giao diện đồ họa.

- 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**.
 - * 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.

Chủ đề 2

Mạng Internet với Raspberry Pi

2.1 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.

2.2 Kết nối internet với USB Wifi

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

2.2.1 Cài đặt Drive

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 drive:

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

- Cài đặt drive, 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-2015yyzz.tar.gz là drive phù hợp với phiên bản hệ điều hành của bạn.

Bạn có thể tải drive 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 Drive mới cho USB Wifi.

2.2.2 Cài đặt địa chỉ IP tĩnh

Tham khảo tại địa chỉ:

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

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 #Địa chỉ IP của bạn
20 netmask 255.255.255.0 #Thay đổi
21 gateway 192.168.0.1 #Thay đổi
22 #Để xem các thông số, dùng lệnh: ifconfig và 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 }
```

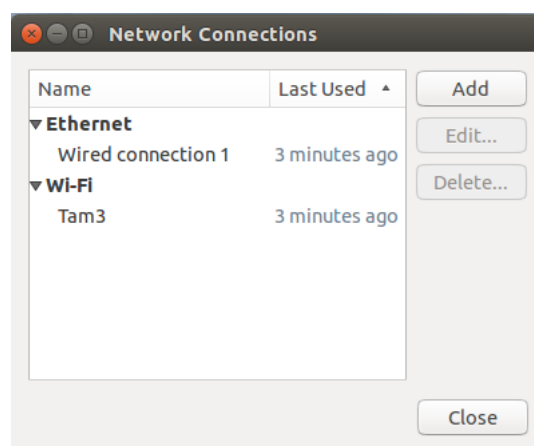
2.3 Sử dụng chung Wifi với Laptop

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

2.3.1 Hệ điều hành Ubuntu

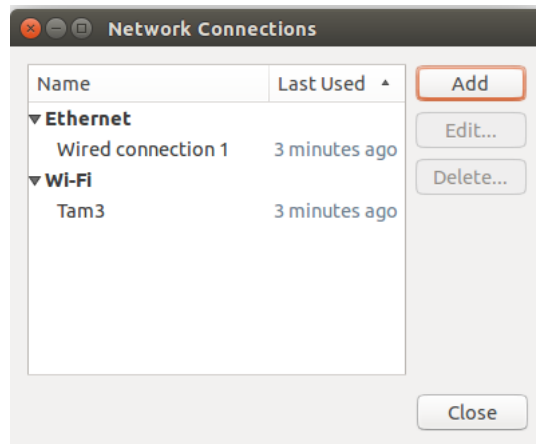
Thực hiện theo các bước sau:

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



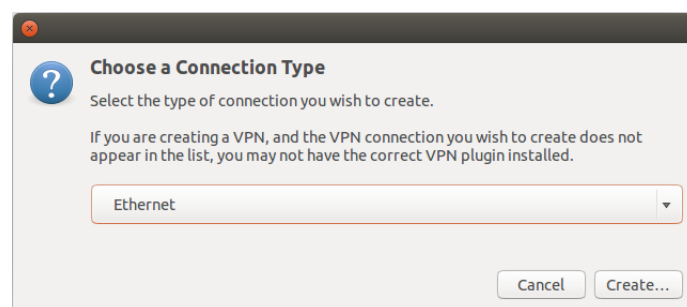
Hình 2.1: Mở Network Connections

- Chọn `Add`:



Hình 2.2: Chọn Add

- Chọn Create:



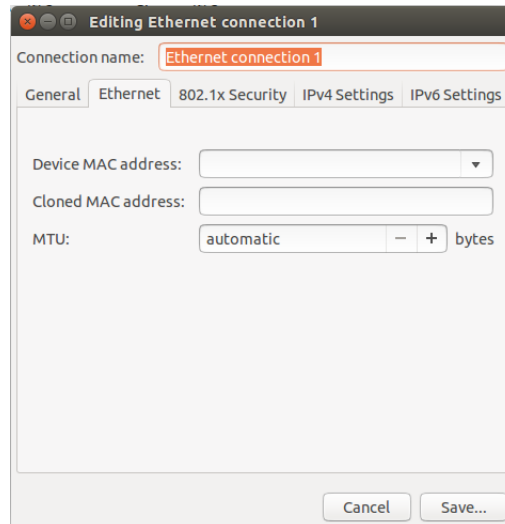
Hình 2.3: Chọn Create...

- Điền tên trong ô Connection name:
- Chọn tab IPv6 Settings, chọn Method là Automatic:
- Chọn tab IPv4 Settings, chọn Method là Share to other computers:
- Chọn Save rồi chọn Close.
- Mở cửa sổ lệnh Terminal gõ lệnh:

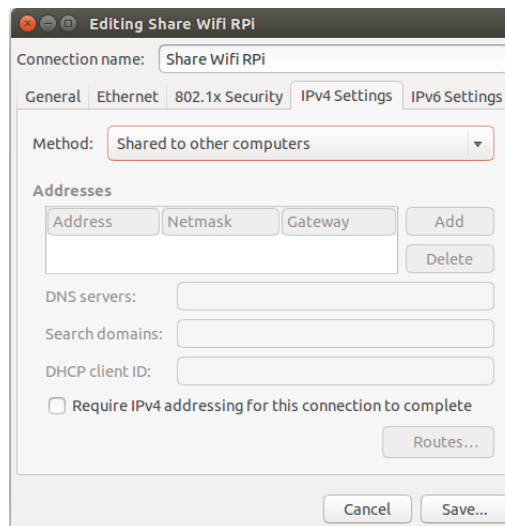
```
1 $ sudo cat /var/lib/misc/dnsmasq.leases
2 1461927547 b8:27:eb:6a:bf:9a 10.42.0.31 raspberrypi
   ff:eb:6a:bf:9a:00:01:00:01:1c:dd:60:6b:b8:27:eb:6a:bf:9a
```

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

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

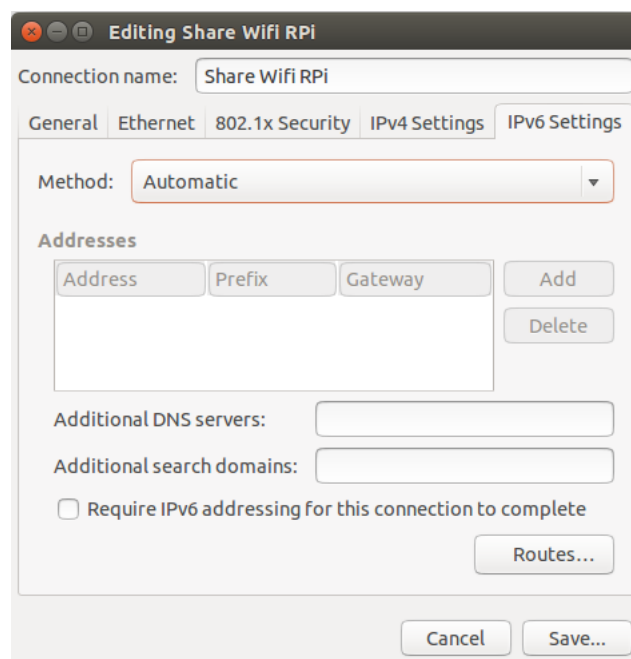


Hình 2.4: Điền tên trong ô Connection name



Hình 2.5: Chọn tab IPv6 Settings, chọn Method là Automatic

- Nhập password đăng nhập tài khoản username là pi.



Hình 2.6: Chọn tab IPv6 Settings, chọn Method là Share to orther computers

Chủ đề 3

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 4

Chạy chương trình Python trên Raspberry Pi

4.1 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).

4.2 Tự động chạy một chương trình python sau khi Reboot

Ở phần 4.1, 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`.

4.2.1 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.
- Copy file myfile.py đến thư mục \home\pi (dùng lệnh cp).
- Mở file crontab gõ lệnh:

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

- Thêm dòng sau vào cuối file:

```
1 @reboot sudo python myfile.py &
```

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

```

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 cron¹ cron là một lịch trình, được khai báo với cú pháp sau:

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

Ý nghĩa của các tham số:

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

- 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

4.2.2 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 3 trang 13*).
- 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`.

4.2.3 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 5

Điều khiển phần cứng qua các chân GPIO

5.1 Giới thiệu

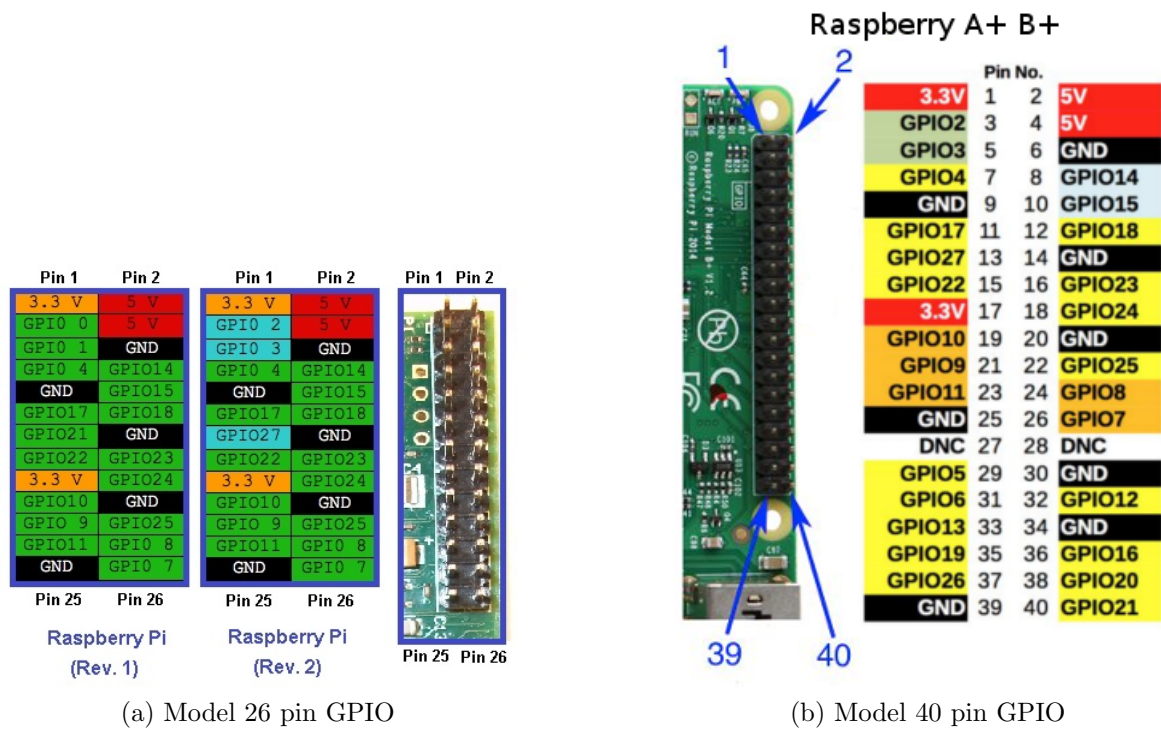
Ngoài chức năng như một máy tính mini (dùng để học tập, giải trí,...), Raspberry Pi còn có khả năng giao tiếp và điều khiển các phần cứng khác (cảm biến, các module mở rộng khác, động cơ, giao tiếp với các IC khác,...) thông qua các chân GPIO.

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).

5.2 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 \rightarrow 40 như trên hình 5.1):

- *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.



Hình 5.1: Số chân GPIO của các phiên bản Raspberry Pi

- *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).

5.3 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 (Phython 2) để điều khiển các chân GPIO của Raspberry Pi. Những dòng lệnh bên dưới, sẽ được thực hiện trên cửa sổ dòng lệnh.

5.3.1 Cài đặt thư viện RPi.GPIO

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

5.3.2 Xác nhận cấu hình I2C

Chỉ khi nào giao tiếp I2C thì mới thực hiện *mục 5.3.2*.

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).

Thực hiện theo các bước từ *hình 5.2a* đến *hình 5.2h* trong *hình 5.2*. Sau khi thực hiện bước cuối cùng trong *hình 5.2h*, 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).
- 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).
- Sau khi thực xong, chúng ta reset lại Raspberry Pi: `sudo reboot`

5.3.3 Xác nhận cấu hình SPI

Chỉ khi nào giao tiếp SPI thì mới thực hiện *mục 5.3.3*.

Thực hiện Enable SPI với lệnh `sudo raspi-config`, tương tự như Enable I2C (đến *hình 5.2b* ta chọn A6 SPI).

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

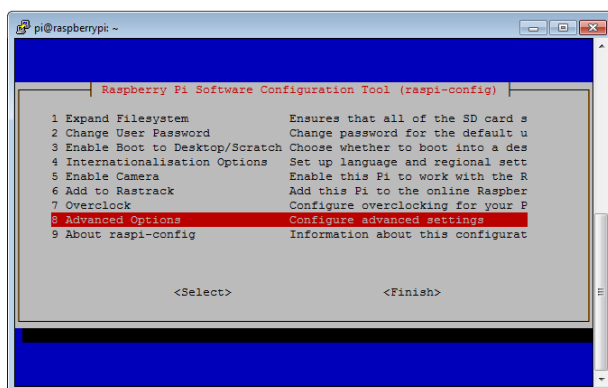
- Mở file `raspi-blacklist.conf`, dùng lệnh:

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

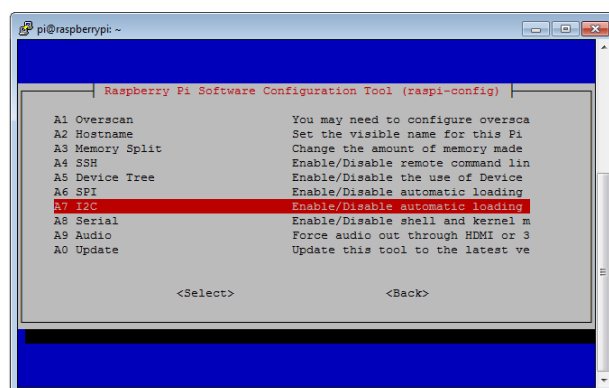
- Thêm dấu `#` vào trước 2 dòng: `blacklist spi-bcm2708` và `blacklist i2c-bcm2708` (Nhấn Ctrl + X + Y để lưu file và thoát khỏi). Nếu không có file này thì hãy tạo lại với nội dung như bên dưới.

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

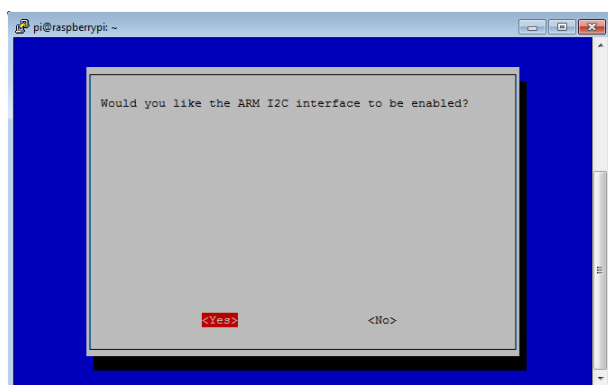
Phần này dựa vào bài viết của tác giả: Anonymous với chủ đề:



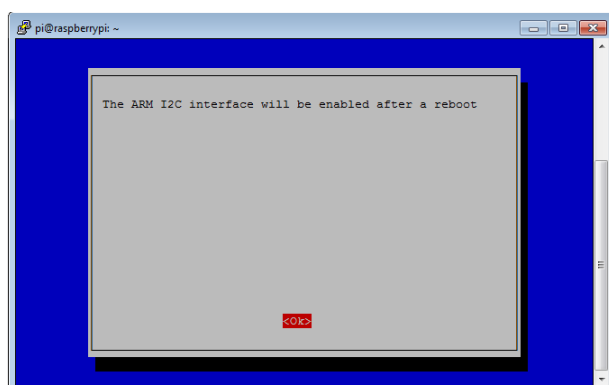
(a) Chọn 8 Advance Options rồi Enter



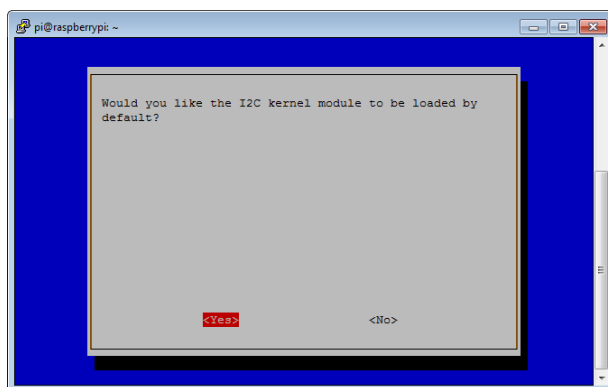
(b) Chọn A7 I2C rồi Enter



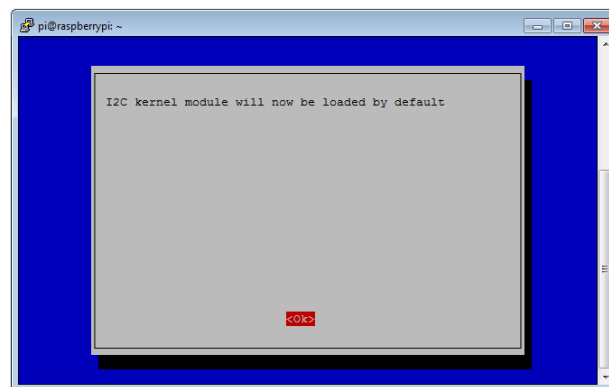
(c) Chọn Yes rồi Enter



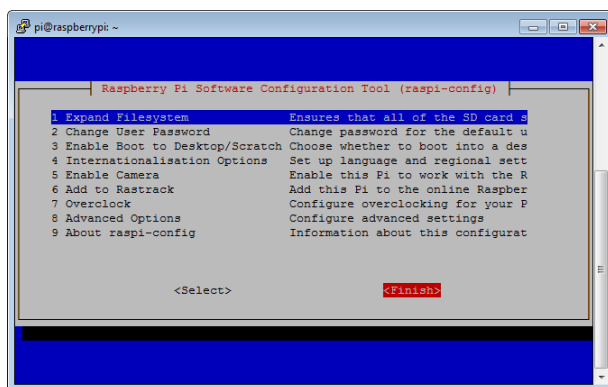
(d) Chọn OK rồi Enter



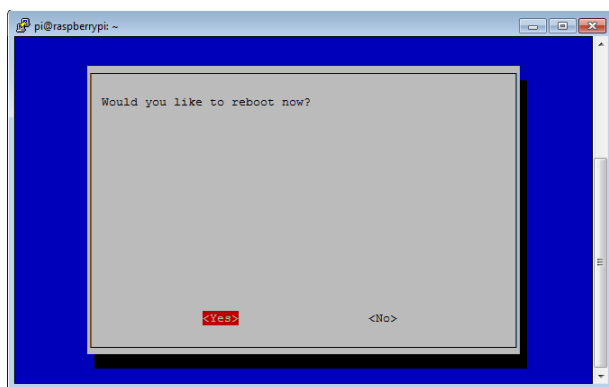
(e) Chọn Yes rồi Enter



(f) Chọn OK rồi Enter

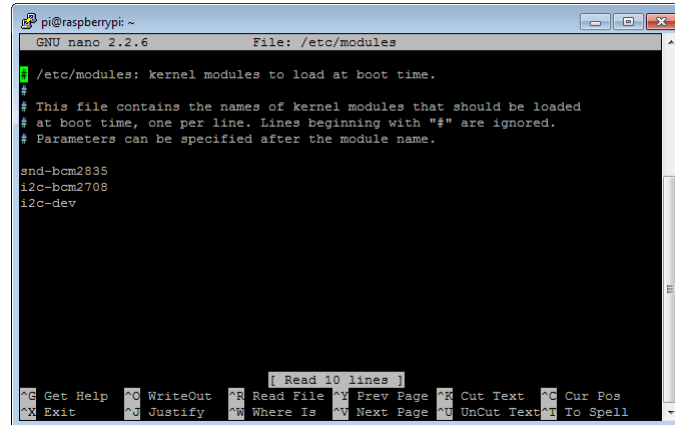


(g) Chọn Finish rồi Enter



(h) Chọn Yes rồi Enter

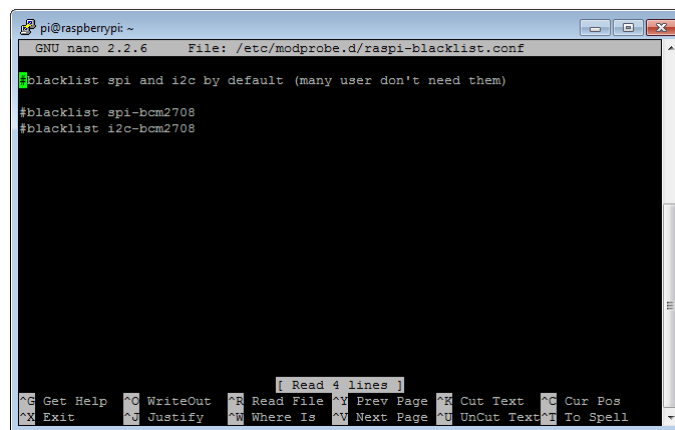
Hình 5.2: Cách cấu hình I2C trên Raspberry Pi



```
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
```



```
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
```

raspberry-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.

5.4.1 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)
```

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

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

5.4.2 Các lệnh cơ bản trong thư viện RPi.GPIO

- Khai báo thư viện RPi.GPIO:

```
1 import RPi.GPIO as GPIO #Dung 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,
  GPIO14,...
```

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 GPIO
```

- 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 xuất nó ra chính nó 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 tất cả các chân GPIO
2 GPIO.cleanup([pin_1, pin_2]) #Chỉ clear một số chân GPIO

```

5.4.3 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 chuyển từ 0 lên 1
3 GPIO.wait_for_edge(pin, GPIO.FALLING) #FALLING chuyển từ 1 xuống 0
4 GPIO.wait_for_edge(pin, GPIO.BOTH) #BOTH chỉ cần có tín hiệu cạnh
5 #GPIO.BOTH gồm GPIO.RISING và GPIO.FALLING
6
7 #Khi cần đợi tín hiệu cạnh trong thời gian nhất định
8 #Đợi tín hiệu cạnh (RISING, FALLING, BOTH) trong thời gian timeout = t
9 #Đơn vị của t là ms, ví dụ: timeout = 5000 (t = 5s)
10 #Nếu quá thời gian timeout mà không phát hiện cạnh thì kết quả là
    None
11
12 #Đợi cạnh chuyển từ 0 lên 1 với thời gian t (ms)
13 signal_edge = GPIO.wait_for_edge(pin, GPIO.RISING, timeout=t)
14
15 #Đợi cạnh chuyển từ 1 lên 0 với thời gian t (ms)
16 signal_edge = GPIO.wait_for_edge(pin, GPIO.FALLING, timeout=t)
17
18 #Đợi tín hiệu cạnh với thời 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 *chuong_trinh_ngat* 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

```

5.4.4 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
```
