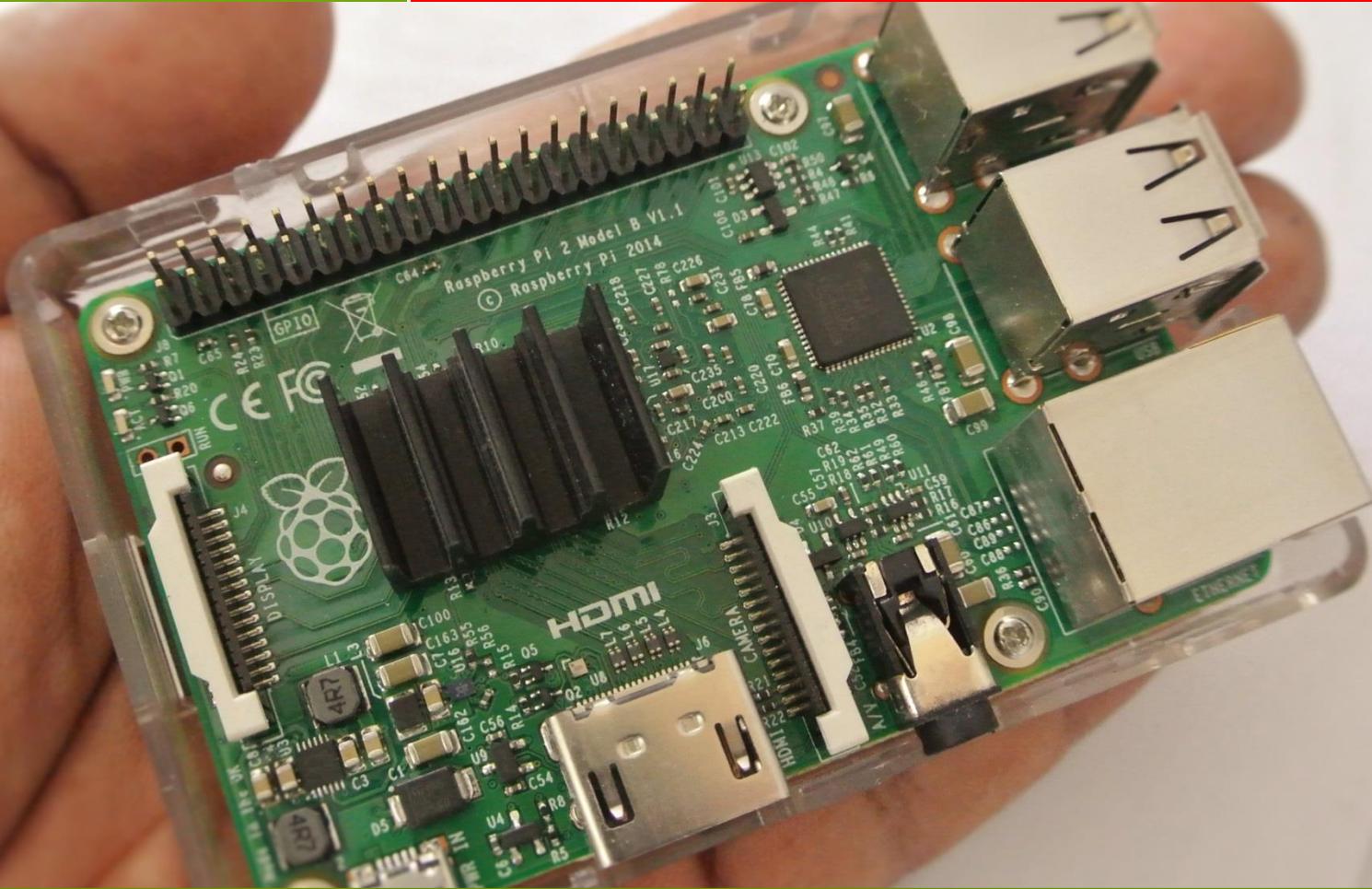


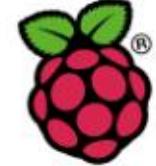
รีวิว
รีวิวจัดและใช้งาน

บอร์ดคอมพิวเตอร์ 32 บิต



Raspberry Pi 2

Raspberry Pi2 Spec



CPU : ARM V7 quad-core 900MHz 512kb Cache

Dual Core GPU

40 ขา GPIO

ไฟเลี้ยง 5V/1.5A microUSB

RAM : 1GByte

USB 2.0 4 ช่อง

แรงกว่าเดิม 6 เท่า

JACK AV 3.5mm

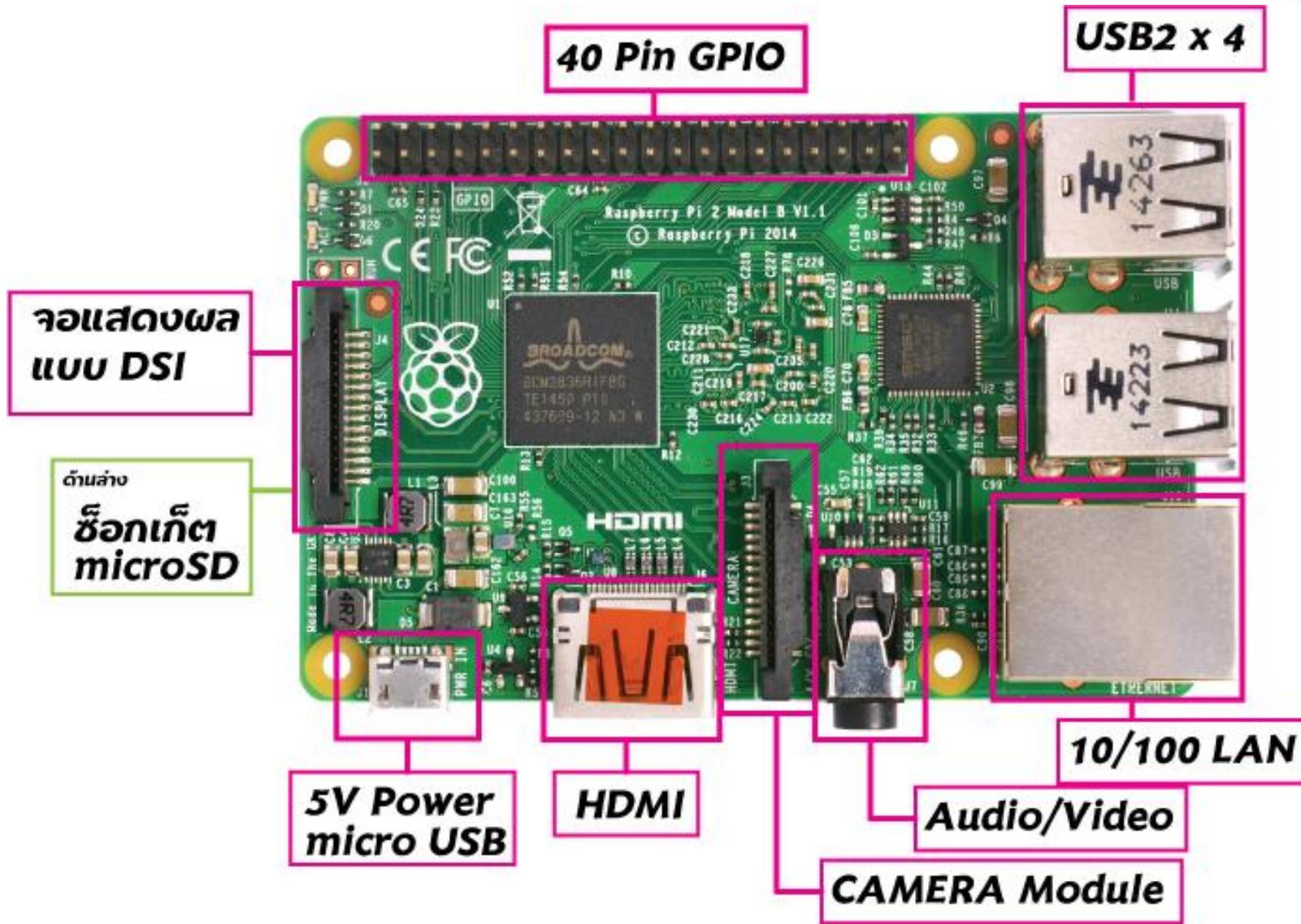
10/100 ETERNET

จุดต่อ HDMI

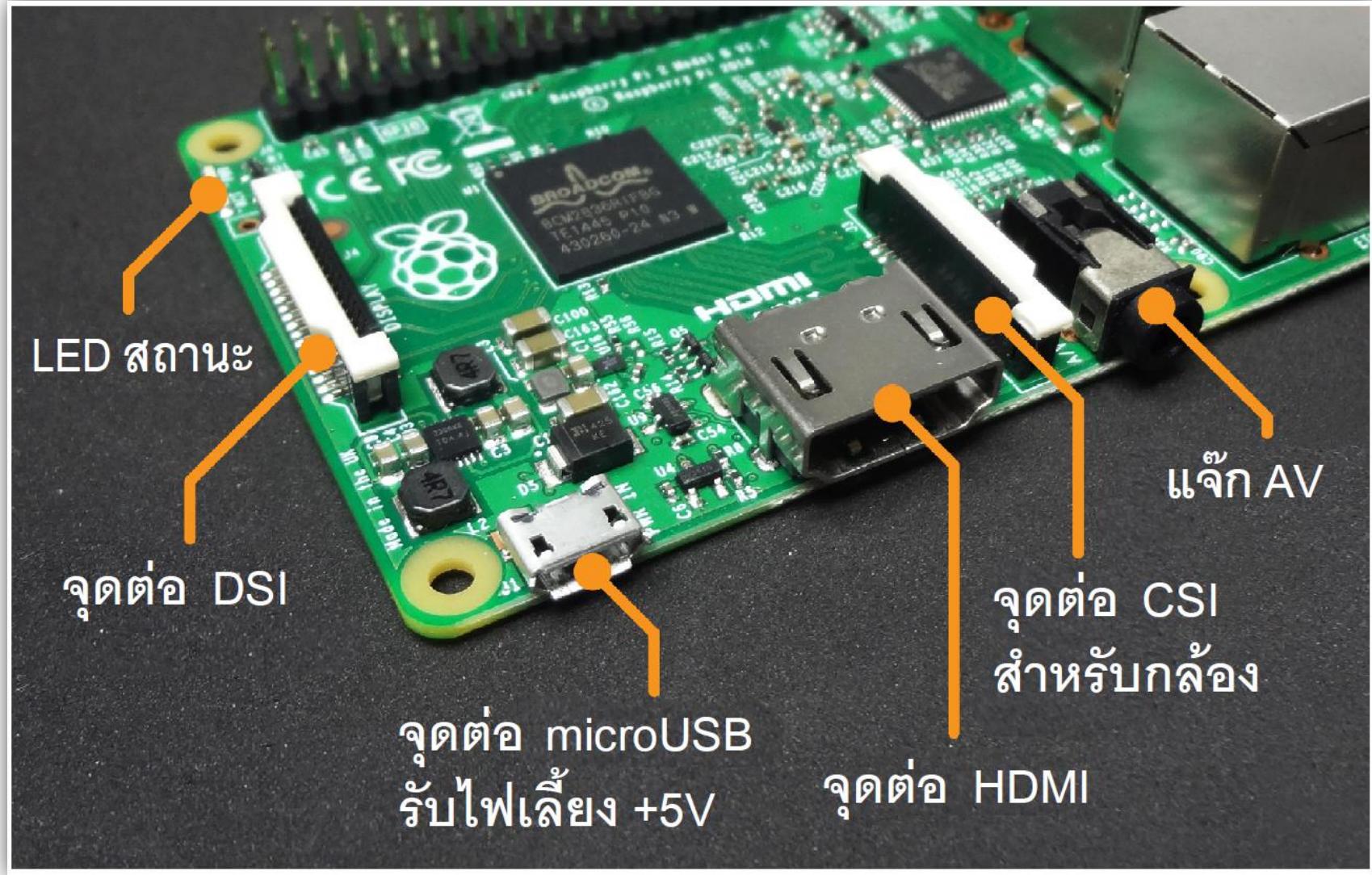
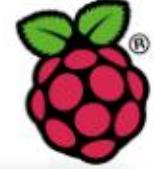
จุดต่อ DSI ต่อจอ LCD

จุดต่อ CSI ต่อกล้อง

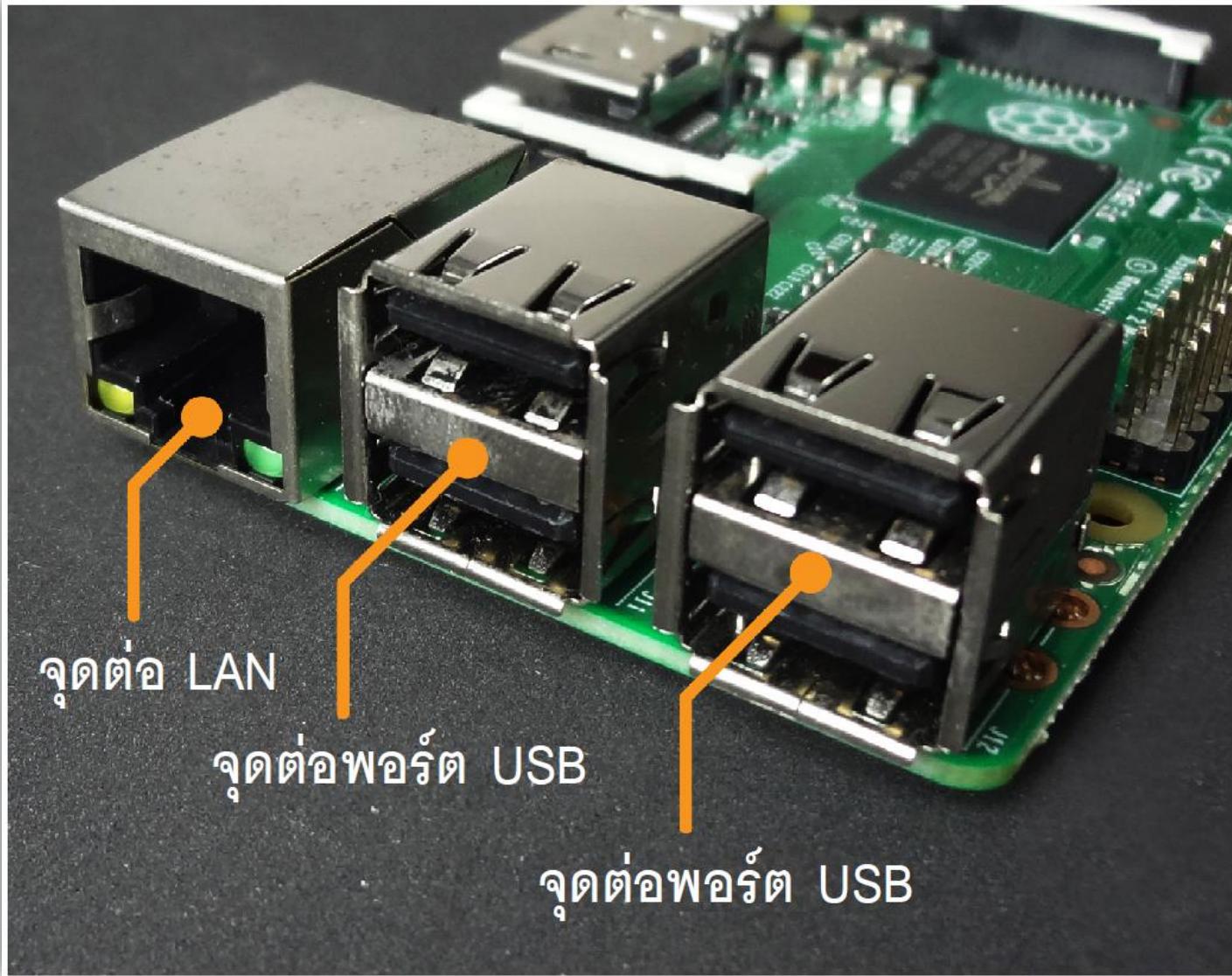
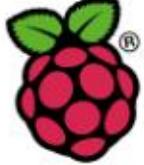
ឧបតម្លៃកម្មពីរបាយការណ៍ Raspberry Pi2



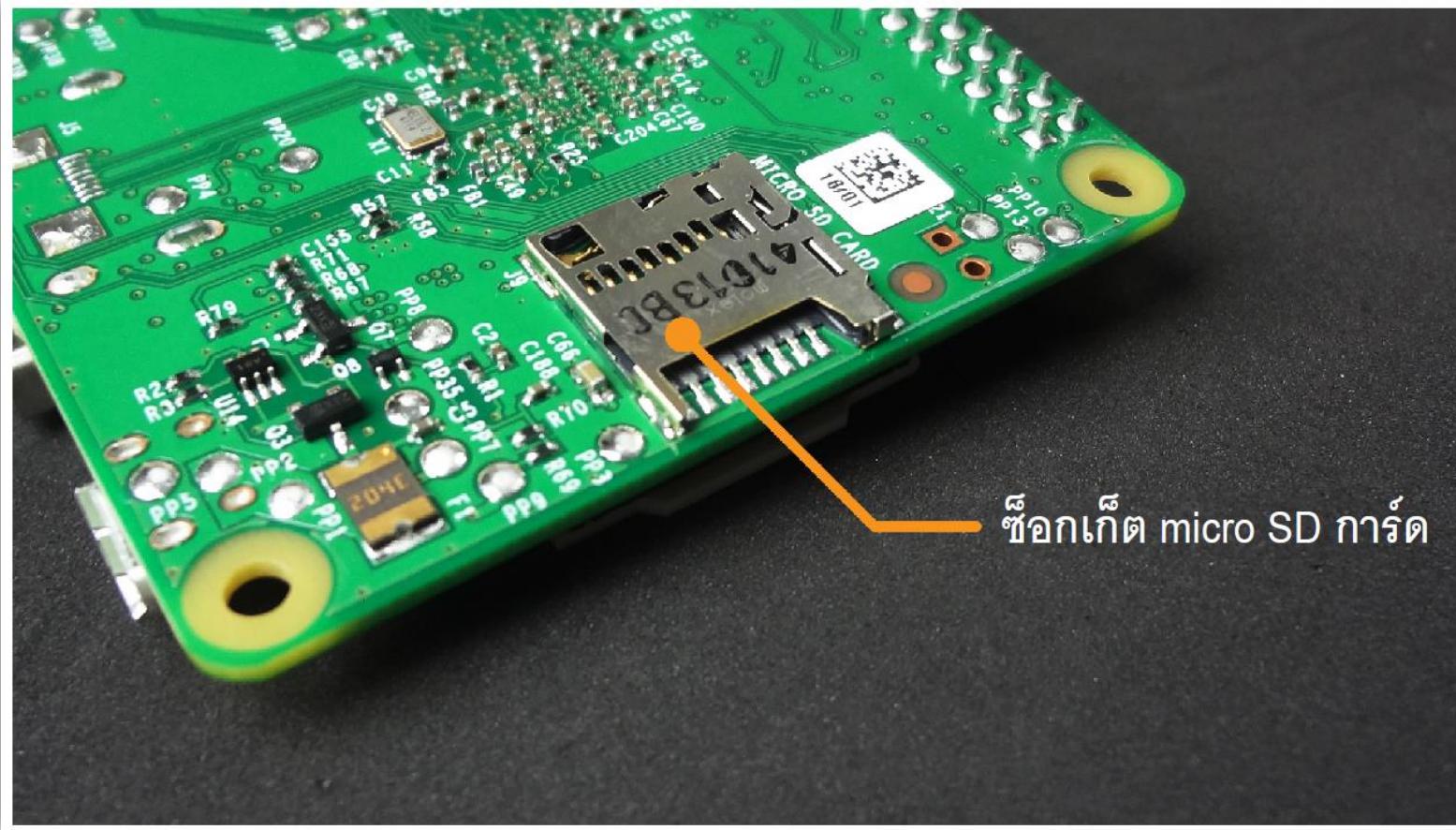
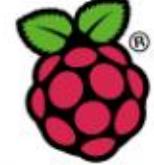
จุดเชื่อมต่อของ Raspberry Pi2



จุดเชื่อมต่อของ Raspberry Pi2

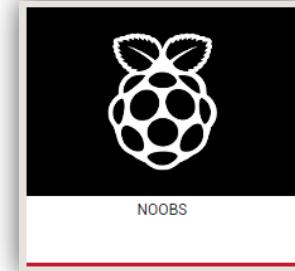
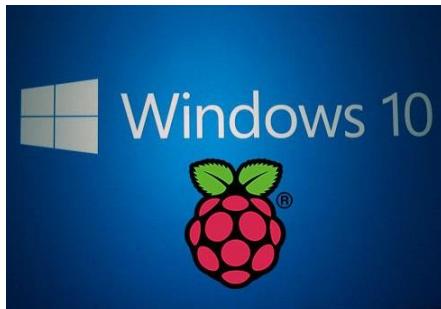
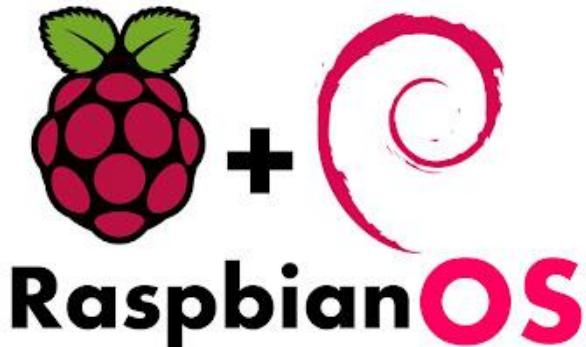
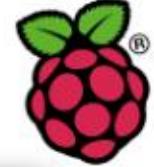


จุดเชื่อมต่อของ Raspberry Pi2

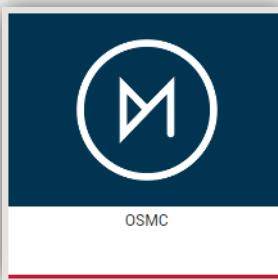


OS ของ Raspberry Pi 2 จะเก็บที่ microSD โดยมีช่องเสียบการ์ดอยู่ใต้บอร์ด

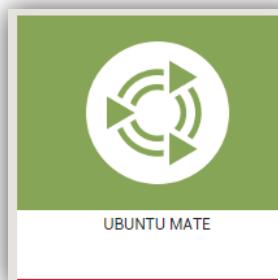
ระบบปฏิบัติการสำหรับ Raspberry Pi2



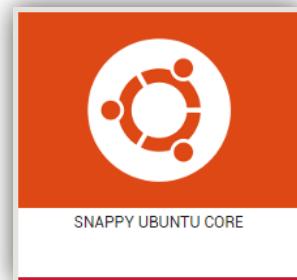
OpenELEC (media center)



(media center)



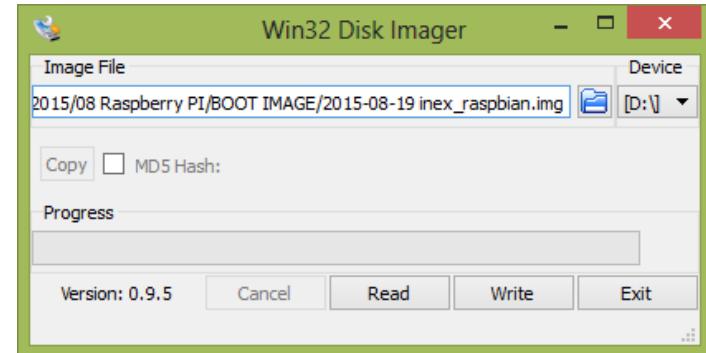
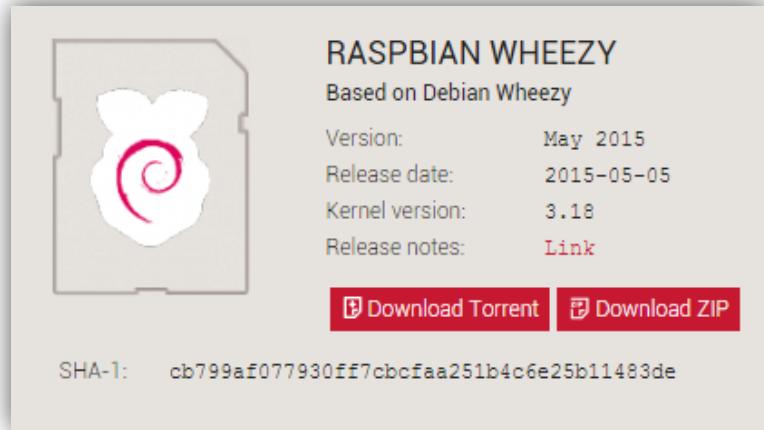
UBUNTU DESKTOP



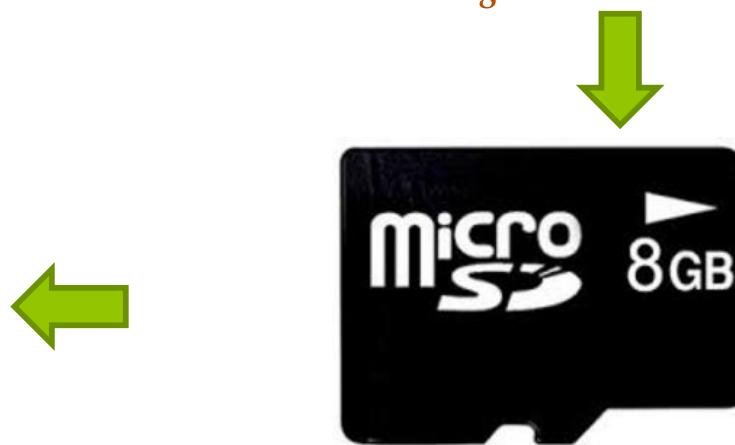
SNAPPY UBUNTU CORE

สำหรับเริ่มต้น

ระบบปฏิบัติการ



ใช้โปรแกรม win32 Disk Image



ดาวน์โหลดไฟล์ Image จาก Raspberrypi.org

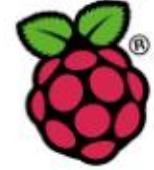


เสียบเข้าไปที่บอร์ด Raspberry Pi

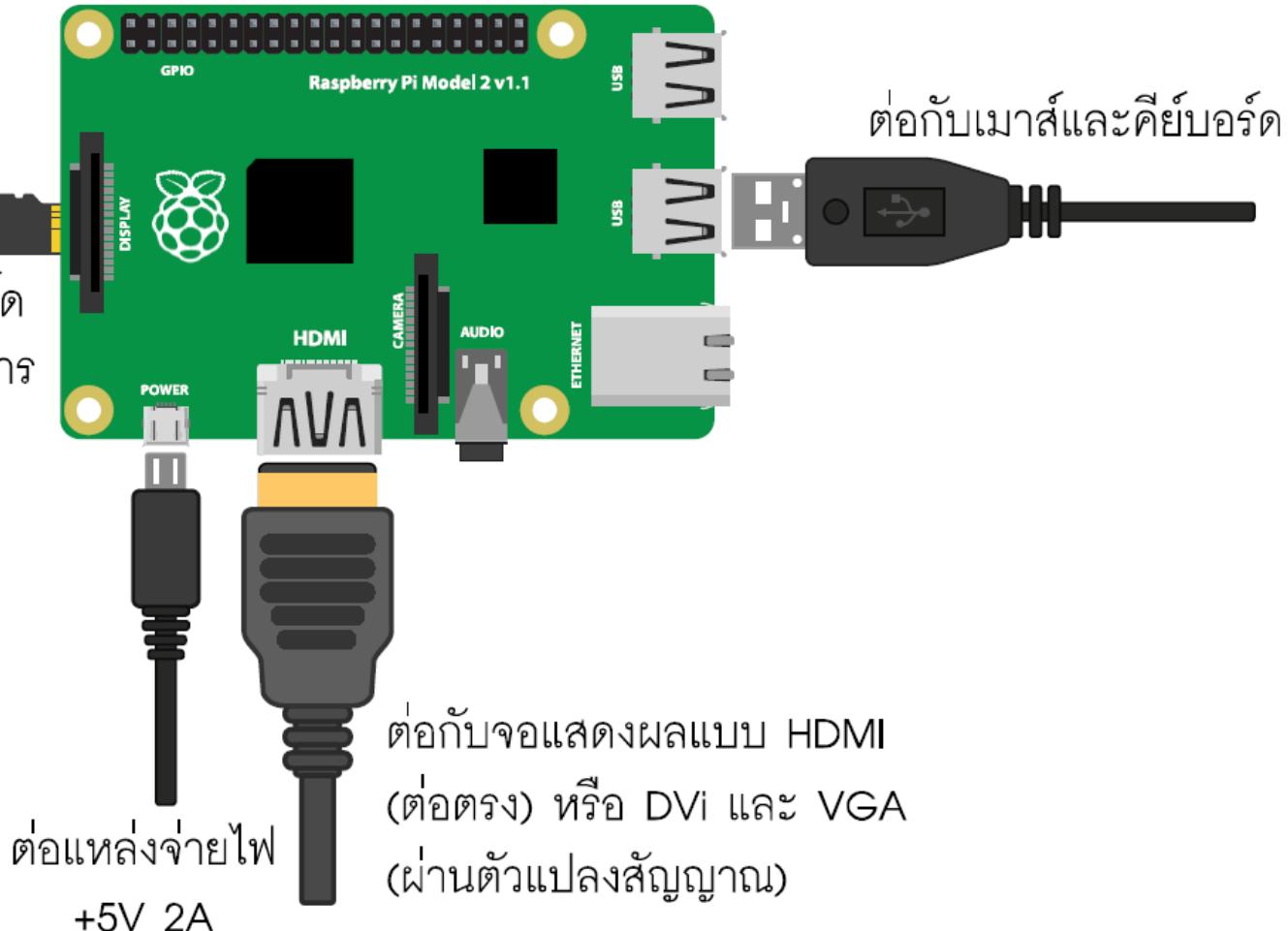


บันทึกไฟล์ลง micro SDCARD

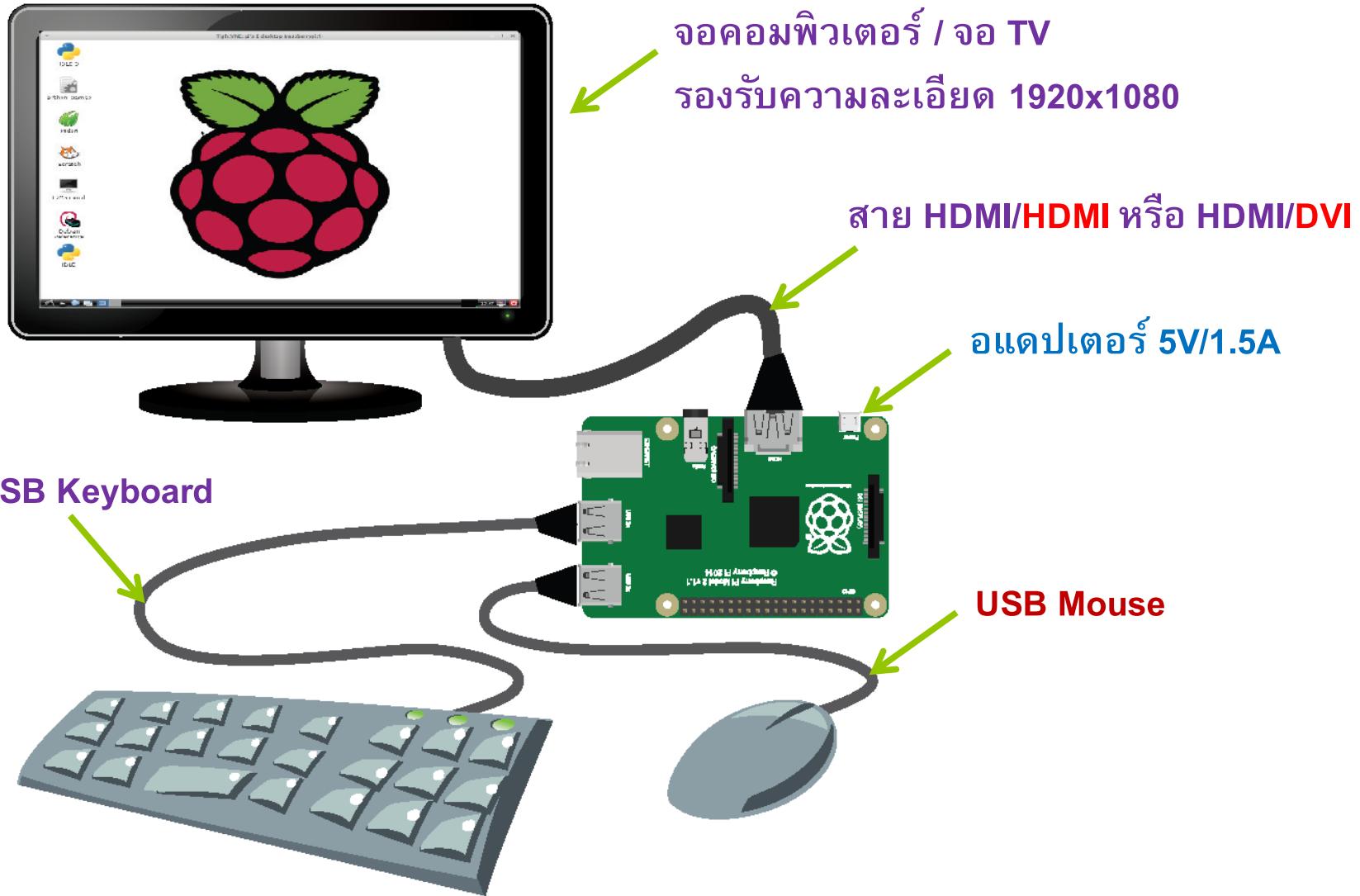
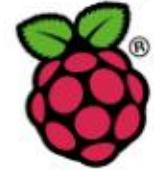
การเชื่อมต่ออุปกรณ์เบื้องต้น



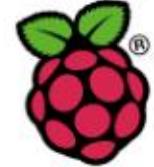
ติดตั้ง microSD การ์ด
ทบroot ระบบปฏิบัติการ
Raspbian แล้ว



อุปกรณ์ต่อพ่วงเบื้องต้น สำหรับโหมด Stand Alone



ប្រព័ន្ធប្រតិបត្តិការ Raspbian

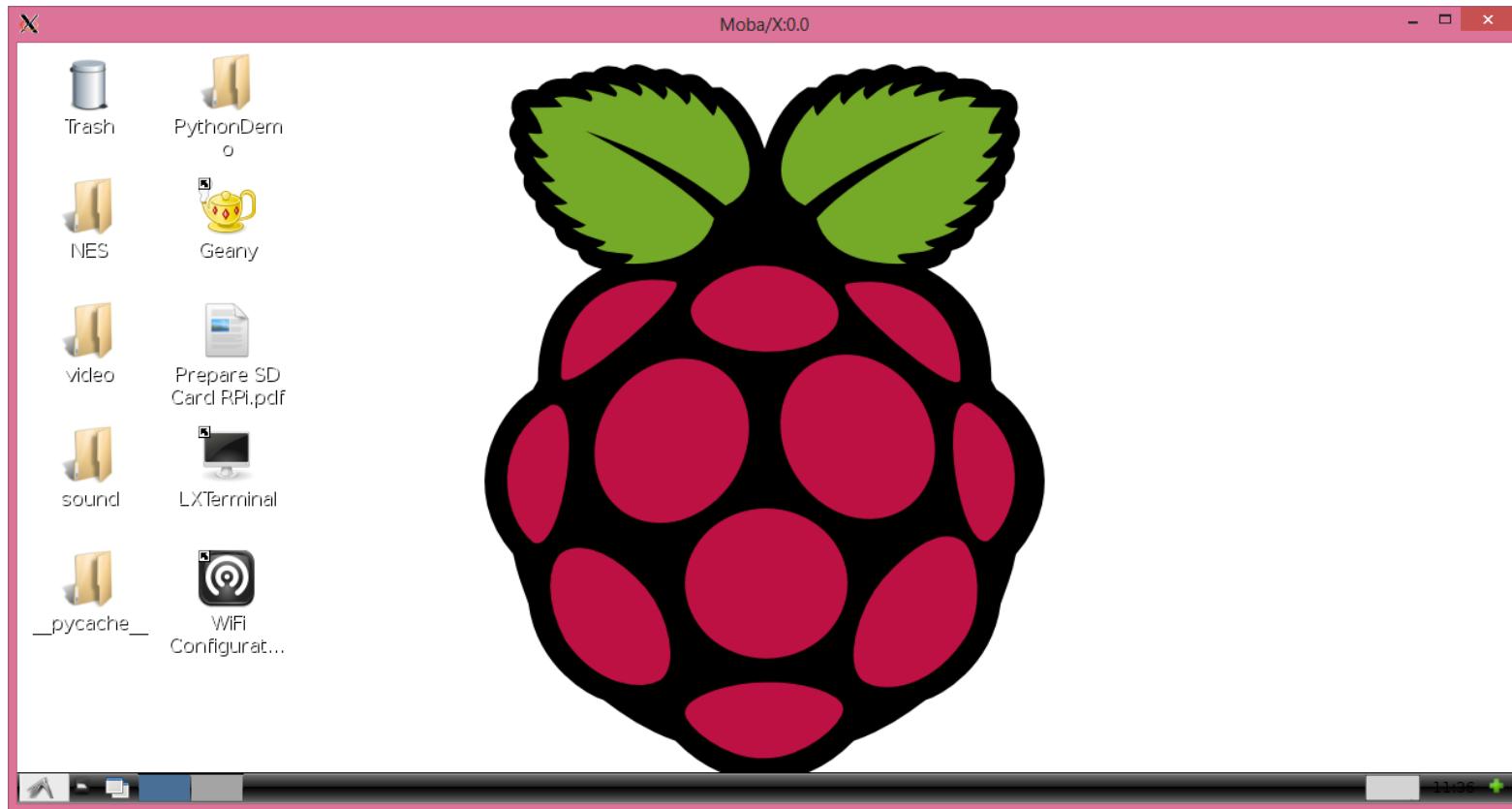


```
t 3.286397] [<801cf49b>] (ext4_lookup) from [<8014c018>] (lookup_real+0x30/0x5c)
t 3.297203] [<8014c010>] (lookup_real) from [<8014ccb8>] (_lookup_hash+0x4/0x4c)
t 3.308955] [<8014ccb8>] (_lookup_hash) from [<8014ec38>] (lookup_slow+0x48/0xb4)
t 3.319425] [<8014ec30>] (lookup_slow) from [<8014fb0c>] (path_lookupat+0x6e8/0x730)
t 3.330678] [<8014fb8c>] (path_lookupat) from [<8014ff98>] (filename_lookup.isra_46+0x30/0x70)
t 3.342818] [<8014fb98>] (filename_lookup.isra_46) from [<80152130>] (user_path_at_empty+0x64/0x8c)
t 3.355400] [<80152130>] (user_path_at_empty) from [<8015217c>] (user_path_at+0x24/0x2c)
t 3.367063] [<8015217c>] (user_path_at) from [<80141f18>] (SyS_faccessat+0xa0/0x1d8)
t 3.378370] [<80141f18>] (SyS_faccessat) from [<80142078>] (SyS_access+0x28/0x2c)
t 3.389433] [<80142078>] (SyS_access) from [<8000cbc0>] (ret_fast_syscall+0x0/0x40)
t 3.400661] Code: e7934004 e3540000 0a00004c e5963014 (e794e003)
t 3.410367] ---[ end trace de8305eb0d5102c5 ]---
t 3.413518] usb 1-1: New USB device found, idVendor=0424, idProduct=9514
t 3.413525] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
t 3.414105] hub 1-1:1.0: USB hub found
t 3.414178] hub 1-1:1.0: 5 ports detected
t 3.454547] Kernel panic - not syncing: Attempted to kill init! exitcode=0x0000000b
t 3.454547]
t 3.470682] CPU0: stopping
t 3.476842] CPU: 0 PID: 0 Comm: swapper/0 Tainted: G      D    3.18.7-0?+ #755
t 3.487908] [<80016d14>] ( unwind_backtrace) from [<80012c40>] (show_stack+0x20/0x24)
t 3.499161] [<80012c40>] (show_stack) from [<8002fc0>] (dump_stack+0x98/0xd8)
t 3.509881] [<80052efe8>] (dump_stack) from [<8001509c>] (handle_IPI+0x234/0x268)
t 3.520759] [<8001509c>] (handle_IPI) from [<80000618>] (do_IPI+0x10/0x1c)
t 3.531122] [<80000618>] (do_IPI) from [<805349b4>] (_irq_svc+0x34/0x14c)
t 3.541471] Exception stack(0x807cbf08 to 0x807cbf50)
t 3.550003] b700:          807e9ccc 00000000 ffffffd 00000000 807ca020 807e8dd4
t 3.561698] b720: 00000000 00000000 807ca000 807e8d70 808238fe 807cbf5c 807cbf50 807cbf50
t 3.573392] b740: 00000f900 00000f904 60000113 ffffff
t 3.581942] [<805349b4>] (_irq_svc) from [<8000f904>] (arch_cpu_idle+0x30/0x4c)
t 3.592860] [<8000f904>] (arch_cpu_idle) from [<8005c5cc>] (cpu_startup_entry+0x314/0x3d4)
t 3.604653] [<8005c5cc>] (cpu_startup_entry) from [<8052a5dc>] (rest_init+0x80/0x98)
t 3.615923] [<8052a5dc>] (rest_init) from [<80767cb4>] (start_kernel+0x370/0x3db)
t 3.626916] CPU2: stopping
t 3.633108] CPU: 2 PID: 0 Comm: swapper/2 Tainted: G      D    3.18.7-0?+ #755
t 3.644297] [<80016d14>] ( unwind_backtrace) from [<80012c40>] (show_stack+0x20/0x24)
t 3.655495] [<80012c40>] (show_stack) from [<8002fc0>] (dump_stack+0x98/0xd8)
t 3.666258] [<80052efe8>] (dump_stack) from [<8001509c>] (handle_IPI+0x234/0x268)
t 3.677172] [<8001509c>] (handle_IPI) from [<80000618>] (do_IPI+0x10/0x1c)
t 3.687551] [<80000618>] (do_IPI) from [<805349b4>] (_irq_svc+0x34/0x14c)
t 3.697916] Exception stack(0xb98c7f58 to 0xb98c7fa0)
t 3.706455] 7f40:          807e9ccc 00000000
t 3.718153] 7f60: ffffffed 00000000 b98c6030 807e8dd4 00000000 00000000 b98c6000 807e8d70
t 3.729063] 7f80: 800238fe b98c7fac b98c7fa0 b98c7fa0 8000f900 8000f904 60000113 ffffff
t 3.741572] [<805349b4>] (_irq_svc) from [<8000f904>] (arch_cpu_idle+0x30/0x4c)
t 3.752507] [<8000f904>] (arch_cpu_idle) from [<8005c5cc>] (cpu_startup_entry+0x314/0x3d4)
t 3.764320] [<8005c5cc>] (cpu_startup_entry) from [<80014bec>] (secondary_start_kernel+0x144/0x164)
t 3.776946] [<80014bec>] (secondary_start_kernel) from [<800086a4>] (0x86a4)
t 3.787565] CPU1: stopping
t 3.793830] CPU: 4 PID: 0 Comm: swapper/1 Tainted: G      D    3.18.7-0?+ #755
t 3.804986] [<80016d14>] ( unwind_backtrace) from [<80012c40>] (show_stack+0x20/0x24)
```

User : pi

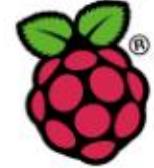
Password : raspberry

ເໜັສູ່ Graphic Mode

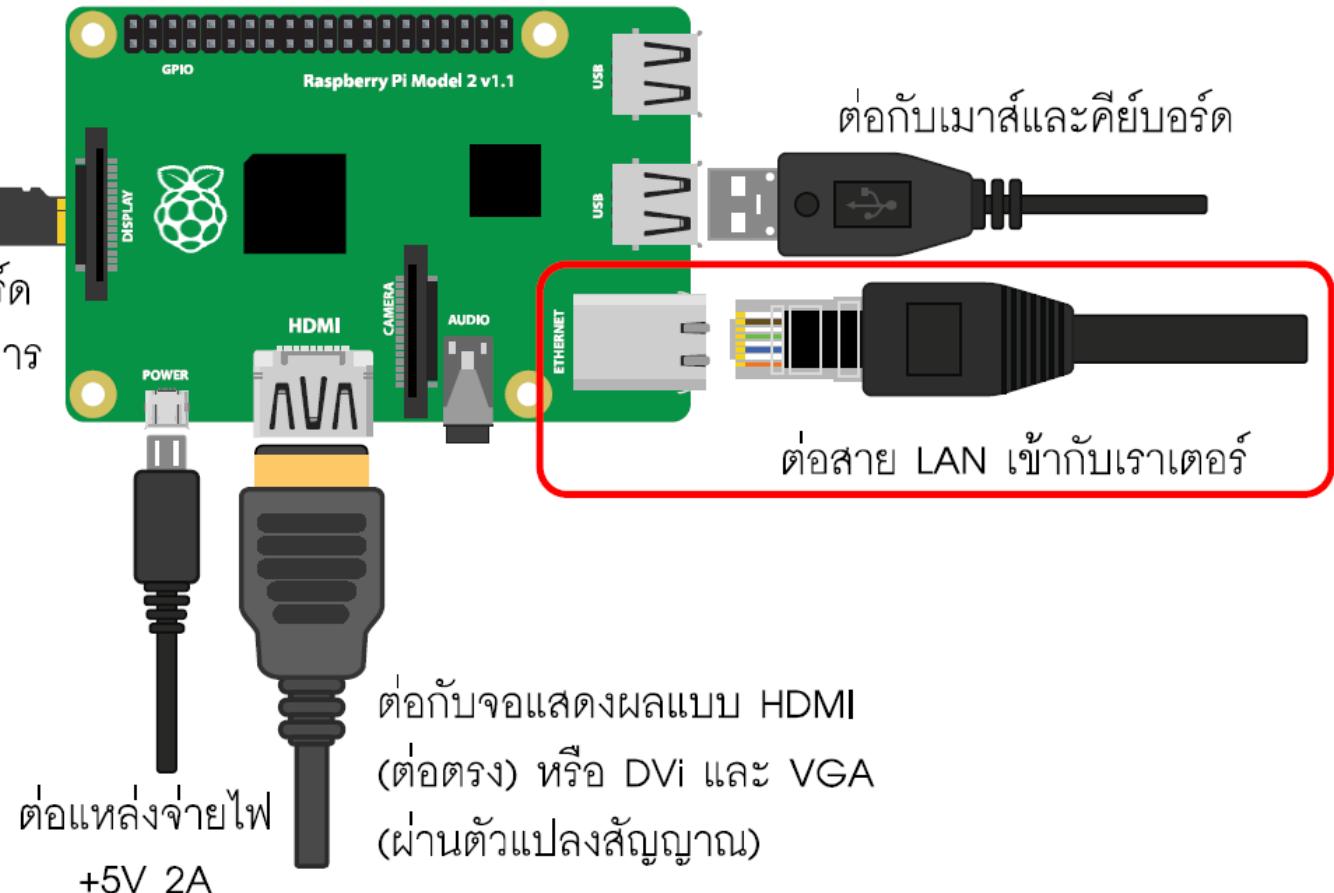


ພິມພໍ STARTX

การเข้าสู่ระบบเครือข่ายและอินเทอร์เน็ต

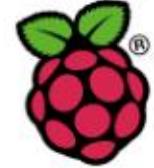


ติดตั้ง microSD การ์ด
ที่บรรจุระบบปฏิบัติการ
Raspbian และ



สามารถต่อสาย LAN เพื่อต่อเข้าอินเทอร์เน็ตได้ทันที

การอัพเดตและติดตั้งโปรแกรม



sudo apt-get update ตรวจสอบการอัพเดตซอฟต์แวร์

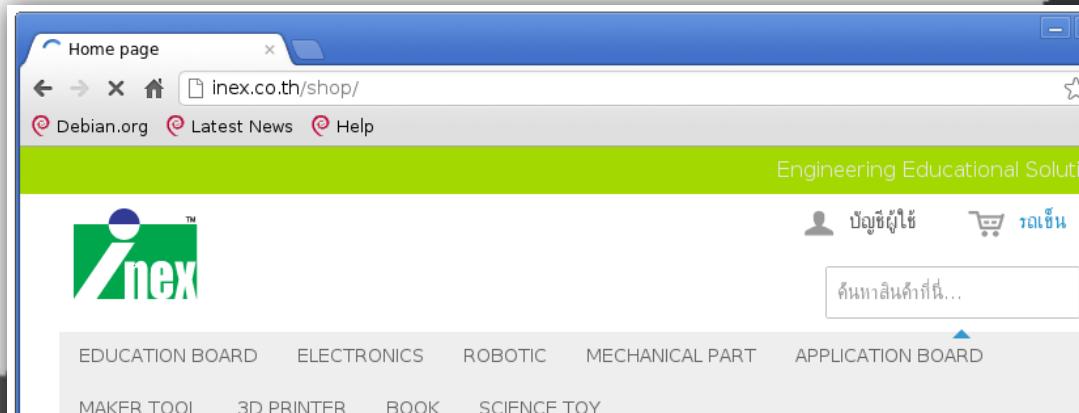
sudo apt-get upgrade อัพเกรดซอฟต์แวร์ทั้งหมดที่ติดตั้ง

ติดตั้งโปรแกรม

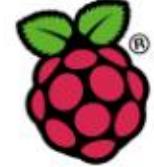
sudo apt-get install (ชื่อโปรแกรม)

ตัวอย่าง ติดตั้ง browser

sudo apt-get install chromium-browser



คำสั่งที่ใช้งานบ่อย



apt-get update อัพเดตเวอร์ชัน Raspbian

apt-get upgrade อัพเกรดซอฟต์แวร์ทั้งหมดที่ติดตั้ง

clear เคลียร์หน้าต่าง Teminal

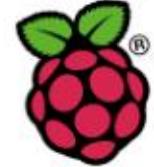
date แสดง วันเวลา ปัจจุบัน

```
pi@raspberrypi ~ $ date
Mon Apr  6 16:44:32 ICT 2015
pi@raspberrypi ~ $
```

find -name *.* ค้นหาชื่อไฟล์ที่ต้องการในทุกโฟลเดอร์

```
pi@raspberrypi ~ $ find -name *.txt
./python_games/starPusherLevels.txt
./arduino/preferences.txt
pi@raspberrypi ~ $
```

คำสั่งที่ใช้งานบ่อย



touch xxx.txt สร้างไฟล์ใหม่

nano xxx.txt เปิดไฟล์ด้วย Text Editor

The screenshot shows a terminal window running the nano 2.2.6 editor. The file being edited is named 'xxx.txt' and has been modified, as indicated by the 'Modified' status bar. The main area contains the text 'Read Me Please !'. At the bottom, a series of keyboard shortcuts are listed, with '^X Exit' highlighted by a red oval. An orange arrow points from this oval down to a confirmation dialog at the bottom of the screen. This dialog asks 'Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?' and provides two options: 'Y Yes' (highlighted by a red oval) and 'N no'. A third option, '^C Cancel', is also visible.

คำสั่งที่ใช้งานบ่อย



sudo poweroff ปิดเครื่องแบบทันที ทันใด

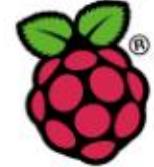
sudo reboot รีสตาร์ตเครื่องใหม่

sudo shutdown -h now ปิดระบบ

startx เปิดหน้าต่าง GUI ของ R-Pi

startlxde เปิดหน้าต่าง GUI ของ R-Pi (Remote)

คำสั่งที่ใช้งานบ่อย



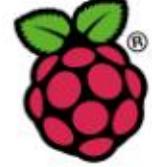
sudo raspi-config เปิดหน้าต่างตั้งค่าของ R-Pi

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem      Ensures that all of the s
2 Change User Password  Change password for the d
3 Enable Boot to Desktop/Choose whether to boot in
4 Internationalisation   Set up language and regio
5 Enable Camera          Enable this Pi to work wi
6 Add to Rastrack        Add this Pi to the online
7 Overclock              Configure overclocking fo
8 Advanced Options       Configure advanced settin
9 About raspi-config     Information about this co

<Select>           <Finish>
```

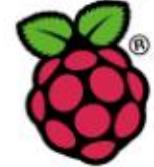
คำสั่งที่ใช้งานบ่อย



File/Directory Basics

ls	List files	แสดงรายชื่อไฟล์และไดเร็คทอรี
cp	Copy files	สำเนาไฟล์
mv	Rename files	เปลี่ยนชื่อไฟล์
rm	Delete files	ลบไฟล์
cd	Change directory	ย้ายไปยังไดเร็คทอรีที่ต้องการ
pwd	Print directory name	แสดงชื่อไดเร็คทอรีปัจจุบัน
mkdir	Create directory	สร้างไดเร็คทอรีใหม่
rmdir	Delete directory	ลบไดเร็คทอรี (ที่ว่างเปล่าเท่านั้น)

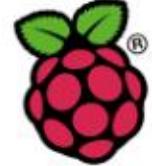
คำสั่งที่ใช้งานบ่อย



File Viewer

cat	View files	ดูเนื้อหาของ text file
less	Page through files	เลื่อนดูเนื้อหาของไฟล์ ออกก่อน Ctrl+Z
head	View file beginning	แสดงส่วนต้นของไฟล์
tail	View files ending	แสดงส่วนท้ายของไฟล์
nl	Number lines	แสดงหมายเลขบรรทัด
od	View binary files	แสดงเนื้อหาในไฟล์ไบนารี่

คำสั่งที่ใช้งานบ่อย



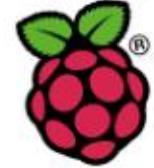
tee

Copy stdin to file and to stdout simultaneously

สำเนาข้อความออกจากไฟล์และ stdout พร้อม ๆ กัน

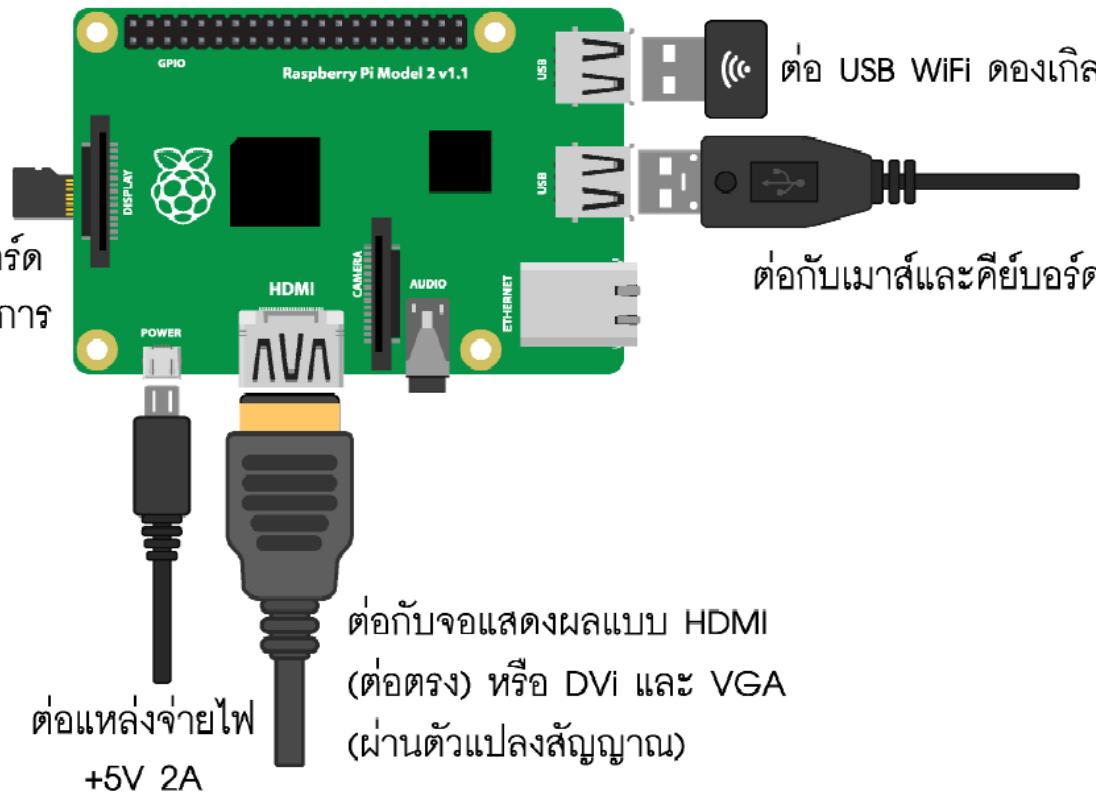
```
echo "Hello, world" | tee hello.txt  
Hello, world
```

การเชื่อมต่อผ่าน WIFI

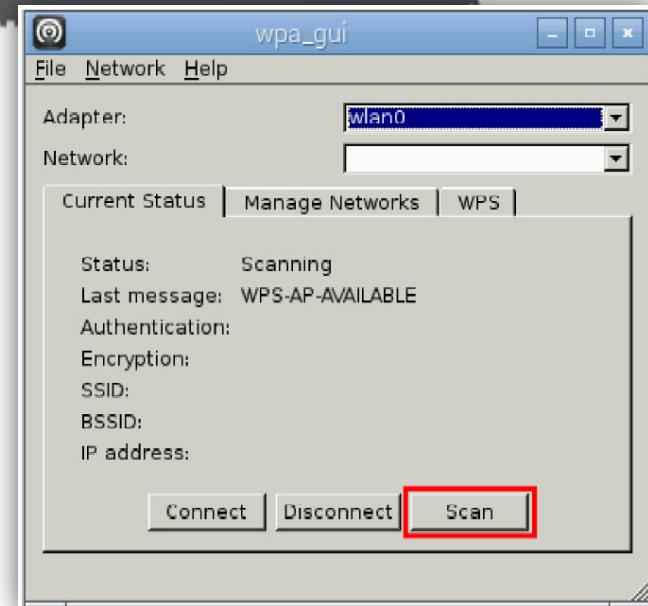
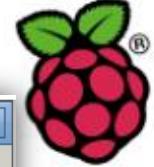


USB WIFI Dongle

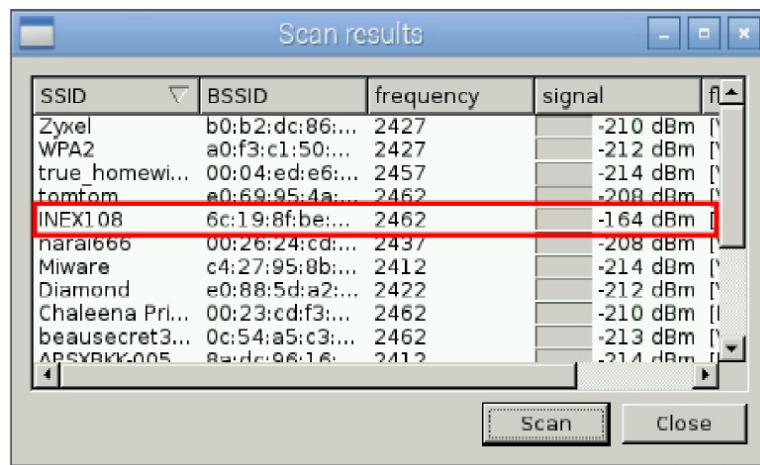
ติดตั้ง microSD การ์ด
ที่บรรจุระบบปฏิบัติการ
Raspbian และ



การตั้งค่าผ่านโปรแกรม WiFi Configuration

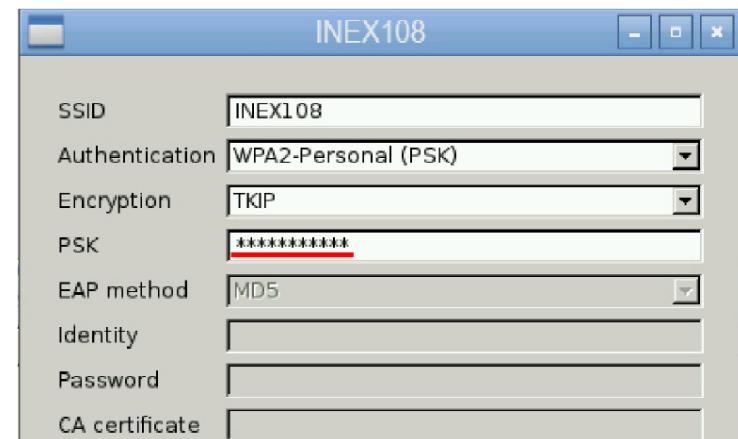


เปิด WiFi Configuration



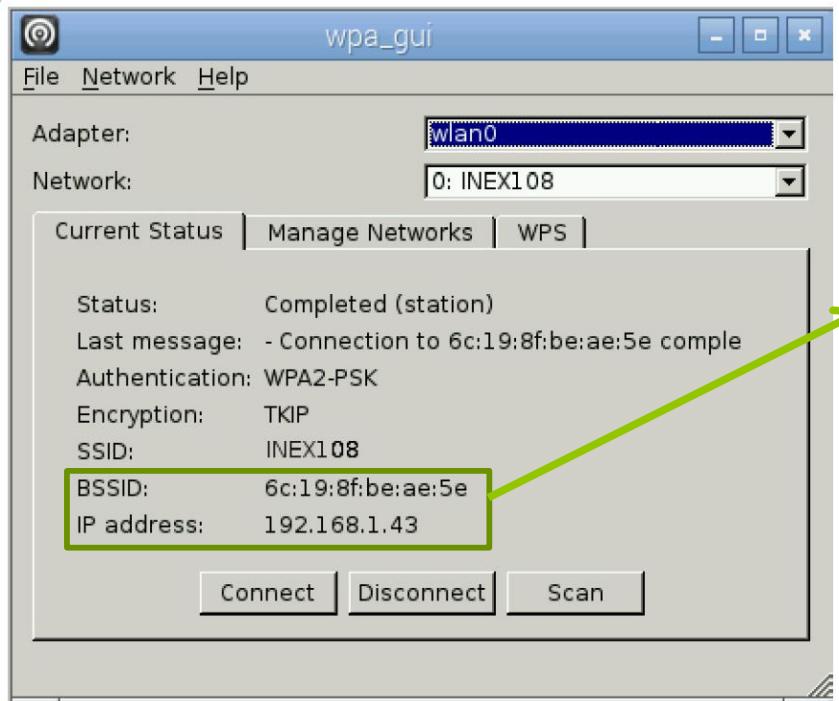
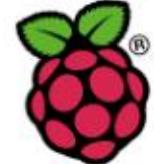
เลือกเครือข่ายที่ต้องการ

กด Scan เพื่อค้นหาเครือข่าย



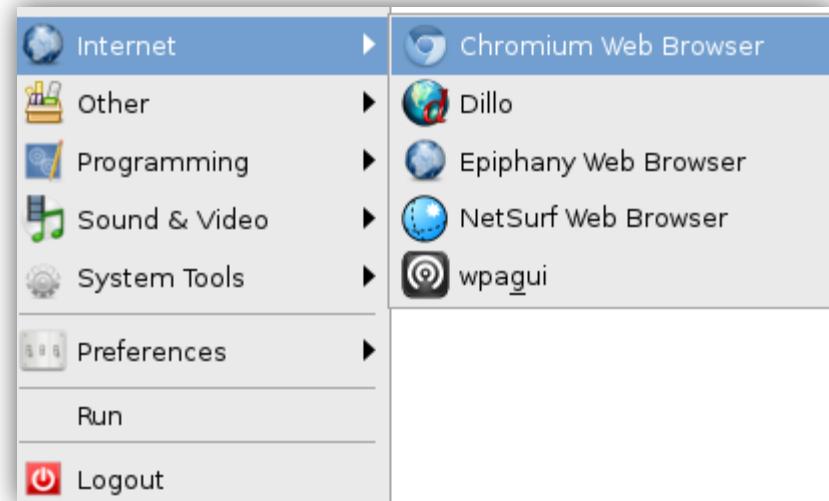
ใส่รหัสผ่าน สำหรับเครือข่ายที่เชื่อมต่อ

การตั้งค่าผ่านโปรแกรม WiFi Configuration



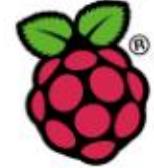
BSSID: 6c:19:8f:be:ae:5e
IP address: 192.168.1.43

IP Address ของ Raspberry Pi



เปิด Browser เพื่อทดสอบเชื่อมต่ออินเทอร์เน็ต

การแบ่งใช้อินเทอร์เน็ตจาก Notebook



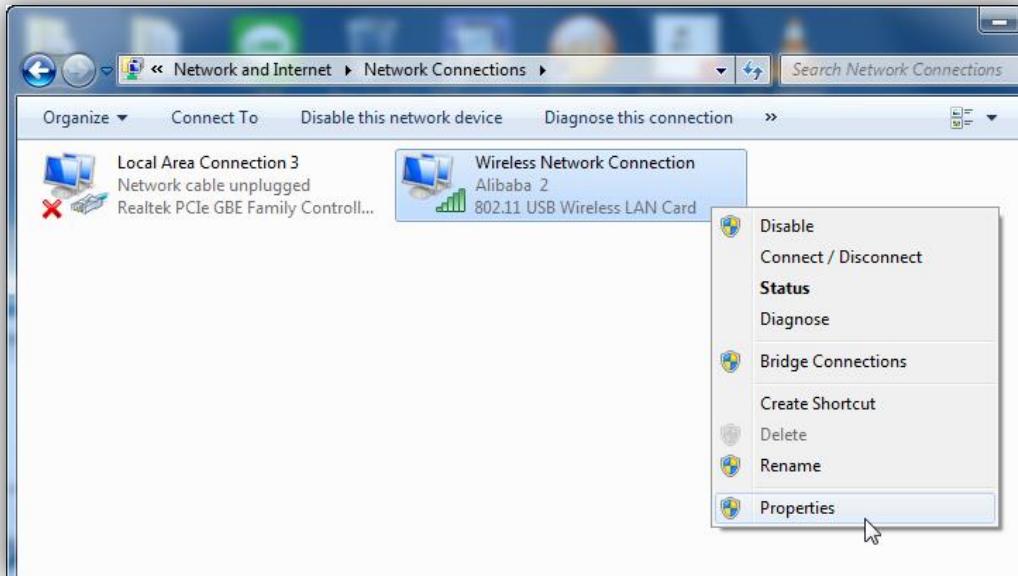
Router หรือ มือถือ

สาย LAN

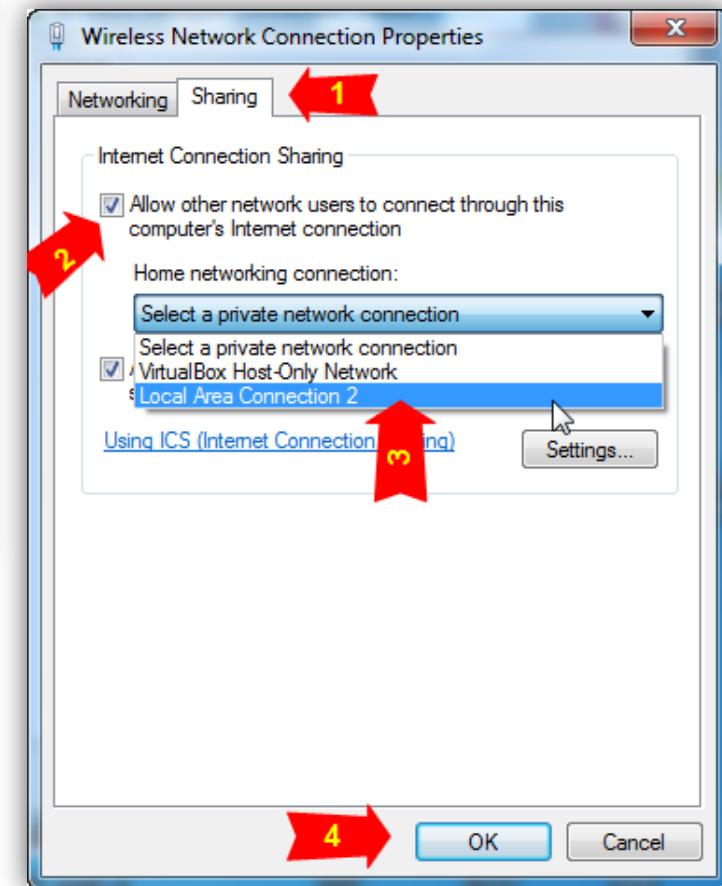
Notebook ต่ออินเทอร์เน็ตด้วย WiFi แชร์ให้ RPi ด้วยสาย LAN

ทำให้อยู่ในเครือข่ายเดียวกันและเล่นอินเทอร์เน็ตได้ด้วย

ตั้งค่า Network ที่ Notebook



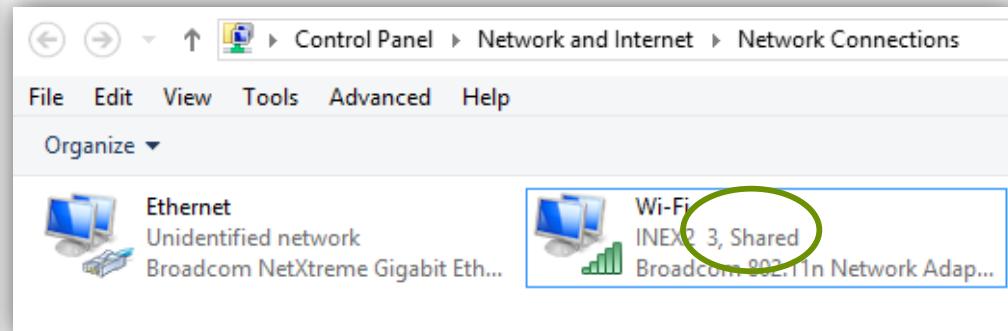
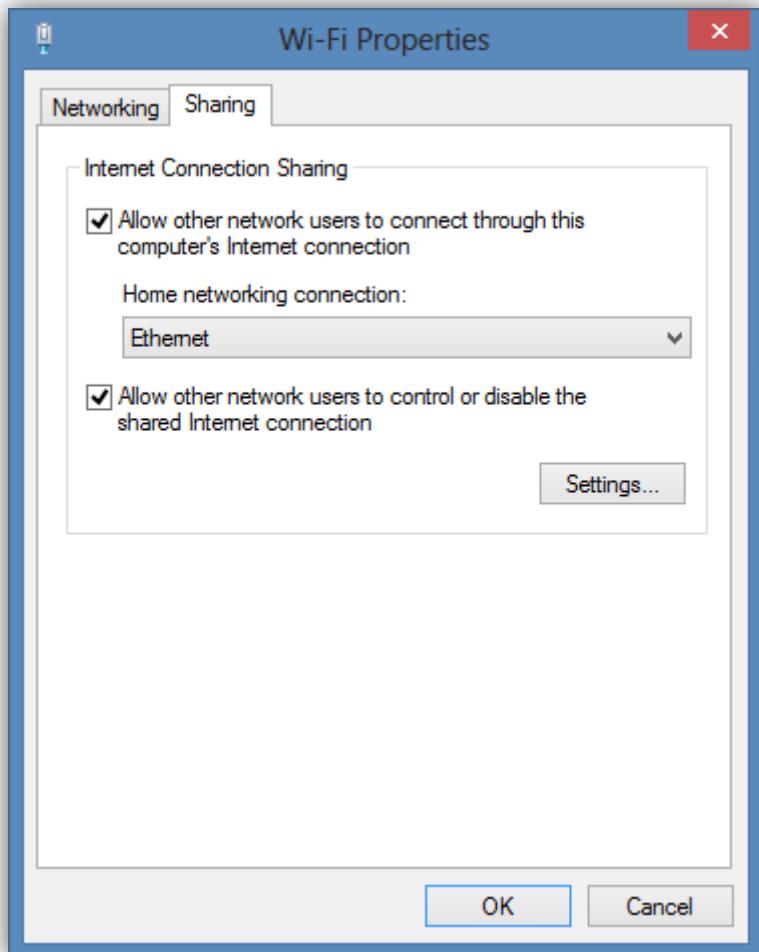
คลิกเม้าส์ขวาเลือก Properties ที่ WIFI



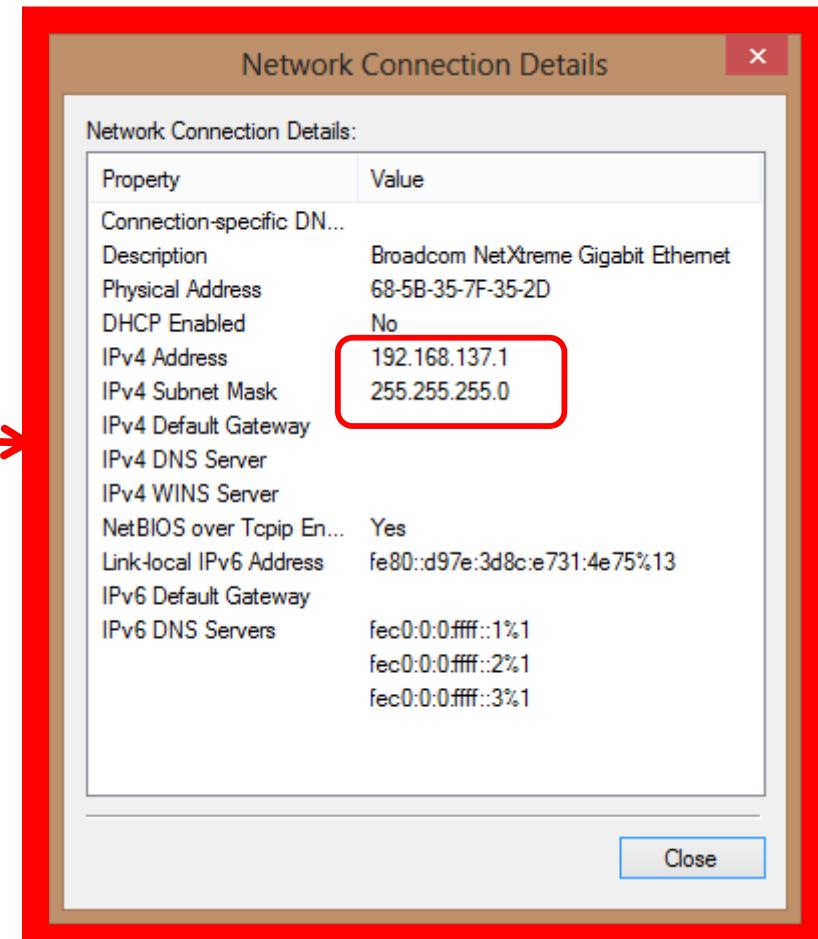
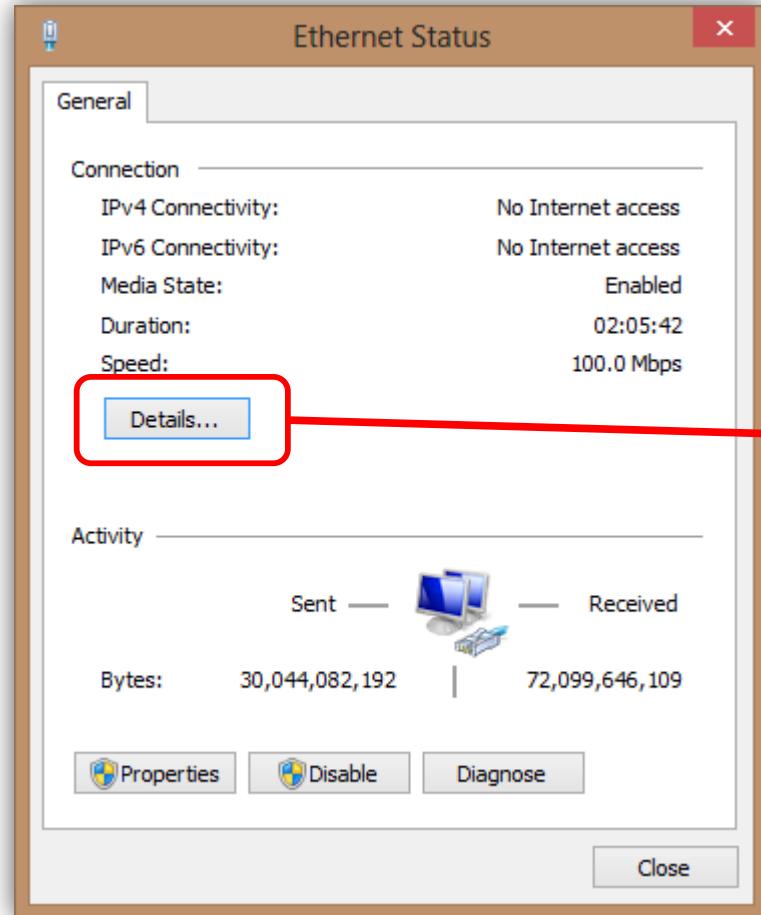
เลือกแชร์อินเทอร์เน็ตให้ LAN

* ถ้าเป็น windows 8 จะใช้คำว่า Ethernet แทน Local Area Network

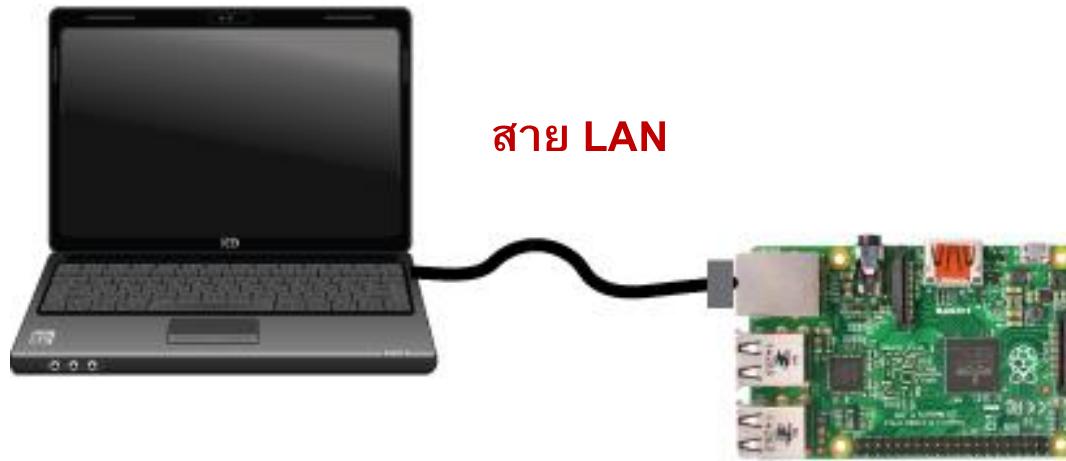
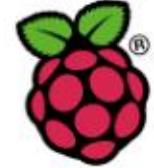
ຕັ້ງຄ່າ Network ທີ່ Notebook (windows 8)



ตรวจสอบ IP Address ตัวที่ปล่อย



การค้นหาตำแหน่ง IP จากบอร์ด RPi



2. ใช้โปรแกรม IPSCAN



1. เชื่อมต่อสาย LAN กับบอร์ด RPi

Free IP Scanner

File Edit View Help

IP Range From 192.168.137. 1 To 192.168.137.255

Customize IP List

IP Address	WorkGroup Name	Host Name	User
✓ 192.168.137.1	INEX	KRITSADA8	N/A
✓ 192.168.137.135	N/A	N/A	N/A
✗ 192.168.137.2	N/S	N/S	N/S
✗ 192.168.137.3	N/S	N/S	N/S
✗ 192.168.137.4	N/S	N/S	N/S
✗ 192.168.137.5	N/S	N/S	N/S
✗ 192.168.137.6	N/S	N/S	N/S

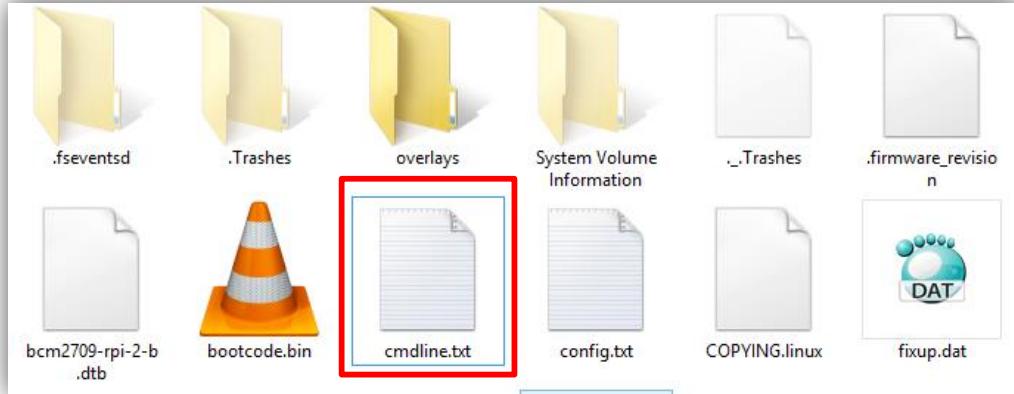
3. ค้นหา IP ของบอร์ด RPi

การ Fix ตำแหน่ง IP ของบอร์ด RPi



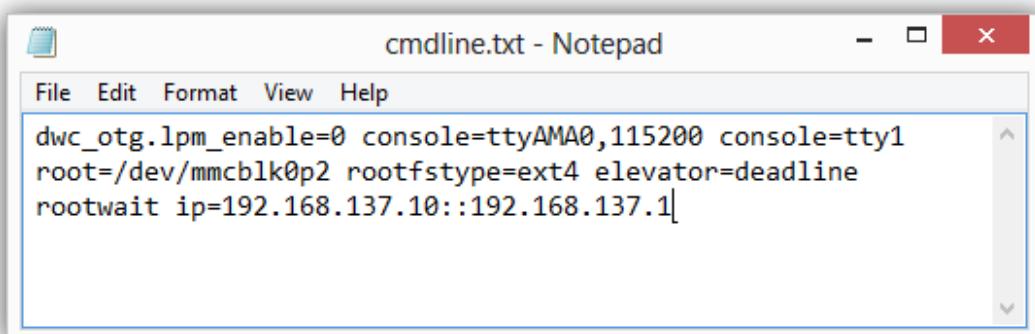
1. นำ SD CARD เสียบที่คอมพิวเตอร์

2. เปิดไฟล์ cmdline.txt ด้วย nodepad

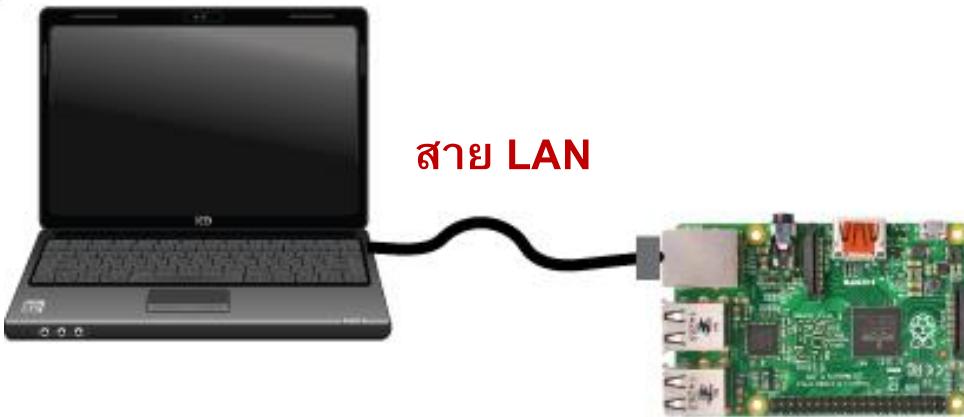


3. เพิ่ม ข้อความ ต่อท้าย แล้ว SAVE

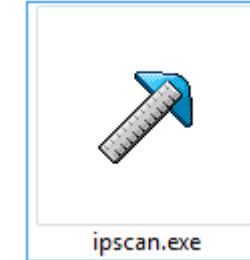
ip=192.168.137.10::192.168.137.1



ตรวจสอบ ip address อีกครั้ง



1. เชื่อมต่อบอร์ด RPi อีกครั้ง

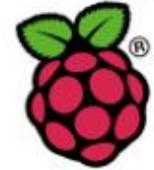


2. ตรวจสอบตำแหน่ง ip Address อีกครั้ง

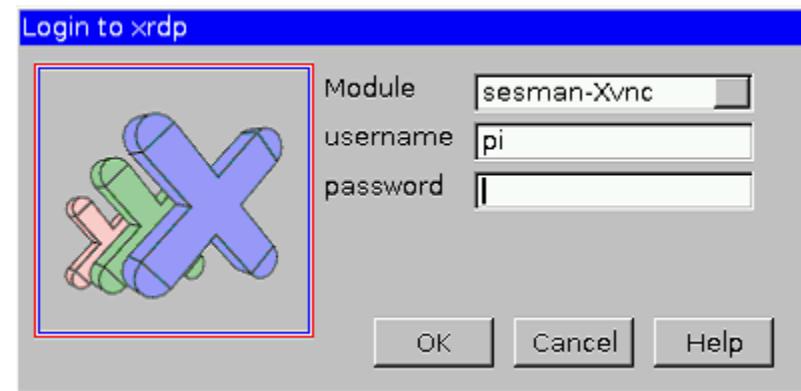
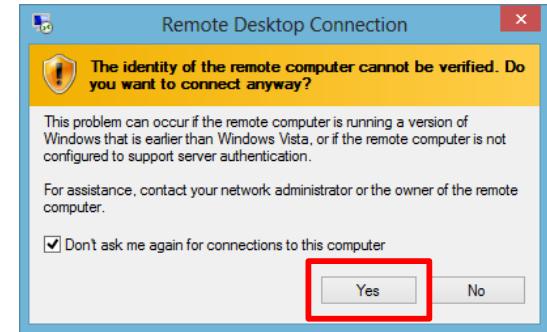
Free IP Scanner

IP Address	WorkGroup Name	Host Name	User	MAC Address
✓ 192.168.137.1	INEX	KRITSADA8	N/A	68-58-35-7F-35-2D
✓ 192.168.137.10	N/A	N/A	N/A	N/A
✗ 192.168.137.2	N/S	N/S	N/S	N/S
✗ 192.168.137.3	N/S	N/S	N/S	N/S
✗ 192.168.137.4	N/S	N/S	N/S	N/S
✗ 192.168.137.5	N/S	N/S	N/S	N/S
✗ 192.168.137.6	N/S	N/S	N/S	N/S

Remote ด้วยโปรแกรม Remote Desktop Connection



XRDP

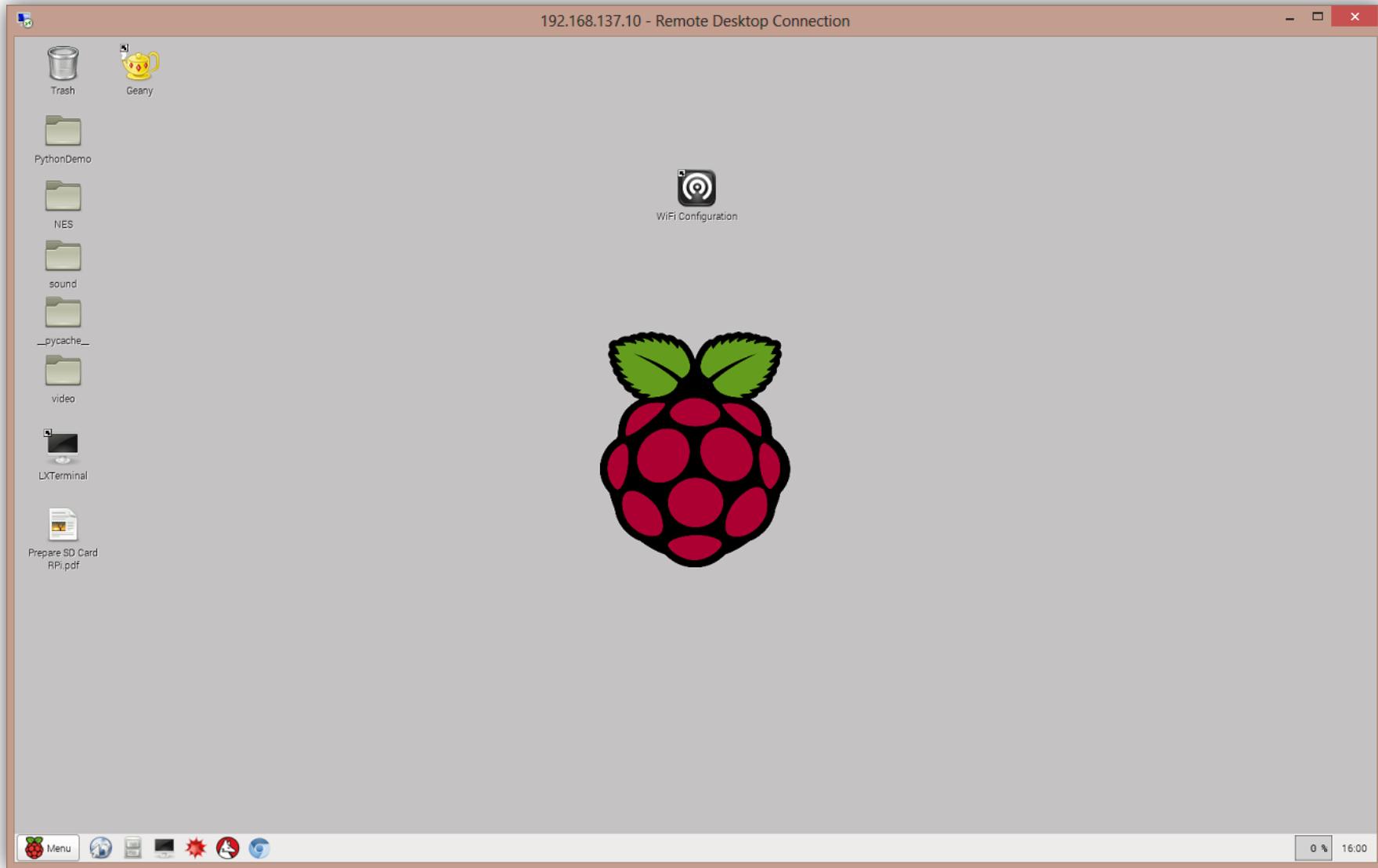
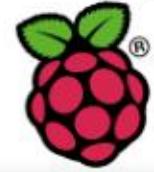


XRDP ติดตั้งบน RPi ไว้แล้ว เพื่อ
รองรับการ Remote ผ่านซอฟต์แวร์
Remote Desktop Connection ของ
windows

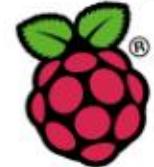
user : pi

password : raspberry

หน้าจอ Remote Desktop

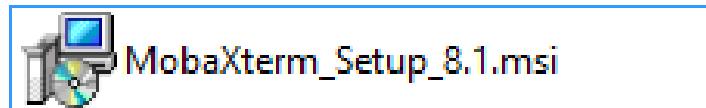


การควบคุมระยะไกลด้วย SSH (Moba XTerm)

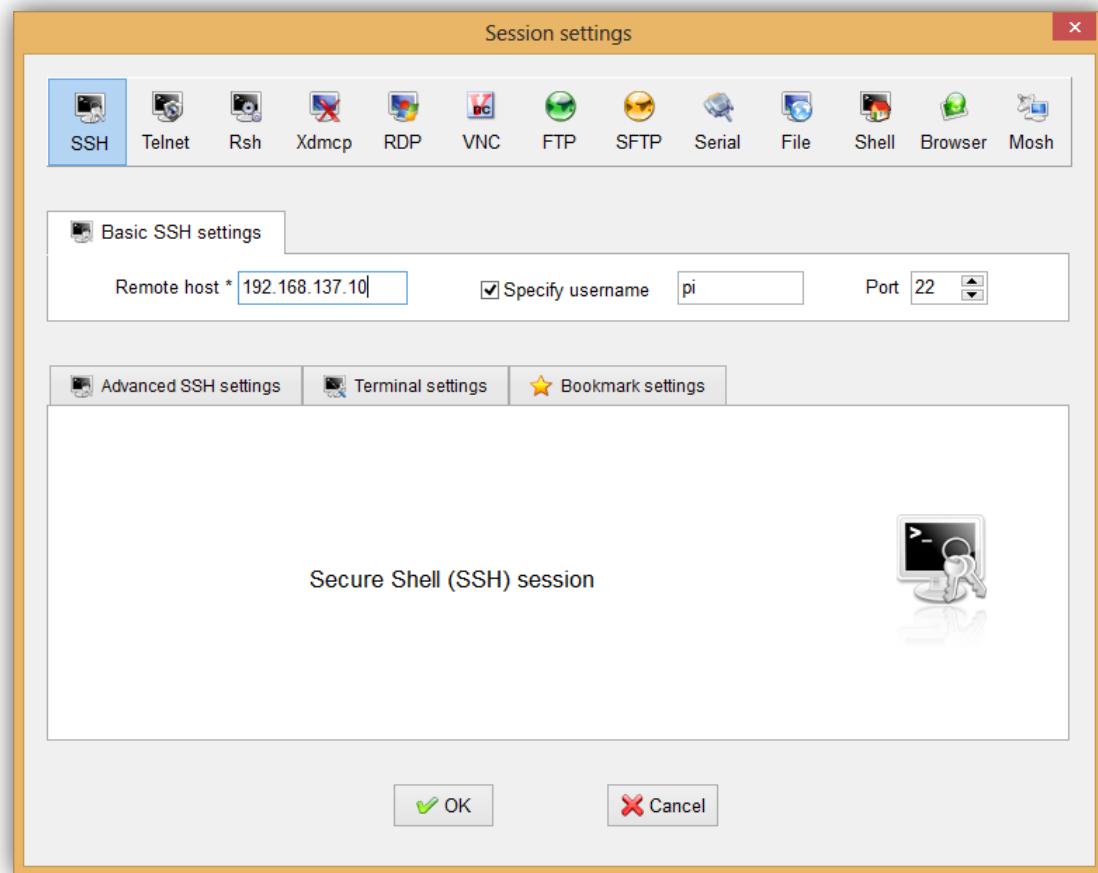


SSH : Secure Shell (เข้ารหัสข้อมูลก่อนส่ง)

1.ติดตั้งโปรแกรม Moba Xterm

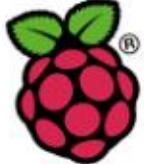


2.สร้าง Session แบบ SSH



3. X มุ่งหน้า สีเขียว

(Moba XTerm)



```
1. ./home/mobaxterm      2. RaspberryPi (pi) +  
• MobaXterm Personal Edition v7.7 •  
(X server, SSH client and GNU/Cygwin tools)  
► Your computer drives are accessible through the /drives path  
► Your DISPLAY is set to 192.168.137.1:0.0  
► When using SSH, your remote DISPLAY is automatically forwarded  
► Each command status is specified by a special symbol (✓ or ✗)  
• Important:  
This is MobaXterm Personal Edition. The Professional edition  
allows you to customize MobaXterm for your company: you can add  
your own logo, your parameters, your welcome message and generate  
either an MSI installation package or a portable executable.  
We can also modify MobaXterm or develop the plugins you need.  
For more information: http://mobaxterm.mobatek.net/versions.php  
pi@192.168.137.10's password: █
```

4. ป้อน password : raspberry

ขณะป้อน password ข้อความไม่ปรากฏ

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

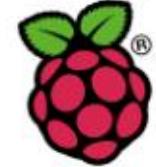
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Fri Nov 13 13:17:14 2015

pi@raspberrypi ~ \$ █

5. พร้อมพิมพ์ Command

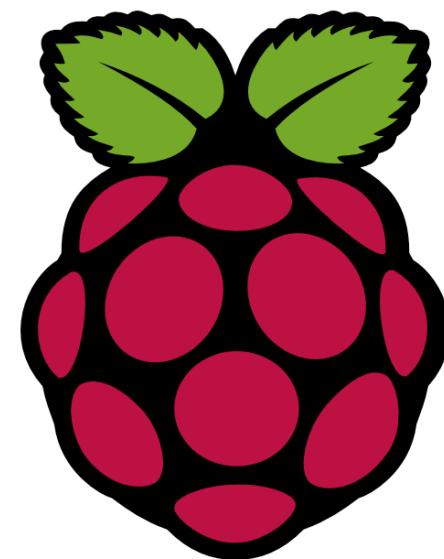
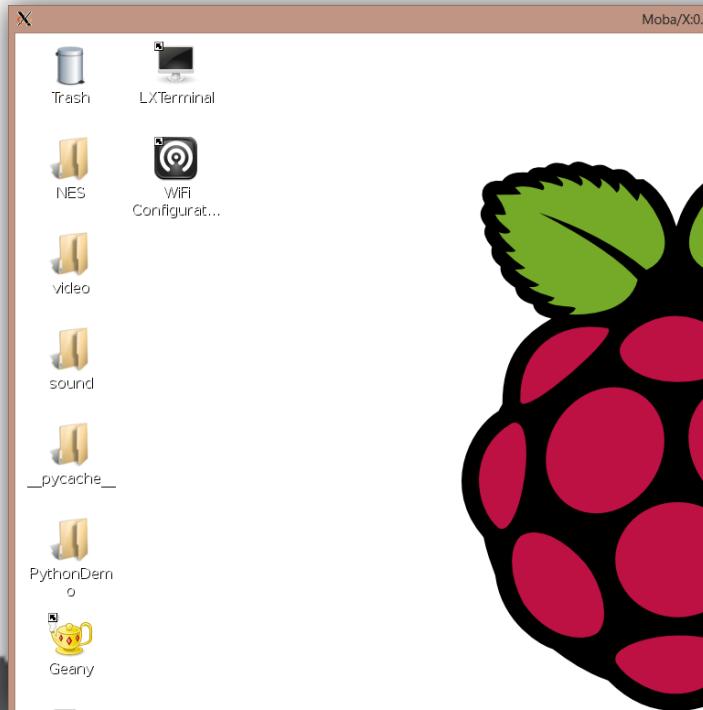
ໂຄມດ Graphic



6. พิมพ์คำสั่ง startlxde เพื่อเรียกໂຄມດ Graphic

```
pi@raspberrypi ~ $ startlxde
warning: output default not found; ignoring
X Error of failed request:  BadValue (integer parameter out of range for operation)
    Major opcode of failed request:  139 (RANDR)
    Minor opcode of failed request:  7 (RRSetScreenSize)
    Value in failed request:  0x0
    Serial number of failed request:  14
    Current serial number in output stream:  16
Obt-Message: Xinerama extension is not present on the server

** (notification-daemon:2280): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not
```



การตั้งค่าหน้าจอของโมดูล X11



MobaXterm Configuration

X11

General Terminal X11 SSH Display Toolbar

Automatically start X server at MobaXterm start up Xorg version 1.16.3

X11 server display mode:
"Windowed mode": X11 server constrained to a single container window

Hardware OpenGL acceleration Clipboard enabled

Unix-compatible keyboard Keyboard Automatic Display offset 0

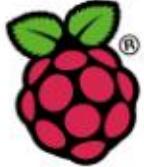
Access control Engine Automatic Run on monitor 1

X11 extensions (for advanced users only)

RANDR Composite DAMAGE XFIXES XTEST XINERAMA

OK Cancel

การคัดลอกไฟล์ผ่าน Moba Xterm



RaspberryPi (pi)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Settings Help

Quick connect...

Sessions Tools Macros Stop

/home/pi/

Name	Size (KB)	Last modified	Owner	Group
..				
Adafruit_Python_DHT	4	2015-08-14 ...	pi	pi
Desktop	4	2015-10-28 ...	pi	pi
Documents	4	2015-10-25 ...	pi	pi
Downloads	4	2015-11-13 ...	pi	pi
examples	28	2015-11-11 ...	pi	pi
logs	4	2015-08-19 ...	pi	pi
python_games	4	2015-01-27 ...	pi	pi
Scratch	4	2015-10-25 ...	pi	pi
sketchbook	4	2015-03-24 ...	pi	pi
Test01	4	2015-09-21 ...	pi	pi
wiringPi	4	2015-04-23 ...	pi	pi

1. /home/mobaxterm 2. RaspberryPi (pi)

```
** Message: x-terminal-emulator has very limited support, consider terminal
** Message: x-terminal-emulator has very limited support, consider terminal

** (pcmanfm:2271): WARNING **: The directory '~/Templates' does not exist
** (lxpanel:2268): WARNING **: The directory '~/Templates' does not exist

(lxpanel:2268): GLib-GObject-WARNING **: Attempt to add properties 'k-can-change-accel' after class was initialised

(lxpanel:2268): GLib-GObject-WARNING **: Attempt to add properties 'k-menu-popup-delay' after class was initialised

(lxpanel:2268): GLib-GObject-WARNING **: Attempt to add properties 'k-menu-popdown-delay' after class was initialised

(lxpanel:2268): GLib-GObject-WARNING **: Attempt to add properties 'k-menu-images' after class was initialised

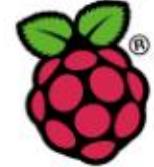
(lxpanel:2268): GLib-GObject-WARNING **: Attempt to add properties 'k-label-select-on-focus' after class was initialised

(lxpanel:2268): GLib-GObject-WARNING **: Attempt to add properties 'k-button-images' after class was initialised

(pcmanfm:2271): GLib-GObject-WARNING **: Attempt to add properties 'k-can-change-accel' after class was initialised

(pcmanfm:2271): GLib-GObject-WARNING **: Attempt to add properties 'k-can-change-accel' after class was initialised
```

ฝึก Python 3 ผ่าน Command Line



```
pi@raspberrypi ~ $ python3
Python 3.2.3 (default, Mar  1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

พิมพ์แล้วเห็นผลลัพธ์ทันที

```
>>> 2 + 2          ผลลัพธ์จากการบวกเป็น int
4
```

```
>>> 50 - 5*6
```

```
20
```

```
>>> (50 - 5*6) / 4
```

```
5.0
```

```
>>> 8 / 5
```

```
1.6          ผลลัพธ์จากการหารเป็น Float
```

```
>>> 17 / 3
```

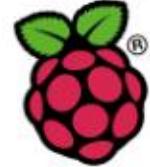
```
5.666666666666667 ยกกำลัง
```

```
>>> 5 ** 2
```

```
25
```

ออกจากโปรแกรมใช้ Ctrl+D

คำสั่ง Print



```
>>> print ("inex")
```

inex

```
>>> print ('inex')
```

inex

```
>>> print ('Line1\nLine2\nLine3')
```

Line1

Line2

Line3

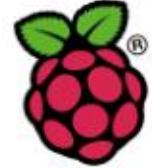
```
>>> x=10
```

```
>>> print ('x=' ,x)
```

x=10

รูปแบบ	ผลลัพธ์
'	'
"	"
\n	ขึ้นบรรทัดใหม่
\t	Tab 1 ครั้ง

บวก ลบ คูณ หาร



```
>>> 2 + 2
```

4

ผลลัพธ์จากการบวกเป็น int

```
>>> 50 - 5*6
```

20

```
>>> (50 - 5*6) / 4
```

5.0

ผลลัพธ์จากการหารเป็น Float

```
>>> 8 / 5
```

1.6

```
>>> 17 / 3
```

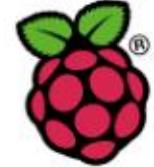
5.666666666666667

```
>>> 5 ** 2
```

25

ยกกำลัง

ตัวแปรภายใน Python



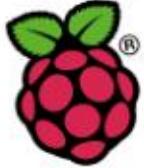
ใช้คำสั่ง **type (var)** ตรวจสอบชนิดตัวแปร

```
>>> x = 400      x เป็น Integer  
>>> y = 4.5      y เป็น Float  
>>> z = True     z เป็น Boolean  
>>> s = 'Rasp'   s เป็น String
```

Python ไม่ใช้ Array ใช้ List แทน หลายชนิดอยู่ที่เดียวกันได้

```
>>> i = [1, 2, 3, 4, 'Rasp', 5, 'Pi']    i เป็น List  
>>> print (i[4])  
Rasp  
>>> print (i[2:6])  
[3,4,'Rasp',5]
```

การกำหนดค่าตัวแปร



เครื่องหมาย = ใช้กำหนดตัวแปร

```
>>> x = 100
```

```
>>> y = 20
```

```
>>> x+y
```

120



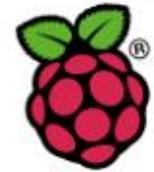
```
>>> x = 3.0
```

```
>>> y = 5
```

```
>>> x*y
```

15.0

ทดสอบเขียนโปรแกรมด้วย nano



เปิด editor

nano prog1.py ➔

เขียนโค้ด

print ("Innovative")
print ("Experiment")

```
GNU nano 2.2.6          File: Prog1.py          Modified  
  
print ("Innovative")  
print ("Experiment")  
  
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U Uncut Text ^T To Spell
```

บันทึก แล้วออก

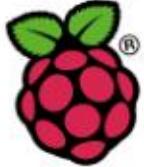
CTRL + X

Y

ENTER

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No ^C Cancel

File Name to Write: Prog1.py
^G Get Help M-D DOS Format
^C Cancel M-M Mac Format



Graphic Mode : Geany

โปรแกรม Editor ที่ช่วยให้การเขียน Python ง่ายขึ้น

sudo apt-get install geany



ติดตั้งโปรแกรมผ่าน

Internet



Prog2.py - /home/pi - Geany

File Edit Search View Document Project Build Tools Help

Symbols Documents Prog2.py

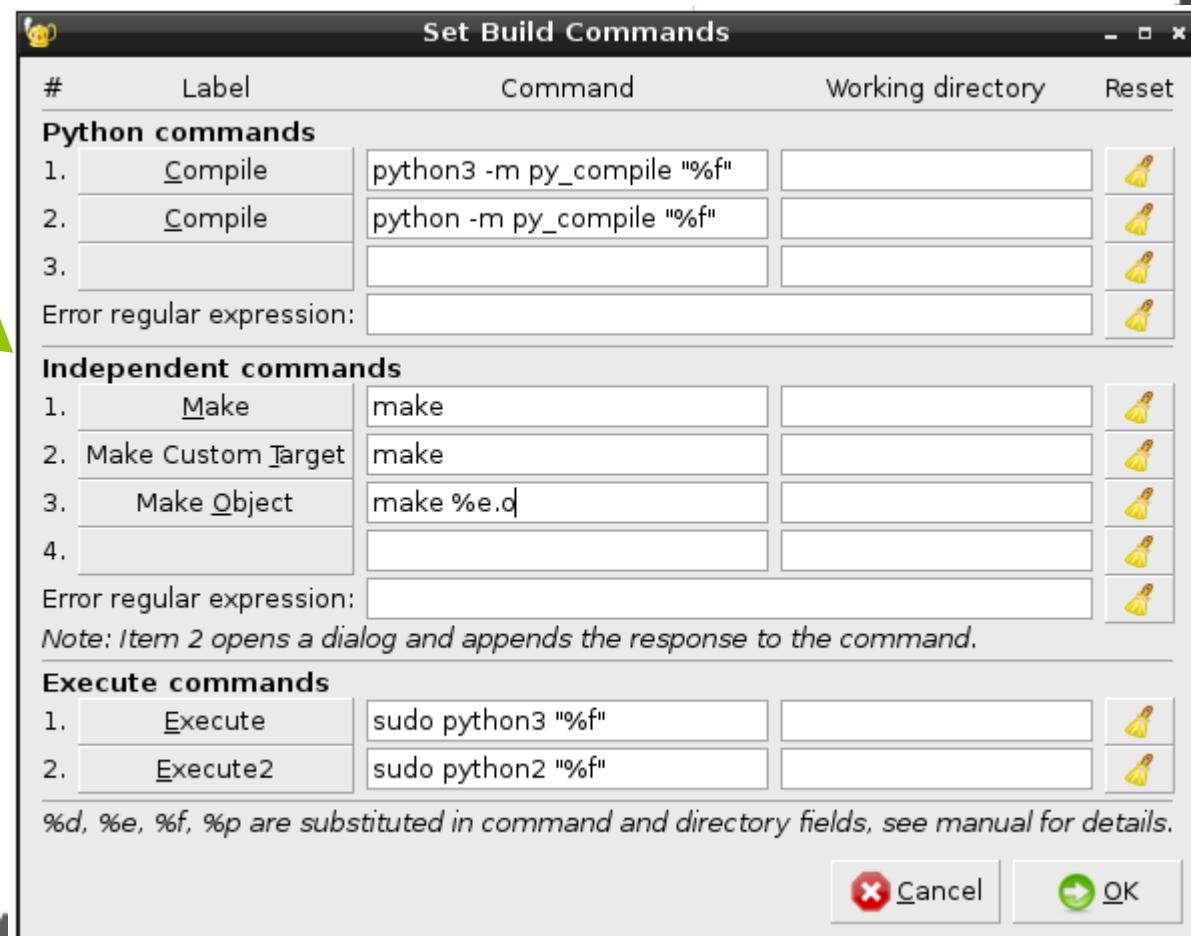
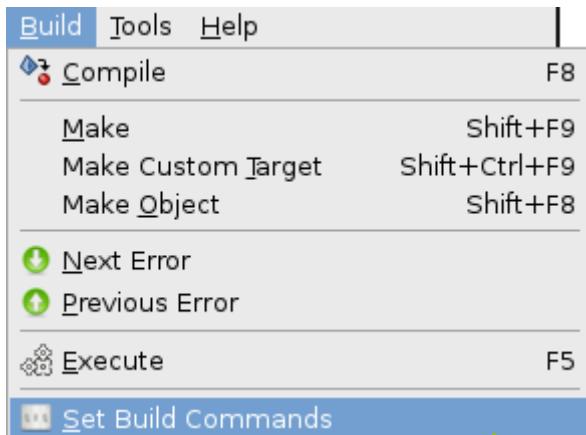
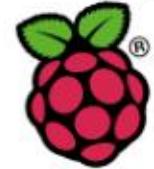
Variables Imports

1 import RPi.GPIO as GPIO
2 import time
3 x=0
4 while (True) :
5 print (x)
6 time.sleep(0.5)
7 x=x+1

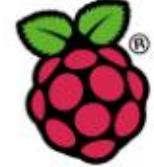
Status: python -m py_compile "Prog2.py" (in directory: /home/pi)
Compiler: Compilation finished successfully.
Messages
Scribble
Terminal

line: 10 / 10 col: 0 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 ...

ตั้งให้คอมไพล์ด้วย python3



ทดสอบโปรแกรม python ผ่าน Geany



```
import time
x=0
while (True) :
    print (x)
    time.sleep(0.5)
    x=x+1
```

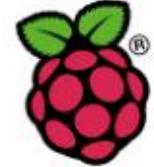
LXTerminal

```
File Edit Tabs Help
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

กด F5 RUN

กด CTRL+C ออก

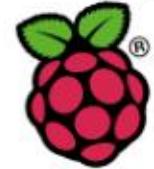
การเปรียบเทียบ



>	มากกว่า
\geq	มากกว่าหรือเท่ากับ
\leq	น้อยกว่าหรือเท่ากับ
<	น้อยกว่า
$=$	เท่ากับ
\neq	ไม่เท่ากับ
in	ตรงกับค่าใน List หรือเปล่า

AND จริงทั้งคู่
OR จริงอันใดอันหนึ่ง
NOT สลับจริงเป็นเท็จ

การวนทำซ้ำ while



```
while Condition:  
    Statements
```

.....
.....

ตัวอย่าง

```
x=1  
while x<5 :  
    print ('Less')  
    x=x+1
```

ออกจากการวนซ้ำได้ด้วย break

LXTerminal

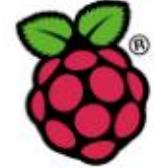
File Edit Tabs Help

Less
Less
Less
Less

(program exited with code: 0)
Press return to continue

A screenshot of a terminal window titled "LXTerminal". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The main area of the terminal shows the word "Less" printed four times on a black background. Below this, there is a dashed horizontal line followed by the text "(program exited with code: 0)" and "Press return to continue". A small gray square cursor is visible at the bottom of the terminal window.

การวนทำซ้ำ while



ตัวอย่าง

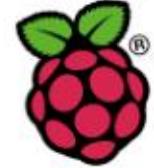
```
s='0'  
while s!='x' :  
    s = input('Press key= ')  
    print (s)
```

คำสั่ง `input` รอรับการกดคีย์เก็บในตัวแปรสตริง

STR

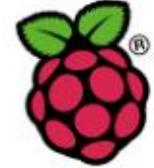
```
LXTerminal  
File Edit Tabs Help  
Press key= 42  
42  
Press key= 5  
5  
Press key= 1  
1  
Press key= 11  
11  
Press key= 1221  
1221  
Press key= x  
x  
-----  
(program exited with code: 0)  
Press return to continue
```

การควบคุมทิศทางของโปรแกรม



1. การวนทำซ้ำ (while)
2. การตัดสินใจ (if)
3. การจัดการความผิดปกติของโปรแกรม (error)

การตัดสินใจ : IF



```
if Condition:  
    Statements
```

.....

.....

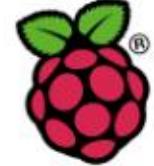
ตัวอย่าง

```
import time  
x=0  
while (True) :  
    if (x is 10) :  
        break  
    print (x)  
    time.sleep(0.2)  
    x=x+1
```

LXTerminal

```
File Edit Tabs Help  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
-----  
(program exited with code: 0)  
Press return to continue
```

การตัดสินใจ : if else



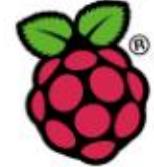
```
if Condition:  
    Statements  
    .....  
    .....  
else:  
    Statements  
    .....  
    .....
```

ตัวอย่าง

```
import time  
x=1  
while True :  
    if (x>5) :  
        print (x*2)  
    else :  
        print (x*5)  
    time.sleep(0.5)  
    x=x+1
```

A screenshot of a terminal window titled "LXTerminal". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal displays a series of numbers: 5, 10, 15, 20, 25, 12, 14, 16, 18, 20, 22, 24, and 26. These numbers are printed in pairs, where the first number in each pair is odd and the second is even, likely demonstrating the output of the provided Python code which prints x*2 for odd x and x*5 for even x.

การตัดสินใจ : if elif else

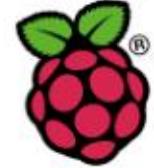


```
if Condition:  
    Statements  
    ....  
    ....  
  
elif Condition:  
    Statements  
    ....  
    ....  
  
else:  
    Statements  
    ....  
    ....
```

ตัวอย่าง

```
import time  
x=1  
while True :  
    if (x<5) :  
        print (x*2)  
    elif (x>=5 and x<10) :  
        print (x*5)  
    else :  
        break  
    time.sleep(0.5)  
    x=x+1
```

การวนทำซ้ำด้วย for



```
for var in range(m, n [, step = 1]):  
    Statements
```

.....

.....

for ตัวแปร in ชุดข้อมูล :

คำสั่ง 1

คำสั่ง 2

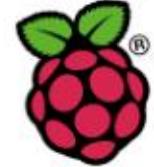
ตัวอย่าง

```
for i in range(0,10) :  
    if i%2 == 0 :  
        print (i, 'even')  
    else :  
        print (i, 'odd')
```

The screenshot shows a terminal window titled "LXTerminal". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The main area of the terminal displays the following text:

```
File Edit Tabs Help  
0 even  
1 odd  
2 even  
3 odd  
4 even  
5 odd  
6 even  
7 odd  
8 even  
9 odd  
-----  
(program exited with code: 0)  
Press return to continue
```

ฟังก์ชันคณิตศาสตร์ภายใน ที่สำคัญ



range(start, end [,step]) กำหนดค่าช่วงของตัวเลข

```
range(1,6) # จะประกอบด้วย 1,2,3,4,5
```

sum(n1,n2 [,nn]) หาผลรวมทั้งหมดในชุด

```
sum(1,2,3,4,5) # ผลลัพธ์คือ 15
```

round(var, digits) กำหนดจำนวนทศนิยม

```
round(1234.56789, 2) # ผลลัพธ์คือ 1234.57
```

min(var), max(var) หาค่าต่ำสุด สูงสุด ของเลขในชุด

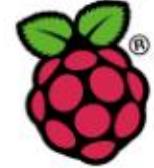
```
min(range(5,10)) # ผลลัพธ์คือ 5
```

```
max(range(5,10)) # ผลลัพธ์คือ 9
```

abs(var) แสดงค่าสัมบูรณ์ (เปลี่ยนเป็นค่าบวก)

```
abs(-50) # ผลลัพธ์คือ 50
```

การดักจับ error

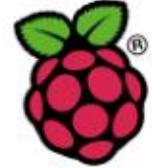


```
try:  
    Standard operation  
except:  
    Error operation  
finally:  
    End operation
```

```
try :  
    x=y  
except :  
    print ('y not set')
```

```
try:  
    main()  
finally:  
    GPIO.cleanup()  
    print("END")
```

การสร้างฟังก์ชัน



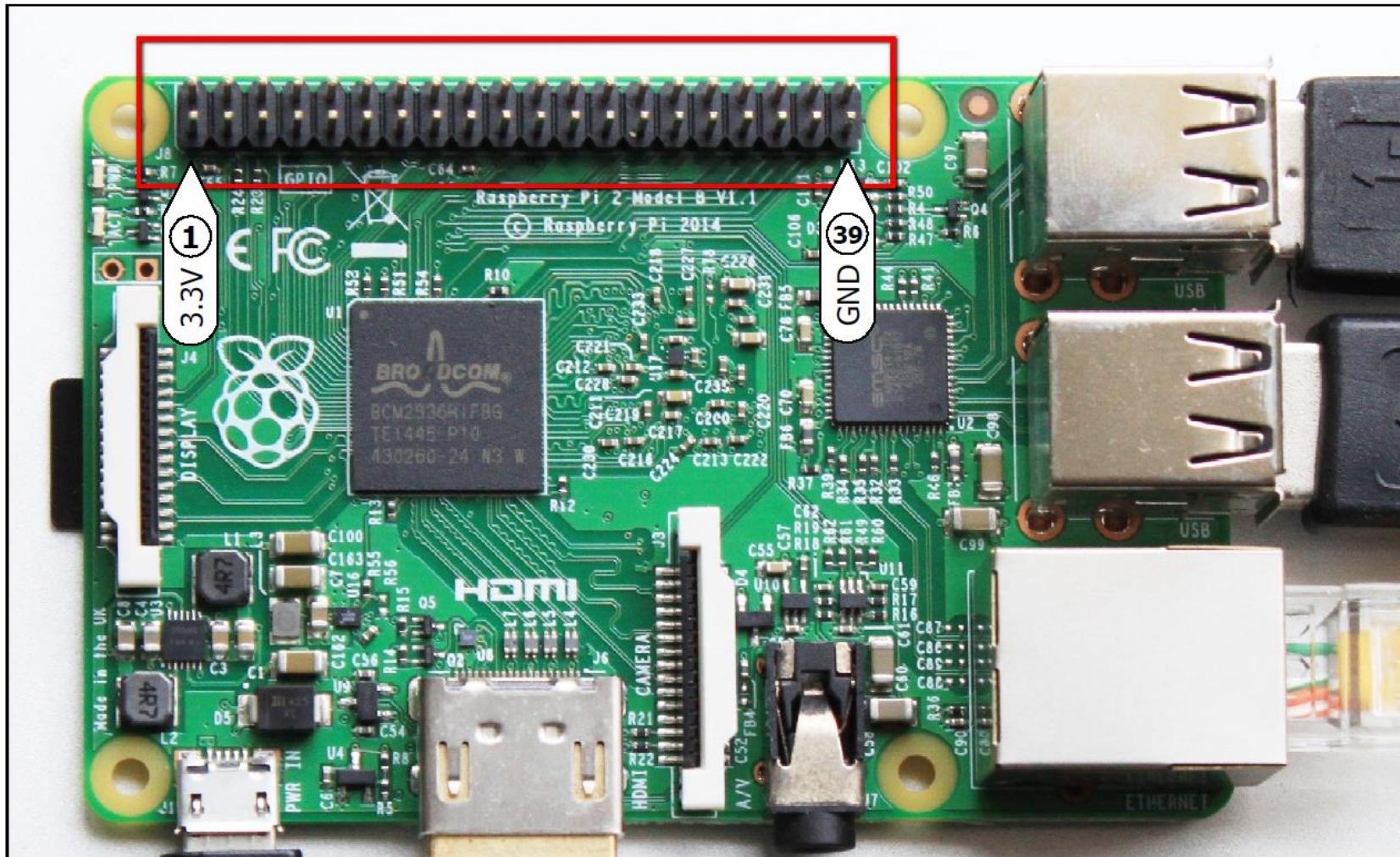
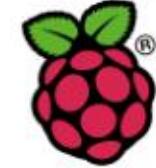
```
def function_name( [Argument] ):  
    Statement  
    .....  
    [return]
```

def	บอกว่าเป็นฟังก์ชัน
function_name	ชื่อฟังก์ชัน
Argument	ค่าที่ส่งเข้าไปในฟังก์ชัน
Statement	ชุดคำสั่ง
return	การคืนค่าของฟังก์ชัน

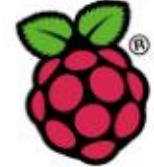
```
def add(x,y):  
    return x + y
```

```
add(3,4)
```

การนับขาของ Rpi



ตำแหน่งของ GPIO ของ Rpi



	+3.3V	1	○ ○	2	+5V
SDA1 -I2C	GPIO2	3	○ ○	4	+5V
SCL1 -I2C	GPIO3	5	○ ○	6	GND
	GPIO4	7	○ ○	8	GPIO14 TxD -UART0
	GND	9	○ ○	10	GPIO15 RxD -UART0
	GPIO17	11	○ ○	12	GPIO18 PCM-CLOCK
	GPIO27	13	○ ○	14	GND
	GPIO22	15	○ ○	16	GPIO23
	+3.3V	17	○ ○	18	GPIO24
MOSI -SPI0	GPIO10	19	○ ○	20	GND
MISO -SPI0	GPIO9	21	○ ○	22	GPIO25
SCLK-SPI0	GPIO11	23	○ ○	24	GPIO8 CE0 -SPI0
	GND	25	○ ○	26	GPIO7 CE1-SPI0
	ID SDA	27	○ ○	28	ID SCL
	GPIO5	29	○ ○	30	GND
	GPIO6	31	○ ○	32	GPIO12
	GPIO13	33	○ ○	34	GND
	GPIO19	35	○ ○	36	GPIO16
	GPIO26	37	○ ○	38	GPIO20
	GND	39	○ ○	40	GPIO21

GPIO 26 ขา

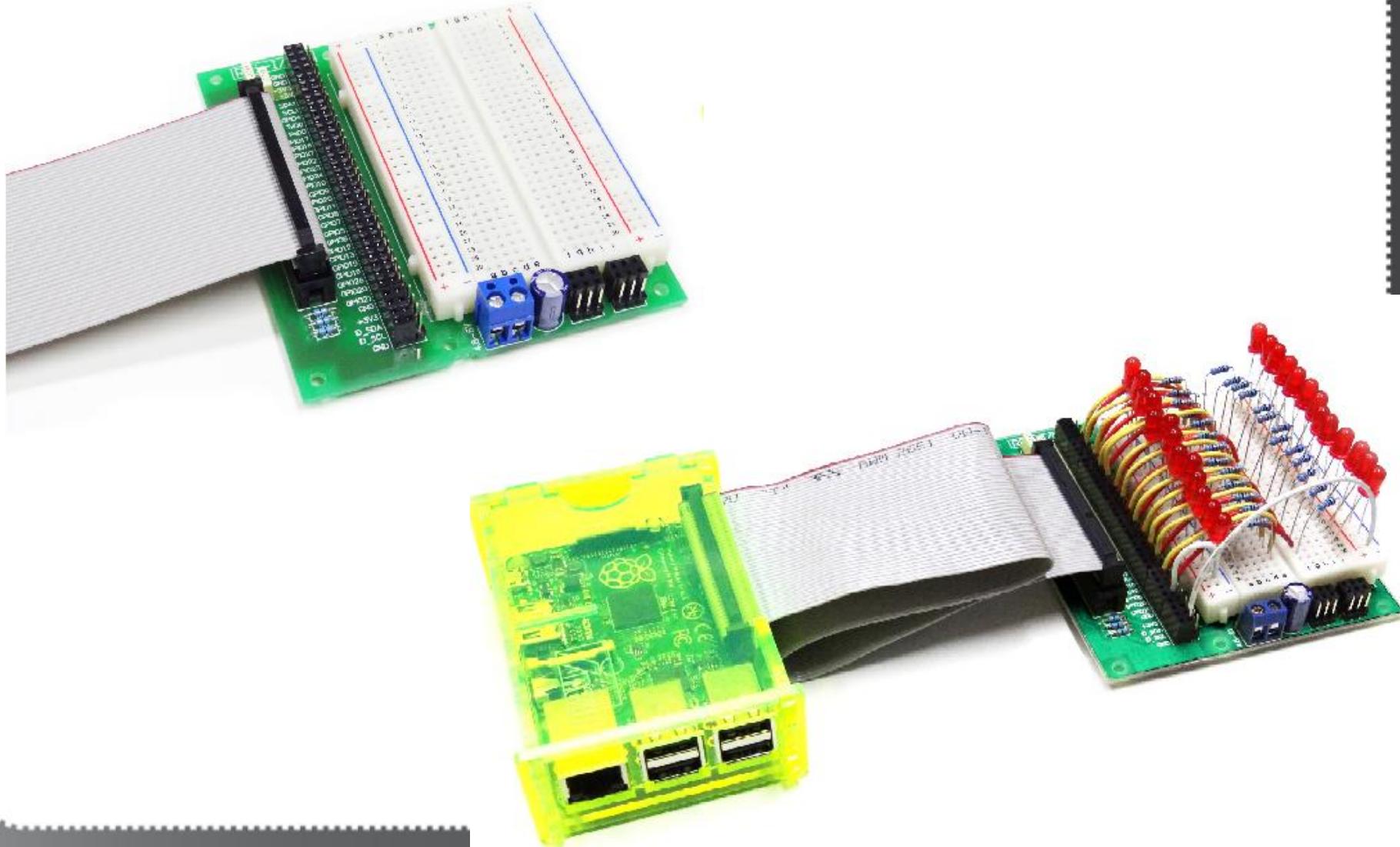
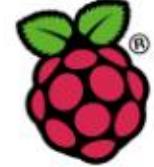
PWM ทุกขา

I2C /UART/SPI

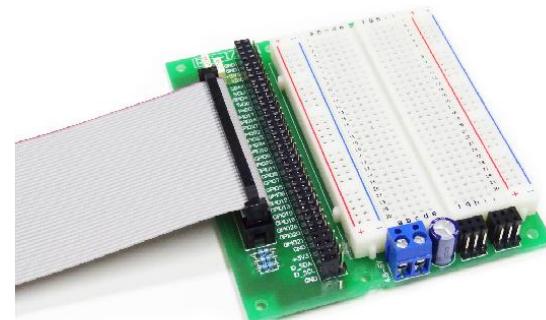
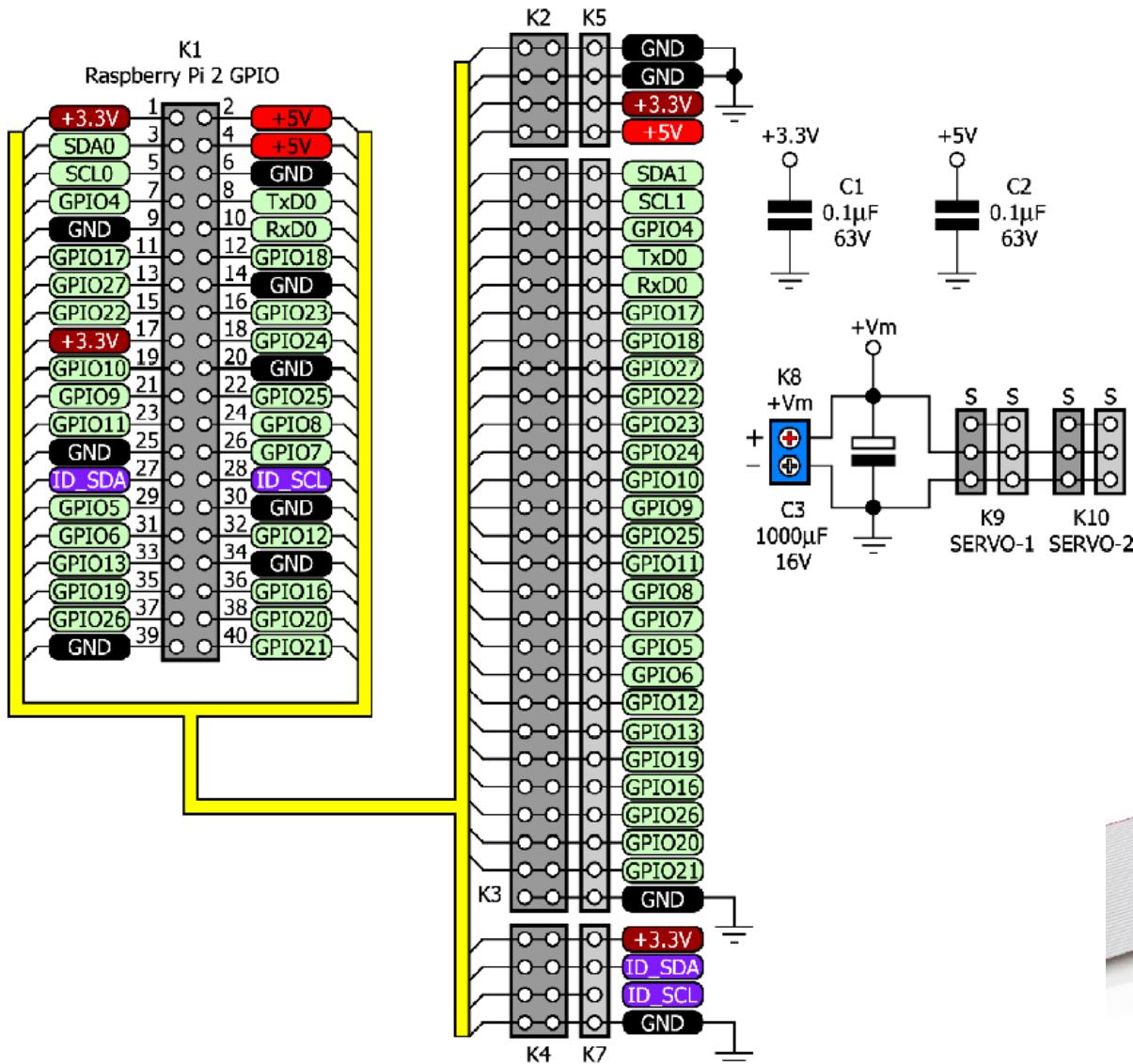
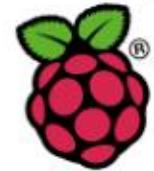
ការរំបូងប្រព័ន្ធគ្នា

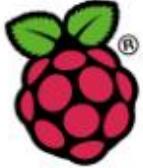


บอร์ด Rpi-I/O 40 และการเชื่อมต่อ

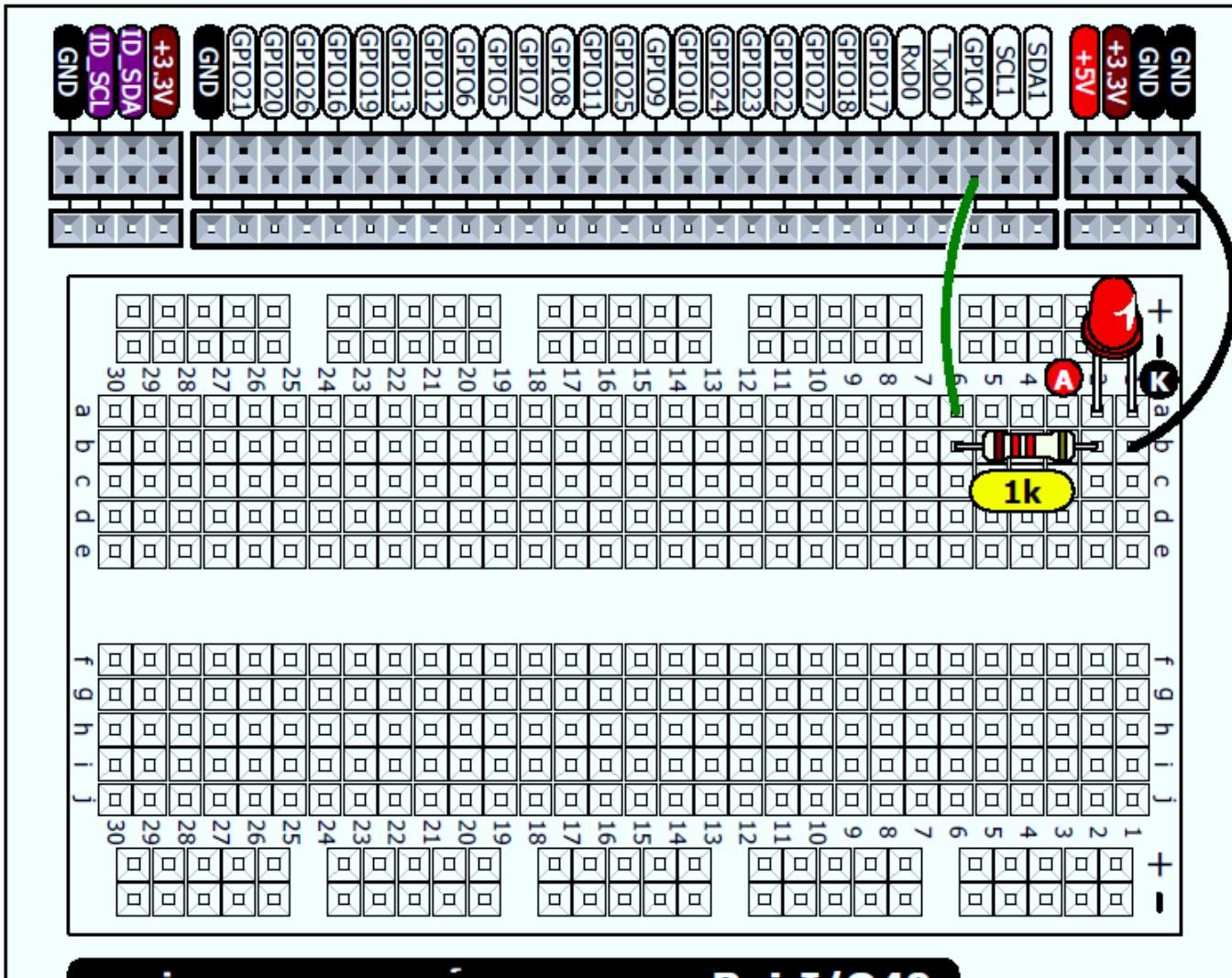


การระบุตำแหน่งขาของ Rpi-I/O 40



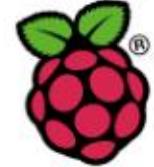


รูปแบบการต่อวงจรกับ Rpi-I/O 40



การต่อวงจรแบบเบตดบอร์ดของแพงวชจ Rpi-I/O40

ไลบรารี RPi.GPIO



เพิ่มไลบรารีในชื่อ GPIO

```
import RPi.GPIO as GPIO
```

setmode กำหนดรูปแบบขา

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setmode(GPIO.BOARD)
```

setup กำหนด input/output

```
GPIO.setup(pin,GPIO.OUT)
```

```
GPIO.setup(pin,GPIO.IN)
```

ปิดการแจ้งเตือน

```
GPIO.setwarnings(0)
```

output ส่งค่าออกไปที่ขา

```
GPIO.output(pin,1)
```

```
GPIO.output(pin,0)
```

input รับค่าล้อจิกจากขา

```
x = GPIO.input(pin)
```

cleanup เคลียร์ค่าพอร์ตกลับ

```
GPIO.cleanup()
```

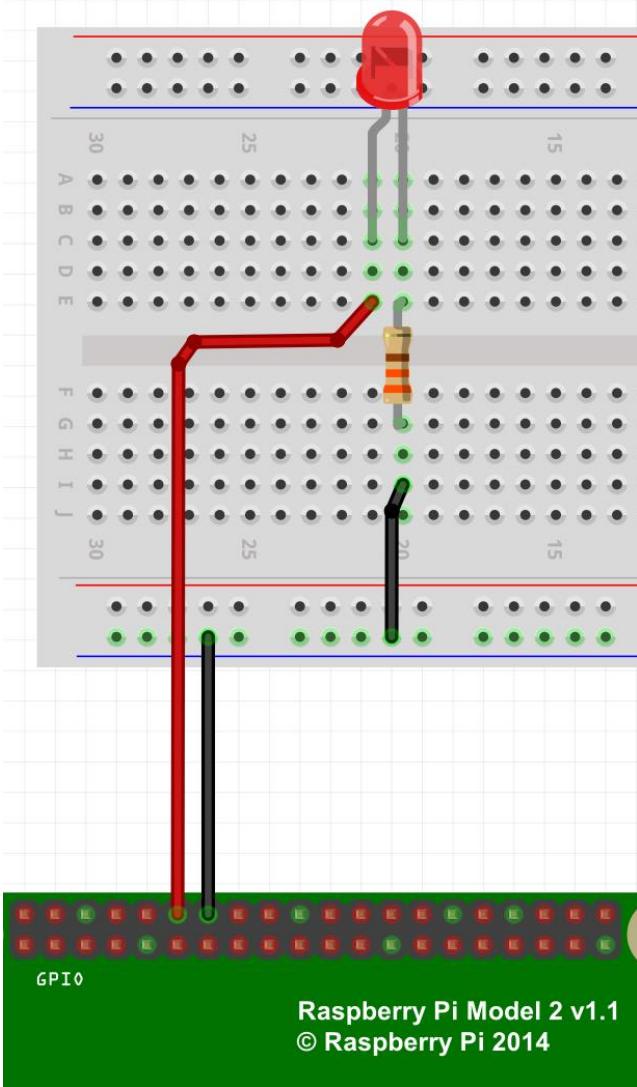
PWM เปิดการทำงาน PWM

```
GPIO.PWM(pin,freq)
```

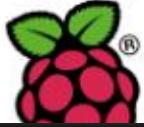
ไฟกะพริบที่ขา 18



```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
while True :
    GPIO.output(18,True)
    time.sleep(0.3)
    GPIO.output(18,False)
    time.sleep(0.3)
```



Cleanup Port หลังเลิกใช้

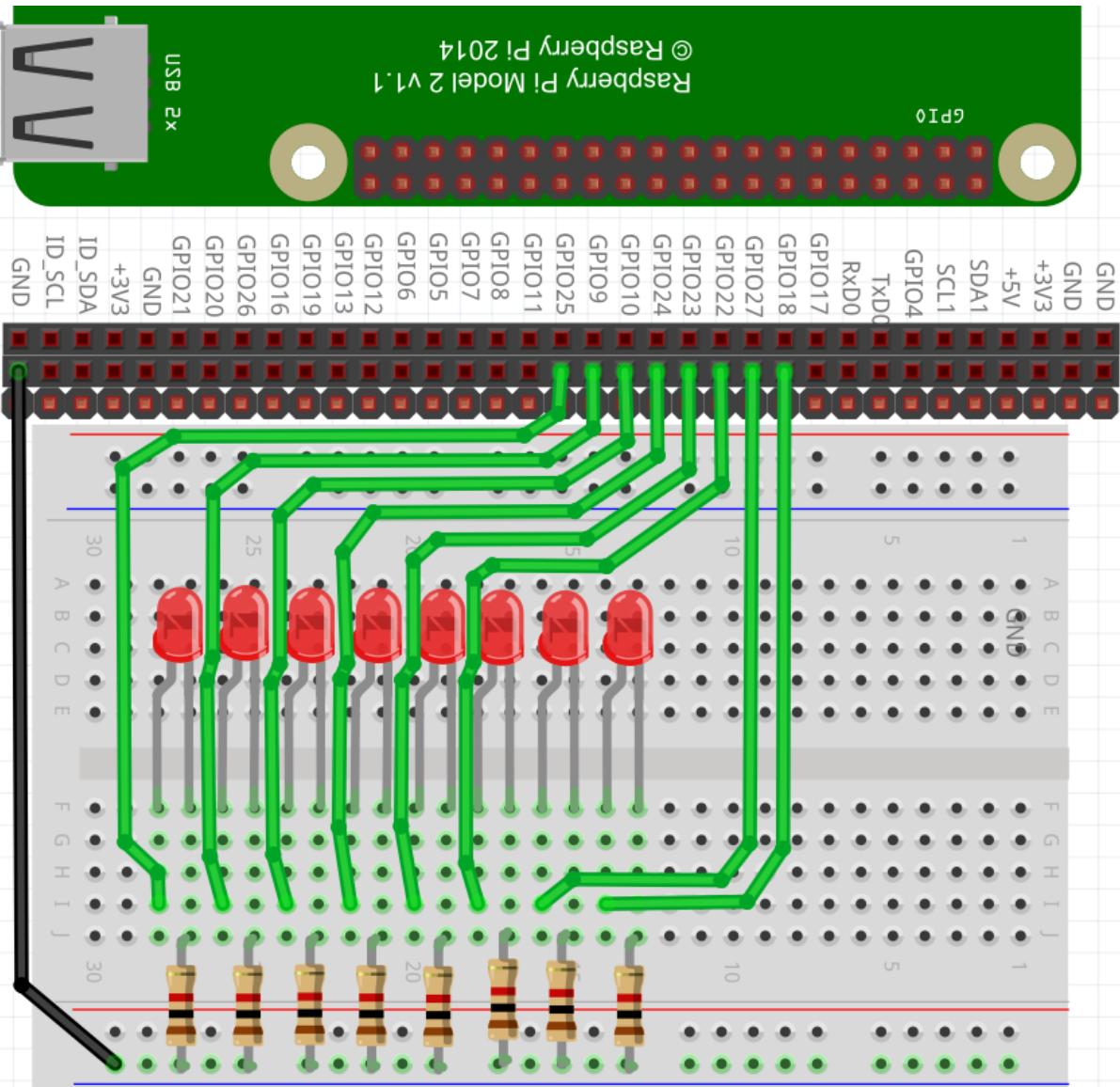
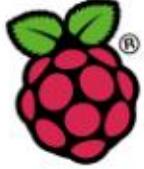


File Edit Tabs Help

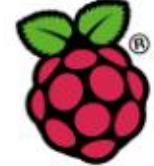
```
Ex-01.py:4: RuntimeWarning: This channel is already in use, continuing anyway.  
Use GPIO.setwarnings(False) to disable warnings.  
    GPIO.setup(18,GPIO.OUT)
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(18,GPIO.OUT)  
try :  
    while True :  
        GPIO.output(18,True)  
        time.sleep(0.3)  
        GPIO.output(18,False)  
        time.sleep(0.3)  
finally :  
    GPIO.cleanup()
```

การจ่าย LED 8 ดวง



ขับ LED ไฟกะพริบ 4 ดวง แบบที่ 1



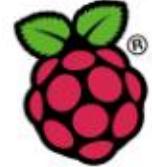
```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)
GPIO.setup(22,GPIO.OUT)
GPIO.setup(23,GPIO.OUT)

try:
    while (True) :
        GPIO.output(18,1)
        GPIO.output(27,1)
        GPIO.output(22,1)
        GPIO.output(23,1)
        time.sleep(0.5)
        GPIO.output(18,0)
        GPIO.output(27,0)
        GPIO.output(22,0)
        GPIO.output(23,0)
        time.sleep(0.5)

finally:
    GPIO.cleanup()
```

คำสั่ง For และตัวแปรแบบ List



Python ไม่ใช้ Array ใช้ List แทน หลายชนิดอยู่ที่เดียวกันได้

```
>>> i = [1, 2, 3, 4, 'Rasp', 5, 'Pi']
```

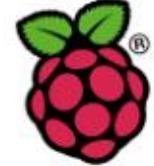
```
pins = [18,27,22,23]
```

↑
ประกาศตัวแปรแบบ List

```
pins = [18,27,22,23]
for x in pins:
    GPIO.setup(x,GPIO.OUT)
```

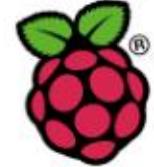
in ดึงค่าใน pins ออกมาทีละตัวเก็บใน x

ขับ LED ไฟกะพริบ 4 ดวง แบบที่ 2



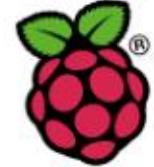
```
import RPi.GPIO as GPIO
import time
pins = [18,27,22,23]
GPIO.setmode(GPIO.BCM)
for x in pins :
    GPIO.setup(x,GPIO.OUT)
try:
    while (True) :
        for x in pins :
            GPIO.output(x,1)
            time.sleep(0.5)
        for x in pins :
            GPIO.output(x,0)
            time.sleep(0.5)
finally:
    GPIO.cleanup()
```

ขับ LED ไฟกะพริบ 8 ดวง



```
import RPi.GPIO as GPIO
import time
pins = [18,27,22,23,24,10,9,25]
GPIO.setmode(GPIO.BCM)
for x in pins :
    GPIO.setup(x,GPIO.OUT)
try:
    while (True) :
        for x in pins :
            GPIO.output(x,1)
            time.sleep(0.5)
        for x in pins :
            GPIO.output(x,0)
            time.sleep(0.5)
finally:
    GPIO.cleanup()
```

ขับ LED ไฟวิ่ง



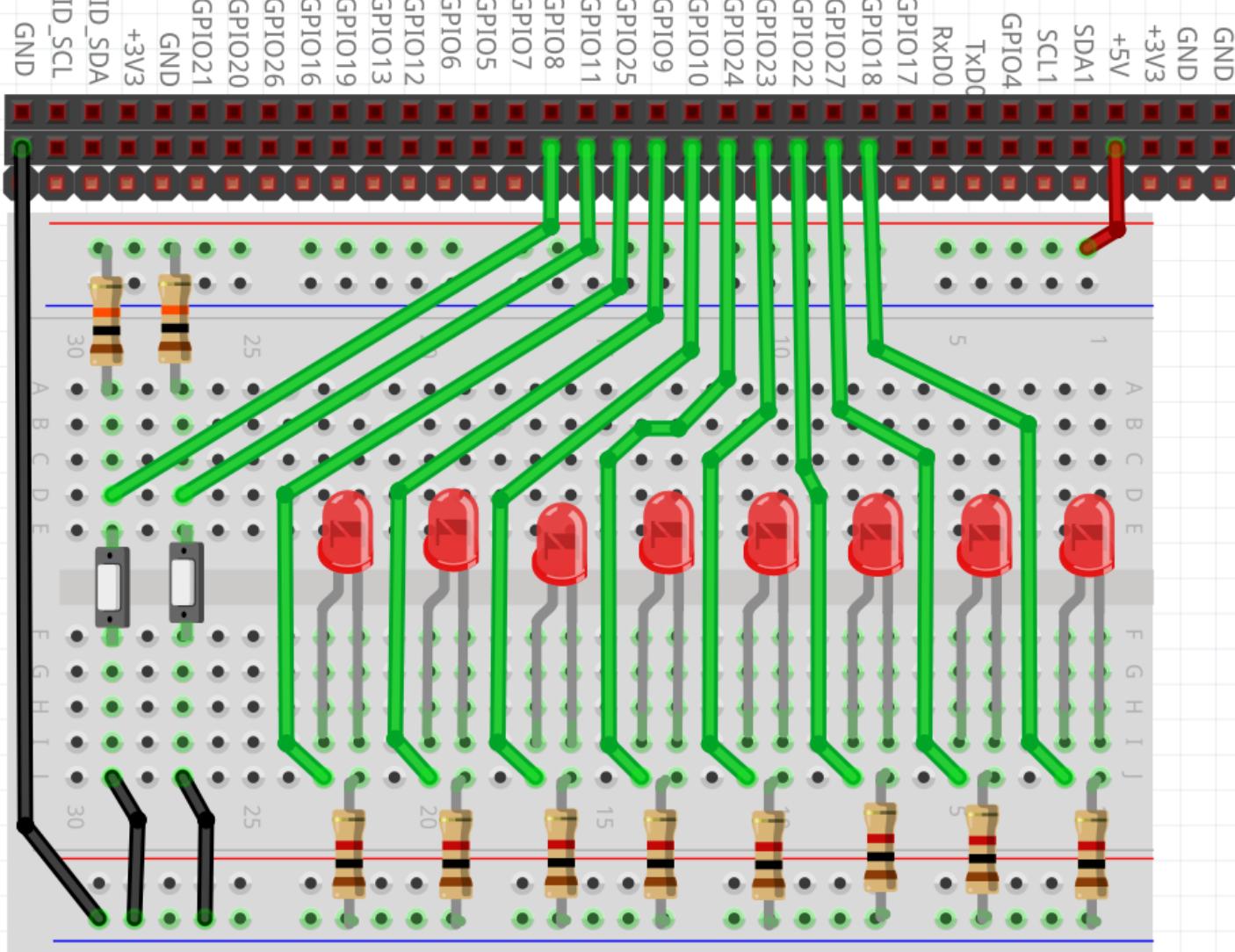
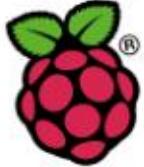
```
import RPi.GPIO as GPIO
import time
pins = [18,27,22,23,24,10,9,25]
GPIO.setmode(GPIO.BCM)
for x in pins :
    GPIO.setup(x,GPIO.OUT)
try:
    while (True) :
        for x in pins :
            GPIO.output(x,1)
            time.sleep(0.5)
        for x in pins :
            GPIO.output(x,0)
            time.sleep(0.5)
finally:
    GPIO.cleanup()
```

ทดลองควบคุมสีสั่ง Random

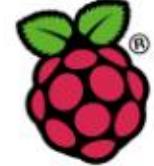


```
import random
import RPi.GPIO as GPIO
import time
pin =[18,27,22,23,24,10,9,25]
GPIO.setmode(GPIO.BCM)
for i in pin : GPIO.setup(pin,GPIO.OUT)
try:
    while True :
        x=random.randint(0,7)
        GPIO.output(pin[x],1)
        time.sleep(0.2)
        for i in pin :
            GPIO.output(i,0)
finally :
    GPIO.cleanup()
```

การอ่านค่าอินพุตจากสวิตช์

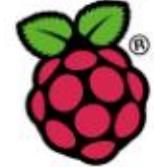


การอ่านค่าอินพุตจากสวิตช์



```
import RPi.GPIO as GPIO
import time
pins = [18,27,22,23,24,10,9,25]
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN)
GPIO.setup(8,GPIO.IN)
for x in pins :
    GPIO.setup(x,GPIO.OUT)
for x in pins :
    GPIO.output(x,0)
try:
    while (True) :
        if GPIO.input(11) : GPIO.output(18,1)
        else : GPIO.output(18,0)
        if GPIO.input(8) : GPIO.output(27,1)
        else : GPIO.output(27,0)
finally:
    GPIO.cleanup()
```

การกระทำการล้อจิก



and ต้องเป็นจริงทั้งคู่ จึงจะเป็นจริง

or เป็นจริง อันใดอันหนึ่ง เป็นจริง

not กลับ จริงเป็นเท็จ เท็จเป็นจริง

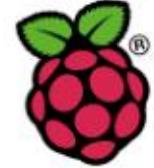
if GPIO.input(11)

if not GPIO.input(11)

if GPIO.input(11) or GPIO.input(8)

if GPIO.input(11) and GPIO.input(8)

ทดลองบากดสวิตซ์



```
import RPi.GPIO as GPIO
import time
pins = [18,27,22,23,24,10,9,25]
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN)
GPIO.setup(8,GPIO.IN)
for x in pins :
    GPIO.setup(x,GPIO.OUT)
for x in pins :
    GPIO.output(x,0)
try:
    while (True) :
        if GPIO.input(11) and GPIO.input(8) :
            GPIO.output(18,1)
        else :
            GPIO.output(18,0)
finally:
    GPIO.cleanup()
```

สั่ง Pull up Pull Down ด้วยซอฟต์แวร์



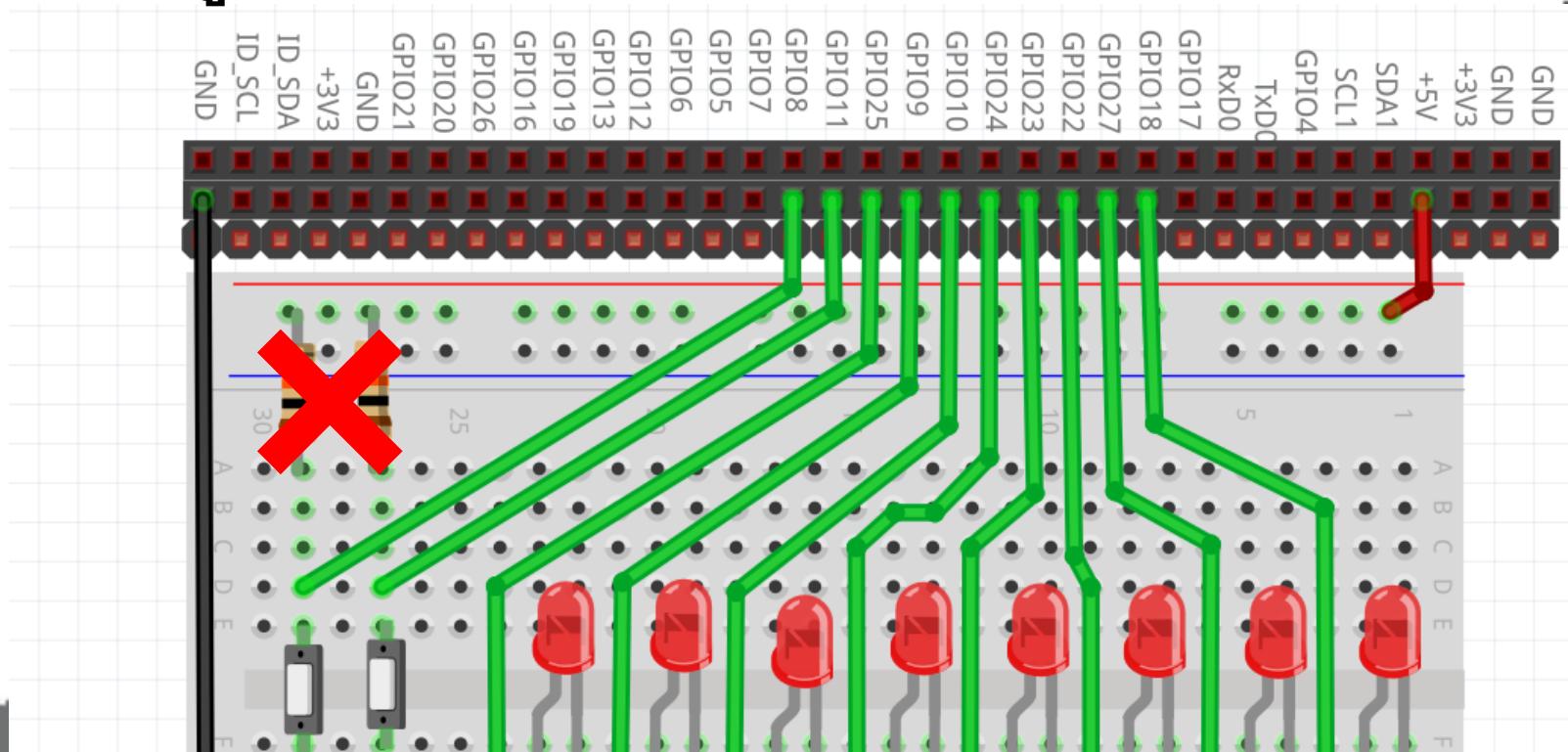
กำหนดให้ขา 20 เป็นขาอินพุต และมีการพูลอัพ

```
GPIO.setup(20,GPIO.IN,pull_up_down=GPIO.PUD_UP)
```

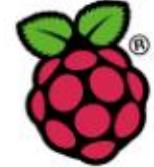
กำหนดให้ขา 21 เป็นขาอินพุต และมีการพูลดาวน์

```
GPIO.setup(21,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
```

50 กิโลโอห์ม พูลอัพ

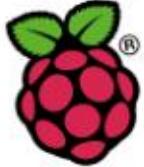


ทดสอบ พูลอัพสวิตช์ด้วยซอฟต์แวร์



```
import RPi.GPIO as GPIO
import time
pins = [18,27,22,23,24,10,9,25]
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(8,GPIO.IN,pull_up_down=GPIO.PUD_UP)
for x in pins :
    GPIO.setup(x,GPIO.OUT)
for x in pins :
    GPIO.output(x,0)
try:
    while (True) :
        if GPIO.input(11) and GPIO.input(8) :
            GPIO.output(18,1)
        else :
            GPIO.output(18,0)
finally:
    GPIO.cleanup()
```

อ่านค่าสวิตช์ขอบขาขึ้น ขอบขาลง



GPIO.wait_for_edge(23, GPIO.FALLING) ขาลง

GPIO.wait_for_edge(23, GPIO.RISING) ขาขึ้น

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(8, GPIO.IN, pull_up_down = GPIO.PUD_UP)
try :
    while True:
        GPIO.wait_for_edge(11, GPIO.FALLING)
        print('Button 1 Pressed')
        GPIO.wait_for_edge(11, GPIO.RISING)
        print('Button 1 Released')
        GPIO.wait_for_edge(8, GPIO.FALLING)
        print('Button 2 Pressed')
        GPIO.wait_for_edge(8, GPIO.RISING)
        print('Button 2 Released')
finally :
    GPIO.cleanup()
```

รูปแบบการสร้างฟังก์ชัน

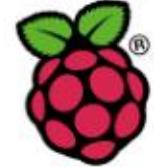


```
def function_name( [Argument] ):  
    Statement  
    .....  
    [return]
```

def	บอกว่าเป็นฟังก์ชัน
function_name	ชื่อฟังก์ชัน
Argument	ค่าที่ส่งเข้าไปในฟังก์ชัน
Statement	ชุดคำสั่ง
return	การคืนค่าของฟังก์ชัน

```
def add(x,y) :  
    return x + y
```

ทดลองใช้งานพีบีชีน

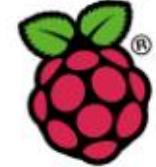


```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN)

def blink(CH,delay,cnt):
    GPIO.setup(CH,GPIO.OUT)
    for i in range(1,cnt):
        GPIO.output(CH,1)
        time.sleep(delay)
        GPIO.output(CH,0)
        time.sleep(delay)

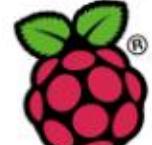
try :
    while GPIO.input(11):
        blink(18,0.5,5)
        time.sleep(2.0)
finally :
    GPIO.cleanup()
```

EVENTS AND CALLBACK FUNCTIONS



```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(18,GPIO.OUT)
def prints(CH) :
    print ('Button 1 Pressed')
try :
    GPIO.add_event_detect(11,GPIO.RISING,callback=prints,bouncetime=300)
    while True :
        GPIO.output(18,1)
        time.sleep(0.2)
        GPIO.output(18,0)
        time.sleep(0.2)
finally :
    GPIO.cleanup()
```

เพิ่มคำแนะนำให้เกิด Event



```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(8,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(7,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(5,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(18,GPIO.OUT)

def prints(CH) :
    print ('Button', CH, 'Pressed')

try :
    GPIO.add_event_detect(11,GPIO.RISING,callback=prints,bouncetime=300)
    GPIO.add_event_detect(8,GPIO.RISING,callback=prints,bouncetime=300)
    GPIO.add_event_detect(7,GPIO.RISING,callback=prints,bouncetime=300)
    GPIO.add_event_detect(5,GPIO.RISING,callback=prints,bouncetime=300)

    while True :
        GPIO.output(18,1)
        time.sleep(0.2)
        GPIO.output(18,0)
        time.sleep(0.2)
finally :
    GPIO.cleanup()
```

ปิด Event

GPIO.remove_event_detect(23)

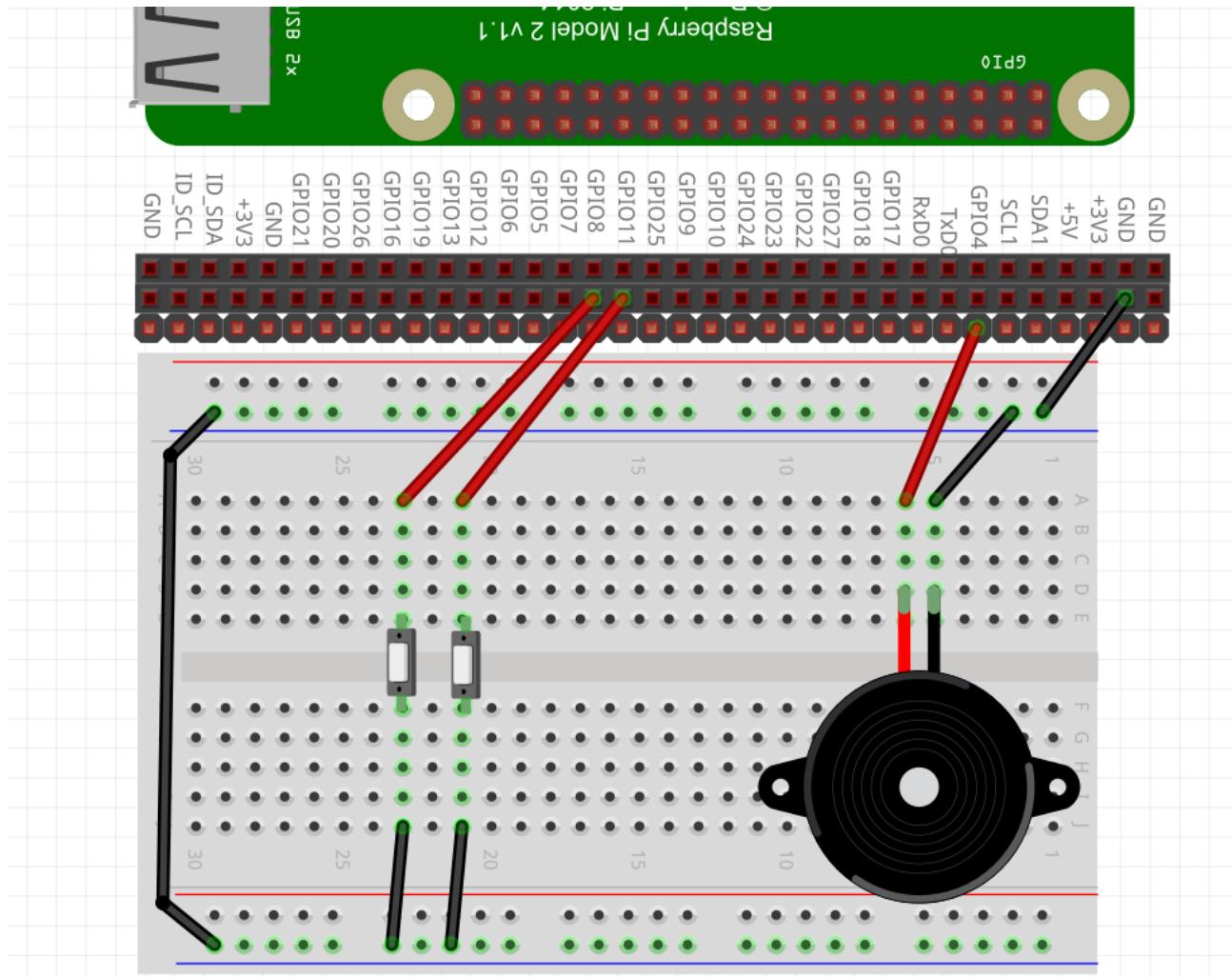
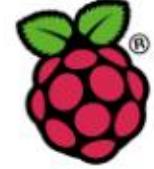
```

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(8,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(7,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(5,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(18,GPIO.OUT)
def prints(CH) :
    print ('Button', CH, 'Pressed')
try :
    GPIO.add_event_detect(11,GPIO.RISING,callback=prints,bouncetime=300)
    GPIO.add_event_detect(8,GPIO.RISING,callback=prints,bouncetime=300)
    GPIO.add_event_detect(7,GPIO.RISING,callback=prints,bouncetime=300)
    while True :
        GPIO.output(18,1)
        time.sleep(0.2)
        GPIO.output(18,0)
        time.sleep(0.2)
        if (not GPIO.input(5)) :
            print ('Close Event')
            GPIO.remove_event_detect(11)
            GPIO.remove_event_detect(8)
            GPIO.remove_event_detect(7)
finally :
    GPIO.cleanup()

```

ทดสอบปิด Event

สร้างเสียงออกลำโพง

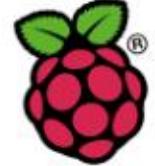


สร้างเสียงออกลำโพง



```
def sound(pins,freq,times):  
    GPIO.setup(pins,GPIO.OUT)  
    for i in range(0,times):  
        GPIO.output(pins,1)  
        time.sleep(0.5/freq)  
        GPIO.output(pins,0)  
        time.sleep(0.5/freq)
```

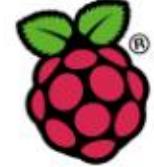
สร้างเสียงอุกอาจ pomoc



```
import RPi.GPIO as GPIO
import time

def sound(pins,freq,times):
    GPIO.setup(pins,GPIO.OUT)
    for i in range(0,times):
        GPIO.output(pins,1)
        time.sleep(0.5/freq)
        GPIO.output(pins,0)
        time.sleep(0.5/freq)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(11,GPIO.IN,pull_up_down= GPIO.PUD_UP)
try:
    while(1):
        if (not GPIO.input(11)) :
            sound(4,1000,100)
            sound(4,2000,100)
            sound(4,500,100)
finally:
    GPIO.cleanup()
```

PWM : พัลส์วิธมอดูเลเตอร์



เปิดการใช้ PWM

```
p=GPIO.PWM(pin,freq)
```

เริ่มต้น PWM

```
p.start(%dutycycle)
```

เปลี่ยนความถี่ PWM

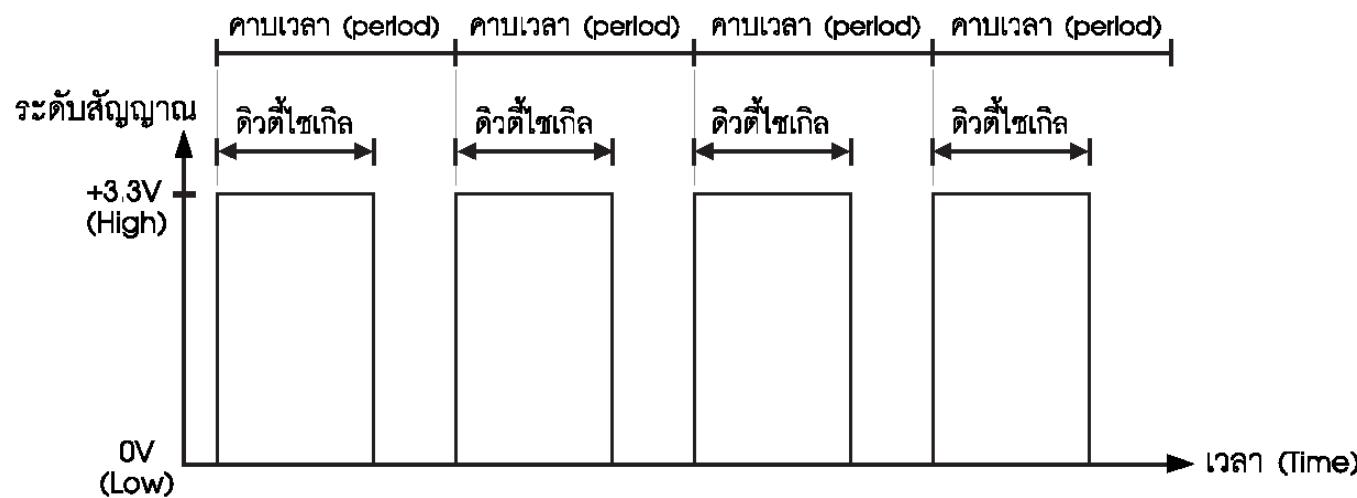
```
p.ChangeFrequency(freq)
```

เปลี่ยนค่าความกว้างพัลส์

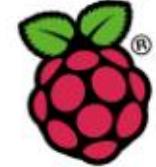
```
p.ChangeDutyCycle(%duty)
```

หยุด PWM

```
p.stop()
```



ไฟกะพริบด้วย PWM



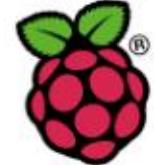
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
p=GPIO.PWM(18,1)← 1 Hz
p.start(50)
input('Press return to stop')
p.stop()
GPIO.cleanup()
```

ไฟกะพริบด้วย PWM

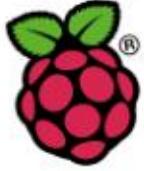


```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
p=GPIO.PWM(18,50)
p.start(0)
try:
    while True :
        for x in range(0,101,5) :
            p.ChangeDutyCycle(x)
            time.sleep(0.1)
        for x in range(100,-1,-5) :
            p.ChangeDutyCycle(x)
            time.sleep(0.1)
except KeyboardInterrupt:
    pass
p.stop()
GPIO.cleanup()
```

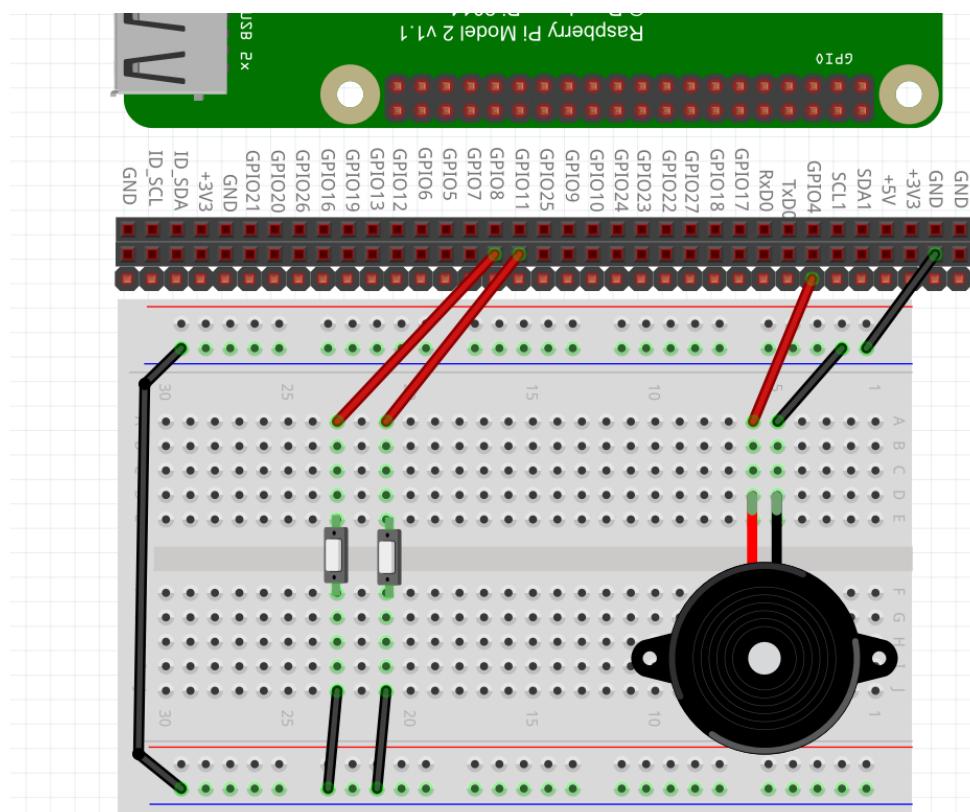
ໄຟຟ້າຕກ



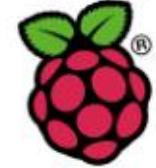
```
import RPi.GPIO as GPIO
import time
pin =[18,27,22,23,24,10,9,25]
p=[0,1,2,3,4,5,6,7,8]
GPIO.setmode(GPIO.BCM)
for i in pin : GPIO.setup(i,GPIO.OUT)
for i in range(0,8) : p[i]=GPIO.PWM(pin[i],50)
for i in range(0,8) : p[i].start(0)
try:
    while True :
        for i in range(0,8) :
            for x in range(0,101,5) :
                p[i].ChangeDutyCycle(x)
                time.sleep(0.01)
        for i in range(0,8) :
            for x in range(100,-1,-5) :
                p[i].ChangeDutyCycle(x)
                time.sleep(0.01)
except KeyboardInterrupt:
    pass
for i in range(0,8) : p[i].stop()
GPIO.cleanup()
```



```
import RPi.GPIO as GPIO
import time
x=2000
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)
GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(8,GPIO.IN,pull_up_down=GPIO.PUD_UP)
p=GPIO.PWM(4,2000)
p.start(50)
try:
    while True:
        if not GPIO.input(11):
            x=x+100
            print (x)
            p.ChangeFrequency(x)
            time.sleep(0.2)
        if not GPIO.input(8):
            if x>100:
                x=x-100
                print (x)
                p.ChangeFrequency(x)
                time.sleep(0.2)
finally:
    p.stop()
    GPIO.cleanup()
```

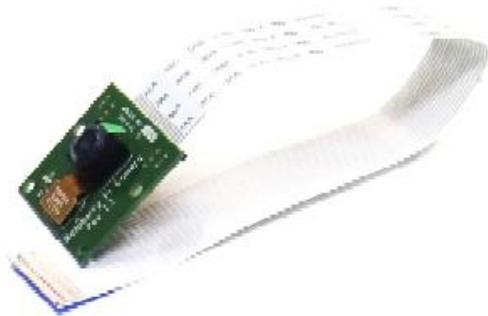
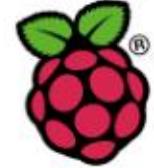


การใช้งานโมดูลกล้อง Pi-camera



- ✓ ความละเอียดถึง 5 ล้านพิกเซล
- ✓ เลนส์แบบ Fixed-Focus
- ✓ ภาพขนาดสูงสุด 2592x1944 พิกเซล
- ✓ ภาพเคลื่อนไหวคุณภาพ ระดับ 1080p
- ✓ 30 เฟรมต่อวินาที (1080p)
- ✓ ติดต่อกับบอร์ด Raspberry Pi ด้วย
บัส CSI (Common System Interface)

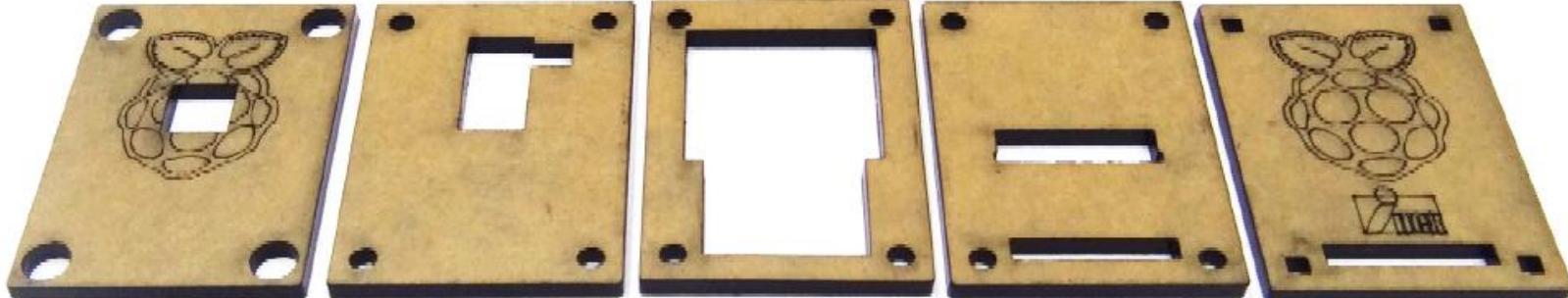
เคสอะคริลิกป้องกันตัวกล้อง



ไม่ดูกล้อง Raspberry Pi



สกรูหัวตัด 3x15 มม. 4 ตัว



1

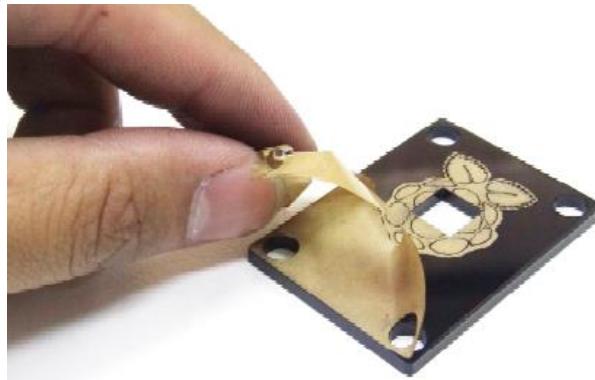
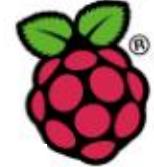
2

3

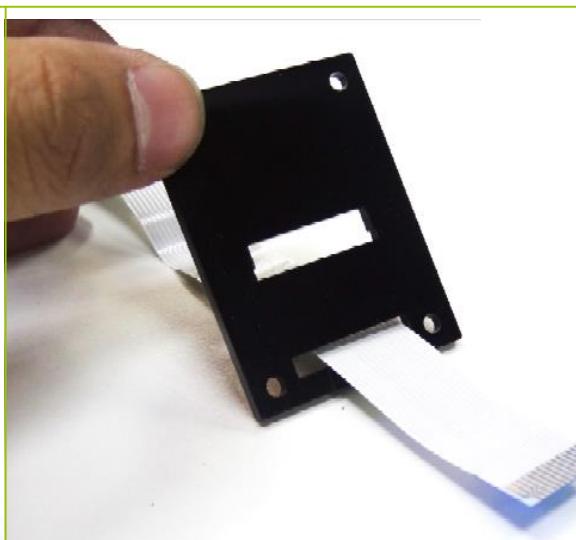
4

5

การประกอบเคสอะคริลิก



1. แกะกระดาษออกก่อน

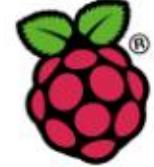


2. วางกล้องลงไป

3. สอดสาย

4. ประกอบเข้าไป

การประกอบเคสอะคริลิก (ต่อ)



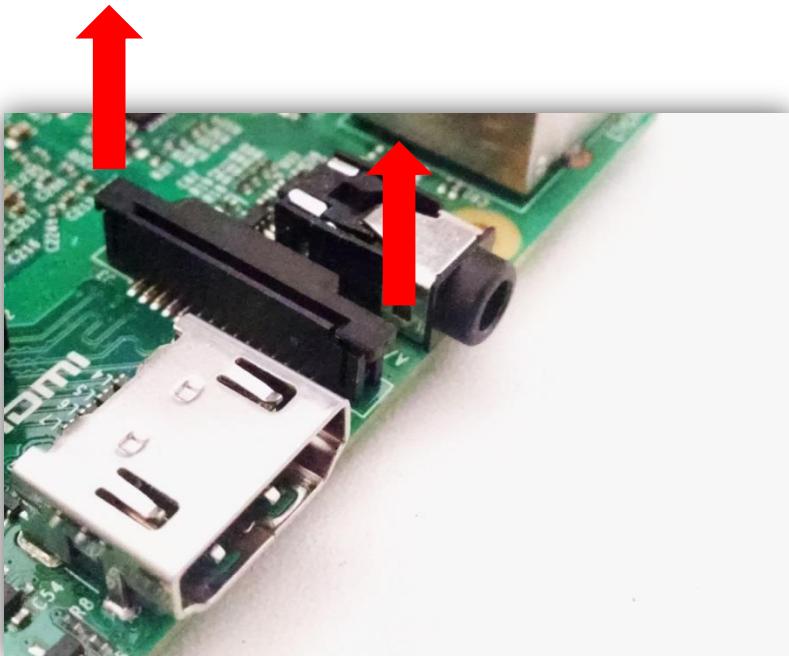
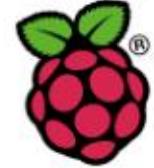
5. ประกอบชิ้นที่ 5 โดยสอดสายเข้าไป



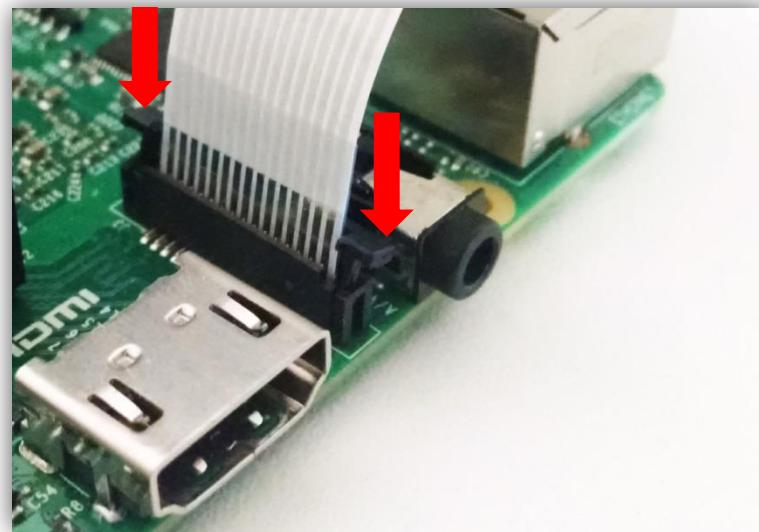
6. ใช้สกรูขันยึดให้แน่น



การติดตั้งกล้อง

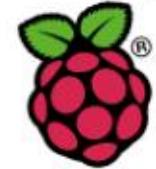


ถึงขาล็อกขึ้นก่อน

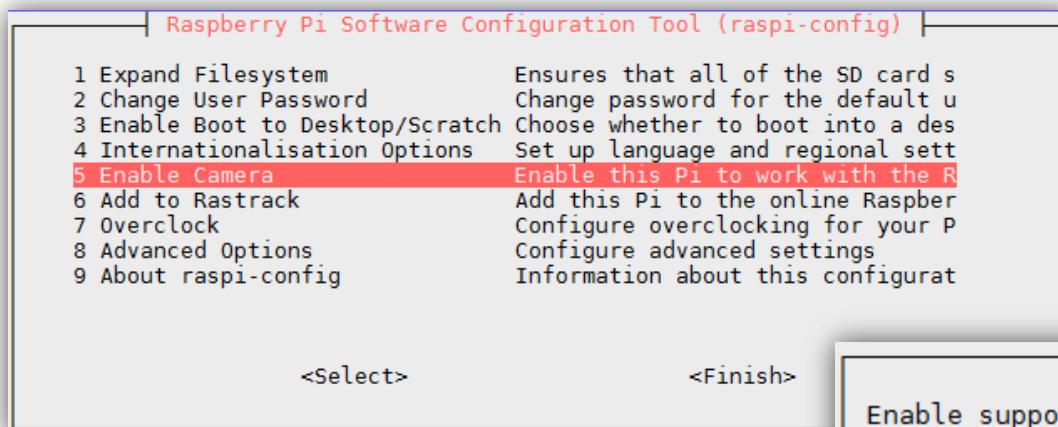


เสียบสายแพ แล้วกด
ขาล็อกให้แน่น

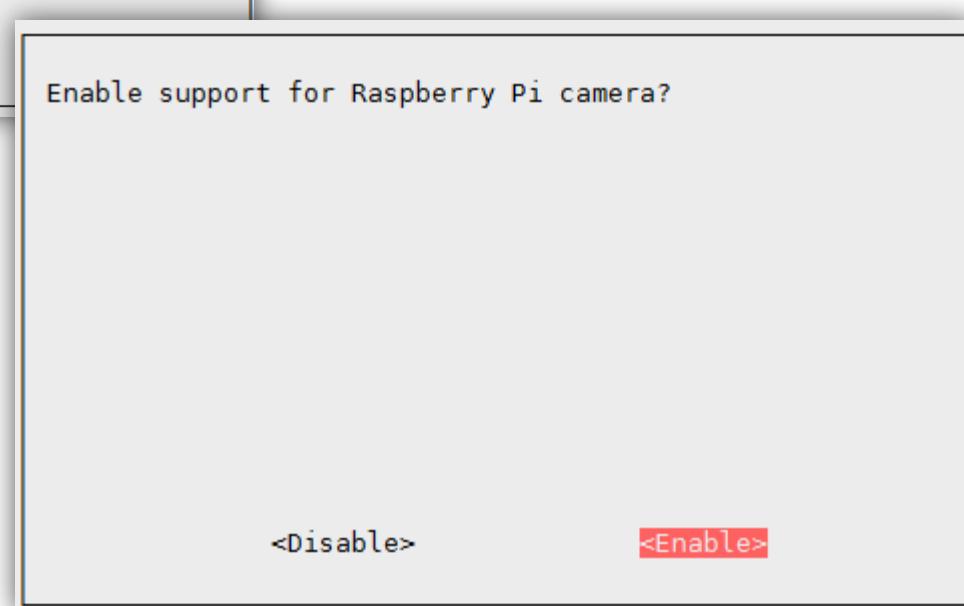
การตั้งค่าเพื่อติดต่อโมดูลกล้อง



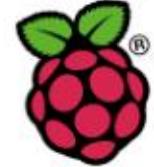
pi@raspberrypi ~ \$ sudo raspi-config



Enable Camera > Enable



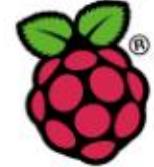
คำสั่งควบคุม荷模ภาพนิ่ง



● **raspistill** เปิดโมดูลกล้องเพื่อถ่ายภาพนิ่ง

- --width , -w ปรับขนาดความกว้างของภาพ
 - --height , -h ปรับขนาดความสูงของภาพ
 - --quality , -q ปรับแต่งคุณภาพของรูปที่ถ่าย ค่า 0-100
 - --output , -o บันทึกภาพในชื่อไฟล์ที่กำหนด
- เช่น **raspistill -o image.jpg**

คำสั่งควบคุมโหมดภาพนิ่ง (ต่อ)



--timeout , -t ใช้กำหนดช่วงเวลาเปิดโมดูลกล้อง ปกติ 5 วินาที

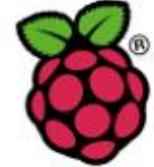
เช่น raspistill -t 10000 (สั่งเปิดโมดูลกล้องนาน 10 วินาที)

--verbose , -v แสดงรายละเอียดต่าง ๆ ออกแบบและเก็บภาพ

```
raspistill Camera App v1.3.8
Width 2592, Height 1944, quality 85, filename x%08d.jpg
Time delay 5000, Raw no
Thumbnail enabled Yes, width 64, height 48, quality 35
Link to latest frame enabled no
Full resolution preview No
Capture method : Single capture

Preview Yes, Full screen Yes
Preview window 0,0,1024,768
Opacity 255
Sharpness 0, Contrast 0, Brightness 50
Saturation 0, ISO 0, Video Stabilisation No, Exposure compensation 0
Exposure Mode 'auto', AWB Mode 'auto', Image Effect 'none'
Metering Mode 'average', Colour Effect Enabled No with U = 128, V = 128
Rotation 0, hflip No, vflip No
ROI x 0.000000, y 0.000000, w 1.000000 h 1.000000
Camera component done
Encoder component done
Starting component connection stage
Connecting camera preview port to video render.
Connecting camera stills port to encoder input port
Opening output file x0000000.jpg
Enabling encoder output port
Starting capture 0
Finished capture 0
Closing down
Close down completed, all components disconnected, disabled and destroyed
```

คำสั่งควบคุมโหมดภาพนิ่ง (ต่อ)



--timelapse , -tl ให้กล้องถ่ายเป็นช่วง ๆ หน่วยเป็น millisec

เช่น raspistill -t 20000 -tl 2000 -o x%03d.jpg

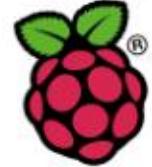
เปิดกล้องนาน 20 วินาที ถ่ายภาพนิ่งทุก ๆ 2 วินาที บันทึกภาพในชื่อ x001.jpg - x011.jpg

```
pi@raspberrypi ~/Mypicture $ raspistill -t 20000 -tl 2000 -o x%03d.jpg
pi@raspberrypi ~/Mypicture $ ls
x001.jpg  x003.jpg  x005.jpg  x007.jpg  x009.jpg  x011.jpg
x002.jpg  x004.jpg  x006.jpg  x008.jpg  x010.jpg
```

--encoding , -e บีบอัดไฟล์รูปแบบอื่น เช่น png ,bmp ,gif

เช่น raspistill -o x%03d.png -e png

คำสั่งควบคุมอิน 7



สำหรับต่อท้าย raspistill

- --preview , -p แสดงภาพออกจอด้วยขณะถ่ายรูป (รีโมตใช้ไม่ได้)

กำหนดพิกัด x,y หน้าจอ และขนาดภาพด้วยเช่น

`raspistill -p 100,150,1000,800`

- --fullscreen , -f แสดงรูปภาพเต็มจอ เช่น `raspistill -f`

- --nopreview , -n ปิดการแสดงออกหน้าจอ เช่น `raspistill -n`

- --opacity , -op กำหนดความเข้มของรูปภาพ ช่วง 0-255 255 ชัดสุด

คำสั่งควบคุมโหมดภาพเคลื่อนไหว



raspivid เปิดโมดูลกล้องเพื่อถ่ายภาพเคลื่อนไหว

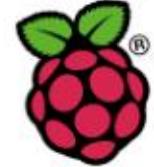
- --width , -w ปรับขนาดความกว้างอยู่ในช่วง 64 ถึง 1920p
- --height , -h ปรับขนาดความสูงอยู่ในช่วง 64 ถึง 1080p
- --output , -o บันทึกภาพเคลื่อนไหวในชื่อและนามสกุล h264

เช่น `raspivid -o movie1.h264`

- --timeout , -t กำหนดช่วงเวลาเปิดกล้อง ปกติ 5 วินาที
หรือกด `Ctrl+C` เพื่อหยุดการทำงานได้

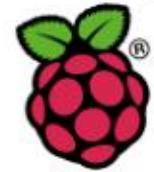
```
raspivid -t 60000 -o Clip1.h264
```

สั่งการกล้องด้วย Python

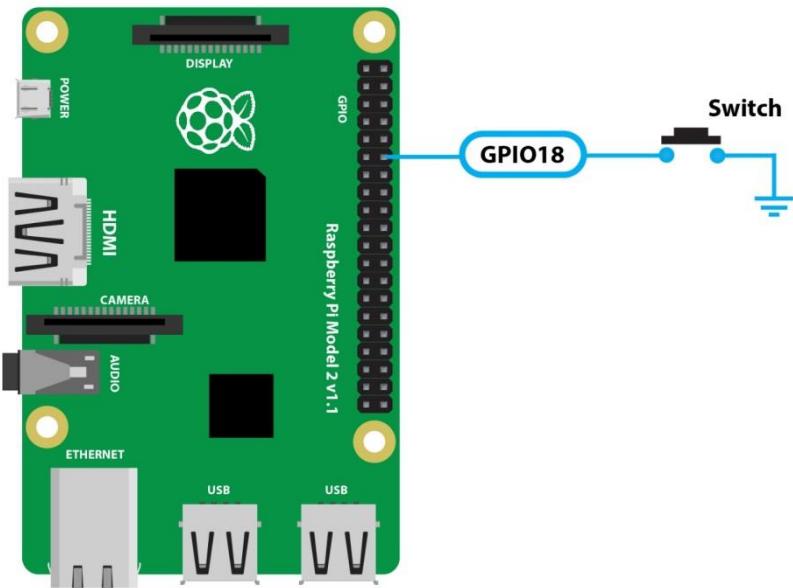


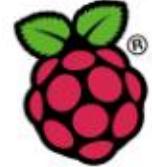
```
import time  
import picamera  
  
with picamera.PiCamera() as camera:  
    camera.resolution = (1024, 768)  
    camera.start_preview()  
    time.sleep(2)  
    camera.capture('Picam_P1.jpg')
```

กดปุ่ม ถ่ายรูป



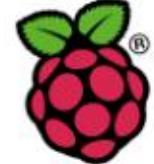
```
import picamera  
import time  
import RPi.GPIO as GPIO  
  
GPIO.setwarnings(0)  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(18,GPIO.IN)  
  
camera= picamera.PiCamera()  
camera.resolution =(1024,768)  
camera.start_preview()  
x=0  
while 1:  
    if(not GPIO.input(18)) :  
        camera.capture(str(x)+'.jpg')  
        time.sleep(0.5)  
        x=x+1
```





```
camera.sharpness = 0
camera.contrast = 0
camera.brightness = 50
camera.saturation = 0
camera.ISO = 0
camera.video_stabilization = False
camera.exposure_compensation = 0
camera.exposure_mode = 'auto'
camera.meter_mode = 'average'
camera.awb_mode = 'auto'
camera.image_effect = 'none'
camera.color_effects = None
camera.rotation = 0
camera.hflip = False
camera.vflip = False
camera.crop = (0.0, 0.0, 1.0, 1.0)
```

ถ่ายรูปป้องกันที่



```
import picamera
import time
import RPi.GPIO as GPIO
import datetime

GPIO.setwarnings(0)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.IN)
namex=str(datetime.datetime.now())+".jpg"
camera= picamera.PiCamera()
camera.resolution =(1024,768)
camera.start_preview()
camera.rotation =45
while 1:
    if(not GPIO.input(17)) :
        camera.capture(namex)
        time.sleep(0.5)
```

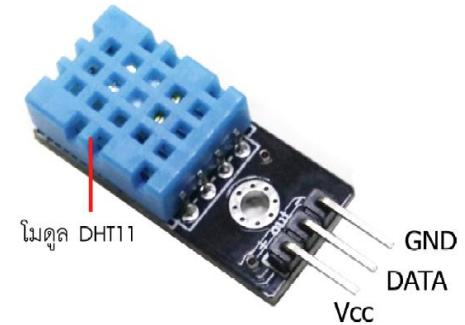


2015-11-14 11:38:27.989836.jpg



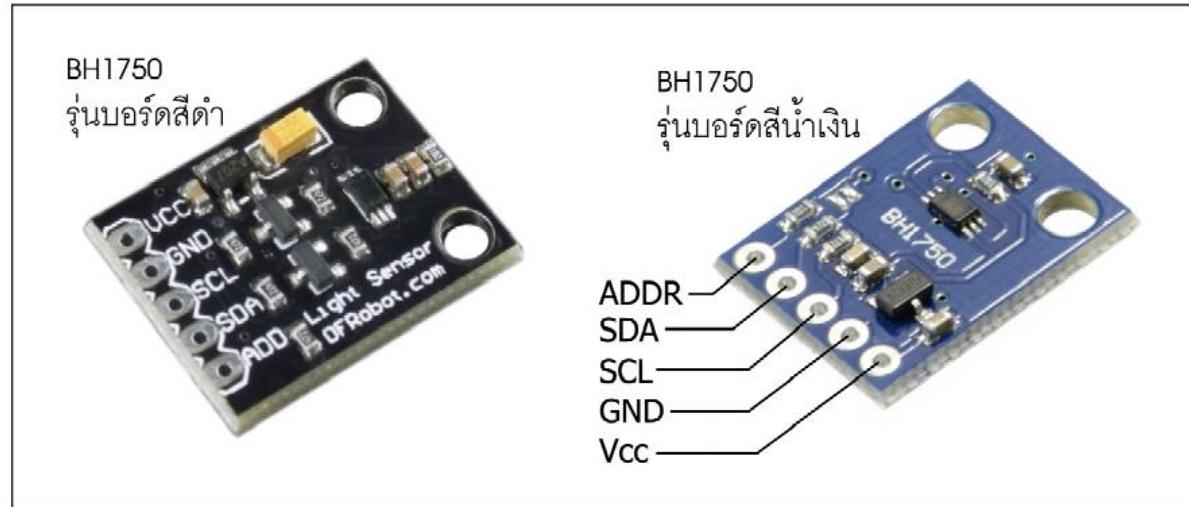
2015-11-14 11:40:28.680059.jpg

Sensor



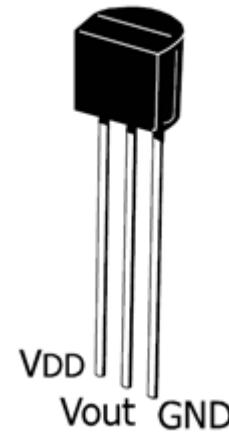
DHT11

reolu**Pi**R

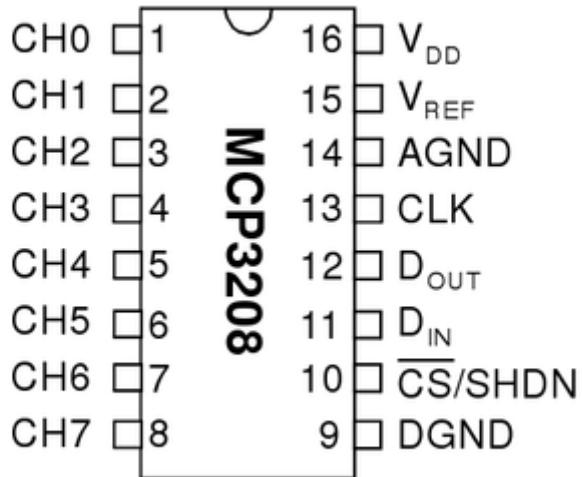
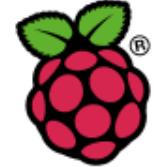


MCP9701

TO-92



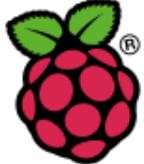
การอ่านค่า Analog



คุณสมบัติ

- แปลงสัญญาณสูงสุดถึง 12 บิต(0-4095)
- ติดต่อ แบบระบบบัส SPI (Serial Peripheral Interface)
- 8 ช่องสัญญาณ(CH0-CH7)

ตั้งค่าเปิด SPI



sudo raspi-config เปิดการตั้งค่า

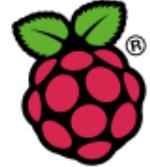
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem	Ensures that all of the SD card s
2 Change User Password	Change password for the default u
3 Enable Boot to Desktop/Scratch	Choose whether to boot into a des
4 Internationalisation Options	Set up language and regional sett
5 Enable Camera	Enable this Pi to work with the R
6 Add to Rastrack	Add this Pi to the online Raspber
7 Overclock	Configure overclocking for your P
8 Advanced Options	Configure advanced settings
9 About raspi-config	Information about this configurat

1

<Select> <Finish>

ตั้งค่าเปิด SPI



เลือก A6 SPI

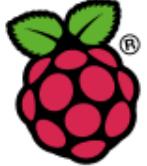
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan	You may need to configure oversca
A2 Hostname	Set the visible name for this Pi
A3 Memory Split	Change the amount of memory made
A4 SSH	Enable/Disable remote command lin
A5 Device Tree	Enable/Disable the use of Device
A6 SPI	Enable/Disable automatic loading
A7 I2C	Enable/Disable automatic loading
A8 Serial	Enable/Disable shell and kernel m
A9 Audio	Force audio out through HDMI or 3
A0 Update	Update this tool to the latest ve

<Select> <Back>

2

ตั้งค่าเปิด SPI



เลือก Yes

Would you like the SPI interface to be enabled? **3**

<Yes> <No>

ตั้งค่าเปิด SPI



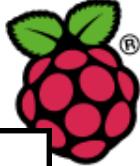
เลือก ok

The SPI interface will be enabled after a reboot

4

<0k>

ໂຄນດກາຣອ່າຄ່າ

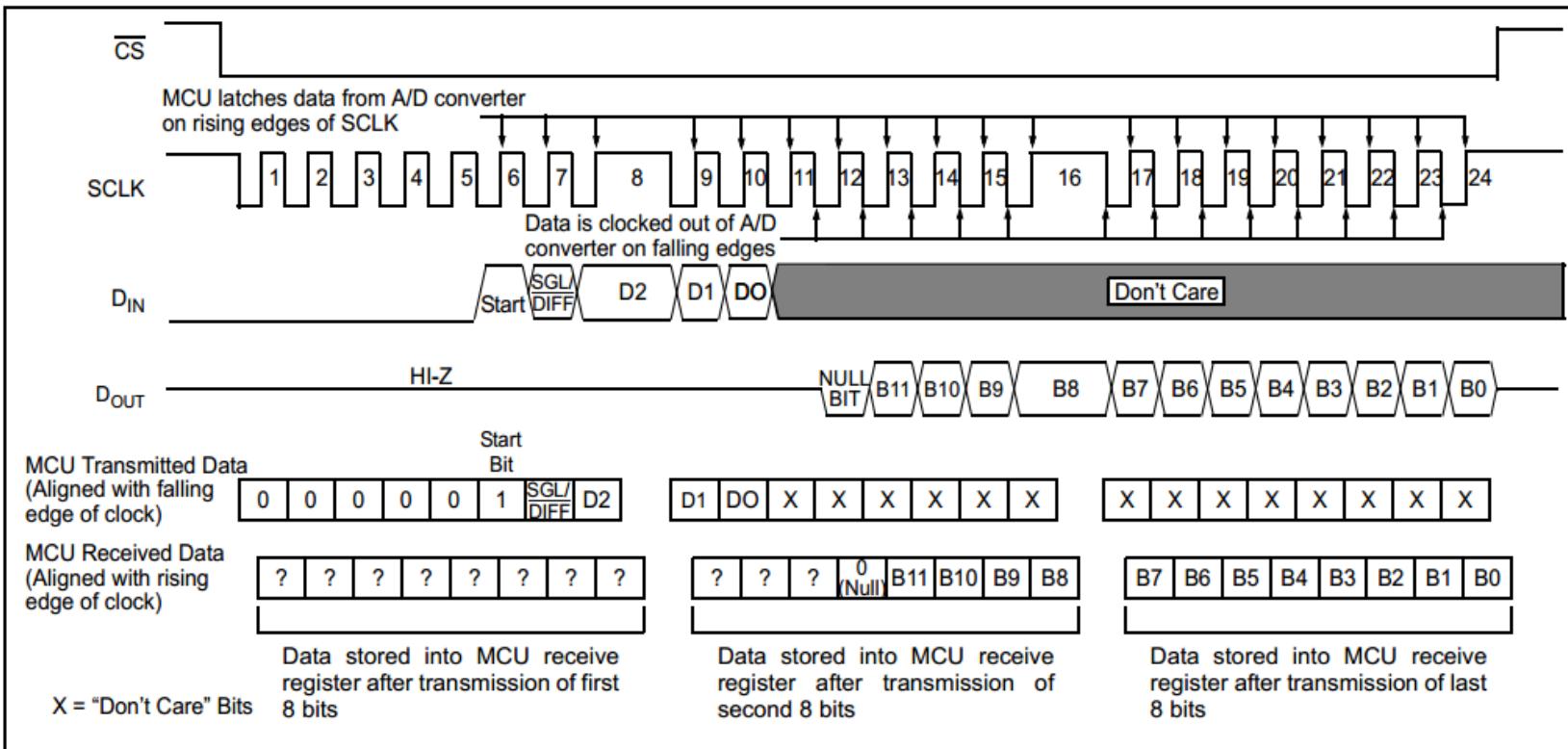


ຊີງເກີລເວັນດ

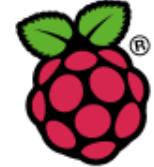
ດີຝເພື່ອເຮັນເຂີຍລ

ບົດຄວບຄຸມ				ໄທມດຂອງອິນພຸຕ	ໜ້ອງອິນພຸຕທີ່ເລືອກ	
D3 Single Diff	D2	D1	D0			
1	0	0	0	ຊີງເກີລເວັນດ	CH0	
1	0	0	1	ຊີງເກີລເວັນດ	CH1	
1	0	1	0	ຊີງເກີລເວັນດ	CH2	
1	0	1	1	ຊີງເກີລເວັນດ	CH3	
1	1	0	0	ຊີງເກີລເວັນດ	CH4	
1	1	0	1	ຊີງເກີລເວັນດ	CH5	
1	1	1	0	ຊີງເກີລເວັນດ	CH6	
1	1	1	1	ຊີງເກີລເວັນດ	CH7	
0	0	0	0	ດີຝເພື່ອເຮັນເຂີຍລ	CH0=IN+	CH1=IN-
0	0	0	1	ດີຝເພື່ອເຮັນເຂີຍລ	CH0=IN-	CH1=IN+
0	0	1	0	ດີຝເພື່ອເຮັນເຂີຍລ	CH2=IN+	CH3=IN-
0	0	1	1	ດີຝເພື່ອເຮັນເຂີຍລ	CH2=IN-	CH3=IN+
0	1	0	0	ດີຝເພື່ອເຮັນເຂີຍລ	CH4=IN+	CH5=IN-
0	1	0	1	ດີຝເພື່ອເຮັນເຂີຍລ	CH4=IN-	CH5=IN+
0	1	1	0	ດີຝເພື່ອເຮັນເຂີຍລ	CH6=IN+	CH7=IN-
0	1	1	1	ດີຝເພື່ອເຮັນເຂີຍລ	CH6=IN-	CH7=IN+

ส่งคำสั่ง-รับค่ากับMCP3208



ส่งค่าสั้ง-รับค่ากับ MCP3208



ส่งค่าสั้งอ่านค่าในหนึ่งครั้งจะใช้ทั้งหมด 3 ไบต์

↓ กำหนดช่อง ↓

ส่ง



ไบต์แรก

ไบต์ที่สอง

ไบต์ที่สุดท้าย

รับ



รับค่าที่อ่านได้ใช้ 2 ไบต์

↑ ← ค่าอนาล็อก → ↑

ส่งคำสั่งไปยัง MCP3208



ส่งคำสั่งอ่านค่าในหนึ่งครั้งจะใช้ทั้งหมด 3 ไบต์



ไบต์แรก

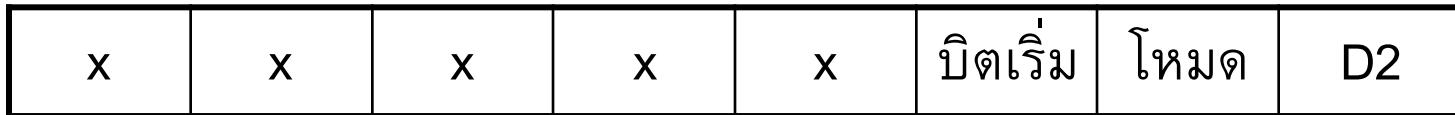


ไบต์ที่สอง



ไบต์ที่สุดท้าย

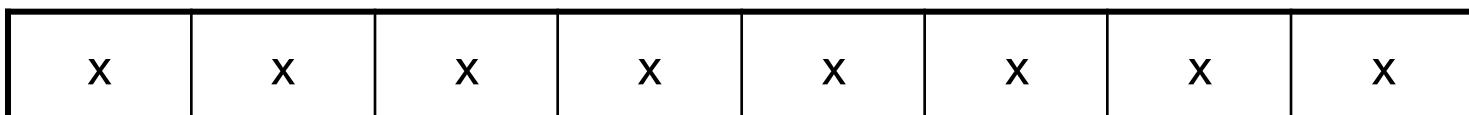
ไบต์แรก



ไบต์ที่สอง



ไบต์ที่สุดท้าย



ส่งคำสั่งไปยัง MCP3208



ไบต์แรก

X	X	X	X	X	บิตเริ่ม	โหมด	D2
---	---	---	---	---	----------	------	----

ไบต์ที่สอง

D1	D0	X	X	X	X	X	X
----	----	---	---	---	---	---	---

ไบต์ที่สุดท้าย

X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---

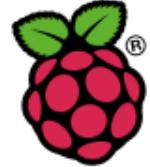
บิตเริ่ม มีค่าเท่ากับ 1 เสมอ

โหมด 0=ดิฟเฟอเรนเชียล, 1=ชิงเกิลเอ็นด์

D2 D1 D0 เลขฐานสองของช่องสัญญาณที่เลือก เช่น CH3=011

X = เป็นอะไรได้ แต่จะกำหนดเป็น 0 เสมอ

ส่งคำสั่งไปยัง MCP3208



ตัวอย่าง เลือกอ่านสัญญาณที่ช่อง CH3 ในโหมดซิงเกิลเอนเดอร์
บิตเริ่ม=1:โหมด=1:CH3=011 ดังนั้น D2=0,D1=1,D0=1:x=0

ไบต์แรก

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

ไบต์ที่สอง

1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ไบต์ที่สุดท้าย

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ผลลัพท์: 00000110 11000000 00000000 เลขฐานสอง

รับค่าจาก MCP3208



รับค่าอนาล็อกจะใช้ 2 ใบต์สุดท้าย

--	--	--	--	--	--	--	--

ใบต์แรก

			x	x	x	x
--	--	--	---	---	---	---

ใบต์ที่สอง

x	x	x	x	x	x	x	x
---	---	---	---	---	---	---	---

ใบต์ที่สุดท้าย

ใบต์แรก

?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---

ใบต์ที่สอง

?	?	?	?	B11	B10	B9	B8
---	---	---	---	-----	-----	----	----

ใบต์ที่สุดท้าย

B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----

รับค่าจากMCP3208



ตัวอย่างค่าที่อ่านได้

ใบต์แรก

?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---

ใบต์ที่สอง

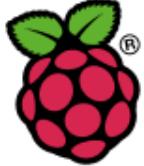
?	?	?	?	1	0	0	1
---	---	---	---	---	---	---	---

ใบต์ที่สุดท้าย

0	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

ใช้ 2 ใบต์สุดท้ายจะได้ **????1001 01110011**

รับค่าจากMCP3208



ใช้ 2 ไบต์สุดท้ายจะได้ **????1001 01110011**

ไบต์ที่ 2 = **????1001**

ไบต์ที่ 3= **01110011**

วิธีทำ

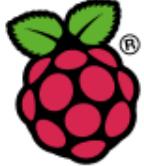
ไบต์ที่ 2 (**???? 1001 AND 0000 1111**) = **0000 1001**

(**0000 1001 << 8**)=**00001001 00000000**

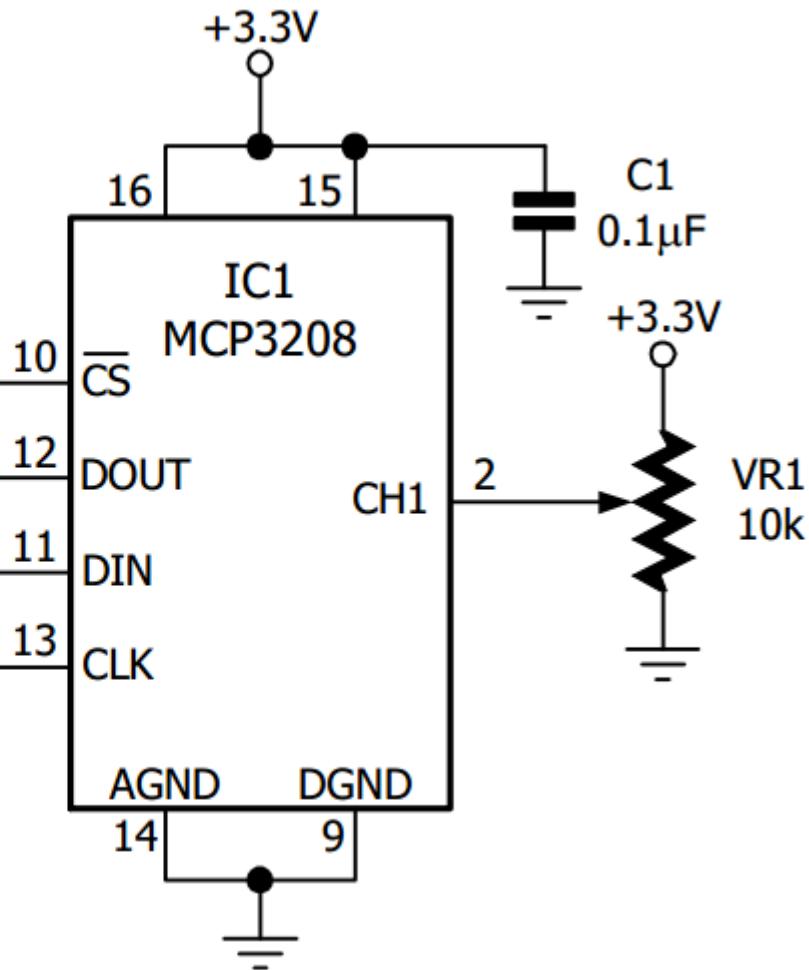
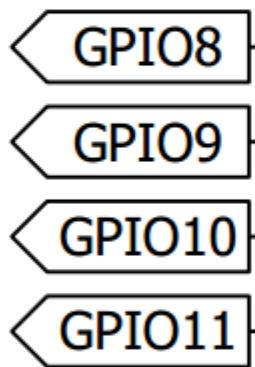
แล้ว OR กับไบต์ที่ 3(**00001001 00000000 OR 01110011**)

ผลลัพธ์ **00001001 01110011 (2419)**

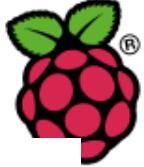
ต่อวงจร



Raspberry Pi 2 GPIO port



รับค่าจากMCP3208



```
import spidev  
  
import time analog_ch = 1  
  
spi = spidev.SpiDev()  
  
spi.open(0,0)  
  
def readADC(adcnum):  
    if adcnum > 7 or adcnum < 0:  
        return -1;  
  
    r = spi.xfer2([4 | 2 | (adcnum >> 2), (adcnum & 3) << 6, 0])  
  
    adcout = ((r[1] & 15) << 8) + r[2]  
  
    return adcout  
  
while True:  
    value = readADC(analog_ch)  
    print("analog_ch1=",value)  
    time.sleep(0.3)
```

รับค่าจาก MCP3208



จาก : `r = spi.xfer2([4 | 2 | (adcnum >> 2), (adcnum & 3) << 6, 0])`

ตัวอย่าง ต้องการอ่านช่อง CH3 โหมด ซิงเกิลเอน

`adcnum=3`

ใบต์ 1: $4|2|(3>>2)=000\ 0110$

ใบต์ 2: $(3&3)<<6=1100\ 0000$

ใบต์ 3: `0000 0000`

ผลลัพท์ `0000110 11000000 00000000`

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

ใบต์แรก

1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ใบต์ที่สอง

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ใบต์ที่สุดท้าย

รับค่าจากMCP3208



?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---

ใบต์แรก

?	?	?	?	1	0	0	1
---	---	---	---	---	---	---	---

ใบต์ที่สอง

0	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

ใบต์ที่สุดท้าย

จาก : $\text{adcout} = ((\text{r}[1] \& 15) \ll 8) + \text{r}[2]$

จากตัวแปร $\text{r}[2] = \text{ใบต์ที่ } 3 : \text{r}[1]=\text{ใบต์ที่ } 2 : \text{r}[0]\text{ใบต์ที่ } 1$

$\text{r}[1] = \text{???? } 1001$

$\text{r}[2]=01110011$

ใบต์ สูงสุด: $(\text{????1001}\&\text{00001111})\ll8=00001001\ 00000000$

ใบต์ ต่ำสุด: 01110011

ใบต์ สูงสุด+ ใบต์ ต่ำสุด= $00001001\ 01110011(2419)$



MCP9701 ไอซีวัดอุณหภูมิ

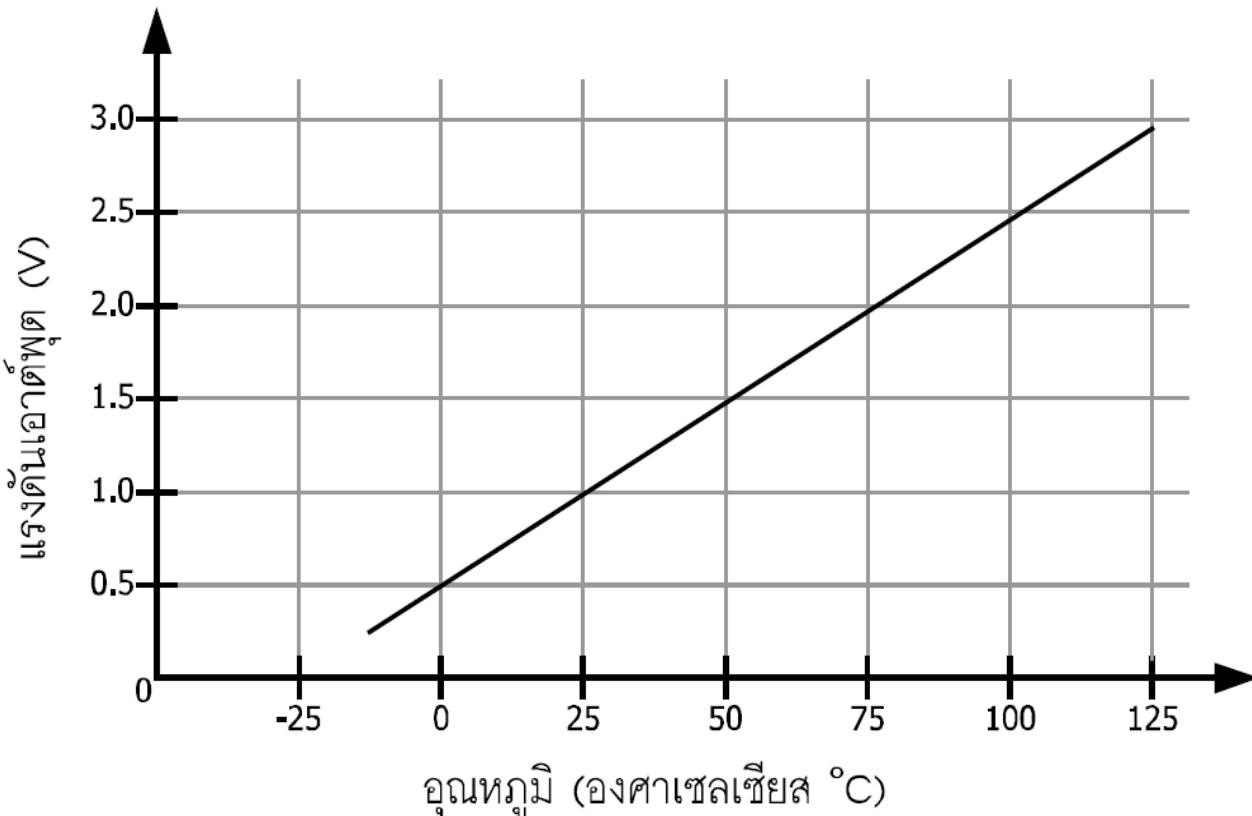
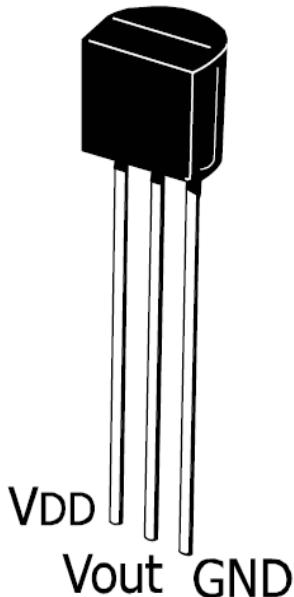
คุณสมบัติทางเทคนิคของ MCP9701 ที่ควรทราบ

- เป็น ไอซีวัดอุณหภูมิในกลุ่มเทอร์มิสเตอร์แบบแบนแอกตีฟที่ให้ผลการทำงานแบบเชิงเส้น
- ย่านวัด -40 ถึง +125 องศาเซลเซียส
- ผลการวัดอ้างอิงกับหน่วยขององศาเซลเซียสโดยตรง
- ความผิดพลาดเฉลี่ย ± 2 องศาเซลเซียส
- ย่านไฟเลี้ยง +3.1 ถึง +5.5V กินกระแสไฟฟ้าเพียง 6mA ใช้แบตเตอรี่เป็นแหล่งจ่ายไฟได้
- ค่าแรงดันเอาต์พุต 500mV (ที่ 0°C) ถึง 2.9375V (ที่ 125°)
- ค่าแรงดันเอาต์พุตต่อการเปลี่ยนแปลงอุณหภูมิ 19.5mV/°C ใช้งานกับวงจรแปลงสัญญาณ
อนalog เป็นดิจิตอลความละเอียดตั้งแต่ 8 บิตได้ โดยมีความคลาดเคลื่อนต่ำ

MCP9701 ໄອซីវេដអុណហក្សមិ



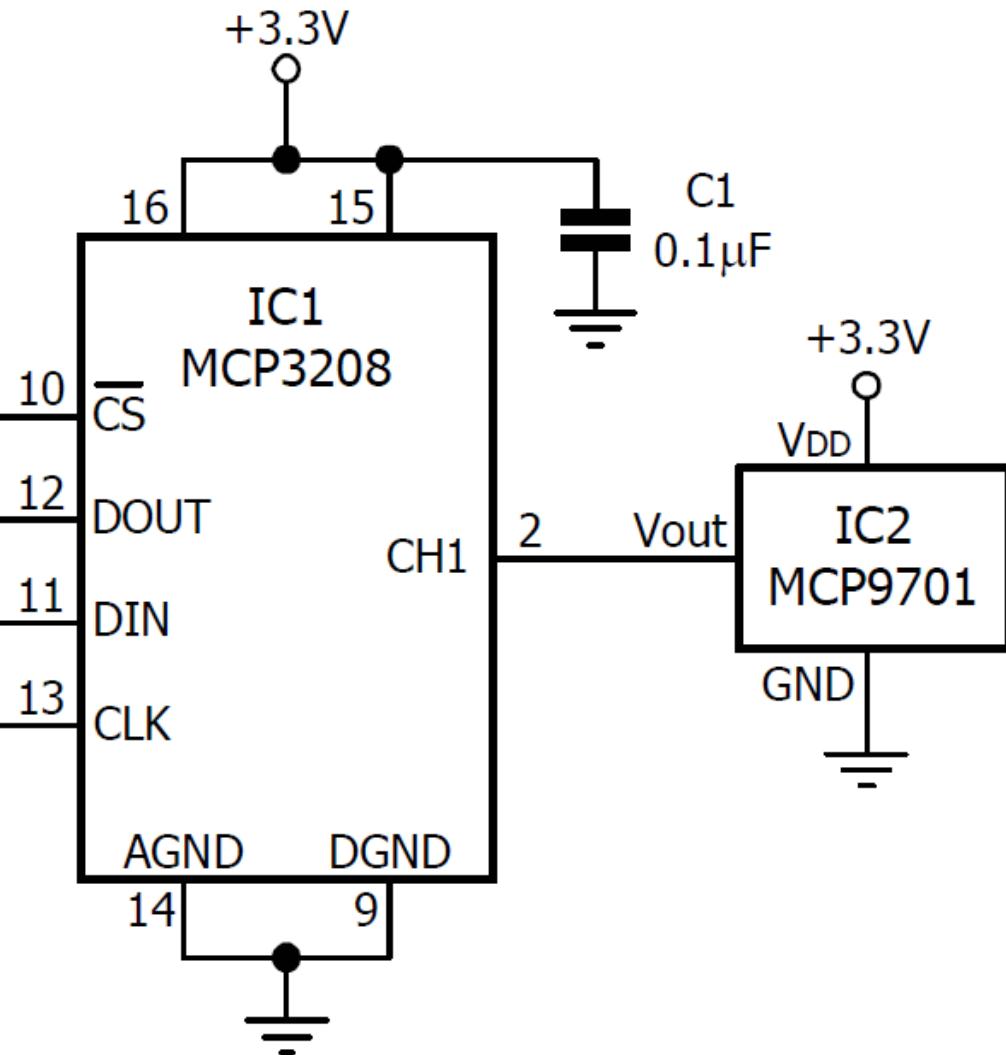
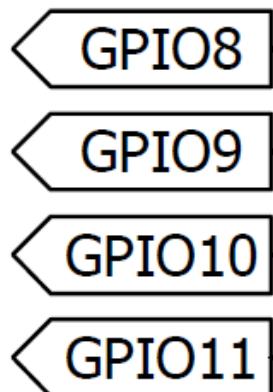
TO-92



ต่อวงจร



Raspberry Pi 2 GPIO port

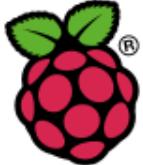


เขียนโค้ด



```
import spidev
import time
analog_ch = 1
spi = spidev.SpiDev()
spi.open(0,0)
def readADCadcnum):
    if adcnum > 7 or adcnum < 0:
        retrun -1;
    r = spi.xfer2([4|2|(adcnum >> 2), (adcnum & 3) << 6, 0])
    adcout = ((r[1] & 15) << 8) + r[2]
    return adcout
```

เขียนโค้ด(ต่อ)



while True:

 value = readADC(analog_ch)

 voltage = value***3.3/4096**

 temp = (voltage-**0.5**)/**0.0195**

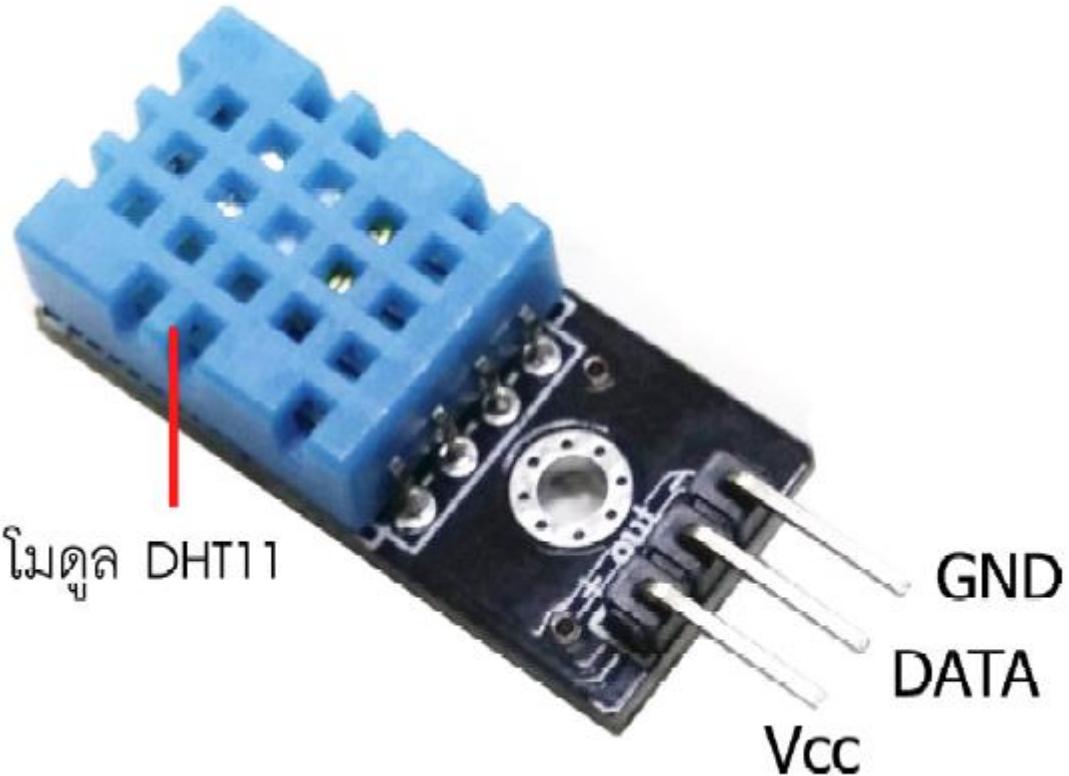
print("Temperature = %2.2f*C" % temp)

 time.sleep(**0.3**)

ZX-DHT11



บอร์ดวัดความชื้นสัมพัทธ์และอุณหภูมิ





รูปแบบข้อมูลของ DHT11

ใบต์ที่ 1 ข้อมูลจำนวนเต็มของความชื้นสัมพัทธ์ (%RH) 8 บิต

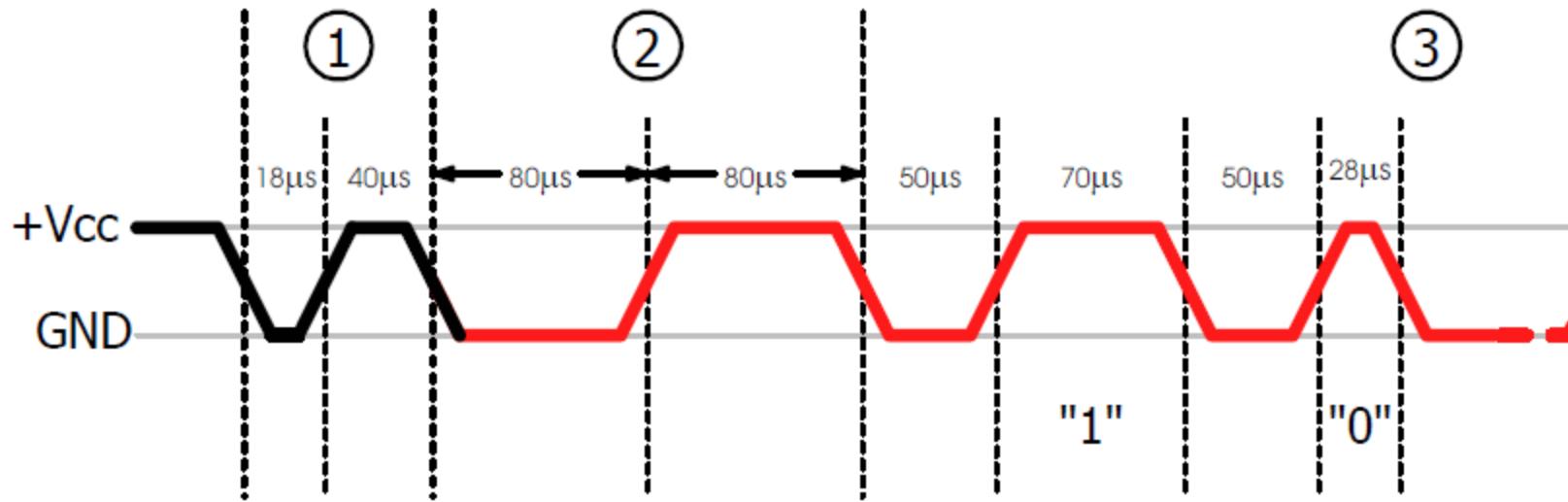
ใบต์ที่ 2 ข้อมูลทศนิยมของความชื้นสัมพัทธ์ (%RH) 8 บิต

ใบต์ที่ 3 ข้อมูลจำนวนเต็มของอุณหภูมิ (องศาเซลเซียส) 8 บิต

ใบต์ที่ 4 ข้อมูลทศนิยมของอุณหภูมิ (องศาเซลเซียส) 8 บิต

ใบต์ที่ 5 ข้อมูลตรวจสอบผลรวม (check sum)

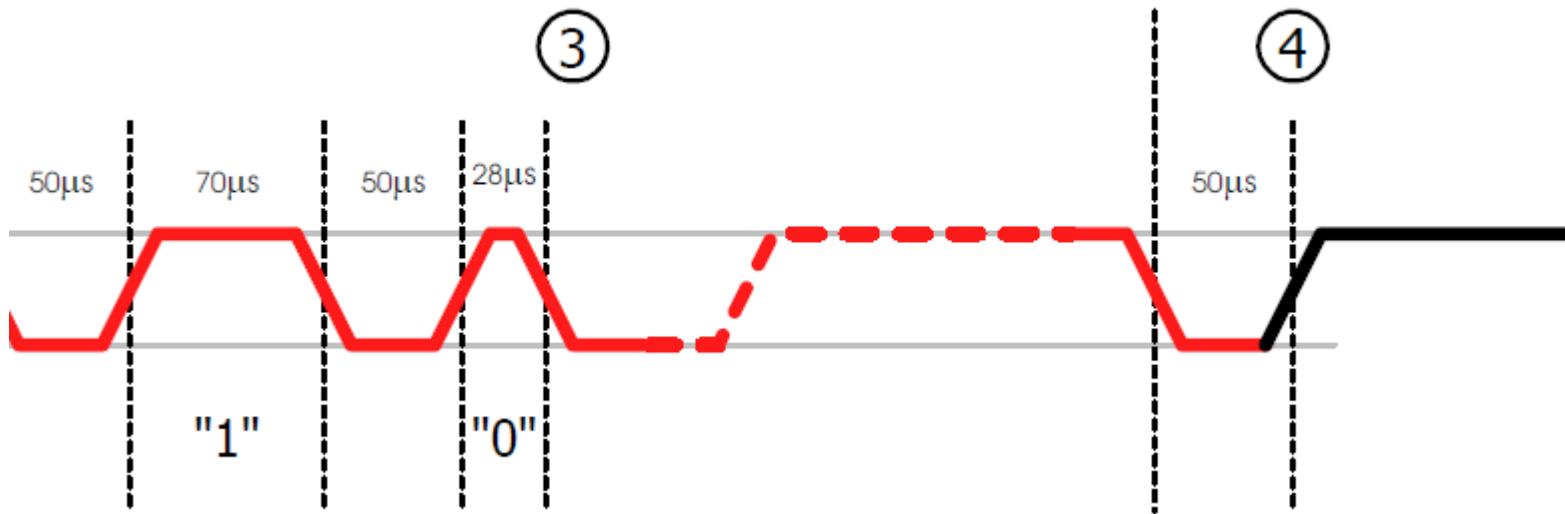
ZX-DHT11



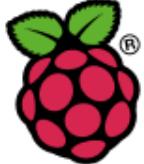
เส้นสีดำ: ช่วงที่ RaspberryPi กำหนดสถานะโลจิกของบัส

เส้นสีแดง: ช่วงที่ DHT11 กำหนดสถานะโลจิกของบัส

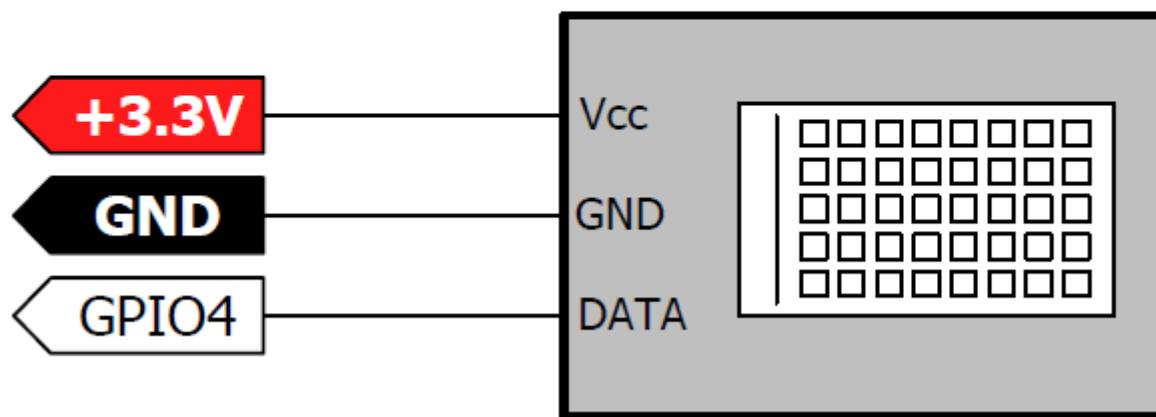
ZX-DHT11



ຕ່ອ DHT11



ZX-DHT11 Humidity/Temperature sensor module



ทดสอบ



```
sudo python3
```

```
import Adafruit_DHT
```

```
Adafruit_DHT.read_retry(11, 4)
```

ກົດສອບ



```
import time
import Adafruit_DHT
Sensor = Adafruit_DHT.DHT11
GPIO = 4
while True:
    humidity, temperature = Adafruit_DHT.read_retry(Sensor,GPIO)
    if humidity is not None and temperature is not None:
        print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature,humidity))
    else:
        print('Failed to get reading. Try again!')
```

วัดความเข้มแสง



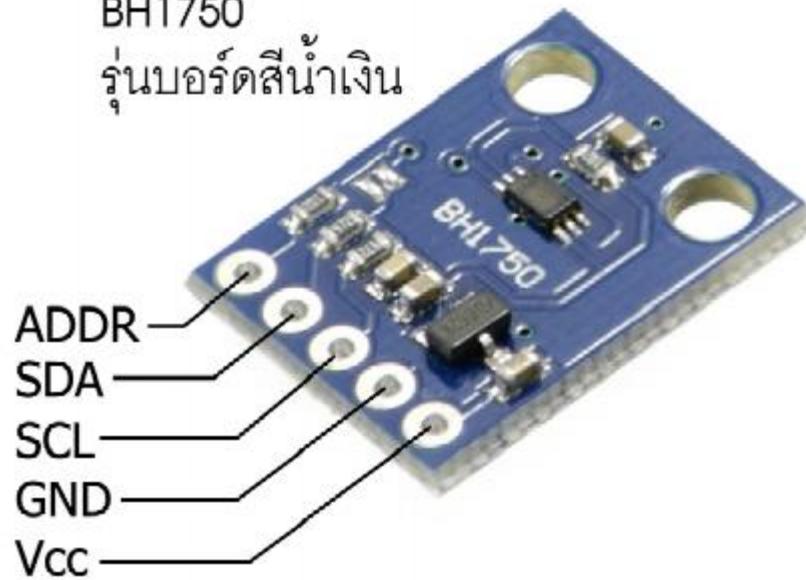
BH1750

รุ่นบอร์ดสีดำ



BH1750

รุ่นบอร์ดสีน้ำเงิน



คุณสมบัติ



ไฟเลี้ยง +3 ถึง +5 V

1-65535 (ลักษ์)

ทดสอบต่อการรับกวนจากแสงอินฟราเรด

ເປີດໃຊ້ i2c



sudo raspi-config

ຈະປຣາກຄູ່ນໍາຕ່າງ Raspberry Pi Software Configuration Tools ດັງລູບ

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem          Ensures that all of the SD card storage is available to the OS
2 Change User Password       Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
4 Internationalisation Options Set up language and regional settings to match your location
5 Enable Camera               Enable this Pi to work with the Raspberry Pi Camera
6 Add to Rastrack             Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock                  Configure overclocking for your Pi
8 Advanced Options            Configure advanced settings
9 About raspi-config          Information about this configuration tool

<Select>                      <Finish>
```

ເປີດໃຊ້ i2c



Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan	You may need to configure overscan if black bars are present on display
A2 Hostname	Set the visible name for this Pi on a network
A3 Memory Split	Change the amount of memory made available to the GPU
A4 SSH	Enable/Disable remote command line access to your Pi using SSH
A5 Device Tree	Enable/Disable the use of Device Tree
A6 SPI	Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)
A7 I2C	Enable/Disable automatic loading of I2C kernel module
A8 Serial	Enable/Disable shell and kernel messages on the serial connection
A9 Audio	Force audio out through HDMI or 3.5mm jack
A0 Update	Update this tool to the latest version

<Select> <Back>

1

ເປີດໃຊ້ i2c



Would you like the ARM I2C interface to be enabled?

2

<Yes>

<No>

The ARM I2C interface will be enabled after a reboot

3

<Ok>

ເປີດໃຊ້ i2c



Would you like the I2C kernel module to be loaded by default?

4

<Yes>

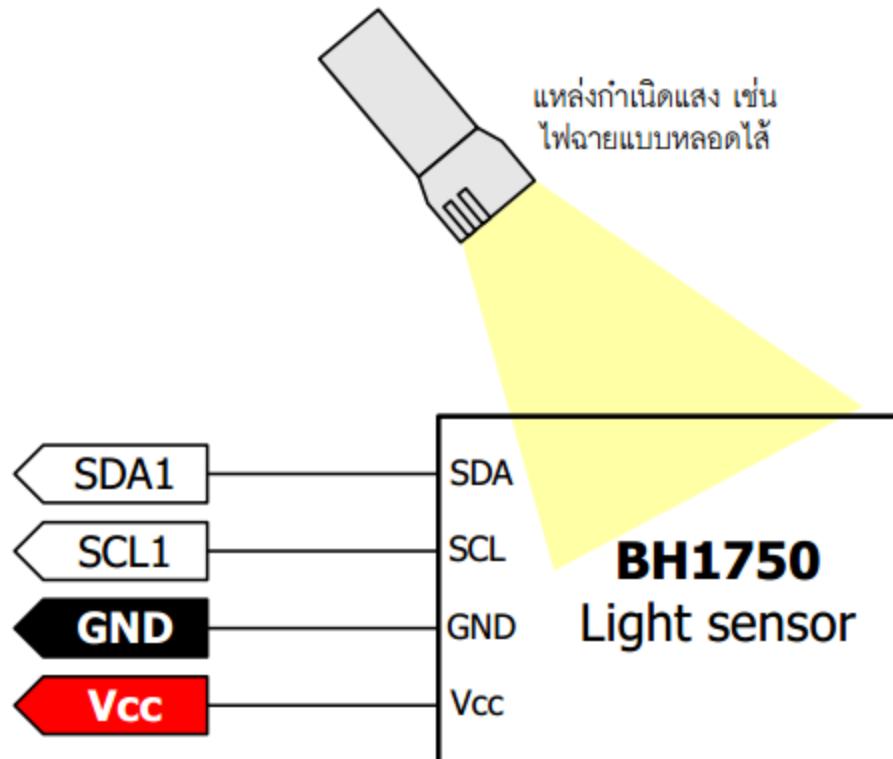
<No>

I2C kernel module will now be loaded by default

5

<Ok>

ต่อวงจร

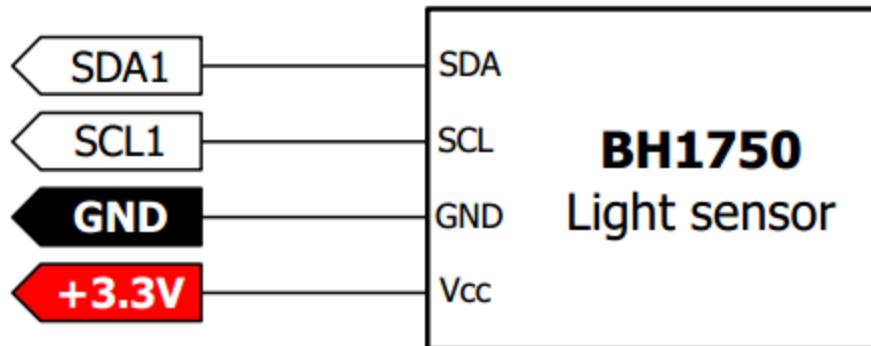
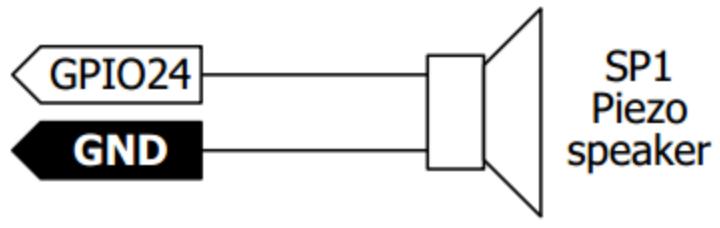


แสดงค่าความเข้มแสง

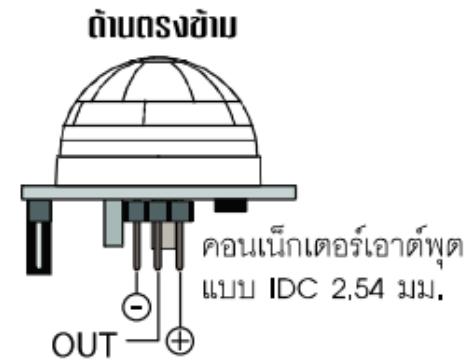
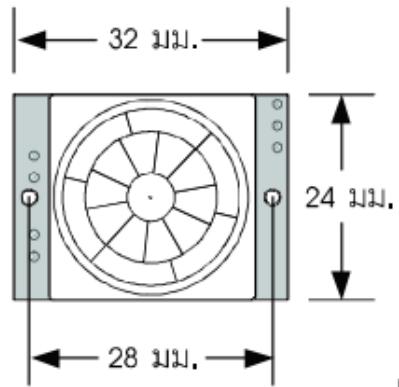


```
import time
import smbus
bus = smbus.SMBus(1)
addr = 0x23 # i2c adress
while True:
    data = bus.read_i2c_block_data(addr,0x11)
    lum=(data[1] + (data[0]<<8) / 1.2)
    print ("Luminosity " ,lum,"lx")
    time.sleep(0.5)
```

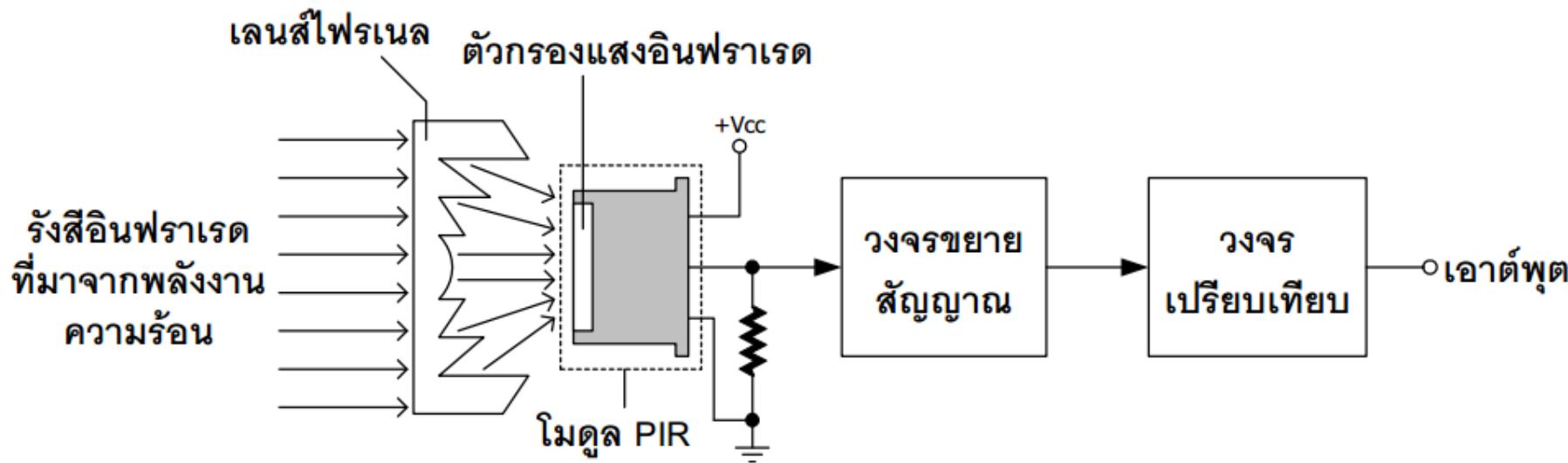
เตือน



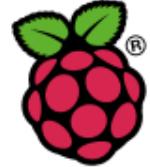
PIR



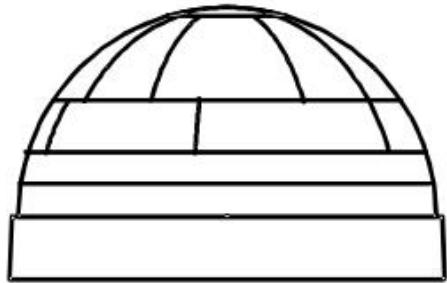
PIR



PIR



ภาพตัวขวางของเลนส์ไฟรีเฟล



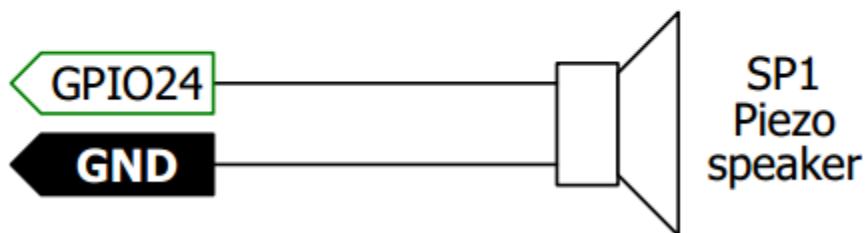
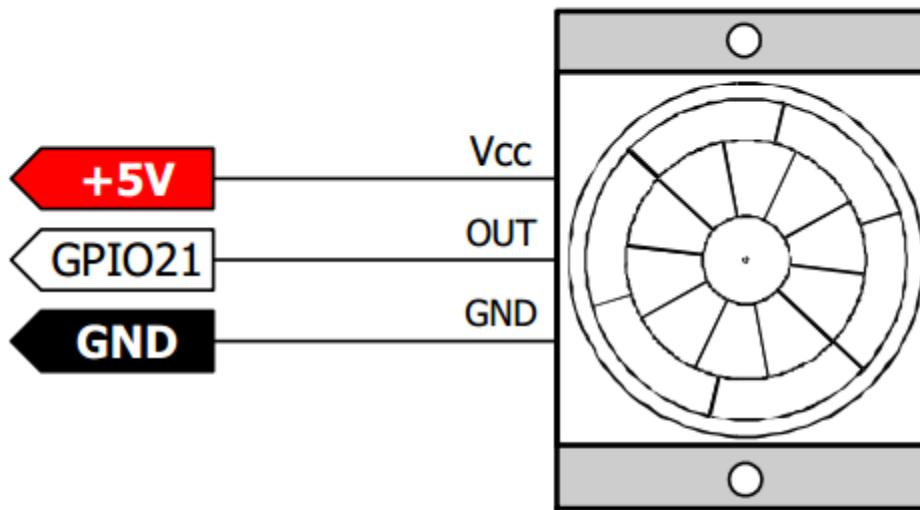
ลักษณะของเลนส์ไฟรีเฟลที่มีการสร้าง
เป็นโดมเพื่อครอบตัวตรวจจับ PIR
ทำให้สามารถรับแสงได้จากทุกทิศทาง
ทั้งด้านหน้าและด้านข้าง



ວັງຈາກ



ZX-PIR2.0 Motion sensor



PIR

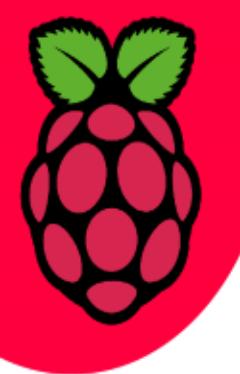


- ระยะการตรวจจับสูงสุด 20 ฟุต
- เมื่อตรวจพบความเคลื่อนไหวจะให้แรงดันเอาต์พุตที่สภาวะสูงที่ข้าเอาต์พุต
- ใช้เวลาในการปรับตัว 10 ถึง 60 วินาทีหลังจากได้รับไฟเลี้ยง
- ใช้ไฟเลี้ยงในย่าน +3.3 ถึง +5V กระแสไฟฟ้าน้อยกว่า 100 mA

เตือน



```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(24,GPIO.OUT)
blink = GPIO.PWM(18,500)
blink.start(0)
GPIO.setup(21,GPIO.IN)
st=0
while (1):
    da=datetime.now()
    microsec=da.microsecond
    if microsec > 700000:
        blink.ChangeDutyCycle(st)
    else:
        blink.ChangeDutyCycle(0)
    state=GPIO.input(21)
    if (state==1):
        st=50
    else:
        st=0
```



Raspberry Pi

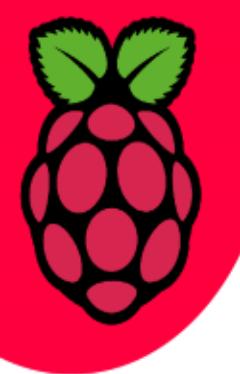


การตั้งค่าให้ชุดคำสั่งเริ่มทำงานอัตโนมัติ

1.สร้างชุดคำสั่งที่ต้องการจะรันแบบอัตโนมัติ โดยในที่นี้ใช้ชุดคำสั่ง LED.py และได้มีการเพิ่มการต่อสวิตซ์ไว้ที่ GPIO 22 (ต่อแบบ Active low) เมื่อมีการกดสวิตซ์โปรแกรมก็จะหยุดการทำงาน

```
import RPi.GPIO as  
GPIO  
  
import time  
GPIO.setmode(GPIO.BC  
M)  
ledPin=16  
swPin=22  
GPIO.setup(ledPin,GP  
IO.OUT)  
GPIO.setup(swPin,GPI
```

```
try:  
    while  
(GPIO.input(swPin)):  
        GPIO.output(ledPin,  
0)  
        time.sleep(0.5)  
    #print("Relay=OFF")
```



Raspberry Pi



2. บันทึกไฟล์ชื่อ **LED.py** และเลือเมนู **Build > Execute** บนโปรแกรม **Geany** หรือ เปิดเทอมิโนลแล้วสั่งรันด้วยคำสั่ง
sudo python3 LED.py
และให้จำตำแหน่งของไฟล์ rc.local ซึ่งในที่นือญู่ที่
/home/pi/Desktop/LED.py

3. เปิด Terminal และทำการแก้ไขไฟล์ **rc.local** โดยพิมพ์ชุดคำสั่ง
sudo nano /etc/rc.local

```
pi@raspberrypi ~ $ sudo nano /etc/rc.local
```



Raspberry Pi

4. จะพับชุดคำสั่งต่าง ๆ แล้วให้พิมพ์ชุดคำสั่ง

`sudo python3 /home/pi/Desktop/LED.py &`
ก่อนบรรทัด `exit0` ดังภาพ จากนั้นบันทึกการแก้ไขไฟล์โดยกด `Ctrl+X`
และ Y แล้วตามด้วย `Enter`

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6          File: /etc/rc.local      Modified
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
sudo python3 /home/pi/Desktop/LED.py &
exit 0
|
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

หมายเหตุ เครื่องหมาย & ต่อท้ายคำสั่งเป็นการบอกว่าคำสั่งนี้ให้ทำงานอยู่เบื้องหลัง
และถ้าหากไม่ใช้งานการทำงานชุดคำสั่งแบบอัตโนมัติให้เข้ามาเพิ่ม # ที่หน้าชุดคำสั่งที่เพิ่มขึ้นนี้



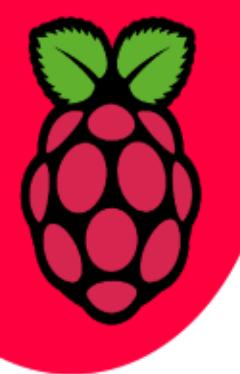
Raspberry Pi

5. เมื่อแก้ไขไฟล์เสร็จเรียบร้อยแล้วให้ลองรีบูตใหม่ด้วยคำสั่ง **sudo reboot**

```
pi@raspberrypi ~ $ sudo reboot
```

6. เมื่อบอร์ดเริ่มกลับมาทำงานใหม่ก็จะเห็นว่าหลอด LED จะกระพริบตลอดเวลาโดยที่ไม่ต้องพิมพ์คำสั่งให้ทำงานเลยหลังจากตัวบอร์ดเริ่มทำงาน

หมายเหตุ ระบบต้องตั้งค่าแบบ **Console Text console, requiring login (default)** ซึ่งยังไม่เข้าสู่ **startx**

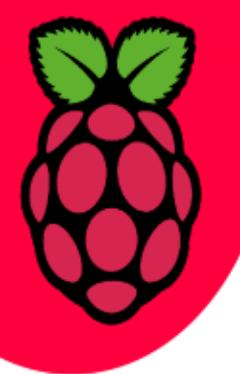


Raspberry Pi



ทดสอบการทำงานชุดคำสั่งเริ่มทำงานอัตโนมัติ

แต่ถ้าหากทดลองกดสวิตช์ก็จะพบว่าโปรแกรมหลักจาก loop while และหยุดทำงาน (เป็นผลมาจากการตั้งค่าเงื่อนไขอ่านค่าการกดสวิตช์)



Raspberry Pi

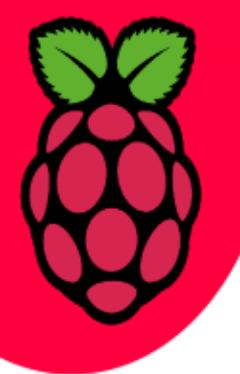


วิธีการ Kill process ที่ทำงานอยู่

1. Remote เข้าไปยังบอร์ดที่ทำงานอยู่ เปิด Terminal ทำการค้นหา Process ID ของโปรแกรมที่ทำงานอยู่ โดยใช้คำสั่ง

```
ps ax | grep {ชื่อไฟล์}
```

```
pi@raspberrypi ~ $ ps ax | grep LED.py
 2207 ?      S      0:00 sudo python3 /home/pi/Desktop/LED.py
 2216 ?      S      0:00 python3 /home/pi/Desktop/LED.py
 2255 pts/0  S+    0:00 grep --color=auto LED.py
pi@raspberrypi ~ $
```



Raspberry Pi



2. จะเห็นได้ว่ามี Process ของ LED.py อยู่ 3 ตัวที่ทำงาน ซึ่งจะเห็นเลข Process ID อยู่ด้วย

```
pi@raspberrypi ~ $ ps ax | grep LED.py
 2207 ?          S      0:00 sudo python3 /home/pi/Desktop/LED.py
 2216 ?          S      0:00 python3 /home/pi/Desktop/LED.py
 2255 pts/0      S+     0:00 grep --color=auto LED.py
pi@raspberrypi ~ $
```

3. ใช้คำสั่ง Kill Process และใช้การระบุ Process ID ที่ต้องการจะปิด
ให้ใช้ Process ID ของตัวไหนก็ได้ที่มาจากการคำสั่ง python3
(จะเป็นตัวที่มีคำสั่ง sudo นำหน้าหรือไม่ก็ได้)

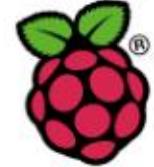
sudo kill 2207



4. หลังจากชุดคำสั่งหยุดทำงานแล้ว ตรวจสอบ Process ที่ทำงานอีกครั้งจะพบว่าไม่มี Process ของชุดคำสั่ง LED.py ลงเหลืออยู่แล้ว

```
pi@raspberrypi ~ $ ps ax | grep LED.py
2260 pts/0    S+    0:00 grep --color=auto LED.py
```

WiringPi เขียนภาษา C บน RPi



ติดตั้ง wiringPi

```
git clone git://git.drogon.net/wiringPi  
cd wiringPi  
git pull origin  
.build
```

ถ้าไม่ได้ต้องลง Make ก่อน
sudo apt-get install Make

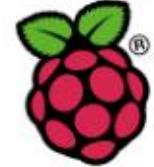
ทดสอบผ่าน Terminal (สั่ง LED ติดและดับที่ขา 18)

```
gpio -g mode 18 output  
gpio -g write 18 1  
gpio -g write 18 0
```

ทดสอบผ่าน Terminal (ให้ขา 22 เป็นอินพุตพูลอัพ และอ่านค่า)

```
gpio -g mode 22 up  
gpio -g read 22
```

WiringPi เขียนภาษา C บน RPi



กำหนดค่าเริ่มต้น

```
wiringPiSetupGpio() ; // กำหนดค่าเริ่มต้นให้ RPi แบบบานา BCM
```

กำหนดค่าเริ่มต้น

```
pinMode(22, INPUT)  
pinMode(23, OUTPUT)  
pinMode(18, PWM_OUTPUT)
```

สั่งงานเอาต์พุต

```
digitalWrite(23, HIGH) ;  
pwmWrite(18, 723) ;
```

อ่านค่าอินพุต

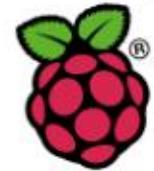
```
if (digitalRead(22))  
    printf("Pin 22 is HIGH\n") ;  
else  
    printf("Pin 22 is LOW\n") ;
```

กำหนดขาพูลอัพ

```
pullUpDnControl(22, PUD_UP) ;
```

หน่วงเวลา

```
delay(2000) ;
```



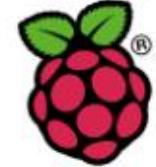
ปรับค่าการคอมไพล์สำหรับ WiringPi



Set Build Commands

#	Label	Command	Working directory	Reset
C commands				
1.	<u>Compile</u>	gcc -Wall -c "%f" -I wiringPi		
2.	<u>Build</u>	gcc -Wall -o "%e" "%f" -I wiringPi		
3.				
Error regular expression:				
Independent commands				
1.	<u>Make</u>	make		
2.	Make Custom Target	make		
3.	<u>Object</u>	make %e.o		
4.				
Error regular expression:				
<i>Note: Item 2 opens a dialog and appends the response to the command.</i>				
Execute commands				
1.	<u>Execute</u>	sudo "./%e"		
2.				
<i>%d, %e, %f, %p are substituted in command and directory fields, see manual for details.</i>				
<input type="button" value="Cancel"/>		<input type="button" value="OK"/>		

การใช้ wiringpi2 กับ Python และ Rpi



by [Gordon‘Dragon’ Henderson](#)

wiringPi พัฒนา Rpi ให้ง่ายเหมือนเขียน Arduino

ติดตั้งผ่าน github

```
git clone https://github.com/Gadgetoid/WiringPi2-Python.git
cd WiringPi2-Python
sudo python3 setup.py
```

ไลบรารี wiringPi2



เพิ่มไลบรารี

```
import wiringpi2 as gpio
```

setmode กำหนดรูปแบบขา

```
gpio.wiringPiSetupGpio()
```

```
gpio.wiringPiSetup()
```

กำหนดขาเป็น output

```
gpio.pinMode(pin,1)
```

กำหนดขาเป็น input

```
gpio.pinMode(pin,0)
```

output ส่งค่าออกไปที่ขา

```
gpio.digitalWrite(pin,1)
```

```
gpio.digitalWrite(pin,0)
```

input รับค่าลอจิกจากขา

```
gpio.digitalRead(pin)
```

pull up

```
gpio.pullUpDnControl(pin,2)
```

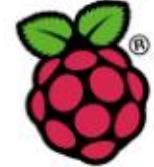
pull down

```
gpio.pullUpDnControl(pin,1)
```

ไม่ pull up/down

```
gpio.pullUpDnControl(pin,0)
```

ใช้คำสั่ง pwm ด้วย wiringPi2



กำหนดขา 18 เป็น PWM

```
wiringpi.pinMode(18,2)
```

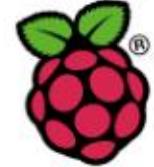
กำหนดค่า dutycycle

```
wiringpi.pwmWrite(18,duty)
```

```
import wiringpi2 as pi
import time

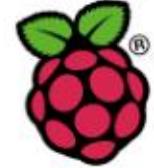
leds = [18,23,24,25]
sw = [16,20,21]
pi.wiringPiSetupGpio()
pi.pinMode(16,0)
pi.pinMode(18,2)

while True:
    for i in range(1025):
        pi.pwmWrite(18,i)
        time.sleep(0.001)
    for i in range(1025,-1,-1):
        pi.pwmWrite(18,i)
        time.sleep(0.001)
```



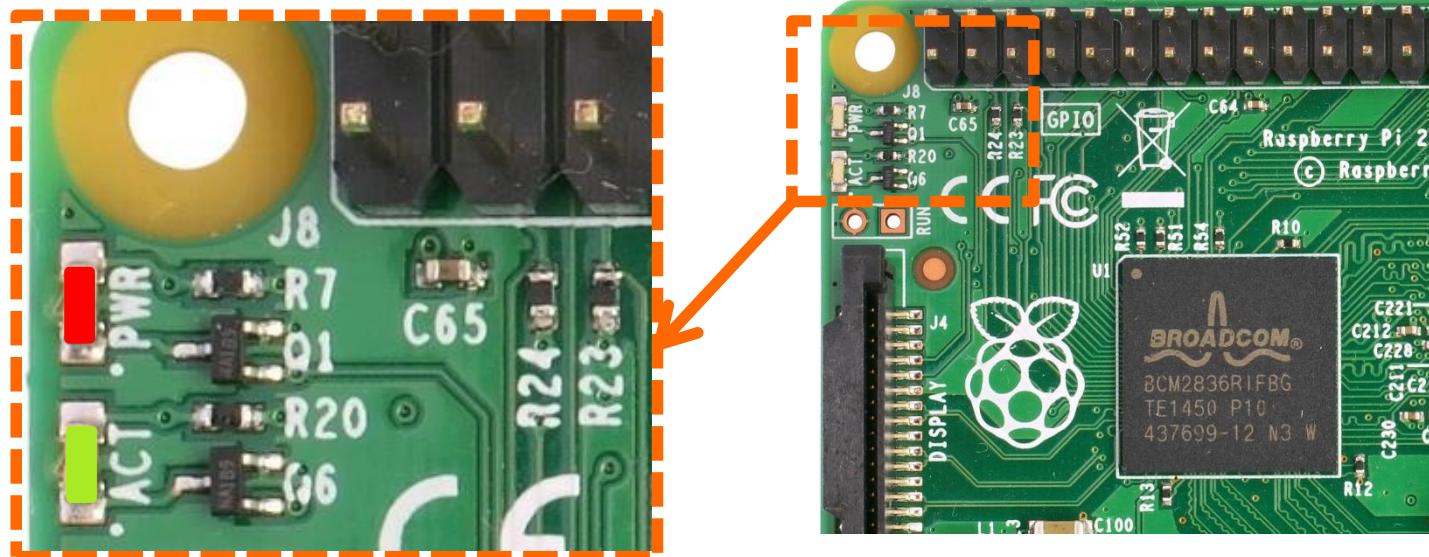
ภาคผนวก

รูปแบบการติดต่อ LED ACT

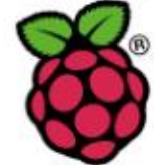


Linux ใช้การตั้งค่า Config ต่าง ๆ ผ่าน Text ไฟล์ใน `/etc/`

การเปลี่ยนสถานะของอุปกรณ์ระบบ ทำกับไฟล์สมุด
จะถูกเก็บใน `/sys/` และ `/proc/`



LED ACT บน R-Pi



`cat /sys/class/leds/led0/trigger`

```
pi@raspberrypi ~/test $ cat /sys/class/leds/led0/trigger  
none mmc0 timer oneshot heartbeat backlight gpio cpu0 cpu1 cpu2 cpu3 default-on input
```

↑
ปกติใช้ว่าค่าการอ่านเขียน SD-CARD

`echo "timer" | sudo tee /sys/class/leds/led0/trigger`

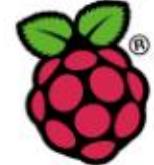
```
pi@raspberrypi ~/test $ echo "timer" | sudo tee /sys/class/leds/led0/trigger  
timer  
pi@raspberrypi ~/test $ cat /sys/class/leds/led0/trigger  
none mmc0 [timer] oneshot heartbeat backlight gpio cpu0 cpu1 cpu2 cpu3 default-on input
```

↑
ย้ายมาใช้ว่าค่า Timer แทน

LED ACT กำพร้า



LED ACT บน R-Pi



```
echo "none" | sudo tee /sys/class/leds/led0/trigger
```

↑
สั่ง LED ไม่ให้ถูกควบคุมจากครอ

```
echo "1" | sudo tee /sys/class/leds/led0/brightness
```

↑
สั่ง LED ติด

```
echo "0" | sudo tee /sys/class/leds/led0/brightness
```

↑
สั่ง LED ดับ

```
echo "mmc0" | sudo tee /sys/class/leds/led0/trigger
```

↑
คืนค่าเดิม (แสดงสถานะ อ่านเขียน SDCARD)



กฤษดา ใจเย็น KRITSADA JAIYEN

บริษัท อินโนเวติฟ เอ็กเพอริเมนต์ จำกัด

108 ซอยสุขุมวิท 101/2 ถ.สุขุมวิท

แขวงบางนา เขตบางนา กรุงเทพฯ 10260

โทรศัพท์ 02-7477001-4 โทรสาร 02-7477005

www.inex.co.th



Email : kritsada@inex.co.th