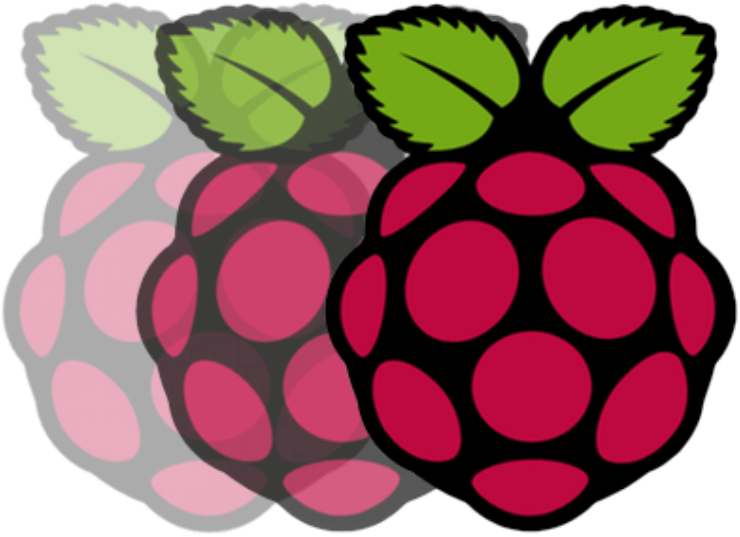


# BASIC RASPBERRY PI

*For Internet of Things*

Max Innovation Technology Co., Ltd.
Worachet@MaxInnoTech.com

# Contents

ขั้นตอนการเตรียม....3
การ Format SD Card....4
การ Burn Image File....5
การกำหนดค่า IP แบบ Direct Config เพื่อใช้ในการกำหนดค่าครั้งแรก....8
การกำหนดค่า IP แบบ Static....11
การตั้งค่า IP Address สำหรับ Wi-Fi ด้วย Command Line....15
การตั้งค่าทั่วไปของ Raspberry Pi....19
การขยายขนาด Space ใน SD Card ให้เต็มพื้นที่....19
การกำหนดการ Boot เข้าหน้าจอของ Raspberry Pi....20
การกำหนด Raspberry Pi Name....21
การกำหนด SSH....22
การตั้งเวลา Raspberry Pi....24
การอัพเดท Raspberry Pi....28
การติดตั้ง Remote Desktop....29
การติดตั้ง XRDP....29
การติดตั้ง VNC....31
การใช้งาน GPIO....33
การสั่งงานผ่าน GPIO ด้วย Command line....34
การเขียน Program Python พื้นฐาน....35
ตัวแปรในภาษา python....35
Syntax ในภาษา python....35

DO NOT COPY

การกำหนดตัวแปร ..... 36
ชนิดข้อมูล ..... 37
Operator ..... 40
Condition ..... 42
Loop ..... 42
Function ..... 43
การใช้ Python เพื่อควบคุม GPIO ..... 44
GPIO Output ..... 44
GPIO Input ..... 45
การใช้งานกล้อง USB Camera ..... 46
การหากล้อง USB ..... 46
ส่งภาพจาก WebCam ไปแสดงบน Web Browser ..... 46
บันทึกรูปจากกล้อง USB Camera ..... 47
ทำ Raspberry Pi เป็น MQTT Broker ..... 51
Socket Programming ..... 53
ทดลองทำ Raspberry Pi เป็น Socket Server ..... 55
ทดลองทำ Raspberry Pi เป็น Socket Server รับค่า JSON ..... 58
ทดลองส่งข่อมูลผ่าน JSON และบันทึกลง Database SQLite ..... 60
การใช้ MQTTSpy ..... 66
ทดลองใช้งาน Blynk ..... 67
QT programming ..... 68
Reference ..... 69

DO NOT COPY

# Basic Raspberry Pi

## ขั้นตอนการเตรียม

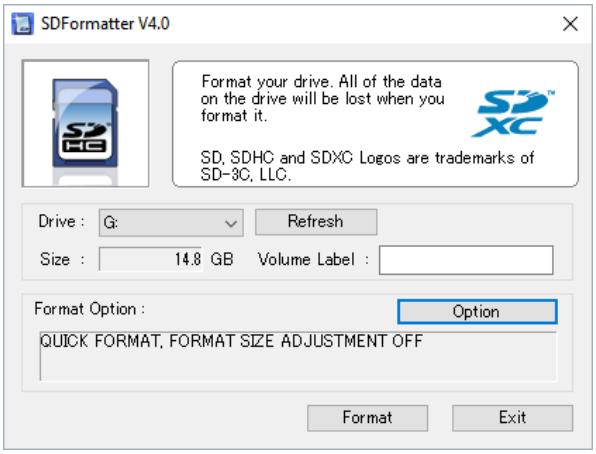
Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กมีขนาดใหญ่กว่าบัตรเครดิตเล็กน้อยซึ่งประสิทธิภาพของมันนั้นมีมากเกินกว่าราคาและนับว่าเป็นกระแสที่เปลี่ยนแปลงวงการ Embedded System นอกจากในตัว Raspberry Pi ที่สามารถติดตั้ง OS ลงไปได้แล้วความสามารถของมันในการเชื่อมต่อกับอุปกรณ์ Sensor ภายนอก อีกทั้งยังทำให้การควบคุมอุปกรณ์ต่าง ๆ ง่ายขึ้น เกิดความนิยมในการนำไปใช้ในการเรียนการสอนการและทดลองต่าง ๆ

Raspberry Pi นับว่าเป็น Embedded System อีกตัวหนึ่งที่ง่ายต่อการพัฒนา และปัจจุบัน Raspberry Pi ได้ออกผลิตภัณฑ์มาหลายรุ่นตั้งแต่ Raspberry Pi Model A, Model B, Model B+, Raspberry Pi 2 และ Raspberry Pi 3

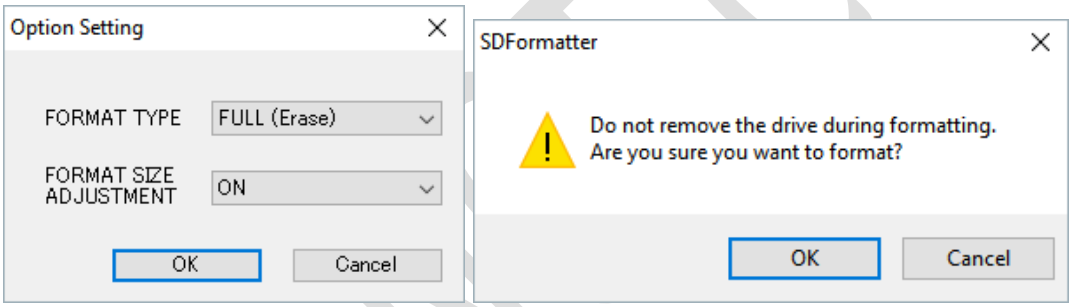
COPY

## การ Format SD Card

ใช้โปรแกรม SD Formatter โดยเลือก Drive ที่เป็น SD Card ของเรา และกำหนด Option ในการ Format



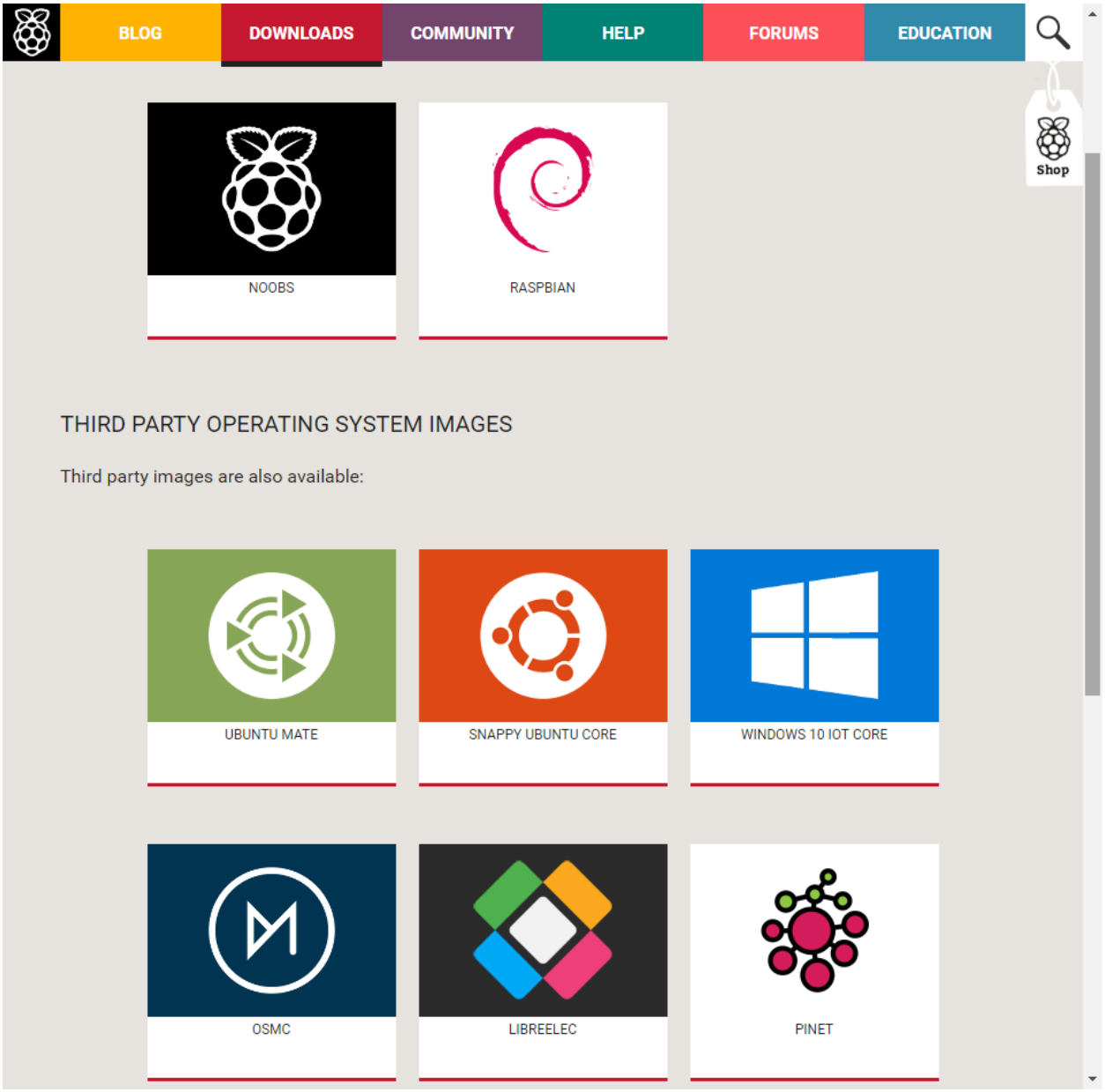

โดยเลือก Option Format Type เป็น Full (Erase) และ Format Size Adjustment เป็น ON




DO NOT COPY

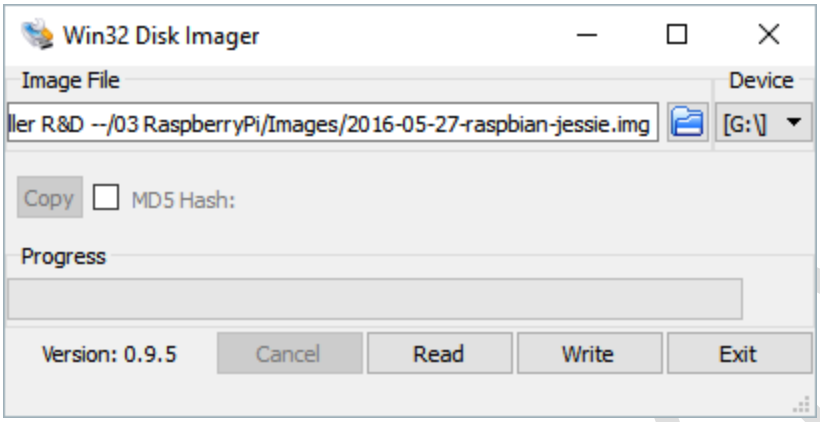
## การ Burn Image File

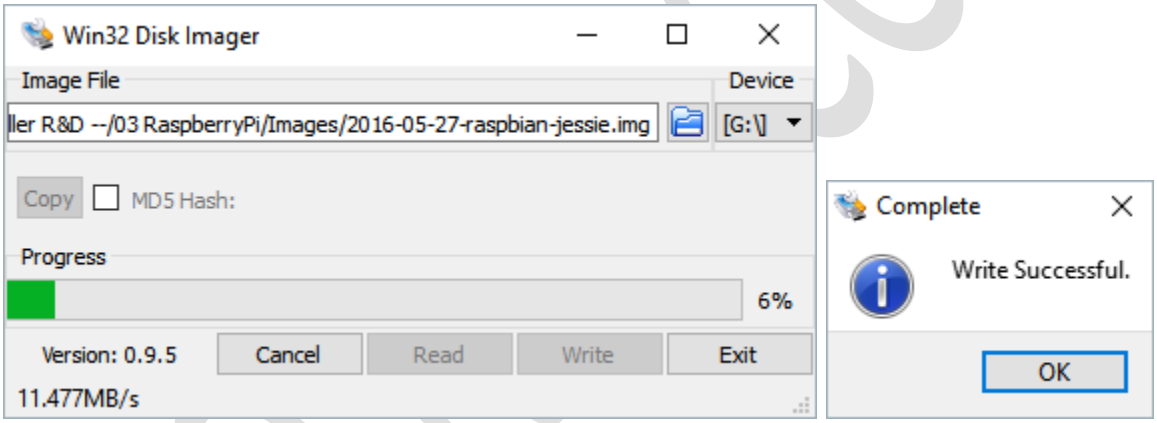
สำหรับ Raspberry Pi นั้นสามารถเลือกใช้ OS ในการใช้งานได้มากมาย โดยเราสามารถ Download OS ได้จาก https://www.raspberrypi.org/downloads/




ซึ่ง OS ที่นิยมใช้ บน Raspberry Pi นั้นนิยมใช้ RASPBIAN ซึ่งเป็น Official ของ Raspberry Pi โดย Version ที่จะใช้ในหนังสือเล่มนี้จะเป็น RASPBIAN Jessie

หลังจากที่เรา Download Jessie จาก Web Site ของ Raspberry Pi แล้วจะได้ File Image เราจะติดตั้งลงบน SD Card โดยใช้ Program Win32DiskImage เลือก Image File และเลือก Device ที่เป็น SD Card แล้วทำการ Write




หลังจากที่เขียน Image ลงบน SD Card แล้วให้เราลองเสียบ SD Card แล้ว Boot หนึ่งรอบ รอประมาณ 2 หรือ 3 นาที เพราะค่า Configuration ตั้งต้นจะถูกกำหนดมาในไฟล์ cmdline.txt ดังกรอบด้านล่าง

```
dwc_otg.lpm_enable=0 Console=serial0,115200 Console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait quiet init=/usr/lib/raspi-
config/init_resize.sh
```

หลังจากที่ Boot เสร็จเรียบร้อยแล้ว File cmdline.txt จะถูกเปลี่ยนค่า Configuration ซึ่งจะมีผลหากเราต้องการ Set ค่า IP Address แบบ direct Config

(จะรู้ได้ยังไงว่าการ Boot ของ Raspberry Pi เสร็จเรียบร้อยแล้วให้เราดูไฟที่แสดงค่า Status สีเขียวบน Board เมื่อ Boot เสร็จไฟสีเขียวนี้จะดับลงและไม่มีการกระพริบ)

ค่า Configuration ใน File cmdline.txt หลังจาก Boot เสร็จเรียบร้อยแล้วจะแสดงตามด้านล่าง ซึ่งหากเรา Login เข้าใช้ตัว Raspberry Pi จะอยู่ใน path /boot

```
dwc_otg.lpm_enable=0 Console=serial0,115200 Console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait
```

DO NOT COPY

## การกำหนดค่า IP แบบ Direct Config เพื่อใช้ในการกำหนดค่าครั้งแรก

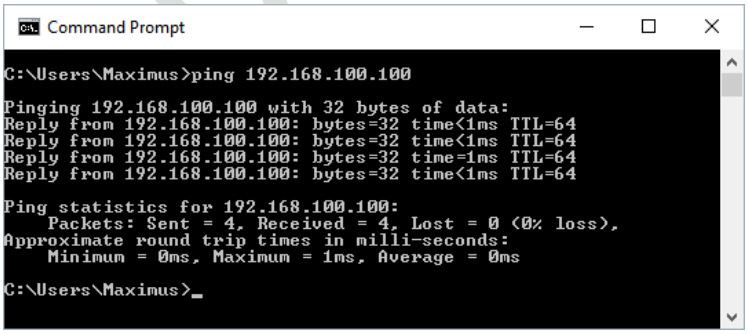
การกำหนดค่าแบบ Direct Config นี้ใช้เพียงสาย LAN ต่อกับ Raspberry Pi กับคอมพิวเตอร์ของเรา ซึ่งไม่จำเป็นต้องต่อ Monitor หรือ Keyboard เลย ทำให้งายและสะดวกในการกำหนดค่า Raspberry Pi เบื้องต้น นอกจากนี้ยังใช้ในกรณีฉุกเฉินที่ไม่สามารถเข้า Raspberry Pi จากปัญหาการกำหนดค่า Network Interface ไม่ถูกต้องอีกด้วย

หลังจากที่เราได้ทำการติดตั้ง OS ลงบน SD Card เรียบร้อยแล้วในการใช้งานจริงเราจำเป็นที่จะต้องเชื่อมต่อสาย LAN กับบอร์ด Raspberry Pi เพื่อใช้ในการ Remote เข้าไป Config ค่าต่าง ๆ แต่การที่เราจะ Remote เข้าไป Config ค่าต่าง ๆ นั้นจำเป็นที่จะต้องกำหนดค่า IP Address ของ Raspberry Pi วิธีการที่ง่ายที่สุดในการกำหนดค่า IP Address ในกรณีที่เราไม่มี Monitor, Keyboard, Mouse เราสามารถ Config ในไฟล์ cmdline.txt และใช้ Tool อยาง Telnet, Putty เข้าไปกำหนดค่าต่าง ๆ ในเบื้องต้นได้โดยการเพิ่มข้อความ “ ip=192.168.1.100” ท้ายไฟล์ cmdline.txt

```
dwc_otg.lpm_enable=0 Console=serial0,115200 Console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait ip=192.168.100.100
```

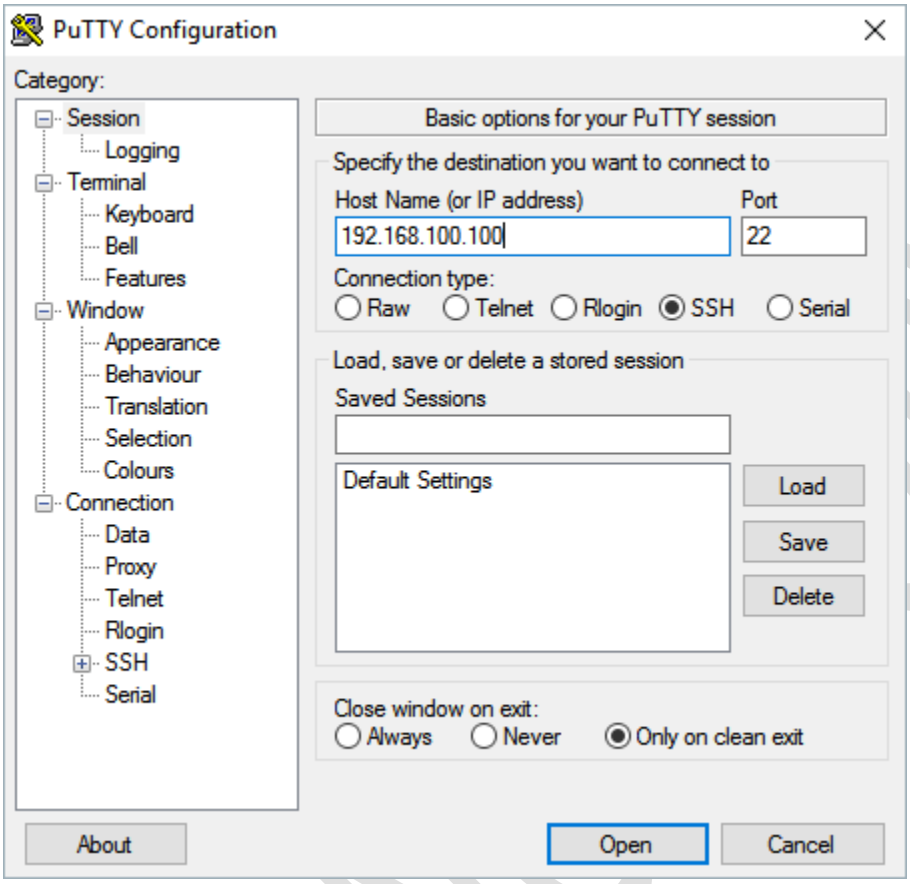
โดยรูปแบบการเชื่อมต่อที่ง่ายที่สุดคือน้ำ Raspberry Pi เสียบสาย LAN ต่อตรงกับ Notebook

หลังจากนั้นทดสอบโดยการ ping ไปที่ IP Address ที่เรากำหนดไว้แต่ต้องอย่าลืม Set IP Address ที่เครื่องเราเป็น IP Address วงเดียวกันกับ Raspberry Pi ด้วย

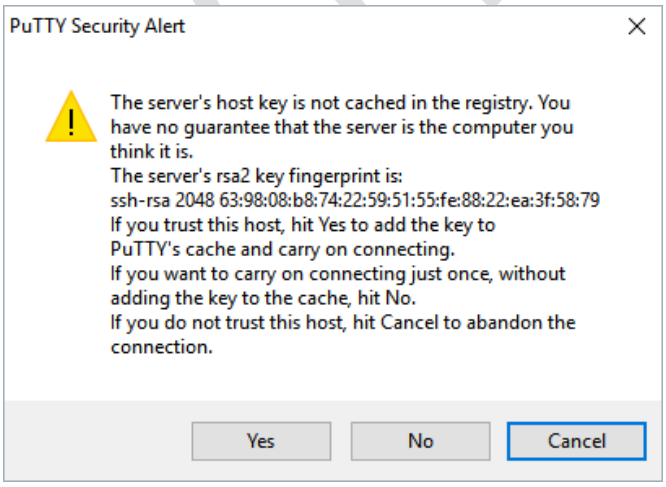



DO NOT COPY

เราสามารถใช้ Putty เพื่อ Remote เข้าไปก็ได้เช่นกัน

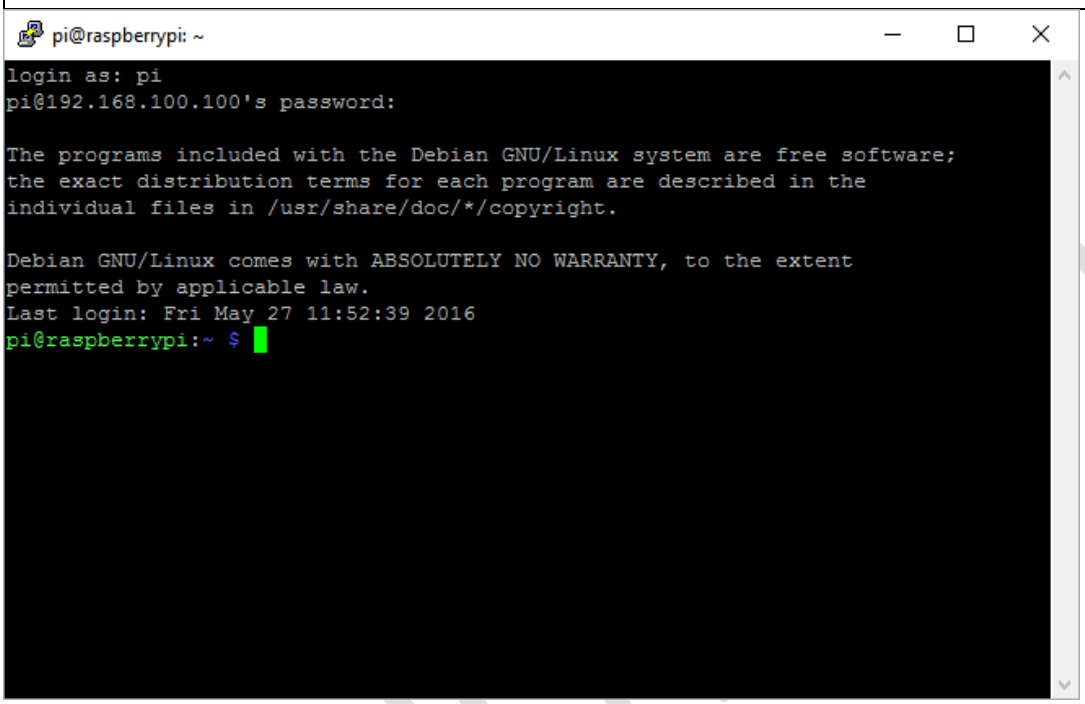



ปกติแล้วการเชื่อมต่อ SSL จะมีการยืนยันโดยใช้ RSA Key ในการเชื่อมต่อโดยเมื่อเรากดปุ่ม yes จะเป็นการเก็บและจำ Key นิไว้ที่ Program Putty ของเรา



เมื่อ Connect เข้ามาที่ Raspberry Pi แล้วจะมีการให้ใส่ Login และ Password โดยค่า Default ของ Raspberry Pi ตัว Login จะใช้

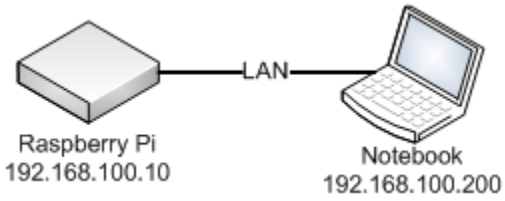
| Login: pi Password raspberry |
|---|




DO NOT

## การกำหนดค่า IP แบบ Static

ในการกำหนด IP Address ของ Raspberry Pi และ Linux Distro อื่นจะมีการกำหนดในรูปแบบคล้าย ๆ กัน

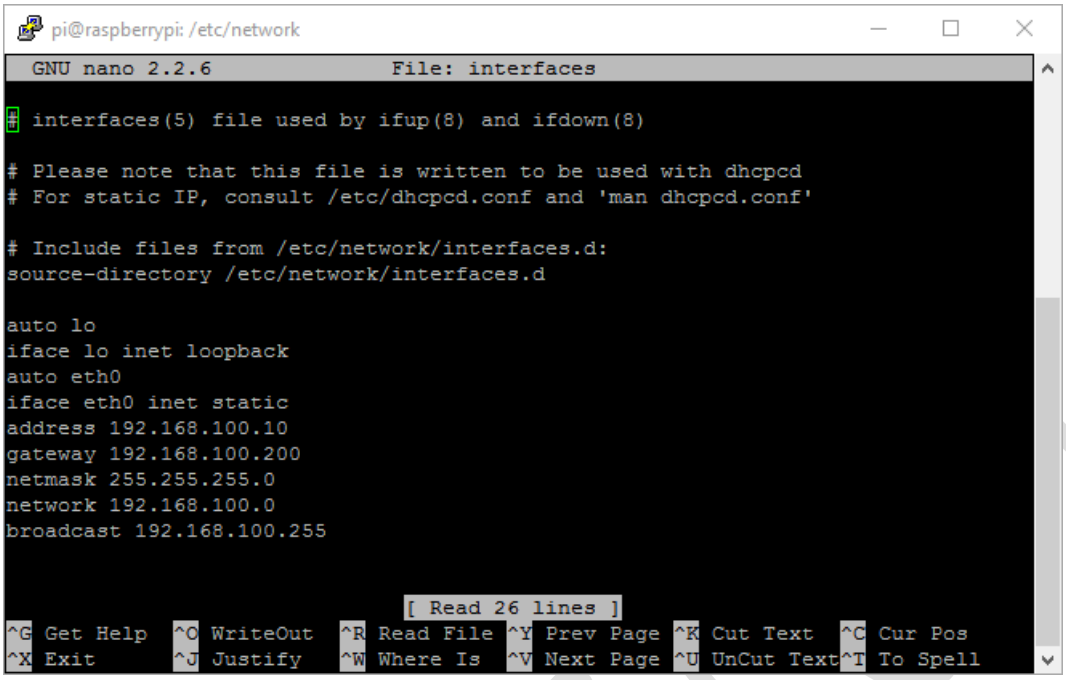



```
pi@raspberrypi: /etc/network
pi@raspberrypi:/etc/network $ cd /etc/network/
pi@raspberrypi:/etc/network $ sudo nano interfaces
```

และให้เราแก้ไข File Interface ตามนี้

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.100.10
gateway 192.168.100.200
netmask 255.255.255.0
network 192.168.100.0
broadcast 192.168.100.255
```

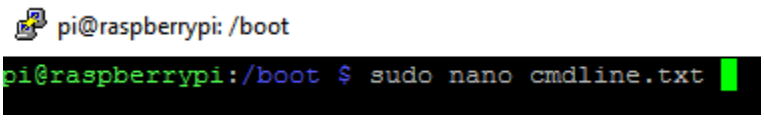
DO NOT COPY




ใน Linux Jessie นี้หาก Set แค่ Interface File นั้นจะไม่สามารถใช้งาน Network ตามค่า Config ของเราได้จำเป็นต้องกำหนดในส่วนของ File /etc/dhcpcd.conf เพิ่มต่อท้าย (สำหรับ OS ที่ใช้ Jessy เท่านั้น ส่วน OS ตัวเก่า ไม่ต้องทำ)

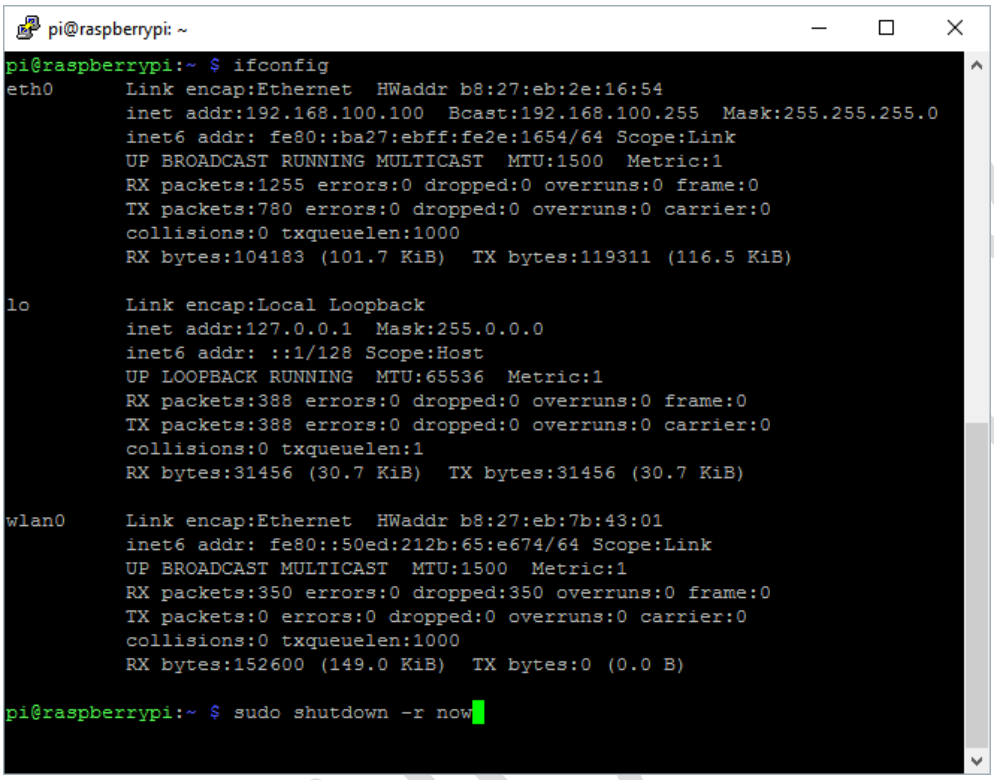
```
interface eth0
static ip_address=192.168.100.10/24
static routers=192.168.100.1
static domain_name_servers=192.168.100.200
```

นอกจากนี้เรายังจำเป็นต้องแก้ไขไฟล์ cmd Line .txt โดยนำ IP Address ออกมิเช่นนั้น IP Address แต่ยังคงเป็นตัวเดิมตามที่กำหนดใน cmdline .txt โดยไฟล์นี้จะอยู่ที่ /boot/cmdline.txt




```
dwc_otg.lpm_enable=0 Console=serial0,115200 Console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait
```

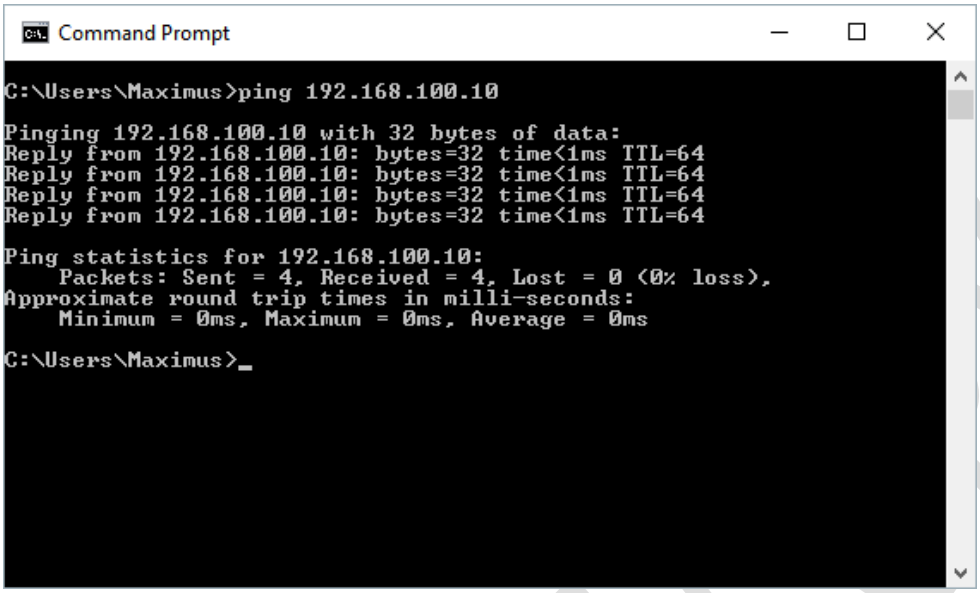
หากเราตรวจสอบ IP Address ด้วยคำสั่ง ifConfig จะพบว่ายังเป็น IP Address เดิมอยู่ ให้เราทำการ restart Raspberry Pi ด้วยคำสั่ง sudo shutdown -r now หรือ sudo reboot




หรือใช้การ Restart Interface eth0 ก็ได้ด้วยคำสั่ง

```
sudo ifdown eth0
sudo ifup eth0
```

เมื่อ Raspberry Pi Reboot มาเรียบร้อยแล้วเราทำการทดสอบโดยการ ping ไปที่ IP ที่เราแก้ไข




## ติดตั้ง RealVNC for Raspberry Pi

Download

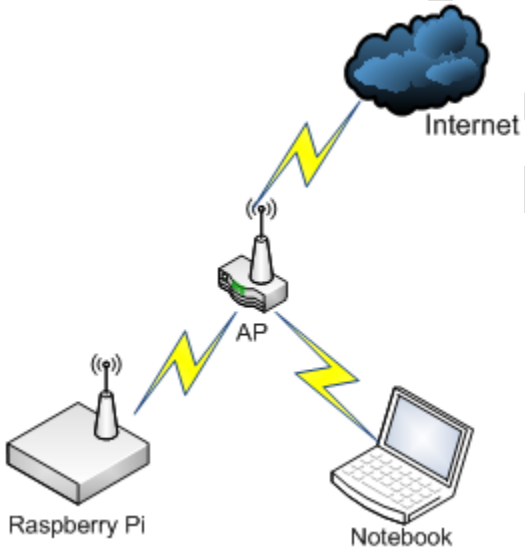
https://www.realvnc.com/download/binary/latest/debian/arm/

ได้ไฟล์ VNC-5.3.2-Linux-ARM-DEB.tar.gz ใช้ winSCP copy ไป

```
tar xvf VNC.tar.gz
sudo dpkg -i <VNC-Server-package-name>.deb [<VNC-Viewer-package-name>.deb]
```

## การตั้งค่า IP Address สำหรับ Wi-Fi ด้วย Command Line

ในการใช้งาน Raspberry Pi นั้นคงเป็นการลำบากที่เราจะหา Mouse, Keyboard และ Monitor เพื่อเริ่มต้นใช้งานในตอนแรกโดยเข้าใช้ GUI mode เพื่อที่จะ Config Wi-Fi อีกทางเลือกหนึ่งเราสามารถกำหนดค่า Wi-Fi Configuration โดยการใช้ Command




เริ่มต้นเราควรสแกน Wi-Fi ด้วยคำสั่ง

```
sudo iwlist wlan0 scan
```

NOT COPY

และดูในส่วนของ SSID ที่เราต้องการจะเชื่อมต่อ

DO NOT COPY

```
      Cell 02 - Address: BC:98:89:85:48:10
                Channel:12
                Frequency:2.467 GHz (Channel 12)
                Quality=58/70  Signal level=-52 dBm
                Encryption key:on
                ESSID:"BUAKAEW AP01"
                Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                          9 Mb/s; 12 Mb/s; 18 Mb/s
                Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s
                Mode:Master
                Extra:tsf=0000000000000000
                Extra: Last beacon: 90ms ago
                IE: Unknown: 000C4255414B4145575F41503031
                IE: Unknown: 010882848B960C121824
                IE: Unknown: 03010C
                IE: Unknown: 050400010100
                IE: Unknown: 2A0104
                IE: Unknown: 32043048606C
                IE: Unknown: 2D1A4E181EFFFF000000000000000000000000000000
000000
                IE: Unknown: 3D160C000000000000000000000000000000000000

                IE: Unknown: 4A0E14000A00B400C800140005001900
                IE: Unknown: 7F0101
                IE: IEEE 802.11i/WPA2 Version 1
                    Group Cipher : CCMP
                    Pairwise Ciphers (1) : CCMP
                    Authentication Suites (1) : PSK
                IE: Unknown: DD180050F20201010000003A4000027A4000042435E00
00
                IE: Unknown: DD1E00904C334E181EFFFF000000000000000000000000
00000000000000
                IE: Unknown: DD1A00904C340C000000000000000000000000000000
000000
                IE: Unknown: DD0600E04C020160
                IE: Unknown: DD180050F204104A000110104400010210490006003 7
20
```

ให้เราทำการเข้าไปแก้ไขไฟล์ wpa_supplicant.conf

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

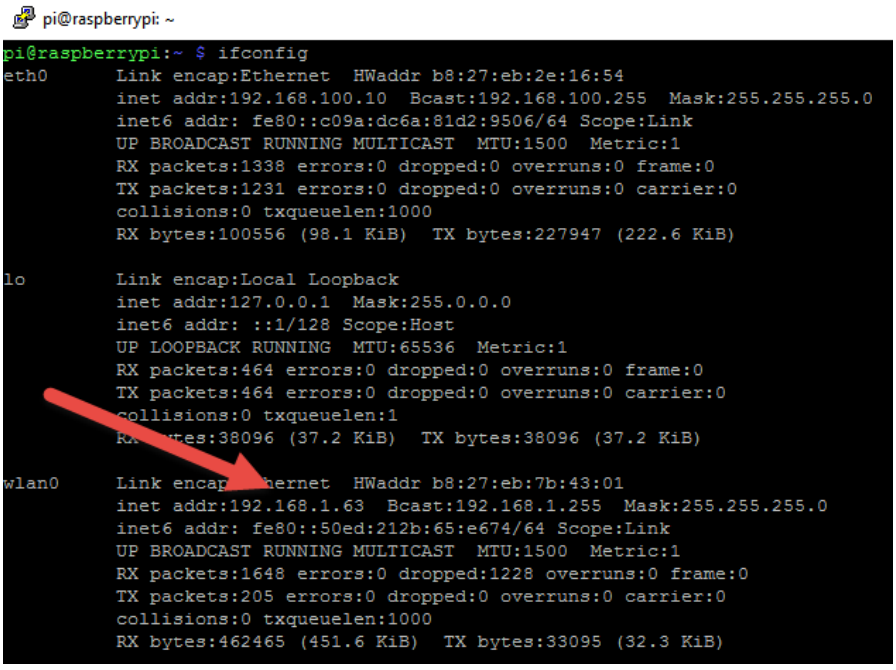
โดยเพิ่มในส่วนของ network

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
 ssid="BUAKAEW_AP01"
 psk="BuaKaew159357"
 key_mgmt=WPA-PSK
}
```

หลังจากนั้นทำการ restart Wi-Fi interface ด้วยคำสั่ง

```
sudo ifdown wlan0
sudo ifup wlan0
```

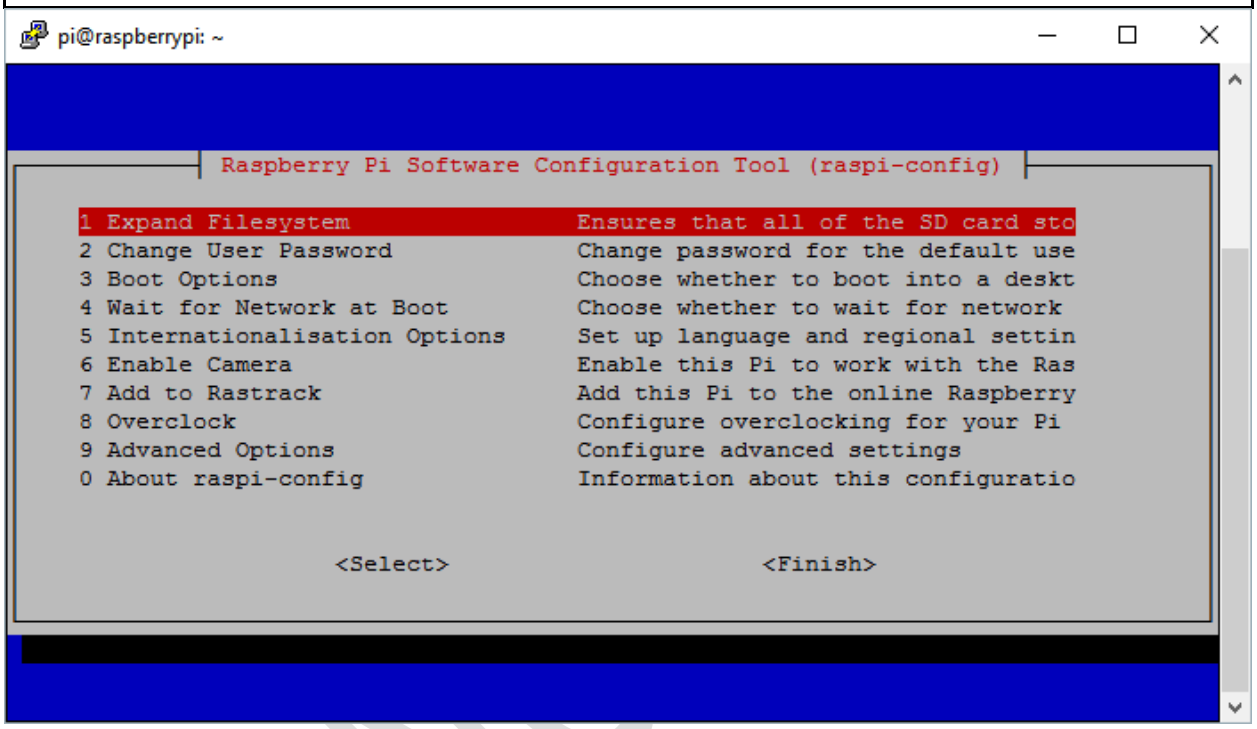
และตรวจสอบ IP Address อีกครั้งหนึ่งจะพบว่า wireless lan จะได้ IP Address ของ access point



# การตั้งค่าทั่วไปของ Raspberry Pi

ในการกำหนดค่าพื้นฐานของ Raspberry Pi ก่อนที่เราจะเริ่มนั้นเราจะใช้คำสั่ง

```
sudo raspi-config
```




## การขยายขนาด Space ใน SD Card ให้เต็มพื้นที่

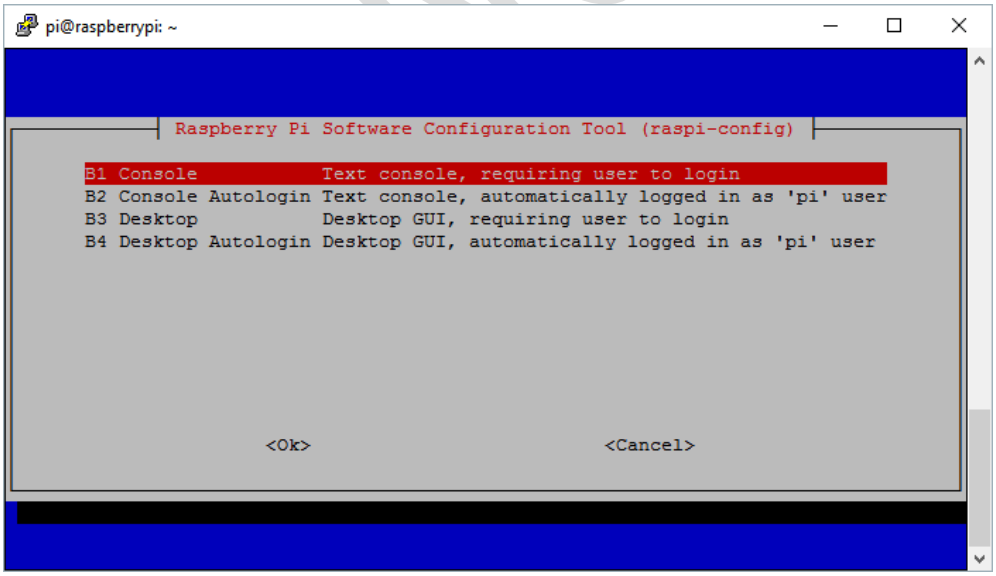
ในการ Burn SD Card จาก Image File ของ Raspberry Pi ครั้งแรกจะมีเนื้อที่ที่ถูกใช้งานใน SD Card ประมาณ 4 GB หากเนื้อที่ SD Card เรามีมากกว่า 4 GB

การที่เราจะมองเห็น Space ของการ์ดได้เต็ม 100% นั้นจำเป็นที่ต้องมีการขยาย File System โดยการเลือกข้อ 1 Expand File System

เมื่อทำการขยายเสร็จจะมีขึ้นหน้าจอ Confirm

## การกำหนดการ Boot เข้าหน้าจอของ Raspberry Pi

ในการ boot เข้าหน้าจอของ Raspberry Pi นั้นหลังจากที่เรา burn Image os ของ Raspberry Pi ลงไปแล้วนั้นค่า Default จัดบูธเข้าในโหมด g u i เราสามารถเปลี่ยนได้โดยการเข้าไปที่เมนู ข้อ 3 boot option

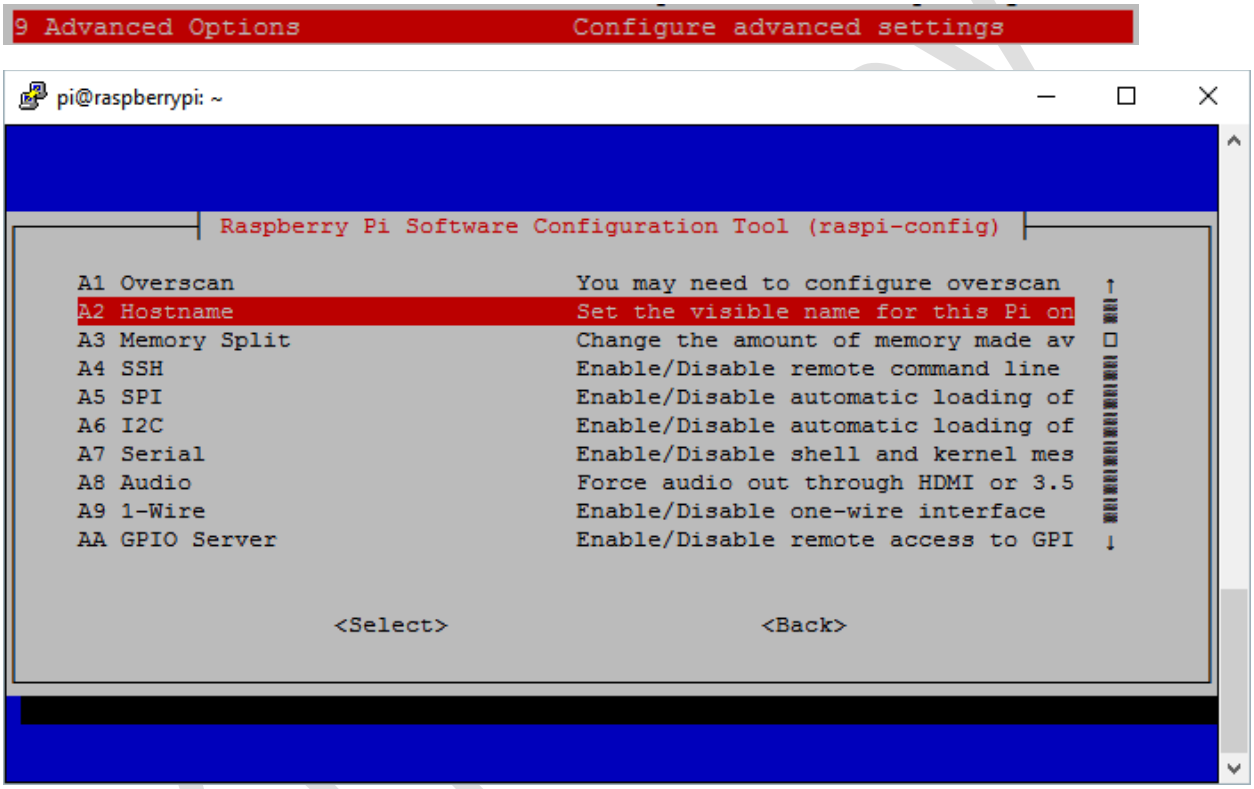


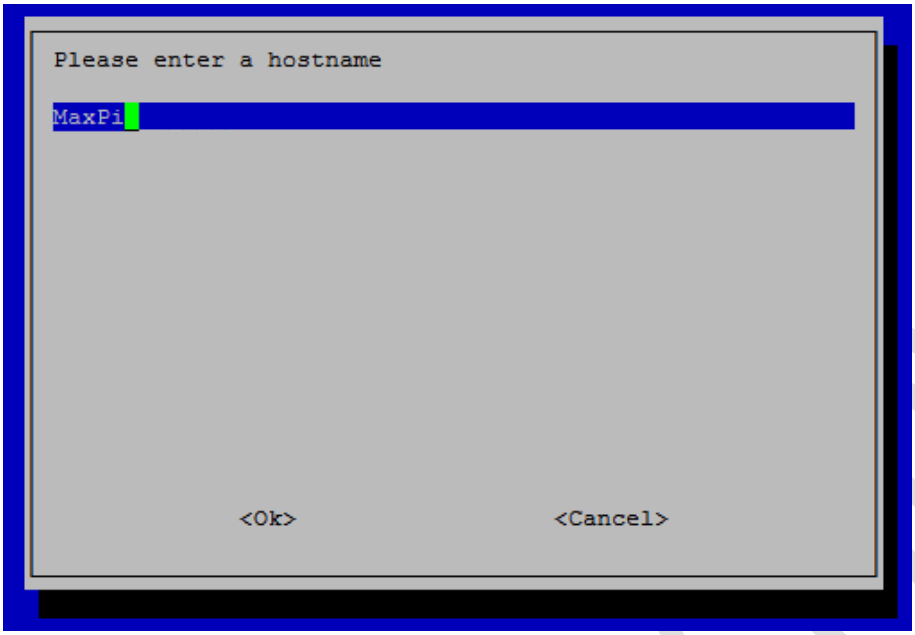

DO NOT COPY

โดยส่วนใหญ่แล้วจะเลือกการ Boot แบบเข้า Console เพื่อให้ Login จะเป็นการประหยัด Resource มากกว่าการเข้าแบบ GUI และคำนึงถึงเรื่อง Security ด้วย

## การกำหนด Raspberry Pi Name

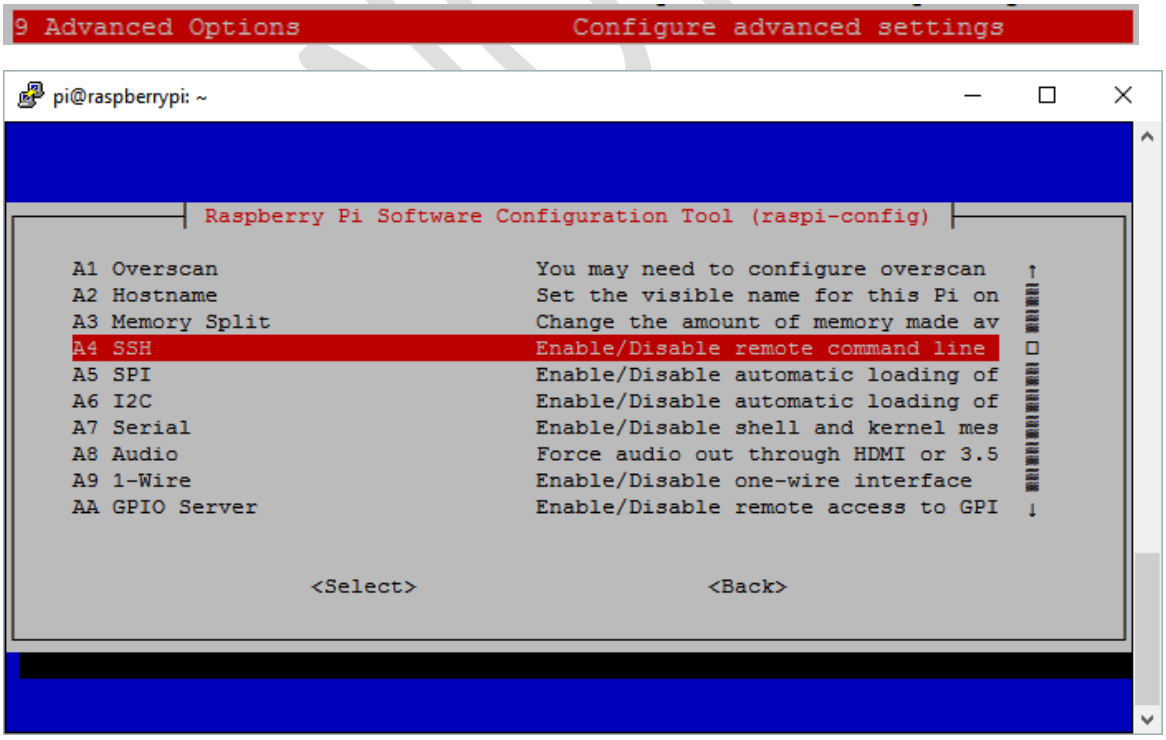
ในการเปลี่ยนชื่อ Raspberry Pi ของเรานั้นให้เข้าไปที่เมนู 9 Advanced Options เลือก Host Name






## การกำหนด SSH

ในการกำหนดให้ Raspberry Pi เปิด Secure Shell ที่ให้เราสามารถ Telnet แบบมีการเข้ารหัสข้อมูลในการรับส่ง หรือเรียกสั้น ๆ ว่า SSH นั้น ให้เข้าไปที่เมนู 9 Advanced Options เลือก A4 SSH




NOT COPY

และ Confirm ในการ Enable SSH

นอกจากนี้ยังมีการกำหนดค่าการใช้งานอื่น ๆ เช่น SPI I2C Serial 1-Wire audio GPIO ก็ทำลักษณะเดียวกัน

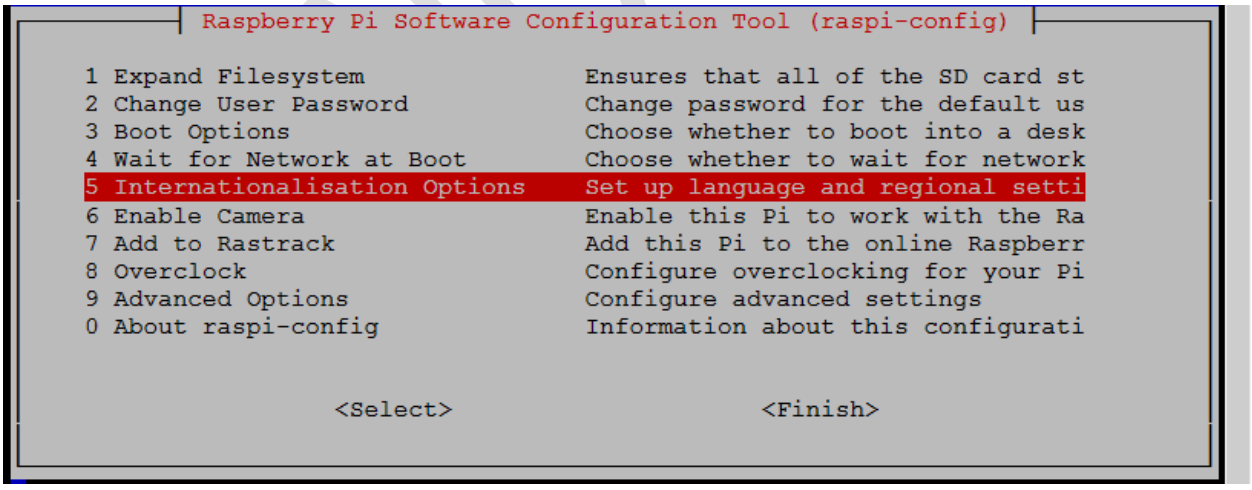
เมื่อเราแก้ไขค่า Config ต่าง ๆ ใน Raspberry Pi Config แล้วกดปุ่ม finish จะให้เรา Boot Raspberry Pi

COPY

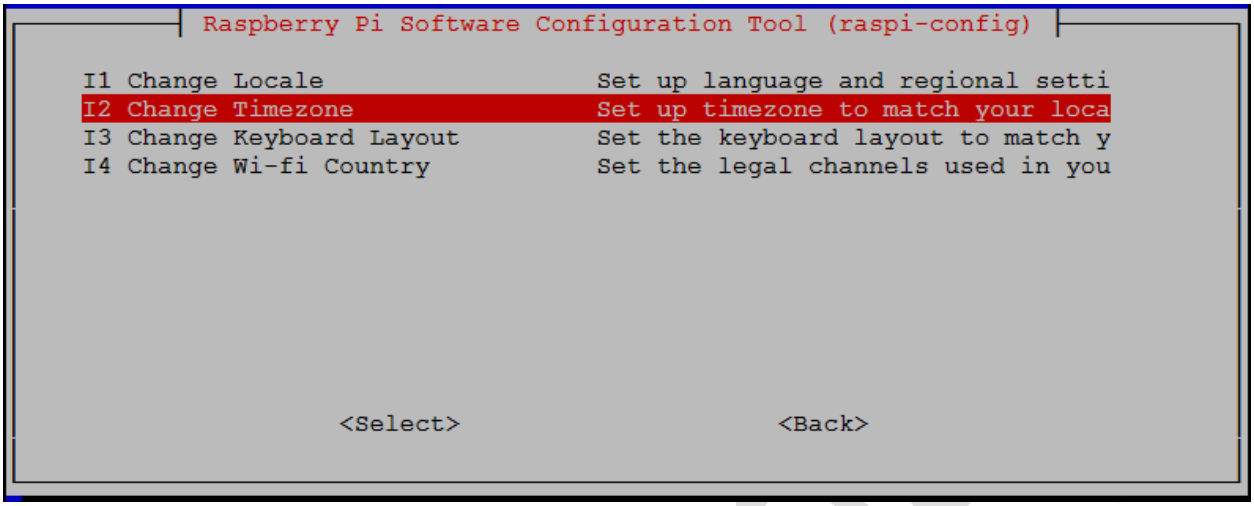
NOT COPY

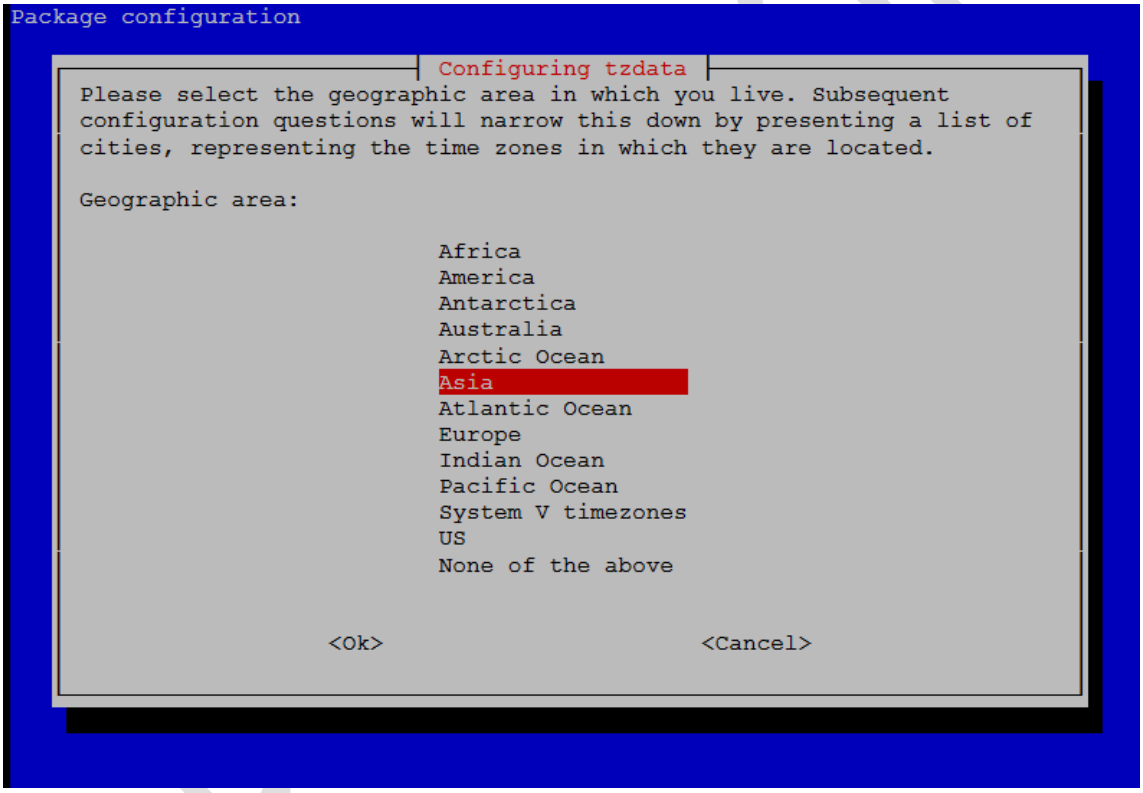
## การตั้งเวลา Raspberry Pi

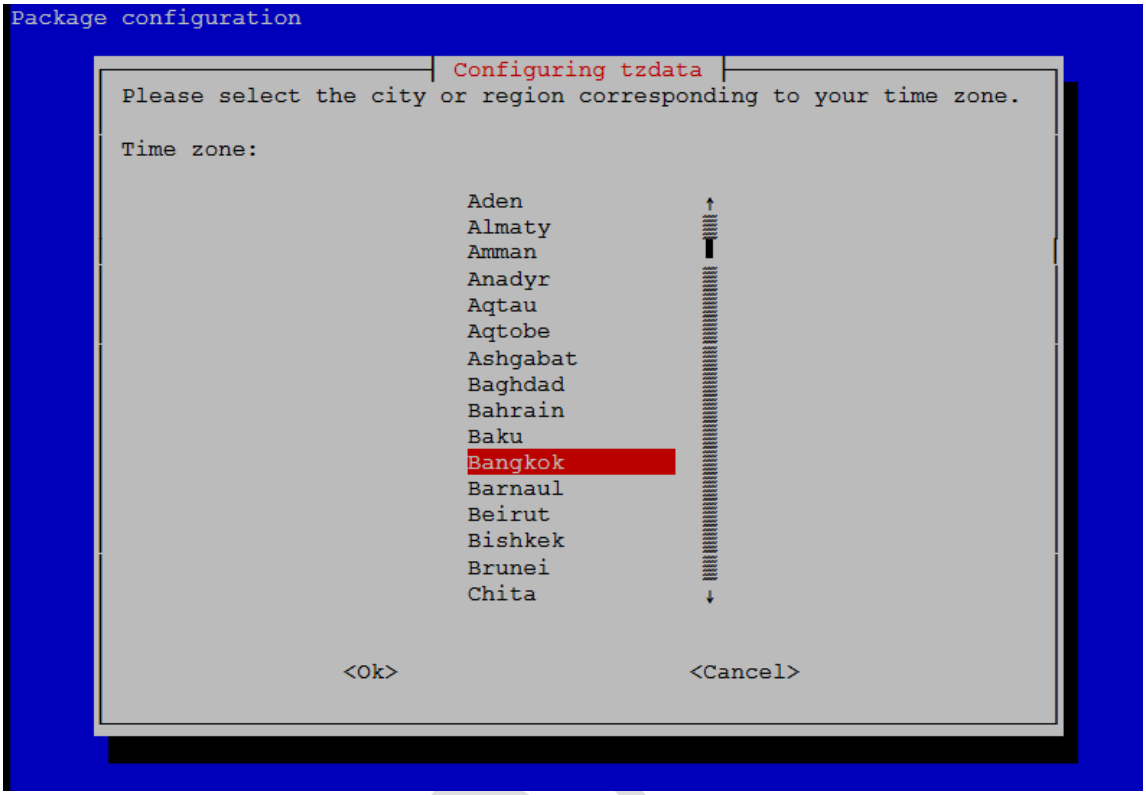
การกำหนดเวลาของ Raspberry Pi สามารถกำหนดได้หลายแบบ แบบง่ายสุดคือการกำหนดใน International Option > Time Zone (การตั้งเวลาแบบนี้ Raspberry Pi จำเป็นต้องมีการเชื่องต่อ internet เพื่อ Update เวลากับ NTP Server)




และเลือก Time Zone เป็น Asia / Bangkok


Raspberry Pi Software Configuration Tool (raspi-config)
I1 Change Locale
Set up language and regional setti
I2 Change Timezone
Set up timezone to match your loca
I3 Change Keyboard Layout
Set the keyboard layout to match y
I4 Change Wi-fi Country
Set the legal channels used in you
<Select>
<Back>


Package configuration
Configuring tzdata
Please select the geographic area in which you live. Subsequent
configuration questions will narrow this down by presenting a list of
cities, representing the time zones in which they are located.
Geographic area:
Africa
America
Antarctica
Australia
Arctic Ocean
Asia
Atlantic Ocean
Europe
Indian Ocean
Pacific Ocean
System V timezones
US
None of the above
<Ok>
<Cancel>




ใช้คำสั่ง date เพื่อดูวันเวลา

```
pi@MaxPi:~ $ date
Wed  3 Aug 22:33:56 ICT 2016
pi@MaxPi:~ $ _
```

แต่ถ้าต้องการกำหนดเวลาแบบ Manual จะใช้ Command

```
sudo date -s “Wed Aug 3 22:30:00 ICT 2016”
```

```
pi@MaxPi:~ $ sudo date -s "Wed Aug 3 22:30:00 ICT 2016"
Wed  3 Aug 22:30:00 ICT 2016
pi@MaxPi:~ $ date
Wed  3 Aug 22:30:04 ICT 2016
```

## การตั้งค่า Keyboard

กำหนดค่าของ Keyboard Layout ที่ File /etc/default/keyboard กำหนด XKBLAYOUT=”us”

```
# KEYBOARD CONFIGURATION FILE

# Consult the keyboard(5) manual page.

XKBMODEL="pc105"
XKBLAYOUT="us"
XKBVARIANT=""
XKBOPTIONS=""

BACKSPACE="guess"
```

DO NOT COPY

## การอัพเดท Raspberry Pi

ในครั้งแรกเราจำเป็นที่จะต้อง Update Package List เพื่อตรวจหารายการที่เป็น Program ใหม่ ๆ จาก Repository หรือจาก Server ของระบบปฏิบัติการนั่นเอง ว่ามีการ update อะไรใหม่ ๆ หรือไม่ด้วยคำสั่ง

```
sudo apt-get update
```

```
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://mirrordirector.raspbian.org jessie InRelease [14.9 kB]
Get:2 http://archive.raspberrypi.org jessie InRelease [13.2 kB]
Get:3 http://mirrordirector.raspbian.org jessie/main armhf Packages [8,980 kB]
Get:4 http://archive.raspberrypi.org jessie/main armhf Packages [144 kB]
Get:5 http://archive.raspberrypi.org jessie/ui armhf Packages [8,966 B]
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Get:6 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [37.5 kB]
Get:7 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [70.3 kB]
Get:8 http://mirrordirector.raspbian.org jessie/rpi armhf Packages [1,356 B]
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Fetched 9,271 kB in 44s (207 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $
```

หลังจากการให้เราทำการอัพเกรด Install Package ต่าง ๆ เมื่อ ทำการ update หรือตรวจสอบรายชื่อใหม่ ๆ มาแล้ว upgrade ก็จะเป็นการดาวน์โหลดและติดตั้ง Program ใหม่ ๆ เหล่านั้นลงไปใน Raspberry Pi นั่นเอง (อันนี้จะนาน)

```
sudo apt-get upgrade
```

```
Removing 'diversion of /boot/start_x.elf to /usr/share/rpikernelhack/start_x.elf by rpikerne
lhack'
Removing 'diversion of /boot/fixup.dat to /usr/share/rpikernelhack/fixup.dat by rpikernelhac
k'
Removing 'diversion of /boot/fixup_cd.dat to /usr/share/rpikernelhack/fixup_cd.dat by rpiker
nelhack'
Removing 'diversion of /boot/fixup_db.dat to /usr/share/rpikernelhack/fixup_db.dat by rpiker
nelhack'
Removing 'diversion of /boot/fixup_x.dat to /usr/share/rpikernelhack/fixup_x.dat by rpikerne
lhack'
Removing 'diversion of /boot/bootcode.bin to /usr/share/rpikernelhack/bootcode.bin by rpiker
nelhack'
Removing 'diversion of /boot/LICENCE.broadcom to /usr/share/rpikernelhack/LICENCE.broadcom b
y rpikernelhack'
Setting up libraspberrypi0 (1.20160620-1) ...
Setting up libraspberrypi-dev (1.20160620-1) ...
Setting up libraspberrypi-doc (1.20160620-1) ...
Setting up libraspberrypi-bin (1.20160620-1) ...
Setting up python-picamera (1.11) ...
Setting up python3-picamera (1.11) ...
Processing triggers for libc-bin (2.19-18+deb8u4) ...
Processing triggers for initramfs-tools (0.120+deb8u2) ...
pi@raspberrypi:~ $
```

# การติดตั้ง Remote Desktop

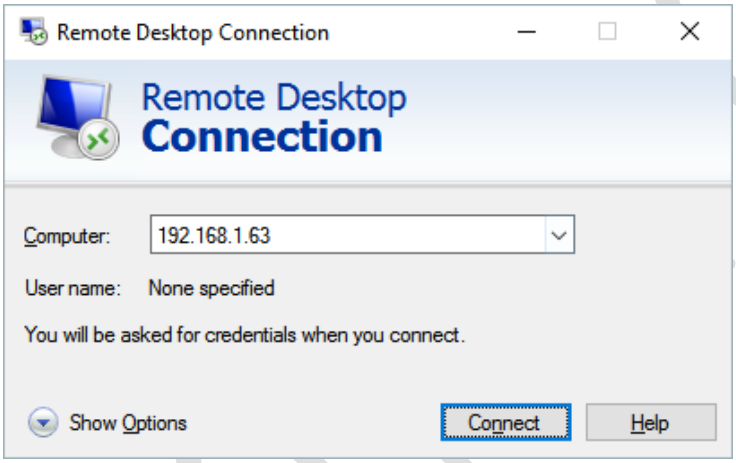
ในการใช้งาน Raspberry Pi นอกจากใช้ Secure Shell เพื่อใช้ Command Line ในการสั่งงานแล้ว ยังสามารถใช้การ Remote เข้าไปใช้งานในลักษณะของ Graphic Mode โดยติดตั้ง Remote desktop ของ Raspberry Pi นั้นจะมีการติดตั้งใช้งาน 2 ตัวได้แก่

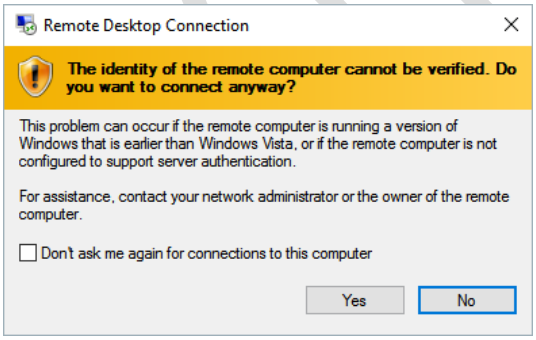
## การติดตั้ง XRDP

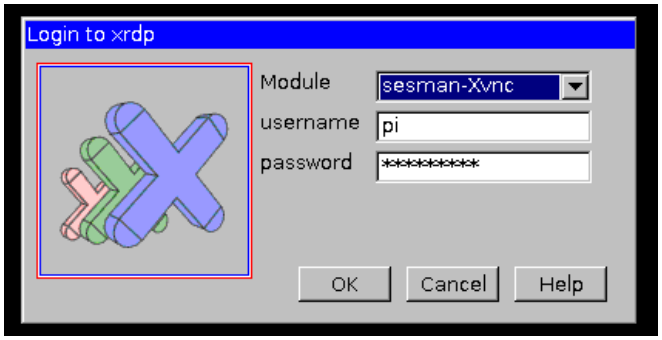
XRDP ใช้โปรโตคอล Remote desktop protocol ในการใช้งาน ซึ่งเราจำเป็นจะต้องติดตั้ง Package

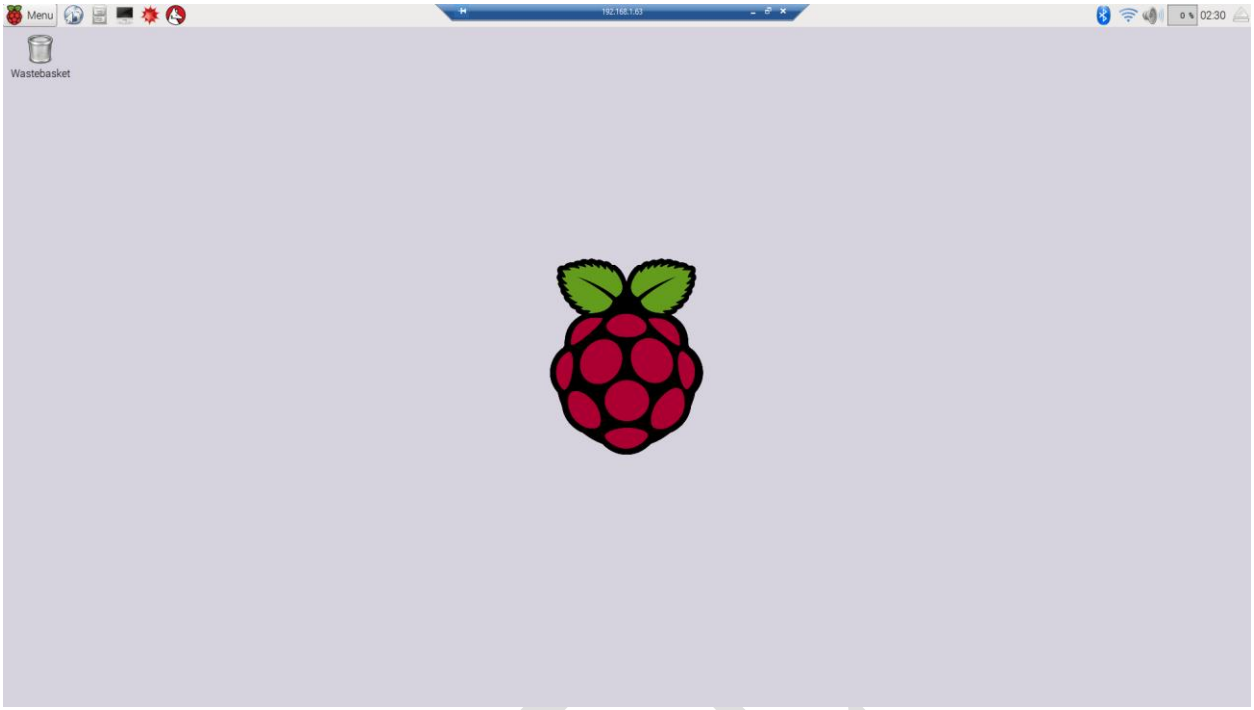
```
sudo apt-get install xrdp
```

ในการใช้งานจากฝั่ง client เราสามารถใช้ Program Remote desktop หรือ Smart TTY






แนะนำให้ใช้ Program SmartTTY เพราะเราสามารถเปิด Program ที่ Run บน XWindows ให้สามารถมาทำงานบนหน้าจอเครื่องที่ Remote ได้

DO NOT C

## การติดตั้ง VNC

สำหรับคนที่ไม่ชอบใช้ Remote Desktop เราสามารถใช้ VNC แทนได้โดยติดตั้ง TightVNC Server ซึ่งเป็น Software สำหรับ Remote ที่ไม่มีค่าใช้จ่าย

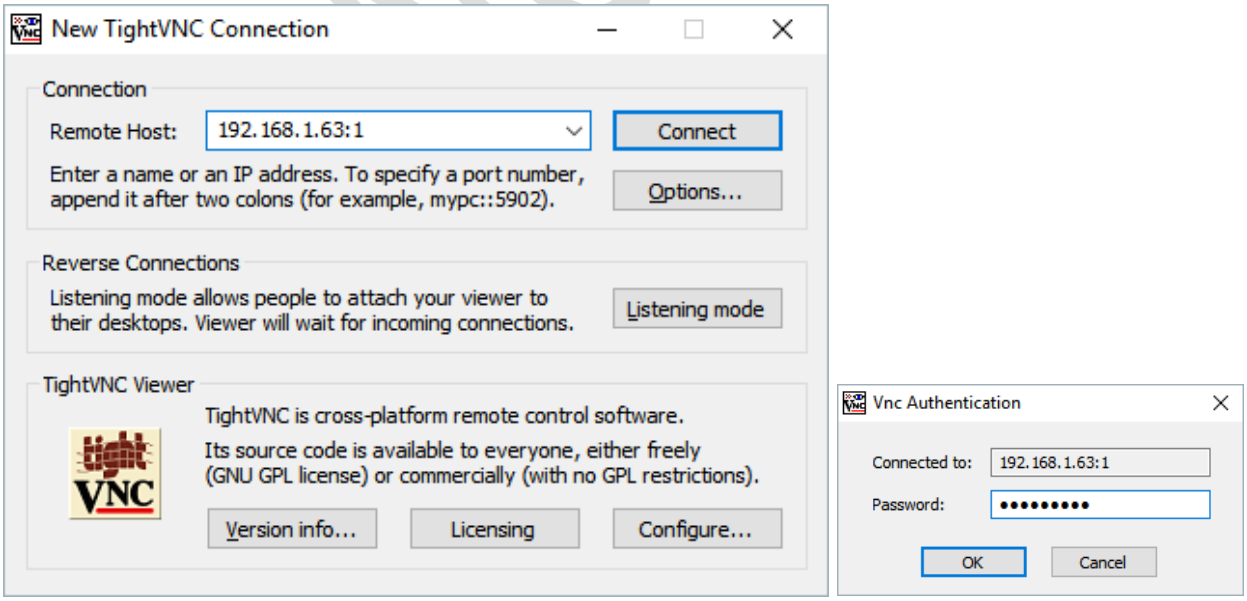
```
sudo apt-get install tightvncserver
```

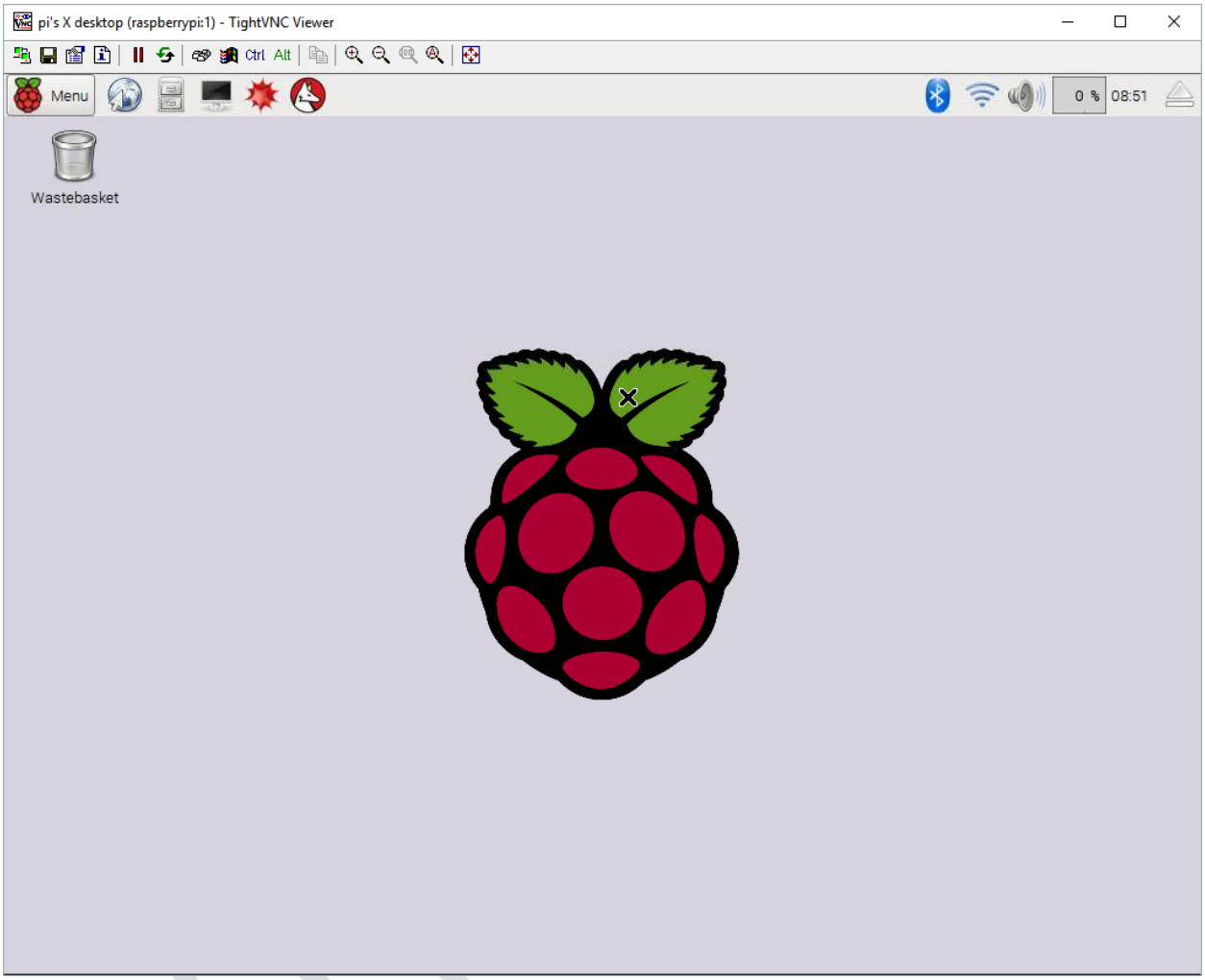
หลังจากนั้นรันใช้ VNC server โดยต้องกำหนด Password ก่อน

```
tightvncserver
```

```
pi@raspberrypi:~ $ tightvncserver

You will require a password to access your desktops.

Password:
Verify:

New 'X' desktop is raspberrypi:1

Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log
```

การใช้งานเพลงเปิด VNC ขึ้นมาและใส่ IP Address : 1




pi's X desktop (raspberrypi:1) - TightVNC Viewer
Ctrl
Alt
Menu
0 %
08:51
Wastebasket

## การใช้งาน GPIO

ในการใช้งาน Raspberry Pi นั้นเราสามารถใช้ขา GPIO ต่าง ๆ ในการเชื่อมต่อกับอุปกรณ์ภายนอกและ Sensor ซึ่งรูปแบบตาจะเป็นตามรูปด้านบนจะสังเกตเห็นว่า Pi นั้นสามารถใช้ได้ทั้ง I2C, SPI และ UART ดังนั้นเราจำเป็นต้องรู้ว่าขาไหนสามารถใช้งาน Function อะไรได้บ้าง ระดับแรงดันที่ใช้ของแต่ละขาจะอยู่ที่ 3.3 v การจัดเรียงขา GPIO เป็นตามภาพ

**Raspberry Pi B**
**Rev 2 P1 GPIO Header**

| | Pin No. | | |
|---|---|---|---|
| 3.3V | 1 | 2 | 5V |
| GPIO2 | 3 | 4 | 5V |
| GPIO3 | 5 | 6 | GND |
| GPIO4 | 7 | 8 | GPIO14 |
| GND | 9 | 10 | GPIO15 |
| GPIO17 | 11 | 12 | GPIO18 |
| GPIO27 | 13 | 14 | GND |
| GPIO22 | 15 | 16 | GPIO23 |
| 3.3V | 17 | 18 | GPIO24 |
| GPIO10 | 19 | 20 | GND |
| GPIO9 | 21 | 22 | GPIO25 |
| GPIO11 | 23 | 24 | GPIO8 |
| GND | 25 | 26 | GPIO7 |

Key

| | |
|---|---|
| Power + | UART |
| GND | SPI |
| I²C | GPIO |

**Raspberry Pi B+**
**B+ J8 GPIO Header**

| | Pin No. | | |
|---|---|---|---|
| 3.3V | 1 | 2 | 5V |
| GPIO2 | 3 | 4 | 5V |
| GPIO3 | 5 | 6 | GND |
| GPIO4 | 7 | 8 | GPIO14 |
| GND | 9 | 10 | GPIO15 |
| GPIO17 | 11 | 12 | GPIO18 |
| GPIO27 | 13 | 14 | GND |
| GPIO22 | 15 | 16 | GPIO23 |
| 3.3V | 17 | 18 | GPIO24 |
| GPIO10 | 19 | 20 | GND |
| GPIO9 | 21 | 22 | GPIO25 |
| GPIO11 | 23 | 24 | GPIO8 |
| GND | 25 | 26 | GPIO7 |
| DNC | 27 | 28 | DNC |
| GPIO5 | 29 | 30 | GND |
| GPIO6 | 31 | 32 | GPIO12 |
| GPIO13 | 33 | 34 | GND |
| GPIO19 | 35 | 36 | GPIO16 |
| GPIO26 | 37 | 38 | GPIO20 |
| GND | 39 | 40 | GPIO21 |

## การสั่งงานผ่าน GPIO ด้วย Command line

**GPIO output**

อย่างแรกเราต้องกำหนด GPIO ว่าขาไหนจะให้เป็น output

```
echo "4" > /sys/class/gpio/export     //ตั้ง GPIO4 เป็น GPIO
echo "out" > /sys/class/gpio/gpio4/direction  //กำหนดโหมดให้เป็น GPIO out
echo "1" > /sys/class/gpio/gpio4/value    // ส่งค่า 1 ออกขา GPIO 4
echo "0" > /sys/class/gpio/gpio4/value    // ส่งค่า 0 ออกขา GPIO 4
```

**GPIO Input**

```
echo "22" > /sys/class/gpio/export     //ตั้ง GPIO4 เป็น GPIO
echo "in" > /sys/class/gpio/gpio22/direction  //กำหนดโหมดให้เป็น GPIO out
cat /sys/class/gpio/gpio22/value     // อ่านค่าขา GPIO 4
```

การยกเลิกการใช้ GPIO

```
echo "22" > /sys/class/gpio/unexport
```

DO NOT COPY

# การเขียน Program Python พื้นฐาน

ในการใช้งาน Microcontroller และ embedded System ส่วนใหญ่นั้นนอกจาก ภาษา C แล้วยังมีภาษาที่นิยมใช้อีก คือ Python ซึ่งเป็นภาที่มี Library ให้เลือกใช้สำหรับเชื่อมต่อกับ Hardware ให้อย่างมากมาย และมีรูปแบบ Syntax ที่ไม่ซับซ้อนจึงง่ายต่อการพัฒนา

Python ก็คือมันเป็นทั้ง Cross Platform และมี Open Source License โดยที่มีอิสระในการแก้ไข Library ต่าง ๆ ที่ Python ให้มา รวมถึงการนำ Software ที่พัฒนาขึ้นจาก Python ไปทำประโยชน์ทางธุรกิจได้อย่างเต็มที่โดยไม่เสียเงินนอกจากนี้ตัวโครงสร้างและ Syntax ของภาษาค่อนข้างอ่านง่าย เข้าใจง่าย และมี Object ด้าน Data Structure รองรับอยู่หลายแบบแล้ว ทำให้ Python เป็นของ ฟรี และของ ดี ที่ช่วยให้พัฒนา Program ได้เร็วขึ้น

## ตัวแปรในภาษา python

การใช้งานตัวแปรในภาษา python นั้นค่อนข้างที่จะง่ายเนื่องด้วยไม่จำเป็นต้องประกาศตัวแปรก็สามารถที่จะนำไปใช้งานได้

```
X = 10

Y = 20.5

Uname = “maximus”
```

## Syntax ในภาษา python

ลักษณะซินแท็กของภาษา python ไม่จำเป็นต้องมีการปิดบรรทัดด้วยโคลอนหรือการใช้เครื่องหมายปีกกาในการกำหนดช่วงของการเขียน Program ซึ่งจะใช้การแท็กหรือเคาะ Space ในการกำหนดช่วงของการเขียน Program โดยใน 1 บรรทัดจะเขียนการทำงานได้แค่สเตจเดียว

DO NOT COPY

| | |
|---|---|
| #include <stdio.h><br><br>Void main(void){<br><br>Int x = 1;<br><br>while (x < 10){<br><br>printf(“%d\n”);<br><br>x++;<br><br>}<br><br>} | x = 1<br><br>While x<10:<br><br>Print x<br><br>x++ |

## การกำหนดตัวแปร

Python ไม่จำเป็นต้องประกาศตัวแปรก่อนใช้งาน สามารถกำหนดค่าขึ้นมา และเรียกใช้ได้เลย แต่การตัวชื่อตัวแปรของ ไพทอนต้องเป็นไปตามกฎดังนี้

- ขึ้นต้นด้วยตัวอักษรตั้งแต่ 1 ตัวขึ้นไป
- ตัวแปรห้ามมีช่องว่าง
- ห้ามมีสัญลักษณ์พิเศษเช่น #,?,$,... (คนที่เขียน PHP มาอย่าลงไปใส่ $ เข้านะครับ)
- ตัวแปรต้องไม่ซ้ำกับคำสงวน ดูคำสงวนได้ท้ายบทความนี้
- ตัวแปรใช้ตัวพิมพ์เล็กพิมพ์ใหญ่ มีความหมายต่างกัน (case-sensitive แบบเดียวกับ PHP)

DO NOT COPY

## ชนิดข้อมูล

ชนิดข้อมูลของไพทอน ไพทอนมีชนิดข้อมูลพื้นฐานอยู่ 5 ชนิดดังนี้

- Numbers เก็บข้อมูลตัวเลข

```
x=10
y=10.5
```

- String เก็บข้อมูลตัวอักษร

```
x=”maximus”
```

- List เก็บข้อมูลได้มากว่า 1 ค่าใน 1 ตัวแปร หรือที่เรียกว่า compound type

```
var_list = [ 'abcd', 123 1.23, 'Mindphp.com', 20.2 ]
var_tinylist = [123, 'mindphp']
print (var_list)  # แสดงค่าทั้งหมดในตัวแปล var_list
print (var_list[0]) # แสดงรายการแรกของตัวแปล var_list
print (var_list[1:3]) # แสดงรายการที่ สอง ถึงสามรายการถัดไป
print (var_list[2:]) # แสดงรายการที่สามถึง สุดท้าย
print (var_tinylist * 2) # แสดงรายการทั้งหมดของตัวแปล var_tinylist สองครั้ง
print (var_list + var_tinylist) # แสดงรายการทั้งหมดของ var_list, var_tinylist รวมกัน
```

- Tuple อ่านว่า "ทูเพิล" เก็บข้อมูลได้มากว่า 1 ค่าใน 1 ตัวแปร ใช้สำหรับเก็บลำดับ หรือที่เรียกว่า sequence type

ทูเบิลเป็นตัวแปลที่เก็บข้อมูลแบบลำดับ (sequence data type) คล้าย ๆ กับ ตัวแปลชนิดลิส ทุเปิลส์จะเก็บค่าของสมาชิกแต่ละตัวไว้ใน เครื่องหมาย (...,...,...)

DO NOT COPY

ตัวอย่าง Tuples.py

```
var_tuples = ( 'abcd', 123 1.23, 'Mindphp.com', 20.2 )
var_tinytuples = (123, 'mindphp')

print (var_tuples)  # แสดงค่าทั้งหมดในตัวแปล var_tuples
print (var_tuples[0]) # แสดงรายการแรกของตัวแปล var_tuples
print (var_tuples[1:3]) # แสดงรายการที่ สอง ถึงสามรายการถัดไป
print (var_tuples[2:]) # แสดงรายการที่สามถึง สุดท้าย
print (var_tinytuples * 2) # แสดงรายการทั้งหมดของตัวแปล var_tuples สองครั้ง
print (var_tuples + var_tinytuples) # แสดงรายการทั้งหมดของ var_tuples, var_tinytuples รวมกัน
```

ผลที่ได้

```
('abcd', 123, 1.23, 'Mindphp.com', 20.2)
abcd
(123, 1.23)
(1.23, 'Mindphp.com', 20.2)
(123, 'mindphp', 123, 'mindphp')
('abcd', 123, 1.23, 'Mindphp.com', 20.2, 123, 'mindphp')
```

Tuples เมื่อสร้างขึ้นมาแล้วเราไม่สามารถเพิ่มค่าให้ตัวแปลได้
เช่น ได้เช่น ตามตัวอย่าง ตัวแปล var_tuples มี index ทั้งหมด 5 ตัว
เราไม่สามารถ

```
var_tuples[6] = My Var' # ไม่สามารถทำได้ จากต่างจาก List ที่สามารถกำหนด ค่าใหม่ให้กับ index เดิม ได้
```

DO NOT COPY

-Dictionary เก็บข้อมูลได้มากว่า 1 ค่าใน 1 ตัวแปรเช่นกัน หรือที่เรียกว่า table type เทียบได้กับตัวแปร array ใน php

ตัวแปลชนิดดิกชันนารี ในไพทอน ดิกชันนารีในไฟทอนเรียกได้ว่าเป็น hash table type คล้ายกับ ตัวแปล array ใน php และ hash ใน ภาษา Perl มาก คือจะมี key และ value คู่กันไปเสมอ key สามารถ เอา data type อะไรของ ไพธอนมากำหนด ก็ได้ แต่แนะนำว่าใช้ แค่ string และ numbers จะดีกว่า ตัวแปล ดิกชันนารี ถ้ากำหนดค่าให้อยู่ในเครื่องหมาย curly braces {...}" และ key จะถูกกำหนดอยู่ในเครื่องหมา square braces "[]"

ตัวอย่าง Dictionary.py

```
var_dict = {}
var_dict['one'] = "This is one"
var_dict[2] = "This is two"

var_tinydict = {'one': 'This is one ของ var_tinydict', 2:'This is two ของ var_tinydict', 'dept': 'sales'}

print (var_dict)  # แสดงค่าทั้งหมดของ var_dict
print (var_dict['one']) # แสดงเฉพาะ key "one" ของ ตัวแปล var_dict
print (var_dict[2])  # แสดงเฉพาะ key "2" ของ ตัวแปล var_dict
print ( var_dict.keys() ) # แสดง key ทั้งหมดของ ตัวแปล var_dict
print ( var_dict.values()) # แสดง values ทั้งหมดของ var_dict

print ('--------------------------')

print (var_tinydict)  # แสดงค่าทั้งหมดของ var_dict
print (var_tinydict['one']) # แสดงเฉพาะ key "one" ของ ตัวแปล var_dict
print (var_tinydict[2])  # แสดงเฉพาะ key "2" ของ ตัวแปล var_dict
print ( var_tinydict.keys() ) # แสดง key ทั้งหมดของ ตัวแปล var_dict
print ( var_tinydict.values()) # แสดง values ทั้งหมดของ var_dictผลที่ได้
```

DO NOT COPY

```
{2: 'This is two', 'one': 'This is one'}
This is one
This is two
dict_keys([2, 'one'])
dict_values(['This is two', 'This is one'])
---------------------------
{'dept': 'sales', 2: 'This is two ของ var_tinydict', 'one': 'This is one ของ var_tinydict'}
This is one ของ var_tinydict
This is two ของ var_tinydict
dict_keys(['dept', 2, 'one'])
dict_values(['sales', 'This is two ของ var_tinydict', 'This is one ของ var_tinydict'])
```

## Operator

| Arithmetic Operator | Description | Example |
|---|---|---|
| + | การบวก | 3+4=7 |
| - | การลบ | 3-4=-1 |
| * | การคูณ | 3*4=12 |
| / | การหาร | 23/4=5.75 |
| % | การหารเอาผลลัพธ์ที่เป็นเศษ | 23%4=75 |
| // | การหารเอาผลลัพธ์ที่เป็นจำนวนเต็ม | 23//4=5 |
| ** | ยกกำลัง | 3**4=81 |

DO NOT COPY

| Comparison Operator | Description | Example |
|---|---|---|
| == | เปรียบเทียบเท่ากันหรือไม่ | If (a == b) : |
| != | เปรียบเทียบไม่เท่ากันหรือไม่ | If (a != b) : |
| > | เปรียบเทียบมากกว่า | If (a > b) : |
| < | เปรียบเทียบน้อยกว่า | If (a < b) : |
| >= | เปรียบเทียบมากกว่าหรือเท่ากับ | If (a >= b) : |
| <= | เปรียบเทียบน้อยกว่าหรือเท่ากับ | If (a <= b) : |

| Logical Operator | Description | Example |
|---|---|---|
| and | เงื่อนไข และ | If (a and b) : |
| or | เงื่อนไข หรือ | If (a or b) : |
| not | เงื่อนไข ตรงข้าม | If not (a and b) : |

| Membership Operator | Description | Example |
|---|---|---|
| in | จะเป็นจริงเมื่อค่าอยู่ในตัวแปร | For x in List : |
| Not in | จะเป็นจริงเมื่อมีค่าไม่อยู่ในตัวแปร | For x not in List : |

DO NOT COPY

## Condition

```
If x > 30 :
        print('a')
Elif x > 15 :
        print(b')
Else:
        print('c')
```

ใน Python ไม่มี Switch case

## Loop

ใน Python Loop สามารถใช้ For Loop และ While

For Loop มีรูปแบบ

| รูปแบบ | ตัวอย่าง | ผลลัพธ์ |
|---|---|---|
| for ตัวแปร in ลำดับขอบเขต :<br>คำสั่ง | for x in 'word':<br>print( x ) | w<br>o<br>r<br>d |
| | for count in range(1, 5):<br>print(count) | 1<br>2<br>3<br>4 |

DO NOT COPY

While

| รูปแบบ | ตัวอย่าง | ผลลัพธ์ |
|---|---|---|
| while นิพจน์:<br>ข้อความสั่ง(s) | count = 0<br>while (count <= 4):<br>print ('Is:', count)<br>count = (count + 1) | Is: 0<br>Is: 1<br>Is: 2<br>Is: 3<br>Is: 4 |
| | while True:<br>n = input("Please enter 'hello':")<br>if n.strip() == 'hello':<br>break | Please enter 'hello': X<br>Please enter 'hello': Max<br>Please enter 'hello': hello |

## Function

การประกาศ function ใน python จะต้องขึ้นต้นด้วย def และตามด้วยชื่อ function (argument) และ return ค่าออกจาก function ซึ่งการเรียกใช้ก็เพียงอ้างอิงชื่อ function และป้อน parameter value ตามตัวอย่าง

| ตัวอย่าง | ผลลัพธ์ |
|---|---|
| def sum(*args):<br>out = 0<br>for i in args:<br>out += i<br>return out<br><br>print(sum(1,3,5,7,9)) | 25 |

DO NOT COPY

# การใช้ Python เพื่อควบคุม GPIO

## GPIO Output

ต่อ LED ที่ GPIO 4 โดย Program จะทำงานสั่งให้ LED ติดดับสลับกัน 1 วินาที

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)

while 1:
    GPIO.output(4, True)
    time.sleep(1)
    GPIO.output(4, False)
    time.sleep(1)
```

*** โหมดของ GPIO บน Raspberry Pi จะมี 2 mode คือ BCM (สำหรับอ้างอิงตาม GPIO) และ BOARD จะนับตามขา pin ที่อยู่บน board

DO NOT COPY

## GPIO Input

ต่อวงจรเพิ่มโดยขา 22 ต่อกับ switch

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)
GPIO.setup(22,GPIO.IN)

while 1:
    in_val=GPIO.input(22)
    if in_val==1 :
        GPIO.output(4,True)
    else :
       GPIO.output(4,False)
```

นอกจากนี้ใน pin ของ Raspberry Pi ยังมีส่วน Interface ในรูปแบบอื่นอีก ได้แก่ SPI (Serial Peripheral Interface) และ I2C (Inter Integrate Circuit Bus)

DO NOT COPY

# การใช้งานกล้อง USB Camera

## การหากล้อง USB

เมื่อเราเสียบกล้องแล้ว สามารถ Scan หาว่าพบกล้องหรือไม่โดยคำสั่ง

lsusb

```
pi@MaxPi:~ $ lsusb
Bus 001 Device 005: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WL
AN Adapter
Bus 001 Device 004: ID 046d:c52b Logitech, Inc. Unifying Receiver
Bus 001 Device 007: ID 1871:0316 Aveo Technology Corp.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Et
hernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

## ส่งภาพจาก WebCam ไปแสดงบน Web Browser

```
sudo apt-get install motion
```

แก้ไขไฟล์ motion.conf

```
sudo nano /etc/motion/motion.conf
```

```
# Start in daemon (background) mode and release terminal ( Default off)
daemon on

# TCP/IP port for the http server to Listen on ( Default 0 = disabled)
webcontrol_port 8080

# Restrict control Connect ions to localhost only ( Default on)
webcontrol_localhost on

# Output for http server, select off to choose raw text plain ( Default on)
webcontrol_html_output on
```

DO NOT CO

และกำหนดให้ motion ทำงานเป็น Daemon Service

```
sudo nano /etc/ Default /motion
```

```
# Set to 'yes' to enable the motion daemon
start_motion_daemon=yes
```

Start Motion Service

```
sudo service motion start
```

## บันทึกรูปจากกล้อง USB Camera

```
sudo apt-get install fswebcam
```

```
fswebcam Image.jpg
fswebcam -r 1280x720 --no-banner Image3.jpg
fswebcam -d /dev/video0 -r 640x480 test123.jpg
```

DO NOT COPY

```
pi@MaxPi:~ $ fswebcam pic1.jpg
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 384x288 to 352x288.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to 'pic1.jpg'.
pi@MaxPi:~ $ ls
Desktop    Downloads  pic1.jpg  Public        Templates
Documents  Music      Pictures  python_games  Videos
```

mkdir webcam

sudo nano webcam.sh

```
#!/bin/bash


DATE=$(date +"%Y-%m-%d_%H%M")


fswebcam -r 1280x720 --no-banner /home/pi/webcam/$DATE.jpg
```

ls -alt webcam

```
pi@MaxPi:~ $ ls -alt
total 436
drwxr-xr-x 21 pi   pi     4096 Aug  4 00:36 .
-rwxrwxrwx  1 root root    103 Aug  4 00:36 webcam.sh
```

./webcam.sh

crontab -e

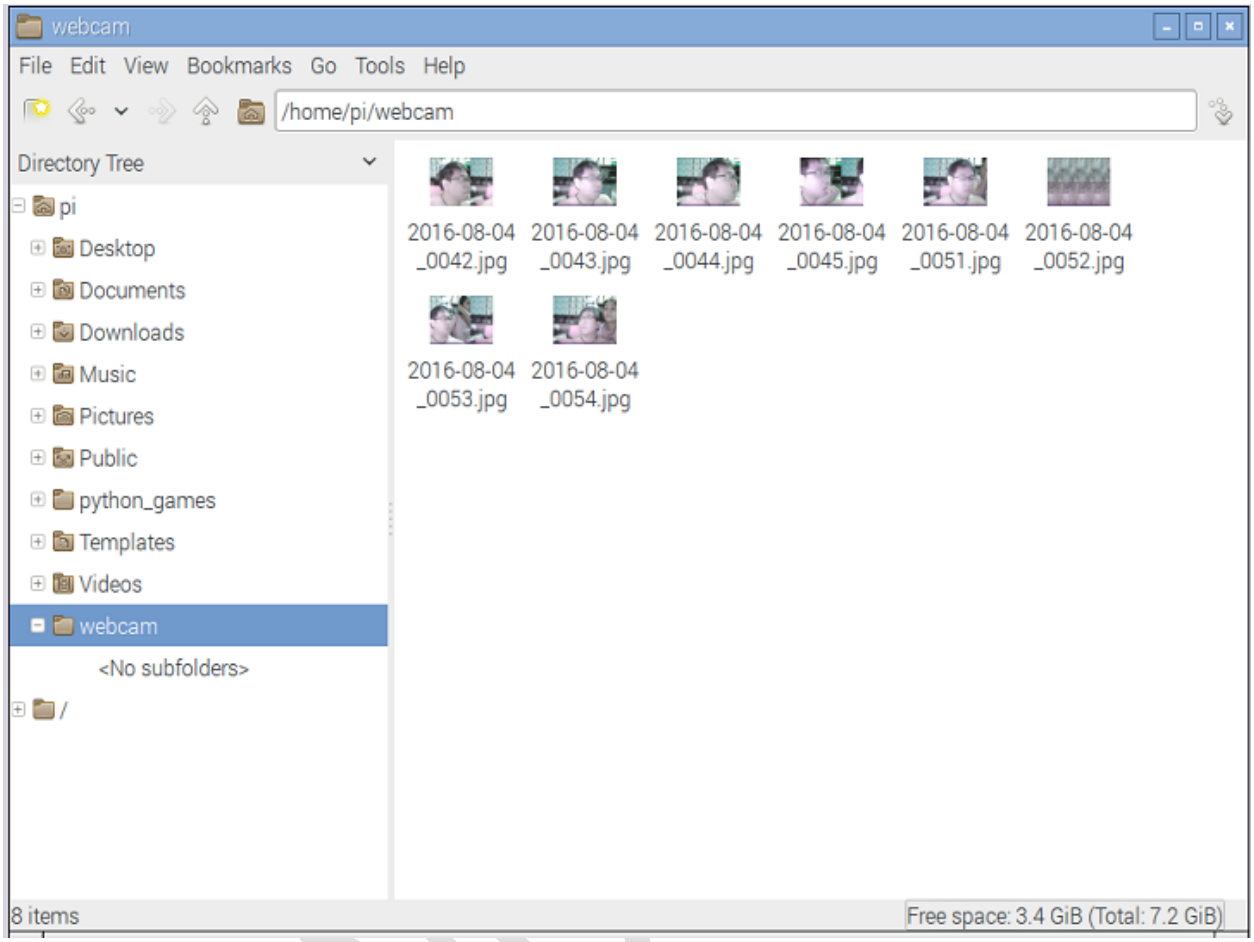
DO NOT COPY

```
pi@MaxPi:~ $ crontab -e
no crontab for pi - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        <---- easiest
  3. /usr/bin/vim.tiny

Choose 1-3 [2]:
```

```
# m h dom mon dow Command
* * * * * /home/pi/webcam.sh 2>&1
```

```
pi@MaxPi:~ $ crontab -e
no crontab for pi - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        <---- easiest
  3. /usr/bin/vim.tiny

Choose 1-3 [2]:
crontab: installing new crontab
```

DO NOT COPY


webcam
File Edit View Bookmarks Go Tools Help
/home/pi/webcam
Directory Tree
pi
Desktop
Documents
Downloads
Music
Pictures
Public
python_games
Templates
Videos
webcam
<No subfolders>
/
2016-08-04_0042.jpg
2016-08-04_0043.jpg
2016-08-04_0044.jpg
2016-08-04_0045.jpg
2016-08-04_0051.jpg
2016-08-04_0052.jpg
2016-08-04_0053.jpg
2016-08-04_0054.jpg
8 items
Free space: 3.4 GiB (Total: 7.2 GiB)

DO NOT

# ทำ Raspberry Pi เป็น MQTT Broker

Update และ Upgrade Raspberry Pi ให้เรียบร้อย

ติดตั้ง Software ที่เกี่ยวข้อง

```
sudo apt-get install libssl-dev
sudo apt-get install cmake
sudo apt-get install libc-ares-dev
sudo apt-get install uuid-dev
sudo apt-get install daemon
```

ติดตั้ง LibWebSocket

```
#Copy libwebsockets-1.4-chrome43-firefox-36.tar.gz by WinSCP
tar zxvf libwebsockets*
cd libwebsockets-1.4-chrome43-firefox-36
mkdir build
cd build
cmake ..
sudo make install
sudo ldconfig                                   #Rebuild Library Cash
cd                                              #go home
```

ติดตั้ง Mosquito

```
wget http://mosquitto.org/files/source/mosquitto-1.4.4.tar.gz
tar zxvf mosquitto-1.4.4.tar.gz
cd mosquitto-1.4.4
```

DO NOT COPY

แก้ไฟล์

| sudo nano config.mk |
| --- |
| WITH_WEBSOCKETS:=yes |

Make and Install

| make<br>sudo make install |
| --- |

Copy config

| sudo cp mosquitto.conf /etc/mosquitto |
| --- |

Edit Websocket เพิ่ม 2 บรรทัด

| sudo nano /etc/mosquitto/mosquitto.conf |
| --- |
| listener 9001<br>protocal websockets |

Add user mosquito ( Password p@ssw0rd)

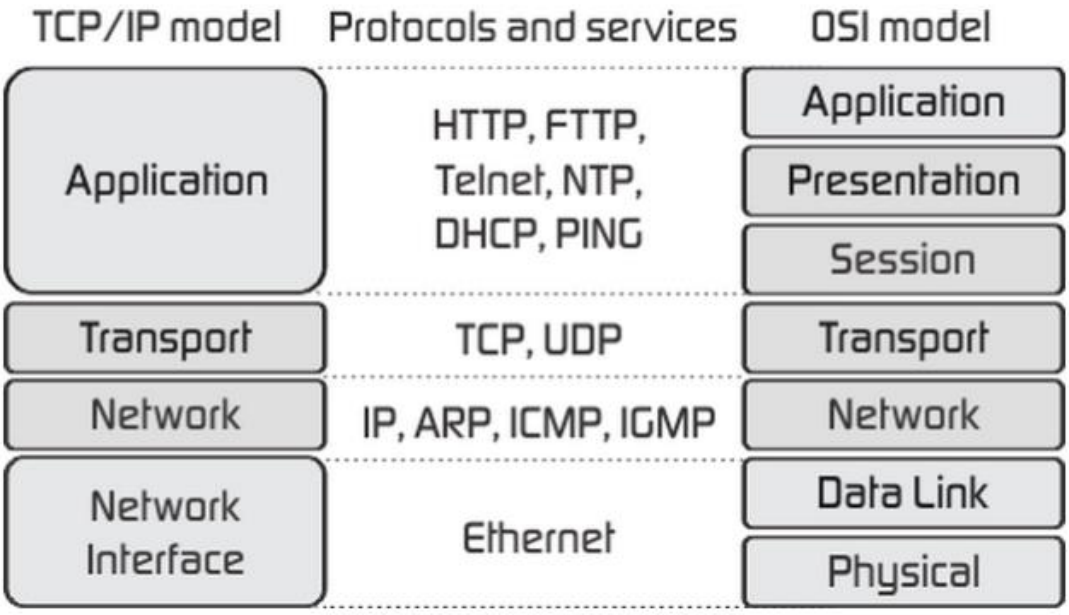
| sudo adduser mosquitto |
| --- |

```
pi@MaxPi:/etc/mosquitto $ sudo adduser mosquitto
Adding user `mosquitto' ...
Adding new group `mosquitto' (1001) ...
Adding new user `mosquitto' (1001) with group `mosquitto' ...
Creating home directory `/home/mosquitto' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for mosquitto
Enter the new value, or press ENTER for the default
        Full Name []: maximus
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
pi@MaxPi:/etc/mosquitto $ _
```

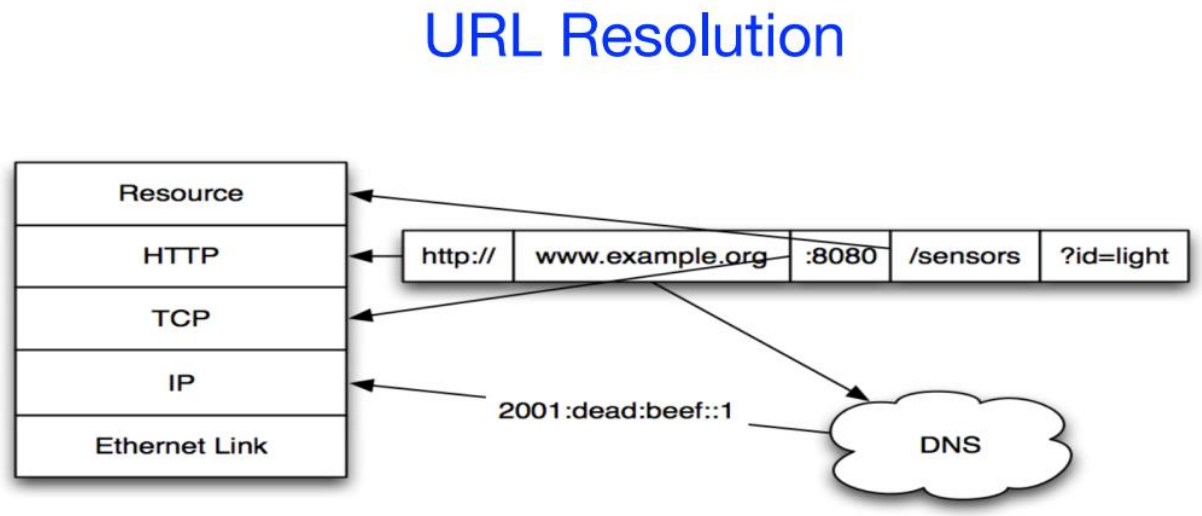
DO NOT COPY

# Socket Programming

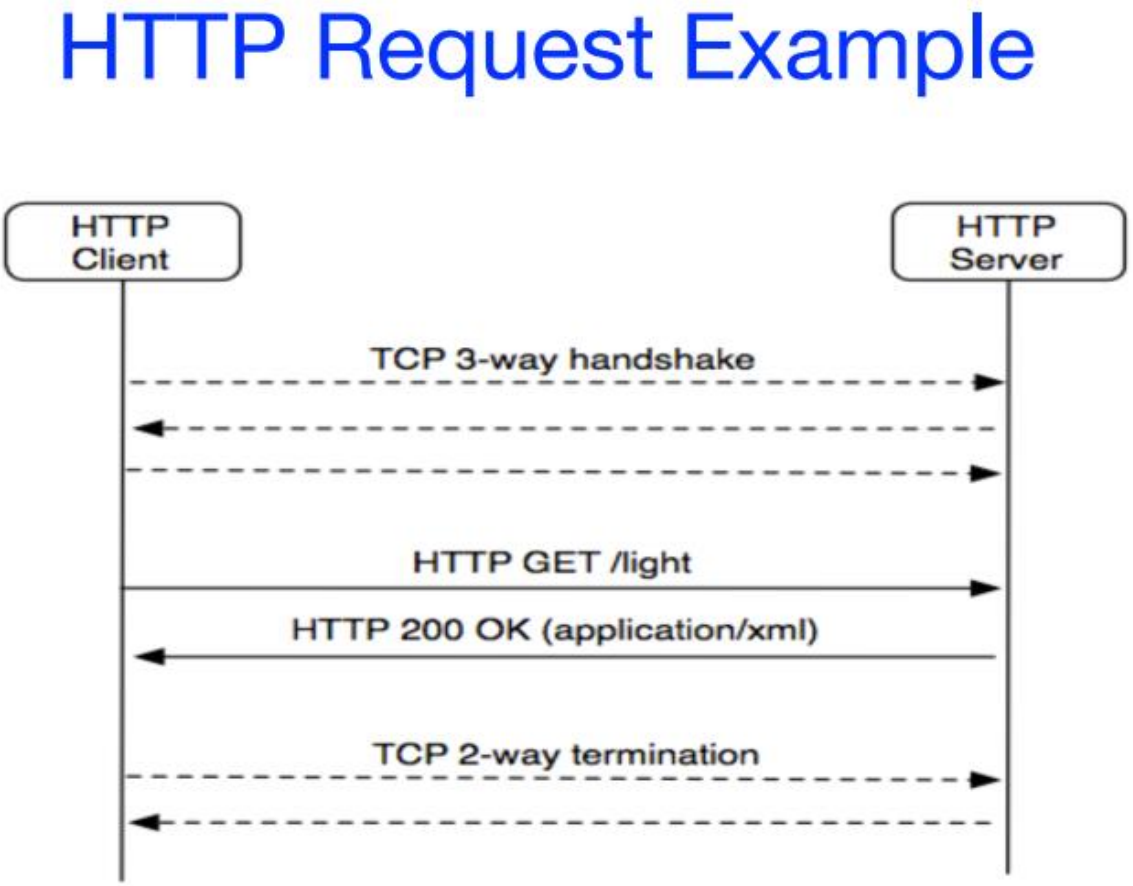
สิ่งที่เราต้องรู้คือ OSI Layer



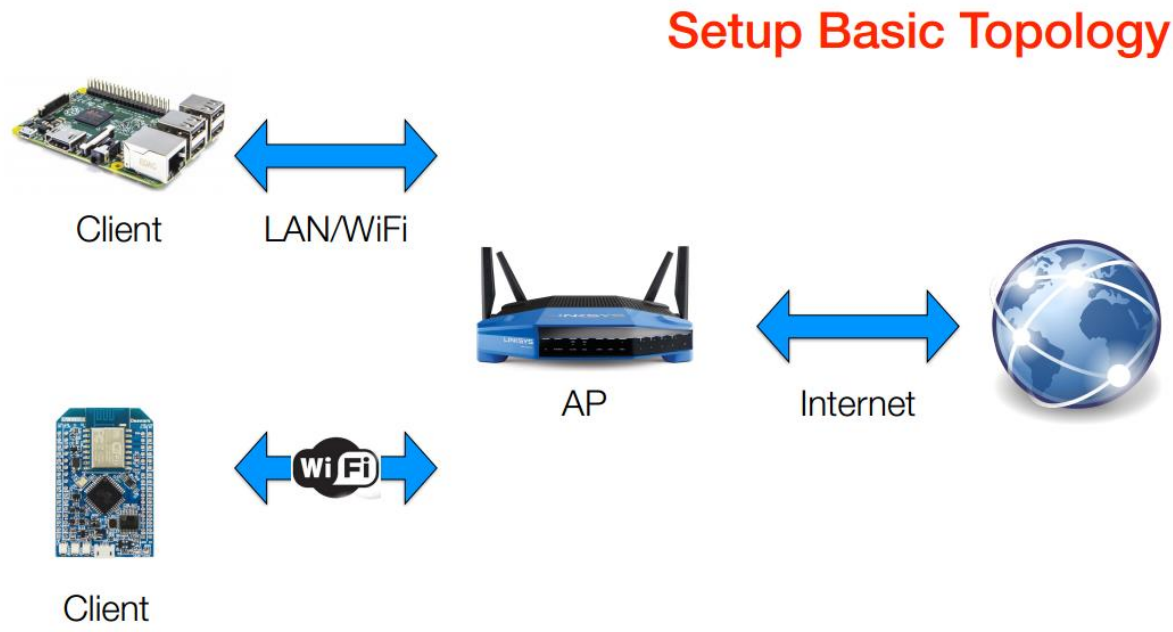

และการสื่อสารผ่าน URL



การทำ Hand Checking



Basic Topology



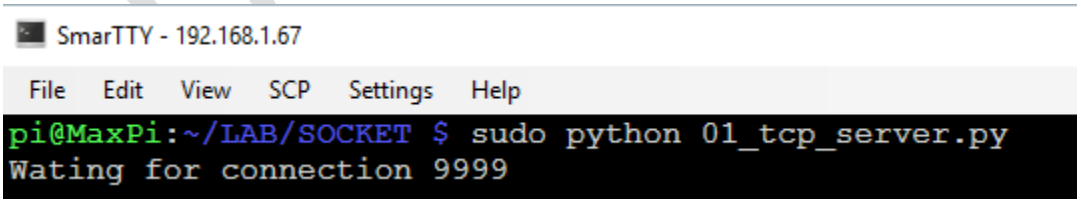

## ทดลองทำ Raspberry Pi เป็น Socket Server

Python

| 01_tcp_server.py |
| --- |
| from socket import *<br>import thread<br><br>BUFF = 1024<br>HOST = ''<br>PORT = 9999<br><br>def handler(clientsock,addr): |

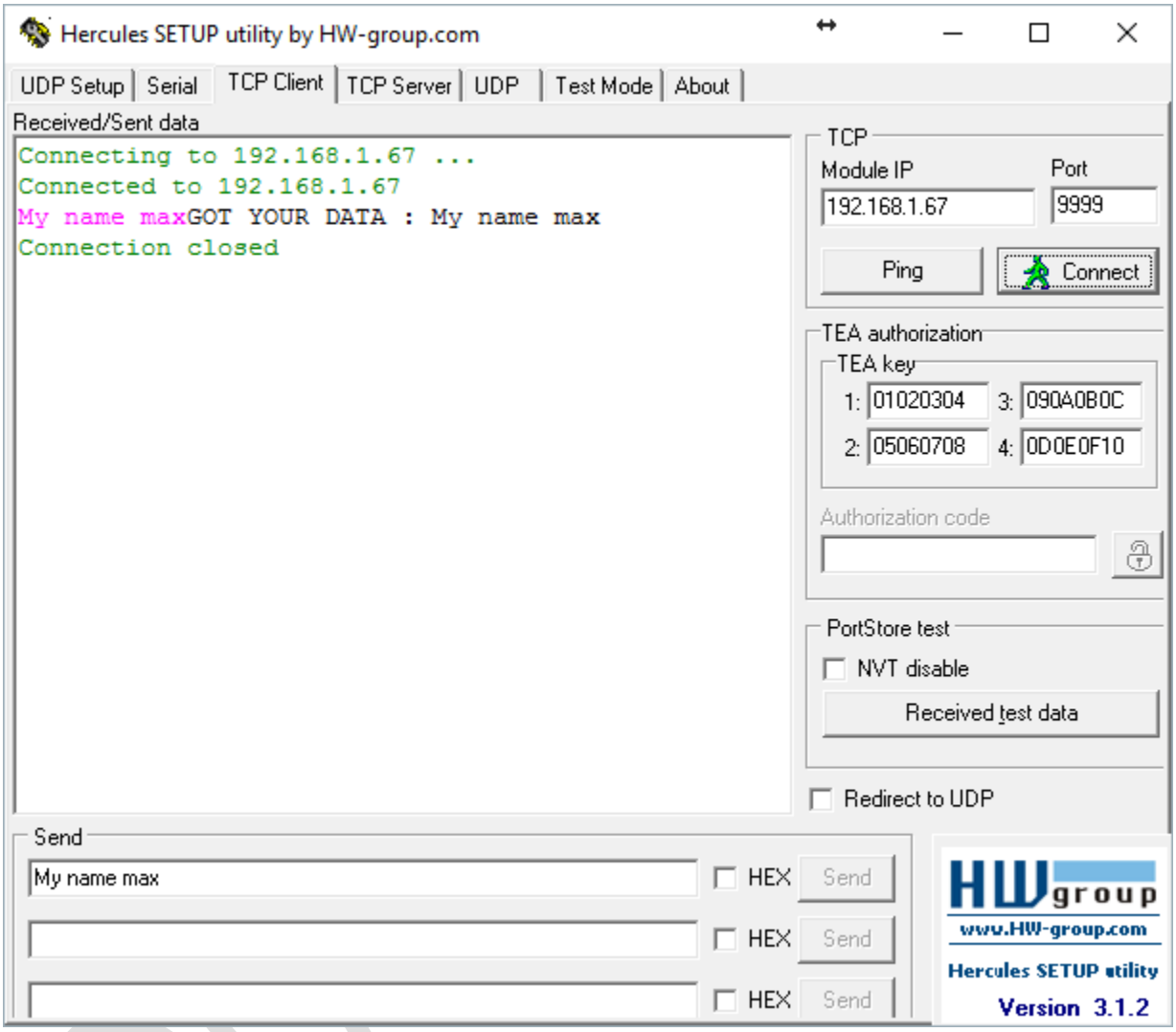
DO NOT

```
while 1:
data = clientsock.recv(BUFF)
if not data: break
print "Receive data + data + "from" + str(addr)
clientsock.send("GOT YOUR DATA " + data)
if "close" == data.rstrip(): break

clientsock.close()
print addr, "closed Connect ion" #log on Console

if __name__=='__main__':
ADDR = (HOST, PORT)
serversock = socket(AF_INET, SOCK_STREAM)
serversock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
serversock.bind(ADDR)
serversock.listen(5)
while 1:
print 'Wating for Connect ion', PORT
clientsock, addr = serversock.accept()
print Connect ed from:', addr
thread.start_new_thread(handler, (clientsock, addr))
```



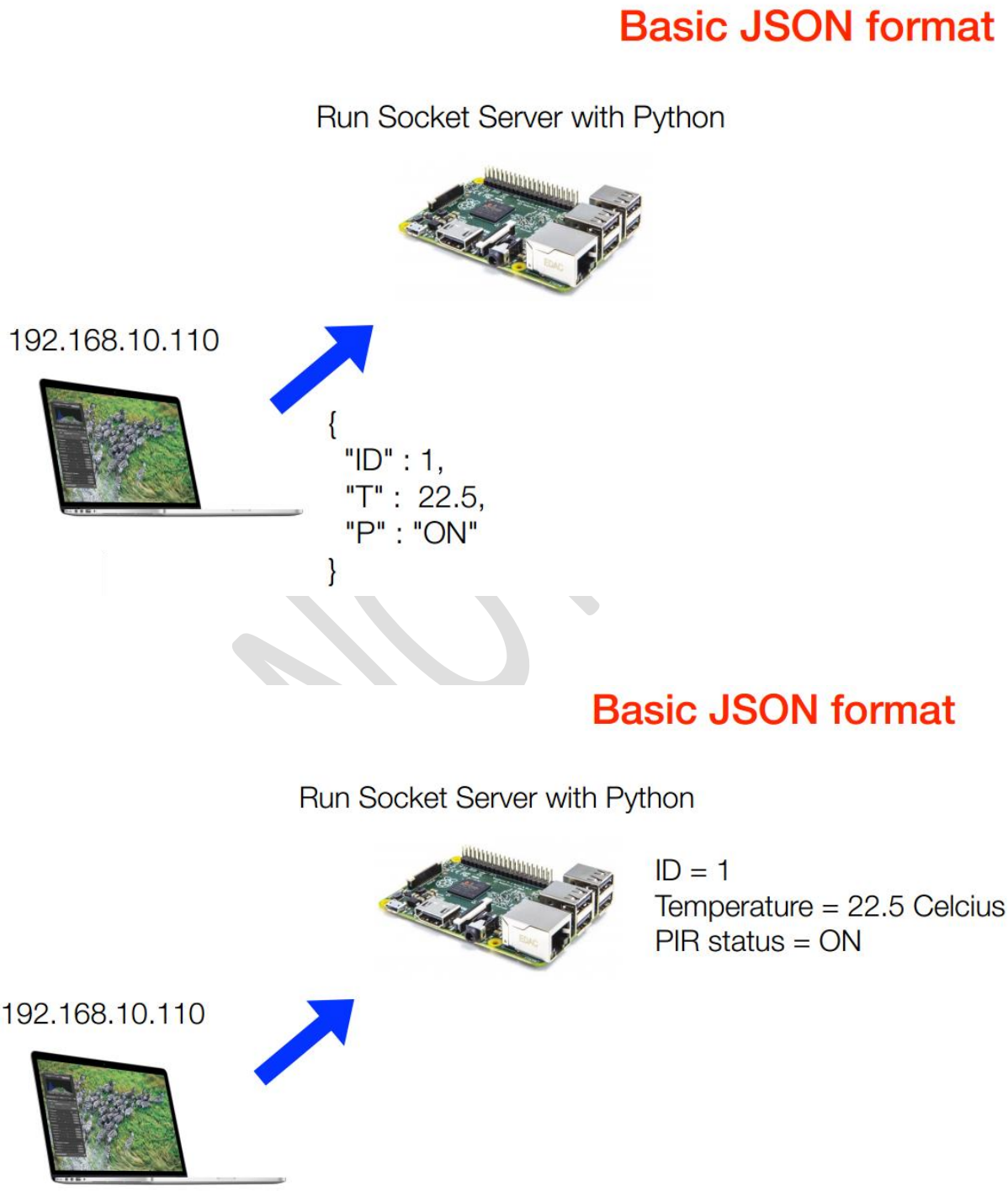

DO NOT COPY

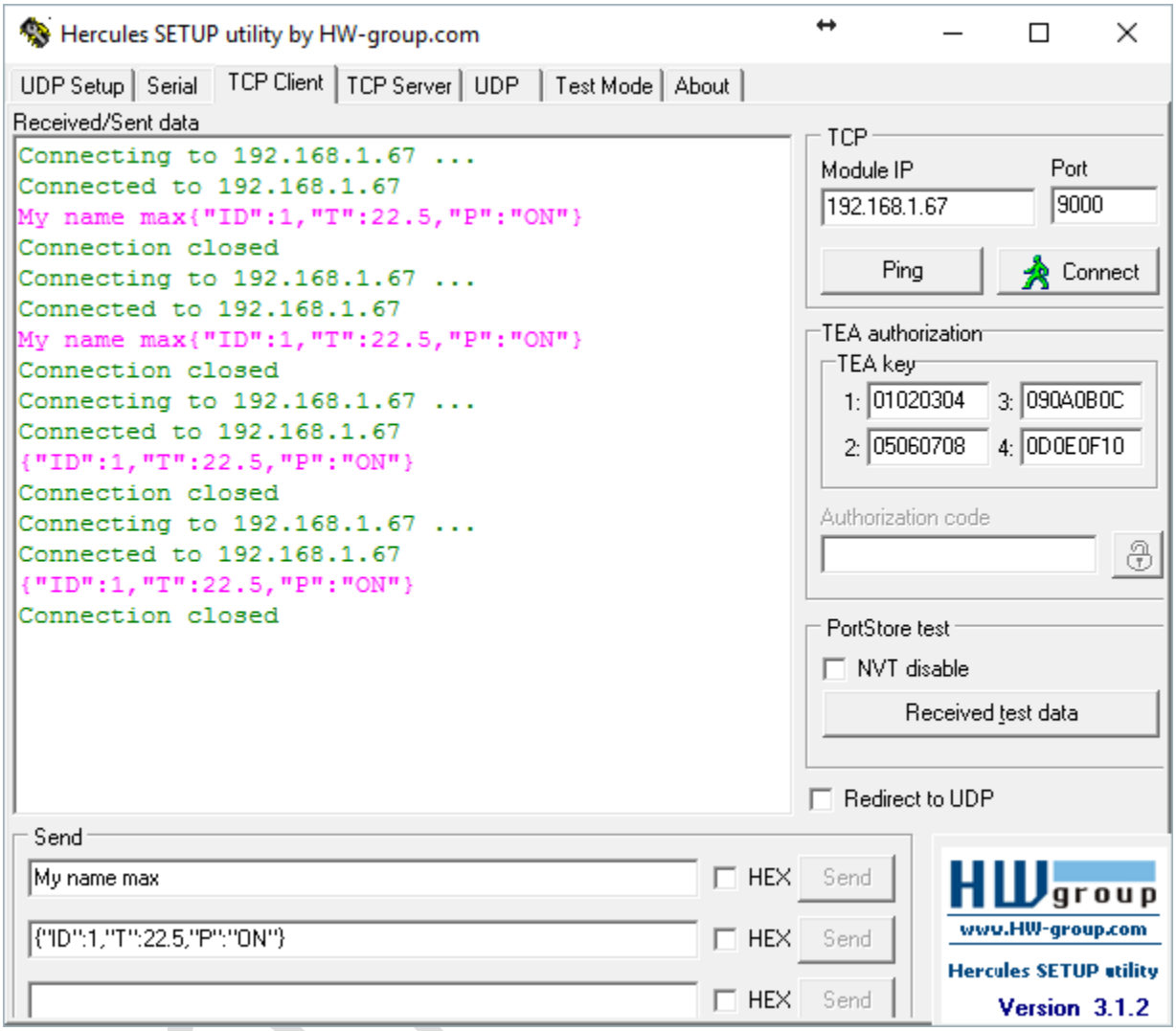
ใช้งานโดย เปิด Heculess




```
pi@MaxPi:~/LAB/SOCKET $ sudo python 01_tcp_server.py
Wating for connection 9999
...connected from: ('192.168.1.20', 53279)
Wating for connection 9999
Receive data :My name maxfrom('192.168.1.20', 53279)
('192.168.1.20', 53279) closed connection
```

## ทดลองทำ Raspberry Pi เป็น Socket Server รับค่า JSON




Hercules SETUP utility by HW-group.com
UDP Setup
Serial
TCP Client
TCP Server
UDP
Test Mode
About
Received/Sent data
Connecting to 192.168.1.67 ...
Connected to 192.168.1.67
My name max{"ID":1,"T":22.5,"P":"ON"}
Connection closed
Connecting to 192.168.1.67 ...
Connected to 192.168.1.67
My name max{"ID":1,"T":22.5,"P":"ON"}
Connection closed
Connecting to 192.168.1.67 ...
Connected to 192.168.1.67
{"ID":1,"T":22.5,"P":"ON"}
Connection closed
Connecting to 192.168.1.67 ...
Connected to 192.168.1.67
{"ID":1,"T":22.5,"P":"ON"}
Connection closed
TCP
Module IP
Port
192.168.1.67
9000
Ping
Connect
TEA authorization
TEA key
1: 01020304
3: 090A0B0C
2: 05060708
4: 0D0E0F10
Authorization code
PortStore test
NVT disable
Received test data
Redirect to UDP
Send
My name max
HEX
Send
{"ID":1,"T":22.5,"P":"ON"}
HEX
Send
HEX
Send
HW group
www.HW-group.com
Hercules SETUP utility
Version 3.1.2

## ทดลองส่งข่อมูลผ่าน JSON และบันทึกลง Database SQLite

**SQLite Database**

- ในหลาย ๆ ระบบที่ใช้ Embedded System เป็นส่วนประกอบต้องการดูค่าและสามารถตรวจสอบข้อมูลย้อนหลังได้

- การเก็บข้อมูลต้องสามารถดูย้อนหลังโดยใชวันที่ เดือน หรือปี เป็นตัวกำหนดในการเลือกดูได้

- ในกรณที่ข้อมูลมีมหาศาลตองมีกลไกในการเลือกแสดงข้อมูล

**ทำไมเราเลือก SQLite**

- เล็ก เร็ว ดูแลง่าย ด้วยซอร์สโค๊ดที่บรรจุอยู่ในไฟลเดียว ( แต่เป็นแสนบรรทัด )

- นิยมใช้ในอุปกรณ์ขนาดเล็กเช่น Smart phone, Embedded Linux

- มีระบบการทำงานเช่น in-memory database

- ซัพพอร์ตชุดคำสัง SQL

- Flat File Friendly

ไฟล์ที่ใช้ในการติดตัง

```
sqlite3_3.7.13-1+deb7u1_armhf.deb
```

คำสั่งที่ใช้ในการติดตั้ง

```
sudo dpkg -i sqlite3_3.7.13-1+deb7u1_armhf.deb
```

ถ้ามี Error จากการติดตั้งให้ลอง

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -f install
```

DO NOT COPY

ทดสอบเข้า SQL Lite ด้วย

Sqlite3 test.db

```
pi@MaxPi:~ $ sqlite3 test.db
SQLite version 3.8.7.1 2014-10-29 13:59:56
Enter ".help" for usage hints.
sqlite> _
```

สร้าง Table

```
CREATE TABLE TEMP_T(NUM INTEGER PRIMARY KEY AUTOINCREMENT, ID INTEGER,TEM FLOAT,
PIR TEXT, TIME TEXT );
```

| Integer | Integer | FLOAT | TEXT | TEXT |
|---|---|---|---|---|
| NUM | ID | TEM | PIR | TIME |
| 1 | 1 | 23.45 | ON | 12:45 |
| 2 | 2 | 24.45 | OFF | 12:50 |
| 3 | 1 | 23.45 | OFF | 12:55 |
| 4 | 3 | 23.45 | ON | 13:00 |

Insert Data

```
insert into TEMP_T VALUES(NULL, 01, 23.12, "ON", "12:45:00");
insert into TEMP_T VALUES(NULL, 02, 24.45, "OFF", "12:50:00");
insert into TEMP_T VALUES(NULL, 01, 23.45, "OFF", "12:55:00");
insert into TEMP_T VALUES(NULL, 03, 23.45, "ON", "13:00:00");
```

Select Data

Select * from TEMP_T;

```
pi@MaxPi:~ $ sqlite3 test.db
SQLite version 3.8.7.1 2014-10-29 13:59:56
Enter ".help" for usage hints.
sqlite> CREATE TABLE TEMP_T(NUM INTEGER PRIMARY KEY AUTOINCREMENT, ID INTEGER,TEM FLOAT, PIR TEXT, TIME TEXT );
sqlite> insert into TEMP_T VALUES(NULL, 01, 23.12, "ON", "12:45:00");
sqlite> insert into TEMP_T VALUES(NULL, 02, 24.45, "OFF", "12:50:00");
sqlite> insert into TEMP_T VALUES(NULL, 01, 23.45, "OFF", "12:55:00");
sqlite> insert into TEMP_T VALUES(NULL, 03, 23.45, "ON", "13:00:00");
sqlite> Select * from TEMP_T;
1|1|23.12|ON|12:45:00
2|2|24.45|OFF|12:50:00
3|1|23.45|OFF|12:55:00
4|3|23.45|ON|13:00:00
sqlite>
```

**Insert Data ด้วย Python**

insert_SQLite.py

```
import sqlite3 as lite
import sys
con = lite. Connect ('test.db')
with con:
 cur = con.cursor()
 cur.execute("insert into TEMP_T VALUES(NULL, 01, 23.12, 'ON', '12:00:00');")
if con:
 con.close()
```

```
pi@MaxPi:~ $ sqlite3 test.db
SQLite version 3.8.7.1 2014-10-29 13:59:56
Enter ".help" for usage hints.
sqlite> select * from temp_t;
1|1|23.12|ON|12:45:00
2|2|24.45|OFF|12:50:00
3|1|23.45|OFF|12:55:00
4|3|23.45|ON|13:00:00
5|1|23.12|ON|12:00:00
sqlite> _
```

DO NOT COPY

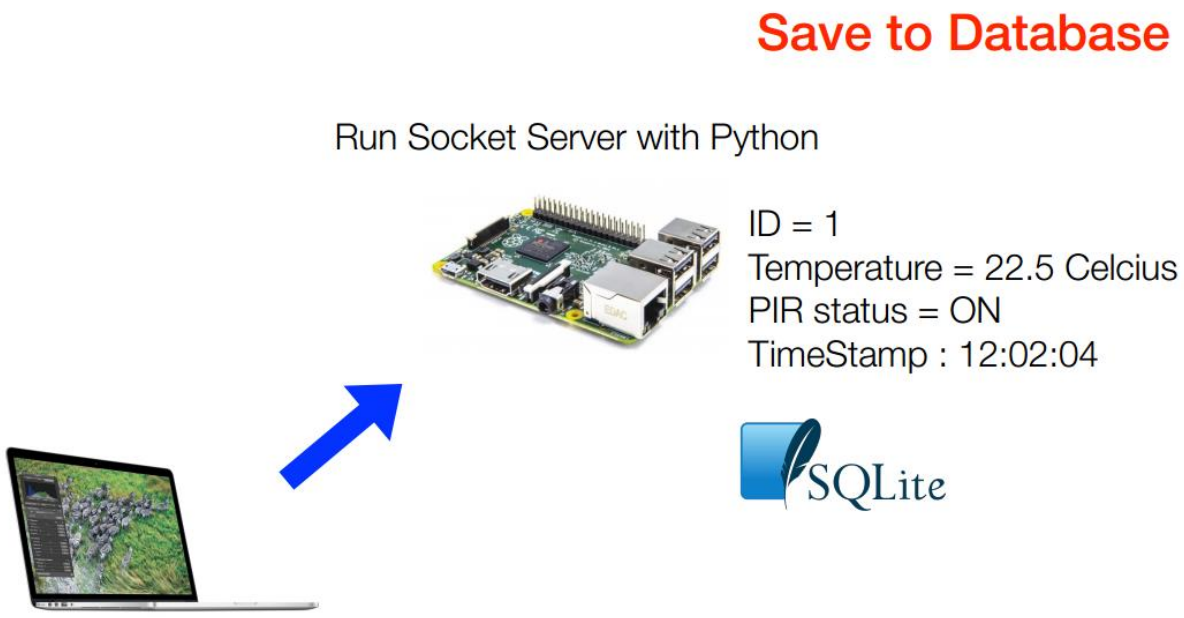
## Select Data ด้วย Python

select_SQLite.py

```
import sqlite3 as lite
import sys
con = lite. Connect ('test.db')
with con:
 cur = con.cursor()
 cur.execute("SELECT * FROM TEMP_T")
 rows = cur.fetchall()
 for row in rows:
 print row
```

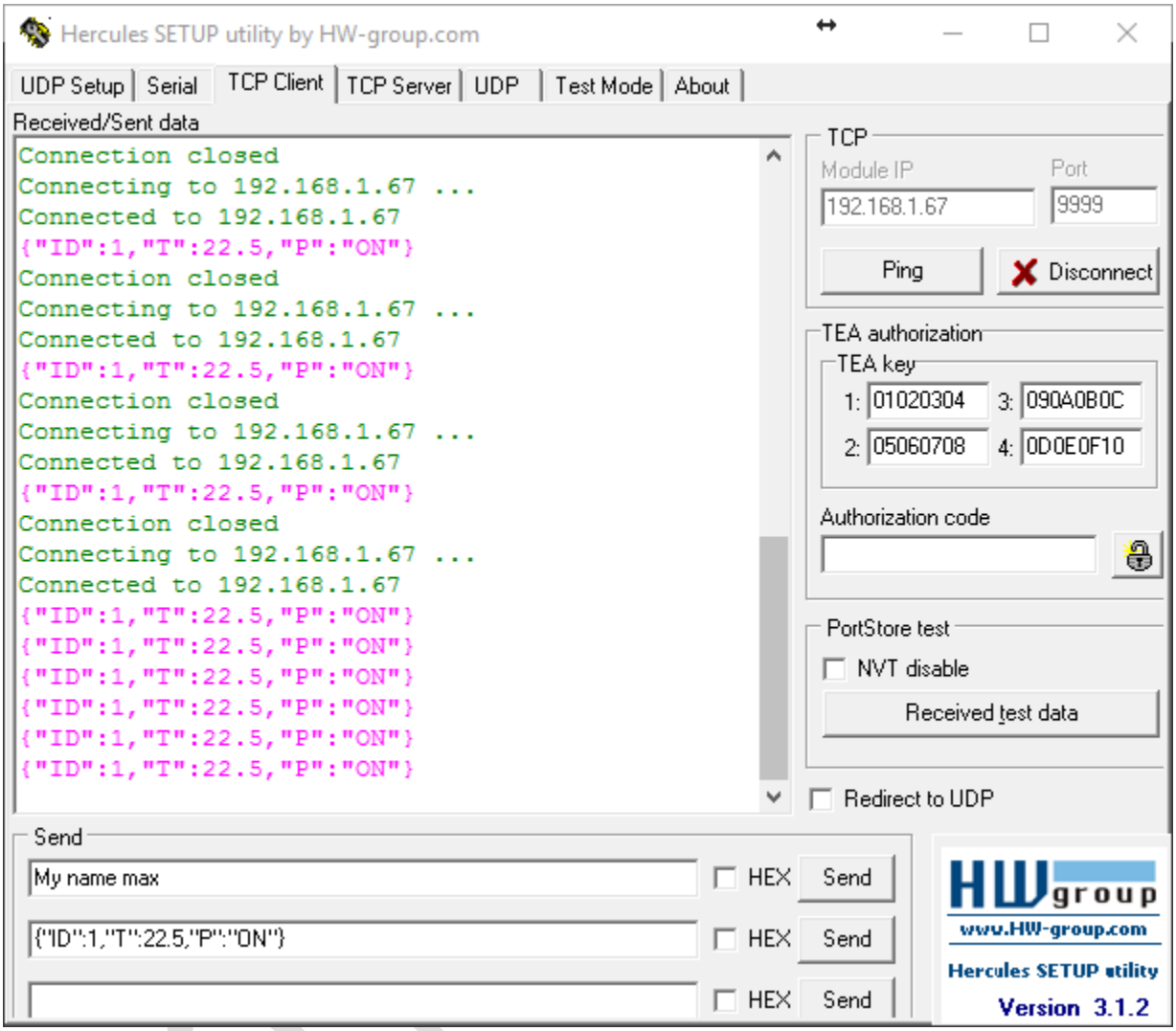
```
pi@MaxPi:~ $ sudo nano select_SQLite.py
pi@MaxPi:~ $ sudo python select_SQLite.py
(1, 1, 23.12, u'ON', u'12:45:00')
(2, 2, 24.45, u'OFF', u'12:50:00')
(3, 1, 23.45, u'OFF', u'12:55:00')
(4, 3, 23.45, u'ON', u'13:00:00')
(5, 1, 23.12, u'ON', u'12:00:00')
pi@MaxPi:~ $
```

DO NOT COPY




File 03_tcp_server_json_db.py

```
pi@MaxPi:~ $ sudo python 03_tcp_server_json_db.py
Wating for connection 9999
...connected from: ('192.168.1.20', 57599)
Wating for connection 9999
Receive data :{"ID":1,"T":22.5,"P":"ON"}from('192.168.1.20', 57599)
JSON DECODE
JSON DECODE and SAVE TO DATABASE SUCCESS
Receive data :{"ID":1,"T":22.5,"P":"ON"}from('192.168.1.20', 57599)
JSON DECODE
JSON DECODE and SAVE TO DATABASE SUCCESS
Receive data :{"ID":1,"T":22.5,"P":"ON"}from('192.168.1.20', 57599)
JSON DECODE
JSON DECODE and SAVE TO DATABASE SUCCESS
Receive data :{"ID":1,"T":22.5,"P":"ON"}from('192.168.1.20', 57599)
JSON DECODE
JSON DECODE and SAVE TO DATABASE SUCCESS
Receive data :{"ID":1,"T":22.5,"P":"ON"}from('192.168.1.20', 57599)
JSON DECODE
JSON DECODE and SAVE TO DATABASE SUCCESS
```




```
sqlite> select * from temp_t;
1|1|23.12|ON|12:45:00
2|2|24.45|OFF|12:50:00
3|1|23.45|OFF|12:55:00
4|3|23.45|ON|13:00:00
5|1|23.12|ON|12:00:00
6|1|23.12|ON|12:00:00
7|1|23.12|ON|12:00:00
8|1|22.5|ON|20:22:41
9|1|22.5|ON|20:23:05
10|1|22.5|ON|20:23:05
11|1|22.5|ON|20:23:05
12|1|22.5|ON|20:23:06
13|1|22.5|ON|20:23:06
```

*** ต้อง Run Script ใน Parth ที่มี File test.db อยู่

**การใช้ MQTTSpy**

DO NOT COPY

# ทดลองใช้งาน Blynk

DO NOT COPY

# QT programming

DO NOT COPY

# Reference

http://mosquitto.org/

http://mqtt.org/

DO NOT COPY

Project TieText

sudo apt-get -y install fswebcam

sudo apt-get -y install libav-tools

apt-get install streamer

sudo streamer -c /dev/video0 -o /home/pi/camera/outputfile.jpeg

streamer -t 0:0:2 -c /dev/video0 -f rgb24 -r 3 -o /home/pi/camera/outputfile.avi

**Install FFmpeg**

cd /usr/src

sudo git clone git://git.videolan.org/x264

cd x264

sudo ./configure --host=arm-unknown-linux-gnueabi --enable-static --disable-opencl

sudo make

sudo make install

cd /usr/src

sudo git clone https://github.com/FFmpeg/FFmpeg.git

cd ffmpeg

sudo ./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfree

DO NOT COPY

make

sudo make install

```
ffmpeg -f video4linux2 -r 25 -s 640x480 -i /dev/video0 out.avi
```

```
ffmpeg -f v4l2 -r 25 -s 640x480 -i /dev/video0 out.avi
```

DO NOT COPY

## Open CV

http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_video_display/py_video_display.html

sudo apt-get install python-opencv

Python Capture VDO

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Display the resulting frame
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

Python Saving VDO

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

# Define the codec and create VideoWriter object
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))

while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
        frame = cv2.flip(frame,0)
```

```
        # write the flipped frame
        out.write(frame)

        cv2.imshow('frame',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

# Release everything if job is finished
cap.release()
out.release()
cv2.destroyAllWindows()
```

Python Play VDO from File

```
import numpy as np
import cv2

cap = cv2.VideoCapture('vtest.avi')

while(cap.isOpened()):
    ret, frame = cap.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

http://www.pygame.org/docs/tut/camera/CameraIntro.html

QT Wiring Pi

WiringPi คือ กลุ่มของไลบรารี่ที่พัฒนาโดย http://wiringpi.com สำหรับเข้าถึงการใช้งาน GPIO ของชิพ BCM2835 ที่เป็นชิพประมวลผลหลักของบอร์ด Raspberry Pi ซึ่งสามารถใช้งานได้จากภาษา C/C++ และภาษาอื่นๆ อีกหลากหลายภาษาที่ WiringPi ออกแบบชุดคำสั่งให้มีลักษณะคล้ายกับการใช้งานภาษา Wiring ทำให้ผู้ที่เคยใช้งานบอร์ด Arduino รู้สึกคุ้นเคยกับรูปแบบชุดคำสั่งและทำให้การควบคุมฮาร์ดแวร์ของบอร์ด Raspberry Pi สามารถทำได้ง่ายขึ้น

วิธีติดตั้ง WiringPi ของบอร์ด Raspberry Pi

- เชื่อมต่อบอร์ด Raspberry Pi กับ Internet

- เปิดโปรแกรม LXTerminal

- ในขั้นตอนการใช้คำสั่ง update และ upgrade จะใช้เวลาในการค่อนข้างนาน หากเคย update และ upgrade ไปแล้ว อาจทดลองใช้คำสั่ง git clone git://git.drogon.net/wiringPi ในหัวข้อถัดไปดูก่อน แล้วหากไม่ Error ก็สามารถข้ามขั้นตอนนี้ไปได้

    o พิมพ์คำสั่ง sudo apt-get update แล้วรอจนอัพเดทเสร็จ

    o พิมพ์คำสั่ง sudo apt-get upgrade แล้วรอจนอัพเกรดเสร็จ

- พิมพ์คำสั่ง sudo apt-get install git-core เพื่อติดตั้ง Git Core

- พิมพ์คำสั่ง sudo apt-get install libi2c-dev เพื่อติดตั้งไลบรารี่ของ I2C

- พิมพ์คำสั่ง git clone git://git.drogon.net/wiringPi เพื่อสร้างลิงค์ WiringPi จาก Git

- พิมพ์คำสั่ง cd wiringPi กด Enter

DO NOT COPY

- พิมพ์คำสั่ง git pull origin เพื่อดึงซอร์สโค้ดมาจากต้นทาง

- พิมพ์คำสั่ง ./build เพื่อคอมไพล์ WiringPi

- ทดสอบการติดตั้ง WiringPi พิมพ์คำสั่ง gpio –v จะปรากฏข้อความดังภาพ

```
pi@MDRS01:~/wiringPi $ gpio -v
gpio version: 2.32
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Pi 2, Revision: 01, Memory: 1024MB, Maker: Sony
  * Device tree is enabled.
  * This Raspberry Pi supports user-level GPIO access.
    -> See the man-page for more details
    -> ie. export WIRINGPI_GPIOMEM=1
```

- พิมพ์คำสั่ง gpio readall จะปรากฏตาราง GPIO ดังภาพ

```
pi@MDRS01:~/wiringPi $ gpio readall
 +-----+-----+---------+------+---+---Pi 2---+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 |   IN | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 |   IN | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT0 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 1 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI |   IN | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO |   IN | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK |   IN | 0 | 23 || 24 | 1 | IN   | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | IN   | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 2---+---+------+---------+-----+-----+
```

COPY

QT Camera ใช้ ต้องใช้ Version 5.3 ขึ้นไป และต้องลง QtMultimedia ด้วย

- Check Version

$qmake –version

```
pi@MaxMetal:~ $ qmake --version
QMake version 3.0
Using Qt version 5.3.2 in /usr/lib/arm-linux-gnueabihf
pi@MaxMetal:~ $
```

- Install QTMultimedia

$sudo apt-get install qtmultimedia5*

$sudo apt-get install qtdeclarative5-dev

$sudo apt-get install qtbase5-dev

sudo apt-get install qt5-default libqt5multimedia5 libqt5multimedia5-plugins \

libqt5multimediaquick-p5 libqt5multimediawidgets5 \

qtmultimedia5-dev qtmultimedia5-examples

sudo apt-get install gstreamer0.10-plugins-*

apt-get install python-wxgtk3

Python GUI Kivy

```
sudo apt-get update
sudo apt-get install libsdl2-dev libsdl2-image-dev libsdl2-mixer-dev
libsdl2-ttf-dev \
```

DO NOT COPY

```
pkg-config libgl1-mesa-dev libgles2-mesa-dev \
python-setuptools libgstreamer1.0-dev git-core \
gstreamer1.0-plugins-{bad,base,good,ugly} \
gstreamer1.0-{omx,alsa} python-dev cython
```

เอารูปรวมเป็น VDO

```
sudo apt-get install mencoder
```

ถ้าติดตั้งไม่ได้ใช้วิธีนี้

```
wget http://www.mplayerhq.hu/MPlayer/releases/mplayer-export-snapshot.tar.bz2
bunzip2 *
tar xvf mplayer-export-snapshort.tar
cd mplayer-export-2016-03-26/
sudo apt-get install ffmpeg        ---→ ถ้าไม่สามารถลงได้ให้ไปทำ configure เลย มันจะ download ffmpeg มาให้
./configure
make
< ... wait a long time here ...>
./mencoder
sudo make install
```

```
ls *.jpg > stills.txt
```

```
mencoder -nosound -ovc lavc -lavcopts
vcodec=mpeg4:aspect=16/9:vbitrate=8000000 -vf scale=1920:1080 -o
timelapse.avi -mf type=jpeg:fps=24 mf://@stills.txt
```

DO NOT COPY

DO NOT COPY

**Mount Windows Share Folder in Linux**

$sudo mount -t cifs //192.168.87.248/GreigeInspection /mnt -o user=MDRS,password=159357

$sudo umount /mnt

sudo mount -t cifs //192.168.84.38/GreigeInspection /mnt -o user=MDRS,password=159357

sudo mount -t cifs -o username=administrator,password=p@ssw0rd,domain=192.168.84.3 //192.168.84.6/machine /mnt

**ทำ Auto mount**

วิธีที่ 1 ที่ไฟล์ /etc/fstab

//192.168.84.38/GreigeInspection /mnt cifs defaults,rw,username=administrator,password=Asdf1234,domain=192.168.84.3,uid=pi,gid=pi,x-systemd.automount 0 0

*****ถ้า boot มาแล้วมันไม่ mount ต้องไปเพิ่มใน /etc/rc.local

sudo mount -av

****** ถ้าไม่ได้ ใช้วิธีที่ 2

วิธีที่ 2 ใน File /etc/rc.local ******** วิธีนี้ OK

สร้าง script mountshare.sh มี delay นิดนึง

```
#! /bin/sh -e
sleep 50
sudo mount -t cifs -o username=administrator,password=p@ssw0rd,domain=192.168.84.3
//192.168.84.9/machine /mnt
exit 0
```

```
sudo mount -t cifs -o username=administrator,password=Asdf1234,domain=192.168.84.3
//192.168.84.38/GreigeInspection /mnt

#!/bin/sh -e

sudo /home/pi/mountshare.sh &
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
```

DO NOT COPY

```
# value on error.
...
```

Refresh mount table

sudo mount -av

******* ถ้าไม่ได้ ใช้วิธีสุดท้าย

วิธีที่ 3 ใช้ crontab

สร้าง script mountshare.sh มี delay นิดนึง

```
#! /bin/sh -e
sleep 50
sudo mount -t cifs -o username=administrator,password=p@ssw0rd,domain=192.168.84.3
//192.168.84.6/machine /mnt
```

crontab -e

@reboot /home/pi/mountshare.sh &

```
  GNU nano 2.2.6                File: /tmp/crontab.E6785R/crontab

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

การทำ Script ให้เป็น service start|stop ได้ใน init.d

**Create script in /etc/init.d**

```
sudo nano /etc/init.d/mountshare.sh
```

mountshare.sh

```
#! /bin/sh
# /etc/init.d/mountshare

### BEGIN INIT INFO
# Provides:          mountshare
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Simple script to start a program at boot
# Description:       A simple script from www.stuffaboutcode.comwhich
will start / stop a program a boot / shutdown.
### END INIT INFO

# If you want a command to always run, put it here

# Carry out specific functions when asked to by the system
case "$1" in
  start)
    echo "Starting mountshare"
    # run application you want to start
   sudo mount -t cifs -o
username=administrator,password=p@ssw0rd,domain=192.168.84.3
//192.168.84.6/machine /mnt
    ;;
  stop)
    echo "Stopping mountshare"
    # kill application you want to stop
    sudo umount /mnt
    ;;
  *)
    echo "Usage: /etc/init.d/mountshare {start|stop}"
    exit 1
    ;;
```

DRAFT COPY

```
esac

exit 0
```

**Make script executable**

```
sudo chmod 755 /etc/init.d/mountshare.sh
```

**Test starting the program**

```
sudo /etc/init.d/mountshare.sh start
```

**Test stopping the program**

```
sudo /etc/init.d/mountshare.sh stop
```

**Register script to be run at start-up**

To register your script to be run at start-up and shutdown, run the following command:

```
sudo update-rc.d mountshare.sh defaults
```

f you ever want to remove the script from start-up, run the following command:

```
sudo update-rc.d -f  mountshare.sh remove
```

**ทำ Automatic Run Python in X-Windows**

**วิธีที่ 1**

ที่ File: /etc/xdg/lxsession/LXDE-pi/autostart

sudo nano /etc/xdg/lxsession/LXDE-pi/autostart

เพิ่ม @/usr/bin/python /home/pi/camera/finalMDRS.py **** ต้องอยู่ก่อน @xscreensaver

```
@lxpanel --profile LXDE
@pcmanfm --desktop --profile LXDE
@lxterminal
@leafpad
@xscreensaver -no-splash@
```

**<u>วิธีที่ 2 ****** ใช้ได้</u>**

ที่ File: /home/pi/.config/lxsession/LXDE-pi/autostart

```
@lxpanel --profile LXDE
@pcmanfm --desktop --profile LXDE
@/usr/bin/python /home/pi/MDRS/finalMDRS.py
@xscreensaver -no-splash@
```

หรือ

@lxterminal --command “/home/pi/MDRS/RunFinalMDRS.sh”

Df

DO NOT COPY

**Tkinter Graphic**

ต้องติดตั้ง

sudo apt-get install python-tk

sudo apt-get install python-imaging-tk

sudo apt-get install idle python-pmw python-imaging

DO NOT COPY

## Setting up I2C

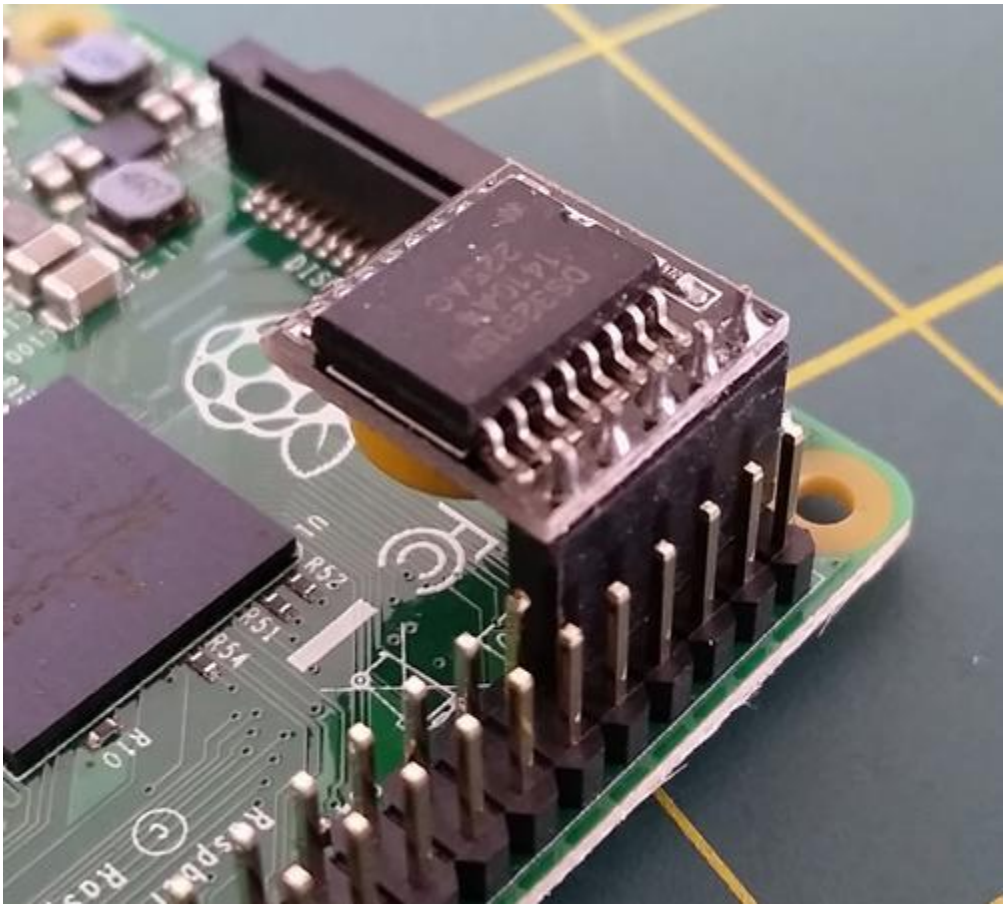
The RTC board uses what is known as the I2C (pronounced 'eye squared see') protocol to communicate between itself and the Pi. The I2C allows the Pi to communicate with a lot of different devices at the same time that are also connected to the same I2C pins on the Pi.

Simon Monk has written very good instructions about how to set up your Pi to be able to use i2c on the Adafruit site. Follow the full instructions, including checking that everything has worked by going through the manual instructions too.

## Plugging in the Board

It is always best to turn your Pi off before plugging anything onto the GPIO pins. Always do this using the `sudo halt` or `sudo shutdown -h now`commands.

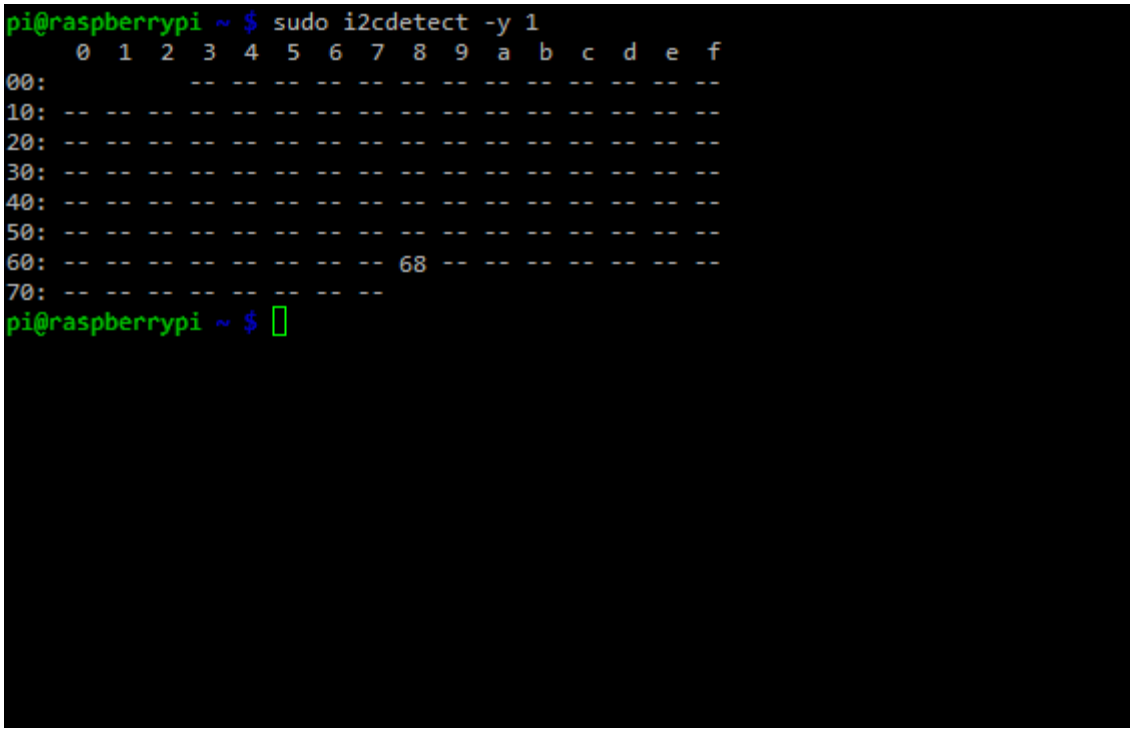
Now plug the board onto your Raspberry Pi GPIO pins on the inside corner with the RTC board pointing inwards, as shown below.



## Setting up the RTC

You now need to configure your Pi to recognise and use the board.

Turn on the Pi and verify that the board is seen by running `sudo i2cdetect -y 1` at the command line. You should see ID #68 show up (the I2C address of the DS1307 Real Time Clock chip):




If you have an older Rev. 1 Pi, you will have to run `sudo i2cdetect -y 0` as the I2C bus address changed from 0 to 1 from rev 2 onwards.

## Setting the RTC Clock

The RTC module must be loaded by the kernel by running:

```
sudo modprobe rtc-ds1307
```

Now you need to be running as the super user; type in:

```
sudo bash
```

and then, if you have a Rev.2 Pi or later:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

Or if you have the older Rev.1 Pi, use:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device
```

Type `exit` to stop being the 'root' user.

Check the time on the RTC device using:

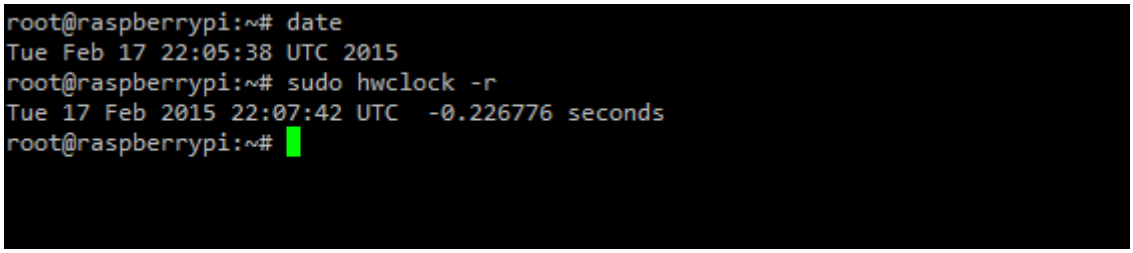
```
sudo hwclock -r
```

If this is the first time the module has been used, it will report back Jan 1 2000, and you'll need to set the time to the current time.

The easiest way is to connect it up to Ethernet or Wifi. It will automatically set the time from the network. You can check the time using the `date`command:

Run `sudo hwclock -w` to write the system time to the RTC board.

You can then verify it with `sudo hwclock -r`:




So, now the RTC clock is set, is that it? Well, no. The next time you boot your Pi, it will not know that it needs to read the clock from the RTC, so if it is not connected to the internet it will again be showing the wrong time.

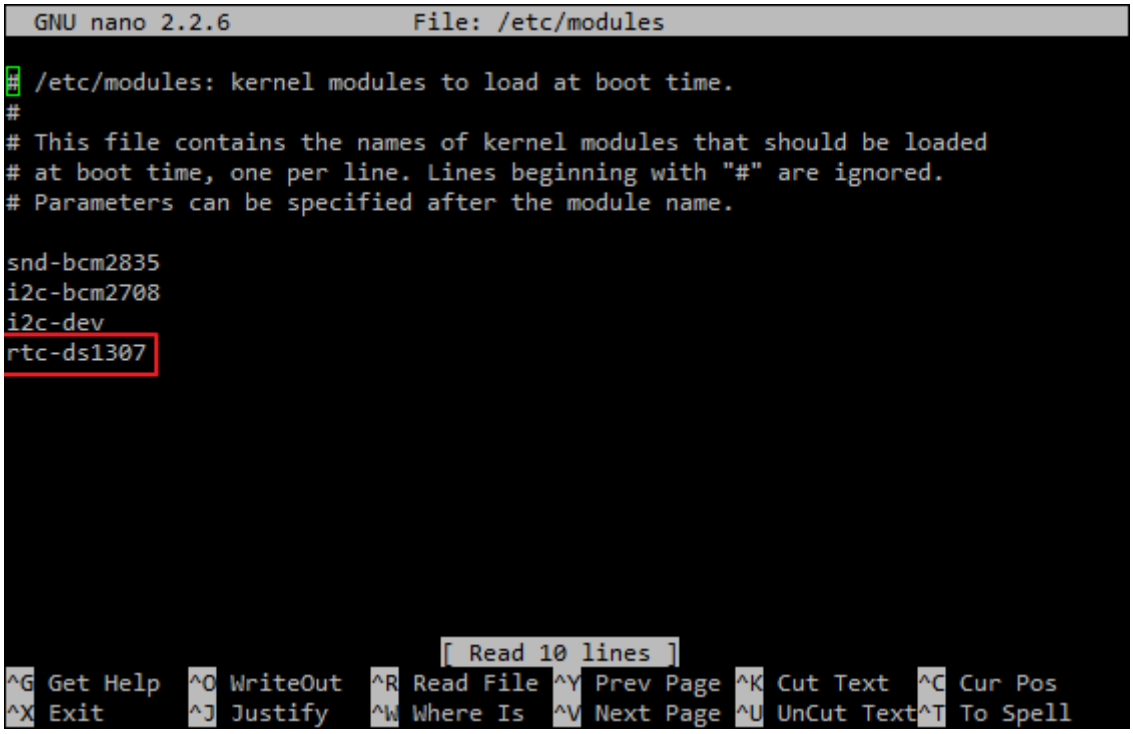
Set Date

```
sudo date -s "Wed Dec 14 08:58:00 UTC 2016"
```

```
sudo date -s "Wed Dec 14 16:14:00 ICT 2016"
```

## Making Your Pi Use the Real Time Clock

You will need to add the RTC kernel module to the file `/etc/modules` so it is loaded when the Pi boots. In a terminal window run `sudo nano /etc/modules` and add `rtc-ds1307` at the end of the file:



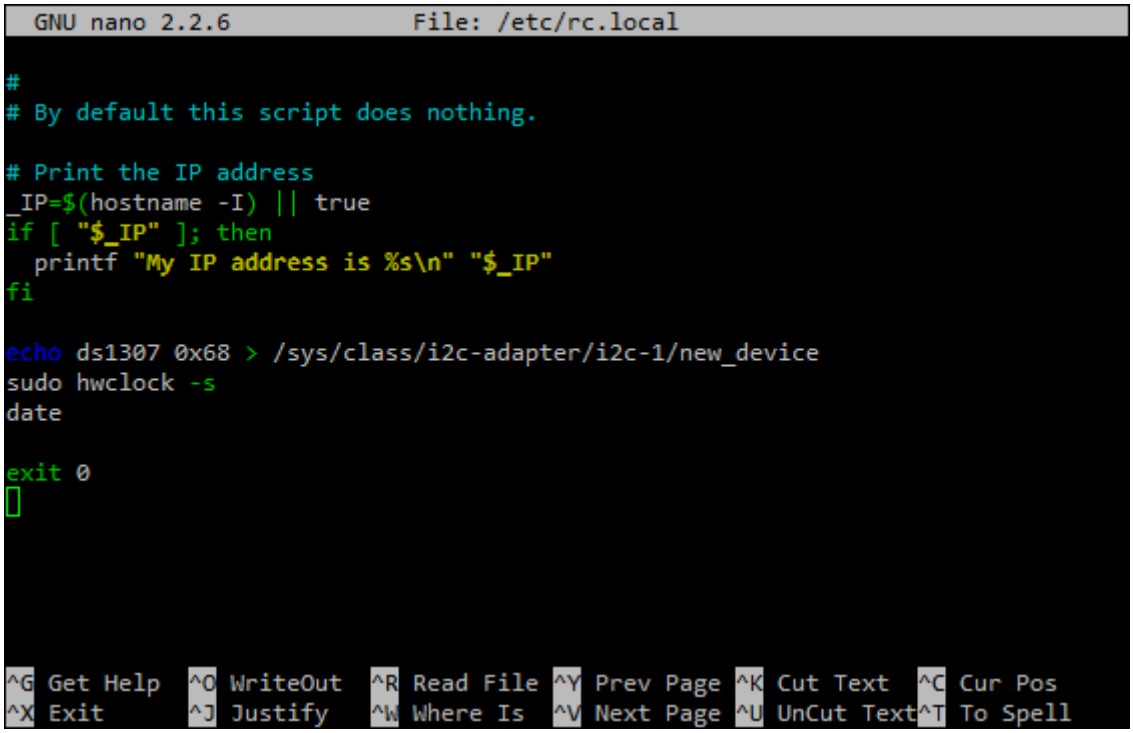

Next you will need to add the DS1307 device creation at boot by editing the `/etc/rc.local` file by running

```
sudo nano /etc/rc.local
```

and add the following lines to the file:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s

date
```

Just before the `exit 0`. ***Note:*** *If you have a Rev 1 Pi, replace* `i2c-1` *by* `i2c-0` *above.*




To save the file, press Ctrl+X, Y then return.

To test this out, shutdown your Pi, unplug any ethernet cable or wifi dongle and then turn the Pi back on. Use the following command in a terminal window to check the date is still set:

```
date
```

DONC