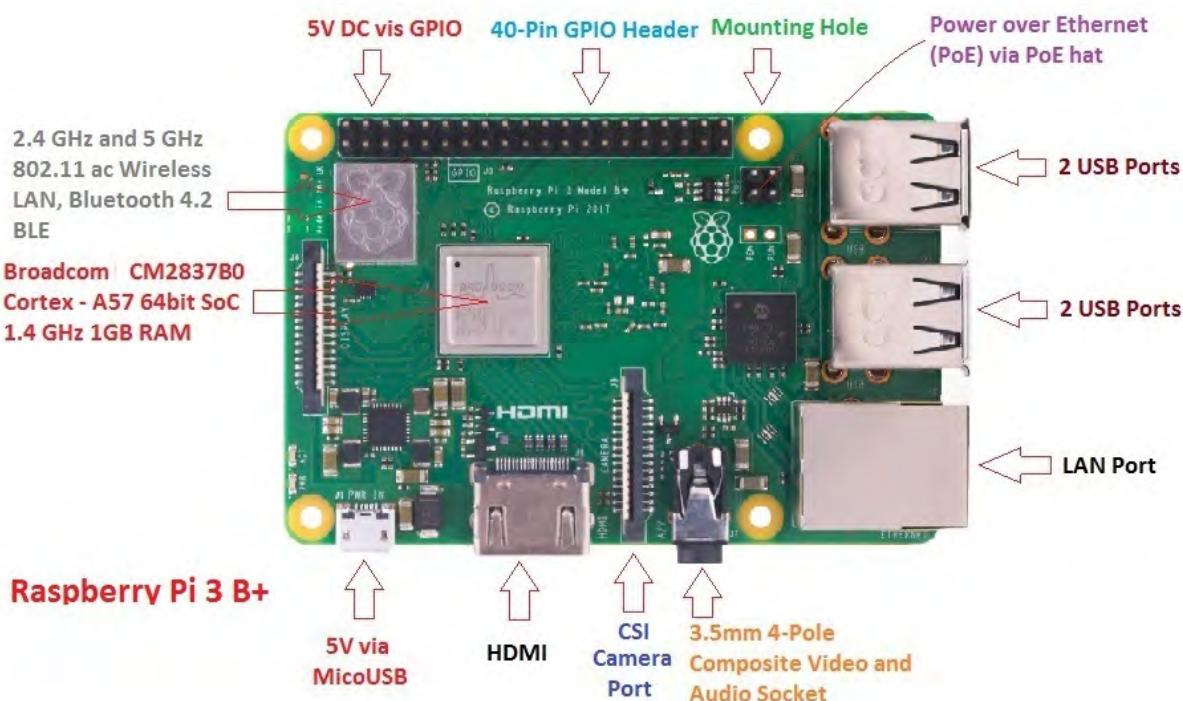


Raspberry Pi ถูกออกแบบมาให้เป็นเครื่องคอมพิวเตอร์ขนาดเล็ก มีข้าวที่สามารถเชื่อมต่อกับจอ Monitor สาย Lan และ USB ที่ไวต่อกับอุปกรณ์ เช่น Keyboard และ Mouse แล้วขึ้น Wireless LAN และ Bluetooth สามารถนำไปทำเป็น Server อุปกรณ์ Embedded System หรือ Internet of Thing (IOT) โดยมีข้าวที่เป็น Pin อยู่ภายในบอร์ดที่เรียกว่า General Purpose Input/Output (GPIO) ไว้ติดต่อภายนอก โดยสามารถเชื่อมต่อและนำໄไปเขียนโปรแกรมเพื่อส่งค่าໄไปทำการควบคุมอุปกรณ์หรือรับค่าข้อมูลจาก Sensor ต่างๆภายนอกได้ ตัวเครื่อง Raspberry Pi นี้สามารถลงระบบปฏิบัติการ (OS) ได้หลากหลายในตระกูล Linux เช่น Raspbian, Ubuntu หรือจะเป็น Windows 10 IoT ที่สามารถลงได้ แต่ที่นิยมใช้มากที่สุดจะเป็น Raspbian ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาใช้กับ Raspberry Pi ซึ่งพัฒนามาจากระบบ Debian Linux จึงทำให้คำสั่งต่างๆเหมือนกับคำสั่งที่ใช้ในระบบปฏิบัติการ Linux โดยที่ตัวเครื่องที่จะนำมาทดลองจะเป็น Raspberry Pi 3 Model B+ มีรายละเอียดดังนี้

#### Specification

- Quad-Core 1.4GHz Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit
- 1GB LPDDR2 SDRAM
- Dual-channel 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)



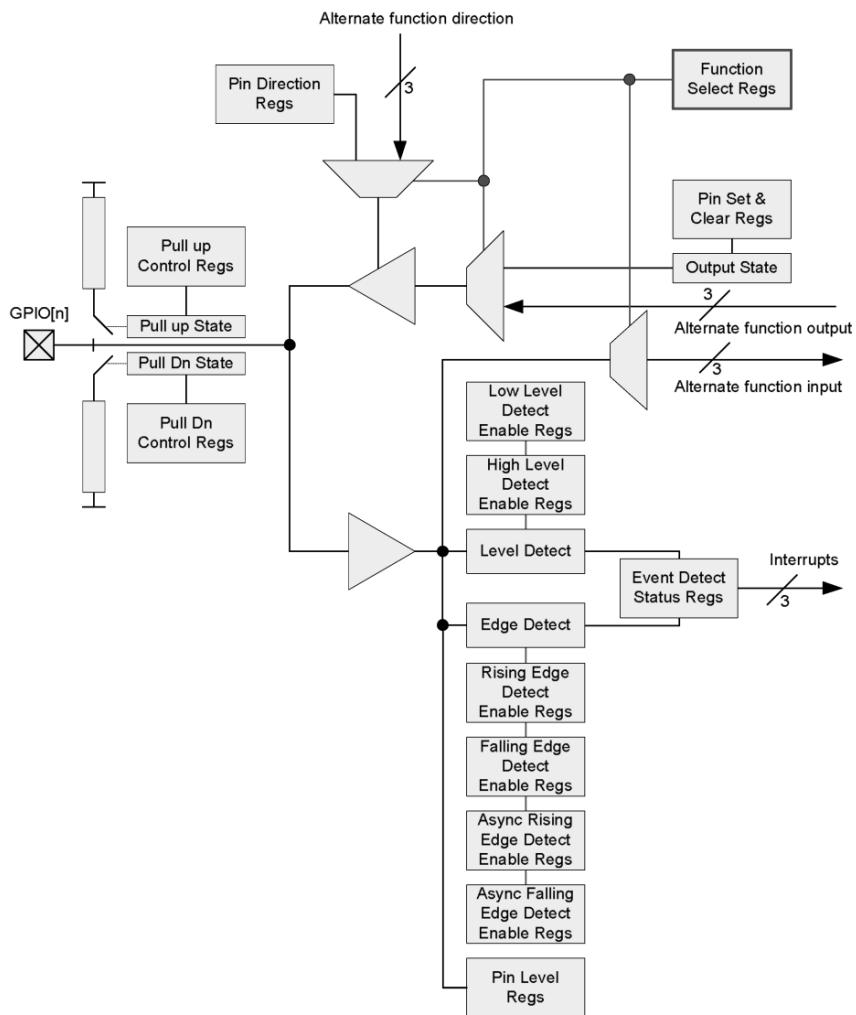
Raspberry Pi 3 B+ จะใช้ชิปประมวลผลเป็น Broadcom BCM2837B0 เป็นชิปที่ใช้ ARM Cortex-A53 ถูกพัฒนาโดยมีสถาปัตยกรรมแบบ 64 บิต ทำงานที่ความเร็ว 1.4 GHz และเป็น Quad-Core ARMv8 หน่วยความจำของบอร์ด Raspberry Pi 3 B+ จะถูกเก็บระบบปฏิบัติการและข้อมูลไว้ในкар์ดหน่วยความจำ Micro SD โดยมีช่องเสียบ Micro SD อยู่ที่ใต้บอร์ด โดยทั่วไปควรเป็น SDHC ที่มีความจุตั้งแต่ 16 GB Class 4 ขึ้นไป ซึ่งจะเก็บข้อมูลเป็นแบบ FAT32 และ Ext4

บอร์ด Raspberry Pi 3+ มีจุดเชื่อมต่ออุปกรณ์ภายนอกดังนี้

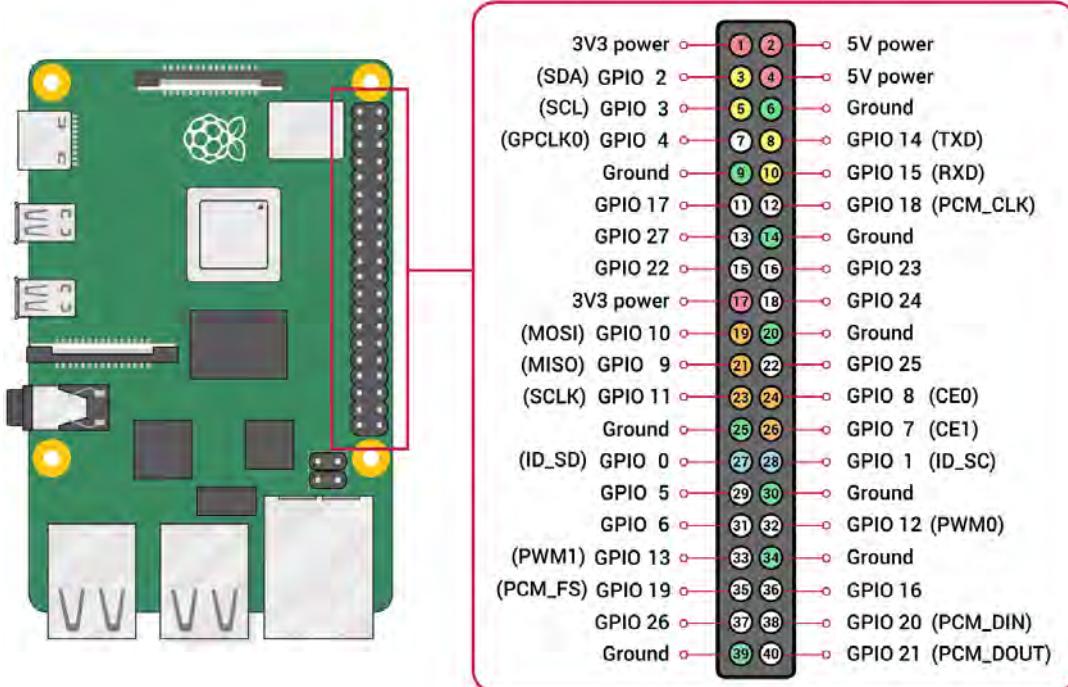
- HDMI สำหรับต่อจอแสดงผล HDMI 1.3
- ช่องเสียบอาต์พูตเสียงและสัญญาณ Component Video
- Micro USB ใช้สำหรับต่อเป็นแหล่งจ่ายไฟ +5V
- Camera Serial Interface (CSI) สำหรับการเชื่อมต่อโมดูลกล้อง
- Display Serial Interface (DSI) สำหรับเชื่อมต่อจอแสดงผล
- LAN สำหรับเชื่อมต่อเครือข่ายผ่านพอร์ต Ethernet
- Wi-Fi และ BLE ภายในเป็นวงจรสื่อสารข้อมูลไร้สายใช้ในการเชื่อมต่อเครือข่ายอินเทอร์เน็ตแบบไร้สาย Wireless LAN และ Bluetooth 4.1 โดยใช้ชิป BCM43438
- USB สำหรับเชื่อมต่ออุปกรณ์ภายนอกจำนวน 4 พอร์ต

General Purpose Input/Output (GPIO) เป็นพอร์ต\_outon กประสงค์ภายใน BCM ที่เป็นจุดใช้สำหรับเชื่อมต่อเข้ากับวงจร อิเล็กทรอนิกส์ต่างๆ ใช้ในการคอนโทรลควบคุมรับและส่งค่าดิจิตอลต่างๆ ทั้งหมดจำนวน 40 Pin มีความยืดหยุ่นในการเลือก ระดับแรงดันไฟฟ้า โดยจะเป็นพอร์ตอินพุต/อาต์พูต 26 Pin เป็นขาอาต์พูตสัญญาณ PWM แบบชาร์ดแวร์ 4 ขา คือ GPIO 12, GPIO 13, GPIO 18 , GPIO 19 และสามารถทำ PWM แบบซอฟต์แวร์ได้ทุกขา แต่จะไม่สามารถรองรับอินพุตอาต์พูตที่เป็น สัญญาณอนาล็อกได้โดยตรง รองรับฟังก์ชันการรับส่งสัญญาณบัสที่เป็นพอร์ตสำหรับการเชื่อมต่อสื่อสารข้อมูลแบบบาน្តอุตุนิยม UART 1 ช่อง โดยมีการแบ่งการทำงานกับพอร์ต GPIO ปกติมีสายสัญญาณ TxD ที่ GPIO 14 และ RxD ที่ GPIO 15 มีพอร์ต เชื่อมต่อการทำงานแบบบัส I2C จำนวน 1 ช่อง โดยที่จะแบ่งการทำงานใช้สายสัญญาณ SDA ที่ GPIO 2 และ SCL ที่ GPIO 3 และบัส I2C สำหรับ EEPROM Data ที่ GPIO0 (Pin27) , EEPROM Clock ที่ GPIO1 (Pin28) นอกจากนี้ยังมีพอร์ตเชื่อมต่อ บัสแบบ SPI จำนวน 1 ช่อง มีสายสัญญาณ 3 เส้น โดยมีการแบ่งการทำกับพอร์ต GPIO คือ MISO ที่ GPIO 9 , MOSI ที่ GPIO 10 และ SCK GPIO 11 และมีพอร์ตเสริมที่ไว้เลือกอุปกรณ์ที่จะเชื่อมต่อคือ CE0 และ CE1 โดยแบ่งการทำงานกับพอร์ต GPIO 8 และ 7 ตามลำดับ

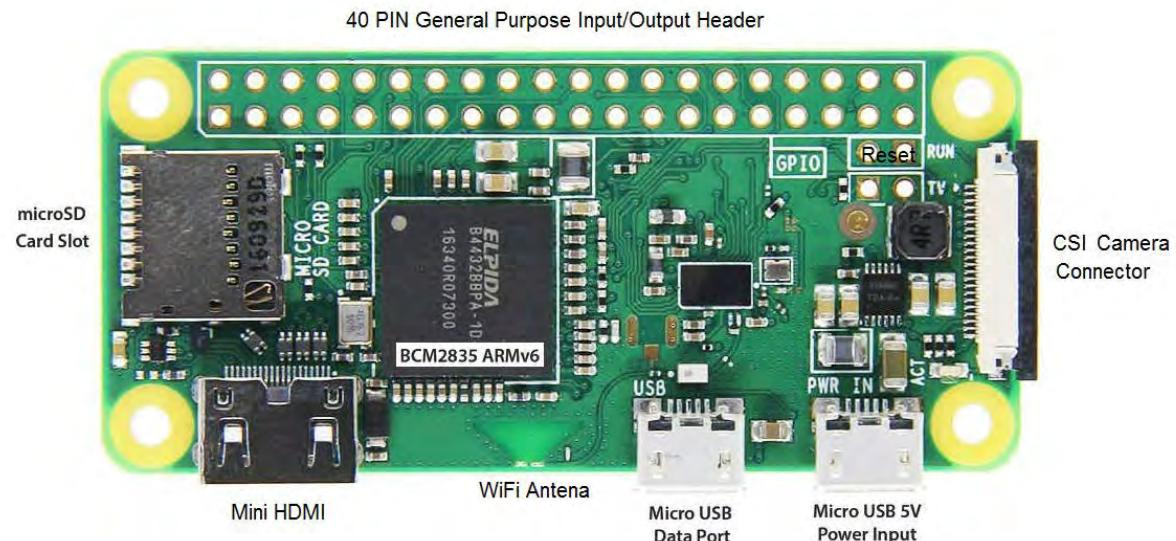
รายละเอียดของ Block Diagram สำหรับฟังก์ชันในการเชื่อมต่อแต่ละขาของ GPIO เป็นดังนี้



GPIO Block Diagram



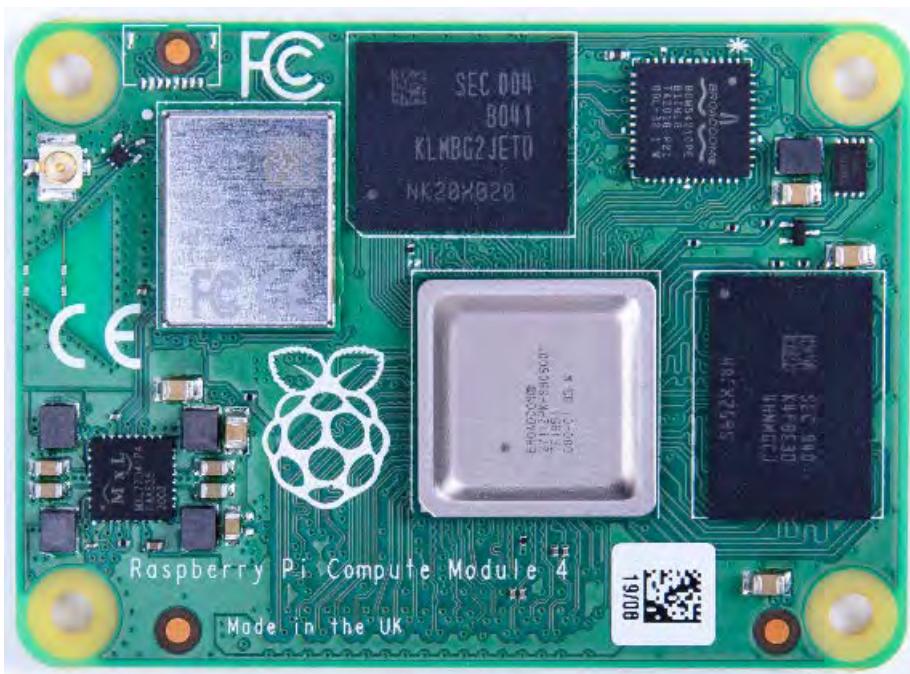
บอร์ด Raspberry Pi Zero W เป็นบอร์ดรุ่นเล็กที่มี Wireless LAN WiFi เป็น WiFi 2.4 GHz IEEE 802.11b/g/n และ Bluetooth 4.1 BLE เพิ่มเติมขึ้นมาจากรุ่น Raspberry Pi Zero ใช้ชิป SoC ของ Broadcom เบอร์ BCM2835 ตัวเดียวกับ Raspberry Pi Model B/A+/B+ ทำงานที่ความเร็ว 1 GHz ตัวบอร์ดมีแรม 512MB ฝั่งมาในชิป SoC เป็น RAM LPDDR2 มีช่องใส่ micro SD card มีพอร์ต mini HDMI ไว้ดูจอแสดงผล มีพอร์ต CSI Camera Interface ที่ต้องใช้สายแปลงสำหรับต่อโมดูลกล้อง Raspberry Pi Camera มีพอร์ต micro USB สำหรับต่อแหล่งจ่ายไฟ และ micro USB OTG / Host สำหรับต่ออุปกรณ์พร้อมขา GPIO 40 ขาที่เหมือนกับของ Raspberry Pi รุ่นอื่นๆ



#### Specification

- 802.11 b/g/n wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI port and micro USB On-The-Go (OTG) port
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector

Raspberry Pi Compute Module 4 (CM4) เป็นวงจรอัญมณี模組 (SoM) ที่มีส่วนประกอบเป็นโปรเซสเซอร์, หน่วยความจำ และ eMMC Flash ไม่ต้องเป็นทางเลือกเพิ่มเติมให้กับนักออกแบบทำให้สามารถใช้ประโยชน์จากอาร์ดแวร์ และซอฟต์แวร์ของ Raspberry Pi ใช้ในระบบและฟอร์มแฟกเตอร์ที่กำหนด เพื่อเป็นการขยายให้มีอินเทอร์เฟซ หรืออุปกรณ์ IO พิเศษเพิ่มมากขึ้นกว่าที่มีอยู่ในบอร์ดของ Raspberry Pi ปกติ สำหรับรุ่นที่ไม่ได้ติดตั้ง eMMC จะเป็นเวอร์ชันนี้เรียกว่า Raspberry Pi Compute Module 4 Lite (CM4Lite)



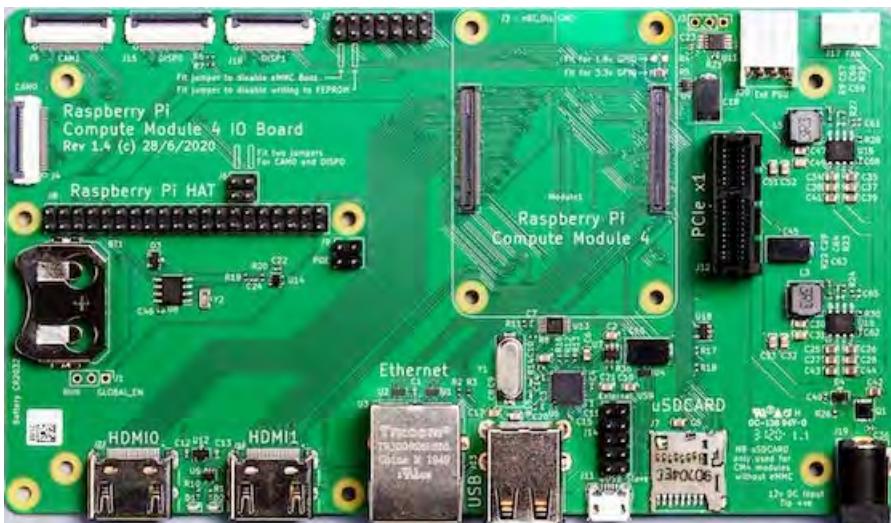
#### Specification

Processor:	Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB, 4GB or 8GB LPDDR4
Connectivity:	<ul style="list-style-type: none"> <li>• Optional wireless LAN, 2.4GHz and 5.0GHz IEEE 802.11b/g/n/ac wireless, Bluetooth 5.0, BLE with onboard and external antenna options</li> <li>• Onboard Gigabit Ethernet PHY supporting IEEE1588</li> <li>• 1 x USB 2.0 interface</li> <li>• PCIe Gen 2 x1 interface</li> <li>• 28 GPIO signals</li> <li>• SD card interface for SD card or external eMMC (for use only with Compute Module 4 variants without eMMC)</li> </ul>
Video:	<ul style="list-style-type: none"> <li>• Dual HDMI interface (up to 4Kp60 supported)</li> <li>• 2-lane MIPI DSI display interface</li> <li>• 4-lane MIPI DSI display interface</li> <li>• 2-lane MIPI CSI camera interface</li> <li>• 4-lane MIPI CSI camera interface</li> </ul>
Multimedia:	H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), OpenGL ES 3.0 graphics
Input power:	5V DC
Operating temperature:	-20°C to +85°C

## Compute Module 4 IO Board

### Specification

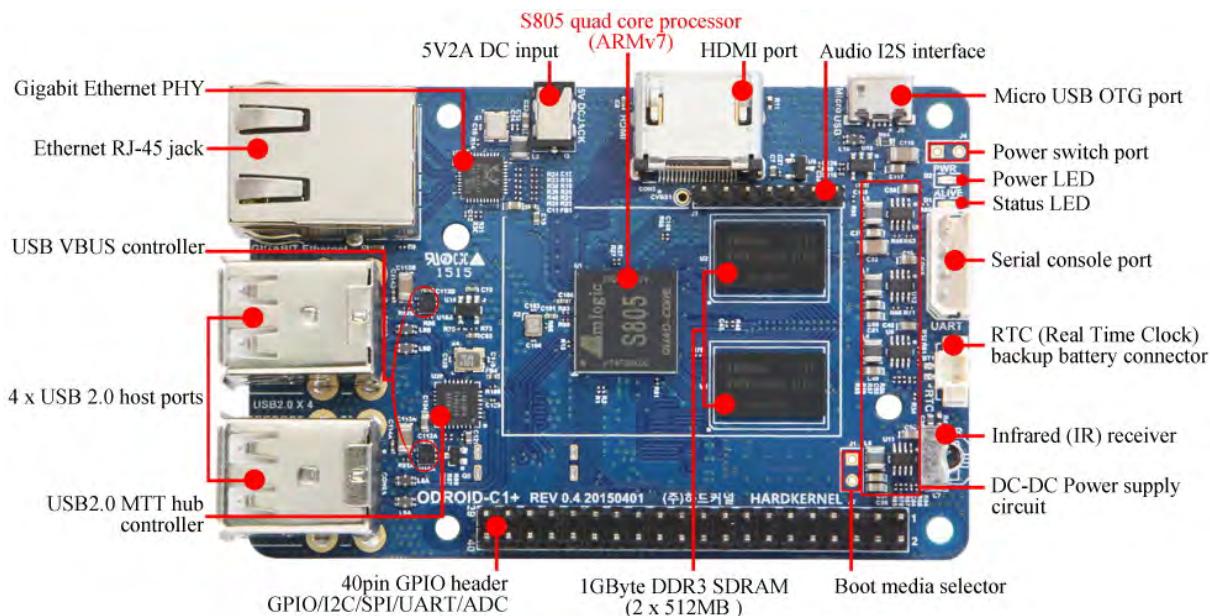
External power connector (+12V, +5V)  
2 x full-size HDMI 2.0 connectors  
2 x USB 2.0 connectors, with header for two additional connectors  
Gigabit Ethernet RJ45 with PoE support  
Micro USB socket for updating Compute Module 4  
MicroSD card socket for Compute Module 4 Lite (without eMMC) variants  
PCIe Gen 2 x1 socket  
Standard fan connector  
2 x MIPI DSI display FPC connectors (22-pin 0.5 mm pitch cable)  
2 x MIPI CSI-2 camera FPC connectors (22-pin 0.5 mm pitch cable)  
Standard Raspberry Pi HAT connectors  
Real-time clock with battery socket and ability to wake Compute Module 4  
Various jumpers to disable specific features, e.g. wireless connectivity, EEPROM writing



Mini Base Board Designed for Raspberry Pi Compute Module 4



ODROID C1+ เป็นคอมพิวเตอร์บอร์ดเดี่ยวที่เป็นอุปกรณ์เนกประสงค์ใช้สถาปัตยกรรมโปรเซสเซอร์ ARM ประกอบด้วยโปรเซสเซอร์ Amlogic แบบสี่แกน GPU Mali และ Gigabit Ethernet เป็นอุปกรณ์สร้างต้นแบบสำหรับการปรับแต่งฮาร์ดแวร์ตัวควบคุมระบบอัตโนมัติแบบเวิร์กสเตชันสำหรับการพัฒนาซอฟต์แวร์พร้อมแพ็คเกจซอฟต์แวร์โอเพ่นซอร์สหลายชนิด ใช้ระบบปฏิบัติการทำงานได้แก่ Ubuntu, Android, Debian และ Linux อีนๆ หมายความว่าใช้เป็นอุปกรณ์ตัวควบคุมแบบฝังตัวขนาดเล็ก



ขา GPIO ในโปรเซสเซอร์ของบอร์ด ODROID ที่ใช้ Ubuntu จะยังไม่สามารถทำงานได้ จะต้องทำการติดตั้งไลบรารีเพื่อให้สามารถเข้าถึง GPIO ที่เป็น PIN บนบอร์ดได้ เพื่อให้สามารถเปลี่ยนคำสั่งจากเทอร์มินัลไปสั่งงานอุปกรณ์ต่างๆ ในการติดตั้งแพ็คเกจเพื่อใช้งาน GPIO จะต้องทำการอัพเดทซอฟแวร์ให้เป็นเวอร์ชันใหม่ก่อนที่จะติดตั้งโปรแกรมให้พิมคำสั่ง

```
sudo apt update
```

```
sudo apt upgrade
```

ติดตั้ง GPIO โดยใช้คำสั่งเพิ่มที่เก็บ PPA ในรายการ อัปเดตรายการแพ็คเกจที่สามารถติดตั้งในระบบ แล้วติดตั้งแพ็คเกจ odroid-wiringpi ด้วยคำสั่ง

```
sudo apt install software-properties-common
```

```
sudo add-apt-repository ppa:hardkernel/ppa
```

```
sudo apt update
```

```
sudo apt install odroid-wiringpi
```

```
sudo apt install libwiringpi-dev libwiringpi2
```

```

root@odroid: ~
root@odroid:~# sudo apt update
Hit:1 http://ports.ubuntu.com/ubuntu-ports focal InRelease
Hit:2 http://ppa.launchpad.net/hardkernel/ppa/ubuntu focal InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports focal-updates InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports focal-backports InRelease
Hit:5 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease
Hit:6 http://deb.odroid.in/c1 focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@odroid:~# sudo apt install odroid-wiringpi
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libwiringpi2
The following NEW packages will be installed:
  libwiringpi2 odroid-wiringpi
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 61.7 kB of archives.
After this operation, 180 kB of additional disk space will be used.
Do you want to continue? [Y/n] 

```

ODROID C1/C1+ (J2 Header)

WiringPi GPIO#	NAME(GPIO#)		WiringPi GPIO#
	3.3 V Power	1	5.0 V Power
	I2CA_SDA (#74)	2	5.0 V Power
	I2CA_SCL (#75)	4	Ground
7	GPIO (#83)	6	TXD1 (#113)
	Ground	8	RXD1 (#114)
0	GPIO (#88)	10	GPIO (#87)
2	GPIO (#116)	12	1
3	GPIO (#115)	14	Ground
	3.3V Power	16	GPIO (#104)
12	MOSI_PWM1 (#107)	18	GPIO (#102)
13	MISO (#106)	20	5
14	SPI_SCLK (#105)	22	Ground
	Ground	24	GPIO (#103)
	I2CB_SDA (#76)	26	6
21	GPIO (#101)	28	GPIO (#117)
22	GPIO (#100)	30	10
23	PWM0 (#108)	32	GPIO (#118)
24	GPIO (#97)	34	11
AIN1	ADC.AIN1 (ADC#1)	36	I2CB_SCL (#77)
	Ground	38	Ground
		40	1.8 V Power
			ADC.AIN0 (ADC#0)
			AIN0

ทดสอบการทำงานของขา GPIO บน ODRID หรือ Raspberry Pi (สำหรับ Raspberry Pi ไม่ต้องติดตั้งแพ็คเกจ) โดยสั่งงานให้รับข้อมูล GPIO ทุกขาและแสดงบนหน้าจอ ด้วยคำสั่ง

`gpio readall`

ทดสอบการส่งข้อมูลของขา GPIO บน ODRID ให้ต่อวงจรที่มีตัว LED อนุกรมกับตัวความดันทาน 300 โอม โดยให้ปลายขาข้างหนึ่งที่เป็นขา Anode ของ LED ต่อ กับขา GPIO 116 และให้ปลายขาอีกข้างของตัวความดันทานต่อลงกราวด์ แล้วให้ทดลองการทำงานของขา GPIO ด้วยคำสั่ง

```
gpio -g mode 116 out
gpio -g write 116 1
gpio -g write 116 0
gpio -g write 116 1
```

ถ้าใช้ Raspberry Pi 4 หรือ kali จะต้องเป็น gpio version 2.52 ขึ้นไป ทดสอบด้วยคำสั่ง  
`gpio -v`

ถ้าไม่ใช่หรือไม่มีคำสั่งนี้ ให้ทำการดาวน์โหลดและติดตั้งรุ่นล่าสุดของ wiringpi ดังนี้

```
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
```

กรณีที่มีปัญหาในการใช้งาน wiringpi หรือติดตั้งไม่ได้ ถ้าใช้ Raspberry Pi OS ของ raspios-bullseye ให้ติดตั้งเป็น version 2.61 แยกตามขนาดของ OS ใช้คำสั่งในการติดตั้งสำหรับ 32 บิตดังนี้

```
wget https://github.com/WiringPi/WiringPi/releases/download/2.61-1/wiringpi-2.61-1-armhf.deb
sudo dpkg -i wiringpi-2.61-1-armhf.deb
```

คำสั่งในการติดตั้งสำหรับ 64 บิตดังนี้

```
wget https://github.com/WiringPi/WiringPi/releases/download/2.61-1/wiringpi-2.61-1-arm64.deb
sudo dpkg -i wiringpi-2.61-1-arm64.deb
```

กรณีที่ใช้ Raspberry Pi ของ raspios-bullseye จะมีรูปแบบคำสั่งของ GPIO ที่ติดตั้งมา กับ OS ดังนี้

#### ***raspi-gpio get [GPIO]***

<code>raspi-gpio get</code>	<i>; print the state of all GPIO pins</i>
<code>raspi-gpio get X</code>	<i>; print the state of GPIO pin X</i>

#### ***raspi-gpio funcs [GPIO]***

<code>raspi-gpio funcs</code>	<i>; dump alt function of all GPIO pins</i>
<code>raspi-gpio funcs X</code>	<i>; dump alt function of GPIO pin X</i>

#### ***raspi-gpio set <GPIO> [options]***

<code>raspi-gpio set X op</code>	<i>; set GPIO pin X as an output</i>
<code>raspi-gpio set X dh</code>	<i>; set GPIO pin X to drive high</i>
<code>raspi-gpio set X dl</code>	<i>; set GPIO pin X to drive low</i>

Valid [options] for raspi-gpio set are:

<b>ip</b>	set GPIO as input
<b>op</b>	set GPIO as output
<b>a0-a5</b>	set GPIO to alternate function alt0-alt5
<b>pu</b>	set GPIO in-pad pull up
<b>pd</b>	set GPIO pin-pad pull down
<b>pn</b>	set GPIO pull none (no pull)
<b>dh</b>	set GPIO to drive to high (1) level (only valid if set to be an output)
<b>dl</b>	set GPIO to drive low (0) level (only valid if set to be an output)

สำหรับ Ubuntu ให้ทำการติดตั้งแพ็คเกจ เพื่อให้สามารถใช้ขา GPIO บนภาษา Python ด้วยคำสั่งดังนี้

```
sudo apt install software-properties-common
sudo add-apt-repository ppa:hardkernel/ppa
sudo apt update
sudo apt install odroid-wiringpi-python
```

```
root@odroid:~# sudo apt install odroid-wiringpi-python
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libexpat1-dev libpython3-dev libpython3.8-dev python-pip-whl python3-dev
  python3-distutils python3-pip python3-setuptools python3-wheel python3.8-dev
  zlib1g-dev
Suggested packages:
  python-setuptools-doc
The following NEW packages will be installed:
  libexpat1-dev libpython3-dev libpython3.8-dev odroid-wiringpi-python
  python-pip-whl python3-dev python3-distutils python3-pip python3-setuptools
  python3-wheel python3.8-dev zlib1g-dev
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 6438 kB of archives.
After this operation, 20.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

หลังจากนั้นให้ติดตั้งแพ็คเกจ odroid-wiringpi โดยใช้ PIP ให้พิมพ์คำสั่งดังนี้

```
sudo apt install python3-pip
sudo odroid-wiringpi --install
```

```
root@odroid:~# Collecting odroid-wiringpi
  Downloading odroid_wiringpi-3.8.tar.gz (142 kB)
    |██████████| 142 kB 1.3 MB/s
Building wheels for collected packages: odroid-wiringpi
  Building wheel for odroid-wiringpi (setup.py) ... done
  Created wheel for odroid-wiringpi: filename=odroid_wiringpi-3.8-cp38-cp38-linux_armv7l.whl size=396065 sha256=2cd26c9cffefdda5ce15646c23f334cd22b28cbe61872090190145b472421c69
  Stored in directory: /root/.cache/pip/wheels/32/bd/ba/01fb4759de54ad78ab1885a5ed39f7a24d6717670d9c5d13a1
Successfully built odroid-wiringpi
Installing collected packages: odroid-wiringpi
Successfully installed odroid-wiringpi-3.8
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...
root@odroid:~# sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
python3-pip set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@odroid:~# sudo odroid-wiringpi --install
Requirement already up-to-date: odroid-wiringpi in /usr/local/lib/python3.8/dist-packages (3.8)
root@odroid:~# 
```

ทดสอบการทำงานของขา GPOI บน ODRID โดยใช้ภาษา Python ด้วยคำสั่งดังนี้

```

import odroid_wiringpi as wpi
import time

channel = 2                      # GPIO 116 on odroid
wpi.wiringPiSetup()
wpi.pinMode(channel, 1)

while True:
    wpi.digitalWrite(channel, 1)
    print ("LED1 on")
    time.sleep(1)
    wpi.digitalWrite(channel, 0)
    print ("LED1 off")
    time.sleep(1)

```

ถ้าต้องการเขียนโปรแกรมเพื่อใช้งานพีซีชั้น GPIO พื้นฐานด้วยภาษา Python บนบอร์ด ODROID โดยใช้คำสั่งเหมือนกับ Raspberry Pi จะต้องทำการติดตั้งแพ็คเกจเพิ่มเติม เพื่อให้ใช้งานไลบรารี RPi.GPIO สำหรับใช้ GPIO บน Raspberry Pi โดยจะต้องโคลนที่เก็บ Github ของเรางและสร้างมันขึ้นมาใหม่ ด้วยคำสั่ง

```

sudo apt install git
git clone https://github.com/awesometric/RPi.GPIO-Odroid
แล้วให้ติดตั้งด้วยคำสั่งดังนี้
cd RPi.GPIO-Odroid
sudo python3 setup.py build install

```

```

root@odroid: ~/RPi.GPIO-Odroid
running install
running install_lib
creating /usr/local/lib/python3.8/dist-packages/RPi
creating /usr/local/lib/python3.8/dist-packages/RPi/GPIO
copying build/lib.linux-armv7l-3.8/RPi/GPIO/__init__.py -> /usr/local/lib/python3.8/dist-packages/RPi/GPIO
copying build/lib.linux-armv7l-3.8/RPi/_GPIO.cpython-38-arm-linux-gnueabihf.so -> /usr/local/lib/python3.8/dist-packages/RPi
copying build/lib.linux-armv7l-3.8/RPi/__init__.py -> /usr/local/lib/python3.8/dist-packages/RPi
byte-compiling /usr/local/lib/python3.8/dist-packages/RPi/GPIO/__init__.py to __init__.cpython-38.pyc
byte-compiling /usr/local/lib/python3.8/dist-packages/RPi/__init__.py to __init__.cpython-38.pyc
running install_egg_info
Writing /usr/local/lib/python3.8/dist-packages/RPi.GPIO-0.6.3.post1.egg-info
root@odroid:~/RPi.GPIO-Odroid#

```

ทดสอบการทำงานของขา GPIO บน ODROID หรือ Raspberry Pi (สำหรับ Raspberry Pi ไม่ต้องติดตั้งแพ็คเกจ) โดยใช้คำสั่งภาษา Python ของ Raspberry Pi ด้วยคำสั่งดังนี้

```

import RPi.GPIO as GPIO
import time

channel = 27                      # GPIO 27 on Raspberry Pi
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.OUT)

while True:
    GPIO.output(channel, True)
    print ("LED1 on")
    time.sleep(1)
    GPIO.output(channel, False)
    print ("LED1 off")
    time.sleep(1)

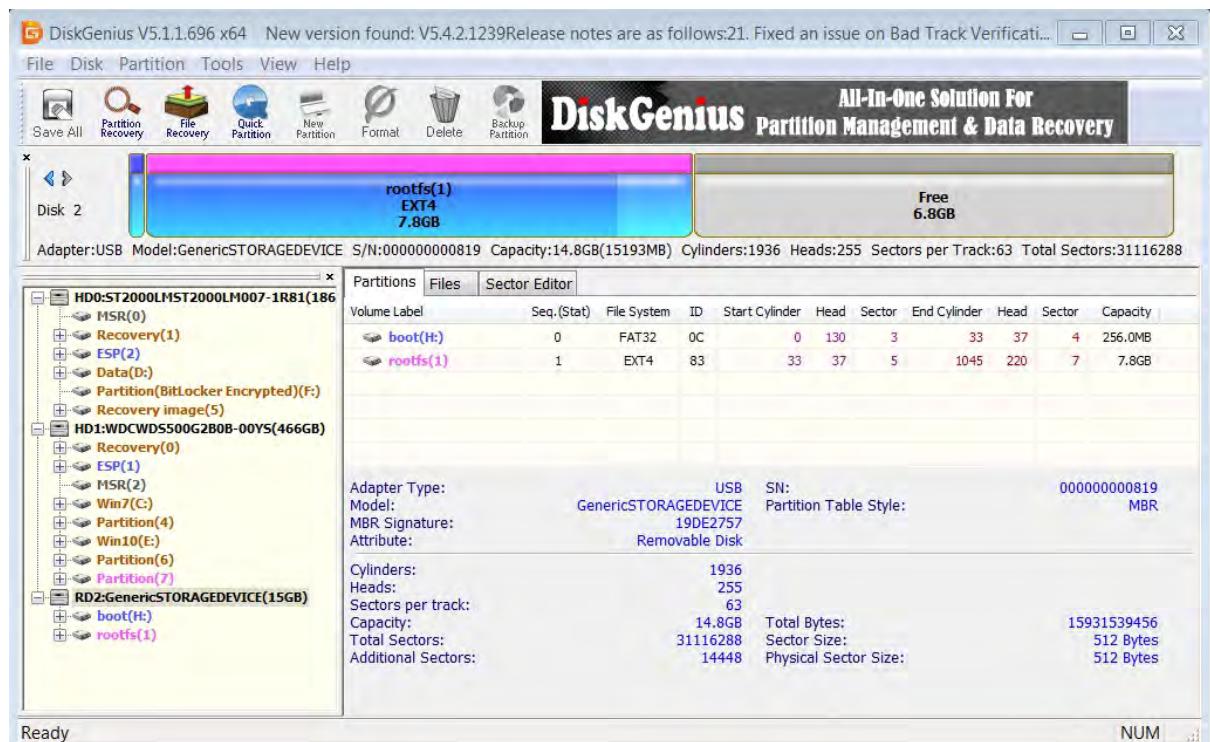
```

การทดลองการทำงานของ Raspberry Pi จะต้องติดตั้งระบบปฏิบัติการให้อยู่ใน Micro SD การ์ดของ Raspberry Pi ซึ่งทาง Raspberry Pi Foundation ได้แนะนำระบบปฏิบัติการเวอร์ชันสำหรับบอร์ด Raspberry Pi 3 ให้ได้ใช้งานคือ Raspbian Jessie สามารถดาวน์โหลดได้ที่ <https://www.raspberrypi.org/downloads/raspberry-pi-os/> โดยให้เลือกเป็น Raspberry Pi OS (32-bit) with desktop and recommended software

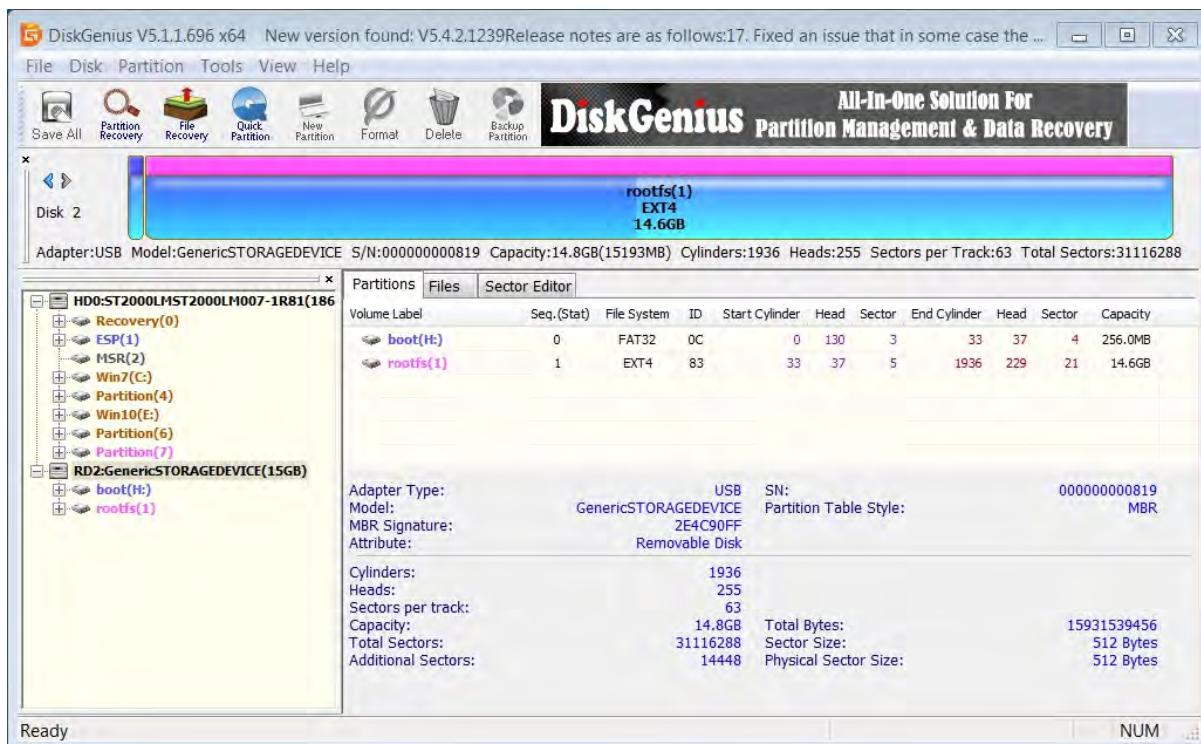
เมื่อได้ไฟล์มาแล้วจะเป็นไฟล์ชนิด Image ให้ติดตั้งโปรแกรมที่จะใช้ในการ Format และเขียนไฟล์ Raspbian ลงใน Micro SD Card โดยทั่วไปมีหลายโปรแกรมให้เลือกใช้ได้หลายโปรแกรมดังนี้

- Win32 Disk Imager - A Windows tool for writing images to USB sticks or SD/CF cards. ดาวน์โหลดได้จาก <https://sourceforge.net/projects/win32diskimager/>
- Flash OS images to SD cards & USB drives, safely and easily. ดาวน์โหลดได้จาก <https://www.balena.io/etcher/>
- SD Memory Card Formatter for Windows Download. ดาวน์โหลดได้จาก [https://www.sdcard.org/downloads/formatter/eula\\_windows/index.html](https://www.sdcard.org/downloads/formatter/eula_windows/index.html)

โดยใช้เลือกดาวน์โหลดแล้วติดตั้ง 1 โปรแกรม เสร็จแล้วใส่การ์ด Micro SD เข้าเครื่องคอมพิวเตอร์ เริ่กใช้โปรแกรมคลิกปุ่ม Browse , เลือกไฟล์.img , เลือกที่บรรจุ Micro SD และคลิกปุ่ม Write เสร็จแล้วก็จะได้ระบบปฏิบัติการ Raspbian ที่ถูกบรรจุลงการ์ด Micro SD ดังรูป



การ Boot ระบบปฏิบัติการในครั้งแรก ขนาดของ Partition ในไดร์ฟ Roots ที่เป็นระบบไฟล์แบบ EXT4 จะถูกขยายให้เต็มความจุของ Micro SD Card ตามรูป



เนื่องจากการเริ่มต้นตั้งค่าเพื่อติดตั้ง Raspbian บน Raspberry Pi นั้นทำได้ 2 แบบคือ แบบต่อจากภาพ กีบอร์ด เม้าส์ ซึ่งสามารถทำได้ปกติ และแบบที่สองคือไม่ต่อจากภาพ กีบอร์ด เม้าส์ ซึ่งแบบนี้จะต้องต่อสาย LAN จาก Raspberry Pi 3 B+ เข้ากับเครื่อง Router โดยบอร์ด Raspberry Pi จะได้รับ IP address ผ่าน DHCP Protocol ซึ่งเป็นค่า default จากระบบปฏิบัติการเมื่อ Router จ่ายค่า IP Address ผ่านสาย LAN และวิธีการเชื่อมต่อจะต้องผ่านระบบปฏิบัติการ Windows ทำการเชื่อมต่อ Raspberry Pi ผ่าน IP Address หรือจะตั้งค่า Connect ผ่าน WiFi เพื่อที่จะสามารถต่อสาย LAN ออกได้ โดยที่การตั้งค่าเริ่มต้นจะสามารถเลือกที่จะตั้งค่าผ่าน Terminal หรือ Desktop ก็ได้ โดยมีขั้นตอนการติดตั้ง Raspbian ลงบน Raspberry Pi 3 B+ ดังนี้

การจัดการในระบบปฏิบัติการ Linux โดยมากจะกระทำการผ่านหน้า Console โดยการใช้ Command Line ในส่วนของ Raspberry Pi จะกระทำการผ่านโปรแกรมที่เรียกว่า Secure Shell V2 หรือเรียกตัวย่อว่า SSH ซึ่งในการเขื่อมต่อจะทำการเปิดโหมด SSH - OpenSSH remote login client เพื่อที่จะให้สามารถรีโมทเข้ามายังคอมพิวเตอร์ตั้งค่าเริ่มต้นผ่านเครื่อง PC/Notebook ได้เนื่องจากในเวลากลางปี 2017 มีการเปลี่ยนแปลงเพื่อเพิ่มความปลอดภัยไม่ให้มีโอกาสจากการท่องเที่ยวจากโจมตีจากแฮกเกอร์ได้ จึงได้มีการปิดโหมด SSH ไว้ ทำให้ไม่สามารถเชื่อมต่อและตั้งค่าเริ่มต้นให้กับ Raspbian ได้ ซึ่งเราสามารถจะเปิดโหมด SSH Service ได้โดยการสร้างไฟล์ชื่อ ssh เก็บไว้ในไดร์ฟ Boot ที่เป็นระบบไฟล์แบบ FAT32 ในการ์ด Micro SD โดยไม่ต้องมีข้อมูลภายในไฟล์ เครื่อง Raspberry Pi ก็จะเปิดโหมด SSH ให้โดยอัตโนมัติ โดยเริ่มจากต่อการ์ด Micro SD เข้ากับเครื่องคอมพิวเตอร์ เริ่กใช้คำสั่ง cmd ของ Windows และให้เรียกไปที่ไดร์ฟของการ์ด SD เช่น E: กด enter พิมพ์คำว่า copy con ssh กด enter และกด ctrl+z ก็จะได้ไฟล์ชื่อ ssh ขนาด 0 ไบต์ ตามที่แสดงในรูป

```
C:\Users\user>H:
H:\>copy con ssh
^Z
    1 file(s) copied.

H:\>dir/w
Volume in drive H is boot
Volume Serial Number is B05C-D0C4

Directory of H:\

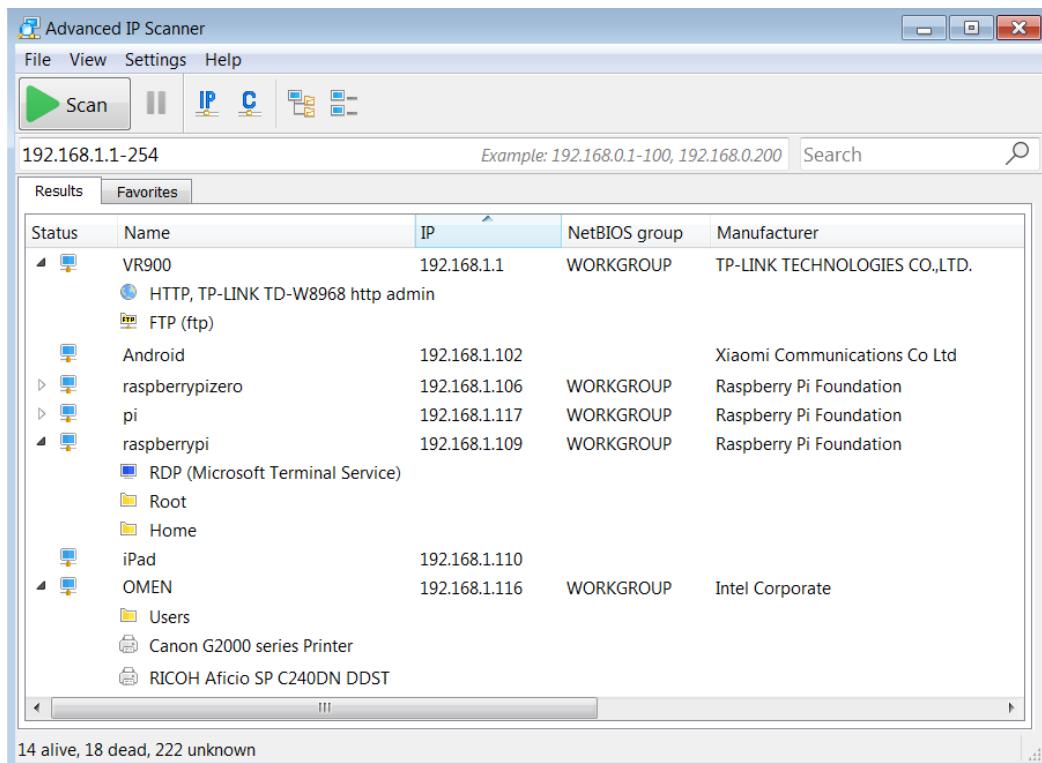
[overlays]           bcm2708-rpi-b-plus.dtb
COPYING.linux        LICENCE.broadcom
issue.txt            bcm2708-rpi-b-rev1.dtb
bcm2708-rpi-b.dtb   bcm2708-rpi-cm.dtb
bcm2708-rpi-zero-w.dtb   bcm2708-rpi-zero.dtb
bcm2709-rpi-2-b.dtb   bcm2710-rpi-2-b.dtb
bcm2710-rpi-3-b-plus.dtb   bcm2710-rpi-3-b.dtb
bcm2710-rpi-cm3.dtb   bcm2711-rpi-4-b.dtb
bcm2711-rpi-400.dtb   bcm2711-rpi-cm4.dtb
bootcode.bin         cmdline.txt
config.txt          fixup.dat
fixup4.dat          fixup4cd.dat
fixup4db.dat        fixup4x.dat
fixup_cd.dat        fixup_db.dat
fixup_x.dat         kernel.img
kernel7.img          kernel7l.img
kernel8.img          start.elf
start4.elf          start4cd.elf
start4db.elf        start4x.elf
start_cd.elf        start_db.elf
start_x.elf         ssh

               41 File(s)      49,296,992 bytes
                 1 Dir(s)     214,338,048 bytes free
```

หลังจากนั้นให้กดการ์ด Micro SD ออกจากเครื่องคอมพิวเตอร์ แล้วทำการติดตั้งการ์ด Micro SD ที่บรรจุระบบปฏิบัติการ Raspbian เข้าที่ช่องเสียบได้ด้วยร็อด Raspberry Pi 3 ต่อสาย LAN เข้าระหว่าง Raspberry Pi 3 กับเราเตอร์ ด้วยขัวต่อแบบ RJ-45 โดยจะใช้งานได้ทันทีไม่ต้องตั้งค่าใดๆ เปิดเครื่อง Raspberry Pi 3 โปรแกรมจะเริ่มนูฟเข้าสู่โหมด Desktop รองและรับ โดยระบบจะมีค่า Default คือ Username: pi และ Password: raspberry และจะมีข้อความเตือนว่า SSH ถูก Enable และให้เปลี่ยน Password

กรณีใช้เครื่องคอมพิวเตอร์ PC ให้เรียกใช้โปรแกรมสแกนเครือข่าย Advanced IP Scanner เพื่อหาหมายเลข IP ที่มีชื่อว่า raspberrypi ดาวน์โหลดซอฟต์แวร์ได้จาก <https://www.advanced-ip-scanner.com/th/> โดยที่การเรียกใช้อุปกรณ์ต่างๆที่อยู่ในเครือข่ายเดียวกันสามารถเรียกใช้จากชื่อ Name หรือหมายเลข IP ก็ได้ ดังรูป

]



กรณีที่สแกนเครือข่ายโดยใช้เป็นโทรศัพท์ Android สามารถดาวน์โหลดโปรแกรมเพื่อติดตั้งได้ที่

<https://play.google.com/store/apps/details?id=com.overlook.android.fing&hl=th&gl=US>

ตัวอย่างโปรแกรมที่ใช้สแกนเครือข่ายบนโทรศัพท์ Android แสดงดังรูป

Devices Network • Security • Internet

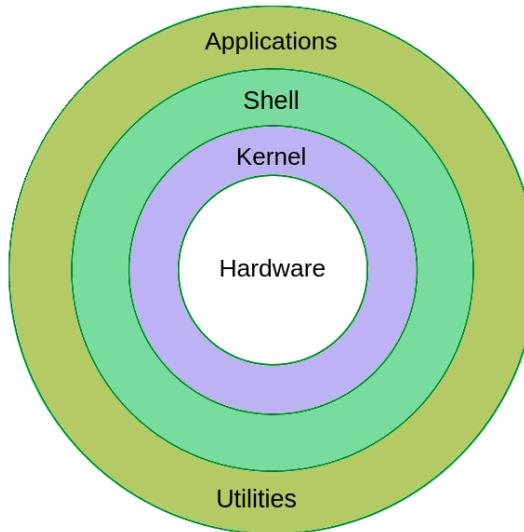
17 devices of 31 1 min ago

- Archer\_VR900 TP-Link  
192.168.1.1 Archer VR900
- Laptop Apple  
192.168.1.100 MacBook
- RASPBERYPIZERO Raspberry Pi  
192.168.1.106 Raspberry Pi
- RASPBERYPPI Raspberry Pi  
192.168.1.109 Raspberry Pi
- PI Raspberry Pi  
192.168.1.117 Raspbian

🔍 ⚡ ⚡ ⚡

### Linux Architecture

สถาปัตยกรรมของระบบปฏิบัติการ Linux จะเป็นอินเทอร์เฟซระหว่างฮาร์ดแวร์คอมพิวเตอร์และผู้ใช้คอมพิวเตอร์ เป็นกลุ่มของซอฟต์แวร์ระบบภายในคอมพิวเตอร์ที่จัดการทรัพยากรของฮาร์ดแวร์และอำนวยความสะดวกในการให้บริการพื้นฐานจัดทำแพลตฟอร์มที่ผู้ใช้สามารถเรียกใช้โปรแกรมได้อย่างมีประสิทธิภาพ โดยมีส่วนประกอบหลักคือคอร์เนลแล้วนำข้ามูลหรือแอปพลิเคชันของผู้ใช้ผ่านทางชั้น เพื่อให้เข้าถึงระบบฮาร์ดแวร์และดำเนินการต่างๆ ตามระบบปฏิบัติการ โครงสร้างของระบบ Linux แสดงตาม Layer ได้ดังรูป



Kernel เป็นส่วนประกอบหลักของระบบปฏิบัติการ มีหน้าที่อยู่เบื้องหลังบริหารทรัพยากรของระบบปฏิบัติการลีนุกซ์ ประกอบด้วยโมดูลประเภทต่างๆ โดยทำงานประสานเป็นสื่อกลางเชื่อมระหว่างฮาร์ดแวร์กับซอฟต์แวร์ เคอร์เนลทำหน้าที่จัดการทรัพยากรในคอมพิวเตอร์ ควบคุมการทำงานตั้งแต่เริ่ม Boot การใช้หน่วยประมวลผลกลาง (CPU) การใช้พื้นที่หน่วยความจำ (RAM) การจัดการไฟล์ การควบคุมอุปกรณ์ Input/Output

System Libraries เป็นชุดของฟังก์ชันไลบรารีสามารถระบุเป็นฟังก์ชันพิเศษบางอย่างได้ ฟังก์ชันเหล่านี้ใช้งานโดยระบบปฏิบัติการ และไม่ต้องการสิทธิ์การเข้าถึงโ苦คบันโมดูลของคอร์เนล

Shell เป็นอินเทอร์เฟซระหว่างคอร์เนลและผู้ใช้ ผู้ใช้ (user) สามารถสั่งงานระบบ Linux และเรียกใช้ฟังก์ชันของคอร์เนลได้ โดยผ่านทาง Shell เช่นในระบบปฏิบัติการแบนงอกเป็นสองประเภทซึ่งมีลักษณะแตกต่างกัน ได้แก่ เชลล์แบบกราฟิกรับคำสั่งให้ใช้งานสำหรับอินเทอร์เฟซส่วนติดต่อผู้ใช้แบบกราฟิก และเชลล์แบบบรรทัดรับคำสั่งให้ใช้งานอินเทอร์เฟซสำหรับรับคำสั่งเป็นบรรทัด Shell ขึ้น มีคุณสมบัติของ Programming Language ทำให้ผู้ใช้สามารถนำคำสั่งต่างๆ ของ Shell มาเขียนรวมกันเป็นโปรแกรม Shell Script เก็บเป็นไฟล์ได้โดยกำหนดเป็นชนิดไฟล์ ชื่อ.sh และกำหนดให้ไฟล์ Attributes เป็น Execute permissions ด้วยคำสั่ง

```
sudo chmod +x ชื่อ.sh
```

ถ้าต้องการดู Attributes ให้ใช้คำสั่ง

```
ls -l
```

ถ้าต้องการ Execute ให้โปรแกรมทำงานใช้คำสั่ง

```
./ชื่อ.sh
```

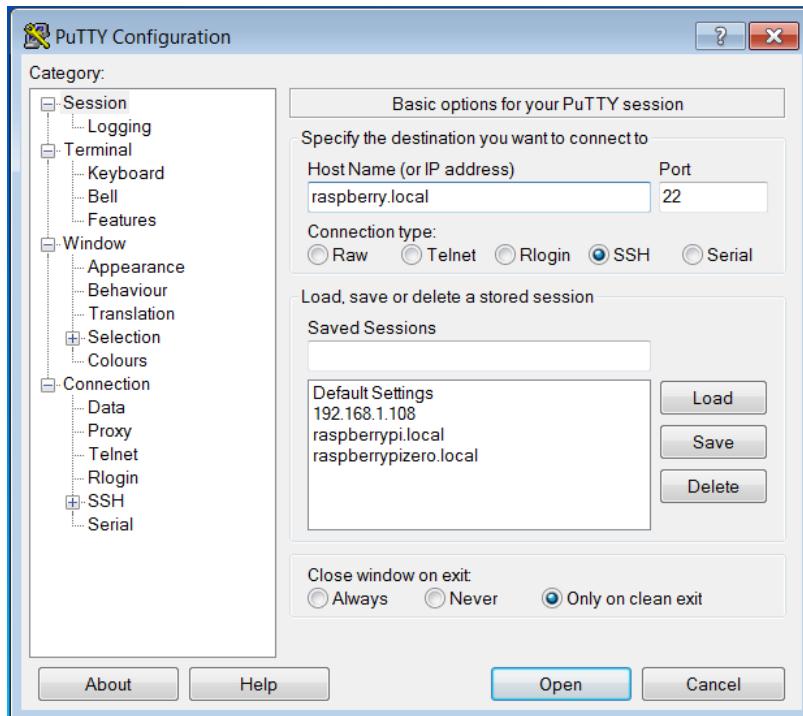
ถ้าต้องการทราบข้อมูลเวอร์ชันคอร์เนลของระบบปฏิบัติการ Linux ให้ใช้คำสั่ง

```
hostnamectl
```

```
pi@raspberrypi:~ $ hostnamectl
  Static hostname: raspberrypi
    Icon name: computer
      Machine ID: 989c759ce07a4fd39dae7f75ef824048
          Boot ID: f3916c5cc962479cbfa4d4ff2eb2b269
        Operating System: Raspbian GNU/Linux 11 (bullseye)
              Kernel: Linux 6.1.21-v8+
            Architecture: arm64
pi@raspberrypi:~ $
```

SSH (Secure Shell) คือ Network Protocol ที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์บนระบบเครือข่ายผ่าน Port หมายเลข 22 โดยมีระบบปฏิบัติการ Linux เป็นพื้นฐานในการเข้าถึงบัญชีผู้ใช้ (Shell Accounts) การทำงานจะอยู่ในลักษณะ ไคลเอนต์และเซิร์ฟเวอร์ (Client-Server) โดยประกอบด้วยโปรแกรม 2 ส่วน ส่วนแรกทำหน้าที่เป็นเครื่องที่ให้บริการ (Server) อำนวยความสะดวกแก่ผู้ใช้งานจะถูกติดตั้งลงที่เครื่อง Raspberry Pi และส่วนต่อมาจะทำหน้าที่เป็นผู้เชื่อมต่อ (Client) ไปยังเครื่องที่ให้บริการ SSH เช่น เครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Windows หรือ macOS และเครื่องโทรศัพท์ที่ใช้ระบบปฏิบัติการ Android หรือ iOS เพื่อให้ผู้ใช้งานสามารถเข้าควบคุมหรือสั่งงานเครื่องที่ให้บริการ โดยจะต้องมีการพิสูจน์ตัวตนก่อนใช้งานด้วยการล็อกอิน (Login) ใช้ชื่อผู้ใช้และรหัสผ่าน การส่งข้อมูลจะอยู่ในรูปแบบด้วยกันที่มีการเข้ารหัสข้อมูล (Encryption) สำหรับเครื่องคอมพิวเตอร์ Mac จะมี SSH ฝังอยู่ใน Terminal สามารถเรียกใช้งานเลย แต่ในเครื่องที่ใช้ Windows จะต้องติดตั้งโปรแกรม PuTTY: a free SSH and Telnet client เพื่อใช้เป็น Terminal ในการติดตั้งโปรแกรมต่างๆ โดยดาวน์โหลดได้จาก <https://www.chiark.greenend.org.uk/~sgtatham/putty/>

แล้วเรียกใช้โปรแกรม Putty โดยสามารถ Connect ผ่าน SSH ไม่ต้องใช้อินเทอร์เฟซของ Raspberry Pi ที่ชื่อ Hostname ให้ป้อนเป็นหมายเลข IP Address ของ raspberrypi แต่ถ้าไม่มีรู้หมายเลข IP ให้ใส่เป็น raspberrypi.local หลังจากนั้นให้เลือกหมายเลข Port เป็น 22 เลือกปุ่ม Connection type เป็น SSH ซึ่งจะสามารถบันทึกการตั้งค่าไว้ใช้งานในครั้งต่อไปโดยการกด Saved เสร็จแล้วกดปุ่ม Open เพื่อเชื่อมต่อไปยัง Raspberry Pi



หลังจากนั้นหน้าจอจะขึ้นให้ใส่ login as: ให้ใส่ pi และ password: ให้ใส่ raspberry

```

login as: pi
pi@192.168.1.112's password:
Linux raspberrypi 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

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: Sun Aug 30 18:36:12 2020 from 192.168.1.35

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

A VNC server is already running as :1
pi@raspberrypi:~$ 
```

กรณีใช้บอร์ด Odroid ที่เป็นระบบปฏิบัติการ Ubuntu ให้ใส่ root และ password: ให้ใส่ odroid

ถ้าในระบบปฏิบัติการ Ubuntu ยังไม่เปิดใช้งานเซิร์ฟเวอร์ SSH เช่น Ubuntu บน Desktop จะต้องติดตั้งเซิร์ฟเวอร์ OpenSSH และเปิดใช้งานการเขื่อมต่อ SSH โดยการเปิดเทอร์มินัลบนระบบ Ubuntu และเรียกใช้คำสั่งเพื่อติดตั้งเซิร์ฟเวอร์ OpenSSH ดังนี้

```

sudo apt update
sudo apt install openssh-server 
```

เมื่อการติดตั้งแพ็คเกจเสร็จ ให้ทดสอบตรวจสอบสถานะการเปิดใช้งานด้วยคำสั่ง

```
sudo systemctl status ssh
```

```
root@odroid:~# sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2022-10-17 19:32:11 +07; 6h ago
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 730 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 780 (sshd)
   Memory: 2.7M
      CGroup: /system.slice/ssh.service
              └─780 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 17 19:32:09 odroid systemd[1]: Starting OpenBSD Secure Shell server...
Oct 17 19:32:11 odroid sshd[780]: Server listening on 0.0.0.0 port 22.
Oct 17 19:32:11 odroid sshd[780]: Server listening on :: port 22.
Oct 17 19:32:11 odroid systemd[1]: Started OpenBSD Secure Shell server.
```

ถ้าในระบบ Ubuntu มีการกำหนดค่าเริ่มต้นใช้ UFW firewall จะต้องอนุญาตให้ใช้ SSH พอร์ต 22 สำหรับผู้ใช้ระบบ ประกอบภายนอกด้วยคำสั่ง

```
sudo ufw allow 22/tcp
```

ถ้าต้องการใช้ ssh ไป login เข้าเครื่องอื่น หรือจากเครื่องหนึ่งไปเข้าอีกเครื่องหนึ่ง ตัวอย่างจาก IP 192.168.1.63 (kali) ไปเข้า IP 192.168.1.21 (pi) สามารถทำได้โดยกำหนดชื่อและหมายเลข IP และหลังจากนั้นให้ใส่ password โดยสามารถทำทุกอย่างได้ตามต้องการเหมือนเครื่องตัวเอง ด้วยคำสั่ง

```
ssh pi@192.168.1.21
```

```
pi@cm41: ~
[pi@cm41 ~]$ login as: kali
[pi@cm41 ~]$ kali@192.168.1.63's password:
Linux kali 5.15.44-Re4son-v71+ #1 SMP Debian kali-pi (2022-07-03) armv7l

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun 25 19:00:37 2023 from 192.168.1.64
[kali㉿kali ~]$ 
[kali㉿kali ~]$ ssh pi@192.168.1.21
The authenticity of host '192.168.1.21 (192.168.1.21)' can't be established.
ED25519 key fingerprint is SHA256:7H2bD1Mowx6IQt2IK4eBSAUQHcgj29yWKPhMTHhEGQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.21' (ED25519) to the list of known hosts.
pi@192.168.1.21's password:
Linux cm41 6.1.21-v8+ #1642 SMP PREEMPT Mon Apr  3 17:24:16 BST 2023 aarch64

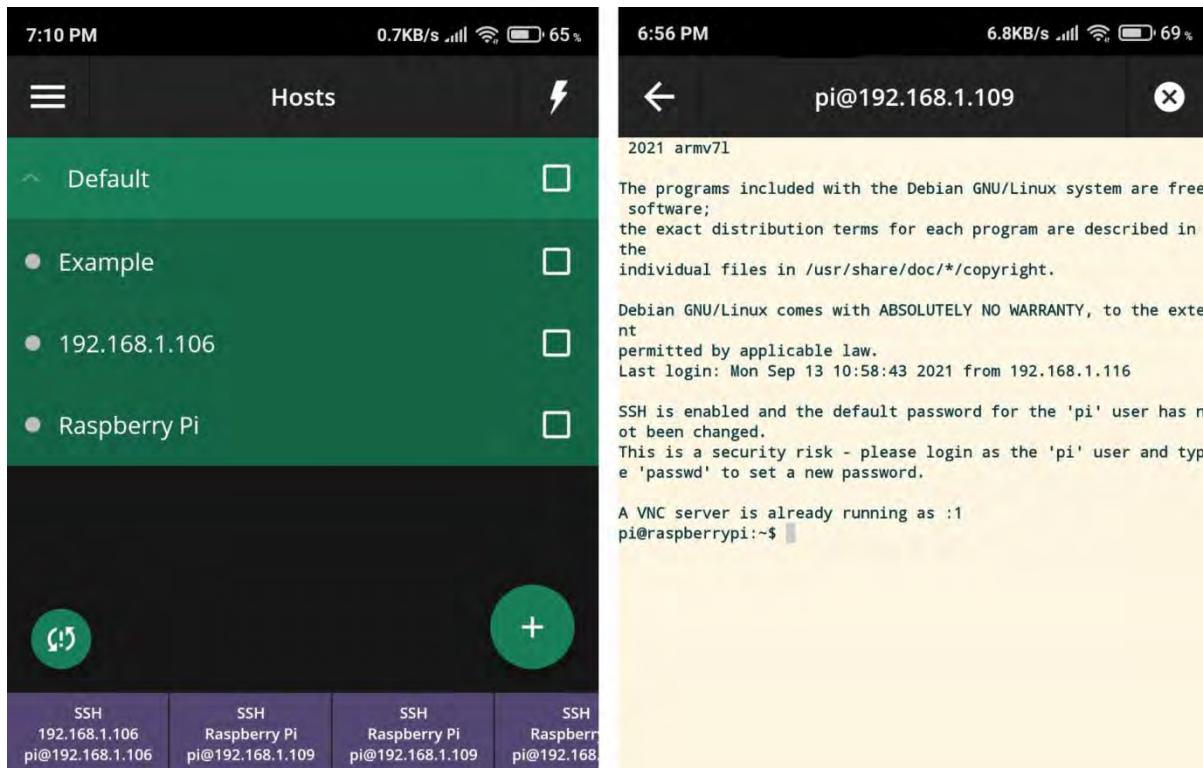
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun 25 00:17:10 2023
pi@cm41: ~ $
```

กรณีที่จะใช้ SSH ในโทรศัพท์ Android สามารถดาวน์โหลดโปรแกรมเพื่อติดตั้งได้ที่

<https://play.google.com/store/apps/details?id=com.arpaplus.adminhands&hl=th&gl=US>

ตัวอย่างโปรแกรมที่แสดงบนโทรศัพท์ Android ซึ่งจะสามารถทำงานได้เมื่อนำเครื่องคอมพิวเตอร์



คำสั่งที่ใช้ตรวจสอบข้อมูลพื้นฐานของอุปกรณ์ต่างๆ ทางด้าน Hardware บนระบบ Linux ดังนี้

lscpu - display information about the CPU architecture เป็นคำสั่งแสดงข้อมูลของ CPU

```
pi@pi4box:~$ lscpu
Architecture:          armv7l
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:  0-3
Thread(s) per core:   1
Core(s) per socket:   4
Socket(s):             1
Vendor ID:             ARM
Model:                 3
Model name:            Cortex-A72
Stepping:               r0p3
CPU max MHz:           1800.0000
CPU min MHz:           600.0000
BogoMIPS:              108.00
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf:     Not affected
Vulnerability Mds:     Not affected
Vulnerability Meltdown: Not affected
Vulnerability Mmio stale data: Not affected
Vulnerability Retbleed: Not affected
Vulnerability Spec store bypass: Not affected
Vulnerability Spectre v1: Mitigation: __user pointer sanitization
Vulnerability Spectre v2: Vulnerable
Vulnerability Srbds:    Not affected
Vulnerability Tsx async abort: Not affected
Flags:                  half thumb fastmult vfp edsp neon vfpv3 tls vfpv4
                           idiva idivt vfpd32 lpae evtstrm crc32
pi@pi4box:~$
```

free - display amount of free and used memory in the system แสดงหน่วยความจำที่ถูกใช้ในระบบ

```
pi@pi4box:~$ free
total        used        free      shared  buff/cache   available
Mem:    3931052     261228      62956      57792    3606868     3473020
Swap:   102396        1024     101372
pi@pi4box:~$
```

lsblk - list block devices แสดงข้อมูลของอุปกรณ์เก็บข้อมูลในระบบ

```
pi@pi4box:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0  1.8T  0 disk 
└─sda1   8:1    0  1.8T  0 part /media/pi/usb
mmcblk0  179:0  0 476.7G  0 disk 
├─mmcblk0p1 179:1  0  256M  0 part /boot
└─mmcblk0p2 179:2  0 476.5G  0 part /
pi@pi4box:~$
```

df - report file system space usage แสดง partitions ของ disk ที่ถูก mount ในระบบ

```
pi@pi4box:~$ df
Filesystem      1K-blocks      Used  Available Use% Mounted on
/dev/root        491692168  28182456  443426600  6% /
devtmpfs          1800660         0  1800660  0% /dev
tmpfs            1965524         0  1965524  0% /dev/shm
tmpfs            786212         2808  783404  1% /run
tmpfs             5120           4   5116  1% /run/lock
/dev/mmcblk0p1    258095       50792  207304  20% /boot
tmpfs            393104         28  393076  1% /run/user/1000
/dev/sdal        1921742316  625185520 1198864740  35% /media/pi/usb
pi@pi4box:~$
```

mount - mount a filesystem เป็นการเชื่อมโยงระบบ file system กับสื่อหรืออุปกรณ์จัดเก็บภายนอก

```
pi@pi4box:~$ mount
/dev/mmcblk0p2 on / type ext4 (rw,noatime)
devtmpfs on /dev type devtmpfs (rw,relatime,size=1800660k,nr_inodes=84203,mode=755)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,size=786212k,nr_inodes=819200,mode=755)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/binfmt_misc type autofs (rw,relatime,fd=29,pgpr=1,timeo=0,minproto=5,maxproto=5,direct)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
sunrpc on /run/rpc_pipefs type rpc_pipefs (rw,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
/dev/mmcblk0p1 on /boot type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed,flush,errors=remount-ro)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=393104k,nr_inodes=98276,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sdal on /media/pi/usb type ext4 (rw,nosuid,nodev,relatime,errors=remount-ro,uelper=udisks2)
pi@pi4box:~$
```

lsusb - list USB devices แสดงข้อมูล USB controllers และอุปกรณ์ที่เชื่อมต่อ

```
pi@pi4box:~$ lsusb
Bus 002 Device 002: ID 0bc2:ab26 Seagate RSS LLC Backup Plus Slim Portable Drive 1 TB
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 007: ID 0c5b:02d1 Hengchangtong HCT USB Entry Keyboard
Bus 001 Device 008: ID 045e:0737 Microsoft Corp. Compact Optical Mouse 500
Bus 001 Device 006: ID 0bda:c811 Realtek Semiconductor Corp. 802.11ac NIC
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@pi4box:~$
```

getconf LONG\_BIT แสดงข้อมูลเครื่องในแบบ 32 บิตหรือ 64 บิต

### โครงสร้างของระบบไฟล์ Linux (Filesystem Hierarchy Standard)

- / ไดเรกทอรี่ 'root' ที่มีไฟล์และไดเรกทอรีอื่น ๆ ทั้งหมด
- /bin (binaries) เป็นไฟล์ปฏิบัติการใช้โดยระบบปฏิบัติการ (OS) มีโปรแกรมและคำสั่งทั่วไปที่ใช้ร่วมกันสำหรับผู้ใช้ทั้งหมด
- /boot (boot loader) เป็นที่เก็บไฟล์ที่จำเป็นสำหรับการ boot ให้สำเร็จในระหว่างกระบวนการเริ่มต้นของระบบ
- /dev (device) ที่เก็บไฟล์พิเศษและไฟล์อุปกรณ์ เช่น /dev/disk0, /dev/sda1, /dev/tty
- /etc (etcetera) เป็นที่เก็บไฟล์การตั้งค่าคอมพิวเตอร์ที่ควบคุมการทำงานของระบบสำหรับ Host
- /home สำหรับเก็บไดเรกทอรีของผู้ใช้แต่ละราย เพื่อจัดเก็บไฟล์ที่บันทึกไว้ในการตั้งค่าส่วนบุคคล
- /lib (libraries) เป็นที่เก็บรหัสไลบรารีที่ใช้ร่วมกันและโมดูลเครื่องเนต ที่ใช้โดยระบบไฟล์และสำหรับการ Boot ซึ่งอาจใช้โดย

#### ไฟล์ปฏิบัติการใน bin

- /media เป็นที่เก็บไดเรกทอรีย่อยที่ได้จากการ Mount เพื่อติดตั้งอุปกรณ์แบบชั่วคราวออดได้ (Removable Media)
- /mnt (mount point) จุดมาท์ต์ระบบไฟล์ที่ติดตั้งชั่วคราวสำหรับผู้คุ้นเคยระบบ
- /opt (optional add-on packages) ที่ติดตั้งเพิ่มจากซอฟต์แวร์แอพพลิเคชันเสริม
- /proc (processes) เป็นระบบไฟล์เสมือนจากการจัดเตรียมกระบวนการและข้อมูลเครื่องเนต โดยทั่วไประบบจะสร้างข้อมูลโดยอัตโนมัติ

- /root ไดเรกทอรีหลักของผู้ดูแลระบบสำหรับผู้ใช้ Root

• /run ที่เก็บข้อมูลตัวแปร Runtime เป็นข้อมูลเกี่ยวกับระบบที่ทำงานตั้งแต่การ启动รีสูดท้ายการที่ผู้ใช้ที่ล็อกอิน ไฟล์ที่อยู่ในไดเรกทอรีนี้ต้องจะถูกลบออกเมื่อเริ่มต้นกระบวนการบูต

- /sbin เป็นที่จัดเก็บ super-binaries ของระบบ คำสั่งที่ผู้ดูแลระบบใช้

• /srv (serve) ไดเรกทอรีนี้มีไว้สำหรับไฟล์ข้อมูลแบบสแตติก สำหรับที่ให้บริการเฉพาะระบบ เช่น ข้อมูลและสคริปต์สำหรับเว็บไซต์ฟอร์ม ข้อมูลที่นำเสนอโดยชิร์ฟเวอร์ FTP

- /sys เป็นข้อมูลเกี่ยวกับอุปกรณ์ไดรเวอร์และคุณสมบัติบางอย่างของเครื่องเนต

- /tmp (temporary) เป็นที่จัดเก็บไฟล์หรือข้อมูลชั่วคราว จะถูกลบออกจากระหว่างการรีบูตระบบ

• /usr (unix system resources) เป็นไดเรกทอรีสำหรับจัดเก็บและแบ่งปันโปรแกรมและข้อมูลของผู้ใช้ มีชื่อพิเศษและแอพพลิเคชันสำหรับผู้ใช้ ไว้เพื่อระหว่างระหว่างโสสต์ และสามารถติดตั้ง NFS กับหลายโสสต์ได้อย่างปลอดภัย

• /var (variable) เป็นไฟล์ข้อมูลตัวแปรที่เนื้อหาจะเปลี่ยนแปลงอย่างต่อเนื่องระหว่างการทำงานปกติของระบบ เช่น logs, spool files และ temporary e-mail files

#### คำสั่ง File/Directory

ls - list directory contents แสดงรายชื่อไฟล์และไดเรกทอรี

pwd - print name of current/working directory แสดงชื่อของไดเรกทอรี

cp - copy files and directories สำเนาไฟล์

mv - move (rename) files เปลี่ยนชื่อไฟล์

rm - remove files or directories ลบไฟล์

cd - change directory ข้ามไปยังไดเรกทอรีที่ต้องการ

mkdir - make directories สร้างไดเรกทอรีใหม่

rmdir - remove empty directories ลบไดเรกทอรี

ถ้าต้องการข้อมูลรูปแบบการใช้งานเพิ่มเติมให้ต่อท้ายคำสั่งด้วย --help

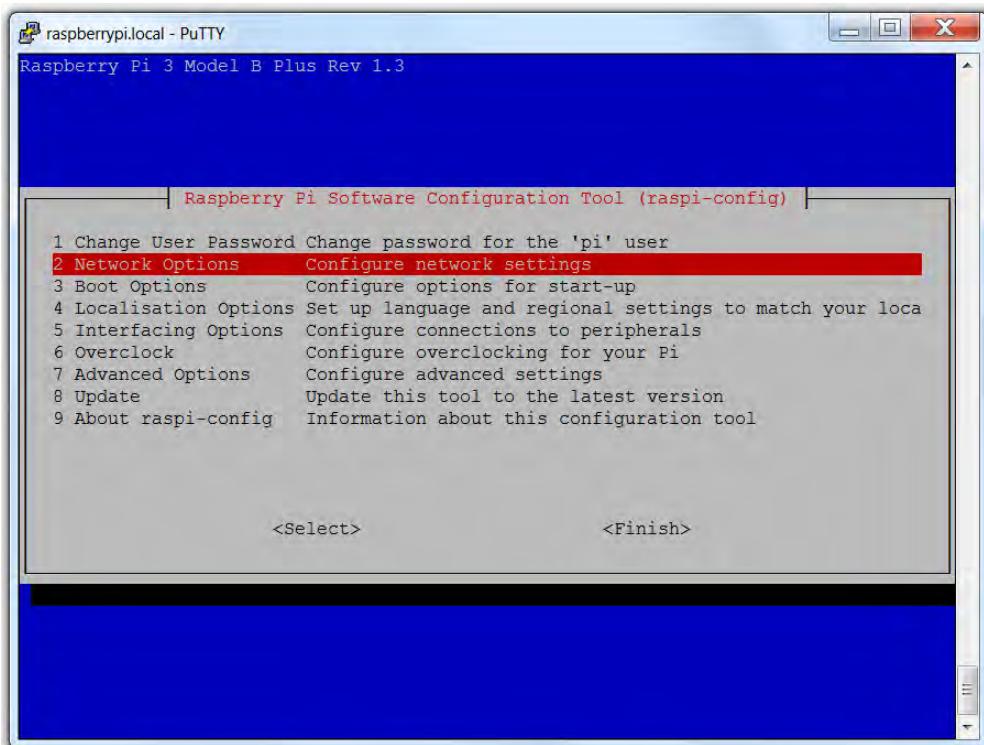
ถ้าต้องการข้อมูลเป็นคู่มือรายละเอียดการใช้งานของแต่ละคำสั่งให้เพิ่มค้านหน้าคำสั่งด้วย man

เมื่อล็อกอิน Raspberry Pi 3 B+ ได้แล้ว ให้ทำการเชื่อมต่อ WiFi แทน เพื่อให้สามารถอุดสาย LAN ออกได้ โดยรีมจากการสแกน WiFi ด้วยคำสั่ง sudo iwlist wlan0 scan แล้วรอให้เครื่อง Raspberry Pi ทำการสแกนหา WiFi ซึ่งจะแสดงรายละเอียดของ WiFi ที่ค้นหาเจอ

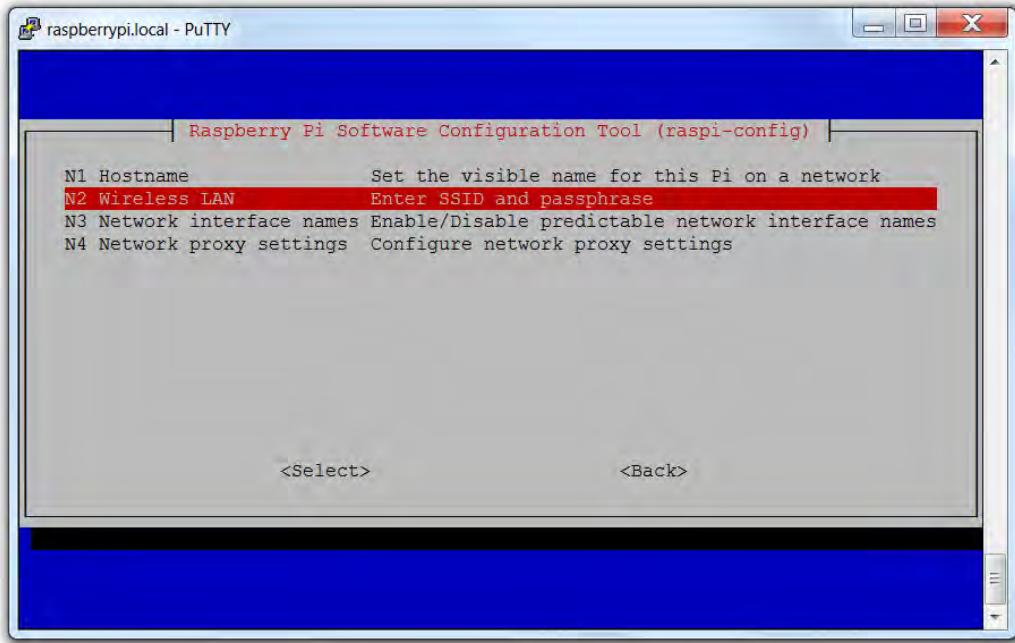
โดยที่ sudo คือคำสั่งที่อนุญาตให้รันคำสั่งที่ผู้ใช้เป็น superuser (sudo) ซึ่งจะเป็นการใช้สิทธิ์สูงสุดของผู้ใช้แบบ Root ถ้าเปรียบเทียบกับระบบปฏิบัติการ Windows ก็คือ Administrator

```
pi@raspberrypi:~$ sudo iwlist wlan0 scan
wlan0      Scan completed :
           Cell 01 - Address: AC:B3:B5:76:F4:48
                     Channel:10
                     Frequency:2.457 GHz (Channel 10)
                     Quality=53/70  Signal level=-57 dBm
                     Encryption key:on
                     ESSID:"TrueGigatexFiber_2.4G"
                     Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
                                   11 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: 60ms ago
                     IE: IEEE 802.11i/WPA2 Version 1
                         Group Cipher : TKIP
                         Pairwise Ciphers (2) : CCMP TKIP
                         Authentication Suites (1) : PSK
                     IE: WPA Version 1
                         Group Cipher : TKIP
                         Pairwise Ciphers (2) : CCMP TKIP
                         Authentication Suites (1) : PSK
```

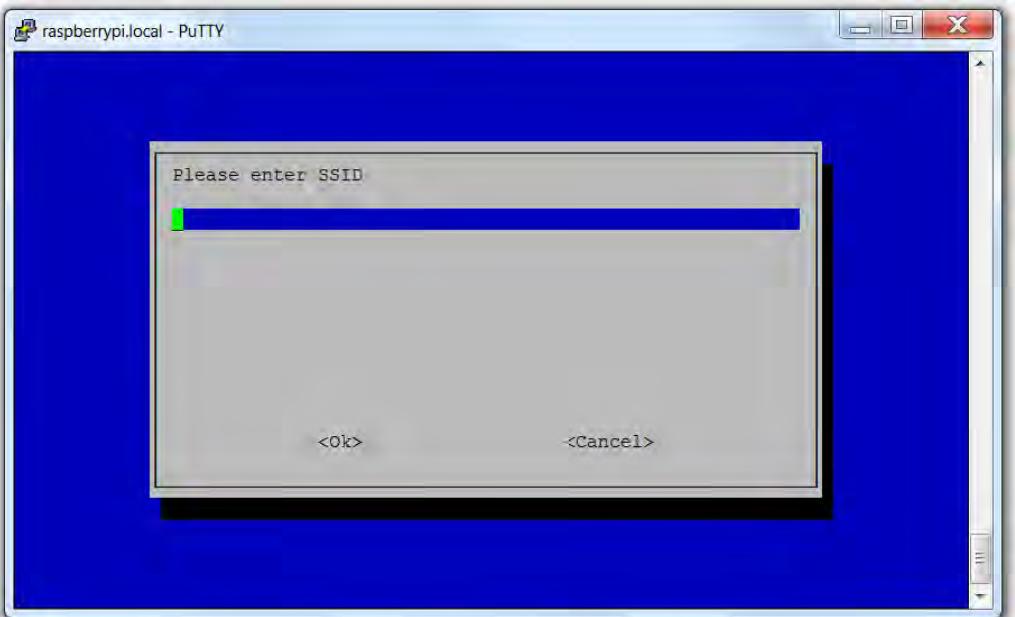
การตั้งค่าชื่อ WiFi ที่ต้องการให้บอร์ดไปเชื่อมต่อ โดยใช้คำสั่งที่แสดง Menu สำหรับกำหนดค่าเริ่มต้นต่างๆ สามารถทำได้โดยพิมพ์คำสั่ง sudo raspi-config จะปรากฏ Menu ดังภาพ



ใช้ปุ่มลูกศรเลื่อนขึ้นและลงเพื่อเลือกเมนู และใช้ปุ่มลูกศรซ้ายขวาเพื่อเลือก Select / Finish โดยให้เลื่อนไปที่ Network Options กดปุ่ม Enter จะเจอน้ากดไป แล้วเลื่อนไปที่ N2 Wireless LAN กดปุ่ม Enter

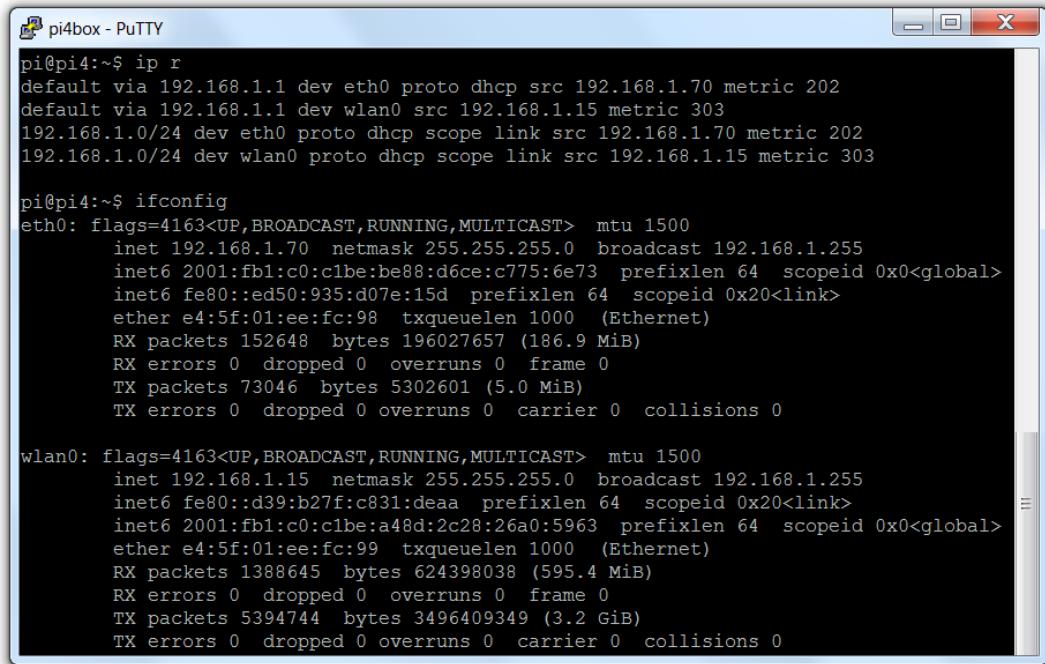


กรอกชื่อ WiFi แล้วกดปุ่ม Enter ให้กรอกรหัสผ่านของ WiFi กดปุ่ม Enter โปรแกรมจะกลับมาหน้าเมนูให้เลื่อนปุ่มลูกศรไปที่ Finish แล้วกด Enter



พิมพ์คำสั่ง `wpa_cli -i wlan0 reconfigure` เพื่ออัพเดทการตั้งค่า WiFi จะขึ้นคำว่า OK

พิมพ์คำสั่ง `ip r` หรือ `ifconfig` แล้วให้ดูที่ wlan0 หากขึ้นหมายเลข IP แสดงว่าสามารถเชื่อมต่อ WiFi ได้แล้ว



```

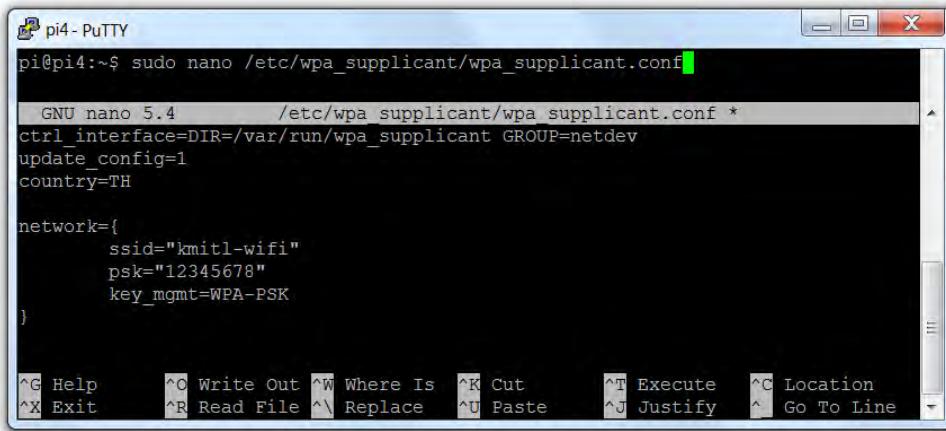
pi@pi4:~$ ip r
default via 192.168.1.1 dev eth0 proto dhcp src 192.168.1.70 metric 202
default via 192.168.1.1 dev wlan0 src 192.168.1.15 metric 303
192.168.1.0/24 dev eth0 proto dhcp scope link src 192.168.1.70 metric 202
192.168.1.0/24 dev wlan0 proto dhcp scope link src 192.168.1.15 metric 303

pi@pi4:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.70 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 2001:fb1:c0:c1be:be88:d6ce:c775:6e73 prefixlen 64 scopeid 0x0<global>
        inet6 fe80::ed50:935:d07e:15d prefixlen 64 scopeid 0x20<link>
          ether e4:5f:01:ee:fc:98 txqueuelen 1000 (Ethernet)
          RX packets 152648 bytes 196027657 (186.9 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 73046 bytes 5302601 (5.0 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.15 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::d39:b27f:c831:deaa prefixlen 64 scopeid 0x20<link>
        inet6 2001:fb1:c0:c1be:a48d:2c28:26a0:5963 prefixlen 64 scopeid 0x0<global>
          ether e4:5f:01:ee:fc:99 txqueuelen 1000 (Ethernet)
          RX packets 1388645 bytes 624398038 (595.4 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 5394744 bytes 3496409349 (3.2 GiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  
```

บังນือวิธีหนึ่งในการเชื่อมต่อ WiFi ด้วยการแก้ไขข้อมูลของไฟล์ที่เก็บค่า SSID โดยตรง ให้พิมพ์คำสั่ง  
`sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`

โดยที่ nano จะเป็น Text Editor และใช้ทำการแก้ไขข้อมูลในไฟล์ ด้วยการเปลี่ยน SSID และ Password ให้ตรงกับ SSID ของ WiFi ที่ใช้งาน เตรียมแล้วให้ทำการบันทึกไฟล์



```

pi@pi4:~$ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
GNU nano 5.4           /etc/wpa_supplicant/wpa_supplicant.conf *
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=TH

network={
    ssid="kmitl-wifi"
    psk="12345678"
    key_mgmt=WPA-PSK
}

  
```

กรณีใช้บอร์ด Odroid จะต้องเชื่อมต่อโดยใช้ USB WiFi และใช้คำสั่ง iwconfig เพื่อตรวจสอบ Wireless Adapter ที่ไม่พบ WiFi จะต้องติดตั้งโปรแกรมเพิ่ม เพื่อให้เครื่องสามารถเข้าถึงชาร์ดแวร์ที่เชื่อมต่อกับอุปกรณ์ในระบบได้ ตัวอย่างเป็นการติดตั้งไดรเวอร์ของอุปกรณ์ Realtek RTL8811CU บนบอร์ด Raspberry Pi ด้วยคำสั่งดังนี้

```

sudo apt update
sudo apt install build-essential bc dkms git
sudo apt install raspberrypi-kernel-headers
git clone https://github.com/morrownr/8821cu-20210916.git
cd 8821cu-20210916
make
sudo make install
  
```

กรณีใช้ USB WiFi เป็นรุ่นอื่น เช่น Realtek RTL8812AU ให้แก้ไข 8821cu-20210916 ไปเป็น 8812au-20210629

โคร์แนลที่เป็น Linux 6.1.x-v8+ ในระบบปฏิบัติการ 32 บิต จะให้คำเริ่มต้นการทำงานเป็นโคร์แนล 64 บิต ซึ่งอาจทำให้เกิดปัญหาในการติดตั้งซอฟต์แวร์บางตัว เช่น เมื่อใช้คำสั่ง make จะเกิด Error ขึ้นว่า No such file or directory. Stop. เพื่อแก้ปัญหานี้ให้เพิ่มคำสั่ง arm\_64bit=0 ลงในไฟล์ /boot/config.txt

เราสามารถตรวจสอบรายการโมดูลโคร์แนลที่โหลดทั้งหมด (List Kernel Modules) ด้วยคำสั่ง lsmod

Module	Size	Used by
rfcomm	49152	12
snd_seq_dummy	16384	0
snd_hrtimer	16384	1
snd_seq	69632	7 snd_seq_dummy
cmac	16384	3
algif_hash	16384	1
aes_arm_bs	24576	2
crypto_simd	16384	1 aes_arm_bs
cryptd	24576	2 crypto_simd
algif_skcipher	16384	1
af_alg	28672	6 algif_hash, algif_skcipher
bnef	20480	2
hci_uart	40960	1
btbcm	20480	1 hci_uart
bluetooth	434176	39 hci_uart, bnef, btbcm, rfcomm
ecdh_generic	16384	2 bluetooth
ecc	40960	1 ecdh_generic
8821cu	2400256	0
hid_logitech_hidpp	45056	0
joydev	20480	0
sg	28672	0
snd_usb_audio	262144	0
snd_hwdep	16384	1 snd_usb_audio
snd_usbmidi_lib	32768	1 snd_usb_audio
snd_rawmidi	36864	1 snd_usbmidi_lib
snd_seq_device	16384	2 snd_seq, snd_rawmidi
uvccvideo	102400	0
hid_logitech_dj	28672	0
binfmt_misc	20480	1
brcmfmac	339968	0
brcmutil	24576	1 brcmfmac
bcm2835_isp	32768	0
bcm2835_v412	49152	0
bcm2835_codec	45056	0
cfg80211	786432	2 brcmfmac, 8821cu
snd_bcm2835	28672	1
bcm2835_mmal_vchiq	36864	3 bcm2835_isp, bcm2835_codec, bcm2835_v412
rfkill	32768	8 bluetooth, cfg80211
v412_mem2mem	36864	1 bcm2835_codec
snd_pcm	122880	2 snd_usb_audio, snd_bcm2835
videobuf2_dma_contig	20480	2 bcm2835_isp, bcm2835_codec
videobuf2_vmalloc	16384	2 uvccvideo, bcm2835_v412
videobuf2_memops	16384	2 videobuf2_dma_contig, videobuf2_vmalloc
videobuf2_v412	32768	5 bcm2835_isp, uvccvideo, bcm2835_codec, bcm2835_v412, v412_mem2mem
videobuf2_common	65536	9 bcm2835_isp, uvccvideo, bcm2835_codec, videobuf2_dma_contig, videobuf2_vmalloc, videobuf2_memops, bcm2835_v412, v412_mem2mem, videobuf2_v412
snd_timer	36864	3 snd_seq, snd_hrtimer, snd_pcm
raspberrypi_hwmon	16384	0
snd	86016	13 snd_hwdep, snd_seq, snd_usb_audio, snd_timer, snd_rawmidi, snd_usbmidi_lib, snd_seq_device, snd_bcm2835, snd_pcm
videodev	249856	7 bcm2835_isp, uvccvideo, bcm2835_codec, videobuf2_common, bcm2835_v412, v412_mem2mem, videobuf2_v412
mc	49152	8 bcm2835_isp, uvccvideo, bcm2835_codec, snd_usb_audio, videobuf2_common, videodev, v412_mem2mem, videobuf2_v412
vc_sm_cma	32768	2 bcm2835_isp, bcm2835_mmal_vchiq
rpiivid_mem	16384	0
nvmem_rmem	16384	0
uio_pdrv_genirq	16384	0
uio	20480	1 uio_pdrv_genirq
drm	544768	0
fuse	135168	3
dm_mod	131072	0
drm_panel_orientation_quirks	16384	1 drm
ip_tables	28672	0
x_tables	32768	1 ip_tables
ipv6	528384	49

ตัวอย่าง หลังจากเพิ่ม Wireless Adapter เมื่อใช้คำสั่ง ifconfig จะเห็นได้ว่ามี wlan0 เซ็ตต่อที่ IP 192.168.1.11 และ wlan1 เซ็ตต่อที่ IP 192.168.1.72

```
kali㉿kali: ~
└─[kali㉿kali)-[~]
  $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.70 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 2001:fb1:c0:b239:e65f:1ff:feee:fca7 prefixlen 64 scopeid 0x0<global>
        inet6 fe80::e65f:1ff:feee:fca7 prefixlen 64 scopeid 0x20<link>
            ether e4:5f:01:ee:fca7 txqueuelen 1000 (Ethernet)
            RX packets 55663 bytes 3806687 (3.6 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 50307 bytes 42223972 (40.2 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 84 bytes 6720 (6.5 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 84 bytes 6720 (6.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.11 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 2001:fb1:c0:b239:5a32:3584:16af:138b prefixlen 64 scopeid 0x0<global>
        inet6 fe80::2311:c570:935f:919 prefixlen 64 scopeid 0x20<link>
            ether e4:5f:01:ee:fca8 txqueuelen 1000 (Ethernet)
            RX packets 18906 bytes 4905128 (4.6 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 264 bytes 58911 (57.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.72 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::70a4:f3be:e05c:a56d prefixlen 64 scopeid 0x20<link>
        inet6 2001:fb1:c0:b239:9859:d4f4:5d:ef6f prefixlen 64 scopeid 0x0<global>
            ether 90:de:80:92:6a:5b txqueuelen 1000 (Ethernet)
            RX packets 3667 bytes 649835 (634.6 KiB)
            RX errors 0 dropped 83 overruns 0 frame 0
            TX packets 274 bytes 21058 (20.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

เมื่อใช้คำสั่ง iwconfig จะเห็นได้ว่าอเดปต์เตอร์ wlan ถูกตั้งค่า Mode เป็น Managed ซึ่งเป็นค่าเริ่มต้นของอุปกรณ์ WiFi ทั้งหมด ทำให้อุปกรณ์นี้จะจับเฉพาะแพ็กเก็ตที่มีค่า MAC Address เป็นของอุปกรณ์นี้เท่านั้น

```
kali㉿kali: ~
└─[kali㉿kali)-[~]
  $ iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   IEEE 802.11  ESSID:"WIFI"
        Mode:Managed  Frequency:2.467 GHz  Access Point: E0:0E:E4:0A:62:B5
        Bit Rate=24 Mb/s  Tx-Power=31 dBm
        Retry short limit:7  RTS thr:off  Fragment thr:off
        Power Management:on
        Link Quality=70/70  Signal level=-32 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:2  Invalid misc:0  Missed beacon:0

wlan1   IEEE 802.11AC  ESSID:"WIFI5"  Nickname:<WIFI@REALTEK>
        Mode:Managed  Frequency:5.805 GHz  Access Point: E0:0E:E4:0A:62:B1
        Bit Rate:434 Mb/s  Sensitivity:0/0
        Retry:off  RTS thr:off  Fragment thr:off
        Power Management:off
        Link Quality=90/100  Signal level=59/100  Noise level=0/100
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

ถ้าต้องการให้สามารถดักจับแพ็กเก็ตข้อมูลการสื่อสารทั้งหมดของสัญญาณ WiFi ที่อยู่ในขอบเขตเครือข่ายไร้สาย ลิ้งแม่บ้านแพ็กเก็ตนั้นจะถูกส่งไปยังเราเตอร์หรือถูกตั้งค่าไปยังอุปกรณ์อื่นที่ไม่ใช่คอมพิวเตอร์ของเรา โดยที่เครื่องของเราจะไม่ได้เข้ามาร่วมต่อภัยเครือข่ายเป้าหมายและแม่จะไม่รู้รหัสผ่านไปยังเครือข่ายเป้าหมาย เพื่อวิเคราะห์ข้อมูลหรือแม้แต่จะเข้าไปในเครือข่ายได้อย่างไร จะต้องตั้งค่าเปลี่ยนโหมดการทำงานของอินเทอร์เฟซไร้สายให้ทำงานในโหมด Monitor และก่อนที่จะเปลี่ยนโหมดของอินเทอร์เฟซจะต้องปิดการทำงานการเข้ามาร่วมต่ออินเทอร์เน็ตของมันเสียก่อน จากนั้นจึงจะสามารถเปิดใช้งานโหมด Monitor ได้ กรณีใช้ WiFi บนบอร์ด Raspberry Pi จะไม่รองรับการทำงานในโหมด Monitor จะต้องหา USB WiFi มาเสียบเพิ่มที่ช่อง USB เช่นตัวอย่างของ Realtek RTL8811CU จะได้อินเทอร์เฟซเป็น wlan1 ต้องใช้คำสั่งดังนี้

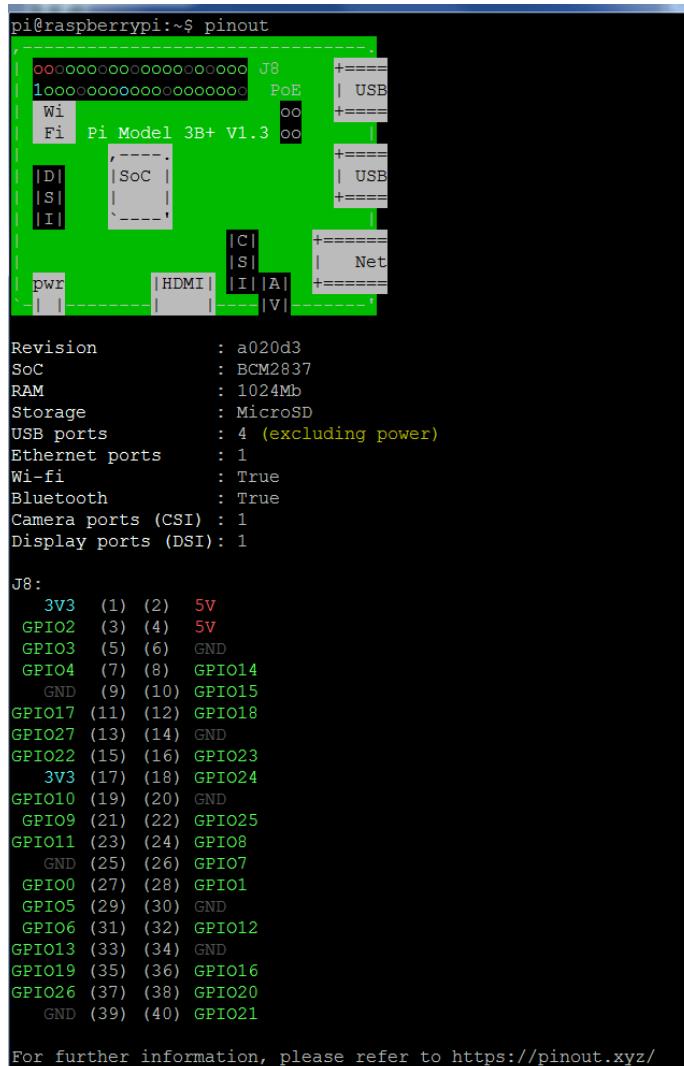
```
sudo ifconfig wlan1 down
sudo airmon-ng check kill
sudo iwconfig wlan1 mode monitor
sudo ifconfig wlan1 up
iwconfig
```

สำหรับชิปเซ็ต USB WiFi ที่มีโหมด Monitor และมีไครเวอร์ที่สนับสนุนระบบปฏิบัติการ Linux หรือมีไครเวอร์ในเโคर์เนล (Linux In-Kernel Driver) ตามมาตรฐาน Linux Wireless Standards (mac80211) ซึ่งจะมีอยู่ภายในเโคร์เนลของตั้งแต่รุ่นที่กำหนด ทำให้อะแดปเตอร์ที่ใช้ได้รับการคุ้มครองรักษาและปราศจากปัญหาในระยะยาว โดยมีชิปเซ็ตในปัจจุบันดังนี้

Chipset	Interface	Standard	MIMO	WiFi 2.4	WiFi 5	WiFi 6	Linux In-Kernel Driver	Monitor Mode
Mediatek MT7922u	USB3 / 5 Gbps	WiFi 6e	2x2	40	160	160	✓ 5.16+	✓
Mediatek MT7921au	USB3 / 5 Gbps	WiFi 6e	2x2	40	80	80	✓ 5.18+	✓
Mediatek MT7662u	USB3 / 5 Gbps	WiFi 5	2x2	40	80	N	✓ 5.9+	✓
Mediatek MT7612u	USB3 / 5 Gbps	WiFi 5	2x2	40	80	N	✓ 4.19+	✓
Mediatek MT7610u	USB2 / 480 Mbps	WiFi 5	1x1	20	80	N	✓ 4.19+	✓
Realtek RTL8822bu	USB2 / 480 Mbps	WiFi 5	2x2	40	80	N	✓ 6.2+	✓
Realtek RTL8812bu	USB3 / 5 Gbps	WiFi 5	2x2	40	80	N	✓ 6.2+	✓
Realtek RTL8812au	USB3 / 5 Gbps	WiFi 5	2x2	40	80	N	-	✓
Realtek RTL8821cu	USB2 / 480 Mbps	WiFi 5	1x1	40	80	N	✓ 6.2+	✓
Realtek RTL8811cu	USB2 / 480 Mbps	WiFi 5	1x1	40	80	N	✓ 6.2+	✓
Realtek RTL8821au	USB2 / 480 Mbps	WiFi 5	1x1	40	80	N	-	✓
Realtek RTL8811au	USB2 / 480 Mbps	WiFi 5	1x1	40	80	N	-	✓
Realtek RTL8192cu	USB2 / 480 Mbps	WiFi 2.4	2x2	40	N	N	✓ 2.6.33+	✓
Ralink RT3573	USB2 / 480 Mbps	WiFi 2.4	3x3	40	40	N	✓ 3.12+	✓
Ralink RT5572	USB2 / 480 Mbps	WiFi 2.4	2x2	40	40	N	✓ 3.10+	✓
Ralink RT3572	USB2 / 480 Mbps	WiFi 2.4	2x2	40	40	N	✓ 2.6.31+	✓
Ralink RT5372	USB2 / 480 Mbps	WiFi 2.4	2x2	40	N	N	✓ 3.0+	✓
Ralink RT5370	USB2 / 480 Mbps	WiFi 2.4	1x1	40	N	N	✓ 3.0+	✓
Ralink RT3070	USB2 / 480 Mbps	WiFi 2.4	1x1	40	N	N	✓ 2.6.31+	✓
Atheros AR9271	USB2 / 480 Mbps	WiFi 2.4	1x1	40	N	N	✓ 2.6.35+	✓

โดยที่มีอะแดปเตอร์ที่ใช้ชิปเซ็ตบางตัวของ Realtek มีโหมด Monitor แต่ก็ควรหลีกเลี่ยงเนื่องจากไครเวอร์ของ Linux นั้นไม่ได้มีปัญหา เช่น 8814au , 8832au , 8852au , 8832bu และ 8852bu

ถ้าต้องการคุ้มครองตำแหน่งขาต่างๆที่จะใช้งานบนบอร์ด Raspberry Pi สามารถทำได้ด้วยคำสั่ง pinout ดังนี้



เนื่องจาก Raspberry Pi มีการอัพเดทซอฟแวร์เพื่อแก้ไขแก้ไขบັນດາและเพิ่มฟีเจอร์ต่างๆอยู่เสมอ จึงควรทำการอัพเดทซอฟแวร์ต่าง ๆ ให้เป็นเวอร์ชันใหม่ ก่อนที่จะติดตั้งโปรแกรมเพิ่มเติมทุกครั้งในหน้าต่างของ PuTTY ให้พิมพ์คำสั่ง

`sudo apt-get update` อัพเดตฐานข้อมูลของแพ๊กเกจด้วยการตรวจสอบรายการที่เป็นโปรแกรมใหม่ๆ จาก Server ของระบบปฏิบัติการ

`sudo apt-get upgrade` ตรวจสอบแพ๊กเกจที่ติดตั้งไว้ทั้งหมดว่ามีแพ๊กเกจใดมีการอัพเดทบ้าง ต่อจากนั้นก็จะถามว่าต้องการดาวน์โหลดและติดตั้งหรือไม่

โดยที่ apt-get คือคำสั่งในการให้ติดตั้งแพ็คเกจแบบออนไลน์จากเซิร์ฟเวอร์ที่ให้บริการ โดยตรง

เพื่อที่จะให้ง่ายต่อการใช้งานโดยใช้ชีวิตชื่อนี้มต่อจากจะยกให้เห็นหน้าจอ Desktop ของ Raspberry Pi 3 B+ ด้วยสามารถทำได้โดยใช้โปรแกรม Remote Desktop Connection ของ Windows หรือจะใส่โปรแกรมเพิ่มเติมโดยใช้ VNC (Virtual Network Computing) สามารถใช้งานได้หลายโปรแกรม เช่น TightVNC , RealVNC หรือ UltraVNC ช่องทางการใช้งาน VNC Remote Desktop บนบอร์ด Raspberry Pi จะถูกกำหนดเป็น VNC Server ส่วนที่คอมพิวเตอร์ที่ใช้ Remote Desktop จะต้องกำหนดเป็น VNC Client เริ่มจากที่ Raspberry Pi ให้ใช้โปรแกรม PuTTY เพื่อทำการติดตั้งโปรแกรม VNC Server แล้วใส่ Username: pi และ Password: raspberry โดยพิมพ์คำสั่งดังนี้

```

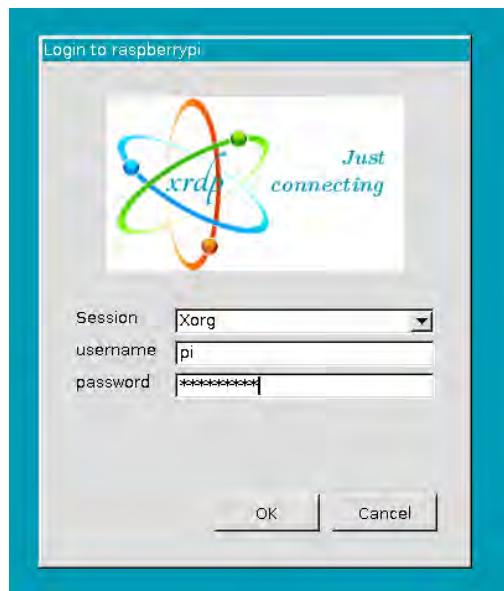
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install tightvncserver
sudo apt-get install xrdp

```

เมื่อติดตั้งเสร็จแล้ว ให้เรียกใช้โปรแกรม Remote Desktop Connection ของ Windows ใส่หมายเลข IP ของ Raspberry Pi หรือใส่เป็น raspberrypi.local



ใส่ Username: pi และ Password: raspberry จะสามารถเข้าไปที่หน้าจอ Desktop ของ Raspberry Pi ได้



โดยที่โปรแกรม Remote Desktop Connection ในการใช้งานจริงมีบางคำสั่งไม่สามารถทำงานได้ เช่น Reboot , Shutdown ดังนั้นเราสามารถแก้ปัญหาเหล่านี้ได้โดยการใส่โปรแกรมเพิ่มเติมในคอมพิวเตอร์ด้วยการใช้ Virtual Network Computing (VNC) เป็นเทคโนโลยีที่ช่วยในการควบคุมระยะไกลของคอมพิวเตอร์เครื่องอื่นโดยใช้โปรโตคอล Remote Frame Buffer (RFB) เพื่อที่จะใช้งานได้เหมือนหน้าจอของ Raspberry Pi 3 B+ ซึ่งสามารถดาวน์โหลดโปรแกรมเพื่อติดตั้งไว้ในเครื่อง

คอมพิวเตอร์ได้จาก <https://www.tightvnc.com/download.php> ส่วนในบอร์ด Raspberry Pi จะต้องทำการตั้งค่าการเรียกใช้ TightVNC Server โดยพิมพ์คำสั่ง

`tightvncserver`

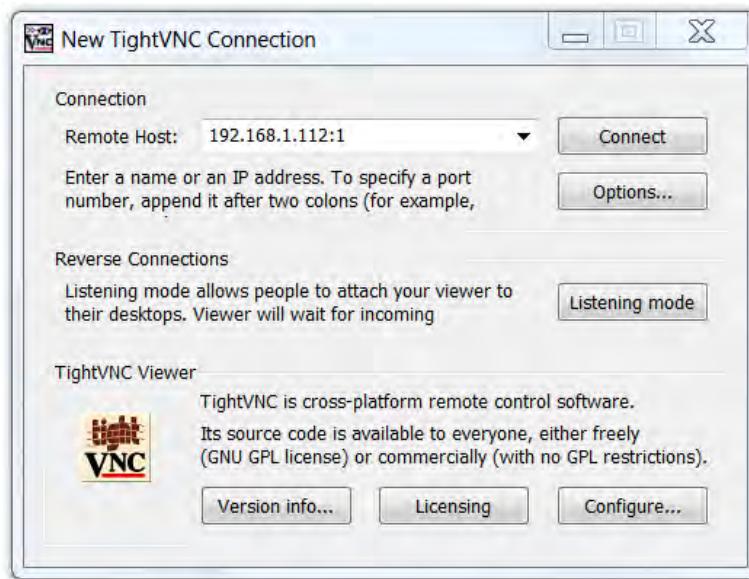
จะมีข้อความ You will require a password to access your desktops. ให้ใส่รหัสผ่าน 8 ตัว ซึ่งรหัสผ่านนี้จะใช้ในการเข้าโปรแกรม TightVNC Remote Desktop จากเครื่องคอมพิวเตอร์ ถ้าต้องการกำหนดขนาดหน้าจอที่จะแสดง เช่น 1920x1080 ด้วยการเปิดเป็น Desktop Display :1 สามารถทำได้ด้วยการใช้คำสั่ง

`vncserver :1 -geometry 1920x1080 -depth 24`

การใช้คำสั่งนี้จะเป็นการกำหนดให้ทำรังเดียว ถ้าต้องการให้เครื่อง Raspberry Pi ทำงานอัตโนมัติทุกครั้งที่บูท Raspberry Pi จะต้องเข้าไปแก้ไขเพิ่มคำสั่งเข้าไปในไฟล์ `.bash_profile` โดยใช้คำสั่งแก้ไขไฟล์คือ

`sudo nano -c ~/.bash_profile`

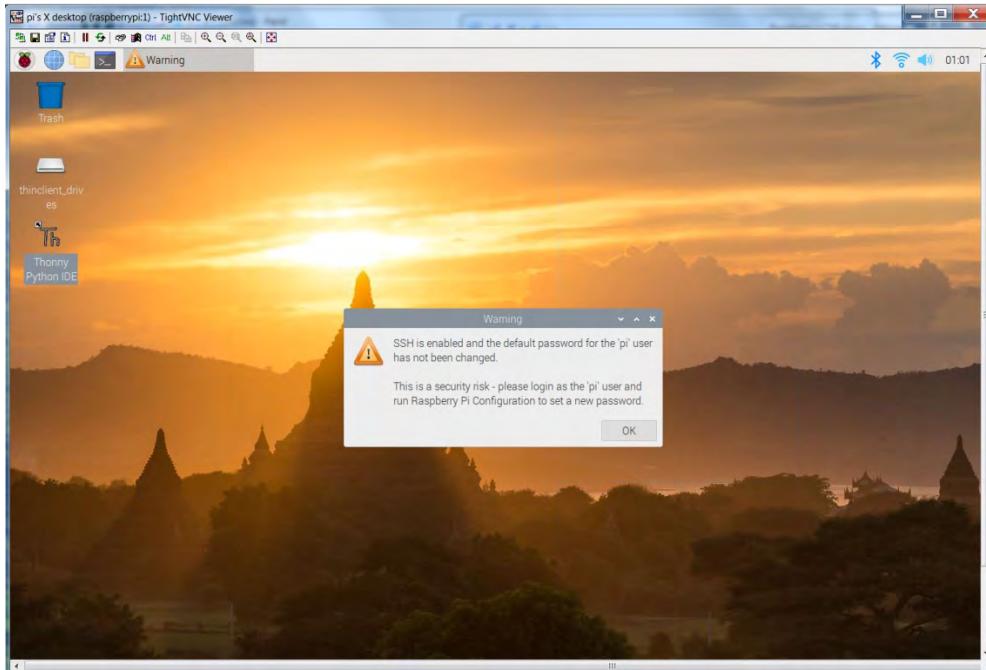
เมื่อติดตั้งโปรแกรมเสร็จแล้ว จากนั้นเราจะสามารถใช้ VNC Viewer บนอุปกรณ์ที่เราต้องการควบคุม ซึ่งจะใช้โปรแกรมใดก็ได้คือ TightVNC , RealVNC หรือ UltraVNC แต่ในที่นี้จะเปิดโปรแกรม TightVNC Viewer เป็นตัวอย่าง โดยจะต้องใส่หมายเลข IP ของ Raspberry Pi ตามด้วย:1 ซึ่งหมายถึง Display 1 หรือจะใส่เป็น `raspberrypi.local:1` แล้วตามด้วยหมายเลขของ Display ในที่นี้เป็น 1 ก็ได้



ให้ใส่รหัสผ่านที่ตั้งไว้



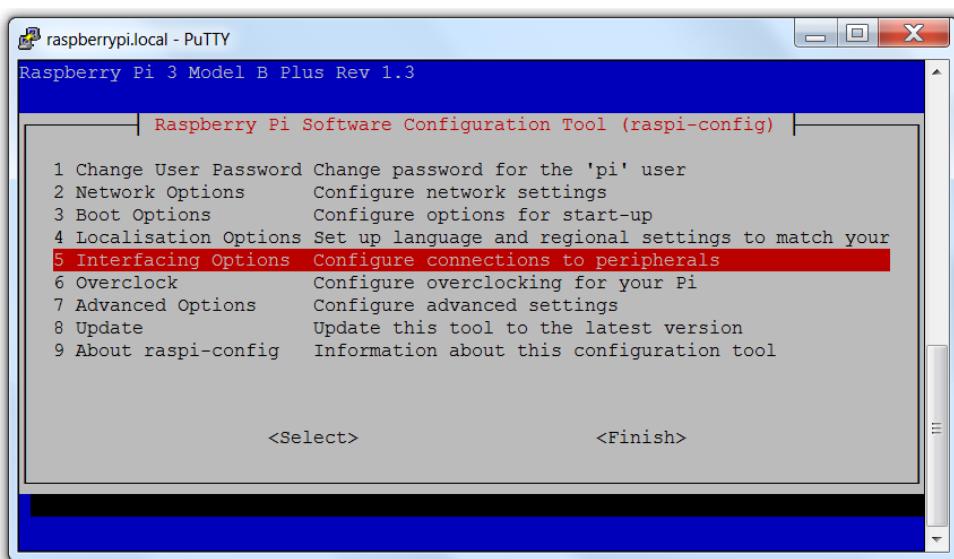
จะได้หน้าจอ Desktop ของ Raspberry Pi 3 B+ ตามรูป



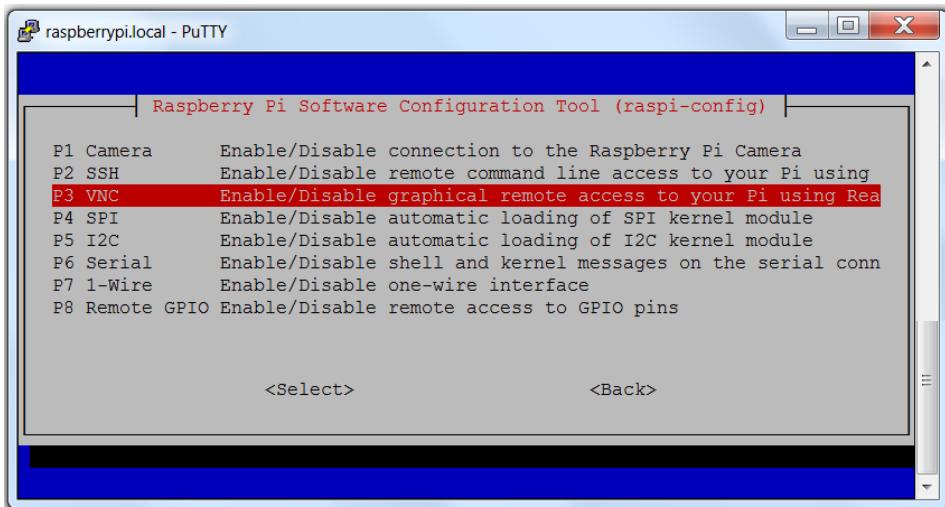
นอกจากวิธีใช้โปรแกรม Remote Desktop ด้วยการติดตั้งโปรแกรม tightvncserver ที่ Raspberry Pi ดังกล่าวแล้ว เรายังสามารถติดตั้งโปรแกรม VNC โดยใช้ Menu ของ Raspberry Pi ได้โดยตรง เนื่องจากใน Raspbian จะมี VNC Connect ซึ่งเป็นซอฟต์แวร์การเข้าถึงระยะไกลของ RealVNC ประกอบด้วย VNC Server และ VNC Viewer สิ่งที่เราต้องทำก่อนคือการตรวจสอบให้แน่ใจว่าเรามี VNC Connect เวอร์ชันล่าสุดแล้ว แต่การเรียกใช้โปรแกรม VNC Viewer ที่ Windows ด้วยวิธีการนี้สามารถใช้งานได้เฉพาะกับโปรแกรม RealVNC Viewer เท่านั้น ซึ่งมีวิธีการติดตั้งได้ 2 แบบดังนี้

แบบแรกด้วยวิธี SSH ใน Raspberry Pi โดยใช้โปรแกรม PuTTY เรียกใช้คำสั่งใน Terminal คือ

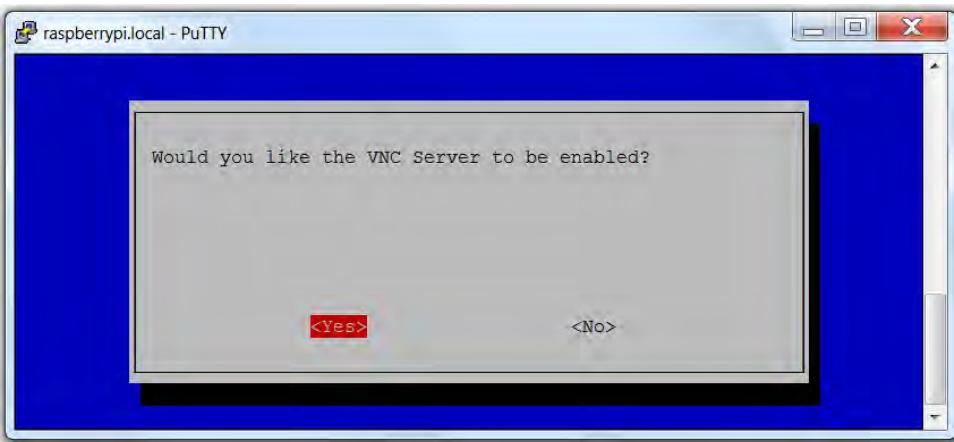
`sudo raspi-config` เพื่อเปิด Raspberry Pi Software Configuration Tool จากนั้นให้เลือก Interfacing Options



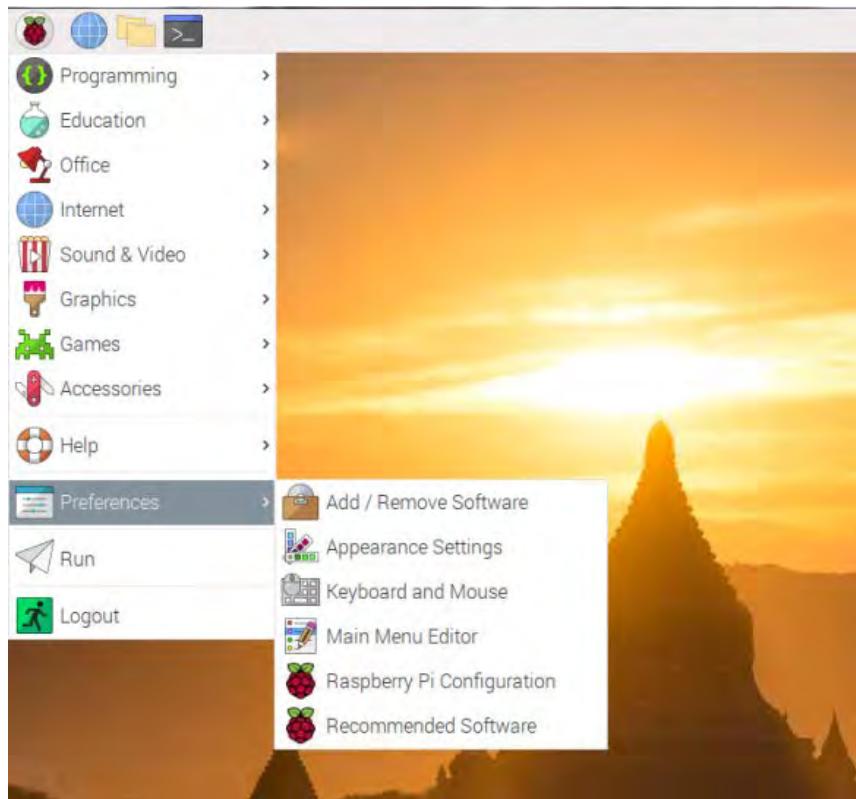
เลือก VNC



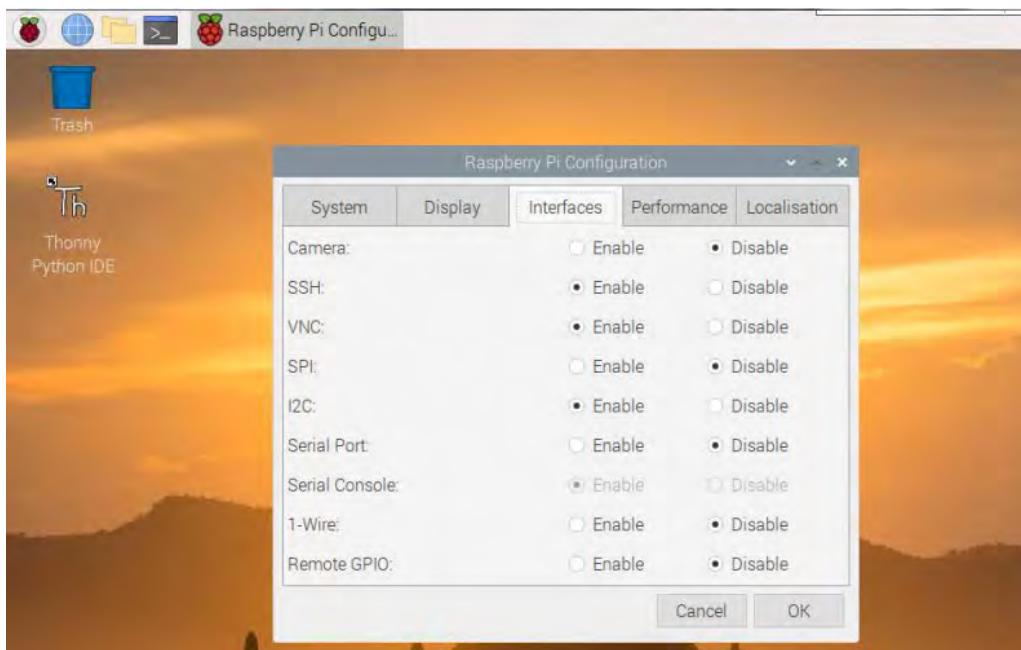
จะเห็นคำว่าต้องการให้ VNC Server เปิดใช้งานหรือไม่ โดยในตัวเลือกจะเป็น No ถ้าแม้จะเคยติดตั้ง tightvncserver ไว้แล้วก็ตาม ให้เปลี่ยนตัวเลือกเป็น Yes ซึ่งโปรแกรมจะทำการติดตั้ง RealVNC ให้และถ้าในเครื่องมีโปรแกรม tightvncserver ซึ่งเคยติดตั้งไว้ก่อนหน้านี้ก็จะถูกถอนการติดตั้งด้วยเป็นผลให้จะไม่สามารถใช้งานโปรแกรม TightVNC หรือ UltraVNC ได้ตัว



แบบที่สองด้วยวิธีการใช้ GUI (Graphic User Interface) บนหน้า Desktop ของ Raspberry Pi ให้ไปที่ตัวเลือก Preferences แล้วเลือก Raspberry Pi Configuration แล้วทำการ click



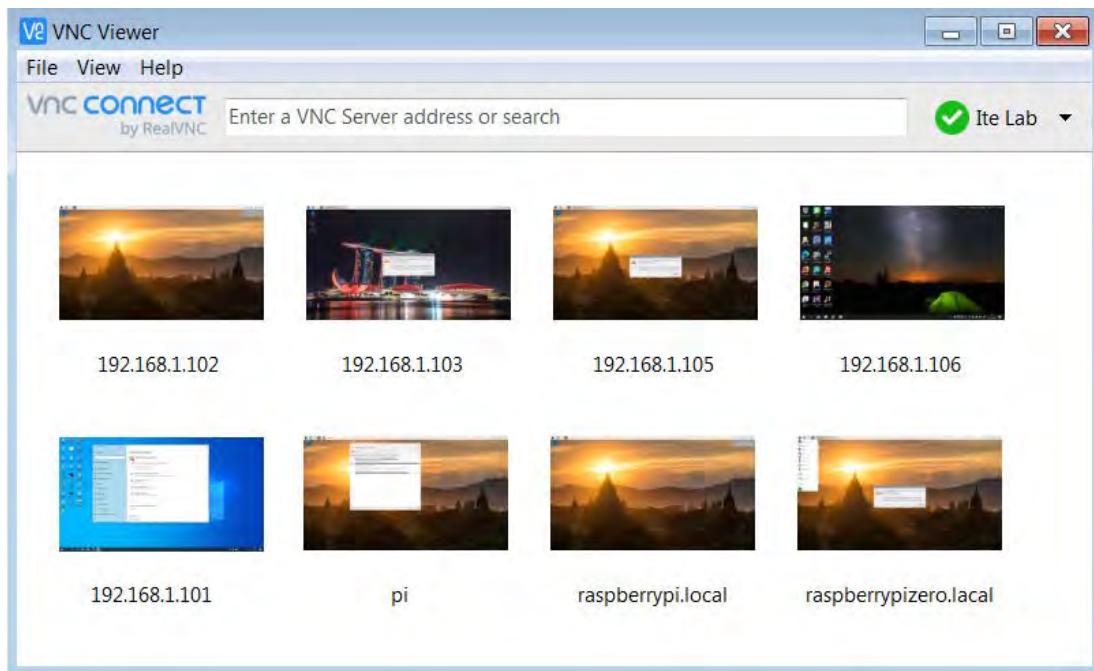
หลังจากนั้นให้ไปที่ Interfaces แล้วให้เลือก VNC Enabled ซึ่งจะได้ผลลัพธ์เหมือนแบบแรก



เมื่อติดตั้งและกำหนดค่าต่างๆแล้วจะสามารถจัดการแอปพลิเคชัน โดยดาวน์โหลดโปรแกรม RealVNC เพื่อติดตั้งไว้ในเครื่องคอมพิวเตอร์ Windows , macOS , Android , iOS , Linux "ได้จาก

<https://www.realvnc.com/en/connect/download/viewer/>

ตัวอย่างหน้าจอการใช้งานแสดงดังรูป



### วิธีการติดตั้งและการกำหนดค่า VNC Server บน Ubuntu

ให้ติดตั้ง Desktop Environment และติดตั้งเซิร์ฟเวอร์ TightVNC โดยใช้คำสั่งในเทอร์มินัลดังนี้

```

sudo apt update
sudo apt install xfce4 xfce4-goodies
sudo apt install tightvncserver

```

เมื่อเริ่มใช้เซิร์ฟเวอร์ TightVNC เป็นครั้งแรก จะได้รับแจ้งให้ป้อนและยืนยันรหัสผ่านเพื่อให้ตั้งค่ารหัสผ่าน ด้วยคำสั่ง vncserver

การกำหนดค่าเซิร์ฟเวอร์ TightVNC จะต้องให้หยุดเซิร์ฟเวอร์ TightVNC เพื่อปรับการตั้งค่าด้วยคำสั่ง vncserver -kill :1

เปิดไฟล์การกำหนดค่า TightVNC

```
nano ~/.vnc/xstartup
```

เพิ่มบรรทัดต่อไปนี้ที่ส่วนท้ายของไฟล์

```
startxfce4
```

การให้เซิร์ฟเวอร์เริ่มทำงานใหม่อีกครั้งจะใช้คำสั่ง

```
vncserver :1 -geometry 800x600 -depth 24
```

การตั้งค่าให้ทำงานอัตโนมัติทุกครั้งที่เปิดเครื่อง โดยสร้างไฟล์หน่วยบริการ สำหรับจัดการเซิร์ฟเวอร์ VNC ด้วยคำสั่ง nano /etc/systemd/system/vncserver.service

เพิ่มการกำหนดค่าต่อไปนี้เข้าไป

```
[Unit]
Description=TightVNC server
After=syslog.target network.target

[Service]
Type=forking
User=root
PAMName=login
PIDFile=/root/.vnc/%H:1.pid
ExecStartPre=/usr/bin/vncserver -kill :1 > /dev/null 2>&1
ExecStart=/usr/bin/vncserver :1 -geometry 1280x720 -depth 24
ExecStop=/usr/bin/vncserver -kill :1

[Install]
WantedBy=multi-user.target
```

โหลดระบบใหม่อีกครั้ง

```
systemctl daemon-reload
```

เปิดใช้งานทำงานอัตโนมัติของเซิร์ฟเวอร์ TightVNC และเริ่มทำงาน

```
systemctl enable --now vncserver
```

ถ้าต้องการตรวจสอบสถานะของการทำงานให้ใช้คำสั่ง

```
systemctl status vncserver
```

การตั้งค่าให้เครื่อง ODROID ที่ใช้ระบบปฏิบัติการ Ubuntu MATE เปิดใช้งานเพื่อให้สามารถเข้าสู่ระบบเดสก์ท็อปโดยอัตโนมัติ ให้เรียกใช้ทอร์มินัลด้วยคำสั่งต่อไปนี้

```
cd /usr/share/lightdm/lightdm.conf.d
```

```
sudo nano 50-slick-greeter.conf
```

ให้เพิ่มส่วนท้ายของไฟล์ด้วยคำสั่งดังนี้

```
[Seat:*]
greeter-sessionslick-greeter
autologin-userodroid
```

หลังจากนั้นให้เริ่มต้นการทำงานของระบบปฏิบัติการใหม่ ด้วยคำสั่ง

```
sudo reboot
```

ผู้ใช้ที่ระบุหลังจาก autologin-user= จะเข้าสู่ระบบโดยอัตโนมัติ และจะทำให้การตั้งค่าต่างๆ ทำงาน เช่น ระบบ WiFi

กรณีที่ติดตั้ง VNC Server ในระบบปฏิบัติการ Kali Linux หรือมีปัญหาขึ้นหน้าจอเป็นสีเทา ให้ติดตั้ง x11VNC ที่เป็นซอฟต์แวร์เข้าถึงจากโคลอ่อนต์รรยะไกลไปที่โฮสต์เชลล์ชั้นเฟรมบัฟเฟอร์ X โดยให้ติดตั้งที่เซิร์ฟเวอร์ด้วยคำสั่งในเทอร์มินัลดังนี้

```
Sudo apt update
sudo apt install x11vnc
```

เมื่อเริ่มใช้เซิร์ฟเวอร์ x11vnc เป็นครั้งแรก จะต้องตั้งค่ารหัสผ่านด้วยคำสั่ง

```
vncserver
```

จากนั้นจะต้องกำหนดค่าความละเอียดของหน้าจอให้ถูกต้อง โดยเปิดไฟล์การกำหนดค่าด้วยคำสั่ง

```
sudo nano ~/.vnc/xstartup
```

และให้ใส่เครื่องหมาย “#” ไว้ที่ด้านหน้าของทุกบรรทัดดังรูป

```
kali@kali: ~
GNU nano 7.2          /home/kali/.vnc/xstartup
#!/bin/sh

#xrdb "$HOME/.Xresources"
#xsetroot -solid grey
#x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
#x-window-manager &
# Fix to make GNOME work
#export XKL_XMODMAP_DISABLE=1
#/etc/X11/Xsession

File to insert [from ./]:
```

เพิ่มบรรทัดต่อไปนี้ที่ส่วนท้ายของไฟล์ xstartup

```
#!/bin/sh
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
startxfce4 &

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
```

สร้างไฟล์สคริปต์ที่จะให้ vncserver เริ่มต้นทำงานโดยอัตโนมัติทุกครั้งที่เปิดเครื่องด้วยคำสั่ง

```
sudo nano /etc/init.d/vncserver
```

เพิ่มบรรทัดต่อไปนี้ในไฟล์ vncserver ดังนี้

```

#!/bin/sh -e
## # BEGIN INIT INFO
# Provides: vncserver
# Required-Start: networking
# Default-Start: 3 4 5
# Default-Stop: 0 6
## # END INIT INFO

PATH="$PATH:/usr/X11R6/bin/"

# The Username:Group that will run VNC
export USER="kali"
#${RUNAS}

# The display that VNC will use
DISPLAY="1"

# Color depth (between 8 and 32)
DEPTH="16"

# The Desktop geometry to use.
#GEOMETRY="x"
#GEOMETRY="800x600"
#GEOMETRY="1920x1080"
GEOMETRY="1280x1024"

# The name that the VNC Desktop will have.
NAME="my-vnc-server"

OPTIONS="-name ${NAME} -depth ${DEPTH} -geometry ${GEOMETRY} :${DISPLAY}"
. /lib/lsb/init-functions

case "$1" in
start)
log_action_begin_msg "Starting vncserver for user '${USER}' on localhost:${DISPLAY}"
su ${USER} -c "/usr/bin/vncserver ${OPTIONS}"
;;
stop)
log_action_begin_msg "Stopping vncserver for user '${USER}' on localhost:${DISPLAY}"
su ${USER} -c "/usr/bin/vncserver -kill :${DISPLAY}"
;;
restart)
$0 stop
$0 start
;;
esac

exit 0

```

เปิดการให้ลิฟท์ที่จำเป็นเพื่อให้ vncserver สามารถทำงานได้อัตโนมัติ ด้วยคำสั่งดังนี้

```
sudo chmod +x /etc/init.d/vncserver
```

```
sudo update-rc.d vncserver defaults
```

ให้ Reboot เริ่มต้นการทำงานของเครื่อง แล้วตรวจสอบว่าสคริปต์เริ่มต้นอัตโนมัติทำงานหรือไม่ โดยสถานะของ vncserver ด้วยคำสั่ง

```
service vncserver status
```

หรือ

```
systemctl status vncserver
```

ตัวอย่างผลลัพธ์ที่ได้แสดงดังรูป

```
kali㉿kali: ~
└─$ service vncserver status
● vncserver.service
    Loaded: loaded (/etc/init.d/vncserver; generated)
    Active: active (exited) since Sun 2023-04-16 23:16:26 +07; 4min 7s ago
      Docs: man:systemd-sysv-generator(8)
   Process: 641 ExecStart=/etc/init.d/vncserver start (code=exited, status=0/s>
     CPU: 59ms

Apr 16 23:16:22 kali systemd[1]: Starting vncserver.service...
Apr 16 23:16:22 kali su[668]: (to kali) root on none
Apr 16 23:16:22 kali su[668]: pam_unix(su:session): session opened for user kali
Apr 16 23:16:25 kali vncserver[641]: Starting vncserver for user 'kali' on local display 1
Apr 16 23:16:25 kali vncserver[811]: xauth: (argv):1: bad display name "kali:1"
Apr 16 23:16:26 kali systemd[1]: Started vncserver.service.

lines 1-13/13 (END)
```

การตั้งค่าให้บอร์ด Raspberry Pi ที่ใช้ระบบ Kali Linux เปิดใช้งานการเข้าสู่หน้าเดสก์ท็อปโดยอัตโนมัติ ให้เรียกใช้ เทอร์มินัลเพื่อแก้ไขไฟล์คอนฟิกurexchange lightdm.conf ด้วยคำสั่ง

```
sudo nano /etc/lightdm/lightdm.conf
```

ให้อาเครื่องหมาย “#” ที่ใส่ไว้ที่ด้านหน้าของบรรทัดที่บรรทัดที่ 83,84 และ 126,127 ออก และตั้งค่าเป็นดังนี้

```
autologin-user = kali
autologin-user-timeout=0
```

ให้อาเครื่องหมาย “#” ที่ใส่ไว้ที่ด้านหน้าของบรรทัดที่ 92 ออก

```
pam-autologin-service=lightdm-autologin
```

เปิดไฟล์เพื่อแก้ไขด้วยคำสั่ง

```
sudo nano /etc/pam.d/lightdm-autologin
```

ถ้ามีเครื่องหมาย “#” ที่ใส่ไว้ที่ด้านหน้าของบรรทัดนี้ให้ออก

```
auth required pam_succeed_if.so user != root quiet_success
```

หลังจากนี้ให้รีเมต์การทำงานของระบบปฏิบัติการใหม่ ด้วยคำสั่ง

```
sudo reboot
```

ผู้ใช้ที่ระบุหลังจาก autologin-user= ตามตัวอย่างคือ kali จะเข้าสู่ระบบโดยอัตโนมัติ

ระบบปฏิบัติการ Kali Linux จะมีการติดตั้ง Service ต่างๆไว้แล้ว เช่น Samba service และ Apache service สำหรับการ เปิดเพื่อใช้งานโดยให้เริ่มต้นการทำงานแบบอัตโนมัติทุกรอบที่ปิดเครื่อง ตัวอย่างเช่น

Samba service ให้ใช้คำสั่ง

```
sudo systemctl enable smbd
```

```
sudo systemctl start smbd
```

Apache2 service ให้ใช้คำสั่ง

```
sudo systemctl enable apache2
```

```
sudo systemctl start apache2
```

โดยที่รูปแบบการทำงานของคำสั่ง systemctl จะเป็นดังนี้

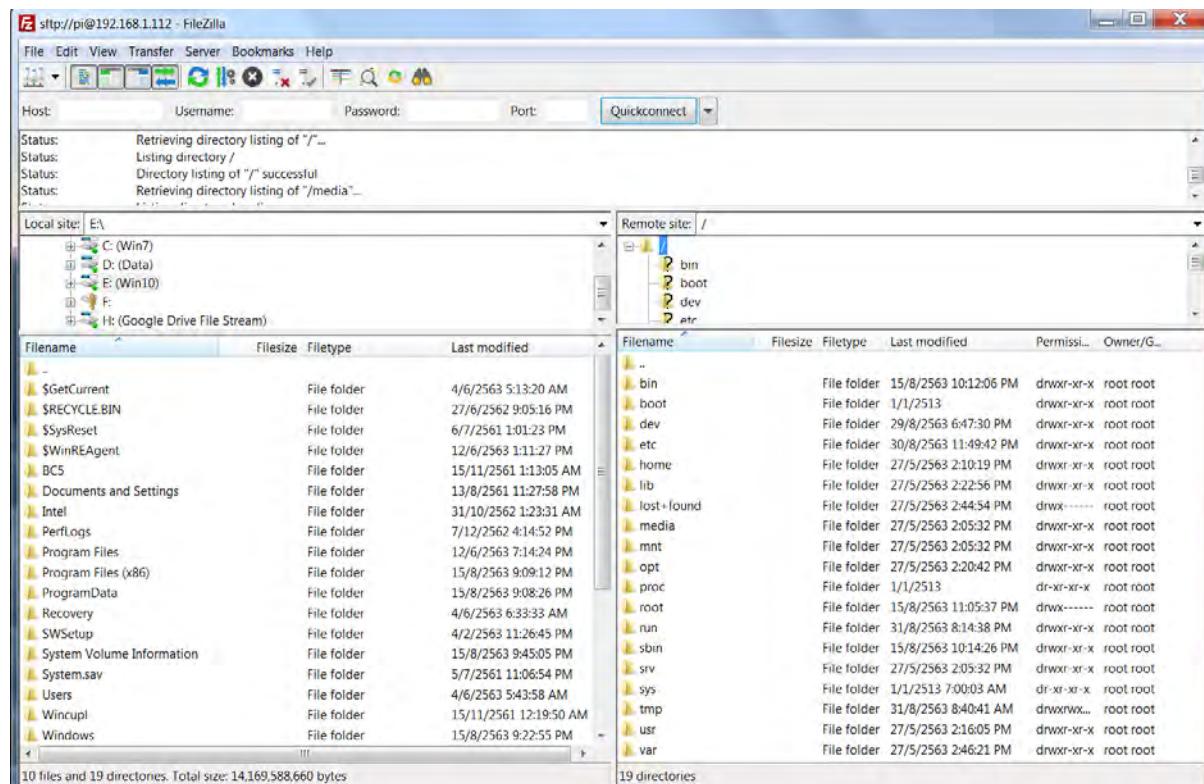
```
systemctl [start|stop|restart|reload|enable|disable|status]
```

ถ้าต้องการตรวจสอบว่ามี service อะไรเริ่มต้นทำงานอัตโนมัติ หรือกำลังทำงานอยู่หรือไม่ ให้ใช้คำสั่ง service --status-all ตัวอย่างผลลัพธ์ที่ได้

```
(kali㉿kali)-[~]
$ service --status-all
[ + ] alsa-utils
[ - ] apache-htcacheload
[ + ] apache2
[ - ] atftpd
[ + ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ - ] cryptdisks
[ - ] cryptdisks-early
[ + ] dbus
[ - ] dns2tcp
[ - ] haveged
[ - ] hwclock.sh
[ - ] inetsim
[ - ] iodined
[ - ] ipsec
[ - ] keyboard-setup.sh
[ + ] kmod
[ - ] lightdm
[ - ] lm-sensors
[ + ] mariadb
[ - ] miredo
[ - ] mosquitto
[ + ] networking
[ - ] nfs-common
[ - ] nginx
[ - ] nmbd
[ - ] openvpn
[ - ] pcscd
[ + ] plymouth
[ + ] plymouth-log
[ - ] postgresql
[ + ] procps
[ - ] ptunnel
[ - ] pulseaudio-enable-autospawn
[ - ] redis-server
[ - ] redsocks
[ - ] rpcbind
[ - ] rsync
[ - ] rwhod
[ - ] samba-ad-dc
[ - ] saned
[ - ] screen-cleanup
[ - ] smartmontools
[ + ] smbd
[ - ] snmpd
[ - ] speech-dispatcher
[ + ] ssh
[ - ] sslh
[ - ] stunnel4
[ - ] sudo
[ - ] sysstat
[ - ] triggerhappy
[ + ] udev
[ + ] vncserver
[ - ] x11-common
[ - ] xl2tpd
```

การ Upload หรือ Download ข้อมูลใน Raspberry Pi Server จะใช้การทำงานของ FTP (File Transfer Protocol) ซึ่งเป็นไฟร์โടกคลอที่ออกแบบมาเพื่อใช้ในการรับส่งไฟล์ระหว่าง Client และ Server ในที่นี้จะใช้โปรแกรม FileZilla โดยคอมพิวเตอร์ที่ทำงานบนระบบปฏิบัติการวินโดว์สามารถโหลดได้ที่ <https://filezilla-project.org/>

เมื่อติดตั้งเสร็จแล้ว ให้เรียกใช้โปรแกรม FileZilla ที่ Host ให้กรอกหมายเลข IP ของ Raspberry Pi หรือใส่เป็น raspberrypi.local และที่ Username เป็น pi และ Password เป็น raspberry ที่ Port จะเป็นการระบุพอร์ตของ FTP ซึ่งค่า default ของ Raspberry Pi คือ port 22 หลังจากนั้นให้คลิกที่ปุ่ม "Quickconnect" เพื่อทำการเชื่อมต่อ กับ Raspberry Pi 3 B+ ดังรูป



การแก้ไขไฟล์ต่างๆ เราอาจจะใช้โปรแกรม FileZilla เพื่อ Download ไฟล์ข้อมูลบางอย่างใน Raspberry Pi มาทำการแก้ไขเสร็จแล้วก็ Upload คืนไป โดยมีซอฟท์แวร์ที่จำเป็นต้องในการใช้งานจัดการคือ Notepad ++ เป็น Text Editor ที่ไว้ช่วยในการตรวจสอบและแก้ไขไฟล์ข้อมูลต่างๆ ข้อควรระวังไม่ควรใช้โปรแกรม Notepad++ บน Window เพราะจะมีปัญหาเรื่องการเว้นบรรทัด ให้ใช้ Notepad ++ แทน นอกจากนี้ Notepad++ ยังสามารถนำไปเขียนโปรแกรมภาษาต่างๆ ให้ง่ายขึ้น เพราะมีการฟอร์แมตหัวสีของคำสั่งต่างๆ ทำให้สะดวกต่อการตรวจสอบโปรแกรม สามารถดาวน์โหลดโปรแกรมเพื่อติดตั้งได้จาก <https://notepad-plus-plus.org/downloads/>

ถ้าต้องการแชร์ไฟล์บางไฟล์ผ่านเครือข่ายโดยสร้างเป็นชิร์ฟเวอร์จัดเก็บข้อมูลภายในบ้านหรือต้องการถ่ายโอนไฟล์ระหว่างเครื่อง Windows กับอุปกรณ์ Raspberry Pi โดยใช้โปรแกรม Windows File Explorer ซึ่งจะมีความสามารถใช้โปรแกรม FileZilla สามารถทำได้ด้วยการใช้ Samba ที่เป็นการใช้งานโปรโตคอล SMB (Server Message Block) และช่วยให้สามารถแชร์การเข้าถึงไฟล์และเครื่องพิมพ์ระหว่างคอมพิวเตอร์ต่างๆบนเครือข่ายได้ ซึ่งเป็น Free ware network file system ทำให้ผู้ใช้สามารถเข้าถึงการแก้ไขไฟล์และใช้ทรัพยากร่วมกันบนระบบเครือข่าย โดยที่ Samba จะสามารถติดตั้งบนระบบปฏิบัติการได้หลายแบบรวมถึงระบบ Raspbian บน Raspberry Pi เพื่อขั้นตอนการติดตั้ง โดยเริ่มจากการตรวจสอบและติดตั้งแพ็คเกจล่าสุดสำหรับ Raspberry Pi เปิดหน้า Terminal ด้วยโปรแกรม PuTTY เข้าสู่ระบบ Pi และป้อนคำสั่ง

```
sudo apt-get update
sudo apt-get upgrade
```

แล้วติดตั้ง Samba โดยพิมพ์คำสั่ง

```
sudo apt-get install samba samba-common-bin
```

เมื่อติดตั้งเสร็จแล้วเราต้องกำหนดค่า Samba เพื่อ Share ไดเรกทอรีของ Raspberry Pi ให้กับคอมพิวเตอร์เครื่องอื่นภายในเครือข่าย ด้วยการป้อนคำสั่งเพื่อเปิดไฟล์ smb.conf และเข้าไปตั้งค่าการใช้ใช้งานด้วยคำสั่ง

```
sudo nano /etc/samba/smb.conf
```

```

pi@raspberrypi: ~
GNU nano 3.2          /etc/samba/smb.conf

[netlogon]
comment = Network Logon Service
path = /home/samba/netlogon
guest ok = yes
read only = yes

# Un-comment the following and create the profiles directory to store
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
[profiles]
comment = Users profiles
path = /home/samba/profiles
guest ok = no
browseable = no
create mask = 0600
directory mask = 0700

[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
printable = yes

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^  Go To Line

```

โดยจะเห็นว่ามีการตั้งค่า Config เดิมของ Samba อยู่แล้ว ซึ่งเราสามารถเลื่อนไฟล์ลงเพื่อจะดูสิ่งที่ต้องการเปิดใช้งาน และสามารถที่จะลบหรือแก้ไขเฉพาะในส่วนที่จำเป็น ดังตัวอย่าง

```

[global]
netbios name = raspberrypi
server string = Raspberry Pi File Center
workgroup = WORKGROUP

[Root]
comment = Root Directories
path = /
browsable = yes
writeable = yes
create mask = 0777
directory mask = 0777
public = no

[Home]
comment = Home Directories
path = /home/pi
browsable = yes
writeable = yes
create mask = 0777
directory mask = 0777
public = no

```

โดยค่าที่กำหนดมีความหมายดังนี้

workgroup คือโ dikmenที่เซิร์ฟเวอร์ Samba จะเป็นส่วนการกำหนดค่า Default ของ Windows

comment คือ ให้ได้รายละเอียดคำอธิบาย

path คือเส้นทางไปยัง Directory ในไฟล์เดอร์ของ Raspberry Pi ที่จะใช้ในการ Share browsable คือ อนุญาติให้ตรวจสอบได้ในเครือข่าย writeable ถ้าตั้งค่าเป็น yes จะอนุญาตให้แก้ไข File ได้ create mask และ directory mask การกำหนดสิทธิ์ของ user เมื่อตั้งค่า Attributes เป็น 777 จะอนุญาตให้ผู้ใช้สามารถทำการ read, write และ execute ได้ public ถ้าตั้งค่าเป็น no จะอนุญาตให้ผู้ใช้ที่ถูกต้องเข้าถึงไฟล์เดอร์ที่ใช้รวมกันเท่านั้น

หลังจากที่ป้อนข้อมูลข้างต้นเสร็จแล้วให้กด Ctrl + o เพื่อ Save ข้อมูล แล้วกด Ctrl + x เพื่อออกจาก Editor เพื่อความปลอดภัยของข้อมูล เราสามารถสร้าง password สำหรับ user ชื่อ pi โดยใช้คำสั่ง

```
sudo smbpasswd -a pi
```

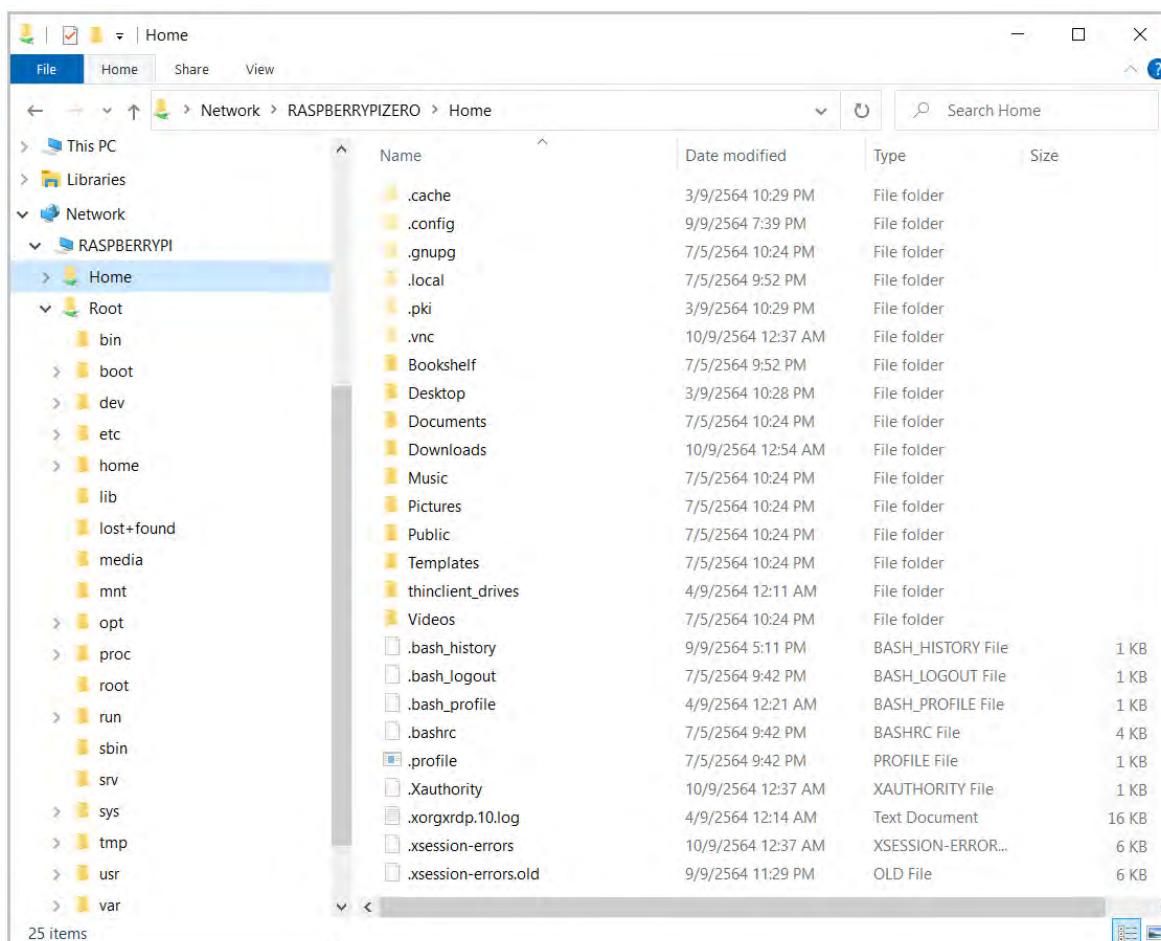
เพื่อให้การบริการ Samba เริ่มทำงานตาม config ที่ตั้งไว้ให้ใช้คำสั่ง

```
sudo service smbd restart
```

ถ้าจะตรวจสอบสถานะการทำงานของ Samba ให้ใช้คำสั่ง

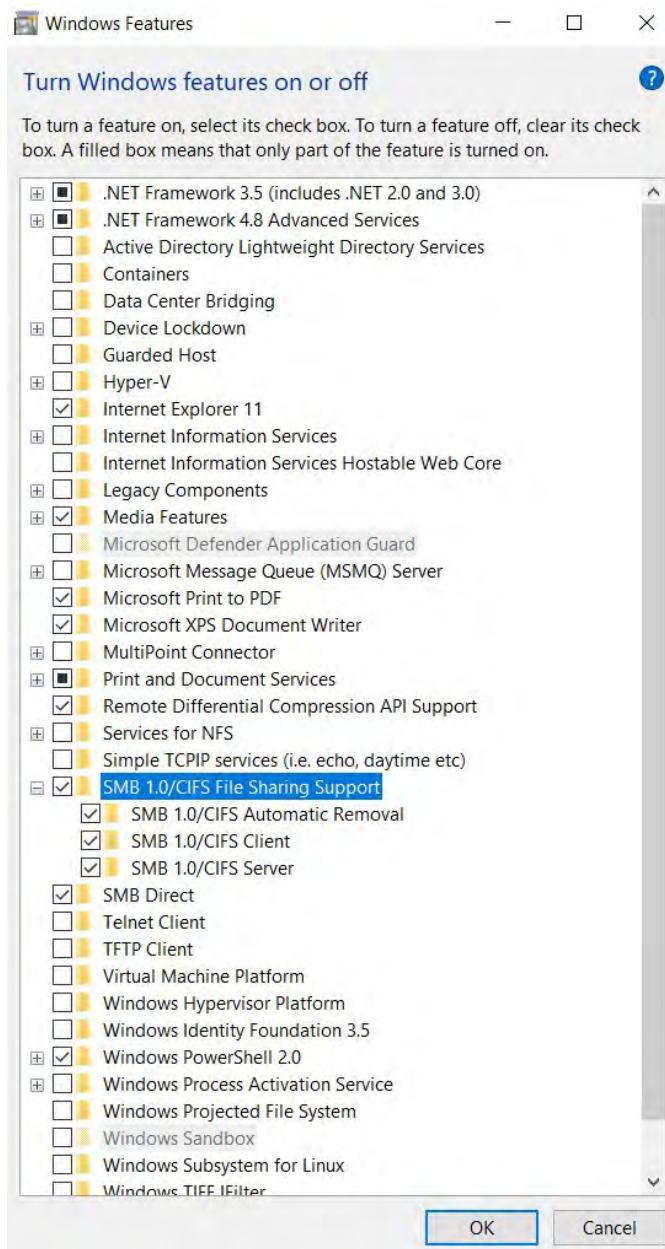
```
sudo systemctl status smbd
```

เมื่อ Raspberry PI เริ่มทำงานใหม่ตามระบบ Samba ที่ได้รับการตั้งค่าไว้แล้ว จะทำให้เราสามารถใช้ Windows File Explorer ในการเข้าถึงไฟล์ต่างๆ ให้คลิกที่ Network เพื่อเข้าถึงไฟล์เดอร์ที่ Share ไว้ของ Raspberry Pi เมื่อคลิกที่ไฟล์เดอร์ป้อนชื่อผู้ใช้ของ Pi แล้วป้อนรหัสผ่าน หลังจากที่เข้าสู่ระบบแล้วจะสามารถจัดการไฟล์เดอร์และไฟล์ของ Raspberry Pi ได้ดังรูป

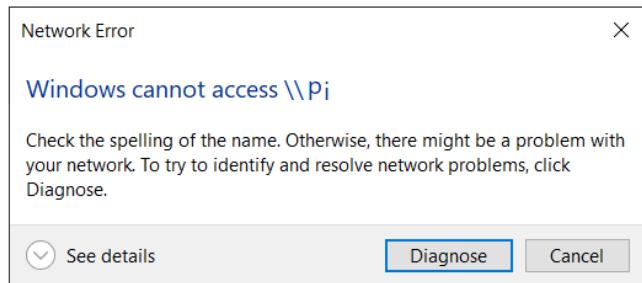


กรณีที่ใช้ Windows10 แล้วไม่สามารถหันการแชร์ของคอมพิวเตอร์หรืออุปกรณ์อื่นบนเครือข่ายได้ มีวิธีแก้ไขคือ

- กดปุ่ม Search บน Windows
- พิมพ์และเลือก Turn Windows features on or off
- กดขยายเมนู SMB 1.0/CIFS File Sharing Support
- คลิกเลือก ตัวเลือกทั้งหมด จะทำให้คอมพิวเตอร์สนับสนุนการแชร์โดยTOCOL SMB (Server Message Block) ทั้งหมด
- กดปุ่ม OK และ Restart now

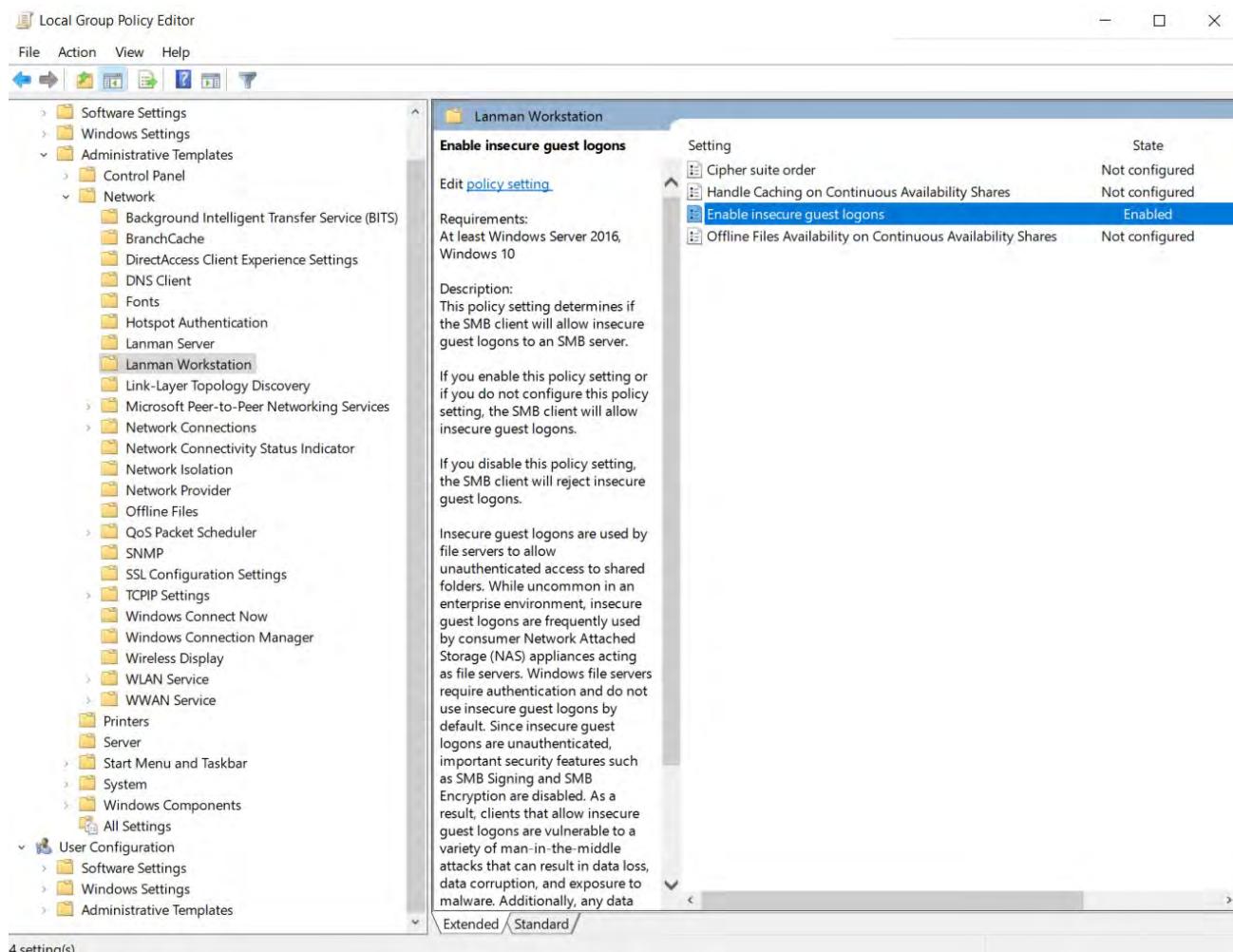


กรณีที่ Windows10 เรียกใช้การแชร์ของคอมพิวเตอร์หรืออุปกรณ์อื่นบนเครือข่ายแล้วเกิด Network Error ขึ้นคำว่า Windows cannot access\\..... ดังรูป



ມີຫັນຕອນກາຣແກ້ໄຂເກືອ

- ກົດປຸ່ມ Search ບນ Windows
- ພິມພີແລະເລືອກ Edit Group Policy
- ກົດຂາຍເມນູ Computer Configuration > Administrative Templates > Network > Lanman Workstation
- ດັບເນື້ອລົດຕິດ Enable insecure guest logons
- ຄລິກເລືອກ Enable
- ກົດປຸ່ມ Apply ແລະ ທຳກາຣ Restart ເຄົ່າງ



หลังจากได้ทำการติดตั้งห้องดูแล Micro SD ที่บรรจุระบบปฏิบัติการ Raspbian และข้อมูลต่างๆไว้แล้ว ซึ่งภายใน Micro SD นี้จะมี 2 Drive ที่ประกอบด้วย ไดร์ฟ boot ที่เป็นระบบไฟล์แบบ FAT32 ซึ่งสามารถเปิดไฟล์ในเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Windows ได้ และไดร์ฟ rootfs ที่เป็นระบบไฟล์แบบ Ext4 ของระบบปฏิบัติการ Linux ซึ่งเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Windows ไม่สามารถเปิดไฟล์ได้

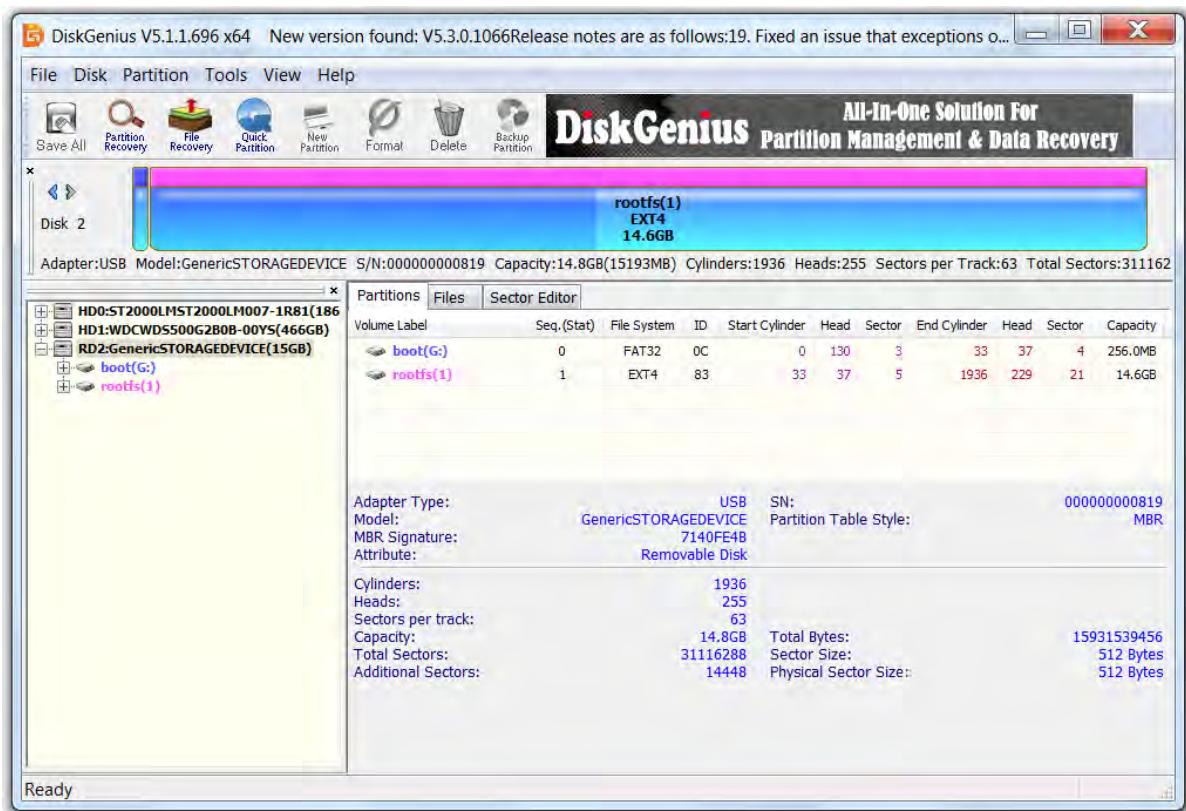
Partition	Capacity	Used	Unused	File System	Type	Status
<b>Disk 3 (Generic STORAGE DEVICE USB, Removable, MBR, 14.84 GB)</b>						
G:boot	256.00 MB	57.27 MB	198.73 MB	FAT32	Primary	None
*:rootfs	14.58 GB	6.54 GB	8.05 GB	Ext4	Primary	None

Disk 3  
 MBR  
 14.84 GB

G:boot(FAT32)  
 256 MB ( 14.6 GB (Used: 44%)

■ GPT/Primary   ■ Logical   ■ Simple   ■ Spanned   ■ Striped   ■ Mirrored   ■ RAID5   ■ Unallocated

ถ้าต้องการจะสำรองข้อมูลเก็บไว้ หรือต้องการ Copy ระบบปฏิบัติการ Raspbian และข้อมูลต่างๆ ไปเก็บไว้ใน Micro SD Card อื่น เพื่อใช้งานกับ Raspberry Pi 3 B+ บอร์ดอื่นโดยไม่ต้องลงระบบปฏิบัติการ Raspbian ใหม่ สามารถทำได้โดยใช้โปรแกรม DiskGenius. Disk Partition And Data Recovery Software ซึ่งเป็นโปรแกรมใช้สำหรับ Clone Disk ที่ภายใน Drive ตัวเดียวกันจะประกอบด้วยระบบปฏิบัติการหลายอย่างรวมกันในแต่ละ Partition ทั้งระบบ File System แบบ FAT , NTFS และ Ext4 ได้ โดยดาวน์โหลดได้จาก <https://www.diskgenius.com/?from=dg>



**OpenCV** (Opensource Computer Vision) เป็นไลบรารีฟังก์ชันที่ใช้ในการเขียนโปรแกรมการแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ มีชุดเครื่องมือที่ใช้ในการประมวลผลภาพที่มีลักษณะ 2D และ 3D สามารถนำไปใช้สำหรับ Image Processing และ Computer Vision ช่วยในการจัดการประมวลผลรูปภาพและวิดีโอ เช่น การ Optimization ระบบฐานข้อมูลนี้ ระบบการจัดจำหน้า ใช้เป็นเครื่องมือการประมาณระยะในขณะเคลื่อนที่ การปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ ในการติดตั้งจำเป็นต้องอัพเกรดแพ็คเกจที่มีอยู่และติดตั้งแพ็คเกจใหม่จำนวนมาก เพื่อให้ OpenCV ทำงานได้อย่างถูกต้อง มีขั้นตอนการติดตั้งดังนี้

1. เริ่มต้นด้วยการอัปเดตและปรับปรุงแพ็คเกจของ Raspberry Pi เป็นเวอร์ชันล่าสุด

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Updating firmware และ kernel modules ของ Raspberry Pi ด้วยคำสั่ง

```
sudo rpi-update
```

```
pi@pi:~$ sudo rpi-update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by Andrews and Dom
*** Performing self-update
*** Relaunching after update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by Andrews and Dom
*** We're running for the first time
*** Backing up files (this will take a few minutes)
*** Backing up firmware
*** Backing up modules 5.10.60-v7+
#####
WARNING: This update bumps to rpi-5.10.y linux tree
See: https://www.raspberrypi.org/forums/viewtopic.php?f=29&t=288234
'rpi-update' should only be used if there is a specific
reason to do so - for example, a request by a Raspberry Pi
engineer or if you want to help the testing effort
and are comfortable with restoring if there are regressions.

DO NOT use 'rpi-update' as part of a regular update process.

#####
Would you like to proceed? (y/N)
*** Downloading specific firmware revision (this will take a few minutes)
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload  Upload Total Spent   Left Speed
100  168  100  168    0     0   349      0 --::-- --::-- --::--  349
100 121M    0 121M    0     0  4349k      0 --::--  0:00:28 --::-- 4924k
*** Updating firmware
*** Updating kernel modules
*** depmod 5.10.63-v7+
*** depmod 5.10.63-v8+
*** depmod 5.10.63+
*** depmod 5.10.63-v71+
*** Updating VideoCore libraries
*** Using HardFP libraries
*** Updating SDK
*** Running ldconfig
*** Storing current firmware revision
*** Deleting downloaded files
*** Syncing changes to disk
*** If no errors appeared, your firmware was successfully updated to bad5282375
a2d1e4947432bd80a8839fbc3504ae
*** A reboot is needed to activate the new firmware
pi@pi:~$
```

ในระหว่างทำการอัพเดตต้องระวังไม่ให้ไฟดับ เพราะจะทำให้เกิดความเสียหายกับเครื่องได้ เมื่อทำการอัพเดตเสร็จล้วนจำเป็นต้องรีบูตระบบเครื่องใหม่ด้วยคำสั่ง

```
sudo reboot
```

3. ติดตั้ง Developer tools ที่จะใช้งานช่วง compile

```
sudo apt-get install build-essential cmake pkg-config
```

```
pi@pi:~$ sudo apt-get install build-essential cmake pkg-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.6).
pkg-config is already the newest version (0.29-6).
The following additional packages will be installed:
  cmake-data libjsoncpp1 librhash0 libuv1
Suggested packages:
  cmake-doc ninja-build
The following NEW packages will be installed:
  cmake cmake-data libjsoncpp1 librhash0 libuv1
0 upgraded, 5 newly installed, 0 to remove and 1 not upgraded.
Need to get 4,326 kB/4,423 kB of archives.
After this operation, 22.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] █
```

4. ทำการติดตั้ง image packages เพื่อใช้งานกับรูปภาพได้หลายรูปแบบ เช่น JPEG, PNG, TIFF

```
sudo apt-get install libjpeg-dev libtiff-dev libpng-dev
```

```
pi@pi:~$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
v
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpng-tools
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libjasper1 libjbig-dev libjpeg62-turbo-dev liblzma-dev libtiff-dev
  libtiffxx5
Suggested packages:
  libjasper-runtime liblzma-doc
The following packages will be REMOVED:
  libfreetype6-dev libpng-dev
The following NEW packages will be installed:
  libjasper-dev libjasper1 libjbig-dev libjpeg-dev libjpeg62-turbo-dev
  liblzma-dev libpng12-dev libtiff-dev libtiff5-dev libtiffxx5
0 upgraded, 10 newly installed, 2 to remove and 1 not upgraded.
Need to get 1,906 kB of archives.
After this operation, 1,305 kB of additional disk space will be used.
Do you want to continue? [Y/n] █
```

5. ทำการติดตั้งแพ็คเกจสำหรับการใช้งานไฟล์วีดีโอชนิดต่างๆ

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libx265-dev
```

```
pi@pi:~$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libx265-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpng-tools
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libavutil-dev libswresample-dev
Suggested packages:
  libx265-doc
The following NEW packages will be installed:
  libavcodec-dev libavformat-dev libavutil-dev libswresample-dev
  libswscale-dev libv4l-dev libx264-dev libx265-dev libxvidcore-dev
0 upgraded, 9 newly installed, 0 to remove and 1 not upgraded.
Need to get 14.0 MB of archives.
After this operation, 50.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

6. ทำการติดตั้งแพ็คเกจ GTK development library ทำให้สามารถใช้งาน HiGUI สำหรับสร้างแบบฟอร์มบนหน้าจอ

```
sudo apt-get install libgtk2.0-dev libgtk-3-dev
```

```
The following NEW packages will be installed:
  autoconf automake autopoint autotools-dev debhelper dh-autoreconf
  dh-strip-nondeterminism dwz gettext gir1.2-atspi-2.0 gir1.2-gtk-2.0
  gir1.2-harfbuzz-0.0 icu-devtools intltool-debian libarchive-cpio-perl
  libarchive-zip-perl libatk-bridge2.0-dev libatk1.0-dev libatspi2.0-dev
  libblkid-dev libcairo-script-interpreter2 libcairo2-dev libdbus-1-dev
  libdebelhelper-perl libegl-dev libegl1-mesa-dev libepoxy-dev libffi-dev
  libfile-stripnondeterminism-perl libfontconfig1-dev libfreetype6-dev
  libfribidi-dev libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-dev libgl-dev
  libgl1-mesa-dev libgl2.0-dev libglib2.0-dev-bin libglx-dev
  libgraphite2-dev libgtk-3-dev libgtk2.0-dev libharfbuzz-dev
  libharfbuzz-gobject0 libicu-dev libltdl-dev liblzma2-2 libmail-sendmail-perl
  libmount-dev libpango1.0-dev libpcre16-3 libpcre3-dev libpcre32-3
  libpcrecpp0v5 libpixman-1-dev libpng-dev libselinux1-dev libsep01-dev
  libsys-hostname-long-perl libtool libwayland-bin libwayland-dev
  libxcb-render0-dev libxcb-shm0-dev libcomposite-dev libxcursor-dev
  libxdamage-dev libxext-dev libxfixes-dev libxft-dev libxi-dev
  libxinerama-dev libxkbcommon-dev libxml2-utils libxrandr-dev libxrender-dev
  libxtst-dev m4 pango1.0-tools po-debconf uuid-dev wayland-protocols
  x11proto-composite-dev x11proto-damage-dev x11proto-fixes-dev
  x11proto-input-dev x11proto-randr-dev x11proto-record-dev x11proto-xext-dev
  x11proto-xinerama-dev
0 upgraded, 90 newly installed, 1 to remove and 1 not upgraded.
Need to get 30.2 MB of archives.
After this operation, 125 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

7. ติดตั้งแพ็คเกจสำหรับ optimize function ต่างๆ ช่วยคำนวณการคำนวณงานด้านแมทริกซ์

```
sudo apt-get install libatlas-base-dev gfortran
```

```
sudo apt-get install libhdf5-dev libhdf5-serial-dev python3-pyqt5
```

```
pi@pi:~$ sudo apt-get install libatlas-base-dev gfortran
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gfortran-8 libatlas3-base libgfortran-8-dev
Suggested packages:
  gfortran-doc gfortran-8-doc libgfortran5-dbg libcoarrays-dev libatlas-doc
  liblapack-doc
The following NEW packages will be installed:
  gfortran gfortran-8 libatlas-base-dev libatlas3-base libgfortran-8-dev
0 upgraded, 5 newly installed, 0 to remove and 1 not upgraded.
Need to get 12.8 MB of archives.
After this operation, 54.3 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

8. ก้าวใน Raspberry Pi ขั้งใหม่เมื่อ python 3 for development จะต้องติดตั้ง python3 ด้วยคำสั่ง

```
sudo apt-get install python3-dev
```

```
sudo apt-get install python3-pip
```

9. การดำเนินการติดตั้ง OpenCV จะต้องดาวน์โหลดซอฟต์แวร์โค้ด opencv กับ opencv\_contrib จาก github โดยสามารถดาวน์โหลดไฟล์และคลายชิปดังนี้

```
cd ~
```

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.5.3.zip
```

```
unzip opencv.zip
```

```
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.5.3.zip
```

```
unzip opencv_contrib.zip
```

```
pi@pi:~$ inflating: opencv_contrib-4.5.3/modules/xphoto/tutorials/images/baboon_oil_painting_effect.jpg
inflating: opencv_contrib-4.5.3/modules/xphoto/tutorials/images/originalVSdistorted.jpg
inflating: opencv_contrib-4.5.3/modules/xphoto/tutorials/images/reconstructed_fastVSbest.jpg
inflating: opencv_contrib-4.5.3/modules/xphoto/tutorials/oil_painting_effect.m
arkdown
inflating: opencv_contrib-4.5.3/modules/xphoto/tutorials/training_white_balance.markdown
creating: opencv_contrib-4.5.3/samples/
creating: opencv_contrib-4.5.3/samples/data/
inflating: opencv_contrib-4.5.3/samples/data/corridor.jpg
creating: opencv_contrib-4.5.3/samples/python2/
inflating: opencv_contrib-4.5.3/samples/python2/common.py
inflating: opencv_contrib-4.5.3/samples/python2/lsd_lines_extraction.py
inflating: opencv_contrib-4.5.3/samples/python2/seeds.py
inflating: opencv_contrib-4.5.3/samples/python2/video.py
pi@pi:~$
```

10. ในระหว่างกระบวนการติดตั้ง OpenCV อาจจะเกิดปัญหาล้มเหลวหรือเครื่องค้างแสงค่าไม่ทำงานต่อ เพื่อหลีกเลี่ยงปัญหานี้จำเป็นต้องเพิ่มขนาด Swap ซึ่งเป็นไฟล์ที่ทำหน้าที่รับข้อมูลบนดิสก์แทนหน่วยความจำ RAM ที่มีขนาดไม่เพียงพอที่จะจัดเก็บข้อมูล โดยค่าเริ่มต้นของการแยกเปลี่ยนข้อมูลที่เก็บไว้บน Raspberry Pi ตั้งไว้ให้มีขนาด 100 Mb อาจจะเลือกไป ซึ่งสามารถแก้ไขได้โดยใช้คำสั่ง

```
sudo nano /etc/dphys-swapfile
```

ให้แก้ไขตัวแปร CONF\_SWAPSIZE ใน swapfile โดยให้เปลี่ยนค่า CONF\_SWAPSIZE = 100 ไปเป็น CONF\_SWAPSIZE = 2048 เมื่อเปลี่ยนการกำหนดค่า swapfile แล้วจำเป็นที่จะต้องรีสตาร์ทระบบโดยใช้คำสั่ง

```
sudo systemctl restart dphys-swapfile
```

หลังจากที่ได้ติดตั้ง OpenCV เรียบร้อยแล้วให้กลับมาแก้ไขขนาดของ swapfile ให้เหมือนเดิมด้วย

11. การติดตั้ง Python virtual environment ( ถ้าไม่ต้องการติดตั้งส่วนนี้ให้ข้ามขั้นตอนที่ 11 ทั้งหมด และในขั้นตอนตั้งแต่ ข้อที่ 12 เป็นต้นไปไม่ต้องทำคำสั่งในส่วนที่เป็น **อักษรสีแดง** ที่ถูกกรองขึ้นจาก virtual environment ของข้อที่ 11 นี้ )  
Python virtual environment จะเป็นเครื่องมือที่ใช้แยกสภาพแวดล้อมเสมือนในแต่ละงานหรือแต่ละ โครงการของ Python ออกจากกัน อาจจะเป็นเพราะไม่ต้องการให้เกิดปัญหาการรับกันของ library เมื่อใช้ Python ต่างเวอร์ชัน กัน ซึ่งใน virtual environment สามารถช่วยแก้ไขปัญหานี้ได้ ในการนี้ของ OpenCV จะมีความสะดวกเมื่อจำเป็นต้อง ลบ OpenCV เวอร์ชันที่ใช้อยู่แล้วเปลี่ยนไปติดตั้ง OpenCV เวอร์ชันใหม่

- 11.1 ก่อนที่จะคอมไพล์ OpenCV บน Raspberry Pi ต้องอัพเดท pip ซึ่งเป็นตัวจัดการแพ็คเกจ (package installer/manager) ของ Python เพื่อลง library ต่างๆจาก PYPI (Python Package Index) ด้วยคำสั่ง

```
wget https://bootstrap.pypa.io/get-pip.py
```

```
sudo python3 get-pip.py
```

```
pi@pi:~$ wget https://bootstrap.pypa.io/get-pip.py
--2021-10-04 19:06:48-- https://bootstrap.pypa.io/get-pip.py
Resolving bootstrap.pypa.io (bootstrap.pypa.io)... 199.232.44.175, 2a04:4e42:48::175
Connecting to bootstrap.pypa.io (bootstrap.pypa.io)|199.232.44.175|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1946810 (1.9M) [text/x-python]
Saving to: 'get-pip.py'

get-pip.py      100%[=====] 1.86M 3.82MB/s   in 0.5s

2021-10-04 19:06:49 (3.82 MB/s) - 'get-pip.py' saved [1946810/1946810]

pi@pi:~$ sudo python3 get-pip.py
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pip
  Downloading https://www.piwheels.org/simple/pip/pip-21.2.4-py3-none-any.whl (1.6 MB)
|██████████| 1.6 MB 158 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 18.1
    Uninstalling pip-18.1:
      Successfully uninstalled pip-18.1
Successfully installed pip-21.2.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
pi@pi:~$
```

- 11.2 ติดตั้งตัวจัดการ virtual environment ด้วยคำสั่ง

```
sudo pip install virtualenv virtualenvwrapper
```

```
sudo rm -rf ~/.cache/pip
```

```

pi@pi:~$ sudo rm -rf ~/.cache/pip
pi@pi:~$ Collecting typing-extensions>=3.6.4
  Downloading https://www.piwheels.org/simple/typing-extensions/typing_extensions-3.10.0.2-py3-none-any.whl (27 kB)
pi@pi:~$ Collecting pbr!=2.1.0,>=2.0.0
  Downloading https://www.piwheels.org/simple/pbr/pbr-5.6.0-py2.py3-none-any.whl (111 kB)
|██████████| 111 kB 152 kB/s
pi@pi:~$ Installing collected packages: zipp, typing-extensions, importlib-metadata, platformdirs, pbr, filelock, distlib, backports.entry-points-selectable, virtualenv-clone, virtualenv, stevedore, virtualenvwrapper
pi@pi:~$ Successfully installed backports.entry-points-selectable-1.1.0 distlib-0.3.3 filelock-3.3.0 importlib-metadata-4.8.1 pbr-5.6.0 platformdirs-2.4.0 stevedore-3.4.0 typing-extensions-3.10.0.2 virtualenv-20.8.1 virtualenv-clone-0.5.7 virtualenvwrapper-4.8.4 zipp-3.6.0
pi@pi:~$ WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
pi@pi:~$ sudo rm -rf ~/.cache/pip
pi@pi:~$ 

```

11.3 เพิ่มข้อความจำนวน 4 บรรทัดเข้าไปในโปรไฟล์ผู้ใช้งานเก็บไว้ที่ `~/profile` โดยใช้คำสั่ง

```

echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.profile
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile

```

```

pi@pi:~$ sudo rm -rf ~/.cache/pip
pi@pi:~$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
pi@pi:~$ echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
pi@pi:~$ echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.profile
pi@pi:~$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile
pi@pi:~$ 

```

เมื่อได้อัพเดตโปรไฟล์แล้วจำเป็นต้องโหลดใหม่ เพื่อให้การเปลี่ยนแปลงมีผล โดยให้เริ่มต้นระบบใหม่ด้วยคำสั่ง

```
sudo reboot
```

11.4 เปิดเทอร์มินัลและป้อนคำสั่งเพื่อเข้าสู่ระบบ

```
source ~/.profile
```

```

pi@pi:~$ source ~/.profile
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/premkproject
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/postmkproject
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/initialize
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/premkvirtualenv
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/postmkvirtualenv
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/prermvirtualenv
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/postrmvirtualenv
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/predeactivate
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/postdeactivate
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/preactivate
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/postactivate
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/get_env_details
pi@pi:~$ 

```

11.5 ให้ดาวน์โหลดลิ้มแม่ข่าย OpenCV ชื่อ cv

```
mkvirtualenv cv -p python3
```

```
pi@pi:~ $ mkvirtualenv cv -p python3
created virtual environment CPython3.7.3.final.0-32 in 3826ms
  creator CPython3Posix(dest=/home/pi/.virtualenvs/cv, clear=False, no_vcs_ignor
e=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle
, via=copy, app_data_dir=/home/pi/.local/share/virtualenv)
    added seed packages: pip==21.2.4, setuptools==58.1.0, wheel==0.37.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/cv/bin/predeactiva
te
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/cv/bin/postdeactivat
e
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/cv/bin/preactivate
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/cv/bin/postactivat
e
virtualenvwrapper.user_scripts creating /home/pi/.virtualenvs/cv/bin/get_env_det
ails
(cv) pi@pi:~ $
```

เมื่อสภาพแวดล้อมเปลี่ยน ได้รับการติดตั้งอย่างถูกต้องแล้ว ให้เริ่มต้นระบบใหม่อีกรั้ง

sudo reboot

11.6 สามารถใช้ `workon` ให้อัญญาติสภาพแวดล้อมเสมือนโดยจะมีคำนำหน้าว่า (`cv`) ที่ดำเนินการนั้นๆ ตั้งแต่ต้นของบรรทัดคำสั่งด้านล่าง

```
source ~/profile
```

workon cv

```
pi@pi: ~
pi@pi:~$ source ~/.profile
pi@pi:~$ workon cv
(cv) pi@pi:~$
```

ผู้ต้องการให้ความสนใจ virtual environment ให้ใช้คำสั่ง

## deactivate

ถ้าต้องการให้ workon cv เริ่มต้นทำงานโดยอัตโนมัติทุกรัชที่เข้ามาโหลด virtual environment ให้ไปเพิ่ม workon cv ต่อท้ายในไฟล์ .profile

12. ติดตั้งแพ็คเกจnumpyและNumPy ใช้ในการดำเนินการครัวเรือนใน Python

pip install numpy (สำหรับ Ubuntu ให้ใช้ sudo apt-get install python3-numpy )

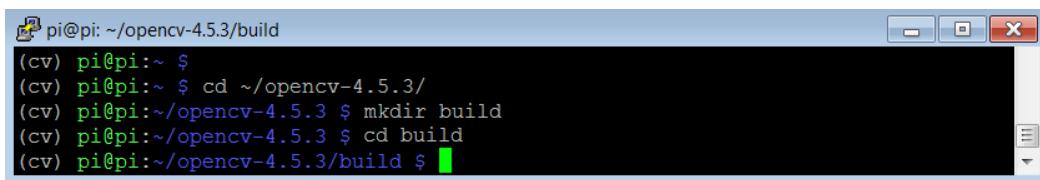
```
pi@pi: ~
pi@pi:~$ source ~/.profile
pi@pi:~$ workon cv
(cv) pi@pi:~$ pip3 install numpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting numpy
  Downloading https://www.piwheels.org/simple/numpy/numpy-1.21.2-cp37-cp37m-linu
x_armv7l.whl (12.3 MB)
|████████████████████████████████████████████████████████████████████████████████| 12.3 MB 12 kB/s
Installing collected packages: numpy
Successfully installed numpy-1.21.2
(cv) pi@pi:~$
```

13. ให้ไปที่โฟลเดอร์เป็นชอร์สโค้ด OpenCV และสร้างโฟลเดอร์ใหม่ชื่อ build เพื่อที่จะใช้ในการคอมไพล์และติดตั้ง OpenCV ด้วยคำสั่ง

```
cd ~/opencv-4.5.3/
```

```
mkdir build
```

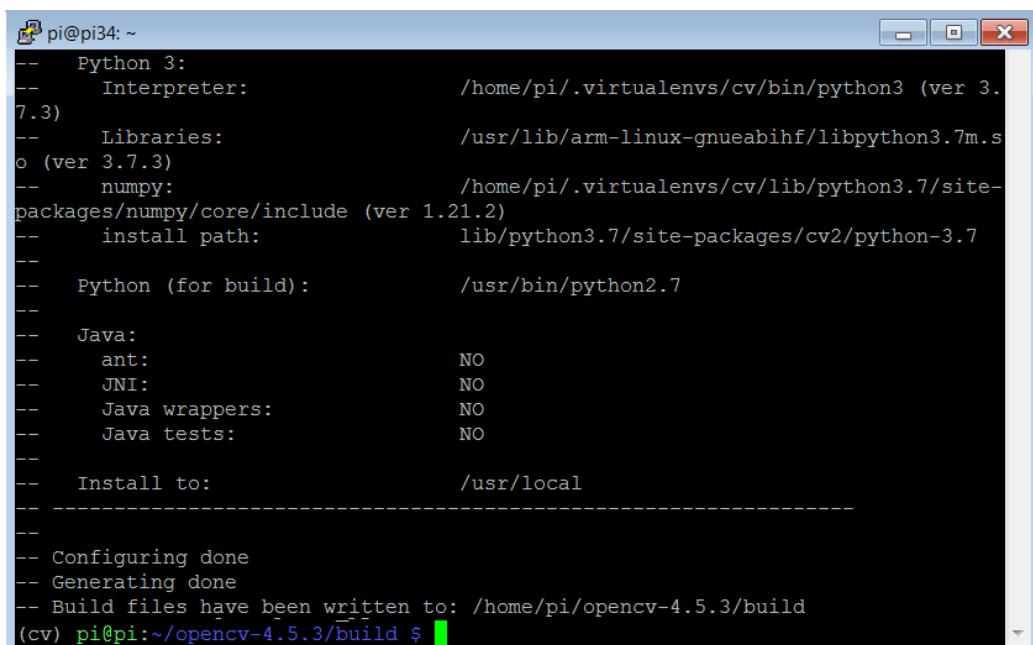
```
cd build
```



```
pi@pi: ~/opencv-4.5.3/build
(cv) pi@pi:~ $ 
(cv) pi@pi:~ $ cd ~/opencv-4.5.3/
(cv) pi@pi:~/opencv-4.5.3 $ mkdir build
(cv) pi@pi:~/opencv-4.5.3 $ cd build
(cv) pi@pi:~/opencv-4.5.3/build $
```

14. ตั้งค่าการสร้าง CMake โดยใส่คำสั่งทั้งหมดนี้พร้อมกัน

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \ -D CMAKE_INSTALL_PREFIX=/usr/local \ -D
INSTALL_C_EXAMPLES=ON \ -D INSTALL_PYTHON_EXAMPLES=ON \ -D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-4.5.3/modules \ -D BUILD_EXAMPLES=ON ..
```



```
pi@pi34: ~
-- Python 3:
--   Interpreter:          /home/pi/.virtualenvs/cv/bin/python3 (ver 3.7.3)
--   Libraries:            /usr/lib/arm-linux-gnueabihf/libpython3.7m.so (ver 3.7.3)
--   numpy:                /home/pi/.virtualenvs/cv/lib/python3.7/site-packages/numpy/core/include (ver 1.21.2)
--   install path:         lib/python3.7/site-packages/cv2/python-3.7
-- 
-- Python (for build):    /usr/bin/python2.7
-- 
-- Java:
--   ant:                  NO
--   JNI:                  NO
--   Java wrappers:        NO
--   Java tests:           NO
-- 
-- Install to:             /usr/local
-- 
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/opencv-4.5.3/build
(cv) pi@pi:~/opencv-4.5.3/build $
```

15. ขั้นตอนการสร้าง OpenCV จะตามด้วย j4 ซึ่งมีหมายความว่าให้สร้างโดยใช้ทั้ง 4 แกนของ CPU ใน Raspberry Pi เพื่อจะเร่งขั้นตอนให้เร็วขึ้น โดยจะใช้เวลาประมาณ 2 ชั่วโมง ด้วยคำสั่ง

```
make -j4
```

```
[ 99%] Built target opencv_test_objdetect
[100%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf_main.cpp.o
[100%] Building CXX object modules/videostab/CMakeFiles/opencv_test_videostab.dir/test_motion_estimation.cpp.o
[100%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf_matchers.cpp.o
[100%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf_stich.cpp.o
[100%] Linking CXX executable ../../bin/opencv_test_videostab
[100%] Built target opencv_test_videostab
Scanning dependencies of target opencv_python2
[100%] Building CXX object modules/python2/CMakeFiles/opencv_python2.dir/_src2/cv2.cpp.o
Scanning dependencies of target opencv_python3
[100%] Building CXX object modules/python3/CMakeFiles/opencv_python3.dir/_src2/cv2.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_stitching
[100%] Built target opencv_perf_stitching
[100%] Linking CXX shared module ../../lib/cv2.so
[100%] Linking CXX shared module ../../lib/python3/cv2.cpython-37m-arm-linux-gnueabihf.so
[100%] Built target opencv_python3
[100%] Built target opencv_python2
(pi@pi:~/opencv-4.5.3/build $
```

16. เมื่อเสร็จแล้วก็นำไฟนารีติดตั้งที่สร้างขึ้นไปยังไฟล์เดอร์ของ Python

```
sudo make install
```

```
sudo ldconfig
```

กรณีที่ต้องการลบการติดตั้ง OpenCV ของเดิมออกให้ใช้คำสั่ง

```
sudo make uninstall
```

```
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalface_improved.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_profileface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_silverware.xml
-- Installing: /usr/local/bin/opencv_annotation
-- Set runtime path of "/usr/local/bin/opencv_annotation" to "/usr/local/lib"
-- Installing: /usr/local/bin/opencv_visualisation
-- Set runtime path of "/usr/local/bin/opencv_visualisation" to "/usr/local/lib"
-- Installing: /usr/local/bin/opencv_interactive-calibration
-- Set runtime path of "/usr/local/bin/opencv_interactive-calibration" to "/usr/local/lib"
-- Installing: /usr/local/bin/opencv_version
-- Set runtime path of "/usr/local/bin/opencv_version" to "/usr/local/lib"
-- Installing: /usr/local/bin/opencv_model_diagnostics
-- Set runtime path of "/usr/local/bin/opencv_model_diagnostics" to "/usr/local/lib"
(pi@pi:~/opencv-4.5.3/build $ sudo ldconfig
(pi@pi:~/opencv-4.5.3/build $
```

17. ข้อนี้ทำเฉพาะในกรณีที่เป็นการติดตั้งจาก virtual environment จะเห็นว่ามีไฟล์ cv2.cpython-37m-arm-linux-gnueabihf.so ปรากฏขึ้นในไฟล์เดอร์การทำงานของ Python ให้เปลี่ยนชื่อเป็น cv2

```
cd /usr/local/lib/python3.7/site-packages/cv2/python-3.7
```

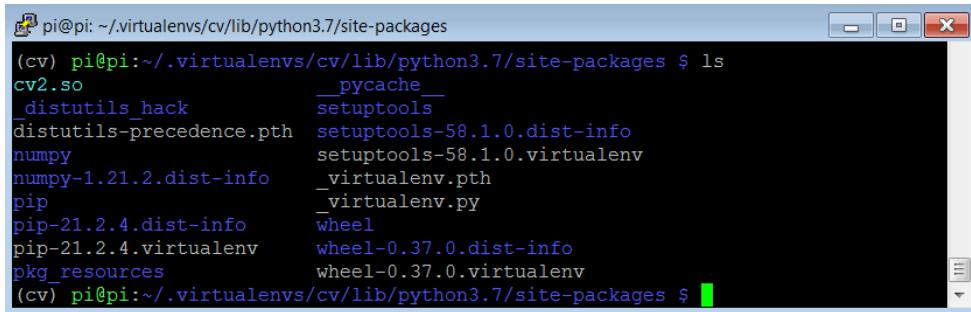
```
sudo mv cv2.cpython-37m-arm-linux-gnueabihf.so cv2.so
```

หลังจากนั้นให้สร้าง symbolic link จากไฟล์ cv2.so ไฟล์ใหม่ที่เรียกว่า cv2 ดังนี้

```
cd ~/virtualenvs/cv/lib/python3.7/site-packages/
```

```
ln -s /usr/local/lib/python3.7/site-packages/cv2/python-3.7/cv2.so cv2.so
```

ดังนั้นจะสามารถใช้ OpenCV ใน virtual environment คือ `~/virtualenvs/cv/lib/python3.7/site-packages/` เพื่อเชื่อมโยงไปยังไฟล์ `cv2.so` ที่อยู่ใน `/usr/local/lib/python3.7/site-packages/cv2/python-3.7` เมื่อใช้คำสั่ง `ls` จะเห็นชื่อไฟล์ `cv2.so` คือ link เป็นลิ้ฟ์



```
pi@pi: ~/virtualenvs/cv/lib/python3.7/site-packages
(cv) pi@pi:~/virtualenvs/cv/lib/python3.7/site-packages $ ls
cv2.so          _pycache_
_distutils_hack    setuptools
distutils-precedence.pth  setuptools-58.1.0.dist-info
numpy           setuptools-58.1.0.virtualenv
numpy-1.21.2.dist-info _virtualenv.pth
pip             _virtualenv.py
pip-21.2.4.dist-info wheel
pip-21.2.4.virtualenv wheel-0.37.0.dist-info
pkg_resources   wheel-0.37.0.virtualenv
(cv) pi@pi:~/virtualenvs/cv/lib/python3.7/site-packages $
```

18. ตรวจสอบการทำงานว่าได้ติดตั้งอย่างถูกต้องหรือไม่ โดยการเข้าสู่ virtual environment และเรียก Python 3

```
source ~/.profile
```

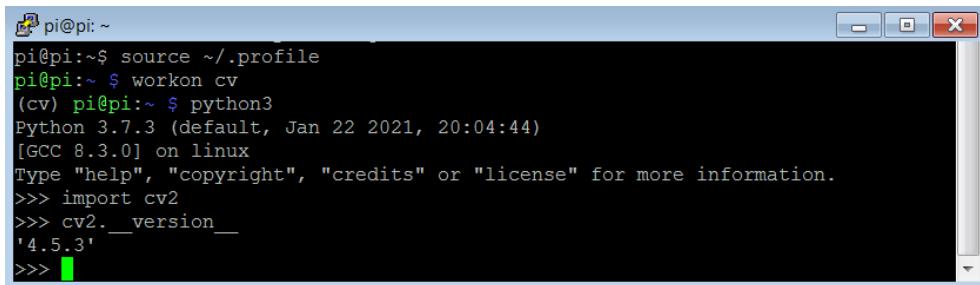
```
workon cv
```

```
python3
```

ทดลองนำเข้าโมดูล `cv2` และตรวจสอบเวอร์ชันของ OpenCV โดยใช้คำสั่ง

```
import cv2
```

```
cv2.__version__
```



```
pi@pi: ~
pi@pi:~$ source ~/.profile
pi@pi:~$ workon cv
(cv) pi@pi:~$ python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.5.3'
>>>
```

กรณีต้องการติดตั้งแพ็คเกจทั้งหมดที่จำเป็นในการเรียกใช้โปรแกรม OpenCV บนเครื่อง Odroid ที่ใช้ระบบปฏิบัติการ Ubuntu 20.04 สามารถทำได้โดยใช้คำสั่ง

```
sudo apt install libopencv-dev python3-opencv
```

ตรวจสอบการติดตั้ง โดยการนำเข้าโมดูล `cv2` จะได้ OpenCV เป็นเวอร์ชัน 4.2

หรืออาจใช้คำสั่ง `pip` ในการติดตั้ง OpenCV แต่จะต้องแก้ไขขนาดของ swap ตามข้อ 10 ก่อน แล้วติดตั้งด้วยคำสั่งดังนี้

```
pip install opencv-contrib-python --no-cache-dir
```

จะได้ OpenCV เป็นเวอร์ชัน 4.6

กรณีต้องการติดตั้งโปรแกรม OpenCV เพื่อที่จะทำการทดลองบนเครื่องคอมพิวเตอร์ที่เป็นระบบปฏิบัติการ Microsoft Windows หรือ macOS สามารถทำได้โดยการติดตั้งโปรแกรมภาษา Python ก่อน ให้ดาวน์โหลดเพื่อทำการติดตั้งได้ที่ <https://www.python.org/downloads/> เมื่อดาวน์โหลดแล้วให้ทำการติดตั้งโปรแกรม และคลิกเลือก Add Python to PATH



หลังจากนั้นให้ทำการติดตั้ง OpenCV โดยมีวิธีการติดตั้งได้ 2 แบบ ดังนี้

- Building OpenCV from source คือ การสร้าง OpenCV จาก source ขึ้นมาจากเครื่อง เนื่องจากต้องใน Raspberry PI
- Installing OpenCV-Python from Pre-built Binaries คือ การติดตั้ง OpenCV ที่สร้างสำเร็จเตรียมไว้แล้ว ไม่ต้องสร้างขึ้นมาใหม่ โดยการติดตั้งในที่นี่จะใช้ wheel file ซึ่งจะถูกบีบอัดไว้เป็นรูปแบบแพ็คเกจมาตรฐาน PEP 427 ที่ใช้สำหรับการแจกจ่าย สามารถทำการดาวน์โหลดไฟล์ในนารี OpenCV เพื่อนำมาติดตั้งได้จาก <https://pypi.org/project/opencv-python/#files> หรือที่ <https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv> ให้ดาวน์โหลด OpenCV เวอร์ชันล่าสุด โดยให้เลือกรุ่นที่ตรงกันกับ Python เวอร์ชันที่ติดตั้งไว้แล้ว และนำไฟล์ที่ดาวน์โหลดไปไว้ที่โฟลเดอร์ที่ติดตั้ง Python เช่น C:\Program Files\Python37

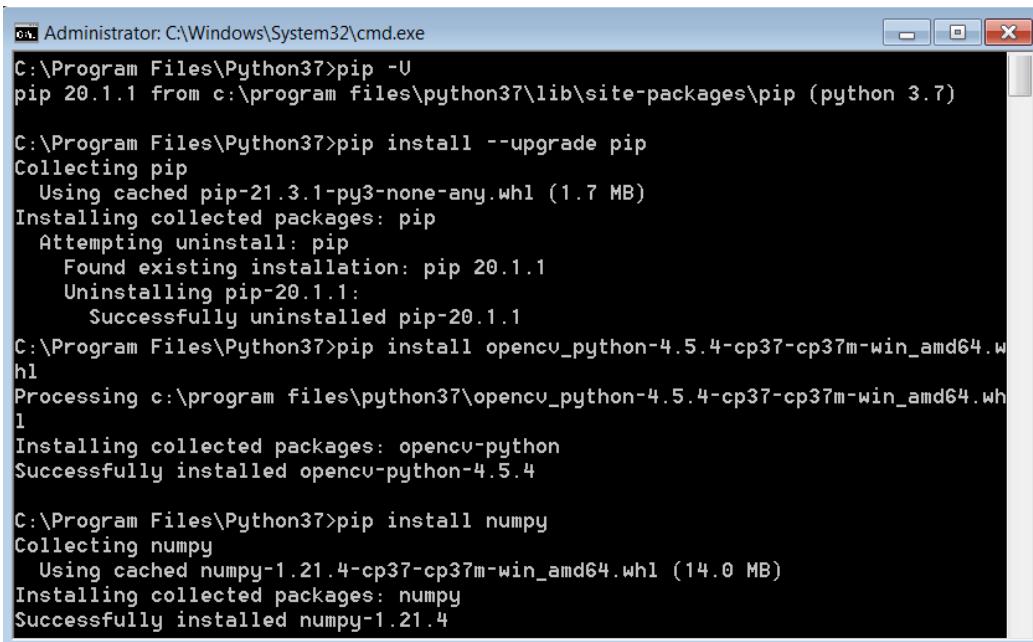
การติดตั้งโปรแกรมเสริมอื่นในภาษา Python ให้เรียกใช้ Command Prompt ของ Windows โดยให้พิมพ์ CMD บนแถบ Search เพื่อค้นหาแล้วให้เลือก Run as administrator นอกจากมอคูลาตรฐานที่มีอยู่แล้ว rekton ติดตั้งภาษา Python แล้วยังจำเป็นต้องติดตั้งมอคูลาเสริม (แพ็คเกจเสริม) เพิ่มเติมที่จำเป็นสำหรับใช้งานที่ต้องการ วิธีการลงมอคูลาใหม่จะใช้คำสั่ง pip

ถ้าต้องการดูว่าอร์ชันของ pip ที่ติดตั้งไว้แล้วตอนติดตั้งภาษา Python ให้ใช้คำสั่ง pip -V

ถ้าต้องการปรับปรุงเวอร์ชันของ pip ให้ใช้คำสั่ง pip install --upgrade pip

ให้ไปยังโฟลเดอร์ที่ติดตั้ง Python ไว้ เช่น cd C:\Program Files\Python37 และทำการติดตั้งโมดูล OpenCV รุ่นที่ดาวน์โหลดมาไว้ โดยใช้คำสั่ง pip install ตามด้วยชื่อแพ็คเกจ เช่น

```
pip install opencv_python-4.5.4-cp37-cp37m-win_amd64.whl
```



```

Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Python37>pip -U
pip 20.1.1 from c:\program files\python37\lib\site-packages\pip (python 3.7)

C:\Program Files\Python37>pip install --upgrade pip
Collecting pip
  Using cached pip-21.3.1-py3-none-any.whl (1.7 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.1.1
    Uninstalling pip-20.1.1:
      Successfully uninstalled pip-20.1.1
C:\Program Files\Python37>pip install opencv_python-4.5.4-cp37-cp37m-win_amd64.whl
Processing c:\program files\python37\opencv_python-4.5.4-cp37-cp37m-win_amd64.whl
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.4

C:\Program Files\Python37>pip install numpy
Collecting numpy
  Using cached numpy-1.21.4-cp37-cp37m-win_amd64.whl (14.0 MB)
Installing collected packages: numpy
Successfully installed numpy-1.21.4

```

ให้ติดตั้งแพ็กเกจเสริมที่จำเป็นต่อการใช้งาน OpenCV เพื่อใช้ในการทดลอง ด้วยคำสั่งดังนี้

pip install numpy

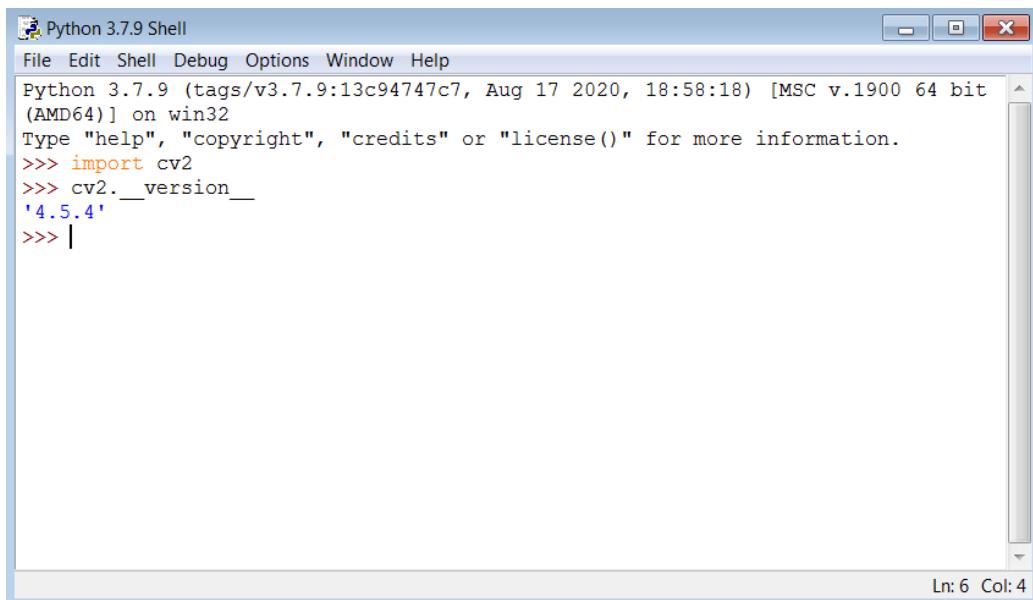
pip install imutils

pip install requests

เมื่อการติดตั้งเสร็จเรียบร้อยแล้ว ตอนนี้เราสามารถทดสอบได้โดยเรียกใช้ IDLE Python 3.7 ทดลองนำเข้าไฟล์รารี cv2 สำหรับ OpenCV และตรวจสอบเวอร์ชันของ OpenCV โดยใช้คำสั่ง

import cv2

cv2.\_\_version\_\_



```

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> cv2.__version__
'4.5.4'
>>>

```

ถ้าต้องการคลิกเลือกไฟล์ภาษา Python จาก Windows Explorer เพื่อไปเรียกใช้ใน IDLE Python 3.7 ให้ไปที่ชื่อไฟล์นั้น แล้วให้เลือกเปิดด้วยโปรแกรม C:\Program Files\Python37\Lib\idlelib\idle.bat

วิธีอัพเกรด Python 3.6.9 บน Ubuntu 18.04 LTS

เริ่มด้วยการตรวจสอบ Python เวอร์ชันที่ติดตั้งอยู่ในระบบด้วยคำสั่ง

```
python3 -V
```

จะเห็นรุ่น Python ที่ติดตั้งอยู่ในเครื่อง Odroid C1+ เป็นรุ่น 3.6.9 ให้ติดตั้ง Python 3.7.12 หรือรุ่นที่สูงกว่า สำหรับ Ubuntu เพิ่มที่เก็บ deadsnakes/ppa โดยใช้คำสั่ง

```
sudo add-apt-repository ppa:deadsnakes/ppa
```

ปรับปรุงรายการแพ็คเกจ โดยใช้คำสั่ง

```
sudo apt-get update
```

ตรวจสอบรายการแพ็คเกจ Python ที่จะปรับปรุงโดยใช้คำสั่ง

```
apt list | grep python3.7
```

ติดตั้ง Python 3.7 โดยใช้คำสั่ง

```
sudo apt-get install python3.7
```

เพิ่ม Python ทั้งเวอร์ชันเก่าและใหม่เพื่ออปเดตทางเลือก

```
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.6 1
```

```
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 2
```

ค่าเริ่มต้น Python3 จะซื้อไปที่ Python 3.6 นั่นหมายความว่าเมื่อเรียกใช้ python3 มันจะดำเนินการเป็น python3.6 ดังนั้นจึงต้องอัพเดตให้ Python 3 ซื้อไปที่ Python 3.7 ให้พิมพ์คำสั่งเพื่อกำหนดค่า python3

```
sudo update-alternatives --config python3
```

ให้เลือกหมายเลขที่ใช้สำหรับ Python 3.7 โดยพิมพ์เลือก 2 สำหรับเข้า Python 3.7

```
root@odroid:~# sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.6 1
update-alternatives: using /usr/bin/python3.6 to provide /usr/bin/python3 (python3) in auto mode
root@odroid:~# sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 2
update-alternatives: using /usr/bin/python3.7 to provide /usr/bin/python3 (python3) in auto mode
root@odroid:~# sudo update-alternatives --config python3
There are 2 choices for the alternative python3 (providing /usr/bin/python3).

      Selection    Path          Priority   Status
      -----      ---          ---        ---
* 0            /usr/bin/python3.7    2        auto mode
    1            /usr/bin/python3.6    1        manual mode
    2            /usr/bin/python3.7    2        manual mode

Press <enter> to keep the current choice[*], or type selection number: [Enter]
```

ทดสอบเวอร์ชันของ python ด้วยคำสั่ง

```
python3 -V
```

ถ้าต้องการขอนกลับให้ลบ python 3.7 ด้วยคำสั่ง

```
sudo apt autoremove python3.7 --purge
```

แล้วให้ลบ Nightly build PPA ด้วยคำสั่ง

```
sudo add-apt-repository --remove ppa:deadsnakes/nightly
```

1. ให้ทดลองเปิดไฟล์รูปภาพชื่อ picture.jpg ด้วยโปรแกรมดังนี้

```
import cv2
img = cv2.imread('picture.jpg')                      # reading from jpg file
cv2.imshow('My photo', img)                          # show the image
cv2.waitKey(0)                                       # check press key
cv2.destroyAllWindows()                                # close any open windows
```

2. ให้ทดลองเปิด Video จากกล้อง USB Webcam และจะปิดเมื่อกด spacebar ด้วยโปรแกรมดังนี้

```
import cv2
import numpy
cap = cv2.VideoCapture(0)                            # initialize the camera
while True:
    ret, frame = cap.read()                         # reading from webcam

    cv2.imshow('Frame BGR',frame)                   # show the frame BGR
    if cv2.waitKey(1) == 32:                         # check press spacebar
        break
cap.release()                                       # cleanup the camera
cv2.destroyAllWindows()                                # close any open windows
```

3. จากข้อ 2 ให้ทดลองเพิ่มโปรแกรมเพื่อเปลี่ยนให้ Video เป็นภาพขาวดำ ซึ่งจะเป็นระดับสีเทา (Gray) ดังนี้

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)          # Converting color image to gray
cv2.imshow('Frame Gray',gray)                          # show the frame gray
```

4. จากข้อ 3 ให้ทดลองแก้ไขโปรแกรมเพื่อเปลี่ยนค่า COLOR\_BGR2GRAY ไปเป็น COLOR\_BGR2RGB และ COLOR\_BGR2HSV และขออธิบายผลที่ได้ .....
- .....
- .....
- .....

5. จากข้อ 2 ให้ทดลองเพิ่มโปรแกรมเพื่อใช้คำสั่ง cv2.split() แยกสีของภาพออกเป็นลิสต์ของแต่ละค่าสี และใช้คำสั่ง cv2.merge() ไว้รวมลิสต์ของแต่ละค่าสีเข้าเป็นอวาร์เรย์ของภาพ โดยให้ออก 2 สีเป็นสูนย์ ซึ่งจะได้ภาพที่มีแม่สี 3 สีแยกกัน ดังโปรแกรมด้านล่าง และให้เขียนโปรแกรมเพิ่มเพื่อแสดงภาพที่เป็น video ของแม่สีทั้งสาม

```
b,g,r = cv2.split(frame)
z = numpy.zeros_like(b)
b = cv2.merge([b,z,z])
g = cv2.merge([z,g,z])
r = cv2.merge([z,z,r])
```

6. จากข้อ 5 ให้ทดลองเพิ่มโปรแกรม mask ด้านล่าง และให้ทดลองแก้ไขค่าตัวเลข

```
lower = numpy.array([127,0,0])
upper = numpy.array([255,255,255])
mask = cv2.inRange(frame, lower, upper)

cv2.imshow('Frame MASK',mask)
```

7. ถ้าต้องการประมวลผลเลือกเฉพาะบางส่วนในภาพ ROI (Region of interest) โดยการตรวจจับรูปทรง ROI ที่ตรวจพบ ซึ่งภาพที่นำมาประมวลผลนั้นจะเก็บข้อมูลอยู่ในรูปแบบทริกซ์ การทดลองนี้จะเป็นตัวอย่างการตัดเฉพาะบางส่วนตามที่กำหนดในกรอบที่ระบุไว้ เพื่อที่จะนำข้อมูลไปใช้ในการประมวลผลของข้อต่อๆไป เริ่มจากเอาโปรแกรมในข้อ 1 ที่เป็นภาพ PNG มาแยกเป็นแม่สี 3 สี แล้วกำหนดตำแหน่งตามค่าข้อมูลในแม่ทริกซ์ ด้วยโปรแกรมดังนี้

```

import cv2
BGR = cv2.imread('BGRcolor.png') # reading from PNG file

cv2.imshow('BGR Frame', BGR) # show the image
cv2.imshow('B Frame', BGR[ 0:500, 0:300, 0]) # y1:y2 , x1:x2 , color
cv2.imshow('G Frame', BGR[200:700, 300:600, 1])
cv2.imshow('R Frame', BGR[400:900, 600:900, 2])
cv2.waitKey(0) # check press key
cv2.destroyAllWindows() # close any open windows

```

8. จากโปรแกรมในข้อ 7 ให้ทดลองการทำงานของโปรแกรม แล้วทดลองแก้ไขตัวเลข หลังจากนั้นให้อธิบายผลการทดลองว่าคำสั่งที่ใช้กำหนดค่าข้อมูลในแม่ทริกซ์มีความสัมพันธ์กับค่า Pixel ของแกน X และ Y ในรูปภาพอย่างไร
- .....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

9. จากโปรแกรมในข้อ 7 ให้ทดลองเพิ่มคำสั่ง  $BGR[:, :, 1] = 0$  หลังคำสั่ง  $BGR = cv2.imread('BGRcolor.png')$  และให้ทดลองการทำงานของโปรแกรม แล้วให้แก้ไขตัวเลข 0 และ 1 ในคำสั่งเป็นค่าอื่น และให้อธิบายผลการทดลองว่าคำสั่งที่ใช้นี้มีความสัมพันธ์ระหว่างค่าแม่ทริกซ์ และรูปภาพอย่างไร
- .....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

10. จากโปรแกรมในข้อ 9 ให้ทดลองแก้ไขเลข 1 เป็น  $[0, 1]$  และให้ทดลองการทำงานของโปรแกรม แล้วให้แก้ไขตัวเลข เป็นค่าอื่น และอธิบายผลการทดลองว่าคำสั่งที่ใช้นี้มีความสัมพันธ์กับแม่สีของรูปภาพอย่างไร
- .....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

11. จากโปรแกรมในข้อ 7 จะทำการแก้ไขโปรแกรมเพื่อนำเอาเฉพาะบางส่วนในภาพ ROI ที่เป็นสัญญาณวิดีโอจากกล้อง USB Webcam แทนการใช้ภาพถ่ายที่มีค่าแกน X และแกน Y คงที่ โดยที่การเขียนโปรแกรมจะต้องมีการกำหนดตัวแปรค่าความสูงและความกว้างของสัญญาณภาพ ซึ่งจะต้องดึงเอาค่าความสูงและความกว้างของภาพจากสัญญาณวิดีโอด้วยแล้วใช้โปรแกรมร่วมกับในข้อที่ 2 สามารถเขียนโปรแกรมได้ใหม่ดังนี้

```

import cv2
cap = cv2.VideoCapture(0)
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
Lo = int(height * 30/100)
Hi = int(height * 70/100)
L = int(width * 35/100)
R = int(width * 65/100)

while(True):
    ret, BGR = cap.read() # reading from webcam
    cv2.imshow('BGR Frame',BGR) # show the frame BGR
    cv2.imshow('B Frame',BGR[ 0:Lo,      0:L,      0]) # y1:y2 , x1:x2 , color
    cv2.imshow('G Frame',BGR[Lo:Hi,      L:R,      1])
    cv2.imshow('R Frame',BGR[Hi:height, R:width,2])
    if cv2.waitKey(1) == 32: # Check press spacebar
        break
cap.release() # cleanup the camera
cv2.destroyAllWindows() # close any open windows

```

12. จากโปรแกรมในข้อ 11 ให้ทดลองการทำงานของโปรแกรม แล้วให้อธิบายว่าค่าแม่ทริกซ์ ที่ใช้ในการเก็บข้อมูลสัญญาณ วิดีโอเป็นอย่างไร และมีความแตกต่างกับค่าแม่ทริกซ์ ที่ใช้ในการเก็บข้อมูลภาพถ่ายชั่วคราวในโปรแกรมในข้อ 7 อย่างไร

13. ถ้าต้องการสร้างโปรแกรมที่ทำหน้าที่ตรวจนับการเข้าออกของคนหรือสินค้า โดยใช้ Sensor จำนวน 2 ตัว ในการตรวจจับสามารถใช้สัญญาณภาพจากกล้อง แล้วกำหนดเส้น 2 เส้นบนภาพนั้นแทนเพื่อเป็น Sensor ตรวจจับความเคลื่อนไหวบนเส้นนั้น เริ่มจากเอาโปรแกรมในข้อ 3 และเพิ่มในส่วนของการตั้งค่า Sensor ที่จะใช้ในโปรแกรม โดยที่ช่องว่างในแต่ละบรรทัดของโปรแกรมจะไว้สำหรับใส่โปรแกรมเพิ่มเติมในข้อต่อไป แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```

import cv2
import numpy as np
countLeft = 0
countRight = 0
tolerance = 2
iframe = 0
diffLeftList = []
diffRightList = []
Left = False
Right = False
prevLeft = False
prevRight = False
cap = cv2.VideoCapture(0) # initialize the camera

while True:
    ret, frame = cap.read() # loop over the frames of the video
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # reading from webcam
    cv2.imshow('Frame',frame) # Converting color image to gray scale image
    iframe += 1 # show the frame

    if cv2.waitKey(1) == 32: # if the spacebar key is pressed, break from the loop
        break
    cap.release() # cleanup the camera
    cv2.destroyAllWindows() # close any open windows

```

14. ให้เพิ่มโปรแกรมเพื่อให้กำหนดคุณภาพและตำแหน่งของเส้นบนสัญญาณภาพ โดยให้ใส่ต่อท้ายคำสั่ง `cv2.VideoCapture` ด้วยคำสั่งดังนี้

```

thick = 20
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) # get width of video resolution
sensorLeft = int(width * 20/100)
sensorRight = int(width * 75/100)

```

15. ให้เพิ่มโปรแกรมเพื่อนำค่าที่กำหนดไว้ในข้อ 14 มาทำการคำนวณแล้วตัดภาพตรงพื้นที่ตำแหน่งเส้นออกมา โดยให้ใส่ต่อท้ายคำสั่ง `gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)` ที่เปลี่ยนสัญญาณภาพวิดีโอที่ได้เป็นภาพระดับลิเทา ด้วยคำสั่งดังนี้

```

leftFrame = gray[:, sensorLeft : sensorLeft + thick] # y1:y2 , x1:x2
rightFrame = gray[:, sensorRight : sensorRight + thick]

```

16. ถ้าต้องการดูรายละเอียดข้อมูลภาพบนเส้นที่ตัดออกมานี้ สามารถทำได้โดยเพิ่มโปรแกรมต่อท้ายข้อ 15 แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```

cv2.imshow('Left Frame',leftFrame)
cv2.imshow('Right Frame',rightFrame)

```

17. ให้เพิ่มโปรแกรมต่อท้ายข้อ 16 เพื่อให้ตั้งค่า `firstFrame` ที่เป็นเฟรมที่หนึ่ง ซึ่งจะใช้ในการเปรียบเทียบเพื่อหาความแตกต่างกับสัญญาณภาพเฟรมอื่นๆ ด้วยคำสั่งดังนี้

```

if iframe == 1: # if first frame , initialize it
    firstFrame = gray
    firstLeftFrame = leftFrame
    firstRightFrame = rightFrame
# update firstFrame

```

18. ให้เพิ่มโปรแกรมต่อท้ายข้อ 17 เพื่อให้คำนวณหาความแตกต่างของสัญญาณวิดีโอ `firstFrame` กับเฟรมอื่นๆ ซึ่งเป็นการเอาภาพ 2 ภาพมาเทียบหาส่วนต่างโดยการลบกัน แล้วใช้ฟังก์ชันนำผลบวกของทุกสมาชิกในเมตริกซ์มารวมกัน ด้วยคำสั่ง

```

if iframe > 1: # difference between the current frame and first frame
    diffLeft = np.sum(cv2.absdiff(leftFrame , firstLeftFrame)) # sum of all elements
    diffRight = np.sum(cv2.absdiff(rightFrame, firstRightFrame))

```

19. จากข้อ 18 ให้ทดลองโปรแกรมเพื่อคุ้มครองความแตกต่างของสัญญาณวิดีโอ firstFrame กับเฟรมอื่นๆ โดยให้ต่อห้ายด้วยคำสั่ง print(diffLeft, diffRight) และหาว่าตกลู่ที่มีสีและขนาดต่างๆ ไปตัดที่ตำแหน่งเด็น และให้อธิบายผลที่ได้จากการทดลองว่า การแสดงผลลัพธ์ที่ได้ออกมา มีความแตกต่างกันอย่างไร .....
- .....  
.....  
.....  
.....

20. ให้เพิ่มโปรแกรมต่อห้ายข้อ 18 ที่จะเอาผลลัพธ์ที่เป็นค่าความแตกต่างของสัญญาณวิดีโอ firstFrame กับเฟรมที่ 2 ถึง 10 และวนนำข้อมูลนั้นมาเพิ่มข้อมูลต่อไปทั้งห้ายข้อมูลเดิมรวมกันอยู่ใน List ด้วยคำสั่งดังนี้

```
if iframe > 1 and iframe <= 10:  
    diffLeftList.append(diffLeft)  
    diffRightList.append(diffRight)
```

21. ให้เพิ่มโปรแกรมต่อห้ายข้อ 20 โดยในช่วงเฟรมที่ 10 จะนำเอาข้อมูลใน List ทั้งหมด 9 ค่า มาหาค่าเฉลี่ยของผลรวมทางคณิตศาสตร์ในความแตกต่างของสัญญาณวิดีโอเฟรมที่ 1 เทียบกับเฟรมที่ 2 ถึง 10 ด้วยคำสั่งดังนี้

```
if iframe == 10:  
    meanDiffLeft = np.mean(diffLeftList)  
    meanDiffRight = np.mean(diffRightList)
```

22. ให้เพิ่มโปรแกรมต่อห้ายข้อ 21 โดยตั้งแต่ช่วงเฟรมที่ 11 เป็นต้นไปจะให้คำนวนหาค่า เพื่อพิจารณาว่าจะให้ Sensor ทำงานหรือไม่ และถ้าเกิดการทำงานให้ตัวแปร count นับจำนวนคนที่เข้าออกด้วย โดยการตั้งค่าขึ้นอยู่กับการกำหนดค่าใน tolerance ด้วยคำสั่งดังนี้

```
if iframe > 10:  
    Left = diffLeft > (1+tolerance) * meanDiffLeft  
    Right = diffRight > (1+tolerance) * meanDiffRight  
  
    if Right and prevLeft and not Left:  
        countLeft += 1  
        prevLeft = False  
    if Left and prevRight and not Right:  
        countRight += 1  
        prevRight = False  
  
    if Right:  
        prevRight = Right  
    if Left:  
        prevLeft = Left
```

23. ให้เพิ่มโปรแกรมต่อห้ายข้อ 22 โดยการนำสัญญาณภาพจากวิดีโอ frame ปัจจุบันที่ได้มารีสอร์ฟรีสีแล้ว เส้นที่ใช้เป็น Sensor ตรวจสอบความเคลื่อนไหวบนตำแหน่งเด็นนั้น โดยที่ปกติจะให้เส้นเป็นสีเหลือง แต่ถ้าพบว่ามีการเคลื่อนไหวบนเส้นจะให้เส้นนั้นเปลี่ยนเป็นสีเทียว ด้วยคำสั่งดังนี้

```

if Left:
    frame[:, sensorLeft : sensorLeft + thick , [0,2]] = 0           # Display Green line
else:
    frame[:, sensorLeft : sensorLeft + thick , 0] = 0                 # Display Yellow line
if Right:
    frame[:, sensorRight : sensorRight + thick , [0,2]] = 0
else:
    frame[:, sensorRight : sensorRight + thick , 0] = 0

```

24. จากโปรแกรมในข้อ 23 ให้ทดลองการปรับเปลี่ยนค่า tolerance ให้มีค่ามากหรือน้อยกว่า 2 แล้วให้ทดลองการแสดงผลของสีบนภาพที่ได้ โดยให้อธิบายผลความแตกต่างของค่าที่ปรับกับวัตถุที่มีขนาดและสีต่างๆ ที่นำไปตัดตรงตำแหน่ง เส้น Sensor ว่ามีผลต่อการทำงานอย่างไร .....
- .....
- .....
- .....
- .....
- .....

25. ให้เพิ่มโปรแกรมต่อท้ายข้อ 23 เพื่อให้ทำหน้าที่แสดงผลของการนับคนหรือสินค้าที่เข้าและออกบันหน้าจอภาพแสดงผล ในสัญญาณวิดีโอ frame ปัจจุบัน แล้วให้ทดลองผลลัพธ์ที่ได้ ด้วยคำสั่งดังนี้

```

cv2.putText(frame, 'IN '+ str(countLeft), (26, 30), cv2.FONT_HERSHEY_SIMPLEX, .5, (255, 255, 0), 1)
cv2.putText(frame, 'OUT '+ str(countRight), (20, 50), cv2.FONT_HERSHEY_SIMPLEX, .5, (255, 255, 0), 1)

```

26. ให้เพิ่มโปรแกรมต่อท้ายคำสั่ง iframe += 1 ในข้อ 25 เพื่อปรับเปลี่ยนสัญญาณวิดีโอ iframe ที่ใช้ในการการเบริกบานเทียน สัญญาณภาพ ณ.เวลาปัจจุบันทุกครั้งที่พบรการเคลื่อนไหวบนเส้น Sensor พร้อมกัน ซึ่งอาจจะเกิดจากสภาพแวดล้อมที่เปลี่ยนไป แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่ง

```

if Left and Right and prevLeft and prevRight :
    iframe = 0
    Left = False
    Right = False
    prevLeft = False
    prevRight = False

```

27. ถ้าในสภาพแวดล้อมของสถานที่จะนำไปใช้งาน เช่นเป็นห้องสีเขียวหรือสนามหญ้าสีเขียว จะมีวิธีแก้ไขโปรแกรมตรงส่วนไหน เพื่อเพิ่มคำสั่งเข้าไปให้โปรแกรมสามารถตรวจจับได้ไวยิ่งขึ้น ให้อธิบายพร้อมทั้งยกตัวอย่าง .....
- .....
- .....
- .....
- .....
- .....
- .....
- .....
- .....

28. เราจะสร้างโปรแกรมที่ทำหน้าที่เป็น Motion Detection โดยเริ่มจากเอาโปรแกรมในข้อ 3 และเพิ่มในส่วนของ datetime เพื่อตั้งค่าของวันและเวลาที่จะใช้ในโปรแกรม โดยบรรทัดที่เว้นว่างไว้จะใช้สำหรับเพิ่มโปรแกรมในข้อถัดไป แล้วให้ทดลองแสดงผลที่ได้ด้วยคำสั่งดังนี้

```

import cv2
import datetime
import imutils
cap = cv2.VideoCapture(0)
startTime = datetime.datetime.now()
firstFrame = None
motionCounter = 0
while True:
    timestamp = datetime.datetime.now()
    ret, frame = cap.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow("GaussianBlur Gray Frame", gray)
    cv2.imshow("Color Frame", frame)

    if cv2.waitKey(1) == 32:
        break
    cap.release()
    cv2.destroyAllWindows()

```

# initialize the camera  
# start time  
# initialize the first frame in the video stream  
# loop over the frames of the video  
# update the timestamp  
# reading from webcam  
# Converting color image to gray scale image  
# show the frame  
# spacebar key is pressed, break from the loop  
# cleanup the camera  
# close any open windows

29. ให้เพิ่มโปรแกรมเพื่อให้นำสัญญาณวิดีโอ gray ที่เป็นระดับสีเทาจากข้อ 28 มาผ่านการใช้ฟิลเตอร์ Gaussian Blur เพื่อให้ภาพเบลอเป็นการลดสัญญาณรบกวนของภาพ แล้วให้ทดลองแสดงผลที่ได้ด้วยคำสั่งดังนี้

```
gray = cv2.GaussianBlur(gray, (21, 21), 0) # Converting gray scale image to Gaussian Blur
```

30. ให้เพิ่มโปรแกรมต่อท้ายข้อ 29 เพื่อให้ตั้งค่า firstFrame ที่จะใช้เป็นเฟรมแรกในการเปรียบเทียบเพื่อหาความแตกต่างกับสัญญาณภาพเฟรมปัจจุบัน ด้วยคำสั่งดังนี้

```
if firstFrame is None:
    firstFrame = gray
    continue
```

# if the first frame is None, initialize it

31. ให้เพิ่มโปรแกรมต่อท้ายข้อ 30 เพื่อให้คำนวณหาความแตกต่างของสัญญาณวิดีโอ firstFrame และ gray แล้วให้ทดลองแสดงผลที่ได้ด้วยคำสั่งดังนี้

```
diff_frame = cv2.absdiff(firstFrame, gray) # difference between the current frame and first frame
cv2.imshow("Difference Frame", diff_frame)
```

32. ให้เพิ่มโปรแกรมต่อจากข้อ 31 โดยให้อยู่ระหว่างกลวงเพื่อคำนวณหากาก่า threshold ที่ตั้งไว้ 30 เป็นขั้นสัญญาณวิดีโอ diff\_frame ที่ได้เป็นภาพขาวและดำ ซึ่งจะมีค่า 0 กับ 255 โดยส่วนของภาพที่มีการเคลื่อนไหวคือมีค่ามากกว่า 30 จะเป็นสีขาว ด้วยคำสั่งดังนี้

```
thresh_frame = cv2.threshold(diff_frame, 30, 255, cv2.THRESH_BINARY)[1] # greater than 30 show white color
cv2.imshow("Threshold Frame", thresh_frame)
```

33. จากข้อ 32 ให้ทดลองโปรแกรม โดยการปรับค่า threshold = 30 ให้มีค่ามากขึ้นหรือน้อยลง แล้วให้อธิบายผลจากการทดลองนี้ว่าการแสดงผลลัพธ์ของภาพ thresh\_frame ที่ได้มีความแตกต่างกันอย่างไร .....

34. ให้เพิ่มโปรแกรมต่อจากข้อ 32 โดยให้ยูร่าห่วงกลาง ที่จะนำสัญญาณวิดีโอ thresh\_frame ที่ได้จากข้อ 32 นำมาคำนวณ ต่อด้วยฟังก์ชัน dilate โดยกำหนดให้ iteration=2 เพื่อบำยความต่อเนื่องของภาพตรงตำแหน่งภาพที่ตรวจจับได้ว่ามีการเคลื่อนไหว ด้วยคำสั่งดังนี้

```
thresh_frame = cv2.dilate(thresh_frame, None, iterations=2) # dilate the thresholded image to fill in holes
```

35. จากข้อ 34 ให้ทดลองโปรแกรม โดยการปรับค่า iterations แล้วทดลองการแสดงผลของภาพ thresh\_frame โดยให้อธิบายว่าก่อนและหลังการใช้ฟังก์ชันในข้อนี้มีความแตกต่างกันอย่างไร .....

36. ให้เพิ่มโปรแกรมต่อท้ายข้อ 34 โดยการนำสัญญาณวิดีโอ thresh\_frame ที่ได้มาคำนวณหา Contours ซึ่งจะได้เป็นเก้าอโกรงตรงตำแหน่งที่ตรวจจับได้ว่ามีการเคลื่อนไหว ROI แล้วให้วัดเส้นล้อมเป็นสีน้ำเงินไว้ ให้ทดลองผลลัพธ์ที่ได้ ด้วยคำสั่งดังนี้

```
(cnts, _) = cv2.findContours(thresh_frame.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(frame, cnts, -1, (255, 0, 0), 2) # draw all contours on the frame
```

37. ให้เพิ่มโปรแกรมต่อท้ายข้อ 36 เพื่อคำนวณขนาด contour ถ้ามีขนาดพื้นที่มากพอตามที่กำหนดให้พล็อตรูปสี่เหลี่ยมสีเทียบคุณลักษณะที่ตำแหน่งนั้นไว้ ด้วยคำสั่งดังนี้

```
for contour in cnts:
    if cv2.contourArea(contour) < 3000: # loop over the contours
        continue # if the contour is too small, ignore it
    (x, y, w, h) = cv2.boundingRect(contour) # compute the bounding box for the contour
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 1) # draw box on the frame
    motionCounter += 1
```

38. จากข้อ 37 ให้ทดลองโปรแกรม โดยการปรับเปลี่ยนค่า contourArea ให้มีค่ามากหรือน้อยกว่า 3000 แล้วทดลองการพล็อตรูปสี่เหลี่ยมที่ได้ว่ามีผลต่อการเปลี่ยนค่าอย่างไร .....

39. ให้เพิ่มโปรแกรมต่อท้ายข้อ 37 เพื่อให้ทำหน้าที่แสดงผลของวัน-เวลาบนหน้าจอแสดงผล ซึ่งเป็นสัญญาณวิดีโอ frame แล้วให้ทดลองผลที่ได้ ด้วยคำสั่งดังนี้

```
ts = timestamp.strftime("%d %b %Y %I:%M:%S%p") # draw timestamp on the frame
cv2.putText(frame, "{}".format(ts), (10, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 1)
```

40. ให้เพิ่มโปรแกรมต่อท้ายคำสั่ง cv2.imshow หลังจากแสดงผลภาพของสัญญาณวิดีโอต่างๆแล้ว จะให้ทำการบันทึกภาพจากสัญญาณวิดีโอ เมื่อมีการตรวจจับว่าเกิดการเคลื่อนไหว โดยให้มีระยะเวลาห่างระหว่างภาพไม่น้อยกว่า 10 วินาที และให้ตั้งไฟล์เป็นชื่อ capture1.png แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```

if (timestamp - startTime).seconds >= 10 :           # if enough time has passed between uploads
    if motionCounter >= 3:                         # number of frames with consistent motion is high enough
        print(format(ts) , ' ', motionCounter , ' ', time = ", (timestamp - startTime).seconds)
        cv2.imwrite('capture1.png', frame)            # write the image to png file

        startTime = timestamp                          # update the last uploaded timestamp
        motionCounter = 0
    
```

41. จากข้อ 40 ให้แก้ไขโปรแกรมที่ใช้ในการบันทึกภาพโดยให้เปลี่ยนการตั้งชื่อไฟล์เป็น วัน-เวลา ที่ทำการบันทึกภาพ แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```

filename = "P"+timestamp.strftime("%Y%m%d-%H%M%S")+".jpg"
cv2.imwrite(filename , frame)                         # write the image to jpg file
    
```

42. ให้เพิ่มโปรแกรมต่อท้ายโปรแกรมข้อ 41 เพื่อให้ทำการบันทึกสัญญาณวิดีโอเมื่อมีการตรวจจับว่าเกิดการเคลื่อนไหว โดยทำการตั้งชื่อไฟล์เป็น วัน-เวลา ที่ทำการบันทึกสัญญาณวิดีโอนั้น ด้วยคำสั่งดังนี้

```

height, width, channels = frame.shape
fourcc = cv2.VideoWriter_fourcc(*'mp4v')           # fourcc = cv2.VideoWriter_fourcc('m', 'p', '4', 'V')
filenameV = "V"+timestamp.strftime("%Y%m%d-%H%M%S")+".mp4"
video = cv2.VideoWriter(filenameV, fourcc, 20, (width, height))
    
```

43. ให้เพิ่มโปรแกรมต่อท้ายโปรแกรมข้อ 42 เพื่อกำหนดให้ทำการบันทึกสัญญาณวิดีโอเป็นเวลา 10 วินาที โดยระยะเวลาที่ทำการบันทึกสัญญาณวิดีโอหาได้จากค่า Frame rate เท่ากับ 20 fps ที่กำหนดไว้ในข้อ 20 คูณกับ 10 Sec ได้เท่ากับ 200 ที่กำหนดในคำสั่ง range(200) ด้วยคำสั่งดังนี้

```

for i in range(200):
    ret, frame = cap.read()                         # Record 10 Sec
    video.write(frame)                            # reading from webcam
    video.release()                             # write the video to mp4 file
    
```

44. จากโปรแกรมในข้อ 43 ให้ทดลองการทำงานของโปรแกรม แล้วให้อธิบายว่าขณะทำการบันทึกสัญญาณวิดีโอโปรแกรมในส่วนของ Motion Detection จะหยุดการทำงานไปด้วยหรือไม่ เพราะเหตุใด แล้วจะมีแนวทางในการแก้ไขโปรแกรมในส่วนไหนเพื่อให้โปรแกรมทำงานได้ตลอดเวลา .....
- .....
- .....
- .....
- .....
- .....
- .....

45. ให้เพิ่มโปรแกรมเพื่อให้สามารถปรับขนาดของความกว้างของสัญญาณวิดีโอได้ แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่ง

```

frame = imutils.resize(frame, width=640)             # resize the frame
    
```

46. ให้เพิ่มโปรแกรมเพื่อปรับเปลี่ยนสัญญาณวีดีโอ firstFrame ที่จะใช้ในการการเปรียบเทียบเป็นสัญญาณภาพ ณ.เวลาปัจจุบัน ทุกครั้งที่พบร่องรอยเคลื่อนไหว แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่ง

```
firstFrame = gray # first frame is current frame
```

47. ให้เพิ่มโปรแกรมเพื่อทำการแจ้งเตือนไปที่ LINE Notify ทุกครั้งที่ตรวจสอบแล้วพบว่ามีการเคลื่อนไหว โดยให้ทำการสมัครใช้งานที่ <https://notify-bot.line.me/th/> กดเข้าสู่ระบบแล้วใส่ email และ password ของ Line ที่ใช้งานอยู่ กดที่ชื่อตระมูลบนหน้าของฉัน กดออก Token และใส่ชื่อ Token (จะแสดงเมื่อมีการแจ้งเตือน) แล้วกดชื่อตัวเอง ถ้าต้องการแจ้งเตือนแบบตัวต่อตัวจาก LINE Notify หรือกดคุณที่ต้องการแจ้งเตือน เมื่อเลือกเสร็จแล้วให้กด ออก Token และไว้ให้ทำการคัดลอก Token ที่ได้เพื่อจะนำมาใส่ในโปรแกรมที่จะใช้งาน
48. ให้เพิ่มโปรแกรมที่ส่วนหัวของโปรแกรมเพื่อกำหนดค่าที่จะใช้ในการส่ง Line Notify โดยในส่วนของ 'XXXXXX' ให้ใส่ เป็นรหัส Token ที่ได้จากข้อ 47 ด้วยคำสั่งดังนี้

```
import requests
LINE_ACCESS_TOKEN = 'XXXXXX'
url = 'https://notify-api.line.me/api/notify'
LINE_HEADERS = {'Authorization': 'Bearer ' + LINE_ACCESS_TOKEN}
session = requests.Session()
```

49. ให้เพิ่มโปรแกรมต่อท้ายโปรแกรมข้อ 41 เพื่อทำการแจ้งเตือนไปที่ LINE Notify ทุกครั้งที่พบร่องรอยเคลื่อนไหว โดยจะส่ง เป็นภาพถ่ายขณะนั้นที่มีการระบุวัน-เวลาในขณะเกิดเหตุการณ์บนภาพถ่ายนั้น และให้พลีอตรูปสีเหลี่ยมคลุมล้อม ตำแหน่งที่ตรวจพบได้ว่ามีการเคลื่อนไหว โดยเมื่อระบบได้ทำการส่ง LINE Notify แล้วจะมีการพิมพ์ข้อความแสดง ออกมาว่าส่งได้สำเร็จหรือไม่ ให้ทดลองการทำงานที่ได้ ด้วยคำสั่งดังนี้

```
_ , line = cv2.imencode('.jpg', frame) # write the image to LINE Notify
response = session.post(url, headers=LINE_HEADERS, params={'message': 'Alert'}, files={'imageFile': line.tostring()})
print(response.text)
```

50. ให้เพิ่มการแจ้งเตือน LINE Notify เป็นแบบกลุ่มไปที่ LINE กลุ่มของวิชานี้ แล้วแก้ไขโปรแกรมใส่รหัส Token เพื่อทำการแจ้งเตือน LINE Notify ไปที่ LINE กลุ่มของวิชานี้ เมื่อโปรแกรม Motion Detection มีการตรวจจับพบร่องรอยเคลื่อนไหว โดยให้ส่งการตรวจจับมาเป็นภาพถ่ายพร้อมรายละเอียดที่ตรวจพบได้รวมทั้งให้ระบุชื่อและรหัสนักศึกษาใน LINE Notify ที่ส่งด้วยอย่างน้อยคนละ 2 ภาพ

51. การทดลองจะนำโปรแกรมใน OpenCV และไฟล์ Cascade ในการตรวจจับวัตถุของ Haar Cascades ซึ่งมีการตรวจจับวัตถุได้หลายชนิด ซึ่งในที่นี้เราจะเริ่มด้านด้วยการตรวจจับใบหน้า (Face detection) และเพื่อให้เกิดความเที่ยงตรงมากขึ้น จะนำรูปใบหน้านี้มาตรวจจับดวงตา (Eye detection) อีกครั้ง เพื่อให้แน่ใจว่าเป็นใบหน้าจริง และจะทำการบันทึกใบหน้านี้ให้เป็นไฟล์หลายไฟล์ ที่จะสามารถนำไปใช้ในการเรียนรู้เชิงลึกเพื่อการจัดจำวัตถุ (Deep Learning for Object Recognition) เริ่มจากให้ทำการดาวน์โหลดไฟล์ซอฟต์แวร์ HaarCascades ที่ผ่านการฝึกฝนเรียนรู้ไว้ล่วงหน้าแล้วใส่ไว้ในไดเรกทอรี่ที่ใช้งาน Python โดยให้ไปที่ <https://github.com/opencv/opencv/tree/master/data> และนำเอาโปรแกรมในข้อ 3 และเพิ่มในส่วนของการตั้งค่า Cascade ที่จะใช้ในโปรแกรม ด้วยคำสั่งดังนี้

```

import cv2
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')           # Detect object in video stream
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
smile_cascade = cv2.CascadeClassifier('haarcascade_smile.xml')
SCALE_FACTOR = 1.3
MIN_NEIGHBORS = 5
count = 0                                         # Initialize face image counter
cap = cv2.VideoCapture(0)                         # Start capturing video
while True:
    frame = cap.read()                           # Capture video frame
    height, width, channels = frame.shape
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Convert frame to grayscale
    cv2.imshow('Frame', frame)                   # Display the video frame
    if cv2.waitKey(1) == 32:                      # if the spacebar key is pressed, break from the loop
        break
cap.release()                                     # cleanup the camera
cv2.destroyAllWindows()                           # close any open windows

```

52. ให้เพิ่มโปรแกรมเพื่อให้นำสัญญาณวิดีโอ gray ที่เป็นระดับสีเทาจากข้อ 51 มาที่ละเพรอมผ่านฟังก์ชัน detectMultiScale โดยใช้ haarcascade\_frontalface\_default.xml เพื่อตรวจจับและพันหาใบหน้า การปรับค่าสเกลในการเบริ่งที่ยืดที่สากล ละเอียดก็จะใช้เวลานานและอาจจะทำให้มีความสามารถระบุวัตถุที่มีขนาดใหญ่ได้ เมื่อพบใบหน้าแล้วจะทำการวนรอบเฟรมภาพเพื่อทำการสร้างกรอบสี่เหลี่ยมสีม่วงบนใบหน้านั้น จากนั้นจะแยก ROI ของใบหน้าจากกรอบที่ได้สร้างไว้เป็นรูปใหม่ 2 รูป คือเป็นรูปสีชื่อ face และรูประดับสีเทาชื่อ faceGray และโปรแกรมทำการแสดงรูปสี face บนจอแสดงผล ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```

faces = face_cascade.detectMultiScale(gray, SCALE_FACTOR, MIN_NEIGHBORS)      # Detect frames of different sizes
for (x,y,w,h) in faces:                                                       # Loop for each faces
    cv2.rectangle(frame,(x,y),(x+w,y+h),(128,0,128),1)                      # Crop the image frame into rectangle
    face = frame[y:y+h, x:x+w]                                              # Convert Crop the image to face
    faceGray = gray[y:y+h, x:x+w]                                             # Display the Face frame
    cv2.imshow('Face', face)

```

53. ให้เพิ่มโปรแกรมต่อท้ายข้อ 52 เพื่อนำรูประดับสีเทาที่เป็นรูปใบหน้าชื่อ faceGray มาผ่านฟังก์ชันเพื่อตรวจจับและคืนหาดวงตาบนใบหน้านั้น เมื่อพบดวงตาจะทำการวนรอบเฟรมภาพเพื่อทำการสร้างกรอบสี่เหลี่ยมสีน้ำเงินบนดวงตาที่พบ ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```

eyes = eye_cascade.detectMultiScale(faceGray)                                # Detect frames of different sizes
for (ex,ey,ew,eh) in eyes:                                                    # Loop for each eyes
    cv2.rectangle(face,(ex,ey),(ex+ew,ey+eh),(255,0,0),1)                  # Crop the image frame into rectangle

```

54. จากโปรแกรมข้อ 53 เมื่อทำการตรวจจับใบหน้าและสามารถตรวจจับดวงตาอยู่บนใบหน้านั้นด้วย ซึ่งจะเป็นการแสดงว่าเป็นใบหน้าคนจริง แล้วก็จะให้ทำการบันทึกภาพของใบหน้านั้นจากสัญญาณสีของวีดีโอที่ชื่อ face มาเป็นภาพถ่ายชนิด jpg ไว้ในโฟลเดอร์ชื่อ images ซึ่งจะต้องสร้างโฟลเดอร์ไว้ก่อน เพื่อจะเก็บไฟล์ไว้ใช้งานต่อไป โดยจะตั้งชื่อไฟล์ในแต่ละภาพขึ้นต้นด้วยอักษร E และตามด้วยหมายเลขเรียงตามลำดับก่อนหลังเริ่มจากหมายเลขหนึ่ง ให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
count += 1 # Increment sample face image
filename = str(count)+".jpg"
cv2.imwrite("images/E"+filename , face) # Save the eye detect to jpg file
```

55. จากโปรแกรมการบันทึกภาพการตรวจจับใบหน้าและดวงตาในข้อ 54 ถ้าในระหว่างบันทึกต้องการที่จะรู้ว่าได้บันทึกภาพถ่ายไปทั้งหมดกี่ภาพจะต้องเพิ่มโปรแกรมต่อท้ายข้อ 54 เพื่อให้แสดงผลที่หน้าจอ และจะมีเพิ่มการแสดงขนาดของภาพจากกล้อง USB Webcam ขึ้นสัญญาณวิดีโอดังนี้ แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
cv2.putText(frame, str(count), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 2)
cv2.putText(frame, str(width)+x+str(height), (10, height-20), cv2.FONT_HERSHEY_SIMPLEX, .6, (0, 255, 255), 1)
```

56. จากโปรแกรมการบันทึกภาพในข้อ 55 ถ้าต้องการให้โปรแกรมบันทึกภาพของแต่ละคนเป็นตัวอย่างที่จะใช้ในการเรียนรู้ของเครื่อง (Machine Learning) !เพื่อจดจำใบหน้าของแต่ละคน เช่นต้องการบันทึกทั้งหมด 100 ภาพ สามารถทำได้โดยเอาค่า count มาใช้ในการตั้งค่าอุอกจากลูป โดยให้เขียนโปรแกรมต่อท้ายคำสั่ง break ตรงบรรทัดว่างของโปรแกรมเดิม แล้วให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
elif count>=100: # If image taken reach 100, stop taking video
    break
```

57. จากภาพถ่ายในการบันทึกใบหน้าของแต่ละคนจะเห็นได้ว่าภาพที่ได้มีเฉพาะใบหน้าไม่ได้รวมส่วนหัวของคนทั้งหมด ให้นักศึกษาทำการแก้ไขโปรแกรม โดยให้ขยายน้ำดของภาพถ่ายออกไปด้านข้างทุกด้าน ให้แต่ละด้านขยายออกไป 25 % แต่ถ้ากรณีที่ขยายแล้วไปชนขอบภาพของ frame ให้สั่นสุดแค่ขอบภาพนั้น

58. กรณีที่การใช้งานบางอย่างที่ต้องการให้เกิดความถูกต้องมากยิ่งขึ้น ไปอีก อาจจะต้องเพิ่มการตรวจจับอย่างที่ 3 คือการตรวจจับปากหรือรอยยิ้ม (Smile detection) จากโปรแกรมในข้อ 53 ให้นำมาแก้ไขเพิ่มเติมแล้วใส่ต่อท้ายโปรแกรมเดิม โดยใช้รูประดับสีเทาที่เป็นรูปใบหน้าชื่อ faceGray มาผ่านฟังก์ชันเพื่อตรวจจับรอยยิ้มหรือปากจากใบหน้านั้น เมื่อพบปากจะทำการวนรอบเฟรมภาพเพื่อทำการสร้างกรอบสีเหลืองสีเขียวบนปากที่พบ ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
smile = smile_cascade.detectMultiScale(faceGray) # Detect frames of different sizes
for (sx,sy,sw,sh) in smile: # Loop for each smile
    cv2.rectangle(face,(sx,sy),(sx+sw,sy+sh),(0,128,0),1) # Crop the image frame into rectangle
```

59. ให้เพิ่มโปรแกรมต่อจากข้อ 58 เพื่อทำการแจ้งเตือน LINE Notify ไปที่ LINE กลุ่มของวิชานี้ เมื่อมีการตรวจจับใบหน้า ดวงตา และปากของนักศึกษา โดยให้สั่งเป็นภาพถ่ายของนักศึกษาที่เห็นส่วนหัวทั้งหมดและมีการระบุส่วนที่ตรวจพบซึ่งจะผลลัพธ์เป็นรูปสีเหลืองกลุ่มล้อมตำแหน่งที่ตรวจพบนั้นอย่างน้อยคนละ 2 ภาพ

60. จากไฟล์ที่บันทึกไว้ในข้อ 54 เมื่อทำการตรวจจับใบหน้านั้นในบางกรณีไม่อาจจะนำไปใช้งานได้ เพราะมีการสร้างกรอบสีเหลืองอยู่บนใบหน้า ให้แก้ไขโปรแกรมเพื่อบันทึกไฟล์ของใบหน้าที่ยังไม่มีการสร้างกรอบสีเหลือง แล้วให้บันทึกไฟล์คู่กัน โดยจะตั้งชื่อไฟล์ในแต่ละภาพขึ้นต้นด้วยอักษร F และตามด้วยหมายเลขเรียงตามลำดับก่อนหลัง

61. การทดลองสร้างโปรแกรมที่ทำหน้าที่ Tracking เพื่อติดตามการเคลื่อนที่ของวัตถุ (Object) จากกล้อง Webcam โดยจะติดตามวัตถุจากสีที่กำหนดบนภาพของวิดีโอ เริ่มจากเอาโปรแกรมในข้อ 2 และเพิ่มในส่วนของการกำหนดพื้นที่สีเหลี่ยม เส้นสีดำที่อยู่ตรงกลางหน้าจอภาพเพื่อที่จะตัดรูปออกมา โดยบรรทัดที่เว้นว่างไว้จะใช้สำหรับเพิ่มโปรแกรมในข้อด้านไปแล้วให้ทดลองการแสดงผลที่ได้ด้วยคำสั่งดังนี้

```

import cv2
import numpy
import imutils
cap = cv2.VideoCapture(0)
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
w = int(width/2)
h = int(height/2)
while True:
    ret,frame = cap.read()                                # reading from webcam
    cv2.rectangle(frame, (w-2, h-2), (w+22, h+22), (0,0,0), 2)      # draw box on the frame
    cv2.imshow("Frame", frame)
    if cv2.waitKey(1)==32:
        break

cap.release()
cv2.destroyAllWindows()

```

62. ถ้าในภาพที่แสดงผลต้องการลับค้าน้ำหน้า หรือบัน-ล่าง เพื่อให้ติดตามวัตถุได้ตรงตามทิศทาง สามารถทำได้โดยเพิ่มโปรแกรมหลังคำสั่ง cap.read() ในข้อ 61 แล้วให้ทดลองแสดงผลที่ได้ด้วยคำสั่งดังนี้

```
frame = cv2.flip(frame,flipCode=0)                      # 0=H-L , 1=L-R
```

63. จะทำการตัดรูปจากพื้นที่สีเหลี่ยมที่กำหนด และตรวจหาค่าเฉลี่ยของสี BGR ภายในเฟรมภาพที่ตัด โดยจะต้องนำวัตถุตามสีที่กำหนดไปไว้ในช่องสีเหลี่ยม เมื่อกด spacebar จะออกจากกลุ่ม แล้วนำมารำนาวน้ำหน้าค่าเฉลี่ยของสีออกมานะ ให้เพิ่มโปรแกรมต่อท้ายหลังคำสั่ง break ในข้อ 62 ด้วยโปรแกรมดังนี้

```

cut = frame[h:h+20, w:w+20]
B,G,R = cv2.split(cut)
B = int(numpy.mean(B))                                # Compute the arithmetic mean (average) of frame B
G = int(numpy.mean(G))                                # Compute the arithmetic mean (average) of frame G
R = int(numpy.mean(R))                                # Compute the arithmetic mean (average) of frame R

```

64. ให้เพิ่มโปรแกรมต่อท้ายข้อ 63 เพื่อนำเฟรมรูปสีเหลี่ยมที่ตัดออกมามาไปหาค่าสี HSV แล้วแยกค่าสีแต่ละค่าออกมานำไปหาค่าต่ำสุดและสูงสุด เพื่อจะนำไปใช้ในการกำหนดสีของวัตถุที่จะใช้ในการติดตาม ด้วยคำสั่งดังนี้

```

cut = cv2.cvtColor(cut, cv2.COLOR_BGR2HSV)
hue,saturation,value = cv2.split(cut)
Hmin = int(numpy.min(hue))                            # Compute the arithmetic min of hue (0-179)
Hmax = int(numpy.max(hue))                            # Compute the arithmetic max of hue
Smin = int(numpy.min(saturation))                    # Compute the arithmetic min of saturation (0-255)
Smax = int(numpy.max(saturation))                    # Compute the arithmetic max of saturation
Vmin = int(numpy.min(value))                         # Compute the arithmetic min of value (0-255)
Vmax = int(numpy.max(value))                         # Compute the arithmetic max of value
Lower = (Hmin-15, 80, 20)                            # HSV color code lower and upper bounds
Upper = (Hmax+15, 255, 255)

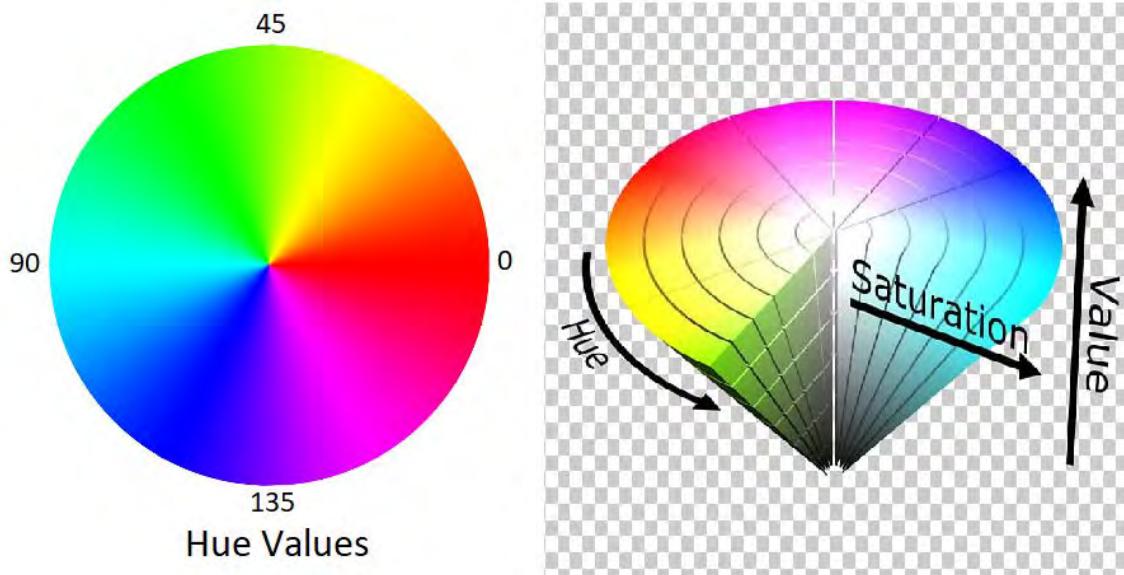
```

โดยที่ระบบสี HSV ประกอบด้วย 3 ค่า ดังนี้

H (hue) ค่าเฉลี่ดสี มีค่าตั้งแต่ 0-179 เป็นค่าที่บ่งบอกว่าสีอยู่ในช่วงไหน เช่น สีแดงเป็น 0 สีเขียวเป็น 60 สีน้ำเงินเป็น 120

S (saturation) ค่าความอิมตัว หรือความสดของสี มีค่าตั้งแต่ 0-255 ถ้าค่ามากสีจะสด

V (value) ค่าความสว่าง มีค่าตั้งแต่ 0-255 ถ้าค่ามากจะสว่าง



65. ให้เพิ่มโปรแกรมต่อท้ายโปรแกรมข้อ 64 เพื่อสร้างลูปที่ใช้ในการติดตามคันหาดแสดงผล โดยในตอนเริ่มต้นนี้จะนำเอารูปภาพสัญญาณวิดีโอจากกล้องและรายละเอียดของสีต่างที่คำนวณได้ก่อนหน้านี้มาแสดงผลบนหน้าจอ ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
while True:
    ret, frame = cap.read()
    frame2 = frame.copy()

    cv2.putText(frame,'B='+str(B)+' G='+str(G)+' R='+str(R), (20,30),cv2.FONT_HERSHEY_SIMPLEX, .5,(B,G,R),2)
    cv2.putText(frame,'H='+str(Hmin)+' S='+str(Smin)+' V='+str(Vmin),(20,50), cv2.FONT_HERSHEY_SIMPLEX,.5,(B,G,R),2)
    cv2.putText(frame,'H='+str(Hmax)+' S='+str(Smax)+' V='+str(Vmax),(20,70), cv2.FONT_HERSHEY_SIMPLEX,.5,(B,G,R),2)
    cv2.putText(frame,str(width) +'x'+str(height),(20,90),cv2.FONT_HERSHEY_SIMPLEX,.5,(B,G,R),2)
    cv2.imshow("Frame", frame)

    if cv2.waitKey(1)==32:
        break
```

66. ให้เพิ่มโปรแกรมเพื่อให้นำสัญญาณวิดีโอ frame ที่เป็นสี BGR จากข้อ 65 หลังคำสั่ง frame2 = frame.copy() มาผ่านการใช้ฟิลเตอร์ Gaussian Blur เพื่อให้ภาพเบลอเป็นการลดสัญญาณรบกวนของภาพ แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
blur = cv2.GaussianBlur(frame, (11, 11), 0)
cv2.imshow('Frame blur',blur)
```

67. ให้เพิ่มโปรแกรมต่อท้ายข้อ 66 เพื่อให้นำสัญญาณวิดีโอ blur ที่ได้เป็นสี BGR เป็นสีที่กำหนดค่า H ทำให้คำนวณ HSV ซึ่งจะทำให้ตัวแปรที่เป็นแมสี 3 สี เปลือดตัวเปรื่องสีเพียงตัวเดียวคือ H ทำให้ง่ายต่อการค้นหา แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)
cv2.imshow('Frame hsv',hsv)
```

68. ให้เพิ่มโปรแกรมต่อท้ายข้อ 67 เพื่อให้คำนวณหาความแตกต่างของสัญญาณวิดีโอ hsv จากค่าช่วงสีที่กำหนดไว้ระหว่าง lower และ upper เพื่อเปลี่ยนเป็น mask ซึ่งจะเป็นภาพขาวและดำ ถ้าจุดภาพ (Pixel) ใดอยู่ในช่วงที่กำหนดจะได้เป็นสีขาว นอกช่วงที่กำหนดจะเป็นสีดำ แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
mask = cv2.inRange(hsv, Lower, Upper)
cv2.imshow('Frame mask1',mask)
```

69. ให้เพิ่มโปรแกรมต่อท้ายข้อ 68 ที่จะนำสัญญาณวิดีโอ mask ที่ได้จากข้อ 68 นำมาผ่านตัวกรอง (filter) ด้วยฟังก์ชันการกัดกร่อน (erode) โดยกำหนดให้ iteration=2 ทำการกรองໄล่ก่าวดบันภาพเพื่อลบบางส่วนที่เป็นเศษทึ่งออกไป แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
mask = cv2.erode(mask, None, iterations=2)
cv2.imshow('Frame mask2',mask)
```

70. ให้เพิ่มโปรแกรมต่อท้ายข้อ 69 ที่จะนำสัญญาณวิดีโอ mask ที่ได้จากข้อ 69 มาผ่านตัวกรอง (filter) ด้วยฟังก์ชันการปองตัว (dilate) โดยกำหนดให้ iteration=2 เพื่อขยายความต่อเนื่องของภาพตรงตำแหน่งการติดตามวัตถุจากสีที่กำหนด แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
mask = cv2.dilate(mask, None, iterations=2)
cv2.imshow('Frame mask3',mask)
```

71. ให้เพิ่มโปรแกรมต่อท้ายข้อ 70 โดยการนำสัญญาณวิดีโอ mask ที่ได้มาคำนวณคืนหา Contours ซึ่งจะได้เป็นเก้าโครงสร้างตำแหน่งการติดตามวัตถุจากสีที่กำหนด ด้วยคำสั่งดังนี้

```
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
```

72. ให้เพิ่มโปรแกรมต่อท้ายข้อ 71 เพื่อตรวจสอบขนาดของ contour ถ้ามีขนาดพื้นที่มากพอตามที่กำหนดให้พล็อตรูปวงกลมคลุมล้อมตามสีของวัตถุที่คืนหานั้น และให้มีจุดศูนย์กลางอยู่ตรงกลาง ซึ่งจะสามารถนำไปประยุกต์เพื่อทำาร์ดแวร์ในการหมุนหรืออวิงเข้าหากาฬได้ต่อไป หลังจากนั้นจะทำการบันทึกไฟล์ภาพถ่ายชื่อ Tracking.png แล้วให้ทดลองแสดงผลที่ได้ ด้วยคำสั่งดังนี้

```
center = None
if len(cnts) > 0:
    c = max(cnts, key=cv2.contourArea) # find the largest contour in the mask
    (x, y), radius = cv2.minEnclosingCircle(c) # it to compute the minimum enclosing circle
    M = cv2.moments(c)
    center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
    if radius > 10: # only proceed if the radius meets a minimum size
        cv2.circle(frame, (int(x), int(y)), int(radius), (B,G,R), 5) # draw the circle and centroid on the frame
        cv2.circle(frame, center, 5, (255, 255, 255), -1) # then update the list of tracked points
        cv2.imwrite("Tracking.png", frame)
```

73. จากโปรแกรมในข้อ 67 เป็นต้นไป สามารถนำมายังไฟล์ โดยนำสัญญาณวิดีโอ hsv มาคำนวณหาความแตกต่างของช่วงสีที่กำหนดไว้ใน lower และ upper เพื่อเปลี่ยนเป็นหน้ากาก (mask) แล้วนำเอามาครอบทับสัญญาณภาพ ก็จะได้ส่วนที่จะให้แสดงเฉพาะสีของวัตถุที่จะติดตามการเคลื่อนที่ได้ ให้ทดลองการแสดงผลที่ได้โดยการปรับเปลี่ยนค่า lower และ upper ให้เหมาะสม ด้วยคำสั่งดังนี้

```
mask = cv2.inRange(hsv, numpy.array([Hmin-20,50,0]), numpy.array([Hmax+20,255,255])) # lower, upper
cv2.imshow('Frame2 mask',mask)
mask = cv2.cvtColor(mask,cv2.COLOR_GRAY2BGR) != 0
cv2.imshow("Frame2", frame2*mask)
```

74. การทดลอง License Plate Recognition ก็คือระบบการตรวจจับและอ่านป้ายทะเบียนรถแบบอัตโนมัติ โดยการตรวจจับภาพป้ายทะเบียนของรถที่ได้จากการถ่ายภาพ นำทำการวิเคราะห์และถอดข้อมูลให้อยู่ในรูปของตัวอักษรและตัวเลข เพื่อจะนำข้อมูลที่ได้ไปใช้ประยุกต์ในงานด้านต่างๆ เช่น ใช้กับงานจราจร งานด้านบริหารจราจร ซึ่งในการทดลองการตรวจจับภาพของป้ายทะเบียนรถนี้จะนำเอาวิธีการที่ผ่านมาแล้วในแต่ละแบบมาเปรียบเทียบความแตกต่างของการดำเนินการเพื่อให้สามารถแก้ไขโปรแกรมให้มีความแม่นยำของข้อมูลได้ดีขึ้น เริ่มจากแบบแรกเป็นการใช้การตรวจจับป้ายทะเบียนของ Haar Cascades เป็นเทคนิคการตรวจจับวัตถุป้าหมายที่ใช้คุณสมบัติของภาพให้ตรงกับภาพเชิงบวก (ภาพป้ายทะเบียน) และภาพเชิงลบ (ภาพที่ไม่มีป้ายทะเบียน) ในขั้นตอนมันต้องการตัวอย่างประมาณ 16000 ภาพ เพื่อทำ Machine Learning ที่ใช้ในการฝึกอบรมเพื่อให้ได้อัลกอริทึมที่ใช้ในการเรียนรู้และจดจำวัตถุป้าหมายมาใช้งาน โดยในการทดลองนี้จะใช้ตัวรู้จำป้ายทะเบียนที่ผ่านการฝึกอบรมไว้ล่วงหน้าแล้วชื่อไฟล์ numberplate\_haarcade.xml หรือ haarcascade\_russian\_plate\_number ให้ใส่ไฟล์ไว้ในไดรริกทอรีที่ใช้งาน Python และนำเอาโปรแกรมในข้อ 1 และเพิ่มในส่วนของการตั้งค่า Cascade ที่จะใช้ในโปรแกรม ด้วยคำสั่งดังนี้

```
import cv2

detector = cv2.CascadeClassifier("numberplate_haarcade.xml")           # initialize cascade classifier
frame = cv2.imread('car.jpg')                                              # reading from jpg file
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)                            # Convert frame to grayscale

cv2.imshow('Frame', frame)                                                 # Show the Frame
cv2.waitKey(0)                                                               # wait until any key is pressed
cv2.destroyAllWindows()                                                       # close any open windows
```

75. ให้เพิ่มโปรแกรมเพื่อให้นำสัญญาณภาพ gray ที่เป็นระดับสีเทาจากข้อ 74 มาเป็นข้อมูลป้อนเข้าผ่านฟังก์ชัน detectMultiScale โดยใช้ numberplate\_haarcade.xml เพื่อตรวจจับป้ายทะเบียนรถจากรูป และทำการพิมพ์ตำแหน่งของป้ายทะเบียนตามจำนวนที่ตรวจพบ ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
plates = detector.detectMultiScale(gray, scaleFactor=1.05, minNeighbors=7,          # Detect Number plates
                                     minSize=(30,30), flags=cv2.CASCADE_SCALE_IMAGE)
print(plates)
```

76. ให้เพิ่มโปรแกรมต่อท้ายข้อ 75 โดยจะทำการวนรอบเฟรมภาพเพื่อทำการสร้างกรอบสี่เหลี่ยมสีขาวบนป้ายทะเบียนรถ ตามจำนวนป้ายทะเบียนที่พบ ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
for (x,y,w,h) in plates:                                                 # iterate through each detected number plates
    cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)               # draw bounding box
```

77. ให้เพิ่มโปรแกรมต่อท้ายข้อ 76 โดยให้อยู่ภายใต้คำสั่ง for เพื่อจะทำการกรองตัดแยก ROI ของพื้นที่ป้ายทะเบียนรถจากกรอบที่ได้สร้างไว้เป็นรูปใหม่จำนวน 1 รูป และโปรแกรมจะทำการแสดงรูปของป้ายทะเบียนบนจอแสดงผล ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
plateGray = gray[y:y+h,x:x+w]                                         # Crop the numberplate
cv2.imshow("Plate", plateGray)                                            # Display the Plate
```

78. ให้เพิ่มโปรแกรมต่อจากข้อ 77 เพื่อนำรูประดับสีเทาที่เป็นป้ายทะเบียนรถที่พูนมาผ่านฟังก์ชัน Optical Character Recognition (OCR) ที่ทำหน้าที่แปลงข้อความจากเอกสารที่อยู่ในรูปแบบของภาพถ่ายให้เป็นรูปแบบของ Text ซึ่งในการทดลองนี้จะใช้เป็นไลบรารีของ Tesseract เพื่อที่จะนำมาใช้ตรวจสอบหาคัดแยกข้อความออกจากป้ายทะเบียนรถที่ตรวจพบ ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
import pytesseract

text = pytesseract.image_to_string( plateGray )           # detect text
if len(text) == 0:
    continue
print(text)
```

การติดตั้งแพ็คเกจ Tesseract ที่เป็นเครื่องมือในการทำ OCR จะมีวิธีการติดตั้งที่แตกต่างกันตามระบบปฏิบัติการดังนี้  
วิธีการติดตั้ง Tesseract บนเครื่อง Raspberry pi ให้ใช้คำสั่งดังนี้

```
sudo apt-get update
sudo apt-get install tesseract-ocr
sudo apt-get install libtesseract-dev
```

ติดตั้งแพ็คเกจภาษาไทยคำสั่ง

```
sudo apt-get install tesseract-ocr-tha
```

ติดตั้งแพ็คเกจ Tesseract เพื่อเรียกใช้บน Python คำสั่ง

```
sudo pip install pytesseract
```

เรียกคุณภาษาที่ติดตั้งไว้ใช้งานคำสั่ง

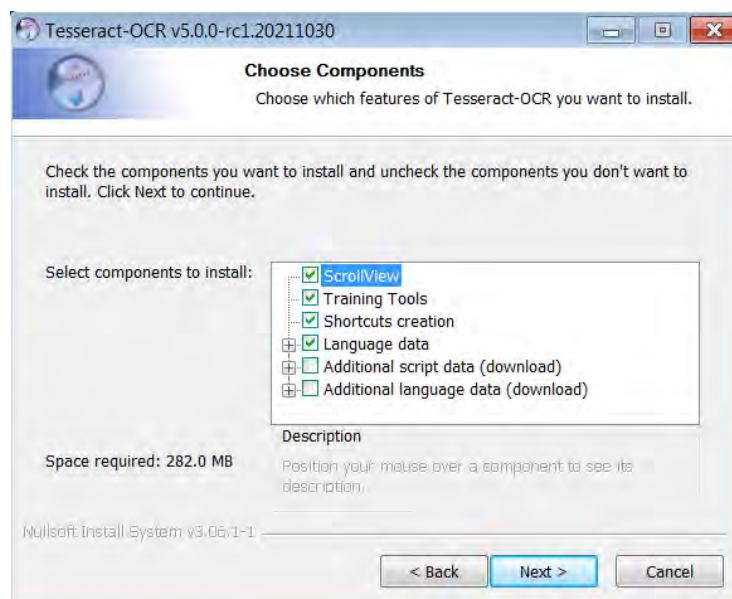
```
tesseract --list-langs
```

```
โดยที่ osd - Script and orientation
eng - English
tha - Thai
```

วิธีการติดตั้ง Tesseract บนระบบปฏิบัติการ MacOS ให้ใช้คำสั่ง

```
brew install tesseract
```

วิธีการติดตั้ง Tesseract บนระบบปฏิบัติการ Microsoft Windows จะต้องทำการติดตั้งโปรแกรม โดยสามารถดาวน์โหลด  
โปรแกรมเพื่อติดตั้งได้จาก <https://github.com/UB-Mannheim/tesseract/wiki> ในการติดตั้งจำเป็นที่จะต้องคลิกเลือกเพิ่ม  
ภาษาไทย



ติดตั้งแพ็คเกจ Tesseract เพื่อเรียกใช้งาน Python จาก Command prompt ด้วยคำสั่ง

```
pip install pytesseract
```

ภายในโปรแกรมจะมีสำหรับการใช้งานบน Windows จะต้องมีการกำหนดค่าสั่งเพิ่มดังนี้

```
pytesseract.pytesseract.tesseract_cmd=r'C:\Program Files\Tesseract-OCR\tesseract.exe' # for windows
```

ภายในคำสั่ง pytesseract.image\_to\_string สามารถกำหนดตัวแปรเพิ่มเติมได้คือ

--psm กำหนดโหมดการแบ่งส่วนของภาพเอกสาร สามารถดูโหมดการทำงานได้ด้วยคำสั่ง

```
tesseract --help-psm
```

Page segmentation modes:

- 0 Orientation and script detection (OSD) only.
- 1 Automatic page segmentation with OSD.
- 2 Automatic page segmentation, but no OSD, or OCR.
- 3 Fully automatic page segmentation, but no OSD. (Default)
- 4 Assume a single column of text of variable sizes.
- 5 Assume a single uniform block of vertically aligned text.
- 6 Assume a single uniform block of text.
- 7 Treat the image as a single text line.
- 8 Treat the image as a single word.
- 9 Treat the image as a single word in a circle.
- 10 Treat the image as a single character.
- 11 Sparse text. Find as much text as possible in no particular order.
- 12 Sparse text with OSD.
- 13 Raw line. Treat the image as a single text line, bypassing hacks that are Tesseract-specific.

```
tesseract --help-oem
```

OCR Engine modes:

- 0 Legacy engine only.
- 1 Neural nets LSTM engine only.
- 2 Legacy + LSTM engines.
- 3 Default, based on what is available.

79. ให้เพิ่มโปรแกรมต่อท้ายข้อ 78 โดยใช้ cv2.putText() เพื่อแสดงข้อความที่ตรวจหาได้ลงบนภาพป้ายทะเบียนรถของ จอแสดงผล ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
cv2.putText(frame,text,(x, y-5),cv2.FONT_HERSHEY_COMPLEX,0.7,(255,128,0),2) # draw text in the frame
```

80. การทดลองเพื่อตรวจจับป้ายทะเบียนรถอีกแบบหนึ่งซึ่งจะมีขั้นตอนที่มากกว่า โดยใช้ฟังก์ชันการเลือกรูปทรงใน OpenCV เพื่อตรวจจับวัตถุที่มีรูปทรงสี่เหลี่ยมแล้วค้นหาป้ายทะเบียน โดยความแม่นยำสามารถปรับปรุงได้หากทราบขนาด ตำแหน่ง สี โดยประมาณของป้ายทะเบียน และตำแหน่งของกล้อง เริ่มจากอาโปรแกรมในข้อที่ 1 และเพิ่มในส่วนของการนำสัญญาณภาพ gray ที่เป็นระดับสีเทามาผ่าน bilateral Filter เพื่อให้ภาพเบลอเป็นการลดสัญญาณรบกวน เพื่อลบรายละเอียดที่ไม่ต้องการออกจากภาพในขณะที่แยกเอาส่วนของออกกัน ซึ่งหากเทียบกับการใช้ Gaussian Blur แล้วผลที่ได้ส่วนที่อยู่ใกล้ขอบจะยังคงรักษาความคมชัดไว้ได้บลอกมาก ในขณะที่ส่วนที่อยู่ไกลจากขอบจะเบลอพอยกัน และให้ทดลองการแสดงผลการทำงานที่ได้ ด้วยคำสั่งดังนี้

```
import cv2
import imutils
import numpy as np

frame = cv2.imread('car.jpg') # reading from jpg file
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #convert to grey scale
gray = cv2.bilateralFilter(gray, 11, 17, 17) #Blur to reduce noise

cv2.imshow('gray',gray)
cv2.imshow('Frame', frame)
cv2.waitKey(0) # Show the Frame
cv2.destroyAllWindows() # wait until any key is pressed
# close any open windows
```

81. ให้เพิ่มโปรแกรมต่อจากข้อ 80 โดยให้อ่านรูปหัวใจกากบาท เพื่อที่จะนำสัญญาณภาพ gray ที่ลดสัญญาณรบกวนของภาพแล้ว มาผ่าน Canny Edge Detection ที่เป็นอัลกอริทึมสำหรับตรวจจับขอบของภาพ แล้วให้ทดลองแสดงผลที่ได้ด้วยคำสั่งดังนี้

```
edged = cv2.Canny(gray, 30, 200) #Perform Edge detection
cv2.imshow('edged',edged)
```

82. ให้เพิ่มโปรแกรมต่อจากข้อ 81 โดยการนำสัญญาณภาพที่ผ่านการหาขอบของภาพ ที่ได้มาคำนวณหา Contours ซึ่งจะมองหารูปทรงบนภาพตามตำแหน่งของขอบภาพ หลังจากนั้นจะตรวจหาขนาดจัดเรียงรูปทรงที่ตรวจพบจากใหญ่ไปเล็ก ด้วยคำสั่งดังนี้

```
contours = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) # find contours
contours = imutils.grab_contours(contours)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:10]
screenCnt = None
```

83. ให้เพิ่มโปรแกรมต่อท้ายข้อ 82 เริ่มจากหาความยาวรวมของเส้นเค้าโครงทั้งหมดด้วยฟังก์ชัน cv2.arcLength() ทำการประมาณเพื่อยุบย่อจุดบนเส้นให้เรียบง่ายขึ้นด้วยฟังก์ชัน approxPolyDP ซึ่งจะได้ขนาด contour ที่มีรูปทรงหลายแบบ โดยแยกกรองเฉพาะรูปทรงของป้ายทะเบียนคือคันหารูปทรงสี่เหลี่ยมผืนผ้าที่มีสี่ด้านและเป็นรูปปิดที่มีขนาดพื้นที่ไม่น้อยกว่าที่กำหนด และทำการเขี่ยเส้นเค้าโครงด้วยฟังก์ชัน convexHull หลังจากนั้นให้ตรวจสอบใหม่ว่าข้างเป็นรูปสี่เหลี่ยม แล้วให้พล็อตรูปสี่เหลี่ยมสีเขียวคลุมล้อมที่ตำแหน่งของป้ายทะเบียนที่ตรวจพบ ให้ทดลองการแสดงผลการทำงานของโปรแกรมที่ได้ด้วยคำสั่งดังนี้

```
for c in contours: # loop over our contours
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * peri, True)
    area = cv2.contourArea(approx)
    if len(approx) == 4 and area >=1000 : # approximated contour has four points
        hull = cv2.convexHull(approx,returnPoints = True) #applying convex hull technique
        if len(hull) ==4: # hull contour has four points
            screenCnt = approx
            break
if screenCnt is None:
    detected = 0
else:
    detected = 1
cv2.drawContours(frame, [screenCnt], -1, (0, 255, 0), 2)
```

84. ให้เพิ่มโปรแกรมต่อท้ายข้อ 83 โดยให้อ่านส่วนของคำสั่ง else เมื่อพบป้ายทะเบียนแล้วให้คำนวณหา mask เพื่อนำมาครอบปิดบังทุกอย่างยกเว้นป้ายทะเบียน แล้วให้ทดลองการแสดงผลการทำงานที่ได้ด้วยคำสั่งดังนี้

```
mask = np.zeros(gray.shape,np.uint8) # Masking the part other than the number plate
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)

new_image = cv2.bitwise_and(frame,frame,mask=mask)

cv2.imshow('new_image',new_image)
cv2.imshow('mask',mask)
```

85. ให้เพิ่มโปรแกรมต่อจากข้อ 84 หลังจากปิดบังภาพทั้งหมดยกเว้นบริเวณป้ายทะเบียนแล้ว จะทำการตัดครอบพื้นที่ป้ายทะเบียนและบันทึกเป็นรูปภาพใหม่ และโปรแกรมจะทำการแสดงรูปของป้ายทะเบียนบนจอแสดงผล แล้วให้ทดลองการแสดงผลการทำงานที่ได้ด้วยคำสั่งดังนี้

```
(x, y) = np.where(mask == 255) # Crop frame
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]
```

```
cv2.imshow('Cropped',Cropped)
```

86. ให้เพิ่มโปรแกรมต่อจากข้อ 85 เพื่อนำรูประดับสีเทาที่เป็นป้ายทะเบียนรถที่พับ มาอ่านอักษรจากภาพนั้น โดยใช้ฟังก์ชัน OCR จากไลบรารี Tesseract และแปลงอักษรที่ได้ไปเก็บอยู่ในตัวแปร text ให้ทดลองการทำงานของโปรแกรม ด้วยคำสั่งดังนี้

```
import pytesseract
pytesseract.pytesseract.tesseract_cmd=r'C:\Program Files\Tesseract-OCR\tesseract.exe' # ---- for windows only ---- **

text = pytesseract.image_to_string(Cropped, lang='eng', config='--psm 7') # Tesseract-OCR Reader object
print("Detected license plate Number is: ",text)

** โปรแกรมในบรรทัดนี้เฉพาะในระบบปฏิบัติการ Microsoft Windows เท่านั้น ถ้าเป็น Raspberry pi ให้ตัดออกไม่ต้องใส่
```

87. ในกรณีที่ป้ายทะเบียนเป็นภาษาไทยให้ใช้เป็นคำสั่ง

```
text = pytesseract.image_to_string(Cropped, lang='tha', config='--psm 7') # Tesseract-OCR Reader object
```

88. จากข้อ 87 ในกรณีที่อ่านป้ายทะเบียนได้ไม่ถูกต้อง อาจจะต้องทำการปรับเปลี่ยนตัวแปรบางตัวของฟังก์ชัน pytesseract.image\_to\_string โดยดูจากข้อ 78 หรืออาจจะต้องปรับปรุงคุณภาพของรูปภาพที่เป็นป้ายทะเบียนรถที่พับ ทั้งนี้ขึ้นอยู่กับสภาพแวดล้อมของรูปถ่าย คุณภาพและรายละเอียดของรูป ด้วยคำสั่ง

```
Cropped = cv2.adaptiveThreshold(Cropped,100, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 15, 16)
```

89. จากข้อความอักษรที่อ่านได้จากป้ายทะเบียนรถและรูปภาพของป้ายทะเบียนรถที่ตรวจจับได้นั้น เราสามารถนำข้อมูลนั้น ไปส่งเป็น email เมื่อตรวจพบป้ายทะเบียนแล้ว โดยจะใช้ไลบรารี smtplib เป็นคำสั่งสำหรับเชื่อมต่อ SMTP เข้ารหัส โดยมี TLS (Transport Layer Security) สำหรับส่งอีเมล์ โดยจะต้องกำหนดที่อยู่ของอีเมล์ที่จะรับและส่ง รหัสผ่านของผู้ส่งอีเมล์ เรื่องดังไว้เป็น “License Plate Recognition” ข้อความที่ส่งอยู่ในตัวแปรชื่อ text1 รูปภาพที่ส่งอยู่ในตัวแปรชื่อ frame และชื่อรูปภาพที่ส่งจะอยู่ในตัวแปรชื่อ text2 เมื่อทำการส่งครั้งแรกจะมีการแจ้งเตือนใน Email โดยจะยังไม่สามารถทำการส่งได้ ให้เข้าไปตั้งค่าความปลอดภัยและการระบุตัวตนก่อนถึงจะทำการส่งได้ แล้วให้ทดลองการทำงานของโปรแกรมที่ได้ ด้วยคำสั่งดังนี้

```
import os
import smtplib
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart
message = MIMEMultipart()
message["From"] = "user1@gmail.com"
password = "*****"
message["To"] = "user2@gmail.com"
message["Subject"] = "License Plate Recognition"

message.attach(MIMEText(text))
image=cv2.imencode(".jpeg", frame)[1].tobytes()
message.attach(MIMEImage(image, name=os.path.basename(text2)))
server=smtplib.SMTP('smtp.gmail.com',port=587)
server.starttls()
server.login(message["From"], password)
server.sendmail(message["From"], message["To"], message.as_string())
server.quit()

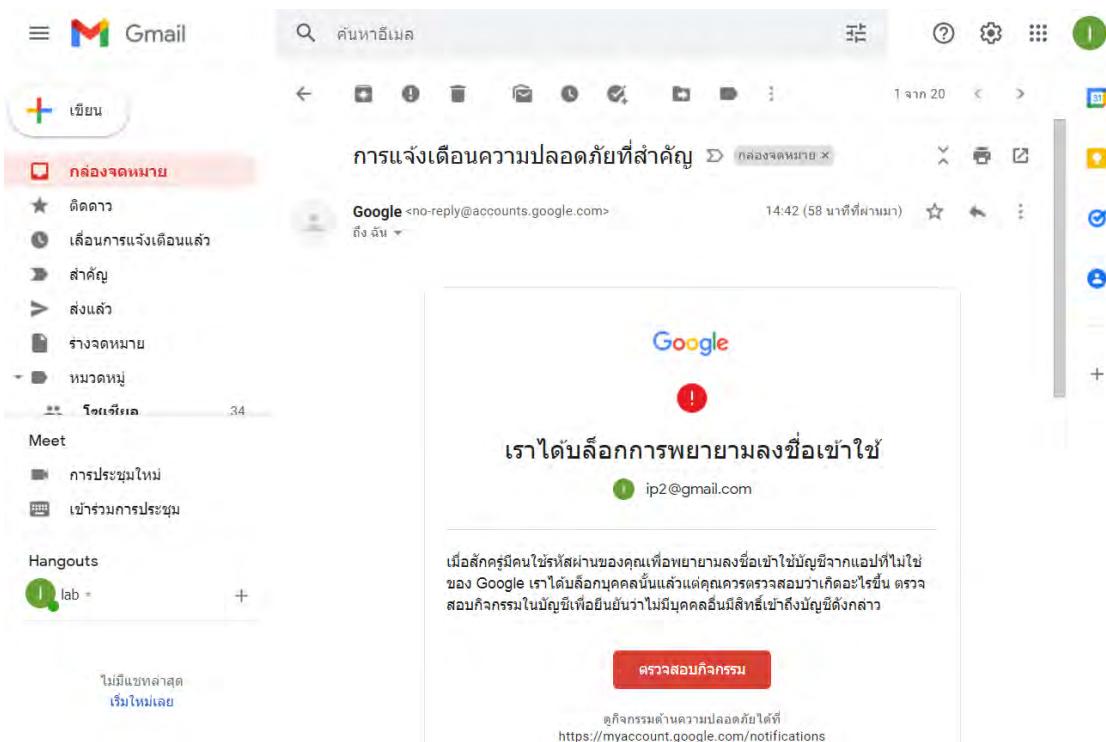
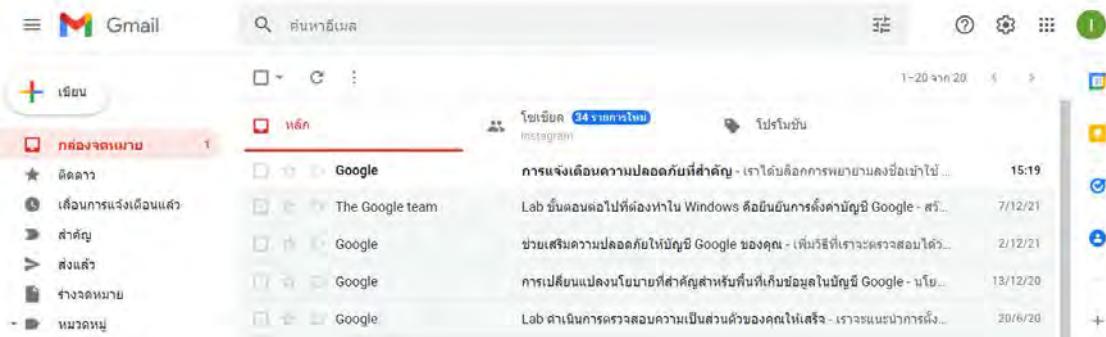
# Import smtplib for the actual sending function
# email package modules

# Sender email address
# Sender email password
# Receiver email address
# Subject

# write text message
# write image in binary mode.
# attach file
# initialize connection to SMTP server
# start communicate with TLS encryption
# login to the email server
# send the mail
# Logout of the email server
```

90. ให้แก้ไขโปรแกรมการทดลอง License Plate Recognition ที่รับสัญญาณภาพจากรูปภาพ ไปเป็นการรับสัญญาณวิดีโอจากกล้อง Webcam เพื่อมาทำระบบการตรวจจับและอ่านป้ายทะเบียนรถแบบอัตโนมัติ

วิธีตั้งค่าความปลอดภัยของแอป สำหรับใช้ส่ง e-mail



**บล็อกแอปที่มีความปลอดภัยน้อยแล้ว**

Google บล็อกแอปที่คุณพยายามไข้เพราะแอบดังกล่าวไม่เป็นไปตามมาตรฐานความปลอดภัยของเรา  
แอปและอุปกรณ์บางอย่างใช้เทคโนโลยีการลงชื่อเข้าใช้ที่มีความปลอดภัยน้อย ซึ่งทำให้บัญชีมีความเสี่ยงมากขึ้น คุณปิดการเข้าถึงของแอปเหล่านี้ได้ (ซึ่งเราแนะนำ) หรือปิดการเข้าถึงหากต้องการใช้งานแม้ว่ามีความเสี่ยงก็ตาม Google จะปิดการตั้งค่าโดยอัตโนมัติหากในมีการใช้งาน  
ดูข้อมูลเพิ่มเติม

ยกเลิก      ฉุกเฉินหั้งหนาม

← → ⌂ ⌄ ⓘ https://myaccount.google.com/u/8/security

Google บัญชี | ศูนย์การช่วยเหลือ Google | ? | ⌂

หน้าแรก      ข้อมูลส่วนบุคคล      ข้อมูลและความเป็นส่วนตัว      ความปลอดภัย **ปิด**      การชำระเงินและการสมัครสมาชิก      เกี่ยวกับ

**การเข้าถึงของแอปที่มีความปลอดภัยน้อย**

ระบบจะบล็อกแอปและอุปกรณ์ที่ใช้เทคโนโลยีการลงชื่อเข้าใช้ที่มีความปลอดภัยน้อยที่บันทึกไว้ในบัญชีของคุณ Google จะรักษาบัญชีที่บล็อกด้วยปีกการฟื้นคืนที่โดยอัตโนมัติหากในมีการใช้งาน

**การลงชื่อเข้าใช้เว็บไซต์อื่น**

การลงชื่อเข้าใช้โดย Google      คุณไม่ได้ใช้บัญชี Google เพื่อลงชื่อเข้าใช้เว็บไซต์หรือแอปใด

เครื่องมือจัดการรหัสผ่าน      คุณไม่มีรหัสผ่านที่บันทึกไว้ในบัญชี Google เครื่องมือจัดการรหัสผ่านช่วยให้การลงชื่อเข้าใช้เว็บไซต์โดยอัตโนมัติในทุกอุปกรณ์ที่คุณลงชื่อเข้าใช้

บัญชีที่ลิงก์      คุณไม่มีบัญชีที่ลิงก์ คุณไม่ได้ใช้ Google เข้าถึงข้อมูลของคนงานที่บันทึกไว้ในบัญชี Google เพื่อลงชื่อเข้าใช้เว็บไซต์หรือแอปใด

← การเข้าถึงของแอปที่มีความปลอดภัยน้อย

แอปและอุปกรณ์บางอย่างใช้เทคโนโลยีการลงชื่อเข้าใช้ที่มีความปลอดภัยน้อย ซึ่งทำให้บัญชีของคุณมีช่องโหว่ คุณปิดการเข้าถึงของแอปเหล่านี้ได้ (ซึ่งเราแนะนำ) หรือปิดการเข้าถึงหากต้องการใช้งานแม้ว่ามีความเสี่ยงก็ตาม Google จะปิดการตั้งค่าโดยอัตโนมัติหากในมีการใช้งาน ดูข้อมูลเพิ่มเติม

อนุญาตแอปที่มีความปลอดภัยน้อย: เปิด

### ขั้นตอนตั้งค่าการยืนยันแบบ 2-step

หน้าแรก

ข้อมูลส่วนบุคคล

ข้อมูลและความเป็นล้วนด้วย

ความปลอดภัย

ผู้คนและการแข่ง

การชำระเงินและการสมัครใช้บริการ

---

เกี่ยวกับ

**กิจกรรมด้านความปลอดภัยล่าสุด**

- ลงชื่อเข้าใช้ครั้งใหม่ใน Windows 12 ต.ค. >
- ความพยายามที่นำส่งสัญญาในการลงชื่อเข้าใช้ด้วยรหัสผ่านของคุณ 12 ต.ค. · ประเทศไทย >
- ลงชื่อเข้าใช้ครั้งใหม่ใน Windows 12 ต.ค. >

ตรวจสอบกิจกรรมด้านความปลอดภัย

**วิธีลงชื่อเข้าใช้ Google**

คุณได้สามารถเข้าถึงบัญชี Google ได้ด้วยรหัสผ่านเดียวที่คุณตั้งไว้ แต่คุณสามารถเปลี่ยนรหัสผ่านของบัญชีนี้ได้

<b>(๑) การยืนยันแบบ 2 ขั้นตอน</b>	<b>การยืนยันแบบ 2 ขั้นตอนมีปิดอยู่</b>
---	>
<b>---</b> รหัสผ่าน	เปลี่ยนแปลงล่าสุด 10 พ.ย. 2556
<b>---</b> หมายเลขอุตสาหกรรมที่รับการอนุมัติ	เพิ่มหมายเลขอุตสาหกรรมที่มือถือ
<b>---</b> อีเมลสำหรับการรีเซ็ต	เพิ่มที่อยู่อีเมล

คุณเพิ่มเติมแล้วก็อีกครั้ง ในการลงชื่อเข้าใช้ได้

**ฟ้าสีฟ้า**

[ความเป็นส่วนตัว](#) | [ข้อกำหนด](#) | [ความช่วยเหลือ](#)

### ← การยืนยันแบบ 2 ขั้นตอน



ปกป้องบัญชีของคุณด้วยการยืนยันแบบ 2 ขั้นตอน

เพิ่มการปกป้องขั้นเพิ่มป้องกันไม่ให้แฮกเกอร์เข้าถึงบัญชี หรือคุณลงชื่อเข้าใช้ การยืนยันแบบ 2 ขั้นตอนจะช่วยข้อมูลส่วนบุคคลของคุณเป็นส่วนตัวและปลอดภัย

รักษาความปลอดภัยได้ง่ายๆ

หากเหตุการณ์ฉุกเฉินที่คุณต้องการเข้าสู่บัญชีของคุณ คุณสามารถรีเซ็ตรหัสผ่านได้โดยกดที่ 2 ที่อยู่ยืนยันตัวตนของคุณได้อย่างรวดเร็ว

ใช้การยืนยันแบบ 2 ขั้นตอนกับทุกบัญชีออนไลน์

การยืนยันแบบ 2 ขั้นตอนเป็นเครื่องมือที่มีประสิทธิภาพและรวดเร็วช่วยป้องกันการโจมตีทางไซเบอร์ที่พบได้ทั่วไป เป็นชื่อของน้ำที่ถูกดึงที่มีให้บริการเพื่อปกป้องบัญชีของคุณให้ปลอดภัย

**Safer with Google**

**เริ่มต้นใช้งาน**

**Google บัญชี**

ผลการค้นหาด้วยคำว่า “ห้องน้ำ” เมื่อวันที่ 16 พ.ค. 2566

ผลการค้นหา

**← การยืนยันแบบ 2 ขั้นตอน****ขั้นตอนที่ 2 ที่ใช้ได้**

ขั้นตอนที่ 2 หลังจากป้อนรหัสผ่านจะยืนยันว่าคุณเป็นผู้ลงชื่อเข้าใช้ คุ้มครองเพิ่มเติม  
หมายเหตุ: หากคุณลงชื่อเข้าใช้บัญชี Google ในโทรศัพท์ที่มีสิทธิ์ ระบบจะเพิ่ม Google Prompt เป็นอีกชั้นการยืนยันแบบ 2 ขั้นตอน

**ข้อความหรือข้อความเสียง (ค่าเริ่มต้น) ②**

099 792 9999 ได้รับการยืนยันแล้ว

ระบบจะส่งรหัสยืนยันทางข้อความ

**เพิ่มขั้นตอนที่ 2 อีก 1 เพื่อยืนยันว่าเป็นคุณ**

ตั้งค่าขั้นตอนสำรองเพิ่มเติมเพื่อให้คุณลงชื่อเข้าใช้ได้เมื่อว่าด้วยกันอีกอีก 1 จะไม่พร้อมใช้งาน

**รหัสสำรอง**

รหัสผ่านแบบใช้ครั้งเดียวและลังทิมที่ได้เหล่านี้ช่วยให้คุณลงชื่อเข้าใช้เมื่อไม่ได้พก  
โทรศัพท์มา เช่น ในขณะเดินทาง

**Google Prompt**

หากต้องการรับ Google Prompt เพียงลงชื่อเข้าใช้บัญชี Google ในโทรศัพท์

หลังจากที่ป้อนรหัสผ่านในอุปกรณ์เครื่องใหม่ Google จะส่งข้อความแจ้งไปยัง  
โทรศัพท์ทุกเครื่องที่คุณลงชื่อเข้าใช้ในอุปกรณ์ แต่ที่ข้อความแจ้งได้เพียงบัญ

บัญชีบันคุณไม่ได้ลงชื่อเข้าใช้อุปกรณ์ที่รองรับ Google Prompt

**แอป Authenticator**

ใช้อปAuthenticator เพื่อรับรหัสยืนยันได้โดยไม่มีค่าใช้จ่ายแม่万一โทรศัพท์จะลอก  
ไลน์ออย พร้อมให้เข้ากันกับ Android และ iPhone

**คีย์ความปลอดภัย**

คีย์ความปลอดภัยต่อวิธีการยืนยันที่ช่วยให้คุณลงชื่อเข้าใช้ได้อย่างปลอดภัย คีย์ความ  
ปลอดภัยเหล่านี้มีอยู่ในโทรศัพท์ ใช้ผ่านบลูทูธ หรือเสียบเข้ากับพอร์ต USB ของ  
คอมพิวเตอร์โดยตรง

**อุปกรณ์ที่ไม่ต้องใช้ขั้นตอนที่ 2**

คุณสามารถข้ามขั้นตอนที่ 2 บนอุปกรณ์ต่างๆ ที่คุณเข้าถึง เช่น คอมพิวเตอร์ของคุณเอง

**อุปกรณ์ที่คุณเข้าถึง**

เพิกถอนสถานะที่เข้าถึงได้จากอุปกรณ์ที่ข้ามการยืนยันแบบ 2 ขั้นตอน

เพิกถอนทั้งหมด

**รหัสผ่านสำหรับแอป**

รหัสผ่านสำหรับแอปจะเป็นในการที่ส่วนใหญ่และเราไม่แนะนำให้ใช้ ใช้ "ลงชื่อเข้าใช้ด้วย Google" เพื่อเข้ามายังแอป  
บัญชี Google เพื่อช่วยรักษาบัญชีให้ปลอดภัยสมอ

**รหัสผ่านสำหรับแอป**

ไม่มี

## ← รหัสผ่านสำหรับแอป

รหัสผ่านสำหรับแอปช่วยให้คุณลงชื่อเข้าใช้บัญชี Google ได้ในแอปและบริการเก่าที่ไม่รองรับมาตรฐานความปลอดภัยสมัยใหม่

รหัสผ่านสำหรับแอปมีความปลอดภัยกว่าการใช้แอปและบริการที่อัปเดตล่าสุดซึ่งไข้มัตตรฐานความปลอดภัยสมัยใหม่ ก่อนสร้างรหัสผ่านสำหรับแอป คุณควรตรวจสอบว่าแอปต้องใช้รหัสผ่านนี้เพื่องชื่อเข้าใช้หรือไม่ ดูข้อมูลเพิ่มเติม

คุณไม่มีรหัสผ่านสำหรับแอปเลย

To create a new app specific password, type a name for it below...

App name  
raspberrypi

สร้าง

## ← รหัสผ่านสำหรับแอป

รหัสผ่านสำหรับแอปช่วยให้คุณลงชื่อเข้าใช้บัญชี Google ได้ในแอปและบริการเก่าที่ไม่รองรับมาตรฐานความปลอดภัยสมัยใหม่

รหัสผ่านสำหรับแอปนี้  
ไข้มัตตรฐานความปลอดภัย  
และต้องใช้รหัสผ่านนี้  
ดูข้อมูลเพิ่มเติม

รหัสผ่านของแอปที่สร้างขึ้น

รหัสผ่านของแอปสำหรับอุปกรณ์

**hmyw sbwi nnpz kiid**

ใช้ใช่  
ไปที่การตั้งค่าสำหรับบัญชี Google ในแอปพลิเคชันหรืออุปกรณ์ที่ต้องการจะตั้งค่า เปลี่ยนรหัสผ่านเป็นรหัสผ่าน 16 วอชระที่แสดงอยู่ด้านบน  
เขียนเทียบกับรหัสผ่านทั่วไป รหัสผ่านสำหรับแอปนี้ให้ลืมรหัสผ่านที่ใช้บัญชี Google อีกต่อไป  
สมบูรณ์ คุณไม่จำเป็นต้องจารหัสผ่านนี้ ดังนั้นอย่าจดไว้ในกระดาษหรือแชร์รหัสผ่านนี้กับใคร

เสร็จ

สร้าง

91. ให้ใส่โปรแกรม DroidCam หรือ Webcams อื่นๆ เพื่อใช้กล้องของเครื่องโทรศัพท์ที่เป็นระบบปฏิบัติการ Android/iOS แล้วเปิดโปรแกรมจะขึ้นหมายเลข IP และ PORT

92. ให้เปิดเครื่องคอมพิวเตอร์ PC หรือ Raspberry Pi เพื่อทดลอง Webcams โดยแก้ไขโปรแกรมการเปิดใช้งานกล้องด้านล่าง ด้วยการใส่หมายเลข IP และ PORT ของโทรศัพท์ลงในโปรแกรม

```
import cv2
cap = cv2.VideoCapture("http://192.168.1.60:4747/video")      # IP:PORT/video
while(True):
    ret, frame = cap.read()
    cv2.imshow('Frame',frame)

    if cv2.waitKey(1) == 32:
        break
    cap.release()
    cv2.destroyAllWindows()
```

93. ถ้าภาพวิดีโอที่ได้มีข้อความหรือโฆษณาบนภาพ ให้ตัดบางส่วนของภาพที่ไม่ต้องการออก โดยกำหนดตำแหน่งที่ต้องการใช้งาน  $y1:y2,x1:x2$  ด้วยคำสั่ง

```
crop = frame[12 :, :]
cv2.imshow('Crop', crop)                                     # y1 = 12
```

94. การทดลองเครื่องเนล เป็นรูปภาพเมมทริกซ์ขนาดเล็กที่ใช้ในการใส่อเฟเฟกต์ เช่น การเบลอภาพ การเพิ่มความคมชัด การหาขอบ โครงสร้างของภาพ การทำให้ภาพมุนขึ้น และยังใช้ในการเรียนรู้เทคนิคเพื่อการแยกคุณลักษณะในการกำหนดส่วนที่ต้องการของรูปภาพ การทำงานเมมทริกซ์ของภาพ 1 พิกเซลของสีแต่ละสีจะประกอบด้วยตัวเลขระหว่าง 0 ถึง 255 ซึ่งแต่ละตัวเลขจะสัมพันธ์กับความสว่างของจุดภายในภาพนั้น ตัวอย่างการใช้เครื่องเนลในพิกเซลแต่ละบล็อกจะเป็นเมมทริกซ์ขนาด  $3 \times 3$  เราจะคุณเมมทริกซ์แต่ละพิกเซลด้วยรายการของเครื่องเนลตามค่าที่ต้องการ จากนั้นจึงหาค่าเฉลี่ยผ่อนรวมกันเป็นพิกเซลค่าใหม่ภายในรูปภาพ ให้ทดลอง kernel โดยเปิดไฟล์รูปภาพชื่อ picture.jpg ด้วยโปรแกรมดังนี้

```
import cv2
import numpy as np

kernel1 = np.array([[ 0,  0,  0],
                   [ 0,  1,  0],
                   [ 0,  0,  0]])                                # Identity kernel

kernel2 = np.array([[ 1,  4,  6,  4,  1],
                   [ 4, 16, 24, 16, 4],
                   [ 6, 24, 36, 24, 6],
                   [ 4, 16, 24, 16, 4],
                   [ 1,  4,  6,  4,  1]]) / 255                 # Gaussian blur

kernel3 = np.array([[ 1,  1,  1],
                   [ 1,  1,  1],
                   [ 1,  1,  1]]) / 9                            # Box blur

kernel4 = np.array([[ 0, -1,  0],
                   [-1,  5, -1],
                   [ 0, -1,  0]])                                # Sharpen

kernel5 = np.array([[-1, -1, -1],
                   [-1,  8, -1],
                   [-1, -1, -1]])                                # Edge detection

kernel6 = np.array([[-2, -1,  0],
                   [-1,  1,  1],
                   [ 0,  1,  2]])                                # Emboss

kernel7 = np.array([[-1,  0,  1],
                   [-2,  0,  2],
                   [-1,  0,  1]])                                # right sobel
```

```
image = cv2.imread('picture.jpg')
print('Image Dimensions : ',image.shape) # height, width, number of channels
cv2.imshow('original image', image)

dst = cv2.filter2D(src=image, ddepth=-1, kernel=kernel1) # depth is same as original
cv2.imshow('filter2d image', dst)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

95. ให้แก้ไขโปรแกรมเปลี่ยนการเปิดไฟล์ภาพไปเป็นการเปิด Video ของ Webcam จากกล้องของเครื่องโทรศัพท์
  96. ให้ทดลองแสดงผลลัพธ์ที่ได้จาก image kernel แบบต่างๆ และให้อธิบายค่าต่างๆ ของเมทริกซ์มีผลต่อภาพที่ได้ในแต่ละแบบอย่างไร .

97. ให้แก้ไขโปรแกรมเพิ่มฟิลเตอร์ใน image kernel มา 10 แบบ และให้ทดลองการแสดงผลลัพธ์ที่ได้พร้อมทั้งอธิบายค่าต่างๆของเมทริกซ์ผลต่อภาพที่ได้อ่านไว้ .....

.....  
.....  
.....  
.....

ภาษาไพทอน (Python programming language) เป็นโปรแกรมภาษาคอมพิวเตอร์ระดับสูง (High-level programming language) สนับสนุนแนวคิดการเขียนโปรแกรมเชิงวัตถุ หรือ OOP (Object Oriented Programming) ถูกออกแบบมาให้เป็นภาษา Script ที่อ่านง่าย มีโครงสร้างข้อมูลระดับสูงประกอบกันเป็น Dynamic Typing คือสามารถเปลี่ยนชนิดข้อมูลได้ง่าย โดยตัดความซับซ้อนทางโครงสร้างและไวยากรณ์ของภาษาออกไป สามารถอ่านชุดคำสั่งได้ง่ายโดยใช้ช่องว่าง (Whitespace) ตามตำแหน่ง Column เพื่อแบ่งชุดคำสั่งแต่ละบรรทัดว่าอยู่ภายใต้ลักษณะใด ระบบปฏิบัติการ Linux สมัยใหม่ส่วนใหญ่ในปัจจุบัน Python จะได้รับการติดตั้งให้เป็นค่าเริ่มต้น และยังมีไลบรารีมาตรฐานจำนวนมากในรูปแบบ Open Source โดยมาพร้อมกับแอปพลิเคชัน IDLE (Integrated Development Environment) อย่างไรก็ตามหากไม่ได้ติดตั้งมาจะสามารถติดตั้งได้โดยใช้ตัวจัดการแพคเกจด้วยคำสั่ง

```
sudo apt-get install idle3
```

การทดลองจะใช้ภาษา Python เป็นภาษาที่ Raspberry Pi แนะนำให้ใช้ โดยที่จะมีไลบรารีในการเข้ามือต่อสารด้วยจำนวนมาก ภาษา Python ไม่ต้องประกาศ Type หน้าตัวแปร เริ่มจากการนำ Raspberry Pi 3 Model B+ ต่อ กับ พอร์ต GPIO แล้วทำการเชื่อมต่ออุปกรณ์ที่ใช้ในการทดลองลงบนพอร์ตGPIO 14, 15, 18 2, 3 ของ Raspberry Pi ส่งค่าสัญญาณเอาท์พุตแบบดิจิตอลออกมานะ และนำ RPi.GPIO ซึ่งเป็นโมดูลสำหรับอินเตอร์เฟสกับพอร์ต GPIO ของ Raspberry Pi มาใช้ ด้วยคำสั่ง `GPIO.output(pin,value)` โดยที่ค่า `value` ของสัญญาณดิจิตอลที่ได้มีอยู่ 2 รูปแบบคือสัญญาณ True (1) และ False (0) ถ้าในบอร์ดบังไม่ได้ติดตั้งสามารถติดตั้ง python module ชื่อ `rpi.gpio` ได้ด้วยคำสั่ง

```
sudo apt install python3-rpi.gpio
```

การเขียนโค้ดคำสั่งในการกำหนดชื่อของขาต่างๆจะมี 2 แบบ คือ Board Number และ BCM Number ใน การทดลองนี้จะเลือกใช้แบบ BCM Number โดยจะอ้างอิง ชื่อขาจากซีพียูประมวลผล BCM เป็นตำแหน่งขา GPIO Pinout โดยในแต่ละขาของ GPIO จะมีรายละเอียดข้อกำหนดของแรงดันไฟฟ้าต่างๆ ดังนี้

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{IL}$	Input low voltage <sup>a</sup>	$V_{DD\_IO} = 1.8V$	-	-	0.6	V
		$V_{DD\_IO} = 2.7V$	-	-	0.8	V
		$V_{DD\_IO} = 3.3V$	-	-	0.9	V
$V_{IH}$	Input high voltage <sup>a</sup>	$V_{DD\_IO} = 1.8V$	1.0	-	-	V
		$V_{DD\_IO} = 2.7V$	1.3	-	-	V
		$V_{DD\_IO} = 3.3V$	1.6	-	-	V
$I_{IL}$	Input leakage current	$TA = +85^{\circ}C$	-	-	5	$\mu A$
$C_{IN}$	Input capacitance	-	-	5	-	pF
$V_{OL}$	Output low voltage <sup>b</sup>	$V_{DD\_IO} = 1.8V, I_{OL} = -2mA$	-	-	0.2	V
		$V_{DD\_IO} = 2.7V, I_{OL} = -2mA$	-	-	0.15	V
		$V_{DD\_IO} = 3.3V, I_{OL} = -2mA$	-	-	0.14	V
$V_{OH}$	Output high voltage <sup>b</sup>	$V_{DD\_IO} = 1.8V, IOH = 2mA$	1.6	-	-	V
		$V_{DD\_IO} = 2.7V, IOH = 2mA$	2.5	-	-	V
		$V_{DD\_IO} = 3.3V, IOH = 2mA$	3.0	-	-	V
$I_{OL}$	Output low current <sup>c</sup>	$V_{DD\_IO} = 1.8V, VO = 0.4V$	12	-	-	mA
		$V_{DD\_IO} = 2.7V, VO = 0.4V$	17	-	-	mA
		$V_{DD\_IO} = 3.3V, VO = 0.4V$	18	-	-	mA
$I_{OH}$	Output high current <sup>c</sup>	$V_{DD\_IO} = 1.8V, VO = 1.4V$	10	-	-	mA
		$V_{DD\_IO} = 2.7V, VO = 2.3V$	16	-	-	mA
		$V_{DD\_IO} = 3.3V, VO = 2.3V$	17	-	-	mA
$R_{PU}$	Pullup resistor	-	50	-	65	$k\Omega$
$R_{PD}$	Pulldown resistor	-	50	-	65	$k\Omega$

<sup>a</sup> Hysteresis enabled

<sup>b</sup> Default drive strength (8mA)

<sup>c</sup> Maximum drive strength (16mA)

- การทดลองต่อวงจรเพื่อสั่งงานให้ Raspberry Pi ส่งข้อมูลต่างๆ ไปออกที่ LED จำนวน 5 หลอด โดย LED ที่เป็นไดโอดเปล่งแสงทุกตัว จะต้องต่อขา Cathode เข้ากับบอร์ดไฟลอนกราวด์ และให้ขา Anode ในแต่ละตัวต่อ กับขาด้วยความต้านทาน  $220\ \Omega$  แล้วให้นำขาด้วยความต้านทานที่เหลือแต่ละตัวต่อเข้ากับขา GPIO 14, 15, 18, 2, 3 ของบอร์ด Raspberry Pi ตามตามลำดับ
- ให้เชื่อมต่อสาย USB ของบอร์ด Raspberry Pi กับอะแดปเตอร์จ่ายไฟ USB 5V. เปิดโปรแกรม Thonny Python IDE ที่มีติดตั้งไว้ใน Raspberry Pi จากนั้นทำการเขียนโปรแกรมที่ทำหน้าที่สั่งงานให้ LED ที่อยู่บนบอร์ด กระพริบทุก 1 วินาที แล้วทำการ Run โปรแกรม

```

import RPi.GPIO as GPIO          # เพิ่มไลบรารีในชื่อ GPIO
import time

channel1 = 14                   # GPIO 14  PIN 8   TX
channel2 = 15                   # GPIO 15  PIN 10  Rx
channel3 = 18                   # GPIO 18  PIN 12
channel4 = 2                     # GPIO 2   PIN 3   SDA
channel5 = 3                     # GPIO 3   PIN 5   SCL

GPIO.setmode(GPIO.BCM)           # กำหนดรูปแบบขา GPIO number (BCM mode)
GPIO.setup(channel1, GPIO.OUT)    # กำหนดขาเป็น input/output

GPIO.output(channel1, True)      # ส่งค่า output เป็น 1 ออกไปที่ขา channel1
print ("LED1 on")               # พิมพ์ข้อความ
time.sleep(1)                   # หน่วงเวลา (วินาที)
GPIO.output(channel1, False)     # ส่งค่า output เป็น 0 ออกไปที่ขา channel1
print ("LED1 off")

GPIO.output(channel2, True)      # ส่งค่า output เป็น 1 ออกไปที่ขา channel2
print ("LED2 on")               # พิมพ์ข้อความ
time.sleep(1)                   # หน่วงเวลา (วินาที)
GPIO.output(channel2, False)     # ส่งค่า output เป็น 0 ออกไปที่ขา channel2
print ("LED2 off")

GPIO.output(channel3, True)      # ส่งค่า output เป็น 1 ออกไปที่ขา channel3
print ("LED3 on")               # พิมพ์ข้อความ
time.sleep(1)                   # หน่วงเวลา (วินาที)
GPIO.output(channel3, False)     # ส่งค่า output เป็น 0 ออกไปที่ขา channel3
print ("LED3 off")

GPIO.output(channel4, True)      # ส่งค่า output เป็น 1 ออกไปที่ขา channel4
print ("LED4 on")               # พิมพ์ข้อความ
time.sleep(1)                   # หน่วงเวลา (วินาที)
GPIO.output(channel4, False)     # ส่งค่า output เป็น 0 ออกไปที่ขา channel4
print ("LED4 off")

GPIO.output(channel5, True)      # ส่งค่า output เป็น 1 ออกไปที่ขา channel5
print ("LED5 on")               # พิมพ์ข้อความ
time.sleep(1)                   # หน่วงเวลา (วินาที)
GPIO.output(channel5, False)     # ส่งค่า output เป็น 0 ออกไปที่ขา channel5
print ("LED5 off")

GPIO.cleanup()                  # เคลียร์ค่าพาวเวอร์หลังเลิกใช้

```

- จากการทดลองให้อธิบายการทำงานของโปรแกรม
- .....
- .....

4. หลังจากที่โปรแกรมทำงานเสร็จแล้ว เมื่อหยุดการทำงานให้สังเกตว่าหลอด LED ได้ดับหมดทุกดวงหรือไม่ ถ้ายังมีหลอดที่ไฟยังติดอยู่ให้นอกกว่ามีขาไฟบ้าง และเหตุใดจึงเป็นเช่นนี้

## คำสั่งการวนทําซํ้า while

## while Condition: Statements

•

โดยที่สามารถออกจาก while ได้คือคำสั่ง break

5. จากโปรแกรมในข้อ 2 ให้แก้ไขโปรแกรมเพื่อให้โปรแกรมทำงานตลอดเวลาโดยไม่ให้หยุด เมื่อมีการทำงานของฟังก์ชัน void loop() ใน Arduino ที่ทำตามคำสั่งต่างๆที่เขียนไว้วนรอบซ้ำกันไป โดยให้ใช้คำสั่ง

while 1 :

6. จากโปรแกรมในข้อ 5 ให้แก้ไขโปรแกรมเพื่อให้โปรแกรมทำงานโดยให้หลอดที่ 1 ถึง 3 ทำงานตามลำดับ  
จำนวน 5 รอบก่อน แล้วจึงให้หลอดที่ 4, 5 ทำงานตามมา หลังจากนั้นให้วุ่นซ้ำหนึ่งครั้ง โดยให้เพิ่มคำสั่ง

```
for i in range(5):
```

โดยที่เล่น 5 หมายความว่าให้วันซึ่งจำนวน 5 รอบ

7. ให้ต่อความดันท่าน 10 K  $\Omega$  อนุกรมกับสวิทช์ แล้วนำปลายขาข้างหนึ่งต่อ กับ Vcc ของบอร์ด Raspberry Pi และปลายขาอีกข้างต่อลงกราวด์ แล้วให้ออก GPIO27 ต่อเข้ากับจุดต่อร่วมระหว่างความดันท่านกับสวิทช์ โดยจะต้องมีการกำหนดค่าขาเพิ่มคือ

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

## កំសៀវភៅការព័ត៌មានឱ្យ if

## if Condition: Statements

1

8. จากระบบในข้อที่ 7 ให้เขียนโปรแกรมทดสอบการกดสวิทช์โดยแก้ไขเพิ่มจากโปรแกรมในข้อที่ 6 กำหนดเงื่อนไขไว้ว่า ถ้ามีการไม่กดสวิทช์ให้ LED ทั้งหมดตั้งแต่หลอดที่ 1 ถึง 5 ทำงานตามลำดับเหมือนในข้อที่ 6 แต่ถ้ากดสวิทช์ให้ LED ทั้งหมดหยุดทำงาน โดยให้ใช้คำสั่ง

```
If GPIO.input(channel) == 1 :
```

โดยที่เลข 1 ในที่นี้อาจจะเป็น 1 หรือ 0 ก็ได้ ขึ้นอยู่กับการต่อตัวความต้านและสวิทช์ว่าเอาด้านไหนต่อกับ Vcc หรือ กราวด์

คำสั่งการตัดสินใจ if else

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

9. จากข้อที่ 8 ให้แก้ไขโปรแกรมใหม่ โดยกำหนดเงื่อนไขไว้ว่า ถ้ามีการไม่กดสวิทช์ให้ LED ตั้งแต่หลอดที่ 1 ถึง 4 ทำงานตามลำดับเหมือนในข้อที่ 6 แต่ถ้ากดสวิทช์ให้ LED เลพาะหลอดที่ 5 ทำงานติดค้างไว้จนกว่าจะปล่อยสวิทช์ โดยให้เพิ่มคำสั่ง

else :

10. บนบอร์ด Raspberry Pi สามารถที่จะกำหนด Internal pull-up/pull-down enable/disable "ได้ ดังนี้  
จากระบบในข้อที่ 9 ให้แก้วงจร โดยเอาตัวความต้านทาน  $10\text{ k}\Omega$  ที่ต่อ กับสวิทช์ออก และไว้ให้แก้ไขโปรแกรมให้วงจรทำงานเหมือนเดิม โดยให้ใช้คำสั่งในการกำหนดขา Input ใหม่เป็น

```
GPIO.setup(channel, GPIO.IN, pull_up_down=GPIO.PUD_UP)  
หรือ  
GPIO.setup(channel, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

11. จากระบบในข้อที่ 10 ให้สังเกตว่า เมื่อกดสวิทช์จะเปลี่ยนการทำงานมาทำหลอดที่ 5 ทันทีหรือไม่ เพราะอะไร  
แล้วก็จะให้ทำทันทีจะสามารถทำได้หรือไม่อย่างไร
- .....  
.....  
.....  
.....

### คำสั่งการสร้างฟังก์ชัน

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

โดยที่

```
def กำหนดค่าเป็นฟังก์ชัน
Function_name ชื่อฟังก์ชัน
Argument ค่าที่รับเข้าไปในฟังก์ชัน
Statement ชุดคำสั่ง
return การส่งค่ากลับของฟังก์ชัน
```

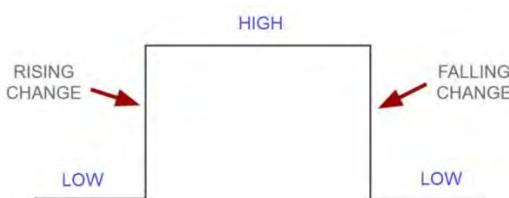
- จากข้อที่ 10 ให้แก้ไขโปรแกรม โดยกำหนดเงื่อนไขไว้ว่า เมื่อมีการกดสวิตช์ขณะที่ LED ตั้งแต่หกอตที่ 1 ถึง 4 กำลังทำงานปกติตามลำดับ ให้ทำการจำเอาไว้จนกว่าการทำงานจะวนมาถึงหลอดสุดท้ายคือหลอดที่ 4 แล้วจึงเปลี่ยนให้มาทำงานที่หลอดที่ 5 ให้ส่วน 1 วินาทีทุกครั้งที่มีการกดสวิตช์ หลังจากนั้นจึงกลับทำงานไปวนซ้ำ ตั้งแต่หกอตที่ 1 ถึง 4 เหมือนเดิมตามปกติ โดยให้ใช้คำสั่งเพิ่มเป็นคำสั่งที่ใช้ในการ Interrupt ซึ่งสามารถกำหนดได้ดังนี้

```
status = 0

def switch1(button):
    global status
    status = 1

GPIO.add_event_detect(channel, GPIO.FALLING, callback=switch1, bouncetime=100)
```

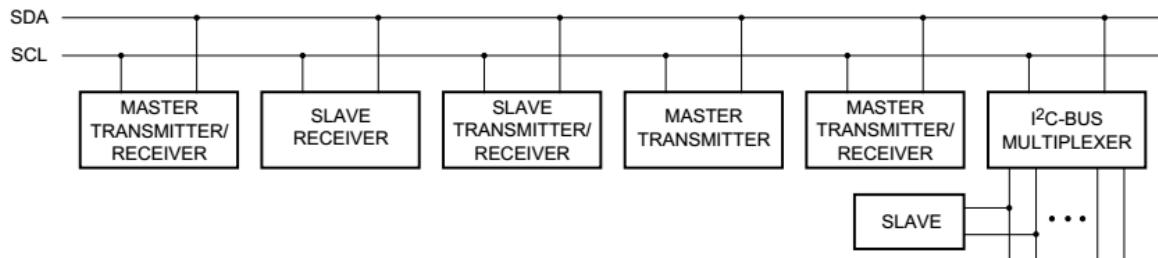
โดยที่การทำงานของการ Interrupt นี้ จะต้องมีการทริกให้ทำงาน ซึ่งสามารถกำหนดได้ว่าจะให้ทำงานเมื่อสถานะของสัญญาณเปลี่ยนจาก LOW ไปเป็น HIGH หรือเปลี่ยนจาก HIGH ไปเป็น LOW



callback คือ การกำหนดฟังก์ชันที่จะถูกเรียกใช้ให้ไปทำงานเมื่อมีการ Interrupt ในที่นี้คือ switch1

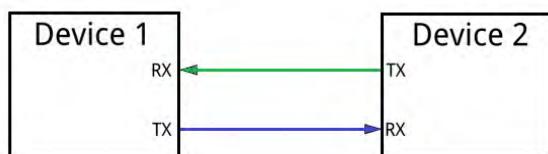
bouncetime คือ ค่าช่วงเวลาของสัญญาณที่เกิดจากการกดปุ่มแต่ละครั้ง โดยที่เมื่อกดปุ่มหน้าสัมผัสที่ทำการสัมผัสนั้นชั่วไม่แนบสนิทจะทำให้เกิดสัญญาณวิ่งขึ้นลงหรือสัญญาณรบกวน ถ้าในช่วงเวลาที่กำหนดมีการขึ้นลงมากกว่าหนึ่งครั้งจะให้คือว่ามีการกดสวิตช์เพียงครั้งเดียว ค่านี้มีหน่วยเป็นมิลลิวินาที

**I<sup>2</sup>C (Inter-Integrated Circuit)** เป็นโปรโตคอลมาตรฐานที่ใช้สำหรับการติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์กับอุปกรณ์ภายนอก ด้วยสายสัญญาอนุกรมแบบซิงโครนัส (Synchronous) โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้นคือ SDA (Serial Line Data) และ SCL (Serial Clock) บางทีอาจเรียกว่า TWI (Two Wire Interface) สามารถเชื่อมต่ออุปกรณ์หรือไอซีจำนวนหลายตัวเข้าด้วยกันได้ แต่จะสามารถส่งข้อมูลระหว่างกันได้ครึ่งละครึ่งเท่านั้น โดยมีตัว Master ควบคุมการส่งข้อมูล ต้องการติดต่อสื่อสารกับอุปกรณ์อะไรได้ครึ่งละตัว และมีตัว Slave จะถูกสั่งให้รับข้อมูลหรือส่งข้อมูลกลับ สามารถมี Slave และ Master รวมกันได้ถึง 128 ถูกพัฒนาขึ้นโดยบริษัท Phillips Semiconductor ตั้งแต่ปี คศ.1982 ซึ่งในปัจจุบันคือบริษัท NXP Semiconductor โปรโตคอลแบบ I<sup>2</sup>C มีโหมดที่ใช้ความเร็วในการรับส่งข้อมูลตั้งแต่ 100 Kbps ไปจนถึง 5 Mbit/s

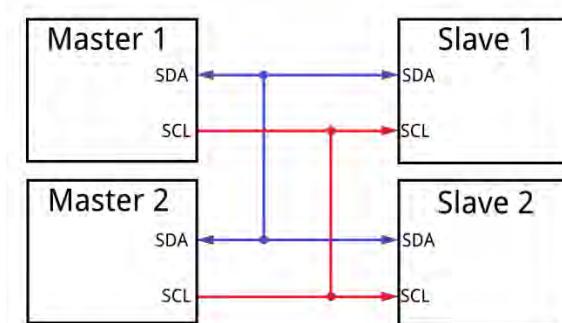


System configuration

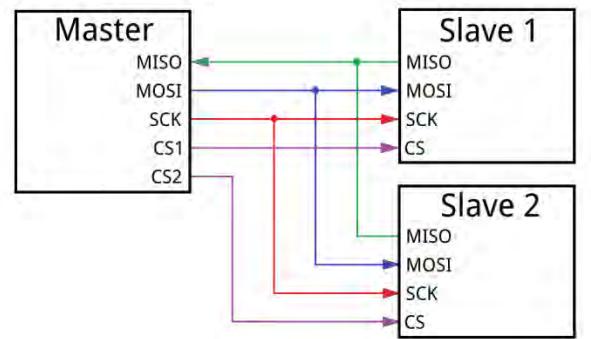
ส่วนอีกโปรโตคอลซึ่งเป็นที่นิยมใช้ เช่นกันคือ SPI (Serial Peripheral Interface) ความแตกต่างระหว่าง I<sup>2</sup>C และ SPI คือจำนวนสายที่ใช้ในการติดต่อ I<sup>2</sup>C ใช้ 2 เส้น ส่วน SPI ใช้สายสัญญาณ 4 เส้นและเพิ่มจำนวนของสายสัญญาณตามจำนวนอุปกรณ์ที่มากขึ้น การสื่อสารของ I<sup>2</sup>C จะใช้สายที่น้อยกว่า เพราะเป็นแบบ Half Duplex และความเร็วที่ใช้ในการส่งของ I<sup>2</sup>C จะมากกว่าแบบ SPI ที่เป็นการสื่อสารแบบ Full Duplex ซึ่งมีสายสัญญาณข้อมูลแยกกันเป็นการรับ (MISO) และส่ง (MOSI) อย่างละ 1 เส้น แต่ก็เพียงพอสำหรับการงานทั่วไป ตัวอย่าง Interface ที่ใช้ในการสื่อสารแต่ละแบบแสดงดังรูป



UART Interface diagram

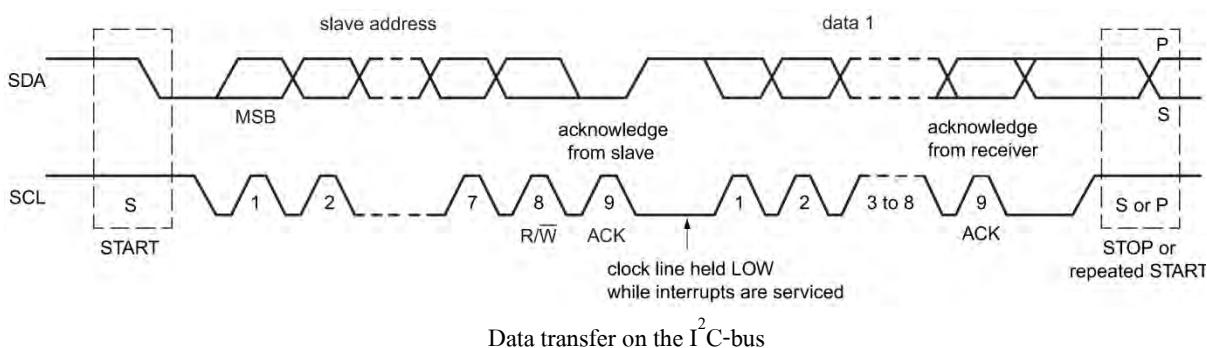


I2C interface diagram

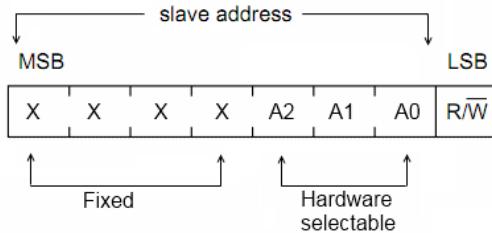


SPI interface diagram

ขั้นตอนการส่งข้อมูลด้วยโปรโตคอล I2C ต้องต่อสายตามที่แสดงในรูปคือ VCC , GND , SDA และ SCL โดยขาสัญญาณ SDA และ SCL จะเป็นขาสัญญาณเอต์พุทที่มีลักษณะเป็นแบบ Open-drain หรือ Open-collector ซึ่งจะต้องใช้ไฟเลี้ยงจากแหล่งจ่ายไฟภายนอกไอซี จึงต้องต่อ กับตัวด้านท่านแบบ Pull up ไว้ เพื่อจะให้อเอต์พุตสามารถชี้อมต่องกันได้หลายตัวพร้อมกัน การทำงานตัว Master จะเป็นตัวเรียกไปที่ Slave โดยส่ง Address ไปที่สายสัญญาณข้อมูล SDA ซึ่ง Master จะเป็นตัวควบคุม Slave ให้ทำงานเข้าจังหวะกับ Clock ซึ่งจะถูกส่งโดยใช้สาย SCL ตัว Slave ที่มี Address ตรงกับที่ถูกเรียกจะส่งสัญญาณ Acknowledge ตอบกลับไปยังสายสัญญาณ SDA ด้วยวิธีการ Pull down สายให้แรงดันเป็น 0 V ดำเนินขั้นตอนการทำงาน (Timing Diagram) เป็นดังนี้

Data transfer on the I<sup>2</sup>C-bus

1. ในสภาวะปกติที่ยังไม่มีการใช้งานจะเป็นสภาวะบัสว่าง โดยสายสัญญาณ SDA และ SCL จะมีแรงดันเป็น High ทั้งคู่
2. เมื่อมาราเตอร์ (Master) ซึ่งอุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อนบัส I2C ต้องการรับ-ส่งข้อมูล จะส่งสภาวะเริ่มต้น (START) เพื่อแสดงการขอใช้บัส โดย Master จะส่งให้มีการเปลี่ยนสัญญาณของสาย SDA จาก High ไปเป็น Low ในขณะที่ SCL จะยังคงเป็น High อยู่
3. ต่อจากนั้นตัว Master จะส่งให้สัญญาณของสาย SCL ลงไปเป็น Low และ Master จะเริ่มส่งข้อมูลค่า Address บิทแรกไปที่สาย SDA ตามด้วยสัญญาณ Clock ไปที่สาย SCL และสเตฟ(Slave) ซึ่งเป็นอุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I2C จะเริ่มอ่านค่าในจังหวะที่ SCL เป็น High โดยอุปกรณ์ที่ต่ออยู่บนบัสอาจจะมีอุปกรณ์ต่อพ่วงกันหลายตัวก็ได้ โดยค่า Address นี้จะเป็นรหัสควบคุม (Control Byte) ที่ประกอบไปด้วย 3 ส่วน คือ
  - รหัสประจำตัวของอุปกรณ์ (Device ID) มีขนาด 4 บิท ถูกกำหนดโดยผู้ผลิต ไอซีจะเปลี่ยนแปลงหรือแก้ไขไม่ได้
  - Device Address มีขนาด 3 บิท สามารถกำหนดเองได้จากการต่อขาสัญญาณโลจิกให้กับไอซี
  - Mode เป็นบิทควบคุมการอ่านหรือเขียนข้อมูลกับอุปกรณ์ภายนอก



สภาวะของช่วงเวลาการรับ-ส่งข้อมูลใน I2C bus จะกระทำในขณะที่ขา SCL เป็น High ในระหว่างการถ่ายทอดข้อมูล สายข้อมูล (SDA) ต้องรักษาข้อมูลไว้ไม่ให้เกิดการเปลี่ยนแปลงเด็ดขาด และสภาวะการเปลี่ยนแปลงข้อมูลจะกระทำในขณะที่ขา SCL เป็น Low เท่านั้น

4. ตัว Master จะส่งค่า Address ของอุปกรณ์ที่ต้องการจะติดต่อด้วยไปเรื่อยๆรวมทั้งหมด 7 บิต แล้วตามด้วยบิตที่ 8 ซึ่งจะเป็นคำสั่ง Mode กือเป็นการระบุว่าสั่งให้ Slave ทำอะไร ถ้าให้เขียน (Write) หรือ Output mode กือ ตัว Master ต้องการส่งข้อมูลไปที่ Slave จะให้สาย SDA เป็น Low และถ้าให้อ่าน (Read) หรือ Input mode กือให้ตัว Master รับค่าที่ส่งมาจาก Slave จะต้องให้สาย SDA เป็น High

5. ส่วนในบิตที่ 9 เป็นการติดต่อด้วยโดยจะต้องส่งสถานะรับรู้ Acknowledge (ACK) เป็นบิตที่ใช้บอกว่า Slave มีการตอบสนองต่อคำสั่งที่ได้รับมาแล้วหรือไม่ กือจะเป็นการตอบกลับจาก Slave ที่มี Address ตรงกับที่ Master ส่งไป ซึ่งถ้ามี Slave ที่ Address ตรงกับที่ระบุไว้ ตัว Slave จะตอบรับ (Acknowledge) โดยการดึงสัญญาณสาย SDA ลงเป็น Low กือพร้อมจะสื่อสารด้วย แต่ถ้าไม่มี Slave ที่ Address ตรงกับที่ Master ต้องการสื่อสารด้วย สายสัญญาณ SDA จะยังคงอยู่ที่ High กือไม่มีการตอบรับ (Not Acknowledge)

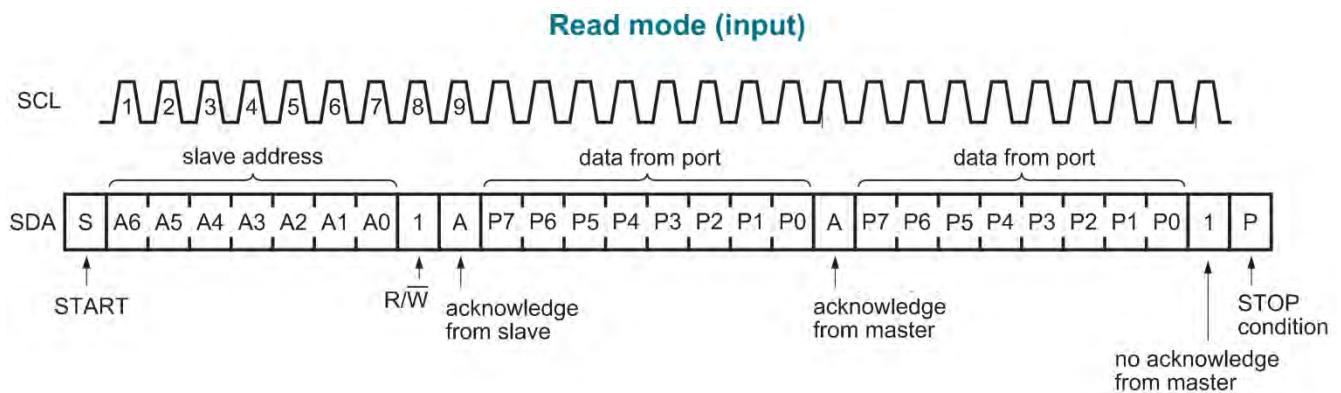
6. เมื่อ Slave มีสัญญาณตอบกลับมาเรียบร้อย กือ Acknowledge แล้ว จะเป็นช่วงเวลาที่สาย SCL ถูกดึงสัญญาณลง Low เป็นเวลาสั้นๆ และสาย SDA จะถูกปล่อยว่าง ก่อนที่ Slave ที่ติดต่อด้วยจะเริ่มส่งค่าข้อมูลนิพจน์มาที่สายสัญญาณ SDA

7. จากนั้นตัว Master จะส่งสัญญาณจากสาย SCL ออกไป เพื่ออ่านค่าข้อมูลเข้ามาจากสาย SDA ในช่วงจังหวะที่สาย SCL เป็น High และจะรับค่าข้อมูล (Data) ไปเรื่อยๆ จนครบข้อมูลชุดแรก 8 บิต ตามจังหวะของสายสัญญาณ SCL โดยในช่วงที่เกิดการอ่านข้อมูลค่าสภาวะของล็อกอิกที่เกิดขึ้นบนสาย SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสภาวะเป็น High ไม่เข่นนั้น แล้วข้อมูลที่ทำการถ่ายทอดจะเกิดความผิดเพี้ยนได้ และข้อมูลจะมีการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็น Low เท่านั้น

8. เมื่อรับข้อมูลครบ 8 บิตแล้ว ในกรณีข้อมูลที่ Slave ต้องการส่งยังมีข้อมูลชุดอื่นอีก Master จะให้สัญญาณ Acknowledge โดยดึงสัญญาณสาย SDA เป็น Low เพื่อบอกให้ Slave รู้ว่า Master จะรอข้อมูลอีก 8 บิตชุดต่อไป และจะเริ่มกลับไปทำงานต่ออีกเหมือนในข้อ 6

9. แต่ถ้ารับข้อมูลครบ 8 บิต และ Slave ไม่มีข้อมูลชุดอื่นที่จะส่งมาอีกแล้ว เมื่อถึงจังหวะการ Acknowledge ในบิตที่ 9 สายสัญญาณ SDA จะถูกปล่อยให้เป็น High

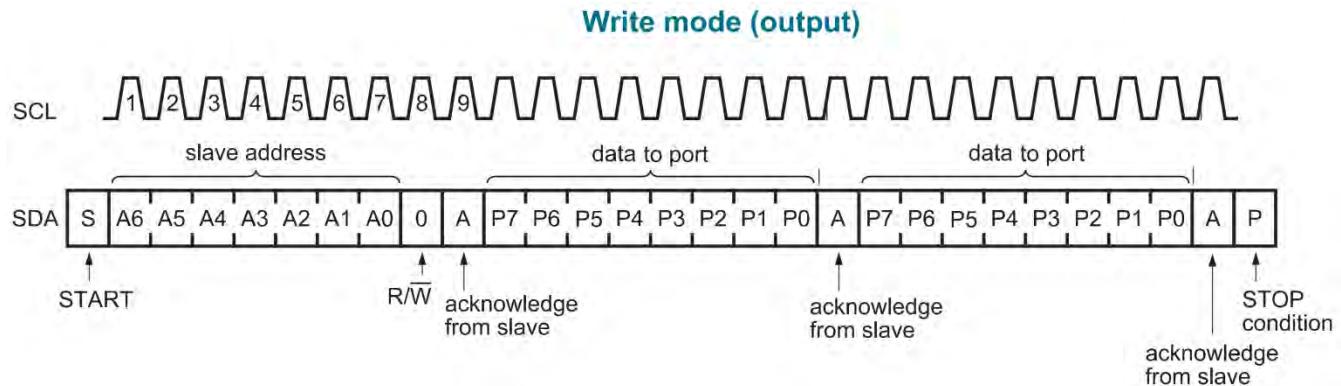
10. เมื่อรับข้อมูลครบหมดแล้ว Master ก็จะต้องสั่งให้หยุด (Stop) เป็นสถานะที่บอกให้อุปกรณ์รู้ว่าถึงสุดการใช้บัสในการรับส่งข้อมูลแล้ว โดยการส่งสัญญาณให้สาย SDA เปลี่ยนจาก Low เป็น High ในขณะที่สัญญาณสาย SCL เป็น High อยู่ ซึ่งจะได้ Timing Diagram ของการอ่านข้อมูลเข้ามาของ I2C แสดงดังรูป



ตัวอย่าง รหัสข้อมูลที่ใช้ในโหมดการอ่าน โดยข้อมูลที่ถูกส่งจาก Slave จะเป็นอักษรตัวหนา

<S> <slave address + read> <ACK> <data in> <ACK> <data in> <ACK> <data in> <NACK> <P>

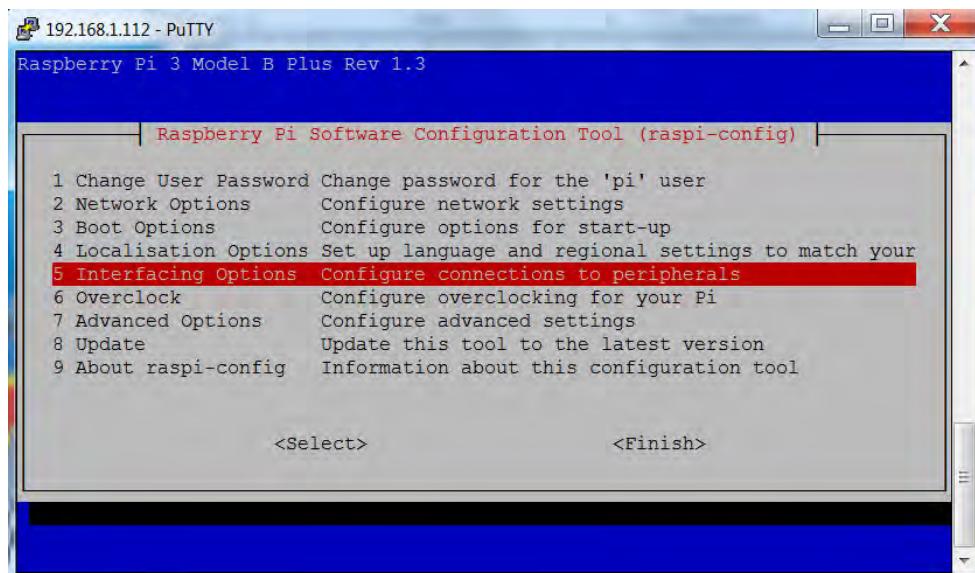
ส่วน Timing Diagram ของ I2C ในการเขียนข้อมูล ซึ่งจะส่งออกจาก Master ไปที่ Slave แสดงได้ดังรูป



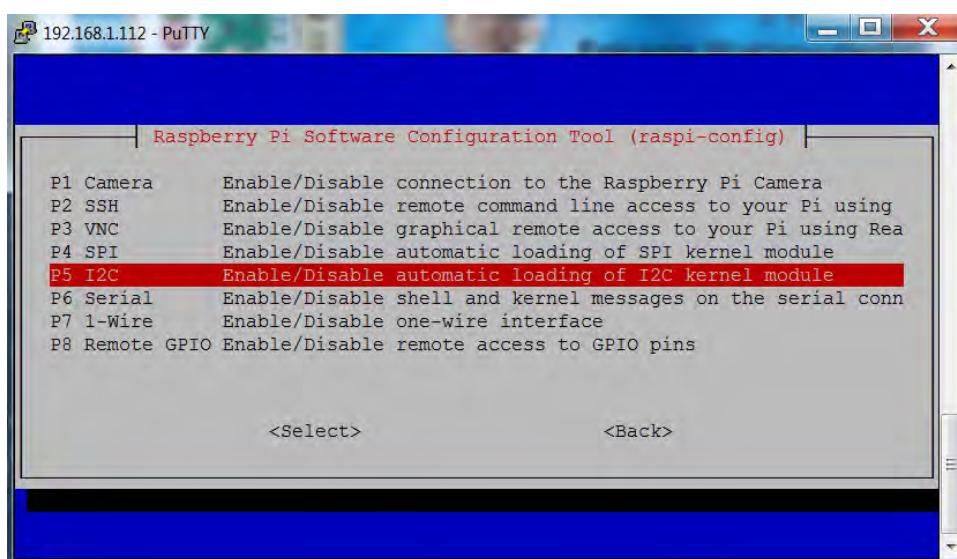
ตัวอย่าง รหัสข้อมูลที่ใช้ในโหมดการเขียน โดยข้อมูลที่ถูกส่งจาก Slave จะเป็นอักษรตัวหนา

<S> <slave address + write> <ACK> <data out> <ACK> <data out> <ACK> <data out> <ACK> <P>

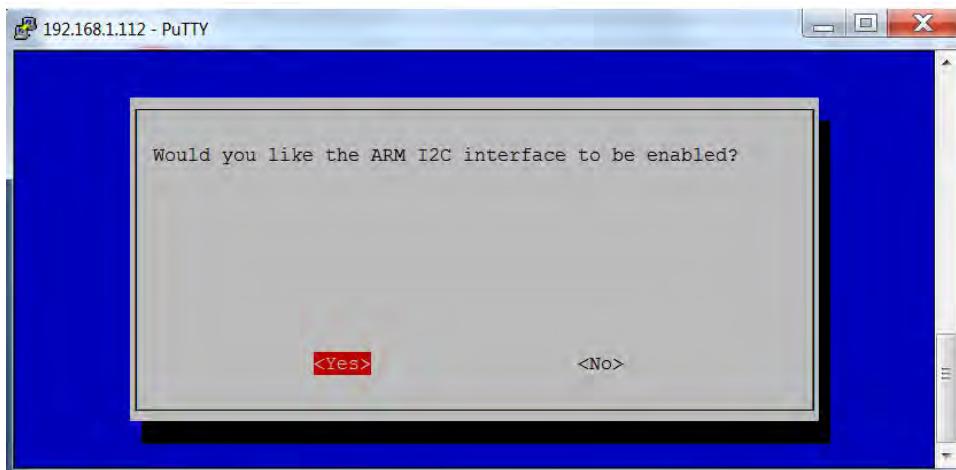
การทดลอง I2C (Inter-Integrated Circuit) ซึ่งเป็นโปรโตคอลมาตราฐานที่ใช้ในการติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์กับอุปกรณ์ภายนอกได้พร้อมกันหลายตัว โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้นคือ SDA (Serial Line Data) และ SCL (Serial Clock) การใช้งาน I2C บนบอร์ด Raspberry Pi จะต้องทำการเปิดการใช้งาน I2C ก่อน โดยสามารถทำได้ 2 แบบ แบบแรกใช้ GUI (Graphic User Interface) บนหน้า Desktop ของ Raspberry Pi ให้ไปที่ตัวเลือก Preferences แล้วเลือก Raspberry Pi Configuration หลังจากนั้นให้ไปที่ interfaces แล้วคลิก Enable ที่ตัวเลือก I2C จากนั้นกด OK หรืออาจจะใช้แบบที่สองโดยใช้โปรแกรม PuTTY เป็นเทอร์มินัล ด้วยการใช้คำสั่ง sudo raspi-config เพื่อเปิด Raspberry Pi Software Configuration Tool จากนั้นให้เลือก Interfacing Options ดังนี้



เลือก I2C



จะเห็นคำถามว่าต้องการให้ ARM I2C interface เปิดใช้งานหรือไม่ โดยในตัวเลือกจะเป็น No ถ้าข้างไม่เคยเปิดใช้ ให้เปลี่ยนตัวเลือกเป็น Yes และกด OK



การเขียนโปรแกรมด้วยภาษาไพทอนจะต้องติดตั้งไลบรารี package python-smbus i2c-tools ก่อน โดยให้พิมพ์คำสั่งต่อไปนี้ในเทอร์มินัล

```
sudo apt-get update
sudo apt-get install -y python-smbus i2c-tools
sudo reboot
```

หลังจากวินาทีระบบแล้วให้พิมพ์คำสั่งต่อไปนี้เพื่อตรวจสอบการติดตั้งซอฟต์แวร์:

```
lsmod | grep i2c_
```

จะได้ผลลัพธ์ในรายการมี `i2c_bcm2835` และ `i2c_dev` ซึ่งหมายความว่าไลบรารีได้รับการติดตั้งสำเร็จแล้ว

```
pi@raspberrypi:~$ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0 MB]
Fetched 13.0 MB in 33s (400 kB/s)
Reading package lists... Done
pi@raspberrypi:~$ sudo apt-get install -y python-smbus i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version (4.1-1).
python-smbus is already the newest version (4.1-1).
The following package was automatically installed and is no longer required:
  rpi-eeprom-images
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 25 not upgraded.
pi@raspberrypi:~$ lsmod | grep i2c_
i2c_bcm2835           16384  0
i2c_dev                16384  0
pi@raspberrypi:~$
```

ขั้นตอนต่อไปจะต้องเป็นการตรวจหาอาร์ดิวาร์ของอุปกรณ์ I2C ที่นำมาต่อกับบอร์ด Raspberry Pi โดยจะต้องต่อขาของอุปกรณ์ I2C ให้ขา SDA ต่อกับขา3(GPIO2) และขา SCL ต่อกับขา5(GPIO3) ของ Raspberry Pi แล้วให้รีบกใช้คำสั่งต่อไปนี้ในเทอร์มินัล

```
i2cdetect -y 1
```

ตามตัวอย่างถ้าพบอุปกรณ์ I2C จะแสดงตัวเลขขึ้นมาในที่นี่มีอุปกรณ์ I2C ต่อໄว์ 2 อุปกรณ์ กือที่ Address 27 และ Address 3f แต่ถ้าหากเห็นเฉพาะ -- ในรายการโดยไม่มีตัวเลขแสดงว่าการเชื่อมต่อของ "ไม่ถูกต้องหรือซอฟต์แวร์ไม่ได้รับการติดต่ออย่างถูกต้อง"

```
pi@raspberrypi:~$ i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: --- - - - - - - - - - - - - - - - - - - - -
10: --- - - - - - - - - - - - - - - - - - - - -
20: --- - - - - - - - 27 - - - - - - - - - -
30: --- - - - - - - - - - - - - - - - - - - 3f
40: --- - - - - - - - - - - - - - - - - - - -
50: --- - - - - - - - - - - - - - - - - - - -
60: --- - - - - - - - - - - - - - - - - - - -
70: --- - - - - - - - - - - - - - - - - - - -
pi@raspberrypi:~$
```

### คำสั่งการวนทำซ้ำด้วย for

```
for Condition:  
    Statements  
....  
....
```

### คำสั่งการจัดการทดสอบข้อผิดพลาด error

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

13. จากวิธีการตรวจหาชาร์ดแวร์ของอุปกรณ์ I2C ด้วยการใช้คำสั่ง SSH จากหน้าต่างเทอร์มินัลนั้น เราจะทำการเขียนโปรแกรมเป็นภาษา Python เพื่อตรวจหาชาร์ดแวร์ของอุปกรณ์ I2C โดยจะใช้คำสั่งที่ได้จาก Library ที่ชื่อว่า smbus ซึ่งจะนำการลีดอสารแบบ I2C มาใช้ แล้วใช้คำสั่งใน smbus จัดการรับส่งค่าให้ เพื่อเป็นการหา Address ที่ใช้ในโมดูลของอุปกรณ์ที่นำมาเชื่อมต่อนั้น เราสามารถเขียนโปรแกรมเพื่อทดสอบหา Address ของอุปกรณ์ได้ดังนี้

```
1. import smbus  
2. bus_number = 1 # 1 indicates /dev/i2c-1  
3. bus = smbus.SMBus(bus_number)  
4. for device in range(128):  
5.     try:  
6.         bus.read_byte(device)  
7.         print(hex(device))  
8.     except: # exception if read_byte fails  
9.         pass
```

14. การทดลองให้ต่อ I2C Board Module For LCD เข้ากับบอร์ด Raspberry Pi ผ่านการเชื่อมต่อทางพอร์ต I2C แล้ว  
ให้ทดลองป้อนโปรแกรมเพื่อตรวจหาอุปกรณ์ต่างๆ ที่ต่ออยู่กับพอร์ต I2C และบันทึกค่าที่ได้
- .....  
.....  
.....  
.....

15. ให้ทดลองแก้ไขโปรแกรมข้อ 13 โดยให้เปลี่ยนคำสั่งจาก pass ไปเป็น print("[“hex(device),”]”)  
เสร็จแล้วให้ทำการ RUN โปรแกรมและให้อธิบายผลลัพธ์ที่ได้
- .....  
.....  
.....  
.....

16. จากโปรแกรมในข้อ 15 ให้อธิบายความหมายและการทำงานของโปรแกรมในแต่ละบรรทัด

- 1.....
- 2.....
- 3.....
- 4.....
- 5.....
- 6.....
- 7.....
- 8.....
- 9.....

17. ให้ต่อวงจรโดยใช้ LED จำนวน 4 หลอด แล้วต่ออนุกรมกับตัวต้านทาน  $300\ \Omega$  เข้าที่ขา P4 P5 P6 P7 ของ I2C Board

18. ให้แก้ไขโปรแกรมข้อ 13 เพิ่มโปรแกรมการส่งค่าไปที่พอร์ต I2C ที่ตรวจสอบ โดยค่าที่ส่งไปมีค่าตั้งแต่ 0 ถึง 255  
และจะพิมพ์ค่าที่ส่งออกไปที่จอ Monitor ด้วยดังนี้

```
for data in range (256):
    print("OUT ",bin(data))
    bus.write_byte(addr, data)
    time.sleep(0.1)
```

19. ให้อธิบายผลลัพธ์ที่ได้ของ LED จากการส่งข้อมูลผ่านทางพอร์ต I2C
- .....  
.....

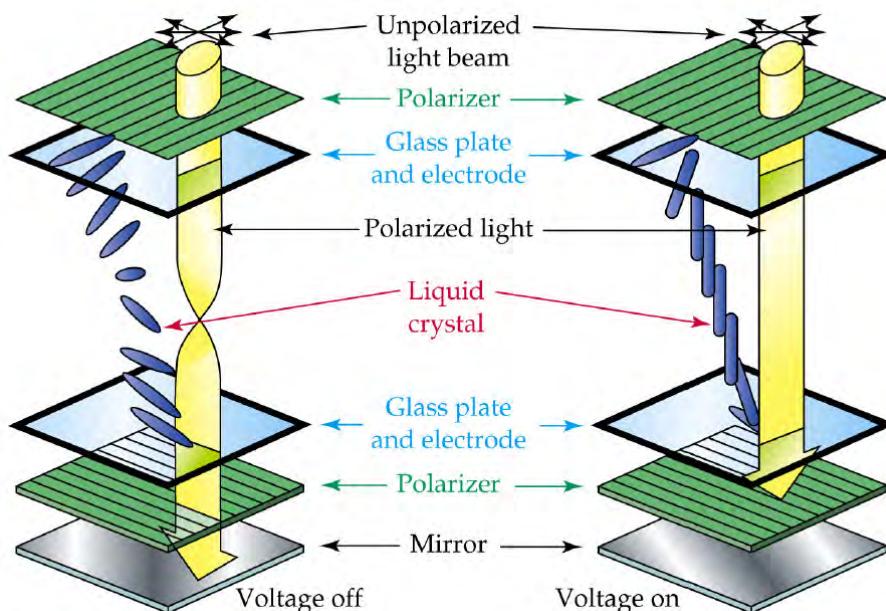
20. ให้แก้ไขโปรแกรมข้อ 18 โดยเพิ่มโปรแกรมการอ่านค่าจากพอร์ต I2C กลับมา หลังจากส่งข้อมูลไปในแต่ละค่า และพิมพ์ผลลัพธ์ของค่าที่รับมาไปออกจอ Monitor ดังนี้

```
dataIN = bus.read_byte(addr)
print(" IN ",bin(dataIN))
```

21. ให้อธิบายผลลัพธ์ที่ได้จากค่าที่รับเข้ามาผ่านทางพอร์ต I2C แล้วแสดงออกทาง Monitor ว่าตรงกับค่าที่ส่งออกไป หรือไม่ เพราะเหตุใด
- .....  
.....  
.....

22. ให้แก้ไขโปรแกรมในข้อที่ 20 โดยเปลี่ยนโปรแกรมการส่งค่าไปที่พอร์ต I2C จากค่าที่ส่งไปทั้งหมด 256 ค่า ไป เป็นค่าที่ส่งให้ Stepping Motor จำนวนทั้งหมด 4 ค่า

การแสดงผลตัวอักษรโดยใช้ Liquid Crystal Display หรือจอภาพ LCD หลักการทำงานจะมีการส่งแสงไฟที่จากด้านหลังของตัวอักษร LED Backlight โดยที่แสงจะเป็นคลื่นแม่เหล็กไฟฟ้า ประกอบด้วยสนามแม่เหล็กและไฟฟ้าที่สั่นตั้งๆ กันอยู่บนระบบที่ตั้งๆ กันทิศทางการแผ่ของคลื่น และเป็นคลื่นตามขวางโดยใช้ปรากฏการณ์โพลาไรเซชัน แสงโพลาไรซ์สามารถผลิตได้โดยการส่งผ่านโพลาไรเซอร์ แสงจะถูกส่งออกมาทางด้านหน้าผ่านทางของเหลวที่เป็นกลาง แสงโพลาไรซ์ที่ประกอบด้วยสนามไฟฟ้า ซึ่งสั่นในแนวใดแนวหนึ่ง จะผ่านลักษณะของ LCD ที่ควบคุมด้วยแรงดันไฟฟ้าเข้าไประหว่างผลึกของสารเหลวซึ่งมีคุณสมบัติการบิดแกนโพลาไรซ์ของแสง ทำให้ดังนีการหักเหของแสงไม่เท่ากันเกิดการเคลื่อนที่ของแสงด้วยความเร็วไม่เท่ากันในแต่ละทิศทางของผลึกเกิดการหักเหหมุนบิดตัวให้อยู่ในแนวตรงกันข้าม เป็นการเปลี่ยนโพลาไรซ์ของแสง เมื่อผ่านตัวกรองโพลาไรซ์ที่จะทำให้คลื่นแสงของโพลาไรซ์เพียงแนวเดียวผ่านได้ที่เป็นโพลาไรเซอร์แนวตรงกันเข้า แนวเดิงกันแนวนอน จึงทำให้แสงสามารถลดลงของความสว่างเป็นสีขาว และถ้าควบคุมแรงดันไฟฟ้าไม่ให้มีการบิดแกนโพลาไรซ์แสงจะทะลุผ่านออกมายังไฉไลจึงเป็นสีดำ



จอแสดงผล LCD ที่ใช้ในการทดลองจะมีอิเล็กโทรดที่บังคับการทำงานของพนักเหลวนี้ มาประกอบกันเป็นเซลล์เด็กๆ ที่มีความละเอียดในรูปแบบ Dot Matrix ทำให้สามารถควบคุมสร้างให้เป็นตัวอักษรได้ โดยมีขนาดของตัวอักษรเป็นแบบ Matrix ประกอบด้วยความกว้าง 5 pixel กับสูง 8 pixel มีจำนวนตัวอักษรรวม 16 ตัว และมีจำนวนบรรทัดทั้งหมด 2 แถวภายในวงจรที่ใช้ควบคุมจะใช้ไอซี HD44780 เป็นไอซีที่ทำหน้าที่ Dot Matrix Liquid Crystal Display Controller/Driver ถูกพัฒนาโดยบริษัท Hitachi เพื่อใช้ในการควบคุมการแสดงผลตัวอักษรแบบ Dot Matrix บนจอ LCD

โดยที่ LCD จะมีการแสดงข้อความเป็นแบบขาวดำ โดยมากมักจะถูกนำมาใช้ในเครื่องถ่ายเอกสาร เครื่องรับโทรศัพท์ เครื่องแท็บเล็ต อุปกรณ์ทดสอบทางอุตสาหกรรม อุปกรณ์เครื่องข่าย และอุปกรณ์จัดเก็บข้อมูล

ชุดของตัวอักษรในไอซี HD44780 จะมีใช้โดยทั่วไปได้ 2 แบบ คือ แบบเอชีและแบบยูโรป โดยที่เป็นชุดตัวอักษรหนึ่งในสองชุดที่แตกต่างกัน ดังนี้

Lower 4 Bits RAM	Upper 4 Bits RAM	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	(1)	0	0	P	^	P	-	タ	ミ	ア	ル	ー	タ	ミ	ア	ル	
xxxx0001	(2)	!	1	A	Q	a	!	ア	チ	6	あ	ー	ア	チ	6	あ	
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	」	」	イ	ツ	」	イ	
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	エ	»	」	ウ	テ	エ	
xxxx0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト	ム	、	エ	ト	ト	
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	バ	セ	・	オ	ナ	バ	
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	ヲ	カ	ニ	ヨ	
xxxx0111	(8)	'	7	G	W	g	w	ア	キ	ア	ラ	9	ア	キ	ア	ラ	
xxxx1000	(1)	(`	8	H	X	h	x	イ	ワ	キ	リ	ト	イ	ワ	キ	リ	
xxxx1001	(2)	)	9	I	Y	i	y	ガ	リ	リ	リ	ー	ガ	リ	リ	ー	
xxxx1010	(3)	*	:	J	Z	j	z	エ	コ	ゴ	レ	じ	エ	コ	ゴ	レ	
xxxx1011	(4)	+	;	K	C	k	c	オ	サ	ヒ	ロ	×	オ	サ	ヒ	ロ	
xxxx1100	(5)	,	<	L	¥	1	l	ア	シ	フ	ワ	¶	ア	シ	フ	ワ	
xxxx1101	(6)	-	=	M	J	m	j	ユ	ス	ヘ	ン	÷	ユ	ス	ヘ	ン	
xxxx1110	(7)	.	>	N	^	n	→	ミ	エ	ホ	ー	ト	ミ	エ	ホ	ー	
xxxx1111	(8)	/	?	O	_	o	े	シ	ウ	カ	■	ঠ	シ	ウ	カ	■	

Lower 4 Bits RAM	Upper 4 Bits RAM	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	(1)	0	0	P	^	P	-	タ	ミ	ア	ル	ー	タ	ミ	ア	ル	
xxxx0001	(2)	!	1	A	Q	a	!	ア	チ	6	あ	ー	ア	チ	6	あ	
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	」	」	イ	ツ	」	イ	
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	エ	»	」	ウ	テ	エ	
xxxx0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト	ム	、	エ	ト	ト	
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	バ	セ	・	オ	ナ	バ	
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	ヲ	カ	ニ	ヨ	
xxxx0111	(8)	'	7	G	W	g	w	ア	キ	ア	ラ	9	ア	キ	ア	ラ	
xxxx1000	(1)	†	8	H	X	h	x	イ	ワ	キ	リ	ト	イ	ワ	キ	リ	
xxxx1001	(2)	‡	9	I	Y	i	y	ガ	リ	リ	リ	ー	ガ	リ	リ	ー	
xxxx1010	(3)	→	*	J	Z	j	z	エ	コ	ゴ	レ	じ	エ	コ	ゴ	レ	
xxxx1011	(4)	←	+	K	C	k	c	オ	サ	ヒ	ロ	×	オ	サ	ヒ	ロ	
xxxx1100	(5)	≤	,	L	¥	1	l	ア	シ	フ	ワ	¶	ア	シ	フ	ワ	
xxxx1101	(6)	≥	-	=	M	J	m	ユ	ス	ヘ	ン	÷	ユ	ス	ヘ	ン	
xxxx1110	(7)	▲	.	>	N	^	n	ミ	エ	ホ	ー	ト	ミ	エ	ホ	ー	
xxxx1111	(8)	▼	/	?	O	_	o	シ	ウ	カ	■	ঠ	シ	ウ	カ	■	

HD44780 สามารถนำไปใช้ควบคุมจอ LCD โดยมีโหมดการเข้ามือต่อได้ 2 แบบคือ โหมด 4 Bit (DB4 - DB7) ใช้สายทึ้งหนึ่ง 8 เส้น และ โหมด 8 Bit (DB0 - DB7) ใช้สายทึ้งหนึ่ง 12 เส้น ไม่ต้องอะแดปเตอร์ของหน้าจอLCD ที่ใช้หัวไปจะนิยมแบบการเข้ามือต่อโหมด 4 บิต การเข้ามือต่อโหมดแบบ 4 บิตนี้จะต้องใช้ส่งข้อมูลไปทึ้งหนึ่ง 2 ครั้ง เพื่อให้ได้ข้อมูลครบ 8 บิต โดยการส่ง 4 บิตแรกที่มีนัยสำคัญสูงสุด(MSB)ไปก่อน แล้วตามด้วย 4 บิตหลังที่มีนัยสำคัญต่ำสุด(LSB) ซึ่งการที่ต้องส่งข้อมูลถึง 2 ครั้งนั้นเวลาที่ใช้ไม่ได้เป็นอุปสรรคเนื่องจากความเร็วในการส่งข้อมูลก็ยังมากจนสายตาของเรามิ่งสามารถแยกออกได้ ส่วนการส่งงานให้ไมค์ลุด LCD ทำงานนี้จะต้องเขียน Software ควบคุมจากไมโครคอนโทรลเลอร์ไปสร้างสัญญาณ Timing เพื่อควบคุมไอซี HD44780 ให้ทำงาน ดังนั้นนอกจากบัสข้อมูล 4 เส้นแล้วจำเป็นต้องมีสายควบคุมเพิ่มเข้ามาได้แก่

RS (Register Select) ขาที่ทำหน้าที่เพื่อเลือกว่าจะถ่ายโอนข้อมูล หรือ ส่งคำสั่งระหว่างไมโครคอนโทรลเลอร์กับไมค์ลุด LCD ถ้าขาที่เป็น High ก็คือ Data register จะเป็นการอ่านหรือเขียนข้อมูลของไปต์ที่ตำแหน่งปัจจุบันของเครอร์เซอร์ใน LCD ที่เวลา\_n และถ้าขาที่เป็น Low จะมี 2 โหมดคือ 1. Instruction Register สำหรับการเขียน จะเป็นการส่งคำสั่งไปยัง LCD , 2. Read busy flag (DB7) and address counter (DB0 to DB6) สำหรับการอ่าน จะเป็นการอ่านสถานะการดำเนินการของคำสั่ง สุดท้ายกลับมา (ไม่ว่าจะเสร็จสมบูรณ์หรือไม่ก็ตาม)

R/W (Read/Write) ขาที่ทำหน้าที่กำหนดพิศทางของข้อมูลว่าจะเป็นการอ่านหรือเขียนข้อมูล แต่จะไม่สามารถอ่านสถานะข้อมูลจากแสดงผลได้

E (Enable) ก็คือ Starts data read/write. ขาที่ใช้เพื่อเริ่มการอ่านหรือเขียนข้อมูลภายใน LCD โดยการกำหนดให้ขาที่เป็น High ก่อนที่จะส่งข้อมูลหรือคำสั่งใดๆ หลังจากให้คีเลียร์และส่งอาร์กิวเมนต์ไปเรียบร้อยแล้วต้องให้ขาที่เป็น Low

#### HD44780 มีรีจิสเตอร์อยู่ 2 ตัว ก็คือ

Instruction Register (IR) มีหน้าที่เก็บคำสั่ง เช่น Clear display, Cursor shift , ข้อมูลของตำแหน่งที่ Display ใน Display Data RAM (DDRAM) และ Character Generator RAM (CGRAM) โดยที่ IR จะถูกเปลี่ยนจากไมโครคอนโทรลเลอร์

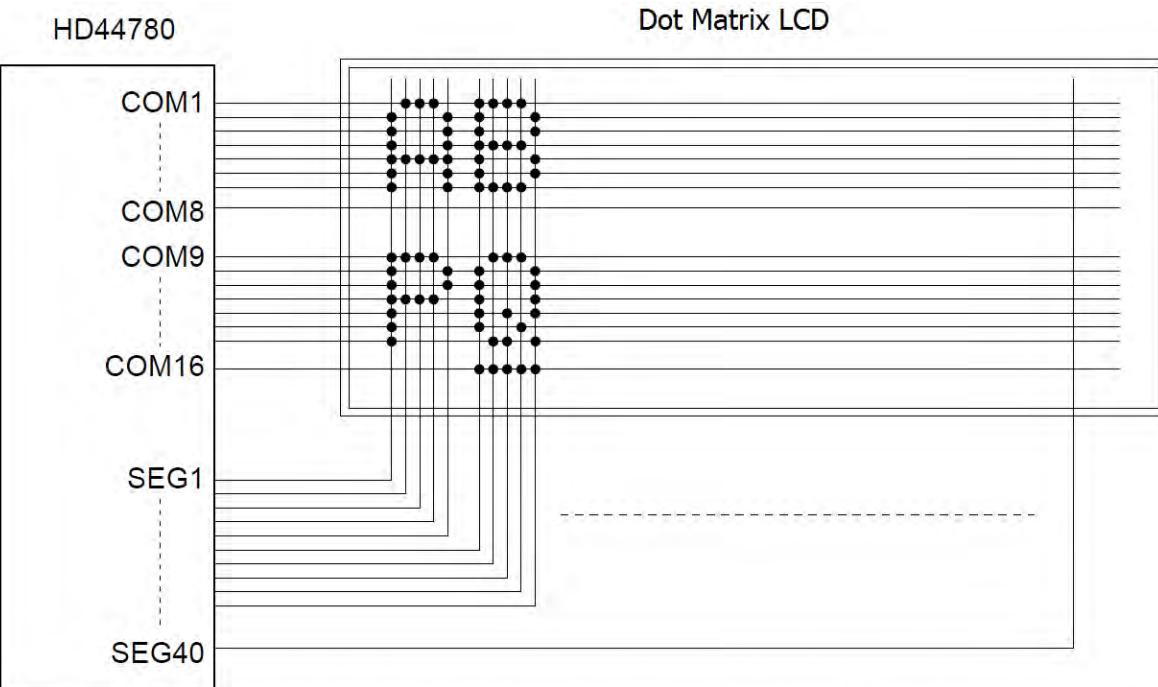
Data Register (DR) ใช้เก็บข้อมูลชี้ว่าควรที่จะเขียนลงใน Data Display RAM (DDRAM) หรือ Character Generator RAM (CGRAM) และเก็บข้อมูลชี้ว่าควรที่อ่านจาก DDRAM หรือ CGRAM

RS	R/W	การทำงาน
0	0	เขียนคำสั่งลงใน IR
0	1	อ่าน Busy Flag (DB7) และ Address Counter (DB0 - DB6)
1	0	เขียนข้อมูลลงใน DD RAM หรือ CG RAM
1	1	อ่านข้อมูลจาก DD RAM หรือ CG RAM

ชุดคำสั่ง (Instruction set) ที่เพิ่มความคุ้มการทำงานของ Dot Matrix LCD HD44780 มีดังนี้

Instruction	Code											Description
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1		Clears entire display and sets DDRAM address 0 in address counter.
Return home	0	0	0	0	0	0	0	0	1	—		Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.
Entry mode set	0	0	0	0	0	0	0	1	I/D	S		Sets cursor move direction and specifies display shift. These operations are performed during data write and read.
Display on/off control	0	0	0	0	0	0	1	D	C	B		Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—		Moves cursor and shifts display without changing DDRAM contents.
Function set	0	0	0	0	1	DL	N	F	—	—		Sets interface data length (DL), number of display lines (N), and character font (F).
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG		Sets CGRAM address. CGRAM data is sent and received after this setting.
Set DDRAM address	0	0	1	ADD		Sets DDRAM address. DDRAM data is sent and received after this setting.						
Read busy flag & address	0	1	BF	AC		Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.						
Write data to CG or DDRAM	1	0	Write data									Writes data into DDRAM or CGRAM.
Read data from CG or DDRAM	1	1	Read data									Reads data from DDRAM or CGRAM.
I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 × 10 dots, F = 0: 5 × 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable												DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses

เมื่อจ่ายไฟเข้ากับจอแสดงผล LCD และจะอยู่ในสถานะพร้อมทำงาน ซึ่งแสดงโดยบรรทัดทั้งหมดเป็นสีดำ หลังจากเริ่มดันระบบสามารถส่งคำสั่งไปยังจอแสดงผลเพื่อข้ามคอร์เซอร์ไปยังตำแหน่งที่ต้องการ สำหรับจอแสดงผล LED ขนาดตัวอักษร  $2 \times 16$  บรรทัดแรกอักษรจะมีแอคเดรஸอยู่ที่  $0x80-0x8F$  ในขณะที่บรรทัดที่สองแอคเดรஸจะอยู่ที่  $0xC0-0xCF$  และถ้า LCD ที่ใช้นั้นมีจำนวนบรรทัดมากกว่าสอง จะทำให้มีบรรทัดที่สามแอคเดรஸอยู่ที่  $0x90-0x9F$  และบรรทัดที่สี่แอคเดรஸจะอยู่ที่  $0xD0-0xDF$  ตัวอักษรจะวางในลำดับการต่อ Dot Matrix ขนาด  $5 \times 8$  แปดตัวอักษร สองบรรทัด แสดงได้ดังนี้



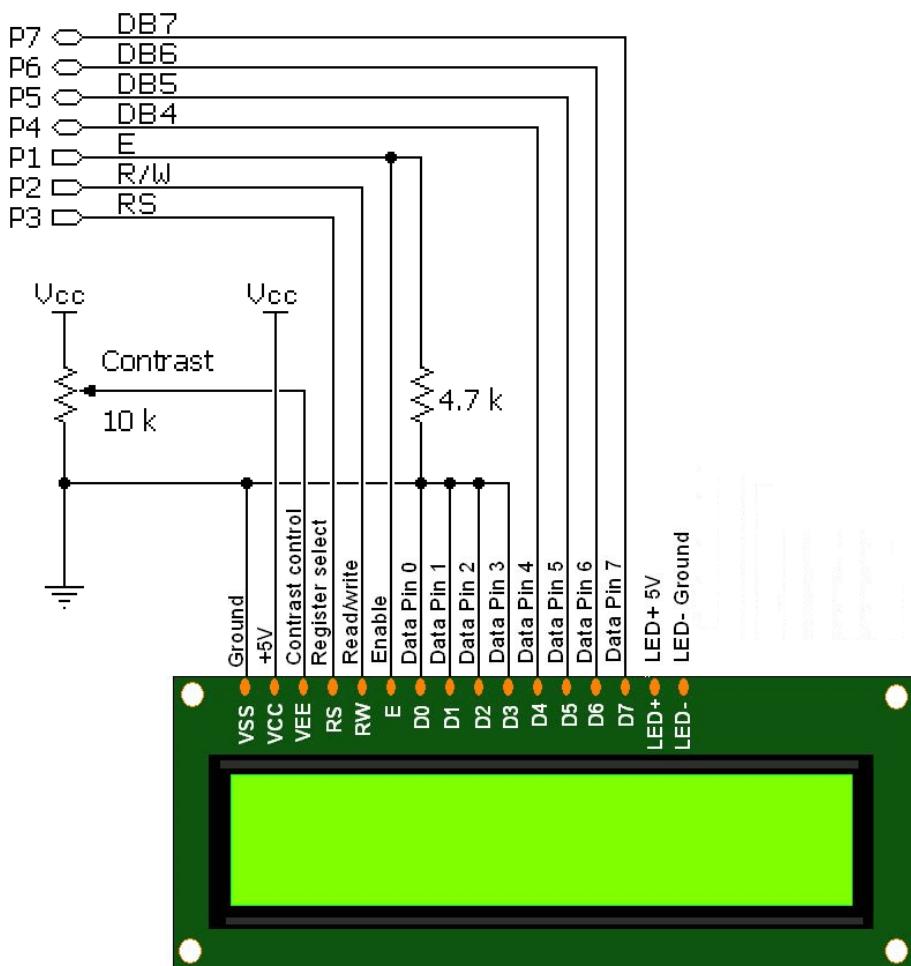
Example of a  $5 \times 8$  dot, 8-character  $\times$  2-line display (1/5 bias, 1/16 duty cycle)

หน้าจอ LCD ที่สามารถนำมาใช้งานร่วมกันกับไอซี HD44780 นี้ได้ผลิตขึ้นมาในรูปแบบมาตรฐานต่างๆ ขนาดทั่วไปคือ มีตั้งแต่หนึ่งถึงแปดตัวอักษร ( $8 \times 1$ ) ส่องแคลวสิบหกตัวอักษร ( $16 \times 2$ ) นอกเหนือนี้ยังมีรูปแบบขนาดที่กำหนดให้กับข้อความอีกประกอบด้วยอักษร 20, 32, 40 หรือ 80 ตัว และมีจำนวนแอกาตั้งแต่ 1, 2, 4 หรือ 8 บรรทัด การผลิตที่มีหน้าจอขนาดใหญ่ขึ้นอีกนั้น โดยทั่วไปมีจำนวนตัวอักษร  $40 \times 4$  ซึ่งต้องใช้คอนโทรลเลอร์ HD44780 ที่กำหนดแอคเดรஸแยกกันสองตัว เนื่องจากไอซี HD44780 ตัวเดียวสามารถระบุตัวอักษรได้สูงสุดไม่เกิน 80 ตัว ส่วน Backlight ที่ใช้ในการให้ความสว่างเพื่อให้พื้นหลังเรืองแสง ทำให้เห็นตัวอักษรของหน้าจอ LCD ได้นั้น อาจจะเป็นหลอด Fluorescent หรือ LED ก็ได้

หน้าจอแสดงผล LCD ที่ใช้ HD44780 ส่วนใหญ่จะใช้ขั้วต่ออินเทอร์เฟชแบบ 16 ขาติดต่อกัน ซึ่งใช้หมุดสำหรับการเชื่อมต่อระยะห่างระหว่างกัน 0.1 นิ้ว (2.54 มม.) โดยมีแต่ละขาที่ต่อเป็นดังนี้

1. Ground (0V)
2. Vcc Power supply ต่อ กับไฟเลี้ยง 2.7V ถึง 5.5V
3. Contrast adjustment (VO) เป็นอินพุตแบบอะนาล็อก ผู้ใช้จะต้องควบคุมแรงดันไฟฟ้าที่ข้างนี้เพื่อเพิ่มประสิทธิภาพในการมองเห็นของจอแสดงผลที่แยกต่างกัน ถ้าปรับไม่ถูกต้องจะไม่เห็นตัวอักษรที่หน้าจอแสดงผล ซึ่งขานี้จะเชื่อมต่อกับแรงดันไฟฟ้าที่ได้จากการแบ่งแรงดันไฟฟ้า (Voltage divider) ของตัวต้านทานปรับค่าได้  $10k$
4. Register Select (RS) ถ้า RS=0 จะเป็นประเภทคำสั่ง (Command) และ RS=1 จะเป็นข้อมูล (Data)

5. Read/Write (R/W) ถ้า R/W=0 จะเป็น Write และ R/W=1 จะเป็น Read
6. Enable (E) เป็นสเมื่อนสัญญาณ Clock ใช้ข้อมูลในการทริก
7. DB0 (ไม่ได้ใช้ในการทำงานแบบ 4 บิต)
8. DB1 (ไม่ได้ใช้ในการทำงานแบบ 4 บิต)
9. DB2 (ไม่ได้ใช้ในการทำงานแบบ 4 บิต)
10. DB3 (ไม่ได้ใช้ในการทำงานแบบ 4 บิต)
11. DB4
12. DB5
13. DB6
14. DB7
15. Backlight Anode (+) ต่อ กับ แรงดันไฟ บวก
16. Backlight Cathode (-) ต่อ กับ กราวด์



การเขียนโปรแกรม Character Pattern การกำหนดรูปแบบตัวอักษรจะต้องมีความสอดคล้องระหว่าง Address และ Data ที่ใช้ในการเขียน โปรแกรมรูปแบบอักษรจะถูกสร้างแล้วเก็บอยู่ใน EPROM ROM HD44780U สามารถสร้างรูปแบบตัวอักษรขนาด 5x8 จุด และ 5x10 จุด โดยมีรูปแบบตัวอักษรทั้งหมด 240 แบบ

ค่า Address กับค่า Data ของ EPROM และรูปแบบ Pattern ของตัวอักษรจะมีความสอดคล้องกัน ในการสร้างตัวอักษรขนาด 5x8 ตัวอย่างอักษร ๖ สามารถแสดงได้ดังในตาราง

โดยที่ค่าตามในตารางจะเป็นดังนี้

- EPROM แอดเดรส A11 ถึง A4 เป็นค่าตำแหน่งของรหัสตัวอักษร
  - EPROM แอดเดรส A3 ถึง A0 เป็นค่าตำแหน่งแต่ละແລະແກ梧ของ Pattern ในตัวอักษรนั้น
  - ข้อมูล EPROM O4 ถึง O0 เป็นค่าข้อมูลแต่ละແລະແກ梧ของ Pattern ตัวอักษรนั้น
  - ข้อมูล EPROM O5 ถึง O7 เป็นค่าข้อมูลของตัวอักษรกำหนดให้เป็น 0
  - ตำแหน่ง Pixel ของตัวอักษรที่ต้องการแสดงผลกำหนดให้เป็น 1
  - ค่าข้อมูลของ Pattern ดังແຕ່ແກວທี่ 9 เป็นต้นไปจะว่างกำหนดให้เป็น 0 สำหรับฟอนต์อักษรขนาด  $5 \times 8$  Pixel

การเขียนโปรแกรมเพื่อแสดงผลบนโมดูล Dot Matrix LCD จะใช้คำสั่งที่นำมาจาก I2C\_LCD\_driver ซึ่งเป็นโมดูลสำหรับอินเตอร์เฟสกับพอร์ต I2C ของ Dot Matrix Liquid Crystal Display HD44780 มาใช้ ปกติแล้วการเชื่อมต่อโมดูล LCD แบบ 4 Bit จะใช้สายจำนวนมาก ทำให้เปลือง Pin ที่อาจจะต้องนำไปใช้งานกับอุปกรณ์อื่นๆ แต่เนื่องจากโหมดการเชื่อมต่อแบบ 4 Bit สามารถแปลงสายสัญญาณให้เข้ากับโมดูลที่มีรองแคปเตอร์ I2C ได้ ทำให้ต่อสายสัญญาณได้ง่ายขึ้น ดังนั้นในการทดลองนี้จะได้นำมาโมดูล I2C มาประยุกต์ใช้งานกับโมดูล LCD เพื่อที่จะลดจำนวน Pin ที่ต้องใช้งานจำนวนมากลงให้เหลือเพียง 2 สาย คือ SCL และ SDA เท่านั้น และยังสามารถติดต่อกับอุปกรณ์อื่นที่ใช้ I2C ร่วมกันได้บนสายสัญญาณเดียวกันได้อีกด้วย

23. ให้เชื่อมต่อโมดูล Dot Matrix LCD ที่มี 16 Pin ต่อเข้ากับโมดูล I2C Board แล้วนำโมดูล I2C Board ไปเชื่อมต่อกับพอร์ต I2C ของบอร์ด Raspberry Pi โดยต่อขาของอุปกรณ์ I2C ให้ขา SDA ต่อ กับขา 13(GPIO2) และขา SCL ต่อ กับขา 15(GPIO3) ของ Raspberry Pi ในกรณีที่ต้องใช้อุปกรณ์ I2C หลายตัวพร้อมกันจะต้องเปลี่ยนค่า Address ของ

อุปกรณ์แต่ละตัวไม่ให้ซ้ำกันด้วย โดยใช้โปรแกรมจาก การทดลองที่ผ่านมาเพื่อตรวจสอบ Address ของอุปกรณ์ต่างๆ ที่ต่ออยู่กับพอร์ต I2C

24. ให้ Copy ไฟล์โมดูล I2C\_LCD\_driver.py ลงในไคลเร็คทอรีและทำการเขียนโปรแกรมที่ทำหน้าที่สั่งงานให้ Dot Matrix LCD แสดงผลลัพธ์ที่หน้าจอเป็นคำว่า LCD I2C จากนั้นทำการ Run โปรแกรม

```
import I2C_LCD_driver
from time import *

mylcd = I2C_LCD_driver.lcd()

mylcd.lcd_display_string("LCD I2C", 1)
sleep(1)
```

25. ให้แก้ไขโปรแกรมในข้อ 24 โดยเพิ่มโปรแกรมสั่งงานให้ LCD นี้แสดงผลลัพธ์เพิ่มเติมที่หน้าจอในบรรทัดที่ 2 ต่อท้ายโปรแกรมเดิมดังนี้

```
mylcd.lcd_display_string(" ASCII Code Test", 2)
```

26. ให้แก้ไขโปรแกรมในข้อ 25 โดยเพิ่มโปรแกรมสั่ง Clear หน้าจอ ต่อท้ายโปรแกรมเดิม แล้วให้อธิบายผลที่ได้

```
mylcd.lcd_clear()
```

.....  
.....

27. จากโปรแกรมในข้อ 26 เพื่อให้โปรแกรมทำงานได้ถูกต้องจะต้องเพิ่มคำสั่งอะไรลงคำสั่ง display หน้าจอ ให้ทำการทดลองและอธิบายผลที่ได้
- .....  
.....  
.....

28. ให้แก้ไขโปรแกรมในข้อ 27 โดยเพิ่มโปรแกรมแสดงชุดของตัวอักษรทั้งหมดภายใน Dot Matrix LCD HD44780 ตั้งแต่รหัส ASC II ตัวที่ 0 ถึง 255 โดยใช้จะต้องใช้คำสั่งเพื่อกำหนดหัวตำแหน่งของ Cursor ด้วยดังนี้

```
row = 1
col = 0
for i in range (256):
    mylcd.lcd_display_string_pos(chr(i), row, col)
    col = col + 1
    if col == 16:
        col = 0
        row = row + 1
        if row == 3:
            row = 1
    sleep(0.05) # 0.05 sec delay
```

29. จากข้อ 28 ให้ทดลองกำหนดตัวแปร font เป็น Array เพื่อสร้างตัวอักษรในรหัส ASC II ขึ้นมาใหม่ตั้งแต่ตัวอักษรตัวที่ 0 ถึง 5 โดยที่อักษรแต่ละตัวจะเป็นขนาด 5x8 ดังนี้

```
font = [
    [0x00, 0x00, 0x03, 0x04, 0x08, 0x19, 0x11, 0x10], # Char 0 - Upper-left
```

```
[0x00, 0x1F, 0x00, 0x00, 0x00, 0x11, 0x11, 0x00],      # Char 1 - Upper-middle
[0x00, 0x00, 0x18, 0x04, 0x02, 0x13, 0x11, 0x01],      # Char 2 - Upper-right
[0x12, 0x13, 0x1b, 0x09, 0x04, 0x03, 0x00, 0x00],      # Char 3 - Lower-left
[0x00, 0x11, 0x1f, 0x1f, 0x0e, 0x00, 0x1F, 0x00],      # Char 4 - Lower-middle
[0x09, 0x19, 0x1b, 0x12, 0x04, 0x18, 0x00, 0x00]       # Char 5 - Lower-right
]
```

# ประกาศตัวแปรแบบ List

30. ให้แก้ไขโปรแกรมในข้อ 29 โดยเพิ่มโปรแกรม Create ตัวอักษร แล้วสั่งให้แสดงตัวอักษรนั้นบน Dot Matrix LCD จากนั้นให้อธิบายผลที่ได้

```
mylcd.lcd_load_custom_chars(font)      # create 6 custom characters
mylcd.lcd_clear()
mylcd.lcd_write(0x80)                  # Write first three chars to row 1 directly
mylcd.lcd_write_char(0)                # write character 0
mylcd.lcd_write_char(1)                # write character 0
mylcd.lcd_write_char(2)                # write character 0
mylcd.lcd_write(0xC0)                  # Write next three chars to row 2 directly
mylcd.lcd_write_char(3)                # write character 0
mylcd.lcd_write_char(4)                # write character 0
mylcd.lcd_write_char(5)                # write character 0
```

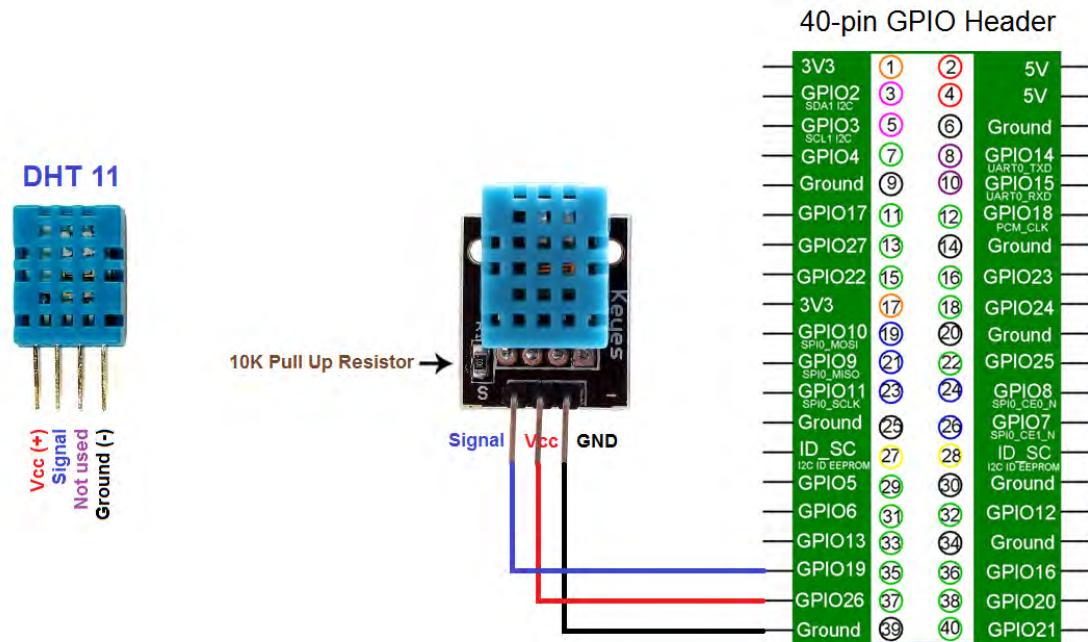
.....  
.....  
.....  
.....  
.....  
.....

31. จากข้อ 30 ถ้านำเอาตัวอักษรหลายตัวมาต่อกันเป็นรูปภาพ จะต้องแก้ไขโปรแกรมอย่างไร ให้ทำการทดลองและอธิบายวิธีการแก้ไข
- .....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

32. ให้เขียนโปรแกรมเพื่อแสดงชื่อนักศึกษาเป็นภาษาไทย บน Dot Matrix LCD โดยทำเป็นตัวอักษรริ้งจากขวาไปซ้าย ทีละตัว

33. จากข้อ 32 ให้นำสวิตช์มาต่อเพิ่ม 1 ตัว กำหนดเงื่อนไขไว้ว่า ถ้ากดสวิตช์ให้ตัวอักษรริ้งจากซ้ายไปขวาทีละตัว และถ้าปลดสวิตช์ตัวอักษรจะริ้งจากขวาไปซ้ายทีละตัว โดยให้แสดงทันทีที่สวิตช์ถูกกด

34. การเชื่อมต่อเซ็นเซอร์วัดอุณหภูมิและความชื้น DHT11 ให้ต่อวงจรเสียงตัว DHT11 เข้ากับขา 40-pin GPIO Header ของ Raspberry Pi ตามรูป โดยมีขา Signal ต่อเข้ากับขา GPIO19 , ขา Vcc ต่อ กับ GPIO13 หรือ GPIO26 ขึ้นอยู่กับตำแหน่งของขาของอุปกรณ์ DHT11 เพื่อทำหน้าที่จ่ายไฟให้กับ DHT11 และขา GND ต่อ กับ Ground



35. ให้ใช้ PIP ที่เป็นตัวจัดการแพคเกจของ Python เพื่อติดตั้งแพคเกจ dht11 ด้วยคำสั่ง
- ```
sudo pip install dht11
```

36. ให้ทำการเขียนโปรแกรมที่ทำหน้าที่อ่านข้อมูลจากเซ็นเซอร์วัดอุณหภูมิและความชื้น DHT11 และ ให้แสดงผลลัพธ์ที่ได้ออกมาทางหน้าจอ Monitor และวิ่งทำการ Run โปรแกรม

```

import RPi.GPIO as GPIO
import dht11
import time

Vcc = 13;                                # Vcc = GPIO13
   # initialize GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()

GPIO.setup(Vcc, GPIO.OUT)                   # Power on DHT11
GPIO.output(Vcc, True)

instance = dht11.DHT11(pin = 19)           # read data using GPIO19
result = instance.read()

if result.is_valid():
    print("Temperature: %-3.1f C" % result.temperature)
    print("Humidity: %-3.1f %%" % result.humidity)
else:
    print("Error: %d" % result.error_code)

```

37. จากการทดลองโปรแกรมในข้อ 36 ในบางครั้งจะเกิดค่า Error ขึ้นให้อธิบายสาเหตุและวิธีการแก้ไข โปรแกรมต้องทำอย่างไร .....
- .....
- .....
- .....
- .....

38. จากข้อ 36 ให้ทำการแก้ไขโปรแกรมเพื่อให้แสดงผลลัพธ์ของการวัดอุณหภูมิและความชื้นที่ได้จากเซ็นเซอร์ DHT11 ออกมาทาง Dot Matrix LCD แทน โดยให้แสดงผลลัพธ์ด้วยคำสั่ง

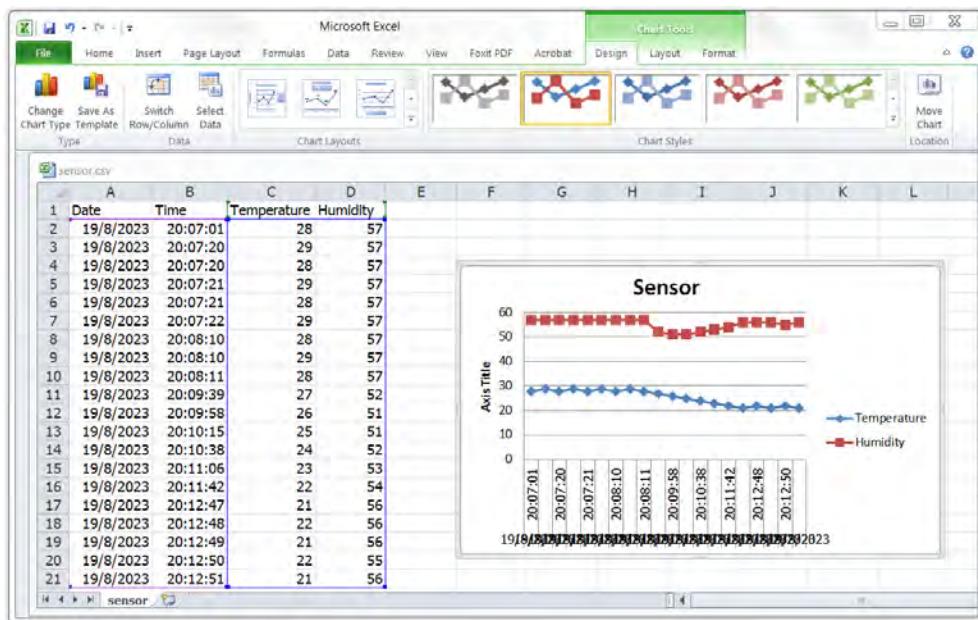
```
mylcd.lcd_display_string("Temp: %d%s C" % (result.temperature, chr(223)), 1)
mylcd.lcd_display_string("Humidity: %d %%" % result.humidity, 2)
```

39. จากข้อ 38 ให้ทำการแก้ไขโปรแกรมเพื่อให้แสดงผลลัพธ์ของอุณหภูมิและความชื้นที่ Dot Matrix LCD ตลอดเวลา ตามจริงแบบ Real-Time โดยใช้คำสั่ง while

40. ให้ต่อสวิตช์เพิ่ม 1 ตัวกับขา GPIO ขาใดก็ได้ และให้แก้ไขโปรแกรมในข้อ 39 เพิ่มโปรแกรมแสดงผลวันที่และเวลาปัจจุบันบนจอ LCD แล้วให้แสดงผลโดยกำหนดเงื่อนไขไว้ว่าทุกครั้งที่มีการกดสวิตช์ จะต้องให้สลับการทำงานระหว่างการแสดงผลของอุณหภูมิและความชื้น กับการแสดงผลของเวลาและวันที่ โดยที่คำสั่งแสดงผลเวลา และวันที่เป็นดังนี้

```
mylcd.lcd_display_string("Time: %s" %time.strftime("%H:%M:%S"), 1)
mylcd.lcd_display_string("Date: %s" %time.strftime("%d/%m/%Y"), 2)
```

41. จากโปรแกรมในข้อ 39 จะทำการเพิ่มโปรแกรมในส่วนของการบันทึกไฟล์ของข้อมูลทั้งหมดที่ได้จากเซ็นเซอร์ และจะบันทึกพร้อมกับวันที่-เวลาที่บันทึกข้อมูลนั้น โดยบันทึกเป็นไฟล์ซีรีส์ Sensor การจัดเก็บข้อมูลภายในจะเป็นรูปแบบฐานข้อมูลที่มีโครงสร้างแบบ CSV มีการจัดเก็บและจัดเรียงข้อมูลแบบตารางสเปรดชีต ตามรูปด้านล่าง



แต่ละแถวในไฟล์จะเป็นรายการบันทึกข้อมูลแต่ละครั้งที่อุณหภูมิการเปลี่ยนแปลงไปจากเดิมมากกว่า 1 องศา การบันทึกค่ามี 4 ฟล็อก (คอลัมน์) ค่านี้ด้วยเครื่องหมายจุลภาค ( , ) โดยเริ่มจากเปิดสร้างไฟล์ CSV ด้วยการฟังก์ชัน open() สำหรับการเขียน (โหมด w) ควรเปิดไฟล์ CSV ด้วย newline=" เพื่อให้อักษรที่เขียนบรรทัดใหม่ภายในช่องที่ยกมาจะได้ตัวความໄດ้ถูกต้อง แล้วเขียนข้อมูลไปยังไฟล์ CSV ด้วยการเรียกใช้เมธอด writerow() โปรแกรมที่ได้จะเป็นดังนี้

```

import RPi.GPIO as GPIO
import dht11
import time
import I2C_LCD_driver
import csv

filename = "Sensor.csv" # name of csv file
Vcc = 13; # Vcc = GPIO13
temp=0

GPIO.setwarnings(False) # initialize GPIO
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()

GPIO.setup(Vcc, GPIO.OUT) # power on DHT11
GPIO.output(Vcc, True)

mylcd = I2C_LCD_driver.lcd()

with open(filename, 'w', newline=") as file : # file is opened in write (w) mode
    writer = csv.writer(file) # creating a csv writer object
    writer.writerow(["Date", "Time", "Temperature" , "Humidity"]) # writing to sensor.csv

    while True : # read data using GPIO19
        instance = dht11.DHT11(pin = 19)
        result = instance.read()

        if result.is_valid() :
            mylcd.lcd_display_string("Temp: %d%6s C" % (result.temperature, chr(223)), 1)
            mylcd.lcd_display_string("Humidity: %d %%" % result.humidity, 2)

            if result.temperature != temp :
                temp=result.temperature
                print(temp)
                writer.writerow([time.strftime("%d/%m/%Y"),time.strftime("%H:%M:%S"), result.temperature, result.humidity])

file.close()

```

42. จากโปรแกรมในข้อ 41 ให้ทดลองการทำงานของโปรแกรมและอธิบายผลที่ได้

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

43. จากไฟล์ Sensor.csv ที่ได้ข้อ 42 ให้ทดลองนำໄปเปิดในโปรแกรม Excel และให้แสดงผลลัพธ์และกราฟที่ได้พร้อมทั้งอธิบายตัวอย่างการนำໄปประยุกต์ใช้งาน

44. จากโปรแกรมในข้อ 41 ให้แก้ไขโปรแกรมเพิ่มข้อ 40 รวมเข้าไปยังโปรแกรม แล้วให้แสดงผลทดลองการทำงาน  
บันทึกเป็นคลิปวิดีโอตั้งแต่เริ่มต้นจนถึงการนำไฟล์ CSV ไปใช้งาน