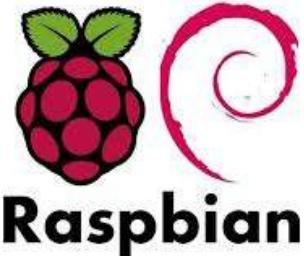


การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ
Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming

1/4 – Getting Start Raspberry Pi4 and GPIO

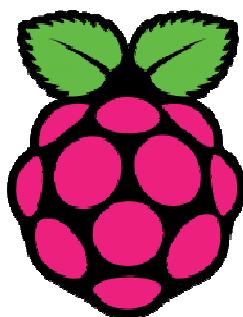
- เริ่มต้นใช้งาน Raspberry Pi
- การติดตั้งระบบปฏิบัติการให้ Raspberry PI
- การโปรแกรมเพื่อควบคุม I/O ด้วย Python
- การโปรแกรมเพื่อควบคุม I/O ด้วย Node-RED
- การถอดและติดตั้ง Node-RED บน Raspberry Pi
- คำแนะนำท้ายบทเพื่อทดสอบความเข้าใจ



1/6 – Raspberry PI

<http://www.raspberryhome.net/article>

- Introduction to Raspberry PI: [Jobs21a Raspberry Pi Basic.pdf](#)
- Raspberry PI SOC architecture: [Jobs21b Raspberry Pi Architecture.pdf](#)
- Other development board compare with Raspberry PI



Raspberry Pi



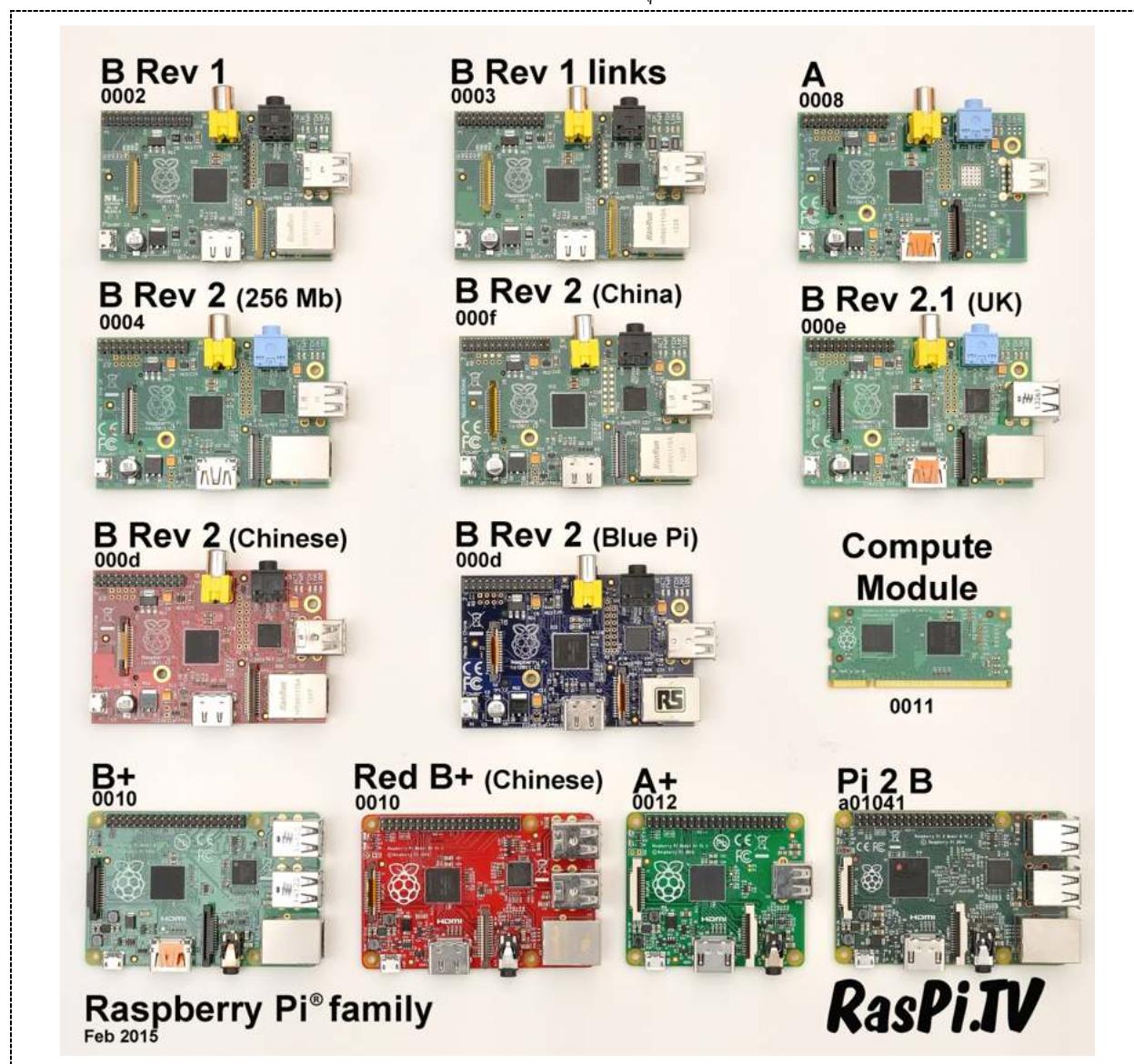
Compute Module 4 *Built-in Wireless

1.1 Raspberry Pi คืออะไร?

Raspberry Pi (ออกเสียงว่า ราส-เบอร์-รี-พาย) เป็นเครื่องคอมพิวเตอร์ขนาดจิ๋ว ที่มีขนาดเพียงเท่ากับบัตรเครดิต ที่สำคัญคือ RPi นี้มีราคาถูกมาก เมื่อเทียบกับคอมพิวเตอร์เดสก์ท็อปปกติ คือ มีราคาเพียงแค่หนึ่งพันกว่าบาทเท่านั้นเอง RPi สามารถทำงานได้เหมือนเครื่องคอมพิวเตอร์ทุกอย่าง เราสามารถต่อ RPi นี้เข้ากับจอคอมพิวเตอร์หรือจอทีวีที่รองรับ HDMI นอกจากต่อจอแสดงผลแล้ว ก็ สามารถต่อมาส์และคีย์บอร์ดผ่าน USB port ปกติ เพราะฉะนั้นสามารถนำมาส์และคีย์บอร์ดที่มีอยู่แล้ว มาต่อได้เลย ระบบจ่ายไฟของraspberrypiใช้สาย Mini USB ที่เราใช้ชาร์จมือถือและอุปกรณ์อื่นๆ เช่น

กับคอมพิวเตอร์ หรือเข้ากับหัวชาร์จไฟมือถือก็ได้ เช่นกัน กรณีเป็น RPi4 จะใช้สาย USB-Type C ขนาดกระแส 3A และมี 2 ช่องต่อจอดisplayแบบ micro-HDMI

raspberrypi (Raspberry Pi) เกิดขึ้นในปี 2549 ที่มหาวิทยาลัยเคมบริดจ์ ประเทศอังกฤษ โดยผู้สร้างทั้งสี่คนคือ อีเบน อัพตัน, ร็อบ มูลลินส์, แจ็ค แลง และ อัลัน มายครอฟท์ มีจุดมุ่งหมายที่จะให้ raspberrypi เป็นคอมพิวเตอร์ราคาถูกเท่าๆ icro ที่สามารถหาซื้อได้ และสามารถศึกษาการทำงานของคอมพิวเตอร์พร้อมทั้งเขียนโปรแกรมง่ายๆ ได้ทันที การที่raspberrypi เป็นบอร์ดวงจรรวมที่เปลี่ยนไปเรื่อยๆ ทำให้เด็กๆ ได้เห็นชิ้นส่วนทั้งหมดที่เป็นส่วนประกอบของคอมพิวเตอร์ได้อย่างชัดเจน ซึ่งจะทำให้เข้าใจการทำงานของคอมพิวเตอร์ในปัจจุบันที่มีในกล่องสวยงามได้มากขึ้น





Network Speed test (transfer 700MB file)

Raspberry Model B+
3.0 MB/sec
(3:40 seconds)



Raspberry Zero v2
using micro USB ethernet adapter
based on 88772a chipset
4.5 MB/sec
(2:35 seconds)



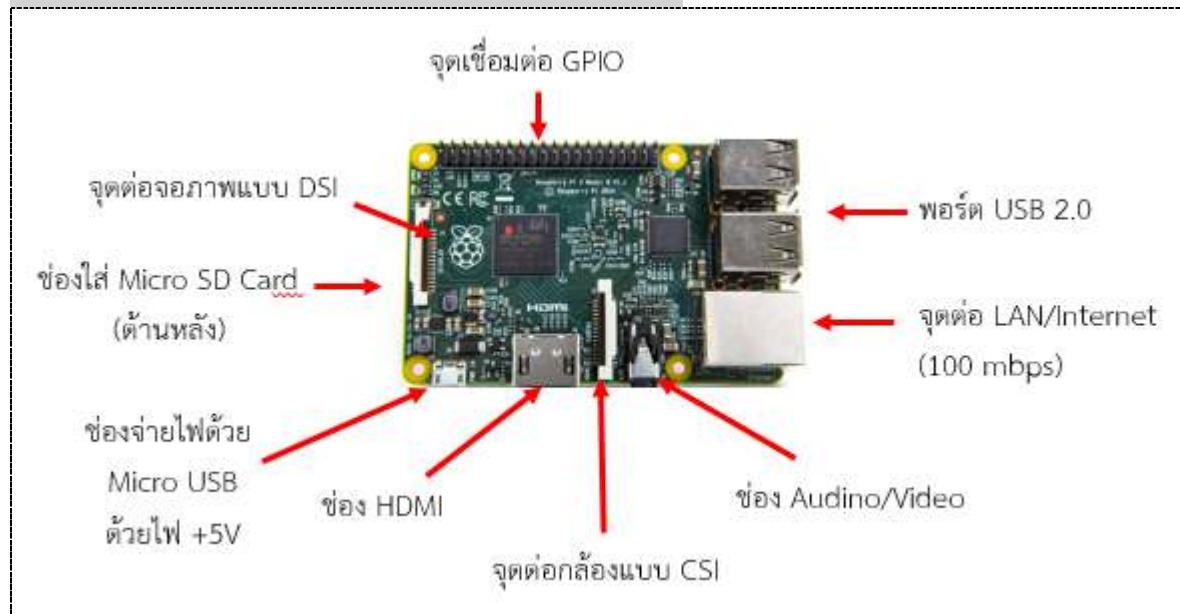
Raspberry 2 Model B
9 MB/sec
(1:20 seconds)



Raspberry 3 Model B
9 MB/sec
(1:20 seconds)

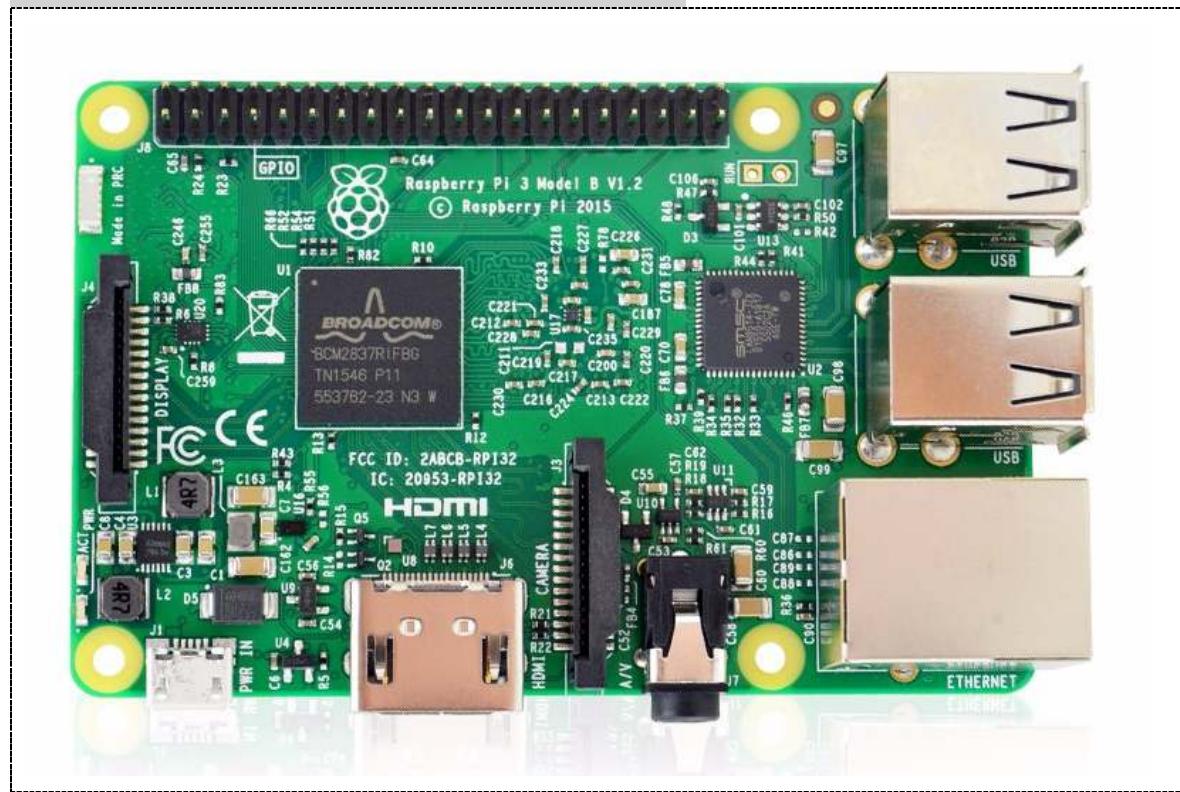
www.pi3dscan.com

1.2 สเปกและส่วนประกอบของ RASPBERRY PI 2 MODEL B

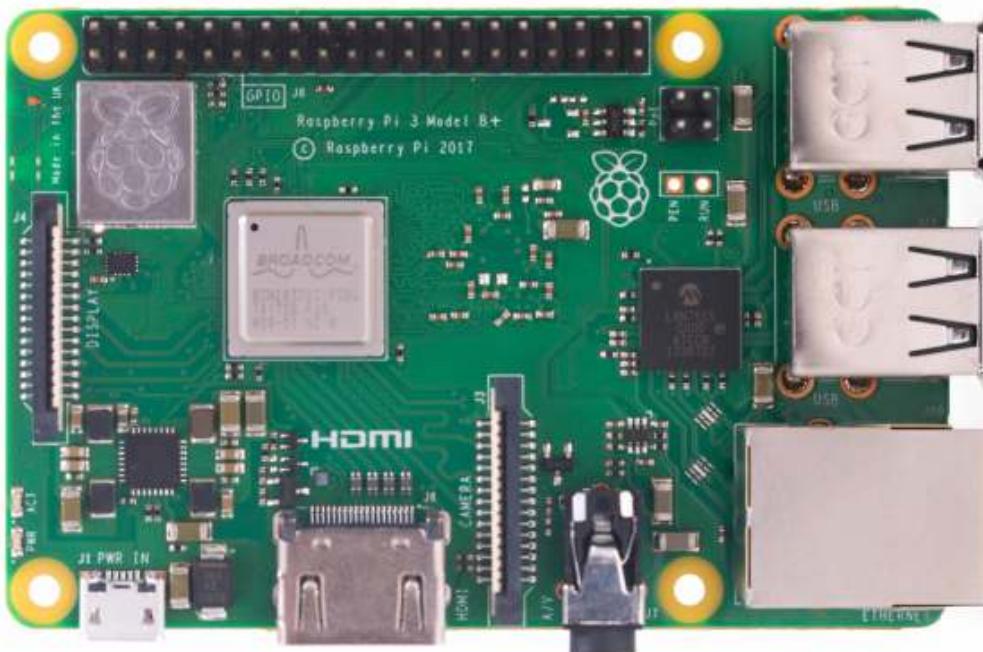


- ชิปปี้ 900 MHz quad-core ARM Cortex-A7 (เร็วกว่า B+ 6 เท่า)
- แรม 1 GB LPDDR2 SDRAM (มากกว่าเดิม 2 เท่า)
- ประมวลผลภาพ VideoCore IV 3D
- USB 2.0 4 พอร์ต
- GPIO (General Purpose Input/Output) 40 จุด
- พอร์ต HDMI
- พอร์ต Ethernet 10/110
- ช่องลับบูนาณเลี้ยง และ ภาพ ขนาด 3.5 มม.
- ช่องต่อลับบูนาณกล้องแบบ CSI
- ช่องต่อลับบูนาณภาพแบบ DSI
- ช่องใส่ Micro SD card

1.3 สเปคและสวนประกอบของ RASPBERRY PI 3 MODEL B



1.4 สเปคและสวนประกอบของ RASPBERRY PI 3 MODEL B+ (@20180404)



The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range.

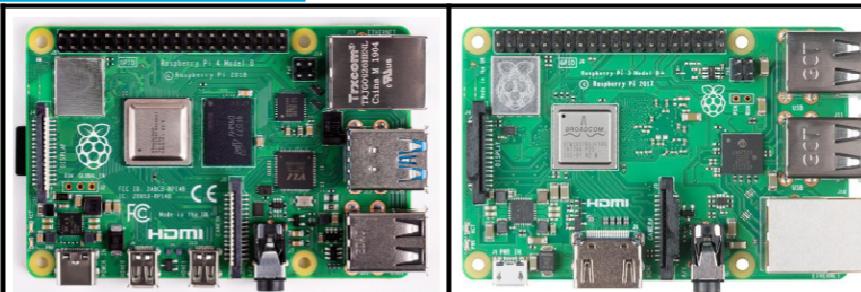
- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN
- Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for 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)

1.5 สเปคและส่วนประกอบของ RASPBERRY PI 4 MODEL B (@24TH JUNE 2019)

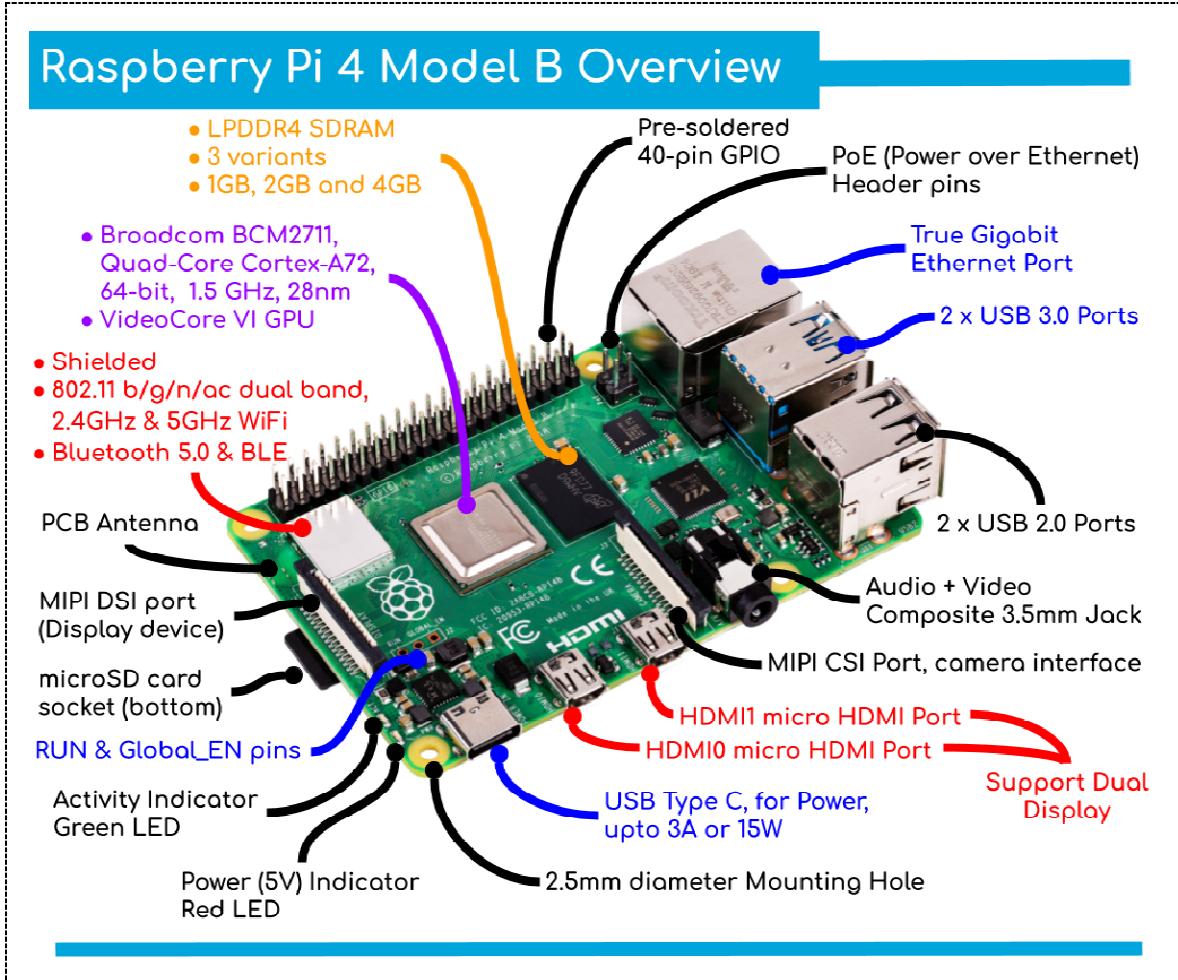
[HTTPS://TH.CYTRON.IO/C-RASPBERRY-PI/C-RASPBERRY-PI-MAIN-BOARD/P-RASPBERRY-PI-4-MODEL-B-4GB](https://th.cytron.io/c-raspberry-pi/c-raspberry-pi-main-board/p-raspberry-pi-4-model-b-4gb)

Announced on the 24th June 2019, the new Raspberry Pi 4 Model B is released and it's available on the Cytron marketplace! Check out the Major Upgrades on Raspberry Pi 4 Model B. And here is the comparison table between Raspberry Pi 4 Model B vs Raspberry Pi 3 Model B+:

Raspberry Pi 4B vs 3B+



Features/Specs	Raspberry Pi 4 Model B	Raspberry Pi 3 Model B+
Release Date	24th June 2019	14th March 2018
SoC Type (Processor)	Broadcom BCM2711 (with metal cover)	Broadcom BCM2837B0 (with metal cover)
Core Type	Cortex-A72 64-bit (ARMv8)	Cortex-A53 64-bit (ARMv8)
No. of Cores	Quad-Core	
GPU	VideoCore VI	VideoCore IV
Multimedia	H.265 decode (4Kp60) H.264 decode (1080p60) H.264 encode (1080p30) OpenGL ES 1.1, 2.0, 3.0 Graphics	H.264, MPEG-4 decode (1080p30) H.264 encode (1080p30) OpenGL ES 1.1, 2.0 Graphics
CPU Clock	1.5 GHz	1.4 GHz
Memory/OS storage	microSD	
RAM	LPDDR4 1GB, 2GB and 4GB options	LPDDR2 1GB
Ethernet	True Gigabit Ethernet	Gigabit over USB 2.0 (Max 300Mbps)
USB Port	2 x USB 3.0 + 2 x USB 2.0	4 x USB 2.0
HDMI	2 x micro HDMI support Dual Display	1 x full size HDMI
WiFi	802.11 b/g/n/ac (2.4GHz+5GHz & Shielded)	
Bluetooth	5.0 + BLE (Shielded)	4.2 + BLE (Shielded)
Antenna	PCB Antenna (Similar to Rpi Zero W)	
GPIO	40 pins (Fully backwards-compatible with previous boards)	
Operating System	Raspbian (> 24 June 2019)	Raspbian (> March 2018)
Dimension	85mm x 56mm	
Power Input	5V via USB Type C (upto 3A) 5V via GPIO header (upto 3A) Power over Ethernet, requires PoE HAT	5V via USB Micro B (upto 2.5A) 5V via GPIO header (upto 3A) Power over Ethernet, requires PoE HAT



As the Raspberry Pi 4 Model B uses newer processors and RAM, it will need the latest Raspbian or NOOBS to boot, please check and get it from here, the official download side of the Raspberry Pi Foundation. Alternatively, get the 16GB microSD card pre-loaded with latest NOOBS here.

Note: Since Raspberry Pi 4 Model B comes with USB type C for the power, you can get the official power adapter for Raspberry Pi 4 Model B that comes with USB type C jack and capable of delivering up to 3A, or 15W of power. Yet, not to disappoint the existing users, the power adapter for Raspberry Pi 3 Model B+ or Raspberry Pi Zero that comes with USB micro B jack can still be used to power RPi 4B, just get the micro B to USB type C adapter here, it can still deliver 2.5A or 12.5W of power, sufficient for most of projects.

Do check out other accessories and raspberry pi boards that we carry here.

Features and Specs:

- Broadcom BCM2711, 64-bit ARM Cortex-A73 (ARMv8), Quad-core, 28nm Processor SoC running @ 1.5GHz, with metal body for better heat dissipation.
- LPDDR4 4GB SDRAM
- 2 x USB3.0 Port + 2 x USB2.0 Port
- Expanded 40-pin GPIO Header
- Video/Audio Out:
 - 4-pole 3.5mm connector
 - 2 x micro HDMI port, support 4K video, dual display
 - Raw LCD (DSI)
- Image/Camera input: CSI port
- Storage: microSD
- True Gigabit Ethernet, >900 Mbps
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 5.0 and BLE (Bluetooth Low Energy)
- Low-Level Peripherals:
 - 27 x GPIO
 - UART
 - I2C bus
 - SPI bus with two chip selects
 - +3.3V
 - +5V
 - Ground
- Power Requirement, 5V/3.0A via USB type C connector.
- Supports Raspbian, and check 3rd party Operating System here.
- Dimensions: 85mm x 56mm x 17mm

1.6 Raspberry Pi Compute Module 4 (CM4)

<https://www.arduintronics.com/product/4659/raspberry-pi-compute-module-4-io-board-wi-fi-100>

<https://www.arduintronics.com/product/4436/raspberry-pi-compute-module-4-1gb-no-mmc-no-wifi-lite>

DESCRIPTION
 <p>The Raspberry Pi Compute Module 4 is based on the Raspberry Pi 4 Model B, but in a smaller form factor - perfect for embedding into products or projects without the bulk of a classic Raspberry pi. You get all the computational power of Raspberry Pi 4 in a compact form factor for deeply embedded applications. The CM4 incorporates the same quad-core ARM Cortex-A72 processor, dual video output, gigabit Ethernet, UART, I2C, SPI, I2S and a few PWM for good measure.</p>

RAM, eMMC and Wireless Options

This module is available in multiple variants, with a range of RAM and eMMC Flash options, and with or without wireless connectivity. The modules are available with 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM with optional storage of 8GB, 16GB or 32GB eMMC Flash. The wireless option includes 2.4GHz and 5GHz 802.11b/g/n/ac wireless LAN and Bluetooth 5.0 for BT classic and BTLE support.

This product version of the CM4 modules comes with 1 GB of RAM, no MMC and no WiFi/BT which makes it very inexpensive!

Specifications

- 1.5GHz quad-core 64-bit ARM Cortex-A72 CPU
- VideoCore VI graphics, supporting OpenGL ES 3.x
- 4Kp60 hardware decode of H.265 (HEVC) video
- 1080p60 hardware decode, and 1080p30 hardware encode of H.264 (AVC) video
- Dual HDMI interfaces, at resolutions up to 4K
- Single-lane PCI Express 2.0 interface
- Dual MIPI DSI display, and dual MIPI CSI-2 camera interfaces
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM
- Optional 8GB, 16GB or 32GB eMMC Flash storage
- Optional 2.4GHz and 5GHz IEEE 802.11b/g/n/ac wireless LAN and Bluetooth 5.0
- Gigabit Ethernet PHY with IEEE 1588 support
- 28 GPIO pins, with up to 6 × UART, 6 × I2C, and 5 × SPI



DESCRIPTION

Exposing every interface from Raspberry Pi Compute Module 4, the Compute Module 4 IO Board provides a development platform and reference base-board design for the most powerful Compute Module yet. The **Compute Module 4 IO Board** is a development board for those who wish to make use of the Raspberry Pi in a more flexible form factor, intended for industrial applications.

This item is just the IO board and does not contain a matching Compute Module. While the Compute Module contains the guts of a Raspberry Pi 4 (1.2GHz, quad-core Broadcom BCM2837 processor), it does not have any easy-to-use ports for development. That's where this IO Board comes in!

Specifications

- 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

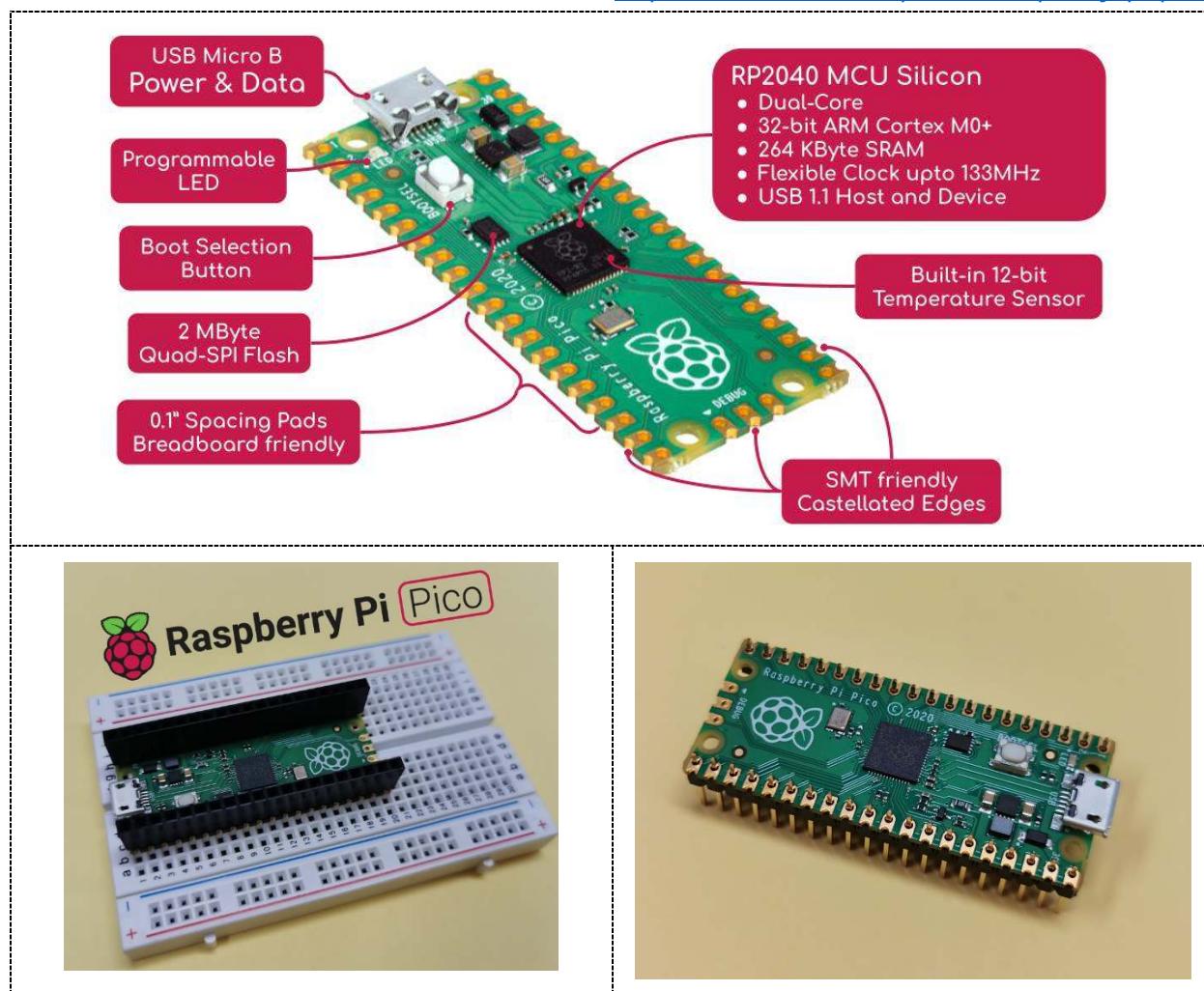


This **PoE CM4 IO board** breakouts quite some extra peripheral ports if we compared it with Raspberry Pi 4 Model B:

- 2 x DSI display port - The DSI connectors are 22-way 0.5mm pitch to pitch. These connectors are the same as the existing CMIO board but different from the Raspberry Pi 4 Model B.
- 2 x CSI camera input port - The CSI connectors are 22-way 0.5mm pitch to pitch. These connectors are the same as the existing CMIO board but different from the Raspberry Pi 4 Model B. Now you can get an extra camera input on the CM4 IO board. The CSI connectors are compatible with the Raspberry Pi Zero CSI connector, so the FFC/FPC cable can be utilized for the camera module interface on the CM4 IO board.
- 15cm Raspberry Pi Zero CSI FFC/FPC cable
- 30cm Raspberry Pi Zero CSI FFC/FPC cable
- 4 x USB 3.2 Gen1 Port - This board provides four, not two USB 3.0 ports and those like to connect to high-speed SSD Storage.
- 2 x Full-Size HDMI port - Again, support dual displays
- 1 x Cooling Fan connector - Support standard +12VDC cooling fan with PWM drive and tacho output. An EMC2301 controls the fan via I2C.
- 1 x RTC with Coin Cell Socket - for onboard Real Time Clock keeping time running even after the CM4 and IO board is power off.
- 1 x DC Barrel Jack - A DC Barrel Jack with 5.5mm*2.1mm, accepting a wide range of input voltage to power the IO board. Ranging from 7V to 36VDC.
- 1 x USB Micro B Slave port - There is a USB Micro B port for CM4 to be updated via rpi boot.
- 1 x MicroSD card socket (at the bottom of PCB) - This is only meant for CM4 Lite as it does not come with onboard eMMC memory for Operating System and will require external memory to boot. This PoE CM4 IO board comes with a push-push type socket :)

1.7 Raspberry Pi Pico

<https://inex.co.th/home/product/raspberry-pi-pico/>



บอร์ดไมโครคอนโทรลเลอร์รุ่นใหม่จาก Raspberry Pi Foundation ผู้ผลิตบอร์ดคอมพิวเตอร์ขนาดเล็กระดับโลกแห่งสหราชอาณาจักร โดยได้ทำการพัฒนาชิปไมโครคอนโทรลเลอร์ชิ้นมาเอง ชิปเบอร์แรกที่ผลิตออกมานี้คือ RP2040 นับเป็นพัฒนาการที่สร้างแรงกระเพื่อมแก่วงการระบบสมองกลฝังตัวทั่วโลก โดยทาง Raspberry Pi Foundation ได้วางแผนเพื่อออกแบบและผลิต RP2040 มาตั้งแต่ปี 2016 ใช้เวลาในการกำหนดแบบและผลิตชิปตัวอย่างทางวิศวกรรมเพื่อส่งให้นักพัฒนาหารดwareทั่วโลกทำการทดสอบ จนกระทั่งพร้อมเพื่อนำเสนอต่อตลาดโลกพร้อมกันในวันที่ 21 มกราคม 2564

โครงการ Raspberry Pi Pico นี้ยังไม่เป็น open source แต่ Raspberry Pi Foundation ได้เปิดเผยข้อมูลทางเทคนิคทั้งハードแวร์และซอฟต์แวร์ ตลอดจนชุด SDK เพื่อให้ผู้สนใจนำไปพัฒนาต่อโดยได้ตามต้องการ รวมถึงไฟล์แผ่นวงจรพิมพ์ของบอร์ดที่รองรับชิป RP2040 ที่สร้างจาก KiCad ให้นักทดลองไปสร้างเองได้ (แต่การจำหน่วยชิป RP2040 แยกจะต้องใช้เวลาอีกสักระยะหนึ่ง) ทาง Raspberry Pi Foundation ได้เตรียมข้อมูลเป็นหนังสือวิจัยทรัพนิกส์ (ภาษาอังกฤษ) จำนวนมาก และมีความละเอียดมากเพียงพอ รวมถึงหนังสือการทดลองเพื่อสนับสนุนการเรียนรู้สำหรับผู้เริ่มต้นด้วย

คุณสมบัติทางเทคนิค

- ติดตั้งชิปไมโครคอนโทรลเลอร์ RP2040 จาก Raspberry Pi สหราชอาณาจักร
- ส่วนของโปรเซสเซอร์เป็น Arm Cortex M0+ 2 เกณ วงจรกำเนิดสัญญาณนาฬิกา มีความถี่ด้วยกัน และทำงานได้ที่ความถี่สูงสุด 133MHz
- มีหน่วยความจำสำหรับแรม 264KB และหน่วยความจำแฟลชแบบ SPI ความจุ 2MB
- รองรับการทำงานเป็น USB1.1 โหมดสต์ เพื่อเชื่อมต่อกับอุปกรณ์ USB ได้อย่างหลากหลาย
- มีการทำงานในโหมดประหยัดพลังงานในแบบสลีป (sleep) และดอร์เมนต์ (dormant)
- ดาวน์โหลดโคเด็ตแบบลากและวางไปยังดิสก์ไดรฟ์ผ่านพอร์ต USB
- ขาพอร์ตอินพุตเอาต์พุตแบบมัลติพังก์ชัน 26 ขา รองรับการสร้างสัญญาณบัส I2S และสัญญาณเชื่อมต่อภาพแบบ VGA
- มีขาพอร์ตรองรับการสื่อสารข้อมูลอนุกรัมผ่านบัส SPI 2 ชุด, I2C 2 ชุด และ UART 2 ชุด
- มีขาพอร์ตอินพุตและออก 3 ขา สำหรับวงจรแปลงสัญญาณอะโนดลีก เป็นดิจิทัล ความละเอียด 12 บิต มีอัตราการสูงสัญญาณ 500 กิโลแซมบลิงต่อวินาที รองรับแรงดันไฟตรงสูงสุด 3.3V
- มีขาพอร์ตรองรับการทำงานสร้างสัญญาณ PWM แบบควบคุมได้ 16 ช่อง
- ไกเมอร์และฐานเวลาความเที่ยงตรงสูง
- มีตัวตรวจจับอุณหภูมิภายในชิป
- มีแลบารีคำนวณคณิตศาสตร์เลขทศนิยมในตัวชิป
- พอร์ตอินพุตเอาต์พุตโปรแกรมได้ (PIO) เป็นสเตดแมชชีนแบบโปรแกรมได้ 8 ชุด ทำให้ปรับแต่งการทำงานของขาพอร์ตได้อย่างเป็นอิสระและหลากหลาย รองรับอุปกรณ์เพอร์เซปต์ในแบบเฉพาะได้
- มีขาพอร์ตรองรับการตีบักแบบ ARM Serial Wire Debug หรือ SWD 3 ขา
- ไฟเลี้ยงบอร์ดยานกาวง 1.8 ถึง 5.5V บนบอร์ดมีวงจรภาคจ่ายไฟควบคุมไฟเลี้ยงคงที่ 3.3V ในแบบเพิ่มและลดอัตโนมัติเมื่อไฟเลี้ยงเข้ามาต่ำกว่า 1.8V และสูงกว่า 3.3V กระแสไฟฟ้าขณะทำงาน 87 ถึง 92mA ใช้แบตเตอรี่ AA หรือ AAA 3 ก้อนเป็นแหล่งจ่ายไฟได้
- การใช้กระแสไฟฟ้าในโหมดสลีป 1.3mA และ 0.8mA ในโหมดดอร์เมนต์

การพัฒนาโปรแกรม

- รองรับการพัฒนาโปรแกรมด้วย microPython และ C/C++
- มีชุด SDK ให้ผู้สนใจนำไปพัฒนาต่อยอดเพิ่มเติมได้

1.8 What is the Raspberry Pi 5 going to be like?

<https://www.upswift.io/post/raspberry-pi-5-what-we-are-going-to-see>

Before going any deeper - have a Raspberry Pi 3 or 4? it's time to manage & access it remotely using Upswift.io

To answer the question, let's first discuss the key aspects of a general-purpose single board computer which are most sought-after.

1. Performance

Raspberry Pi has always been in partnership with Broadcom, trusting their line of SoC to be the heart of the Raspberry Pi lineup. The last member of the Broadcom SoC family to end up in a Raspberry Pi was the Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC, which comes as the main processor of the Raspberry Pi 4B. It's highly unlikely that Raspberry Pi will this time go with a different manufacturer, therefore another beefed-up Broadcom SoC can be expected to be in the Raspberry Pi 5, probably in the range of 2GHz base clock. Also, there's a chance that Raspberry Pi *may* develop its own SoC, although chances of having one as early as in a Raspberry Pi 5 is highly unlikely.

Talking about RAM, we'd like to see one of the Raspberry Pi 5 versions getting as much as 16GB of LPDDR5 RAM, which is not only power efficient, but also faster than its predecessor LPDDR4 SDRAM.

2. Cooling

We all know that the Raspberry Pis can get quite hot when they run on full power. With the reduced size and more processing power cramped in, we can expect the Pi5 to get even hotter than the current ones. An additional cooling system (i.e. copper heat pipes, larger area for mounting heat sinks) would be ideal to increase the lifespan of the computer as well as extract as much juice as possible.

3. Storage

So far, all the Raspberry Pi development boards (except the compute modules) have been equipped with MicroSD or SDHC card slots. In the production run, this makes the BOM cost lower but leaves the user to go through the hassle of swapping SD cards between computers and the Pi, flashing boot images etc. Of course this type of setup has its own advantages such as easy backup and restoration, but it'd be nice to see a Pi5 with a built-in eMMC storage with some form of BIOS configuration that will allow the Pi to be booted from a USB flash drive (how about boot from LAN for a change? :D). Adding eMMC storage will not only provide faster IO, but also the data will be less prone to damage and theft.

4. Ports

There isn't much that can be included into such a small form factor than it already has, but we believe there should be one thing that *must* make a comeback in the next Raspberry Pi; the full-size HDMI port(s). It's been a nightmare to connect all those micro-HDMI ports to convert them into full-size HDMI by adapters that need to be purchased separately. The GPIO header is expected to be of the same setup to maintain the backward-compatibility. If an eMMC storage is used, it'll be nice to have another USB-C port to access the eMMC as a USB-mass storage device, which maintains the existing flexibility

of flashing and extracting boot images from the device while keeping the speed and reliability of the eMMC storage.

5. Connectivity

Other than the default Gigabit Ethernet, USB2.0 and USB3.0 ports, the Raspberry Pi 5 is likely to receive an upgrade for Bluetooth to v5.2 for faster speeds and increased communication range. For added range, SMA connector options for attaching external antennas for WiFi and Bluetooth may also come in handy for production device integration.

6. Display Outputs

Like it's mentioned before, the comeback of a full-size HDMI port is highly anticipated, along with some improvements to the display output system such as 4k 60fps when running dual-displays. The current version only supports 4k30fps when in dual display mode and 4k60fps in single display setup and this is expected to change in the Raspberry Pi 5.

What does the community want the Raspberry Pi 5 to be like?

Obviously, *LESS BUGS!*



As obvious as it may sound, the Pi 5 is expected to mitigate the issues the current versions have such as:

1. USB-C power issue

There are many USB-C power adapters that come with fast charging technologies such as QuickCharge, DashCharge or SuperCharge. A hardware design flaw in the Raspberry Pi 4/4B has been known to cause some adapters to mis-interpret the device type and provide more than 5V for powering it, and inadvertently frying the board's power supply or worse, the whole SoC. There have also been some cases where some adapters have refused to power the Pi at all. Maya Posch explains the cause for this in-depth in her [article](#) in [Hackaday](#). Therefore, we have high hope that the Pi 5 will not be having the same design flaws; or worse.

2. USB hub and Ethernet chip heating up

There have been instances that the Via Labs VL805 4 port USB hub and the LAN9512 Gigabit Ethernet controller on the Pi4/4B reported to be notoriously getting hot, even when there are no devices plugged in. This has been experienced by only some of the users and no specific reason has been found. The Pi 5 is expected to be free from this mishap since increased idle temperature means decreased lifespan of the components.

3. Low Voltage warning

There has been a mysterious pop up in the desktop asking to check the power supply, even when the power supply should be more than sufficient for the Pi. In the Raspberry Pi 5, this issue is expected to be fixed and we'd like to see it trigger only when the Pi is really struggling to draw enough current from the power supply.

A twist of Raspberry Pi Pico in the Raspberry Pi 5?

Not so long ago, Raspberry Pi introduced their very own microcontroller, the Raspberry Pi Pico with a boatload of features such as the Programmable IO state machine subsystem. We know that being a microprocessor-based system, Raspberry Pis are not designed for executing time critical tasks all-too-well such as PWM signal generation, DAC and ADC functions; not to mention interrupt handling. Therefore, how about an integration of some of the most sought-out features from the Pico into the Raspberry Pi 5 development board such as,

1. Analog Inputs (ADC)
2. Hardware PWM outputs
3. Programmable IO state machine system
4. Additional hardware UART interfaces
5. Accelerated on-chip floating point and integer libraries
6. Hardware interrupts

And one or two essential features such as an RTC and low power modes backed by the microcontroller portion of the system.

An afterthought.

These are what we think the core improvements that can be expected to be included in the Raspberry Pi 5. We know it's quite a bit to ask for from the maker of the Raspberry Pi. But some of the features are essential to save the troubleshooting time and focus more on the development of the IoT systems.

Therefore, though the Raspberry Pi 5 may not be perfect, we expect that the design engineers will make sure to include at least some, if not all those improvements in their next release, the Raspberry Pi 5.

Need to manage your Raspberry Pi remotely?

Here is how to access (SSH, VNC), update, monitor and control your Linux device remotely - [Upswift.io](#)

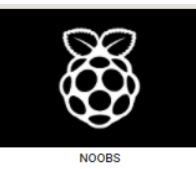
The screenshot shows the Uswift.io dashboard with the following sections:

- PRODUCT FLEET:** Shows 3 ONLINE, 1626 OFFLINE, and 1629 FLEET.
- LIVE ALERTS:** Shows 2 RESOURCES, 0 PROCESS, and 0 DATA.
- FLEET SECURITY:** Shows MEDIUM CVE, LOW AUTH, and 110 LAST SCAN.
- SOFTWARE STATUS:** A pie chart showing software versions: v5.1.0 (green), v2.8 (orange), v5.0.5 (purple), v5.0.4 (pink), and v1.9 (yellow).
- DEVICE REGISTRATION:** A bar chart showing new device registrations: Feb (~1,000) and Mar (~1,500).
- NOTIFICATIONS:** A table listing recent events:

CREATED TIME (PTO)	LEVEL	DESCRIPTION
2021-03-22 14:06	Info	Started update deployment on 1 devices (version: v5.1.0)
2021-03-22 13:56	Info	Started update deployment on 1 devices (version: v5.1.0)
2021-03-22 13:01	Info	Started update deployment on 1 devices (version: v5.0.9)
2021-03-22 09:15	Warning	Update deployment failed on device: smart screen - 4th avenue(17725). (version: v5.0.8)
2021-03-22 09:14	Info	Started update deployment on 1 devices (version: v5.0.8)
2021-03-22 09:11	Info	Started update deployment on 1 devices (version: v5.0.7)
2021-03-21 10:18	Info	Device: New Device 4021135 has been registered to project: Demo, group: Production
2021-03-21 10:18	Info	Device: New Device 40321134 has been registered to project: Demo, group: Production
- Map:** A map of the United States showing device locations, with a red marker in New York.

We know how important it is to manage your deployments based on Raspberry Pi; such as performing remote updates establishing remote connections such as SSH and remote desktop connections. Managing any Linux device remotely is what Uswift specializes at. We provide a complete remote edge Linux IoT management solution that can be accessed from anywhere in the world to manage your devices with just a few clicks.

1.9 ระบบปฏิบัติการของ RASPBERRY PI

Raspberry Pi Desktop (for PC and Mac)  		<ul style="list-style-type: none"> ▪ noobs ▪ raspbian 	
 Raspberry Pi Desktop		<ul style="list-style-type: none"> ▪ Raspberry Pi Desktop (for PC and Mac) 	
Third Party Operating System Images		<ul style="list-style-type: none"> ▪ ubuntu mate ▪ snappy ubuntu core ▪ windows 10 IOT Core ▪ OSMC ▪ libreelec ▪ PINET ▪ RISC OS ▪ Weater Station ▪ IchigoJame RPI 	
 Ubuntu MATE	 Ubuntu Core	 Ubuntu Server	 OSMC
 LibreELEC	 Mozilla WebThings	 PiNet	 RISC OS
 Weather Station	 IchigoJam RPi		

1.10 อ้างอิง

- <http://www.raspberrypi.org/>
- https://en.wikipedia.org/wiki/Raspberry_Pi
- <http://raspi.tv/2015/updated-pi-family-photo-to-include-pi-2-b>
- <http://www.phoronix.com/scan.php?page=article&item=raspberry-pi-3&num=1>
- <http://www.cnx-software.com/2016/03/01/raspberry-pi-3-odroid-c2-and-pine-a64-development-boards-comparison/>

2/6 – การติดตั้งระบบปฏิบัติการให้ Raspberry PI

ในบทความนี้ จะสอนการใช้งาน Raspberry Pi เป็นต้น ตั้งแต่การติดตั้งระบบปฏิบัติการลง Micro SD card ซึ่งในที่นี่คือ Raspbian โดยใช้กับบอร์ด Raspberry Pi 4 โดยทำใช้ระบบปฏิบัติการ Windows ในการติดตั้ง และจำเป็นต้องต่ออินเตอร์เน็ตด้วย

Lab101 – Install Raspberry PI OS

2.1 อุปกรณ์ที่ต้องใช้

- บอร์ด Raspberry Pi 4
- Micro SD card class 10 ขนาด 16 GB ขึ้นไป
- สาย Micro USB
- สาย micro-HDMI
- หน้าจอแสดงผล (สามารถเชื่อมต่อแบบ HDMI ได้)
- USB Keyboard และ USB Mouse
- สาย LAN สำหรับเชื่อมต่อ Internet หรือ USB Wifi
- คอมพิวเตอร์ + Card Reader

2.2 การติดตั้งระบบปฏิบัติการลง MICRO SD CARD

วิธีที่ 1 – Install Raspberry Pi OS using Raspberry Pi Imager

Step 1a: ดาวน์โหลดอิมเมจ OS จาก <https://www.raspberrypi.org/software/> แล้วเลือก Raspberry Pi Image for Windows

**Install Raspberry Pi OS using
Raspberry Pi Imager**

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)

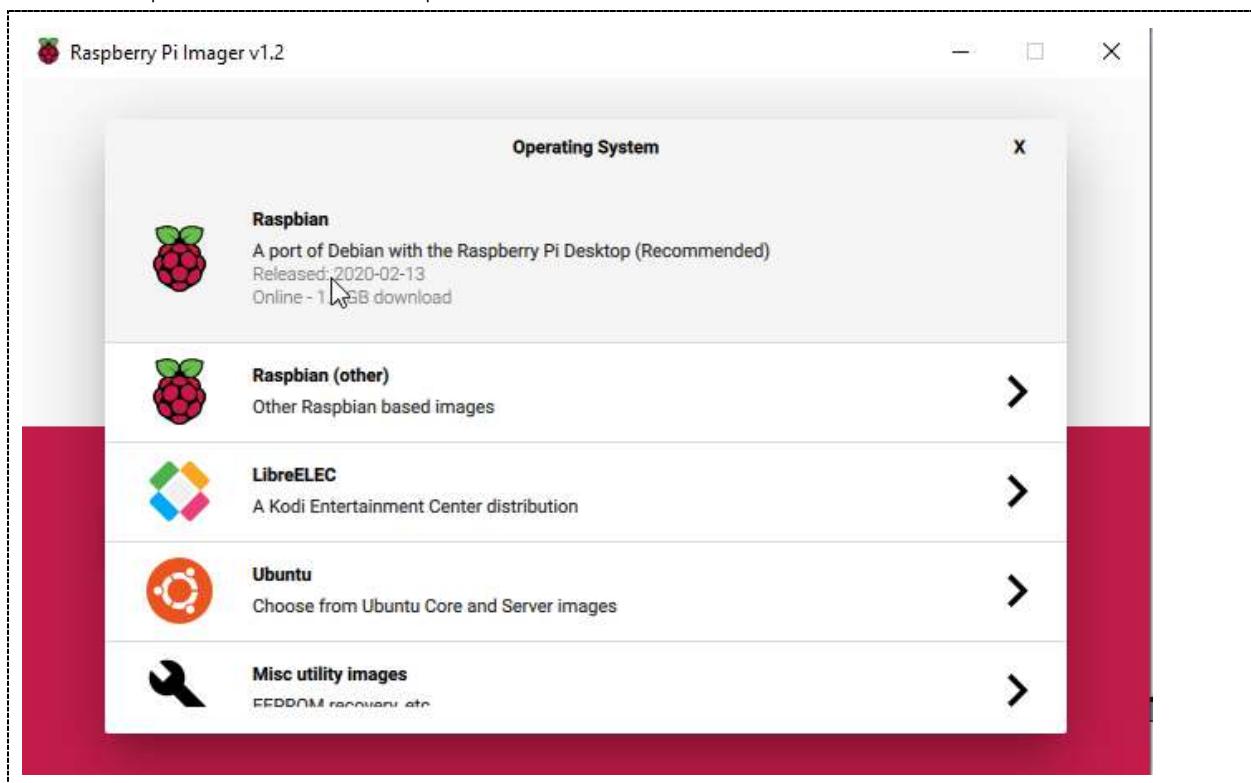
To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`
in a Terminal window.



Step 2a: เมื่อดาวน์โหลดเสร็จเรียบร้อยแล้ว ให้ติดตั้ง image Setup



Step 3a: เลือก OS เป็น Raspbian และเลือก SD Card



Step 4a: Write OS image



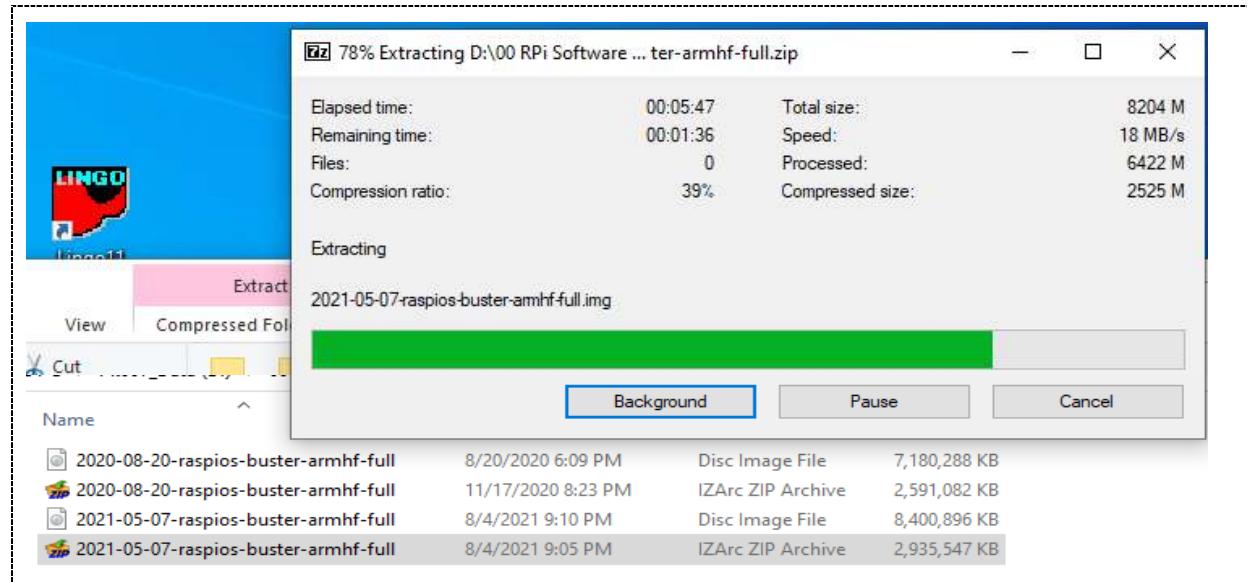
Step 5a: เมื่อติดตั้งเสร็จแล้วให้ถอน Micro SD card ออกจากคอมพิวเตอร์แล้วนำไปที่ Raspberry Pi ต่อไป

วิธีที่ 2 - *Manually install an operating system image*

Step 1b: ดาวน์โหลดอิมเมจ OS จาก <https://www.raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit> และเลือก Raspbian (หรือ Download ด้วย bit torrent ก็ได้)

The screenshot shows the official Raspberry Pi OS download page. At the top, there's a navigation bar with links like "raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit", "Kogoro FC 1412", "EngSUT-WFH2021", "Reg2_Wichai", "Mail_SUT", "Facebook", "OkNation", "สำนักข่าว AMSPORT", "Kapook", "Hindawi", and social media icons. Below the navigation is the title "Raspberry Pi OS". On the left, there's a photo of a Raspberry Pi Model B+ board. Next to it is the text "Compatible with: All Raspberry Pi models". In the center, there are two main download options: "Raspberry Pi OS with desktop software" and "Raspberry Pi OS with desktop". Each option includes details like release date (May 7th 2021), kernel version (5.10), size (2,867MB or 1,180MB), SHA256 hash, and release notes. The "Download" button for the top option is highlighted with a green rounded rectangle. Below each option is a "Download torrent" link.

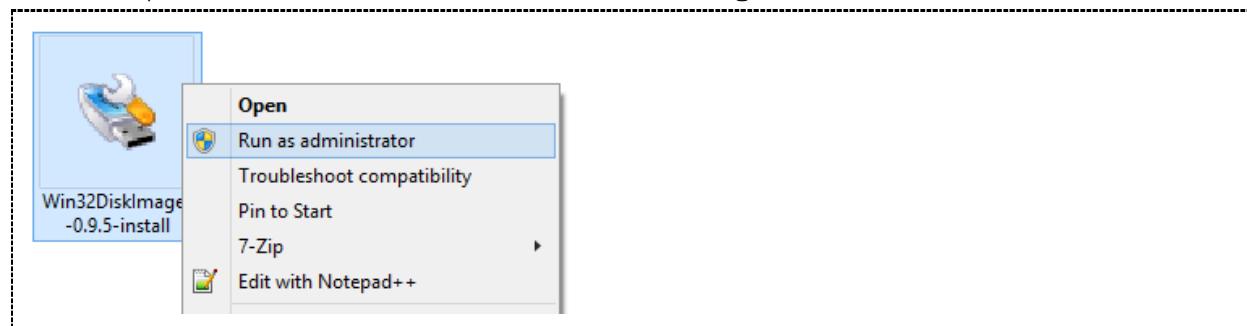
Step 2b: เมื่อดาวน์โหลดเสร็จเรียบร้อยแล้ว ให้ทำการแตกไฟล์จาก 2.9G จะได้ไฟล์ขนาด 8.4G จะได้ไฟล์อิมเมจ (.img) ดังรูป



Step 3b: จาก <https://sourceforge.net/projects/win32diskimager/>



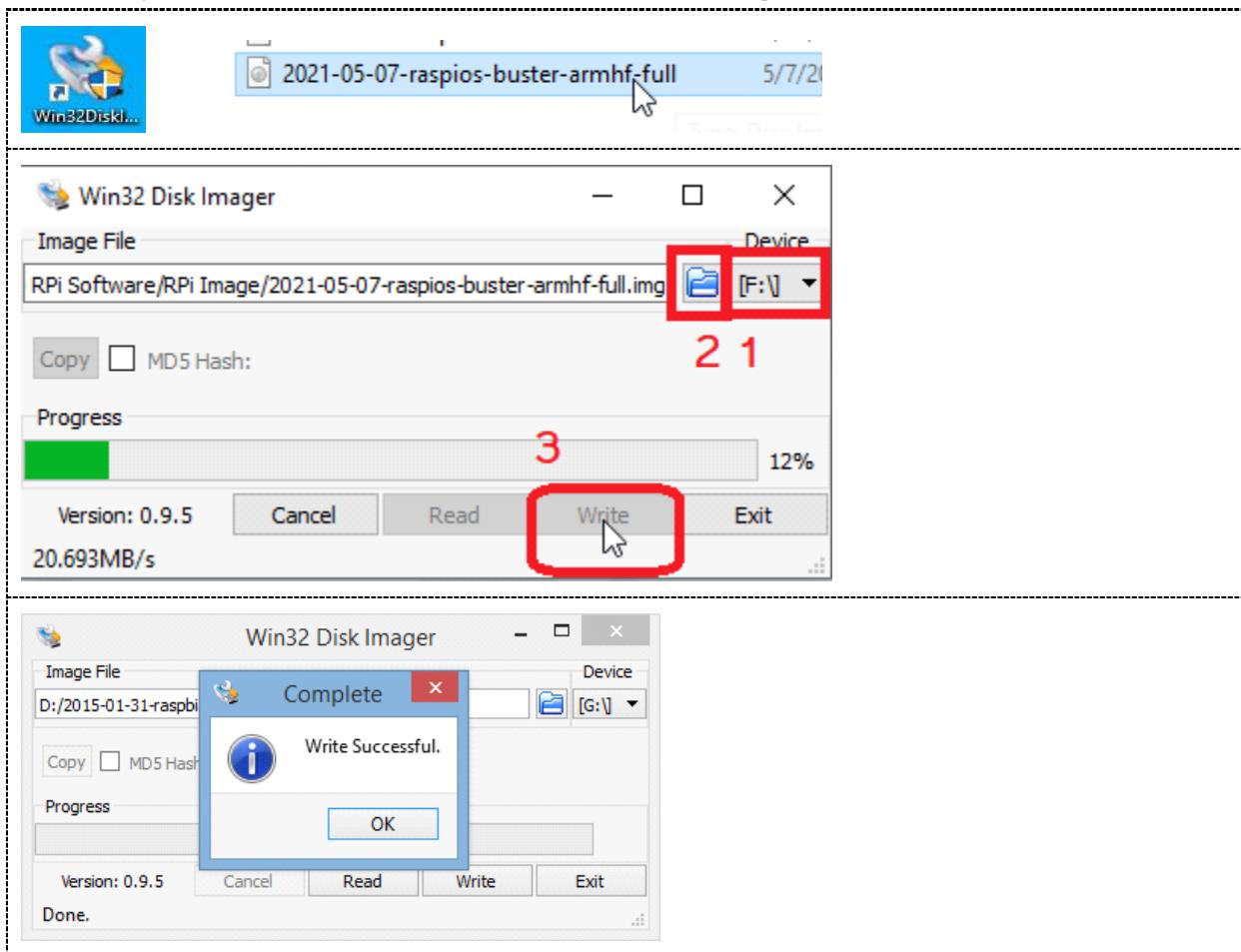
Step 4b: แตกชิปไฟล์ และติดตั้งโปรแกรม Win32DiskImager (คลิกขวา Run as administrator)



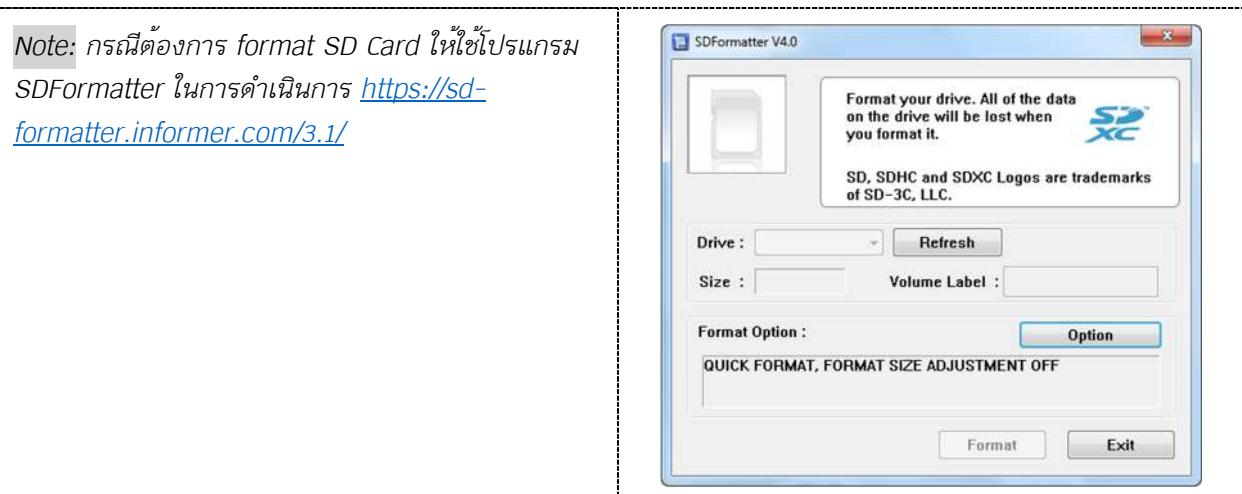
Step 5b: ใส่ SD card เข้ากับเครื่องคอมพิวเตอร์หรือ card reader

Step 6b: เปิดโปรแกรม Win32 Disk Imager

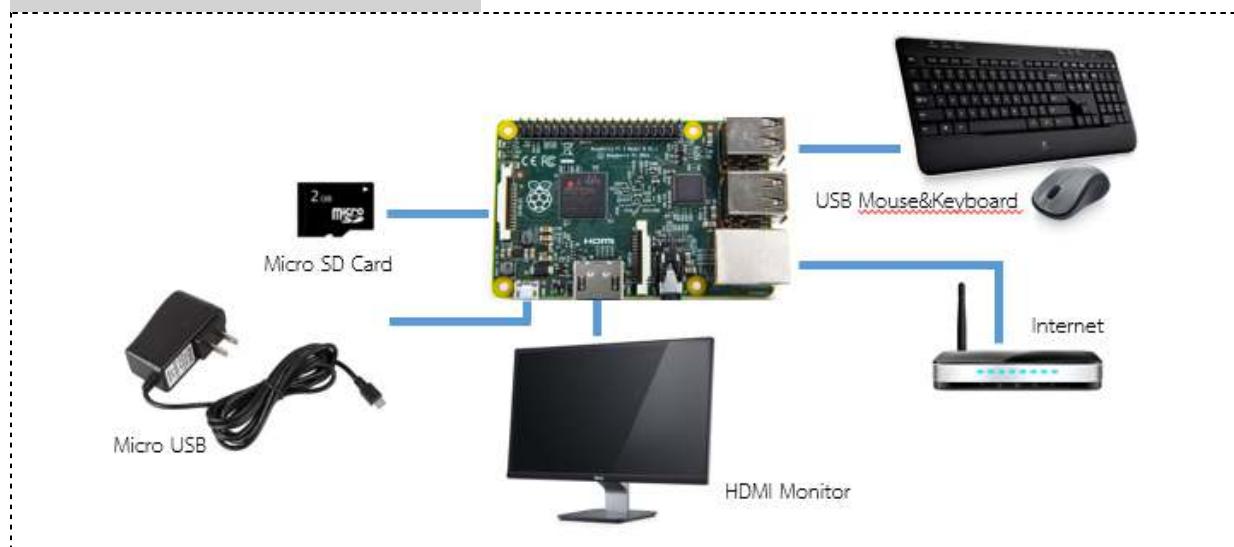
Step 7b: เลือก Dive เป็นไดรฟ์ของ SD card (1), เลือก Image File (2), Write (3)



Step 8b: เมื่อติดตั้งเรียบร้อยแล้วให้ถอน Micro SD card ออกจากคอมพิวเตอร์แล้วนำไปที่ Raspberry Pi ต่อไป



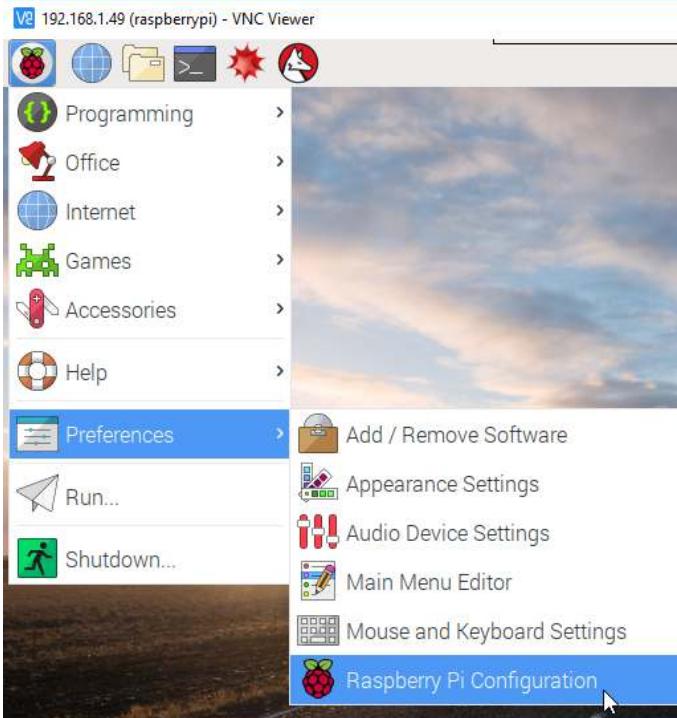
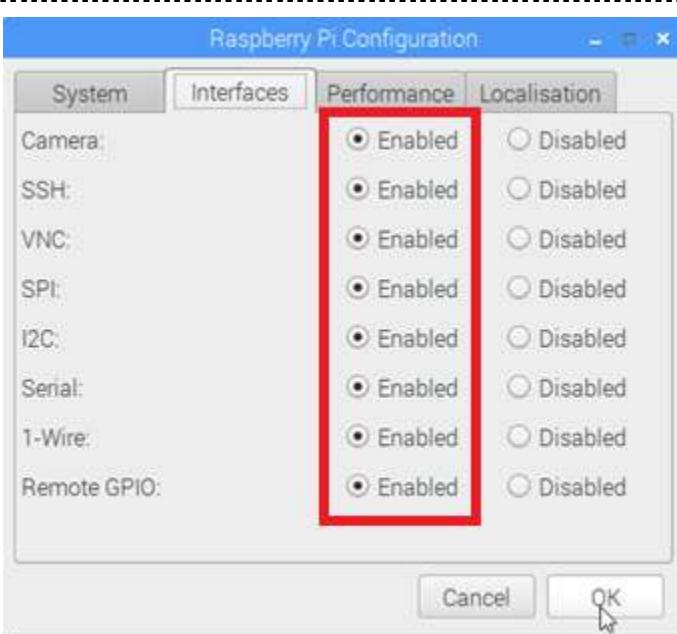
2.3 การเชื่อมต่อ เมื่อเริ่มต้นใช้งานครั้งแรก



Step 1: จ่ายไฟให้ Raspberry Pi ตั้งค่า Remote Password = **raspberry**

Step 2: กำหนดค่าครั้งแรก เพื่อให้สามารถใช้งานแบบ Remote Desktop ได้

 Wireless & Wired Network Settings Add / Remove Panel Items Remove "Wireless & Wired Network" From Panel	Right Click ที่ลูกศรส่วนทาง เลือก Network Setting
 Interface = eth0 <input type="checkbox"/> Automatically configure empty options <input checked="" type="checkbox"/> Disable IPv6 IPv4 Address: 192.168.1.49 IPv6 Address: Router: DNS Servers: DNS Search: Clear Apply Close	<input type="checkbox"/> Automatically <input checked="" type="checkbox"/> Disable IPv6 192.168.1.xx Apply, and Close

	<p>Raspberry → Preference → → RPi Configuration</p>																
	<p>Tab = Interface</p> <table border="0"> <tbody> <tr> <td>Camera</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>SSH</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>VNC</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>SPI</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>I2C</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>Serial</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>1-Wire</td> <td><input checked="" type="radio"/> Enable</td> </tr> <tr> <td>Remote GPIO</td> <td><input checked="" type="radio"/> Enable</td> </tr> </tbody> </table> <p>OK</p>	Camera	<input checked="" type="radio"/> Enable	SSH	<input checked="" type="radio"/> Enable	VNC	<input checked="" type="radio"/> Enable	SPI	<input checked="" type="radio"/> Enable	I2C	<input checked="" type="radio"/> Enable	Serial	<input checked="" type="radio"/> Enable	1-Wire	<input checked="" type="radio"/> Enable	Remote GPIO	<input checked="" type="radio"/> Enable
Camera	<input checked="" type="radio"/> Enable																
SSH	<input checked="" type="radio"/> Enable																
VNC	<input checked="" type="radio"/> Enable																
SPI	<input checked="" type="radio"/> Enable																
I2C	<input checked="" type="radio"/> Enable																
Serial	<input checked="" type="radio"/> Enable																
1-Wire	<input checked="" type="radio"/> Enable																
Remote GPIO	<input checked="" type="radio"/> Enable																

Step 3: กำหนดขนาดหน้าจอเชื่อมต่อ

RPI → Preference → Screen Configure เลือก 1024 x 768, กด เพื่อ Apply

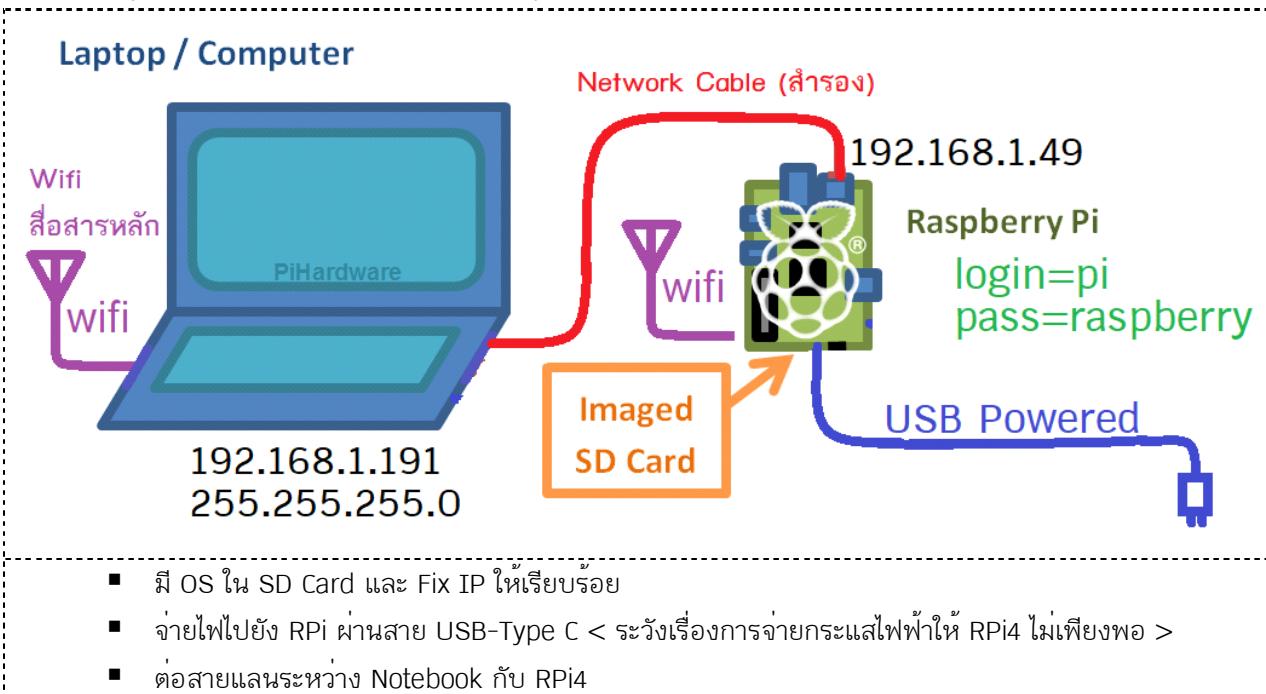


Step 4: Restart Raspberry Pi

- หลังจากนี้จะทำงานแบบ Remote Desktop ซึ่งไม่จำเป็นต้องมี Monitor, Mouse, Keyboard ต่อที่ Raspberry Pi

2.4 การเชื่อมต่อแบบ REMOTE DESKTOP

Step 1: การใช้งานแบบ Remote Desktop



Step 2: Fix IP ที่ Notebook

Apps and Features
Mobility Center
Power Options
Event Viewer
System
Device Manager
Network Connections
Disk Management

→ Win , Right Click
→ Network Connection

Network & Internet

Wi-Fi
Ethernet

Ethernet

Network and Sharing Center

HomeGroup

Network and Sharing Center

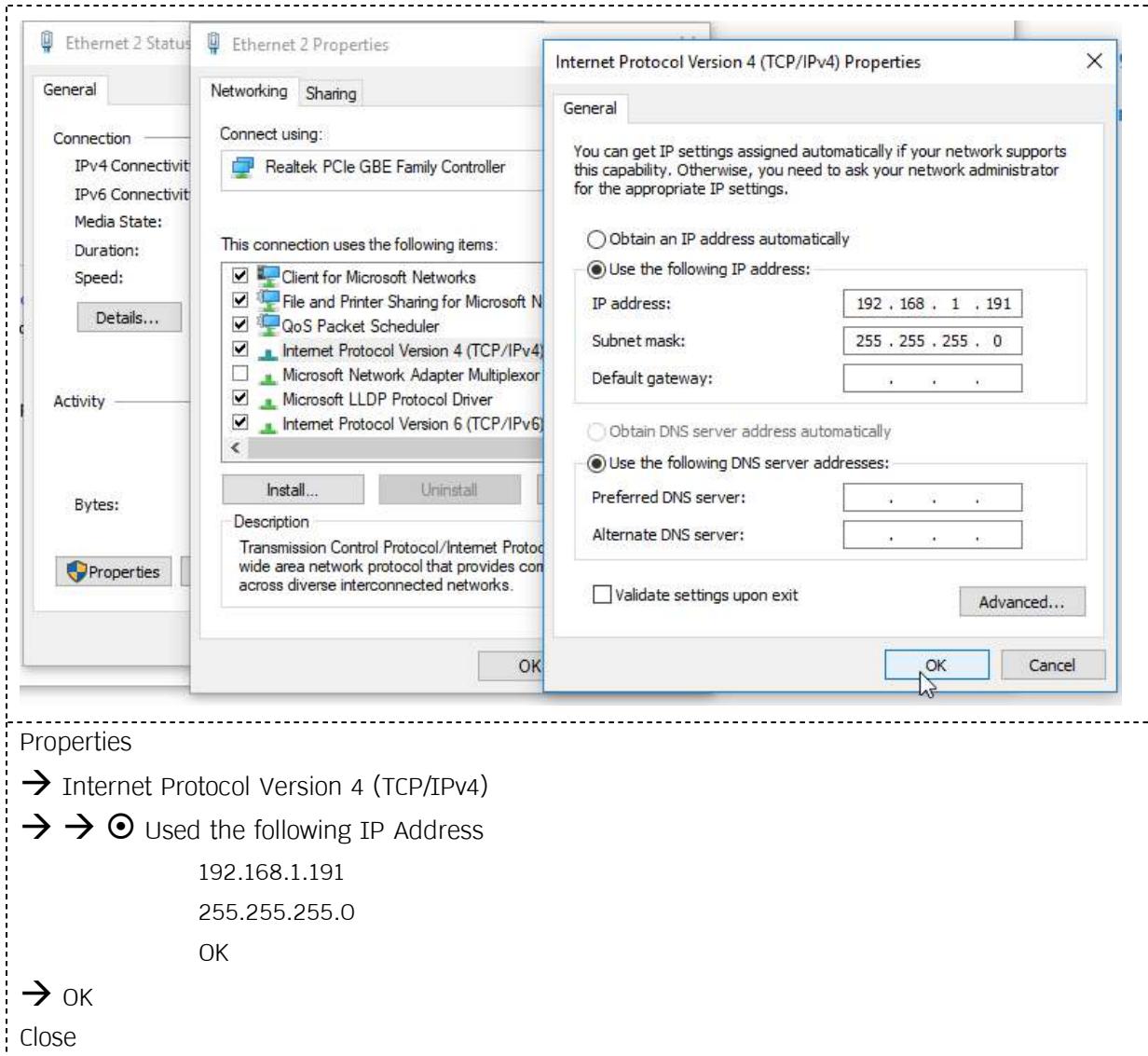
View your basic network information and set up connections

View your active networks

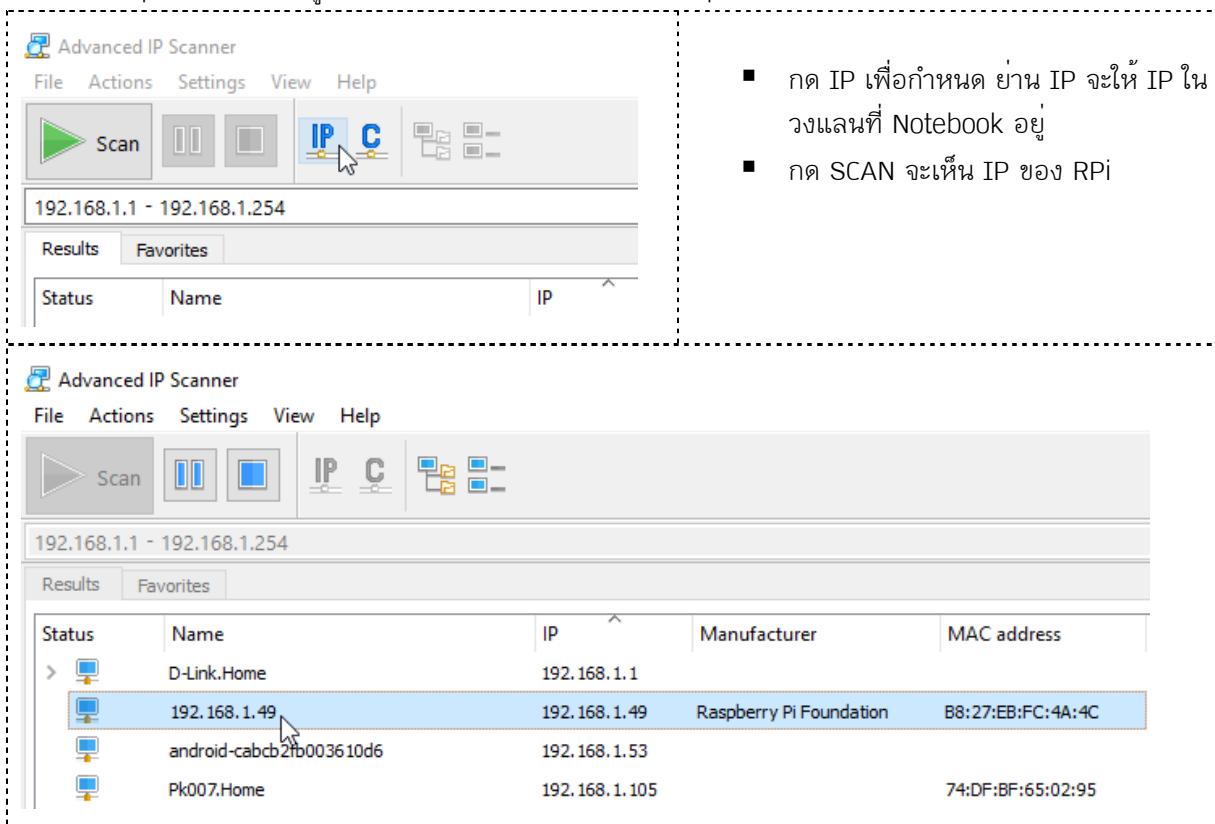
testVirus Private network	Access type: Internet HomeGroup: Ready to create Connections: Wi-Fi (testVirus)	Ethernet 2
Unidentified network Public network	Access type: No network access Connections: Ethernet 2	

Change your networking settings

- Set up a new connection or network**
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.
- Troubleshoot problems**
Diagnose and repair network problems, or get troubleshooting information.



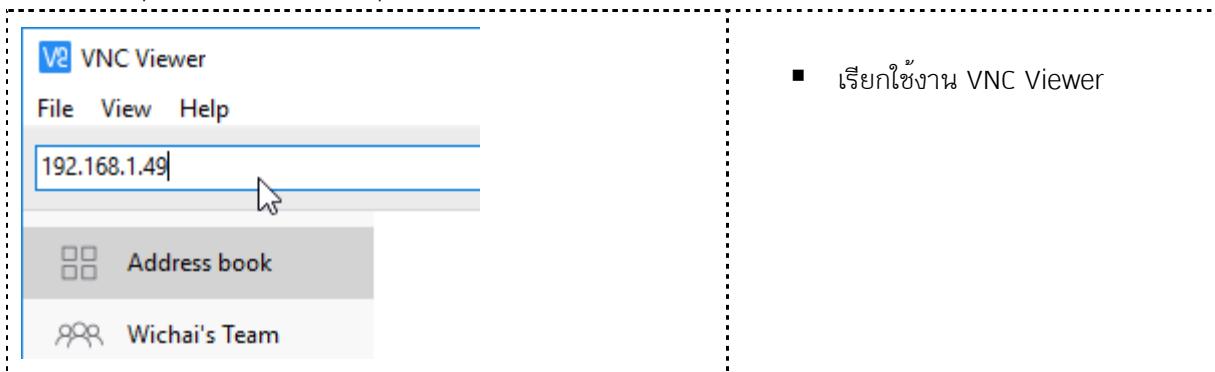
Step 3: ตรวจสอบลูกข่ายที่มีในเครื่องข่าย ด้วยโปรแกรม ipscan



The screenshot shows two instances of the Advanced IP Scanner software. The top instance displays a scan range from 192.168.1.1 to 192.168.1.254. The bottom instance shows a more detailed view of the results table, listing four devices: D-Link.Home (IP 192.168.1.1), 192.168.1.49 (IP 192.168.1.49, highlighted with a blue selection bar), android-cabcb2fb003610d6 (IP 192.168.1.53), and Pk007.Home (IP 192.168.1.105).

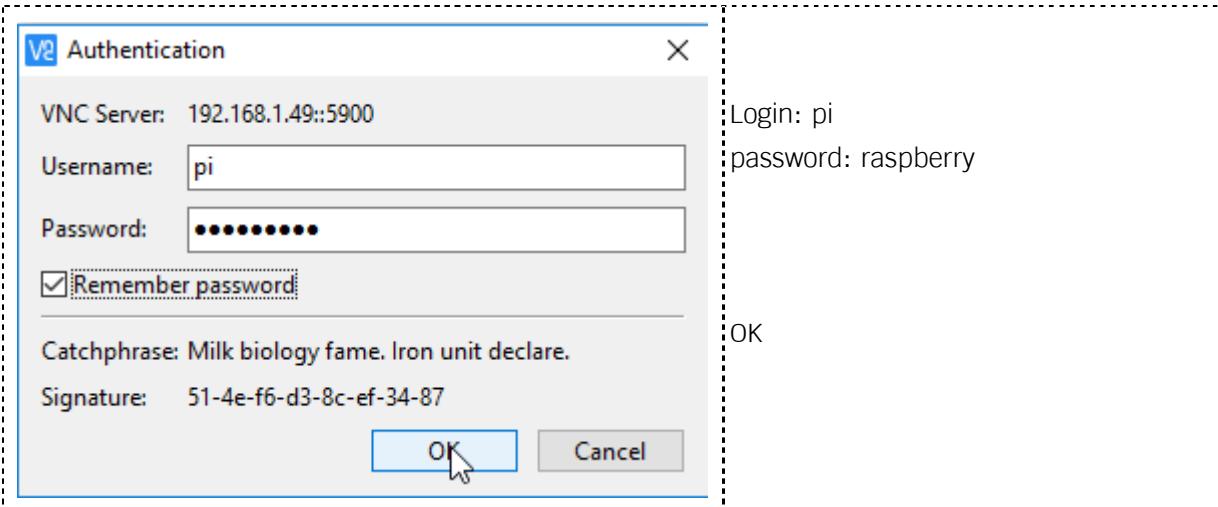
- กด IP เพื่อกำหนด ยาน IP จะให้ IP ในวงแ吝ที่ Notebook อุป
- กด SCAN จะเห็น IP ของ RPi

Step 4: Remote Desktop ด้วยโปรแกรม VNC Viewer



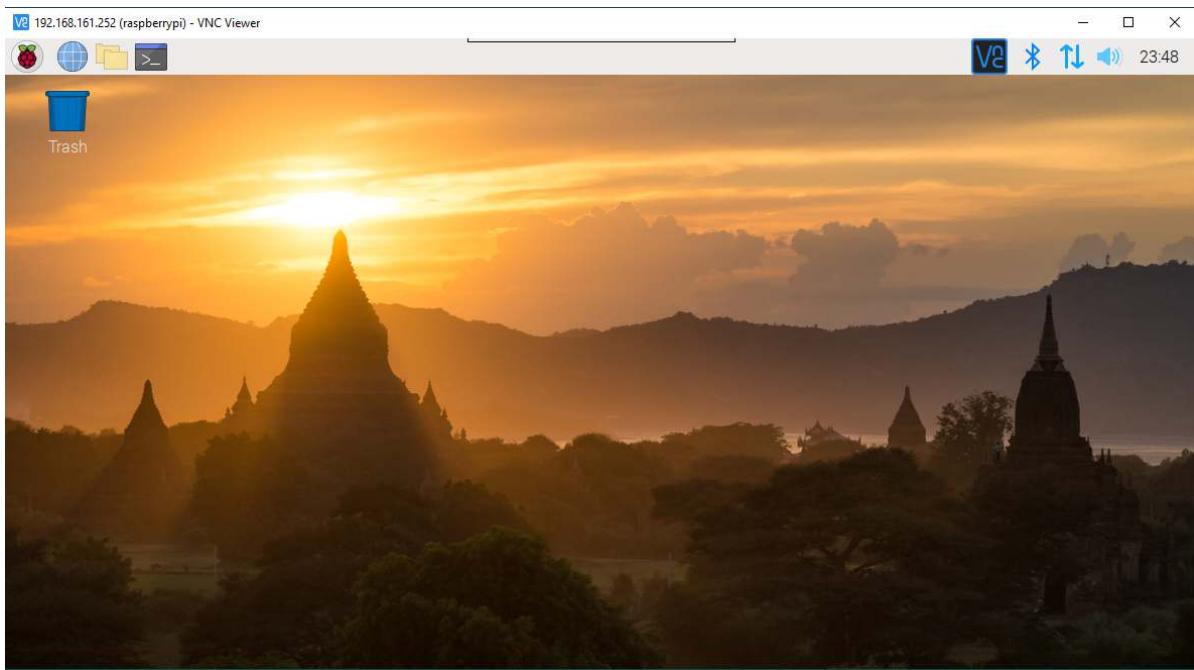
The screenshot shows the VNC Viewer application interface. In the address bar, the IP address 192.168.1.49 is entered. To the right, a list of connections is shown, with 'Wichai's Team' selected.

- เรียกใช้งาน VNC Viewer

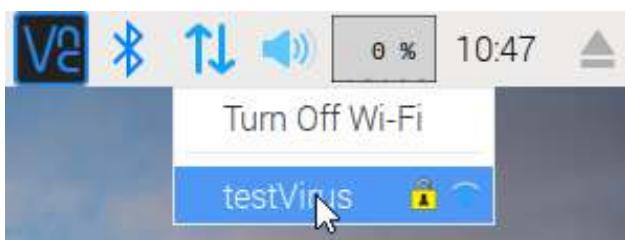


```
Login: pi  
password: raspberry
```

:ok



- พร้อมใช้งาน
 - สามารถตั้งรายละเอียด ขนาดหน้าจอ ได้ที่ Preference → Screen Configure



จำเป็นต้องตั้งการเชื่อมต่อ Wifi เพื่อให้ RPi เชื่อมต่ออินเตอร์เน็ต

2.5 โปรแกรมที่จำเป็นต้องใช้งาน



2.6 สิ่งที่ควรทำหลังจากการ BOOT ครั้งแรก

- อัพเดตโปรแกรมให้ใหม่ล่าสุดด้วย sudo apt-get update
เข้า Terminal และพิมพ์คำสั่ง Update พิม์เสร็จแล้วกด Enter เพื่อทำการอัพเดต

sudo apt-get update

```
pi@raspberrypi ~
File Edit Tabs Help
pi@raspberrypi ~ $ sudo apt-get update
Get:4 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Get:5 http://archive.raspbian.org wheezy Release.gpg [490 B]
Get:6 http://raspberrypi.collabora.com wheezy/rpi armhf Packages [2,214 B]
Hit http://mirrordirector.raspbian.org wheezy/main armhf Packages
Hit http://archive.raspbian.org wheezy Release
Get:7 http://mirrordirector.raspbian.org wheezy/contrib armhf Packages [23.6 kB]
Hit http://mirrordirector.raspbian.org wheezy/non-free armhf Packages
Hit http://archive.raspbian.org wheezy/main armhf Packages
Hit http://mirrordirector.raspbian.org wheezy/rpi armhf Packages
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
Ign http://archive.raspbian.org wheezy/main Translation-en_GB
Ign http://archive.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 49.6 kB in 35s (1,404 B/s)
Reading package lists... Done
pi@raspberrypi ~ $
```

- อัพเดตโปรแกรมให้ใหม่ล่าสุดด้วย sudo apt-get upgrade
เข้า Terminal และพิมพ์คำสั่ง, พิม์เสร็จแล้วกด Enter เพื่อทำการอัพเดต

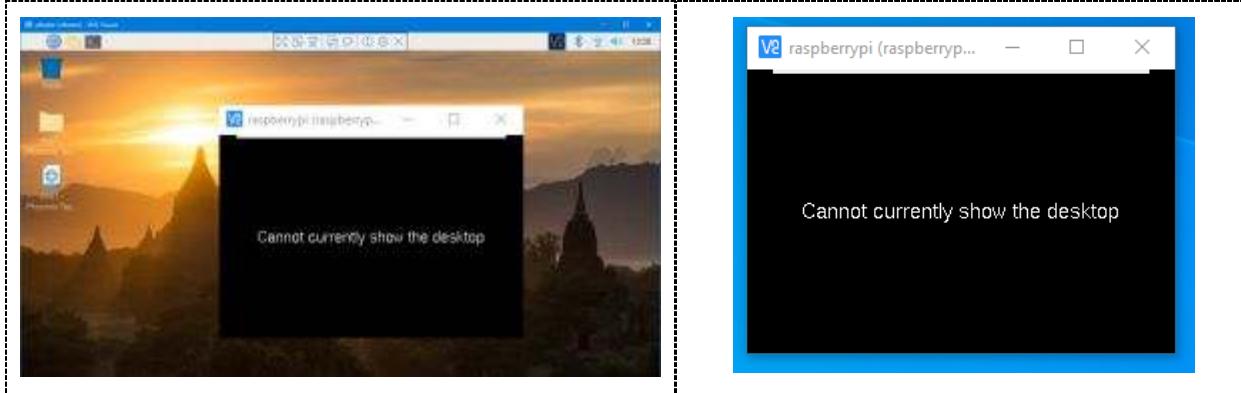
sudo apt-get upgrade

แล้วกด Y และ Enter เพื่อยืนยัน (ขั้นตอนนี้จะนานมาก)

2.7 How to Fix Raspberry Pi's 'Cannot Currently Show the Desktop' Error

<https://www.tomshardware.com/how-to/fix-cannot-currently-show-desktop-error-raspberry-pi>

You get this error when trying to connect via VNC.



It's a truly frustrating error you may encounter when you try to connect to your [Raspberry Pi](#) remotely using VNC. After you've done a [Raspberry Pi headless install](#), you try getting remote access to the desktop environment, but VNC viewer (or some other client) gives you a black screen and with the message "Cannot currently show the desktop" on it.

There are a couple of possible solutions for the "Cannot currently show the desktop" error message when connecting to your Raspberry Pi. Try each and see if it resolves the issue. All of the solutions below involve using the command prompt, either by directly connecting a keyboard and mouse to your Raspberry Pi or by using a remote SSH connection to get to the terminal. Massive hat tip to the folks in this [forum thread](#) who offered solutions and inspired this article.

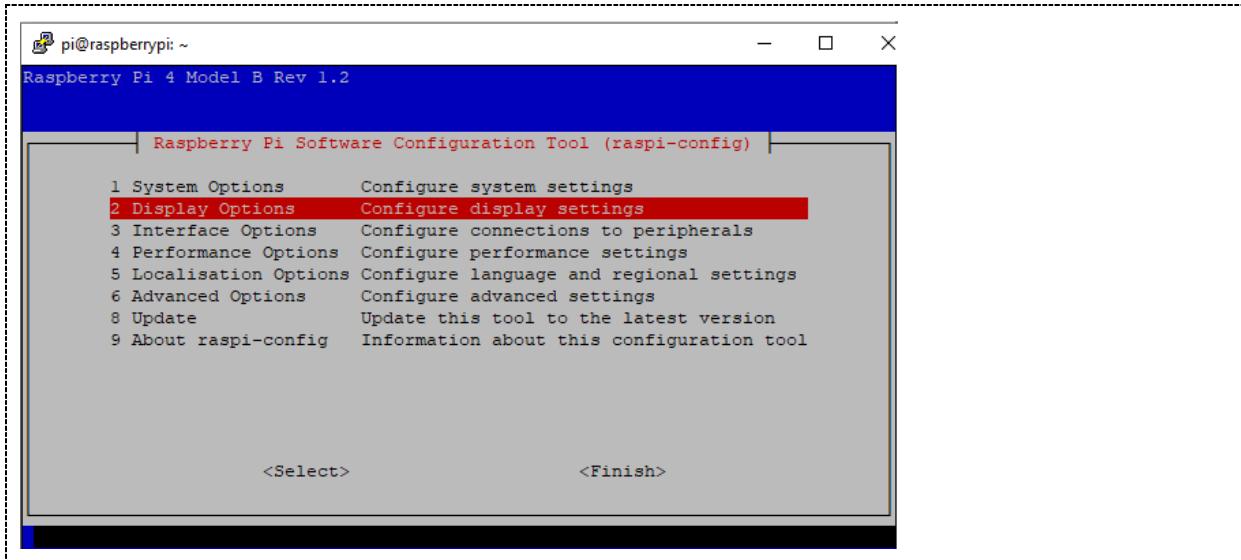
Change Your Screen Resolution

Even though it should have a working default setting, often changing resolution at the command prompt will solve the “Cannot currently show the desktop” error. To do this:

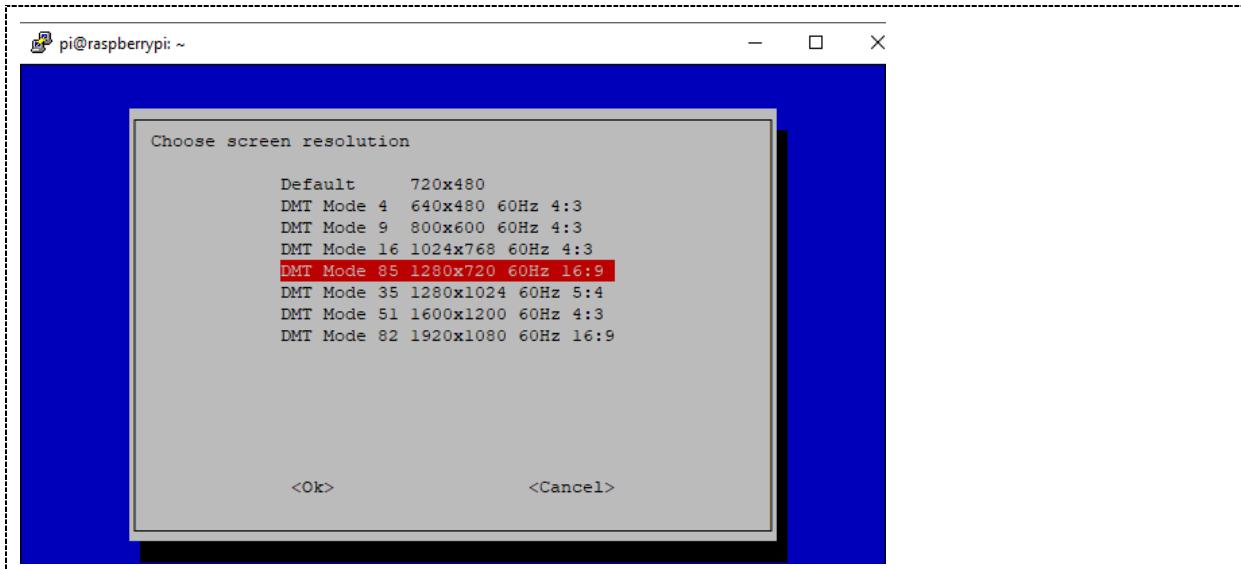
1. Launch raspi-config.

```
sudo raspi-config
```

2. Select Display Options



3. Choose a resolution. We recommend at least 1280 x 720, but some users report that choosing the highest possible (usually 1920 x 1080) is what they needed to do.



4. You'll need to reboot your Raspberry Pi for this change to take effect.

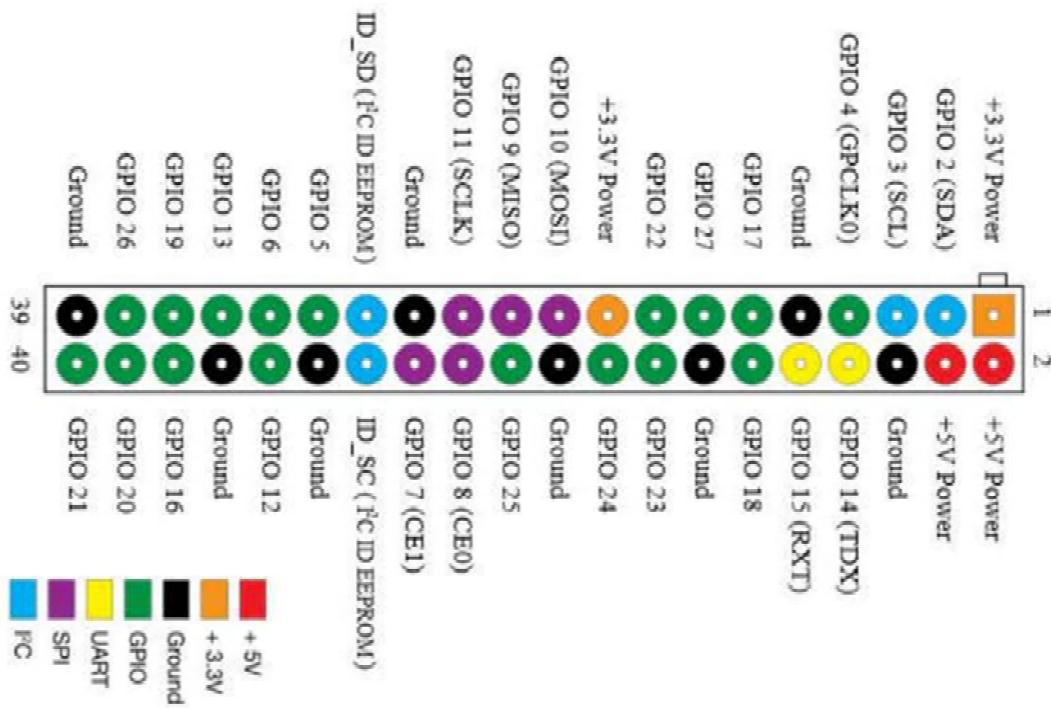
3/6 – การโปรแกรมเพื่อควบคุม I/O ด้วย Python

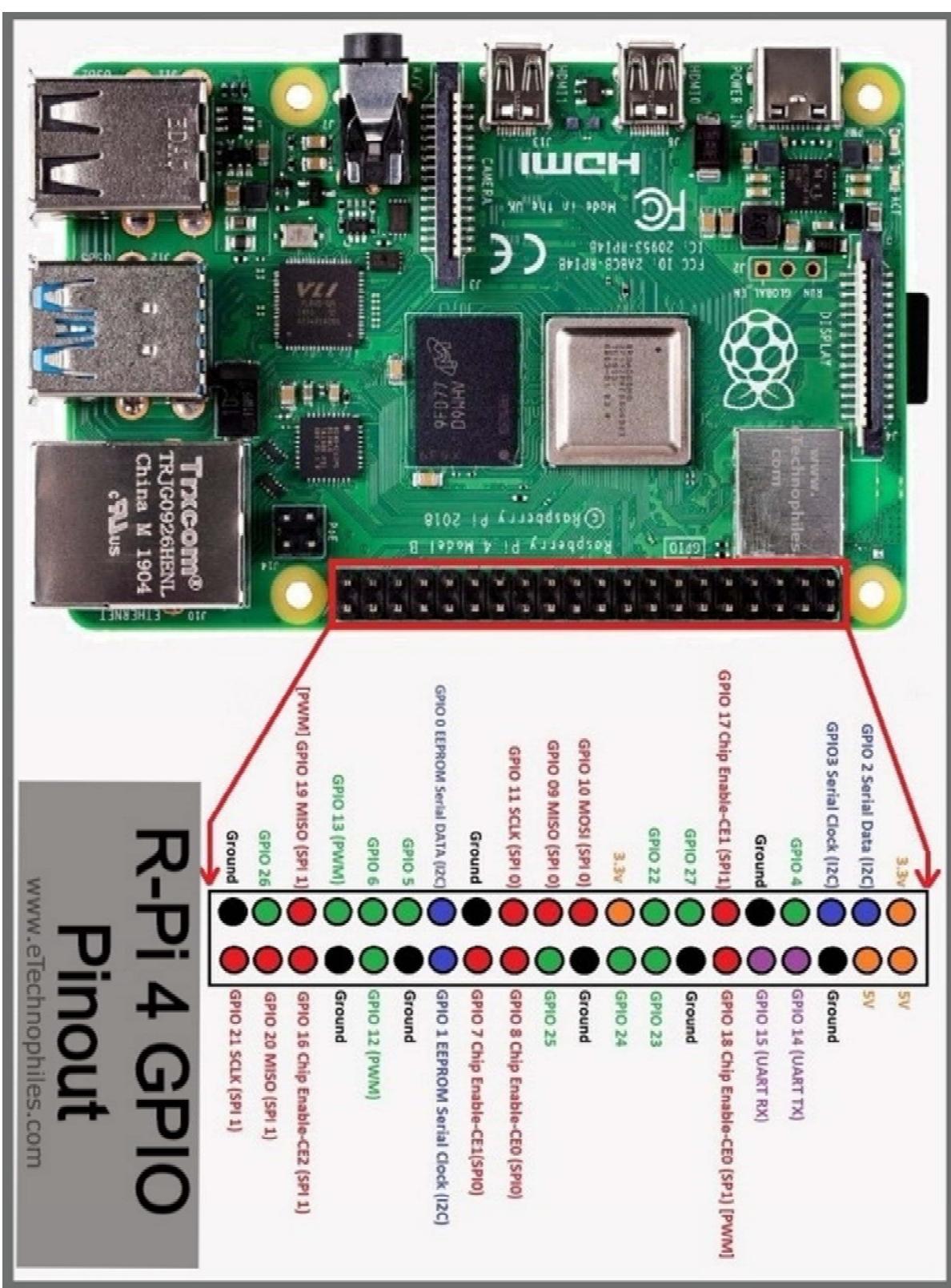
ในตอนนี้เราจะทดลองใช้ GPIO ของบอร์ด Raspberry Pi สั่งเปิด-ปิดไฟ LED และการกดปุ่มเพื่อรับอินพุต เป็นการเรียนรู้การใช้งาน GPIO เป็นต้นๆครับ โดยเราจะเขียนด้วย python

RPi4 และ GPIO Pin



Peripherals	GPIO	Particle	Pin #	Pin #	Particle	GPIO	Peripherals	
3.3V			1	X	5V			
I2C	GPIO2	SDA	3	X	5V			
	GPIO3	SCL	5	X	GND			
Digital I/O	GPIO4	DO	7	X	X	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15 Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18 PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND	
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23 Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24 Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND	
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25 Digital I/O
	GPIO11	SCK	23	X	X	24	CEO	GPIO8 SPI (chip enable)
GND			25	X	X	26	CE1	GPIO7
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND	
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12 Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND	
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16 PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20 Digital I/O
GND			39	X	X	40	D16/A7	GPIO21 Digital I/O





- สามารถเรียกดูขาได้ด้วยคำสั่ง pinout

```
Chris1 — pi@raspberrypi: ~ — ssh pi@192.168.0.51 — 94x50
pi@raspberrypi:~ $ pinout
[Diagram of Raspberry Pi Model 4B Pinout]
Revision      : c03111
SoC          : BCM2711
RAM          : 4096Mb
Storage       : MicroSD
USB ports     : 4 (excluding power)
Ethernet ports : 1
Wi-fi         : True
Bluetooth     : True
Camera ports (CSI) : 1
Display ports (DSI): 1

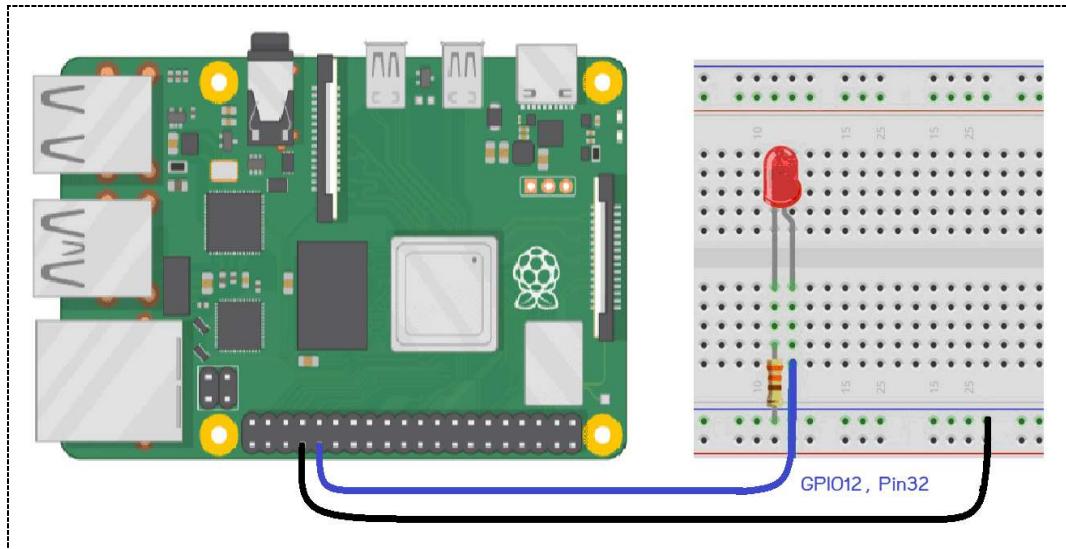
J8:
  3V3  (1) (2)  5V
  GPIO2 (3) (4)  5V
  GPIO3 (5) (6)  GND
  GPIO4 (7) (8)  GPIO14
  GND   (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
  3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
  GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO08
  GND (25) (26) GPIO07
  GPIO08 (27) (28) GPIO01
  GPIO05 (29) (30) GND
  GPIO06 (31) (32) GPIO012
  GPIO13 (33) (34) GND
  GPIO19 (35) (36) GPIO016
  GPIO26 (37) (38) GPIO020
  GND (39) (40) GPIO021

For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~ $
```

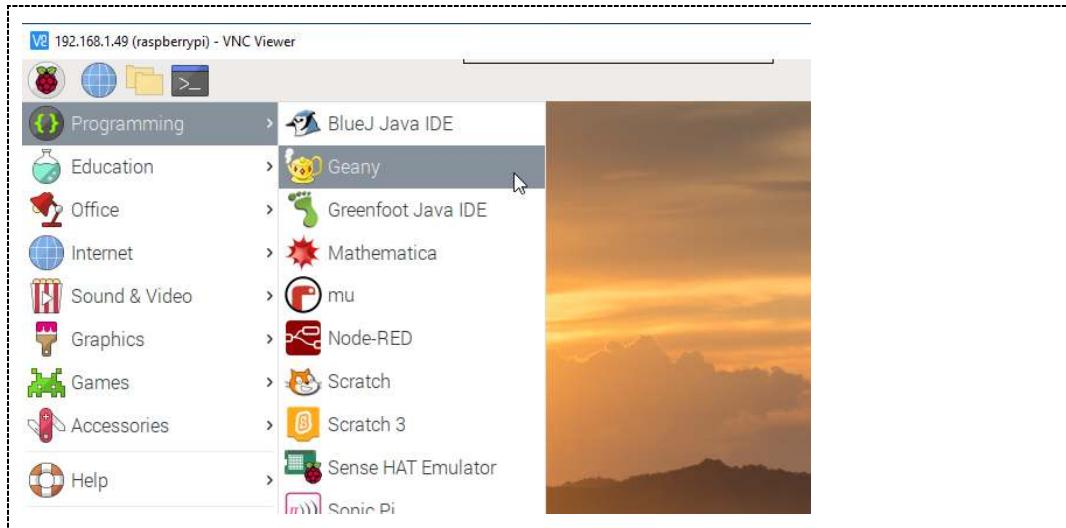
Lab102 – GPIO Output

1. การต่อวงจร

- ใช้ขา 34 (GND) ต่อตัวด้านท่าน 330 โอมห์ และต่อขาลับของ LED
- ใช้ขา 32 (GPIO12) ต่อเข้ากับขาบวกของ LED



2. เข้าโปรแกรม Menu > Programming > Geany



3. ทดสอบโปรแกรม_1 การอ้างอิงขาแบบ BOARD = Pin.32

```

import RPi.GPIO as GPIO          #Add GPIO library to a Python sketch
import time                     #Add time library to a Python sketch
LED_pin = 32                    # Ref Board

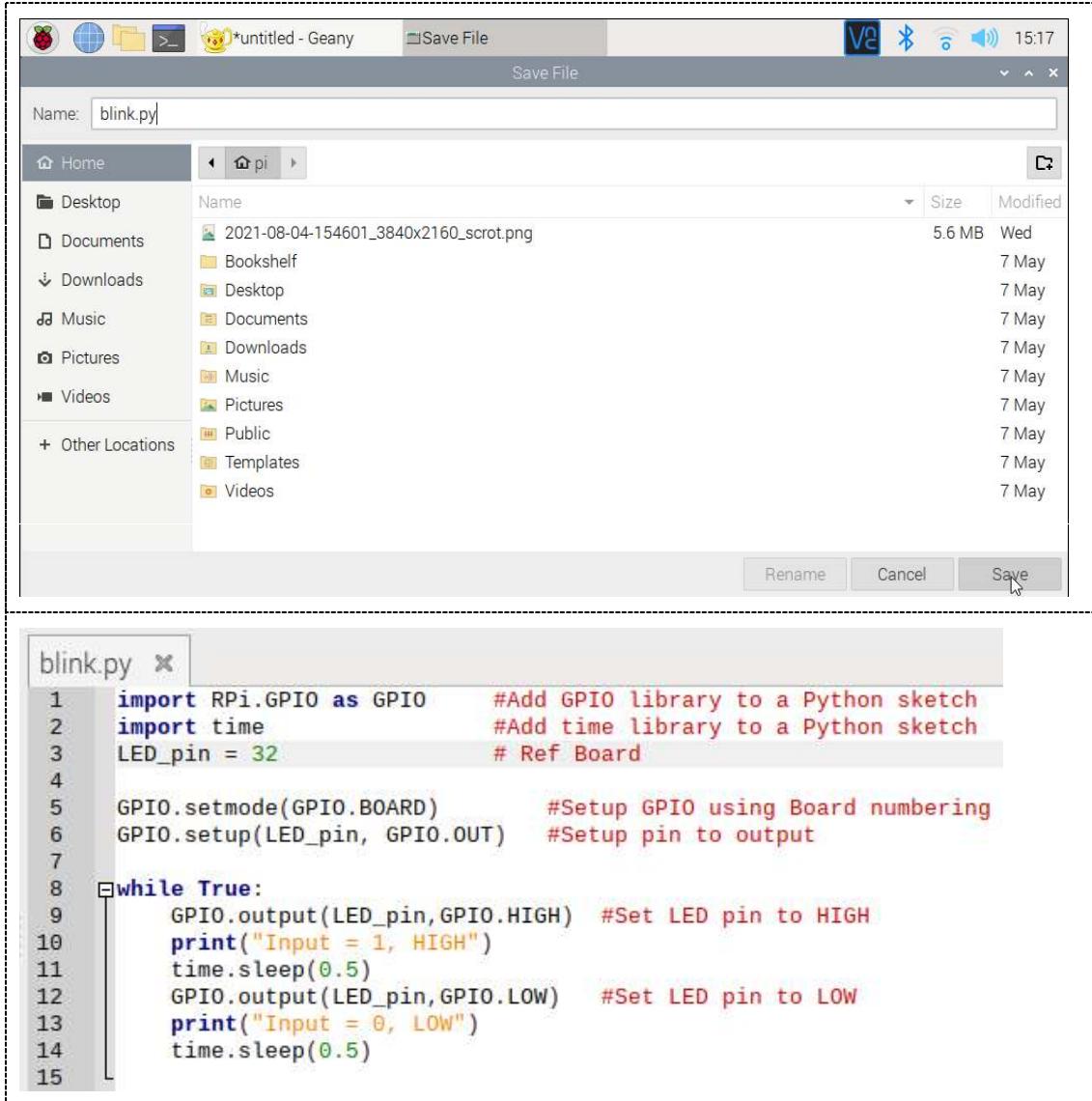
GPIO.setmode(GPIO.BOARD)        #Setup GPIO using Board numbering
GPIO.setup(LED_pin, GPIO.OUT)    #Setup pin to output

while True:
    GPIO.output(LED_pin,GPIO.HIGH)      #Set LED pin to HIGH
    print("Input = 1, HIGH")
    time.sleep(0.5)

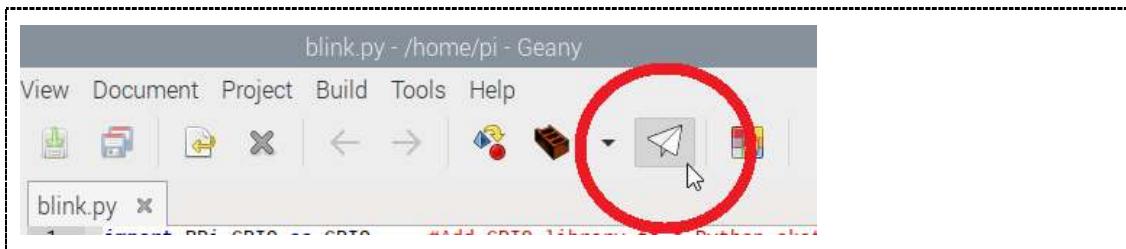
    GPIO.output(LED_pin,GPIO.LOW)       #Set LED pin to LOW
    print("Input = 0, LOW")
    time.sleep(0.5)

```

4. เข้า File > Save As, บันทึกไฟล์ชื่อว่า blink.py ไว้ที่ /home/pi (ตำแหน่ง Default)



5. การทดสอบแบบ A -- กดรัน



6. การทดสอบแบบ B -- รันผ่าน python shell

- เข้าที่ Terminal พิมพ์คำสั่ง

```
sudo python blink.py
```

A terminal window titled "pi@raspberrypi: ~". The command "sudo python blink.py" is run, followed by the script's output:

```
pi@raspberrypi:~ $ dir
2021-08-04-154601_3840x2160_scrot.png  Desktop   Music    Templates
blink.py                                Documents  Pictures  Videos
Bookshelf                               Downloads  Public

pi@raspberrypi:~ $ python3 blink.py
blink.py:5: RuntimeWarning: This channel is already in use, continuing anyway.
Use GPIO.setwarnings(False) to disable warnings.
    GPIO.setup(32, GPIO.OUT)                  #Setup pin 32 to output
Input = 1, HIGH
Input = 0, LOW
Input = 1, HIGH
Input = 0, LOW
```

- จะเห็นว่าไฟ LED จะติดตับครึ่งละ 1 วินาที
- เมื่อต้องการออกให้กด CTRL+Z เพื่อยกเลิกการทำงานของโค้ด

7. ทดสอบโปรแกรม_2 การอ้างอิงขาแบบ BCM = GPIO.12

```
IMPORT RPI.GPIO AS GPIO          #ADD GPIO LIBRARY TO A PYTHON SKETCH
IMPORT TIME                      #ADD TIME LIBRARY TO A PYTHON SKETCH
LED_PIN = 12                     # REF GPIO NAME

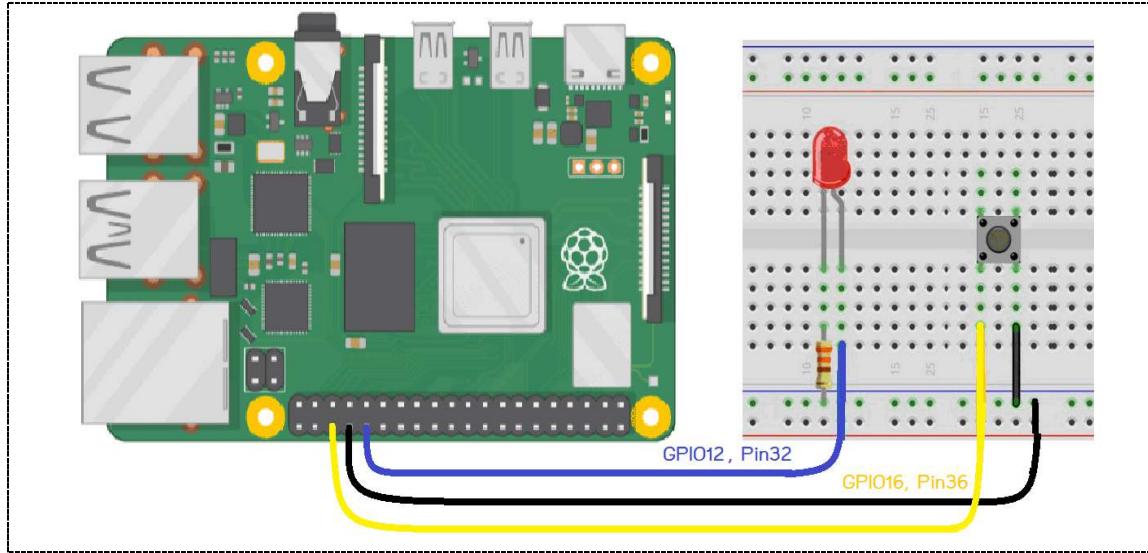
GPIO.SETMODE(GPIO.BCM)           #SETUP GPIO USING GPIO.PIN
GPIO.SETUP(LED_PIN, GPIO.OUT)     #SETUP PIN TO OUTPUT

WHILE TRUE:
    GPIO.OUTPUT(LED_PIN,GPIO.HIGH)  #SET LED PIN TO HIGH
    PRINT("INPUT = 1, HIGH")
    TIME.SLEEP(0.5)

    GPIO.OUTPUT(LED_PIN,GPIO.LOW)   #SET LED PIN TO LOW
    PRINT("INPUT = 0, LOW")
    TIME.SLEEP(0.5)
```

Lab103 – GPIO Input, Output

8. การทดลองนี้จะต่อปุ่มกดแบบ Pull-up โดยปกติไฟ LED จะติดค้าง เมื่อกดปุ่มไฟ LED จะดับ
9. การต่อวงจร LED=Board32, GND=Board34, SW=Board36, แล้วทดสอบโปรแกรม_3



```

import RPi.GPIO as GPIO          # Add GPIO library to a Python sketch
import time                     # Add time library to a Python sketch
LED_pin = 32                    # Ref Board
SW_Pin = 36

GPIO.setmode(GPIO.BOARD)         #Setup GPIO using GPIO.Pin
GPIO.setup(LED_pin, GPIO.OUT)    #Setup pin to output
GPIO.setup(SW_Pin, GPIO.IN, pull_up_down = GPIO.PUD_UP)
                           #Setup pin to input and Pull-Up

while True:
    if (GPIO.input(SW_Pin)==1):   # Read Button pin
        GPIO.output(LED_pin,GPIO.HIGH) # Set LED pin to HIGH
        print("Input = 1, HIGH")
    else:
        GPIO.output(LED_pin,GPIO.LOW) # Set LED pin to LOW
        print("Input = 0, LOW")
    time.sleep(0.5)

```

```

InputOutput.py x
1 import RPi.GPIO as GPIO      #Add GPIO library to a Python sketch
2 import time                  #Add time library to a Python sketch
3 LED_pin = 32                # Ref Board
4 SW_Pin = 36
5
6 GPIO.setmode(GPIO.BOARD)     #Setup GPIO using GPIO.Pin
7 GPIO.setup(LED_pin, GPIO.OUT) #Setup pin to output
8 GPIO.setup(SW_Pin, GPIO.IN, pull_up_down = GPIO.PUD_UP)
                           #Setup pin to input and Pull-Up
9
10
11 while True:
12     if (GPIO.input(SW_Pin)==1): #Read Button pin
13         GPIO.output(LED_pin,GPIO.HIGH) #Set LED pin to HIGH
14         print("Input = 1, HIGH")
15     else:
16         GPIO.output(LED_pin,GPIO.LOW) #Set LED pin to LOW
17         print("Input = 0, LOW")
18     time.sleep(0.5)
19

```

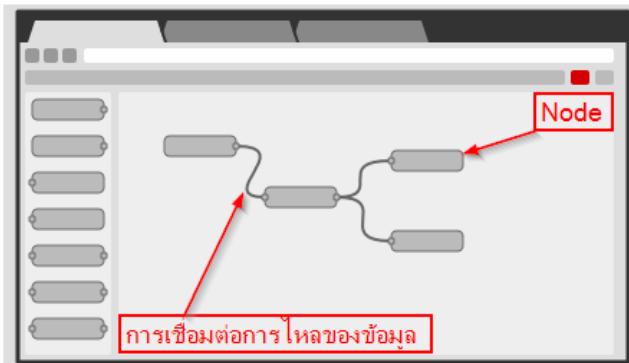
คำถาม Quiz-101

1. จากโปรแกรมข้อ 9 หากบรรทัดที่ 8 เป็น `GPIO.setup(SW_Pin, GPIO.IN)` ผลการทำงานจะเป็นอย่างไร
2. อยากรู้แบบไหน กดติด ปล่อยดับ จะปรับแก้โปรแกรมอย่างไร
3. อยากรู้แบบไหน กดติด กดดับ จะปรับแก้โปรแกรมอย่างไร
4. อยากรู้แบบไหน จำนวนครั้งที่กดสวิตช์

4/6 – การโปรแกรมเพื่อควบคุม I/O ด้วย Node-RED

Lab104 – Node-RED on Raspberry Pi4

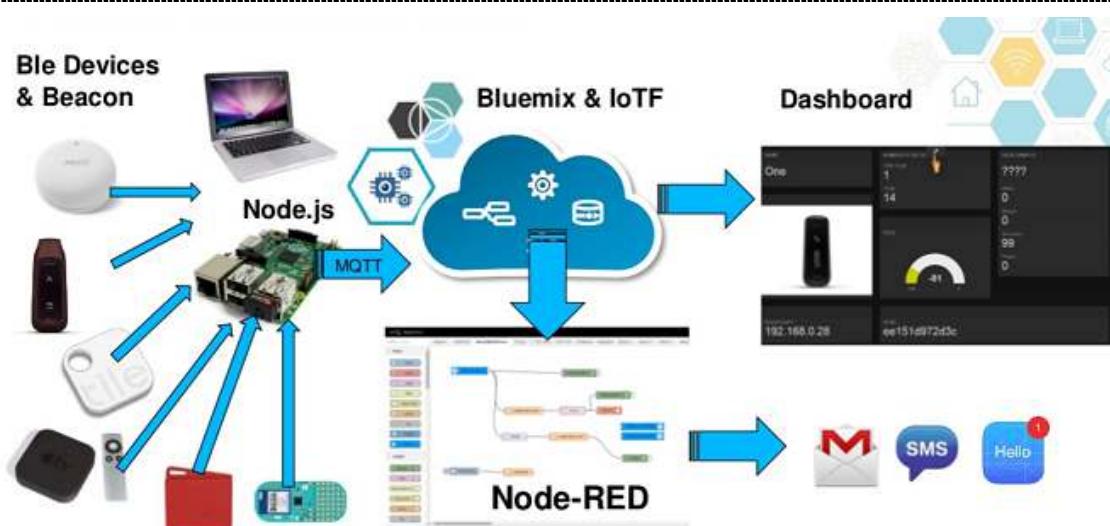
Node-RED เป็นเครื่องมือสำหรับนักพัฒนาโปรแกรมในการเขียนต่ออุปกรณ์ hardware เข้ากับ APIs (Application Programming Interface) ซึ่งเป็นการพัฒนาโปรแกรมแบบ Flow-Based Programming ที่มีหน้า UI สำหรับนักพัฒนาให้ใช้งานผ่าน Web Browser ทำให้การเขียนต่อส่วนทางการไฟล์ของข้อมูลนั้นเป็นเรื่องง่าย



เนื่องจาก Node-RED เป็น Flow-Based Programming นั้นทำให้เราแทบจะไม่ต้องเขียน Code ในการพัฒนาโปรแกรมเลย แค่เพียงเลือก Node มาวางแล้วเชื่อมต่อกันสามารถควบคุม I/O ได้ โดย Node-RED จะมี Node ให้เลือกใช้งานอย่างหลากหลาย

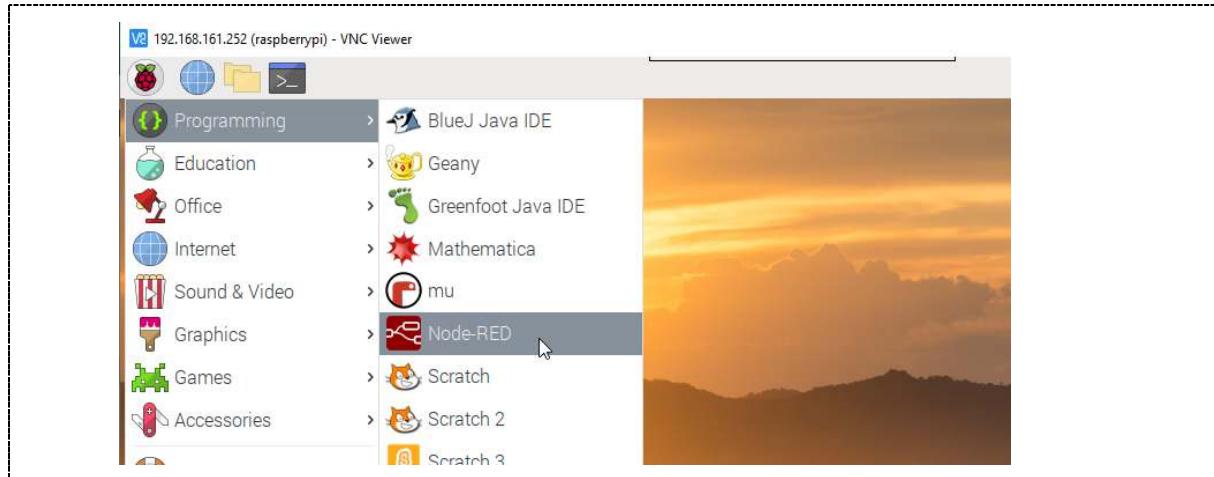
สามารถสร้างฟังก์ชัน JavaScript ได้โดยใช้ Text Editor ที่มีอยู่ใน Node-RED และยังสามารถบันทึก Function, Templates, Flows เพื่อไปใช้งานกับงานอื่นต่อไป

Node-RED นั้นทำงานบน Node.js ทำให้เหมาะสมสำหรับการใช้งานกับ Raspberry Pi เนื่องจากใช้ทรัพยากร้อย ขนาดไฟล์ไม่ใหญ่และ Node.js ยังทำหน้าที่เป็นตัวกลางให้ Raspberry Pi สามารถติดต่อกับ Web Browser และอุปกรณ์อื่นๆ ได้



เริ่มต้นการใช้งาน Node-RED

- ตรวจสอบว่า Raspberry Pi ของเรามี Node-RED หรือไม่ ซึ่ง image ตัวใหม่ จะมีมาพร้อมกับ Raspbian อุปกรณ์แล้ว



- วิธีตรวจสอบ พิมพ์คำสั่งเพื่อ RUN Node-RED ดังรูป
- หากมี Node-RED อยู่แล้ว ก็จะปรากฏข้อความดังรูป

```
pi@raspberrypi:~ $ node-red
```

The terminal window shows the output of the 'node-red' command. It starts with 'Welcome to Node-RED' followed by a series of log messages indicating the version, operating system, and configuration files being loaded. The server is then started at port 1880.

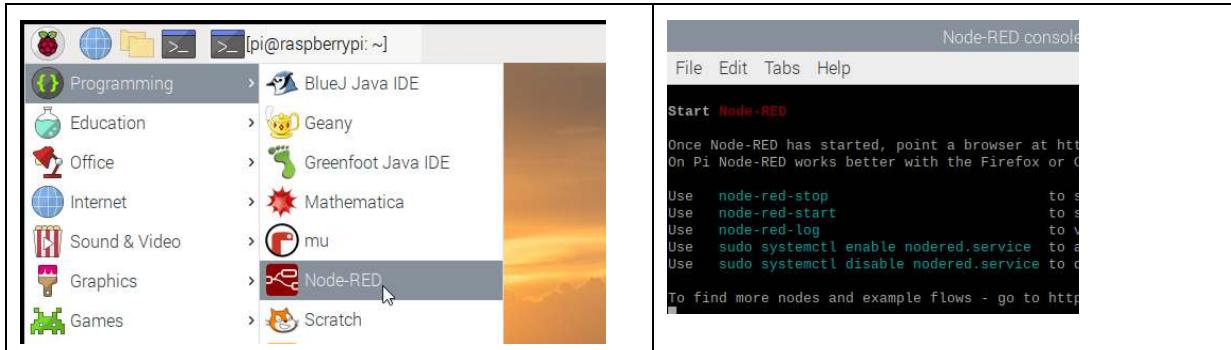
```

pi@raspberrypi:~ $ node-red

Welcome to Node-RED
-----
14 Jun 08:58:59 - [info] Node-RED version: v0.15.3
14 Jun 08:58:59 - [info] Node.js version: v0.10.29
14 Jun 08:59:00 - [info] Linux 4.4.50+ arm LE
14 Jun 08:59:06 - [info] Loading palette nodes
14 Jun 08:59:24 - [info] UI started at /ui
14 Jun 08:59:39 - [info] Settings file : /home/pi/.node-red/settings.js
14 Jun 08:59:39 - [info] User directory : /home/pi/.node-red
14 Jun 08:59:39 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
14 Jun 08:59:39 - [info] Server now running at http://127.0.0.1:1880/
14 Jun 08:59:39 - [info] Starting flows
14 Jun 08:59:40 - [info] Started flows

```

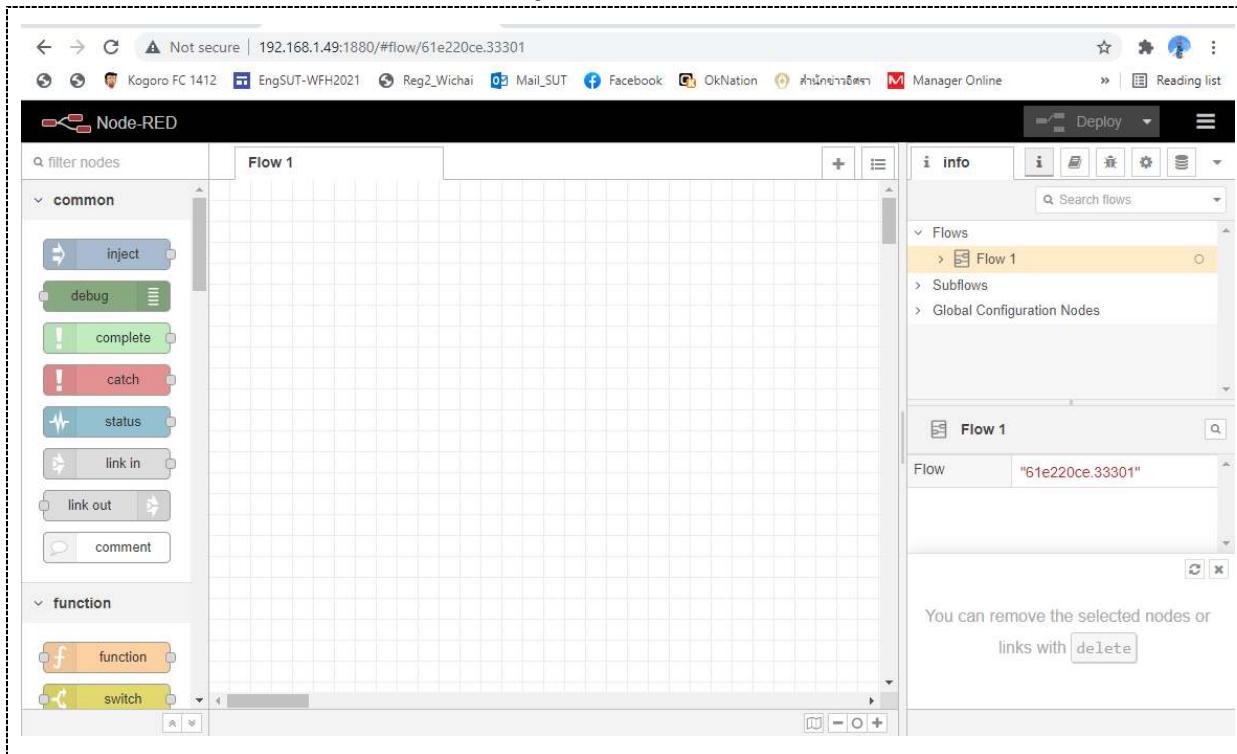
- Start Node-RED ที่ Raspberry Pi 4



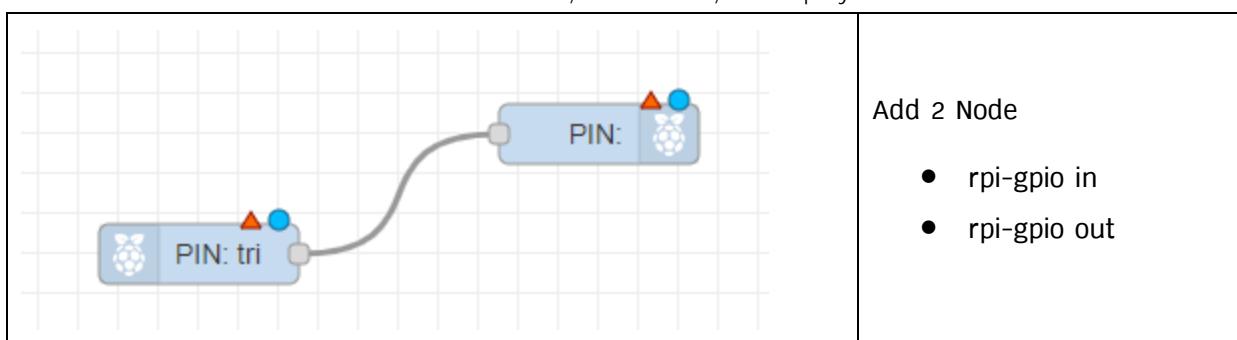
5. จากนั้นเปิด Web Browser ที่ PC และใส่ IP Address ของ Raspberry Pi ของเรา โดยใช้ port 1880 ดังรูป (ในกรณี ที่ทำการเชื่อมต่อ LAN และแชร์สัญญาณอินเตอร์เน็ตจาก Computer ให้ Raspberry Pi สามารถใช้ raspberrypi.mshome.net แทนหมายเลข IP Address ได้)



6. จะปรากฏเป็นหน้า UI ของ Node-RED ดังรูป

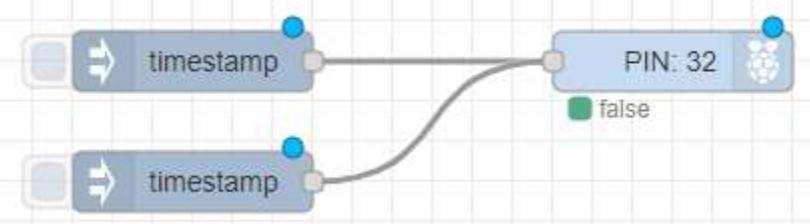
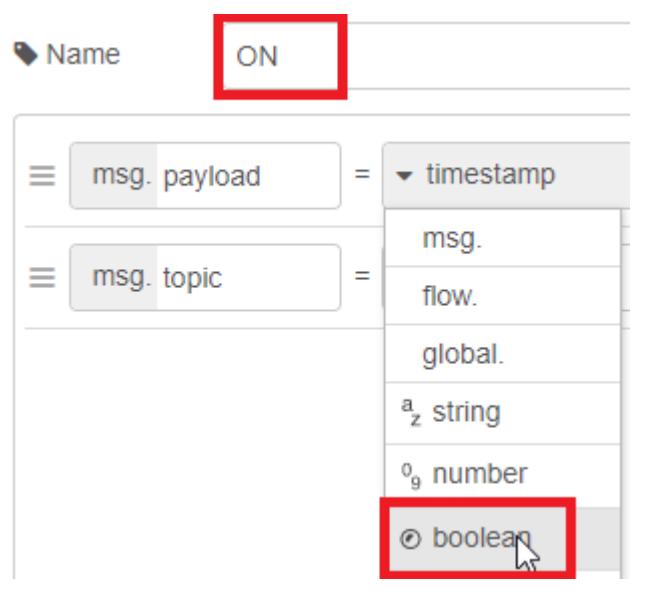
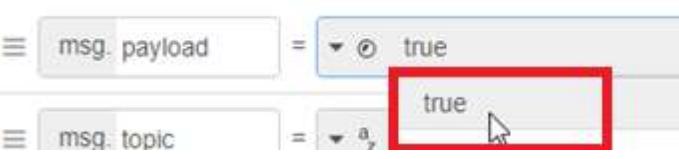
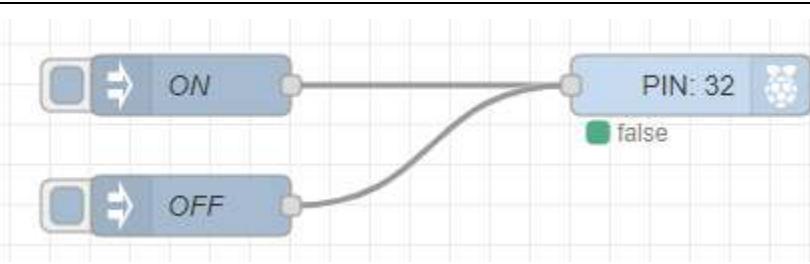
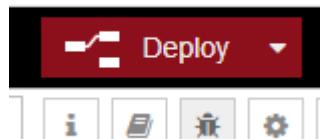


7. Read Pin 36 Send to Pin32 → add node, ตั้งค่า node, กด Deploy และลองกด Switch



<p>● Pin</p> <p>3.3V Power - 1 2 - 5V Power</p> <p>SD - 21 20 - 3G</p> <table border="1"> <tr><td>GPIO05 - 29</td><td>30 - Ground</td></tr> <tr><td>GPIO06 - 31</td><td>32 - GPIO12</td></tr> <tr><td>GPIO13 - 33</td><td>34 - Ground</td></tr> <tr><td>GPIO19 - 35</td><td>36 - GPIO16</td></tr> <tr><td>GPIO26 - 37</td><td>38 - GPIO20</td></tr> <tr><td>Ground - 39</td><td>40 - GPIO21</td></tr> </table> <p>32</p> <p>Type Digital output</p> <p><input type="checkbox"/> Initialise pin state?</p>	GPIO05 - 29	30 - Ground	GPIO06 - 31	32 - GPIO12	GPIO13 - 33	34 - Ground	GPIO19 - 35	36 - GPIO16	GPIO26 - 37	38 - GPIO20	Ground - 39	40 - GPIO21	<p>Config rpi-gpio out</p> <ul style="list-style-type: none"> ● pin 32 ● DONE
GPIO05 - 29	30 - Ground												
GPIO06 - 31	32 - GPIO12												
GPIO13 - 33	34 - Ground												
GPIO19 - 35	36 - GPIO16												
GPIO26 - 37	38 - GPIO20												
Ground - 39	40 - GPIO21												
<p>● Pin</p> <p>3.3V Power - 1 2 - 5V Power</p> <p>SDA1 - GPIO02 - 3 4 - 5V Power</p> <table border="1"> <tr><td>GPIO06 - 31</td><td>32 - GPIO12</td></tr> <tr><td>GPIO13 - 33</td><td>34 - Ground</td></tr> <tr><td>GPIO19 - 35</td><td>36 - GPIO16</td></tr> <tr><td>GPIO26 - 37</td><td>38 - GPIO20</td></tr> <tr><td>Ground - 39</td><td>40 - GPIO21</td></tr> </table> <p>36</p> <p>Resistor? none</p> <p>Name Name</p>	GPIO06 - 31	32 - GPIO12	GPIO13 - 33	34 - Ground	GPIO19 - 35	36 - GPIO16	GPIO26 - 37	38 - GPIO20	Ground - 39	40 - GPIO21	<p>Config rpi-gpio in</p> <ul style="list-style-type: none"> ● pin 36 ● input pullup ● DONE 		
GPIO06 - 31	32 - GPIO12												
GPIO13 - 33	34 - Ground												
GPIO19 - 35	36 - GPIO16												
GPIO26 - 37	38 - GPIO20												
Ground - 39	40 - GPIO21												
<p>PIN: 32</p> <p>OK</p> <p>PIN: ↑ 36</p> <p>1</p>	<p>Deploy</p> <p>Deploy</p>												

8. ทดสอบโปรแกรม On/Off ที่ LED Pin32 → add node, configure node, กด Deploy, On/Off LED

	Add node <ul style="list-style-type: none"> • Inject node • Inject node • Rpi-gpio out
	Config inject 1 <ul style="list-style-type: none"> • Name = ON • Payload = Boolean <ul style="list-style-type: none"> - true
	
	Config inject 2 <ul style="list-style-type: none"> • Name = OFF • Payload = Boolean <ul style="list-style-type: none"> - false
	Deploy 

9. ทดสอบโปรแกรมไฟกระพริบที่ LED Pin32 → import, place flow1, กด Deploy, Start/Stop Blink

The screenshot shows the Node-RED interface with the following steps:

- Top Bar:** Shows "Deploy" (red button), "View" (left arrow), "Import" (yellow arrow pointing left), and "Export" (left arrow).
- Code Area:** Displays the JSON code for the imported flow. The code defines a function node for "startBlink" and "stopBlink" which set a global variable "light" to true or false respectively, and then send a message to the "msg.payload = 1" output.
- Import Nodes Panel:** Shows the "Clipboard" tab with the imported flow's JSON code. It also has "Paste flow json or select a file to import" fields, and "Local" and "Examples" tabs.
- Import Buttons:** "Import to current flow" (gray) and "new flow" (gray).
- Import Buttons:** "Cancel" (gray) and "Import" (red).
- Flow Diagram:** Shows two parallel flows starting from a timestamp node. The top flow goes through a "startBlink" function node to a PIN: 32 node (set to 32). The bottom flow goes through a "stopBlink" function node to the same PIN: 32 node. A yellow arrow points to the "OK" status of the PIN: 32 node.
- Deploy Panel:** Shows a "Deploy" button (red) and a row of icons for "Info", "Logs", "Dashboard", and "Settings".

<pre> var BLINKDELAY = 250; var light = true; var blinker = setInterval(blink, BLINKDELAY); global.set("blinker", blinker); function blink () { if (light) { msg.payload = 1; light = false; } else { msg.payload = 0; light = true; } node.send(msg); } return; </pre>	At Start Blink function
<pre> clearInterval(global.get("blinker")); msg.payload = 0; return msg; </pre>	At Stop Blink function

คำถาม Quiz-102

1. ใช้ Node-RED เพื่อควบคุมสวิตซ์กดแบบ กดติด กดดับ
2. จากข้อ 1 หากต้องการ 2 วงจร จะต้องจราและปรับแก้ Node-RED อย่างไร

5/6 – การถอนการติดตั้งและติดตั้ง Node-RED บน Raspberry Pi

Lab105 – Remove and Install Node-RED

How to uninstall Node-Red

<https://discourse.Node-RED.org/t/how-to-uninstall-node-red/19135>

1. Uninstall Node-RED

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-Node-RED)
```

```
This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED                                \u2714
Remove old version of Node-RED                \u2714
Node option not specified                     :
Leave existing Node.js                         : --node12 or --node14
Clean npm cache                               v14.17.4   Npm 6.14.14
Install Node-RED core                         \u2714    2.0.5
Move global nodes to local                   -
Leave existing nodes                          -
Install extra Pi nodes                       \u2714
Add shortcut commands                      \u2714
Update systemd script                        \u2714

Any errors will be logged to /var/log/nodered-install.log
All done
```

2. Clean you want to make things.

```
sudo npm -g remove node-red
sudo npm -g remove node-red-admin
rm -R ~/.node-red
```

3. Should do everything important. That does leave node.js installed though along with npm. To remove those, you can use something like (I've never tested this because I've never wanted to remove node.js from a device):

```
sudo apt-get remove nodejs
```

Installing Node-RED Locally On A Raspberry Pi 4

<https://medium.com/@thedyslexiccoder/installing-node-red-locally-on-a-raspberry-pi-4-375ab6053708>

In this tutorial, we'll install Node-RED locally on our Raspberry Pi 4 using the terminal. This is one of the best ways to get our Raspberry Pi updated with the latest Node.js, npm, and Node-RED versions.

1. Get the latest system's package list onto our Raspberry Pi

```
sudo apt update
```

```
sudo apt full-upgrade
```

```
pi@raspberrypi:~ $ sudo apt update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
pi@raspberrypi:~ $ sudo apt full-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

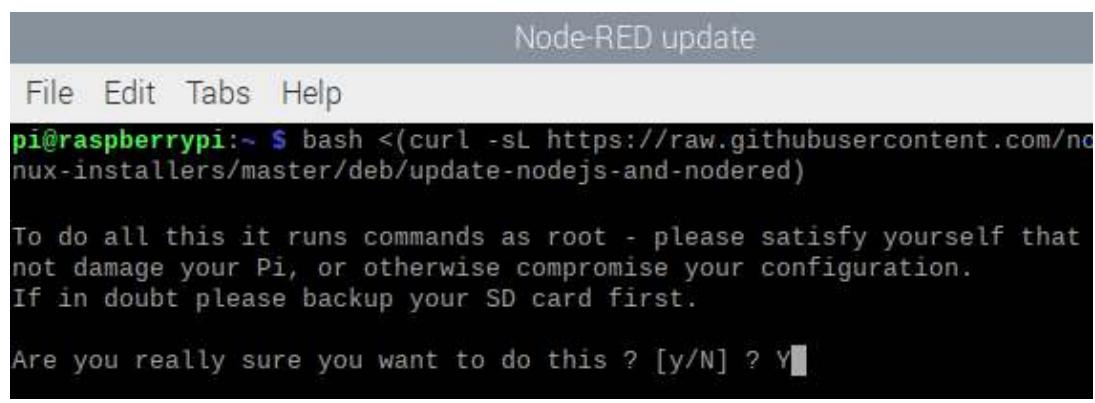
2. To ensure npm can build any modules, run the following command in the command-line:

```
sudo apt install build-essential git
```

```
pi@raspberrypi:~ $ sudo apt install build-essential git
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.6).
git is already the newest version (1:2.20.1-2+deb10u3).
```

3. Install Node-RED. To do this, enter the following script in the terminal:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-Node-RED)
```



4. Node-RED will now begin the process of installing onto our Raspberry Pi. Be patient, as it will take some time to remove and replace the old packages.

```
pi@raspberrypi: Node-RED update
File Edit Tabs Help

Running Node-RED install for user pi at /home/pi on raspbian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED          \u2714
Remove old version of Node-RED    \u2714
Remove old version of Node.js      \u2714
Install Node.js 14 LTS
Clean npm cache
Install Node-RED core
Move global nodes to local
Npm rebuild existing nodes
Install extra Pi nodes
Add shortcut commands
Update systemd script

Any errors will be logged to /var/log/nodered-install.log
```

5. Node-RED is installing on the Raspberry Pi. The script will:
- Remove the pre-packaged version of Node-RED & Node.js.
 - Install the latest version of Node-RED using npm and the latest Node.js LTS (If lower than version 8.x).
 - Install a collection of Pi-specific nodes.
 - Setup specific commands to run the Node-RED service.
6. After the script runs, you'll be greeted with a message, "All done."
7. Node-RED install completed

```
Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip}

Started : Sun 08 Aug 2021 12:15:20 PM +07
Finished: Sun 08 Aug 2021 12:18:38 PM +07
pi@raspberrypi:~ $
```

Lab106 – Node-RED Contribute DHT Sensor

<https://flows.nodered.org/node/node-red-contrib-dht-sensor>

This is a Node Red node to manage connection to a DHT11 or DHT22 sensor on a Raspberry Pi. It allows you to specify the variables that define the connections to the sensor.

This node requires that node-dht-sensor is installed and accessible from Node Red. Since this requires accesss to low-level parts of the Raspberry Pi, you must run your flow with superuser access.

1. Installing this node requires three steps:

- Install the BCM2835 library from here.
- Install the node-dht-sensor dependency

```
sudo npm install --unsafe-perm -g node-dht-sensor
```

```
CXX(target) Release/obj.target/node_dht_sensor/src/dht-sensor.o
CXX(target) Release/obj.target/node_dht_sensor/src/util.o
SOLINK_MODULE(target) Release/obj.target/node_dht_sensor.node
COPY Release/node_dht_sensor.node
make: Leaving directory '/usr/lib/node_modules/node-dht-sensor/build'
+ node-dht-sensor@0.4.3
updated 1 package in 11.009s
pi@raspberrypi:~ $
```

- Install this node

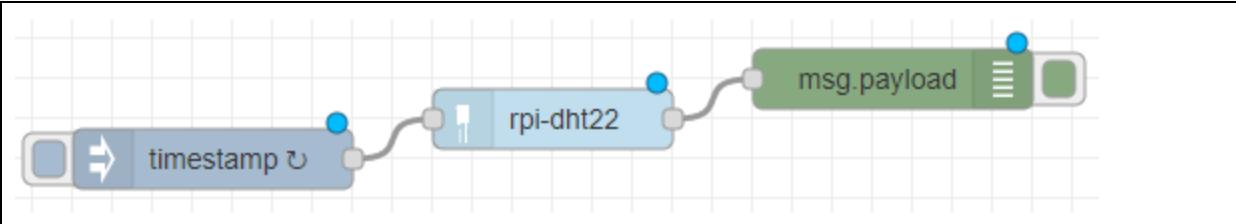
```
sudo npm install --unsafe-perm -g node-red-contrib-dht-sensor
```

```
SOLINK_MODULE(target) Release/obj.target/node_dht_sensor.node
COPY Release/node_dht_sensor.node
make: Leaving directory '/usr/lib/node_modules/node-red-contrib-dht-sensor,
modules/node-dht-sensor/build'
+ node-red-contrib-dht-sensor@1.0.4
added 3 packages from 47 contributors in 14.003s
pi@raspberrypi:~ $
```

- Reboot system

```
sudo reboot
```

2. Using Node



<p>Properties</p> <p>Topic: rpi-dht22</p> <p>Sensor model: DHT22</p> <p>Pin numbering: BCM GPIO</p> <p>Pin number: 18</p>	Sensor = DHT22 Mode = BCM.GPIO Pin = 18
<p>Properties</p> <p>Name: Name</p> <p>msg.payload: timestamp</p> <p>msg.topic: a_z</p> <p>+ add</p> <p>Inject once after 0.1 seconds, then</p> <p>Repeat: interval</p> <p>every 30 seconds</p>	Timestamp No inject once Interval 30
<p>Deploy</p>	Deploy
<p>debug</p> <p>8/8/2021, 2:43:36 PM node: 7ec800daf18fd2ca rpi-dht22 : msg.payload : string[5] "27.90"</p> <p>8/8/2021, 2:44:06 PM node: 7ec800daf18fd2ca rpi-dht22 : msg.payload : string[5] "27.90"</p>	Debug Output

การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ
Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming

ชื่อ-สกุล :

6/6 – คำถ้ามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_101 – ทดสอบ RPi4 GPIO with Python

Python.1 - Python Switch control LED >> กดติด ปล่อยดับ

โปรแกรมที่ใช้ทดสอบ

รูป Code Capture

รูปการทดสอบ 1

รูปการทดสอบ 2

Python.2 - Python Switch control LED >> กดติด กดดับ

โปรแกรมที่ใช้ทดสอบ

รูป Code Capture

รูปการทดสอบ 1

รูปการทดสอบ 2

Python.3 - Python Switch >> Switch Counter

โปรแกรมที่ใช้ทดสอบ

รูป Code Capture

รูปการทดสอบ 1

รูปการทดสอบ 2

Quiz_102 – ทดสอบ RPi4 GPIO with Node-RED

Node-RED.1 – Node-RED เพื่อควบคุมสวิตซ์กดแบบ กดติด กดดับ {Switch-LED 1 คู่}

โปรแกรมที่ใช้ทดสอบ

รูป Code Capture

รูปการทดสอบ 1

รูปการทดสอบ 2

Node-RED.2 – Node-RED เพื่อควบคุมสวิตซ์กดแบบ กดติด กดดับ 2 คู่

โปรแกรมที่ใช้ทดสอบ

รูป Code Capture

รูปการทดสอบ 1

รูปการทดสอบ 2

Node-RED.3 – Node-RED เพื่ออ่าน DHT-22 Sensor

โปรแกรมที่ใช้ทดสอบ

รูป Code Capture

รูปการทดสอบ 1

รูปการทดสอบ 2