

## การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัจฉริยะ ThingsBoard IoTs Platform for smart system

### 3/4 – Widgets and Installing ThingsBoard on Private Server

- แนะนำ Top 5 Hacking IoTs
- การติดตั้งระบบปฏิบัติการให้กับ Raspberry Pi
- การติดตั้ง ThingsBoard Private Server บน Raspberry Pi
- การโปรแกรมเพื่อใช้งาน Rule Chains, MAP Widget และ Multilayer dashboard
- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

<https://colasante-francesco.medium.com/how-to-setup-an-iot-system-using-thingsboard-b705c9189e37>

<https://thingsboard.io/docs/trendz/visualizations-heatmap/>

<https://havespirit.blogspot.com/2017/07/cloud-iot-thingsboard.html>

### 1/5 – แนะนำ Top 5 Hacking IoT

<https://blog.eccouncil.org/top-5-internet-of-things-iot-hacking-tools/>

<https://www.facebook.com/IOT24hours-195875751065834>

<http://thahinc.psru.ac.th/fileupload/4IQymZTkIILY5ws9mG5dnexCHCG4BLR5Ud3.pdf>

การ Hack ความปลอดภัยเพื่อตรวจสอบการทำงานของระบบที่มีประสิทธิภาพ ในขณะที่ปัจจัยหลายประการมีส่วนช่วยในการพัฒนา แต่สิ่งที่สำคัญที่สุดคือความพร้อมใช้งานของเครื่องมือ Hack IoT ที่ทำให้งานง่ายขึ้น Ethical hacker คือ ผู้เชี่ยวชาญด้านความปลอดภัยที่แฮ็กคอมพิวเตอร์อย่างถูกกฎหมายเพื่อตรวจจับความเสี่ยงและการเข้าถึงที่ผิดกฎหมาย

ผู้เชี่ยวชาญด้านความปลอดภัยทุกคนต้องรู้จักอุปกรณ์ทั้งหมดในเครือข่าย รวมถึง Internet of Things (IoT) อย่างละเอียดในแกนหลักของเครื่องมือแฮ็ก IoT เปิดใช้งานการแฮ็กเนื่องจากช่วยให้ขั้นตอนที่จำเป็นเป็นไปโดยอัตโนมัติ ethical hacker ที่มีสามารถใช้พวกมันเพื่อดำเนินการคุณลักษณะบางอย่างที่ช่วยตรวจหาช่องโหว่ของอุปกรณ์



## 1. Wireshark

This is an open-source tool that is used for analyzing and capturing real-time network traffic. The experts can see the comprehensive details about packets and detect the traffic issue while decoding the traffic. The tool can help in detecting the traffic, resolving the performance issues, protocol problems, and misconfigured programming.



## 1. Wireshark



Wireshark เป็นเครื่องมือวิเคราะห์แพ็กเก็ตเครือข่ายที่ให้คุณบันทึกและวิเคราะห์การรับส่งข้อมูลเครือข่ายแบบเรียลไทม์ เป็นโปรแกรมโอเพ่นซอร์สที่หนึ่งในเครื่องมือรักษาความปลอดภัยเครือข่ายที่สำคัญที่สุด

เครื่องมือ Wireshark มีการใช้ โดยปกติจะใช้เพื่อแก้ไขประสิทธิภาพของเครือข่าย ตรวจสอบการรับส่งข้อมูลที่ผิดปกติ การเขียนโปรแกรมที่กำหนดค่าผิดพลาด และปัญหาโปรโตคอล โดยเหล่าแฮกเกอร์ที่มีจะใช้ตัววิเคราะห์โปรโตคอลเครือข่ายเพื่อทำความเข้าใจรายละเอียดของเครือข่ายเป้าหมาย

ประโยชน์ของเครื่องมือ Wireshark:

- สามารถใช้ได้กับแพลตฟอร์มต่างๆ: Windows และ UNIX
- สามารถดูรายละเอียดที่ครอบคลุมเกี่ยวกับแพ็กเก็ตภายในเครือข่ายและไม่ใช่กรรมสิทธิ์
- ตรวจสอบปัญหาที่คุณส่งและรับ
- ถอดรหัสการรับส่งข้อมูลจากบุคคลอื่น

- <https://www.wireshark.org/>
- <https://www.varonis.com/blog/how-to-use-wireshark/>

## 2. Network Mapper (Nmap)

The open-source scanning tool is used for several things like crafting packets, network discovery and operating the data easily while running analyses of responses. The data can help in administering the network for vulnerabilities detection. The tool is used for the interaction with the target host and auditing networking for the detecting of servers.



## 2.Nmap



**Network Mapper (Nmap)** เป็นเครื่องมือโอเพ่นซอร์สที่ใช้กันอย่างแพร่หลายซึ่งใช้ในการค้นพบเครือข่ายและในขณะที่ทำการสแกน จะส่งแพ็กเก็ตที่สร้างขึ้นเพื่อค้นหาอุปกรณ์ที่เชื่อมต่อกับเครือข่าย จากนั้นจะให้ข้อมูลกับระบบปฏิบัติการที่กำลังทำงานเพื่อวิเคราะห์การตอบสนอง **Nmap** ได้รับการพัฒนาเป็นพิเศษสำหรับเครือข่ายระดับองค์กรเพื่อสแกนอุปกรณ์หลายร้อยเครื่อง **Nmap** ช่วยให้ผู้ใช้ดูแลระบบเครือข่ายตรวจสอบหาช่องโหว่

ผู้ตรวจสอบความปลอดภัยใช้ **Nmap** เพื่อระบุช่องโหว่ในระบบของตน แต่แก็กเกอร์ถือว่า **Nmap** เป็นตัวกำหนดความจุพอร์ตที่สำคัญและการค้นพบโฮสต์และบริการบนเครือข่าย คุณสามารถใช้ **Nmap** เพื่อกำหนดเป้าหมายระบบโดยใช้สคริปต์ **Nmap** ที่มีอยู่เมื่อทำการตรวจสอบความปลอดภัยและการสแกนช่องโหว่

ประโยชน์ของ **Nmap**:

- การโต้ตอบกับโฮสต์เป้าหมายสามารถทำได้โดยใช้เอ็นจินสคริปต์ **Nmap**
- **Nmap** ใช้ในการตรวจสอบเครือข่ายเพื่อตรวจหาเซิร์ฟเวอร์ใหม่

- <https://nmap.org/>
- <https://www.bloggang.com/mainblog.php?id=palmnattaphat>

### 3. Fiddler

The web proxy tool is used for pen-testing devices to debug the web traffic and converting the proxy setting as an HTTP proxy. It supports data from applications while sending an HTTP request to the service or website. The tool works on HTTP clients and stops customer traffic.



## 3.Fiddler



Fiddler คือเครื่องมือ Web Debugging Proxy ที่ทำงานร่วมกับเบราว์เซอร์หรือแพลตฟอร์มใดก็ได้ มีคุณสมบัติมากมายที่จะช่วยอุปกรณ์ทดสอบปากกา อนุญาตให้ผู้ใช้ดีนักการเข้าชมเว็บบนระบบใดก็ได้ โปรแกรมยังมีการบันทึกเก็บประวัติ Logs Traffic ได้ทั้งขาเข้าและขาออก และยังสามารทำการแก้ไข Web Sessions ได้อีกด้วยง่ายๆ นอกจากนี้ยังจำลองการเข้าเว็บไซต์ว่าเข้ามาจากหลากหลายพื้นที่ หลากหลายประเทศได้เช่นกัน

ประโยชน์ของนัก Fiddler:

- Works with nearly every HTTP client.
- Can stop customer traffic on non-Windows platforms also.

- <https://www.telerik.com/fiddler>
- <https://docs.telerik.com/fiddler/configure-fiddler/tasks/configurefiddler>



#### 4. Metasploit

The tool is a security information tool that is used for penetration testing to check the vulnerabilities of the system and secure the computer remotely while gaining more access. It is easy to use and offers access source code for users.



## 4. Metasploit



Metasploit เป็นเครื่องมือรักษาความปลอดภัยที่ให้ข้อมูลความปลอดภัยของซอฟต์แวร์และปรับปรุงการทดสอบการเจาะระบบ เป็นเครื่องมือเจาะระบบโอเพ่นซอร์สที่ใช้ในการทดสอบช่องโหว่ในคอมพิวเตอร์และเข้าถึงคอมพิวเตอร์จากระยะไกลและรักษาความปลอดภัยจากการแฮ็กแฮกเกอร์สามารถใช้ Metasploit เพื่อค้นหาและค้นพบช่องโหว่ของซอฟต์แวร์บนเครือข่าย เครื่องมือ Metasploit จะใช้ในการใช้ประโยชน์จากระบบเครือข่ายคอมพิวเตอร์เป้าหมายระยะไกล ผู้ทดสอบการเจาะที่ได้รับอนุญาตในองค์กรสามารถใช้เฟรมเวิร์ก Metasploit เพื่อค้นหาพื้นที่เสี่ยงทั้งหมดด้วยรหัสที่พร้อมใช้งานหรือส่วนบุคคล

Metasploit ยังใช้ในการไล่ล่าภัยคุกคามด้วยการสำรวจช่องโหว่และการรายงาน และข้อบกพร่องที่เป็นระบบจะได้รับการแก้ไขด้วยข้อมูลที่ได้รับ

#### ประโยชน์ของ Metasploit:

- Metasploit เป็นโอเพ่นซอร์สและใช้งานง่ายฟรี
- Metasploit อนุญาตให้ผู้ใช้เข้าถึงซอร์สโค้ดและเพิ่มโมดูลที่กำหนดเองได้

- <https://www.metasploit.com/>
- <https://blog.eldernode.com/introducing-and-install-metasploit-on-kali/>

## 5. Maltego

This tool is a great way to connect and collect the data to investigate graphical relation analysis and open-source intelligence. It is used to focus on the library of transforms for data visualization and open sources to use for data mining and connection analysis. It is used to represent information in an easy to understand and simple way along with showing the severity and complexity.



## 5.Maltego



**Maltego** เป็นเครื่องมือโอเพนซอร์ส มีวิธีการที่ครอบคลุมในการรวบรวมและเชื่อมโยงข้อมูลกับงานตรวจสอบผ่านโอเพนซอร์สและการวิเคราะห์ความสัมพันธ์แบบกราฟิกในรูปแบบเรียลไทม์ **Maltego** มุ่งเน้นไปที่การค้นหาไลบรารีของการแปลงเพื่อค้นหาข้อมูลจากโอเพนซอร์สและแสดงภาพในรูปแบบกราฟที่สามารถใช้สำหรับการวิเคราะห์การเชื่อมต่อและการขุดข้อมูล

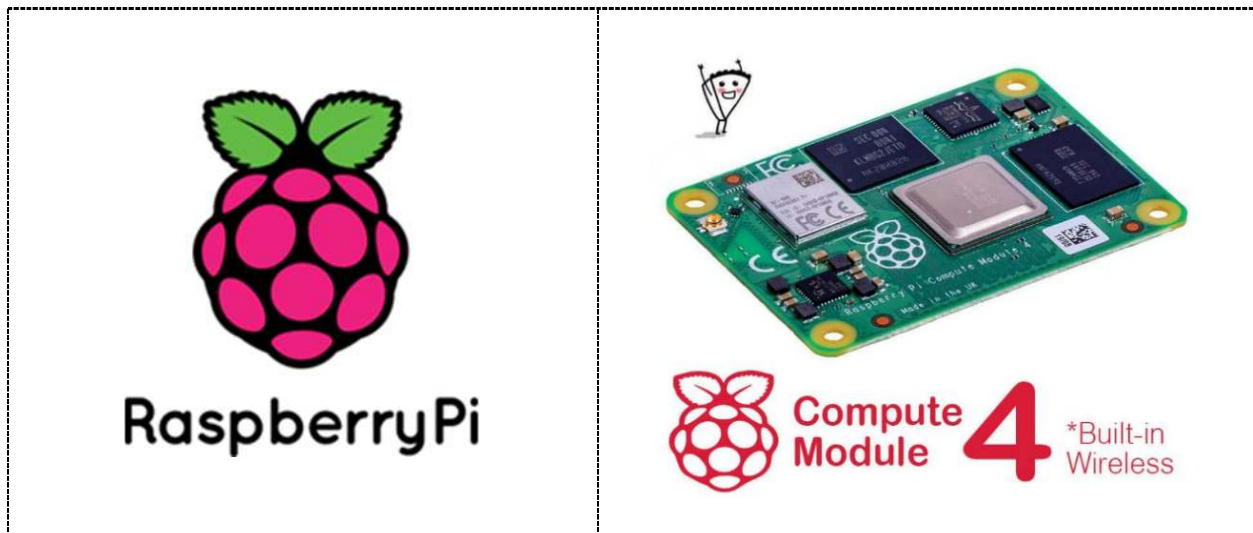
**Maltego** เป็นเครื่องมือขุดข้อมูลที่ทำงานร่วมกันซึ่งมีกราฟการวิเคราะห์ อุปกรณ์นี้ใช้ในการตรวจสอบออนไลน์เพื่อค้นหาการเชื่อมต่อจากแหล่งออนไลน์ต่างๆ **Maltego Tool** มีข้อได้เปรียบในการวิเคราะห์ความเชื่อมโยงระหว่างข้อมูลโอเพนซอร์สแต่ละรายการในโลกแห่งความเป็นจริง สิ่งนี้สามารถเป็นเครื่องมือที่มีค่าในระหว่างขั้นตอนการทดสอบ **Maltego** ทำงานภายใต้ Windows, OS X และ Linux ทำให้เป็นเครื่องมือที่ยืดหยุ่นสำหรับสภาพแวดล้อมทางธุรกิจเกือบทุกประเภท

ประโยชน์ของ **Maltego**:

- **Maltego is to show the complexity and severity of single failure points.**
- เป็นการนำเสนอข้อมูลในลักษณะที่เข้าใจง่าย

- <https://www.ethicalhacker.net/columns/gates/maltego-part-i-intro-and-personal-recon/>
- <https://www.youtube.com/watch?v=9-cd4w7Jl3g>

## 2/5 – Raspberry Pi และการติดตั้งระบบปฏิบัติการให้กับ Raspberry Pi

<http://www.raspberrypi.org/news>

## 2.1 Raspberry Pi คืออะไร?

Raspberry Pi (ออกเสียงว่า ราส-เบอร์-รี่-พาย) เป็นเครื่องคอมพิวเตอร์ขนาดเล็ก ที่มีขนาดเพียงเท่ากับบัตรเครดิต ที่สำคัญคือ RPi นี้มีราคาถูกมาก เมื่อเทียบกับคอมพิวเตอร์เดสก์ท็อปปกติ คือมีราคาเพียงแค่หนึ่งพันกว่าบาทเท่านั้นเอง RPi สามารถทำงานได้เหมือนเครื่องคอมพิวเตอร์ทุกอย่าง เราสามารถต่อ RPi นี้เข้ากับจอคอมพิวเตอร์หรือจอทีวีที่รองรับ HDMI นอกจากต่อจอแสดงผลแล้ว ก็สามารถต่อเมาส์และคีย์บอร์ดผ่าน USB port ปกติ เพราะฉะนั้นสามารถนำเมาส์และคีย์บอร์ดที่มีอยู่แล้วมาต่อได้เลย ระบบจ่ายไฟของราสเบอร์รี่พายใช้สาย Mini USB ที่เราใช้ชาร์จมือถือและอุปกรณ์อื่นๆ เข้ากับคอมพิวเตอร์ หรือเข้ากับหัวชาร์จไฟมือถือก็ได้เช่นกัน กรณีเป็น RPi4 จะใช้สาย USB-Type C ขนาดกระแส 3A และมี 2 ช่องต่อจอภาพแบบ micro-HDMI

ราสเบอร์รี่พาย (Raspberry Pi) เกิดขึ้นในปี 2549 ที่มหาวิทยาลัยเคมบริดจ์ ประเทศอังกฤษ โดยผู้สร้างทั้งสี่คนคือ อีเบน อัฟตัน, ร็อบ มุลลินส์, แจ็ค แลง และ อลัน มายครอฟท์ มีจุดมุ่งหมายที่จะให้ ราสเบอร์รี่พายเป็นคอมพิวเตอร์ราคาย่อมเยาที่ใครๆ ก็สามารถหามาครอบครองได้ และสามารถศึกษาการทำงานของคอมพิวเตอร์พร้อมทั้งเขียนโปรแกรมง่ายๆ ได้ทันที การที่ราสเบอร์รี่พายเป็นบอร์ดวงจรรวมที่เปลือยเปล่า ทำให้เด็ก ๆ ได้เห็นชิ้นส่วนทั้งหมดที่เป็นส่วนประกอบของคอมพิวเตอร์ได้อย่างชัดเจน ซึ่งจะทำให้เข้าใจการทำงานของคอมพิวเตอร์ในปัจจุบันที่มาในกล่องสวยงามได้มากขึ้น

การใช้งาน Raspberry Pi ในปัจจุบัน(2021/08/28) จะเป็นสองลักษณะ คือ เป็น Single Board Computer ได้แก่ Raspberry Pi 4 Board และ ใช้งานเป็น Compute Module ได้แก่ RPi CM4 ทั้งสองรุ่นมีหน่วยประมวลผลเดียวกัน แต่มี IO, RAM, eMMC และ Wifi ที่ต่างกันไปตามรุ่นย่อยของบอร์ด

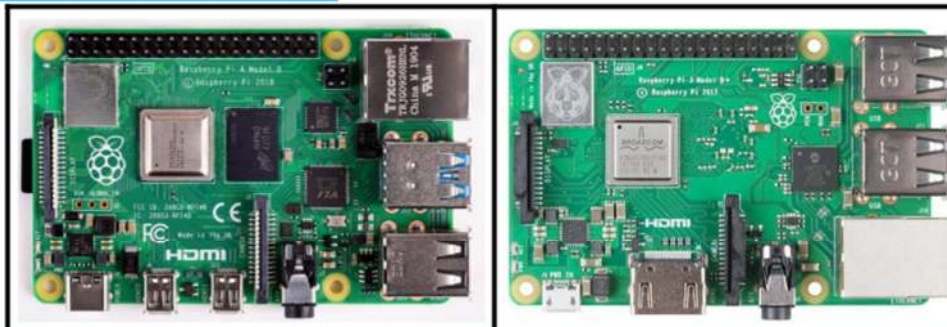


2.2 สเปกและส่วนประกอบของ RASPBERRY PI 4 MODEL B (@24<sup>TH</sup> JUNE 2019)

<https://th.cytron.io/c-raspberry-pi/c-raspberry-pi-main-board/p-raspberry-pi-4-model-b-4gb>

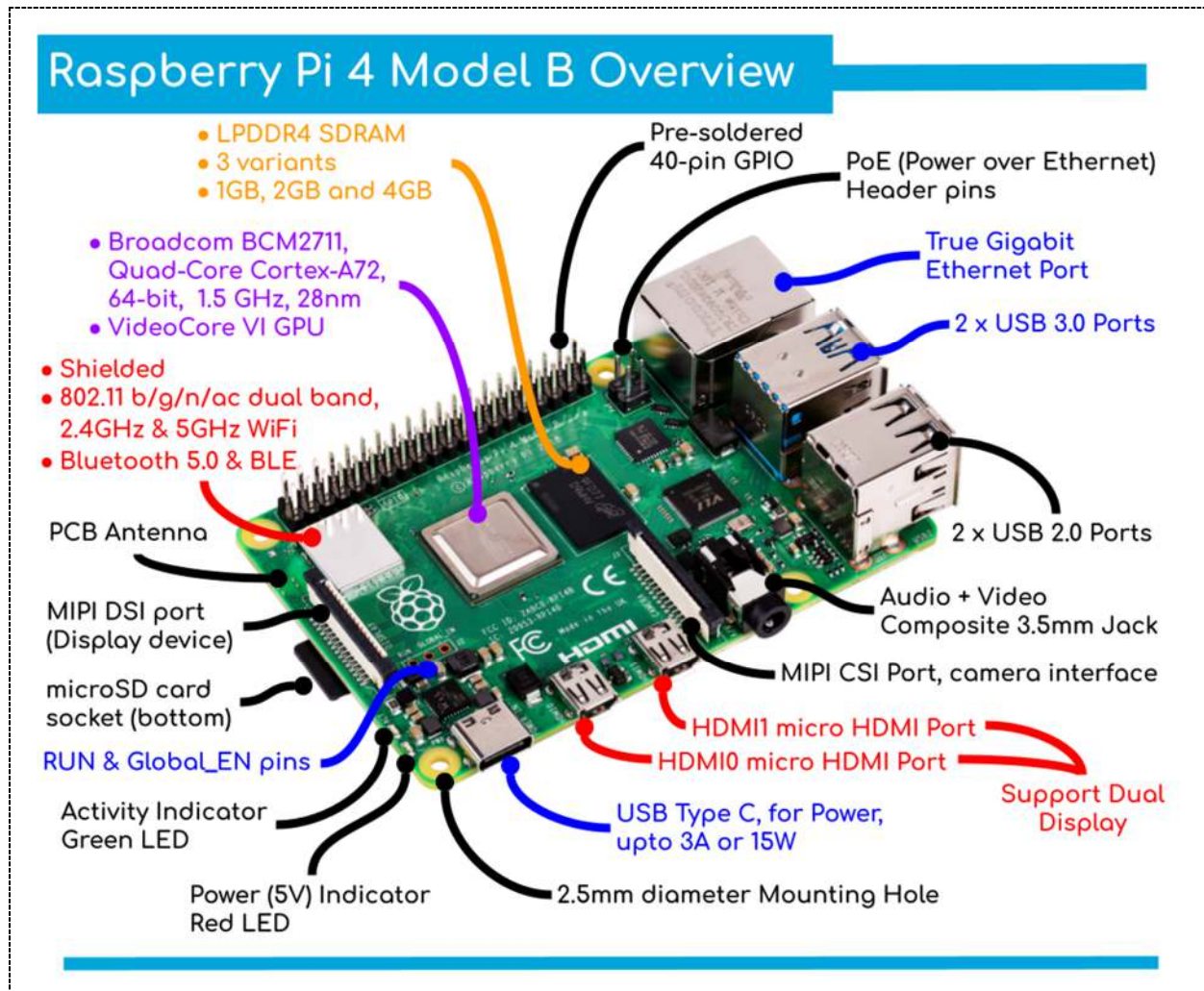
Announced on the 24th June 2019, the new Raspberry Pi 4 Model B is released and it's available on the Cytron marketplace! Check out the Major Upgrades on Raspberry Pi 4 Model B. And here is the comparison table between Raspberry Pi 4 Model B vs Raspberry Pi 3 Model B+:

## 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

## 2.3 Raspberry Pi Compute Module 4 (CM4)

<https://www.arduitronics.com/product/4659/raspberry-pi-compute-module-4-io-board-แท้-100>

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

	<p><b>DESCRIPTION</b></p> <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>
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

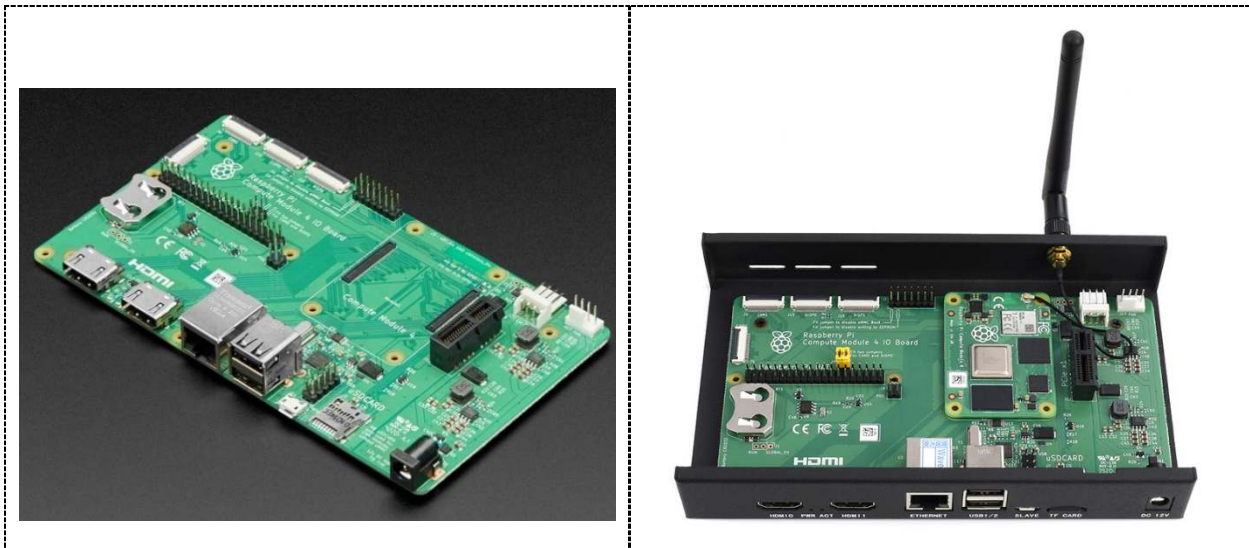
### RAM, eMMC and Wireless Options

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

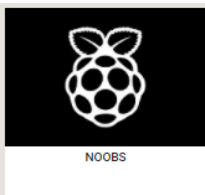
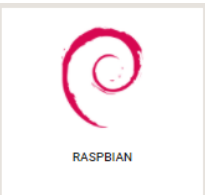







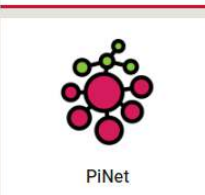







This **PoE CM4 IO board** breakouts quite some extra peripheral ports if we compared it with Raspberry Pi 4 Model B:

- 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 tach 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 :)

## 2.4 ระบบปฏิบัติการของ RASPBERRY PI

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

## 2.5อ้างอิง

- <http://www.raspberrypi.org/>
- [https://en.wikipedia.org/wiki/Raspberry\\_Pi](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/>

## Lab301 -- Raspberry PI Install Operating System

ในบทความนี้ จะสอนการใช้งาน Raspberry Pi เบื้องต้น ตั้งแต่การติดตั้งระบบปฏิบัติการลง Micro SD card ซึ่งในที่นี้คือ Raspbian โดยใช้กับบอร์ด Raspberry Pi 4 โดยทำใช้ระบบปฏิบัติการ Windows ในการติดตั้ง และจำเป็นต้องต่ออินเทอร์เน็ตด้วย

### 3.1 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

### 3.2 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](#)

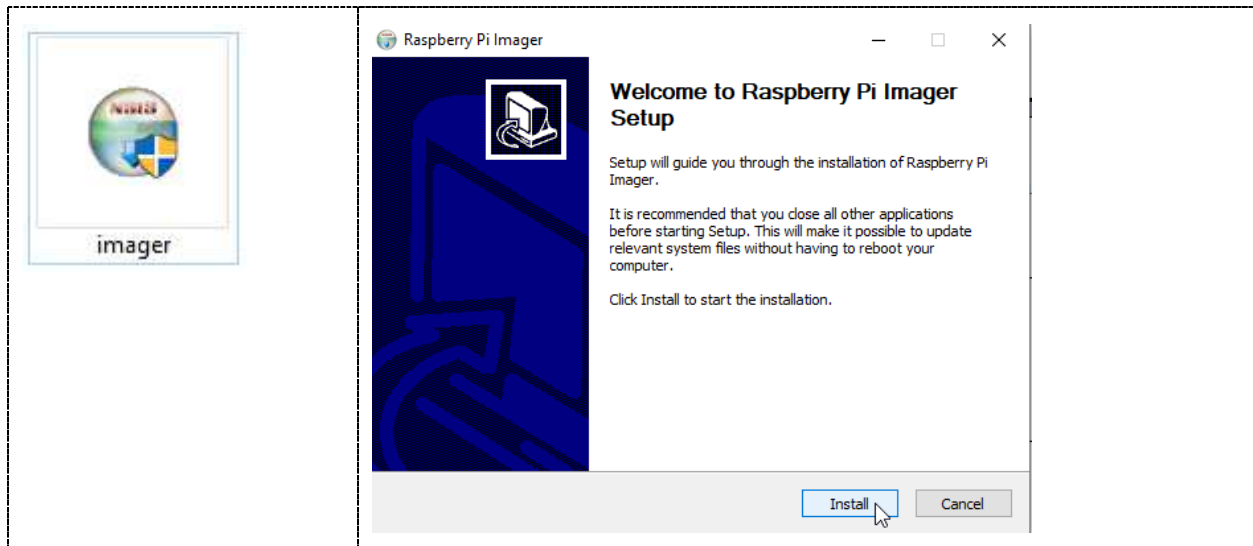
[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type

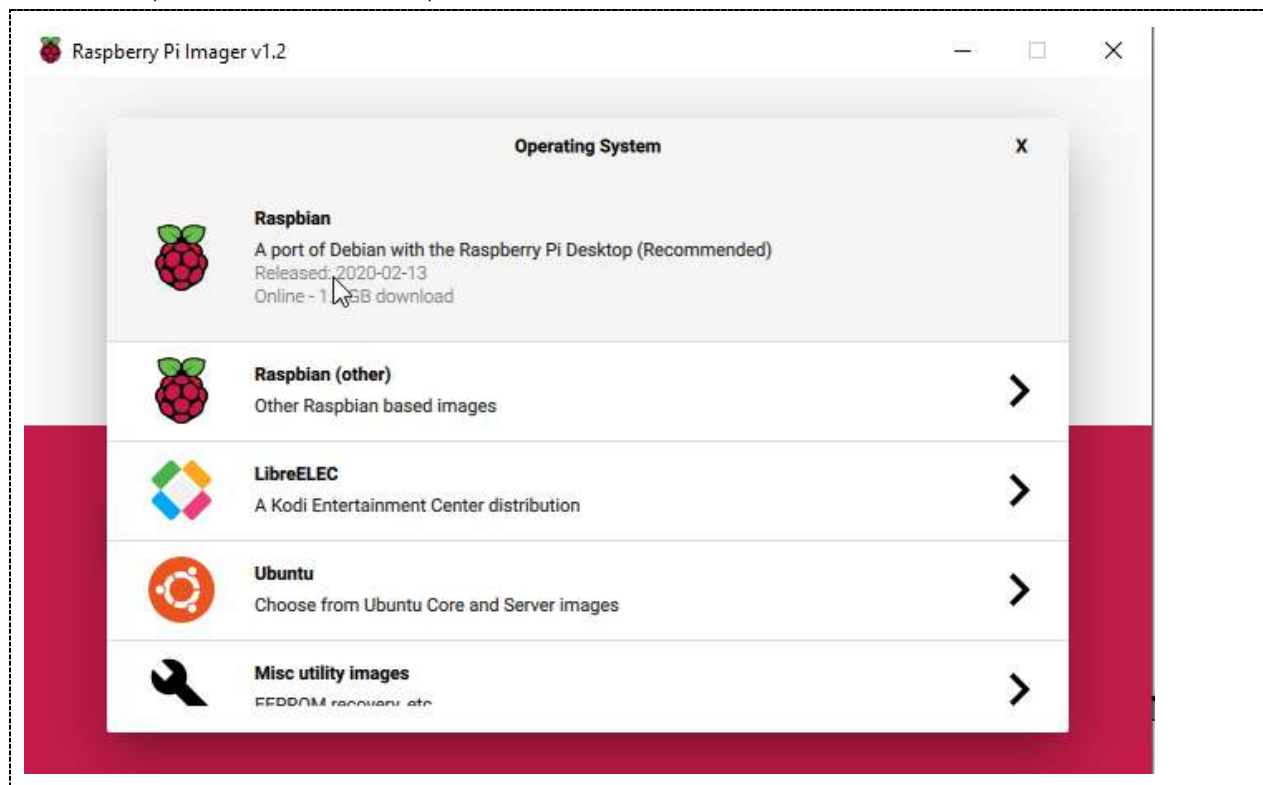
```
sudo apt install rpi-imager
```

in a Terminal window.

Step 2a: เมื่อดาวน์โหลดเสร็จเรียบร้อยแล้ว ให้ติดตั้ง image Setup



Step 3a: เลือก OS เป็น Raspbian และเลือก SD Card





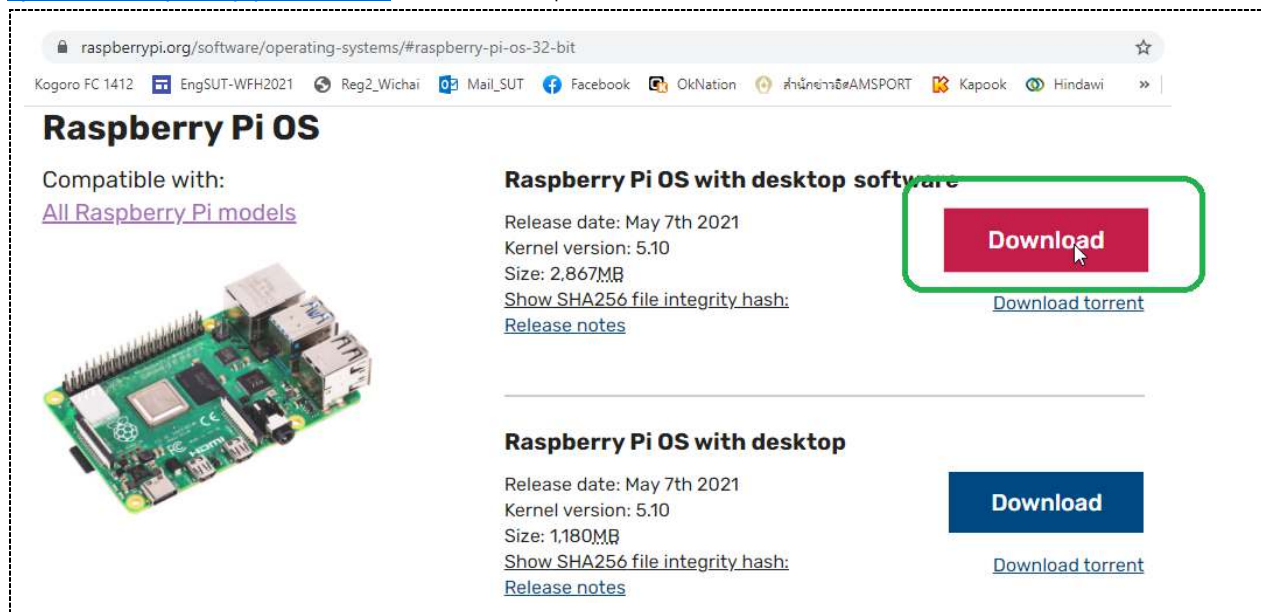
Step 4a: Write OS image



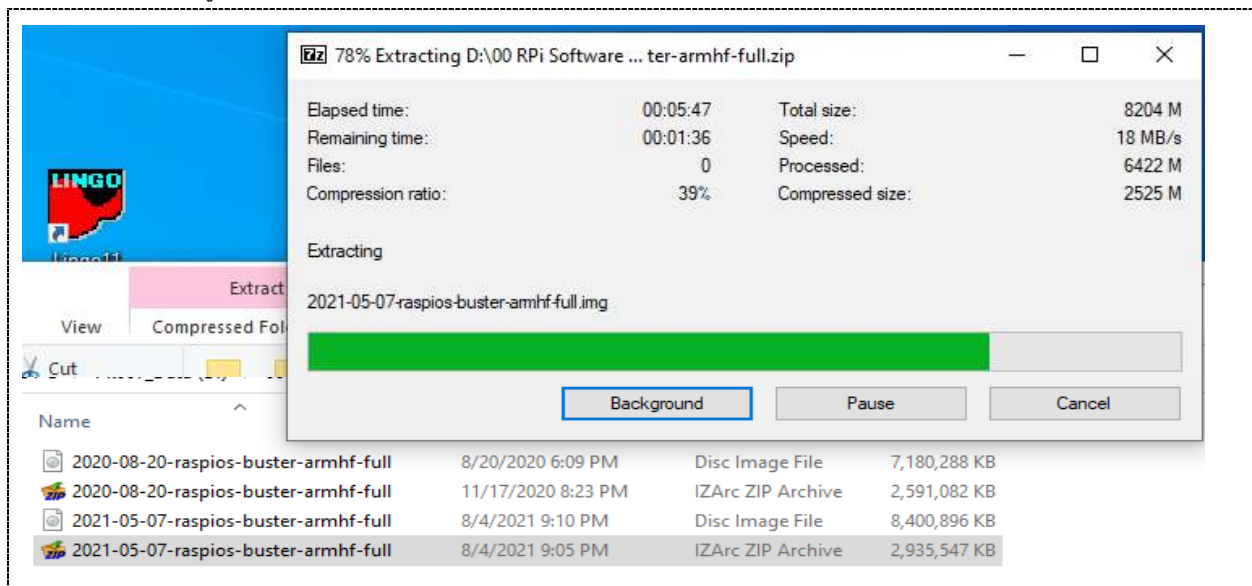
Step 5a: เมื่อติดตั้งเสร็จแล้วให้ถอด Micro SD card ออกจากคอมพิวเตอร์แล้วใส่ไปที่ Raspberry Pi ต่อไป

### วิธีที่ 2 - Manually install an operating system image

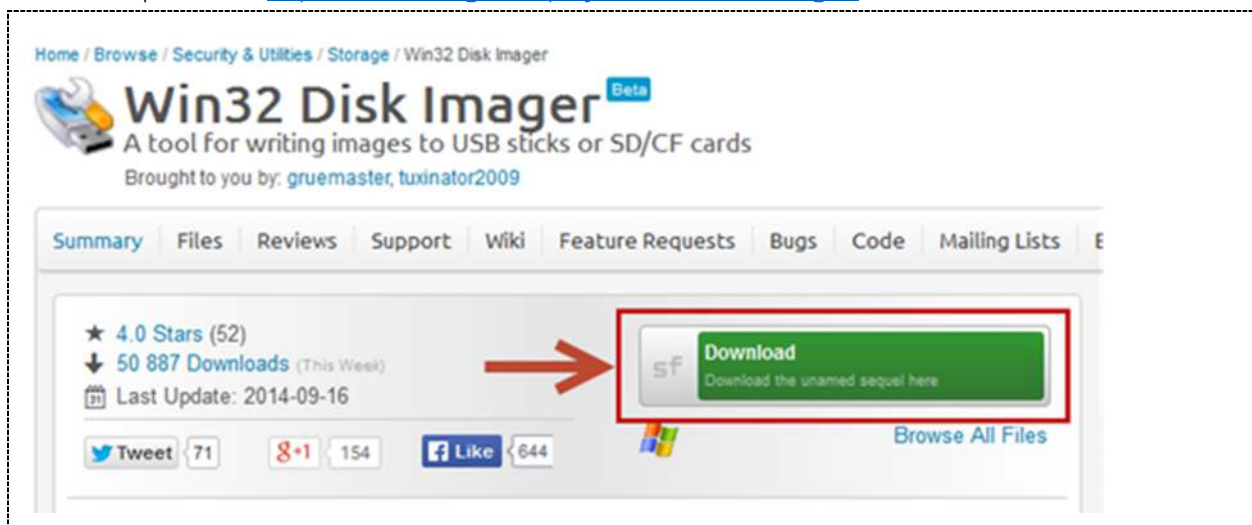
Step 1b: ดาวน์โหลดอิมเมจ OS จาก <https://www.raspberrypi.org/software/operating-systems/#rasberry-pi-os-32-bit> แล้วเลือก Raspbian (หรือ Download ด้วย bit torrent ก็ได้)



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



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



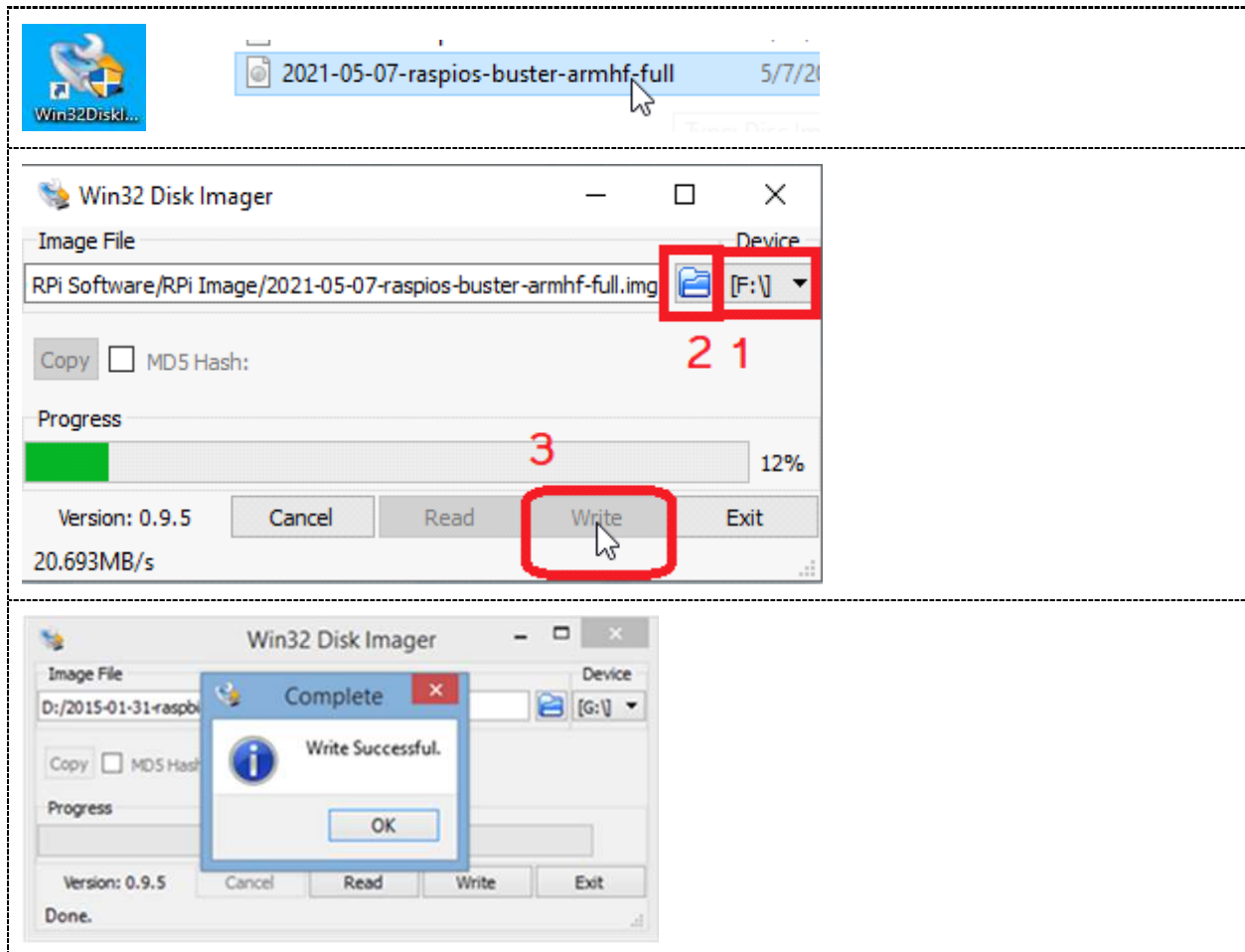
Step 4b: แยกซิปไฟล์ และติดตั้งโปรแกรม Win32DiskImager (คลิกขวา Run as administrator)



Step 5b: ใส่ SD card เข้ากับเครื่องคอมพิวเตอร์หรือ card reader

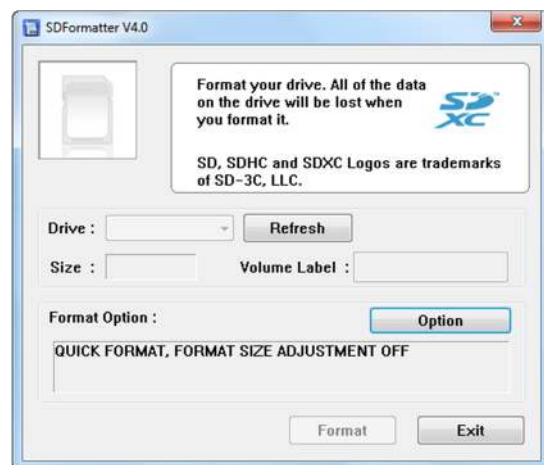
Step 6b: เปิดโปรแกรม Win32 Disk Imager

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

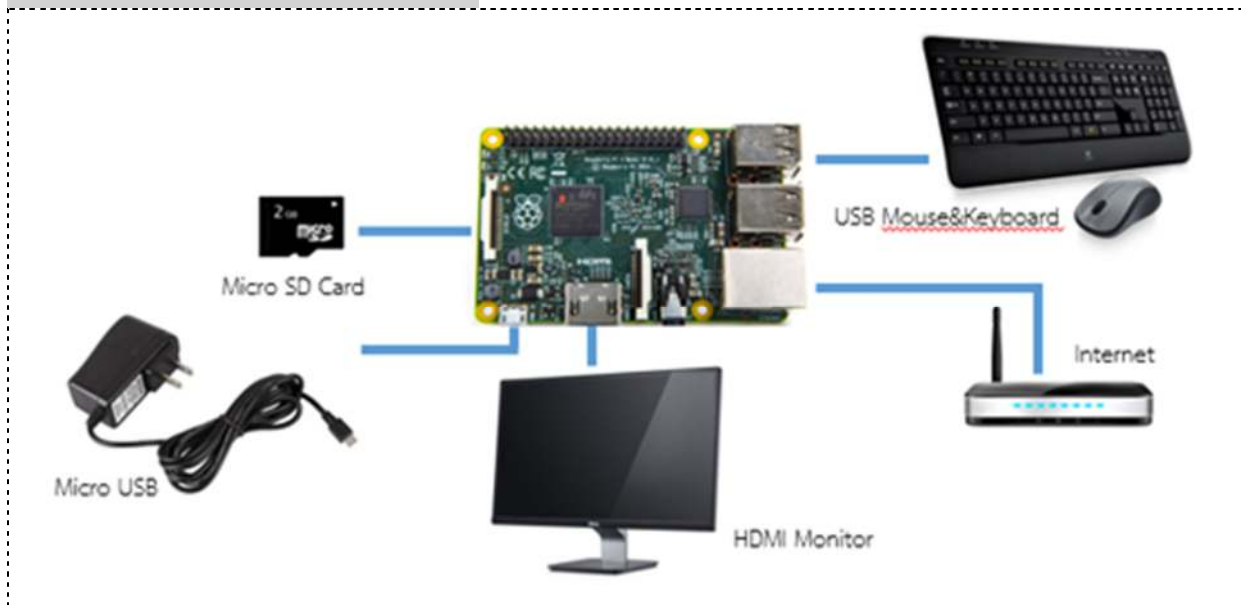


Step 8b: เมื่อติดตั้งเสร็จแล้วให้ถอด Micro SD card ออกจากคอมพิวเตอร์แล้วใส่ไปที่ Raspberry Pi ต่อไป

**Note:** กรณีต้องการ format SD Card ให้ใช้โปรแกรม SDFormatter ในการดำเนินการ <https://sd-formatter.informer.com/3.1/>



### 3.3 การเชื่อมต่อ เมื่อเริ่มต้นใช้งานครั้งแรก

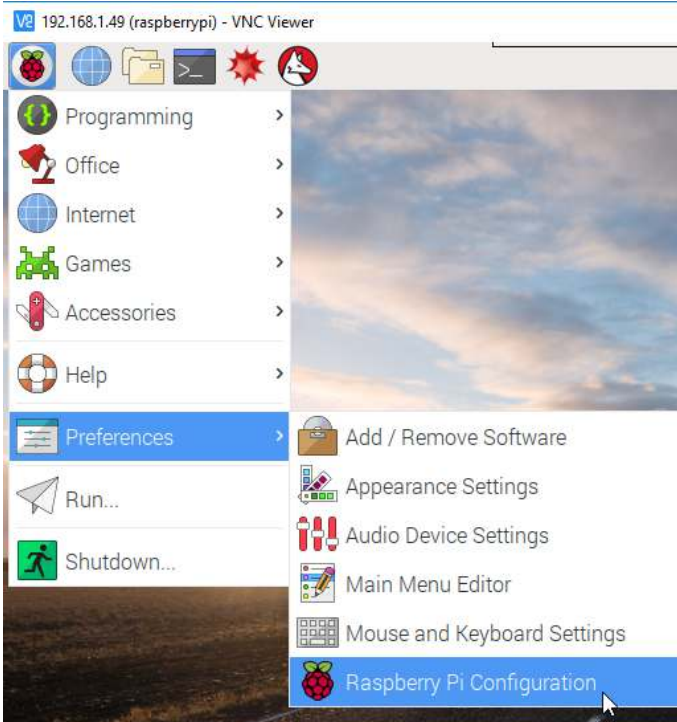


Step 1: จ่ายไฟให้ Raspberry Pi ตั้งค่า Remote Password = **raspberry**

Step 2: กำหนดค่าครั้งแรก เพื่อให้สามารถใช้งานแบบ Remote Desktop ได้

	<p>Right Click ที่ลูกศรสวนทาง</p>
	<p>เลือก Network Setting</p>
	<p>Interface = eth0</p> <p><input type="checkbox"/> Automatically</p> <p><input checked="" type="checkbox"/> Disable IPv6</p> <p>192.168.100.xx</p> <p>Apply, and Close</p>

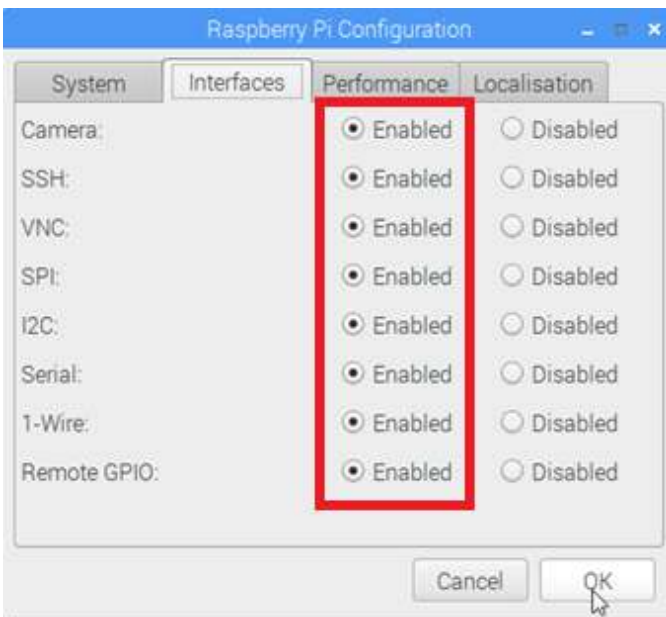




Raspberry

→ Preference

→ → RPi Configuration



Tab = Interface

Camera ☒ Enable

SSH ☒ Enable

VNC ☒ Enable

SPI ☒ Enable

I2C ☒ Enable

Serial ☒ Enable

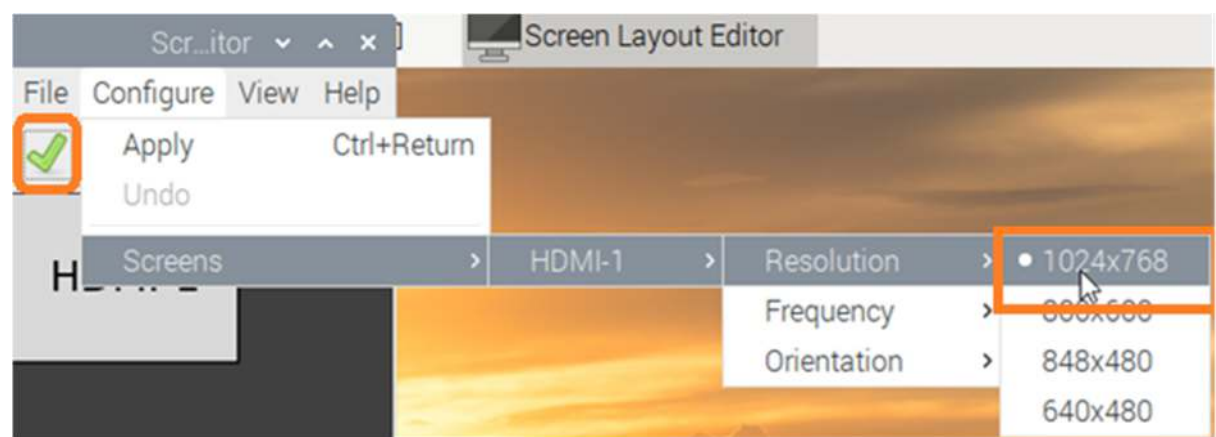
1-Wire ☒ Enable

Remote GPIO ☒ Enable

OK

Step 3: กำหนดขนาดหน้าจอเชื่อมต่อ

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

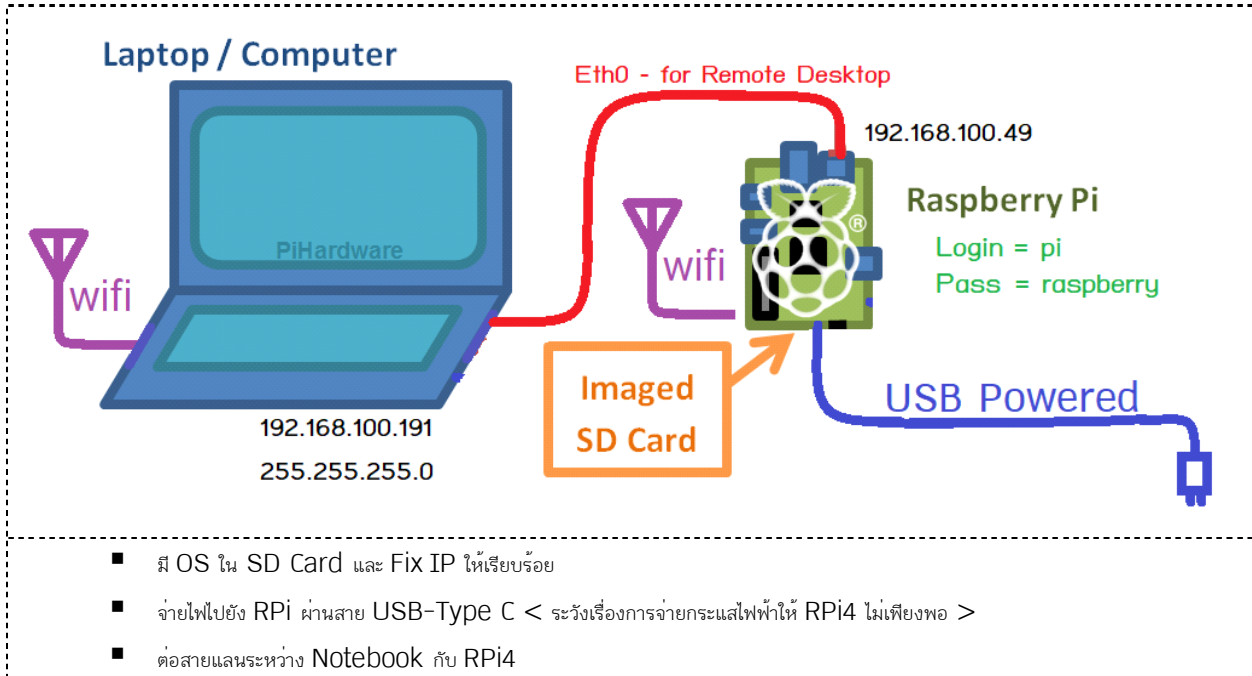


Step 4: Restart Raspberry Pi

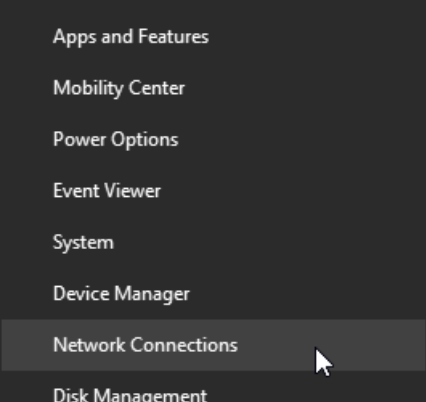
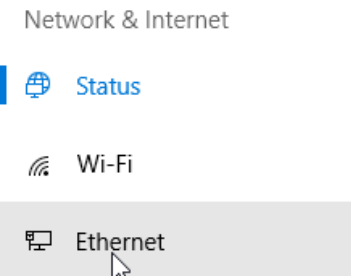
- หลังจากนี้จะทำงานแบบ Remote Desktop ซึ่งไม่จำเป็นต้องมี Monitor, Mouse, Keyboard ต่อที่ Raspberry Pi

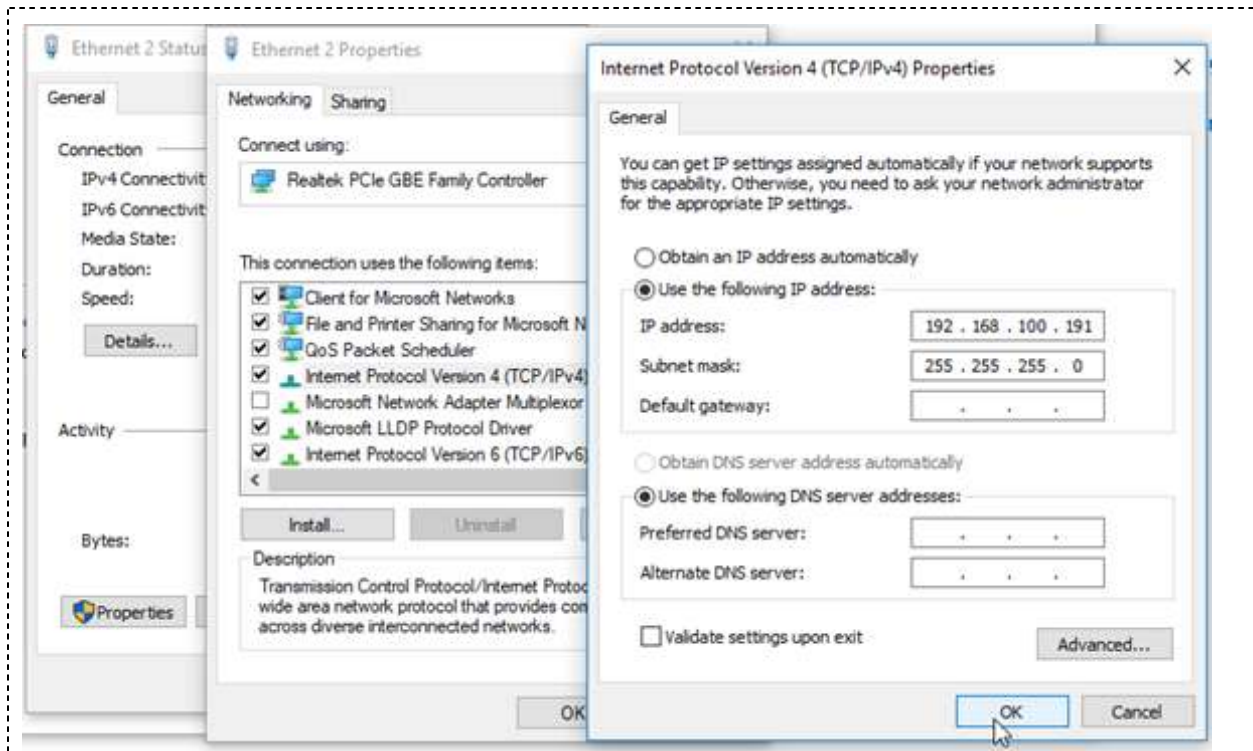
### 3.4 การเชื่อมต่อแบบ REMOTE DESKTOP

Step 1: การใช้งานแบบ Remote Desktop



Step 2: Fix IP ที่ Notebook

	<p>⌘ Win , Right Click → Network Connection</p>				
	<p>Ethernet</p>				
<p><a href="#">Network and Sharing Center</a> <a href="#">HomeGroup</a></p>	<p>Network and Sharing Center</p>				
<p>View your basic network information and set up connections</p> <p>View your active networks</p> <table border="1"> <tbody> <tr> <td> <b>testVirus</b> Private network </td> <td> Access type: Internet  HomeGroup: <a href="#">Ready to create</a>  Connections:  <a href="#">Wi-Fi (testVirus)</a> </td> </tr> <tr> <td> <b>Unidentified network</b> Public network </td> <td> Access type: No network access  Connections:  <a href="#">Ethernet 2</a> </td> </tr> </tbody> </table> <p>Change your networking settings</p> <ul style="list-style-type: none"> <li> <a href="#">Set up a new connection or network</a> Set up a broadband, dial-up, or VPN connection; or set up a router or access point.</li> <li> <a href="#">Troubleshoot problems</a> Diagnose and repair network problems, or get troubleshooting information.</li> </ul>	<b>testVirus</b> Private network	Access type: Internet HomeGroup: <a href="#">Ready to create</a> Connections: <a href="#">Wi-Fi (testVirus)</a>	<b>Unidentified network</b> Public network	Access type: No network access Connections: <a href="#">Ethernet 2</a>	<p>Ethernet 2</p>
<b>testVirus</b> Private network	Access type: Internet HomeGroup: <a href="#">Ready to create</a> Connections: <a href="#">Wi-Fi (testVirus)</a>				
<b>Unidentified network</b> Public network	Access type: No network access Connections: <a href="#">Ethernet 2</a>				



Properties

→ Internet Protocol Version 4 (TCP/IPv4)

→ → ● Used the following IP Address

192.168.100.191

255.255.255.0


OK

→ OK

Close

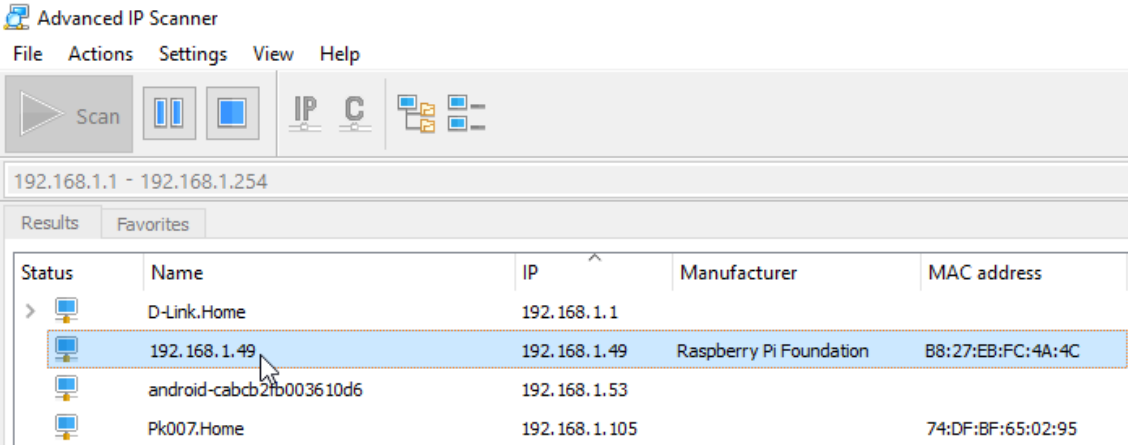


## Step 3: ตรวจสอบลูกข่ายที่มีในโครงข่าย ด้วยโปรแกรม ipscan



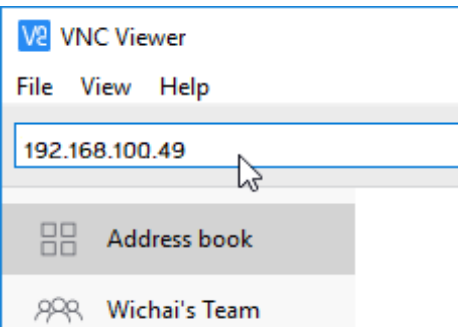
- กด IP เพื่อกำหนด ย่าน IP จะให้ IP ในวงแลนที่ Notebook อยู่
- กด SCAN จะเห็น IP ของ RPi

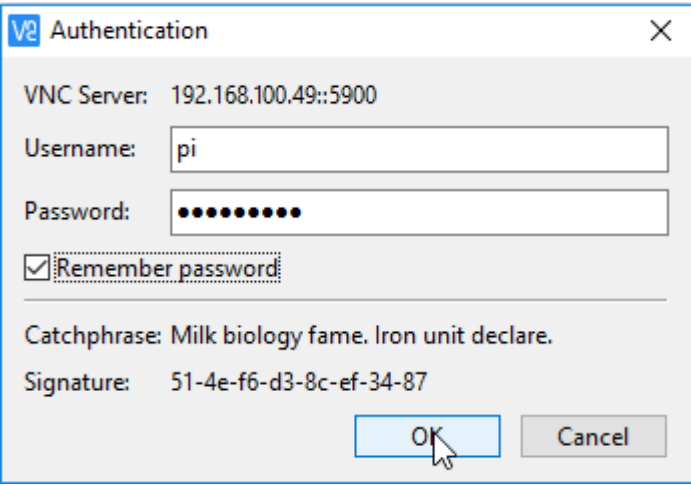
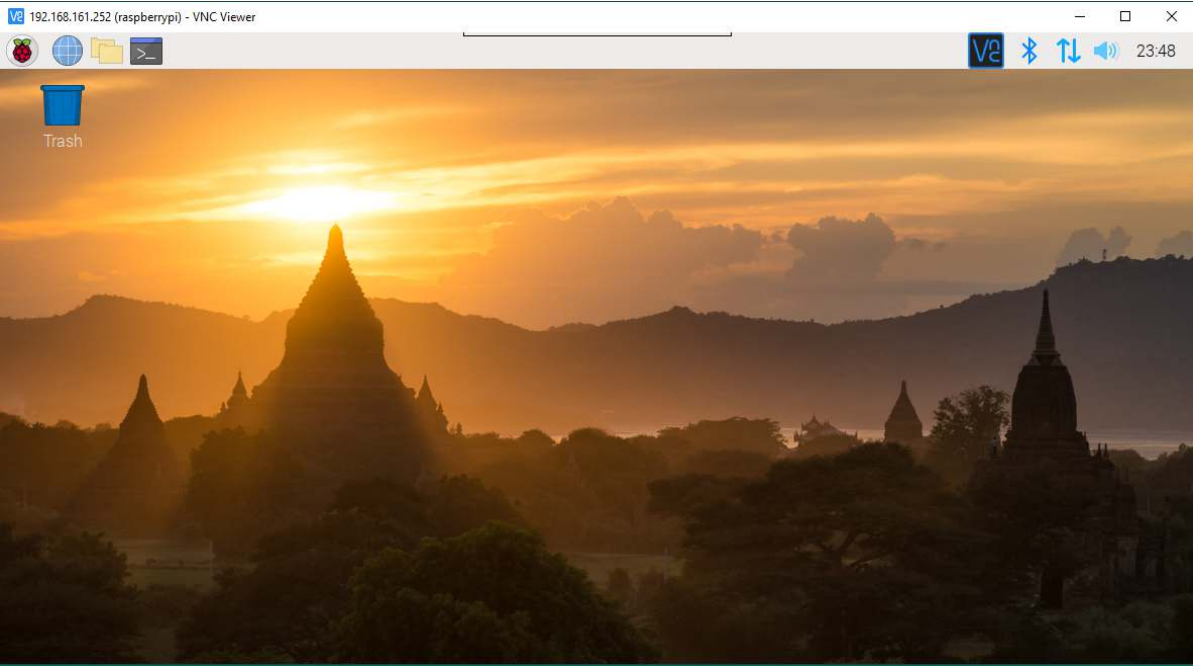
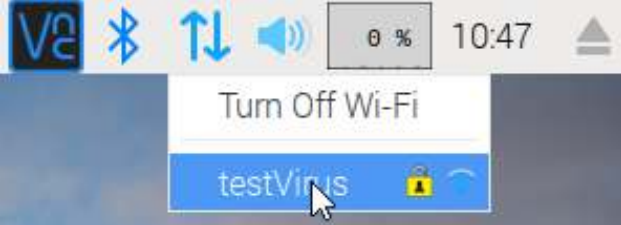


Status	Name	IP	Manufacturer	MAC address
>	D-Link.Home	192.168.1.1		
	192.168.1.49	192.168.1.49	Raspberry Pi Foundation	B8:27:EB:FC:4A:4C
	android-cabcb2fb003610d6	192.168.1.53		
	Pk007.Home	192.168.1.105		74:DF:BF:65:02:95

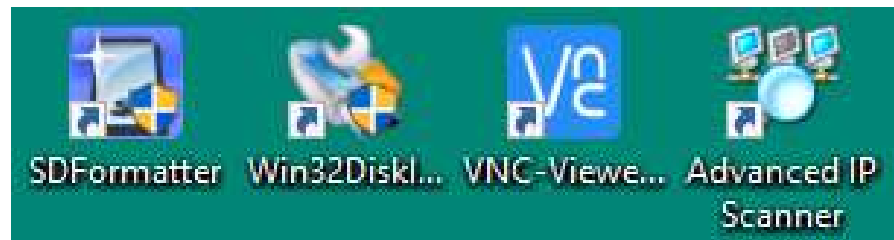
## Step 4: Remote Desktop ด้วยโปรแกรม VNC Viewer



- เรียกใช้งาน VNC Viewer

 <p>VNC Authentication dialog box showing VNC Server: 192.168.100.49::5900, Username: pi, Password: [masked], Remember password: checked, Catchphrase: Milk biology fame. Iron unit declare., Signature: 51-4e-f6-d3-8c-ef-34-87. OK button is highlighted.</p>	<p>Login: pi password: raspberry</p> <p>OK</p>
 <p>VNC Viewer window showing Raspberry Pi desktop with a sunset background, a Trash icon, and a terminal icon. The window title is '192.168.161.252 (raspberrypi) - VNC Viewer'.</p>	
<ul style="list-style-type: none"> <li>■ พร้อมใช้งาน</li> <li>■ สามารถตั้งรายละเอียด ขนาดหน้าจอ ได้ที่ Preference → Screen Configure</li> </ul>	
 <p>Wi-Fi settings overlay showing 'Turn Off Wi-Fi' and 'testVirus' buttons. The status bar shows 0% battery and 10:47 time.</p>	<p>จำเป็นต้องตั้งการเชื่อมต่อ Wifi เพื่อให้ RPi เชื่อมต่ออินเทอร์เน็ต</p>

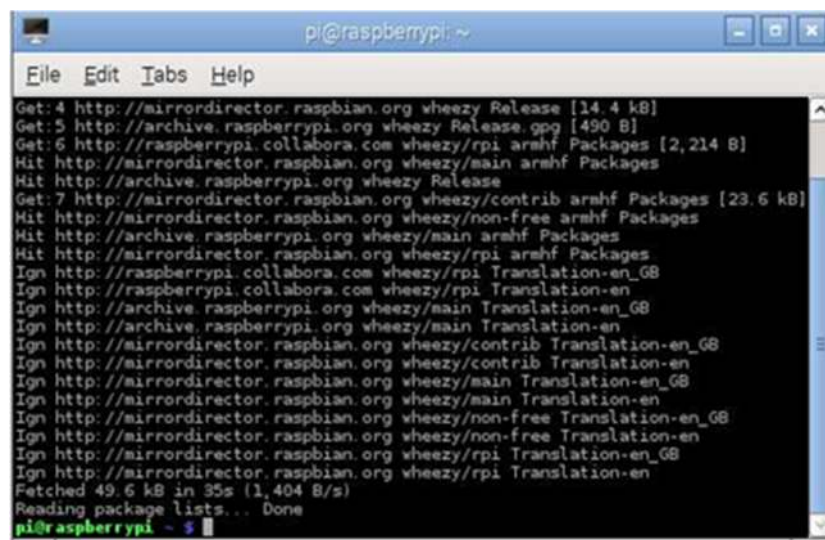
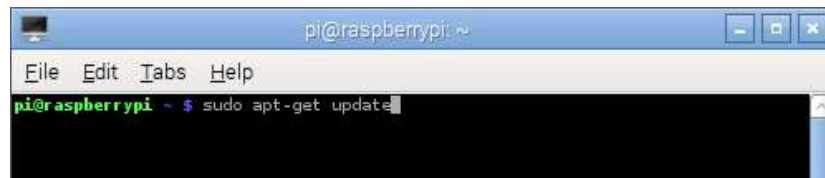
### 3.5 โปรแกรมที่จำเป็นต้องใช้งาน



### 3.6 สิ่งที่ต้องทำหลังจากการ BOOT ครั้งแรก

- อัปเดตโปรแกรมให้ใหม่ล่าสุดด้วย `sudo apt-get update`  
เข้า Terminal แล้วพิมพ์คำสั่ง Update พิมพ์เสร็จแล้วกด Enter เพื่อทำคำสั่ง

#### sudo apt-get update



- อัปเดตโปรแกรมให้ใหม่ล่าสุดด้วย `sudo apt-get upgrade`  
เข้า Terminal แล้วพิมพ์คำสั่ง, พิมพ์เสร็จแล้วกด Enter เพื่อทำคำสั่ง

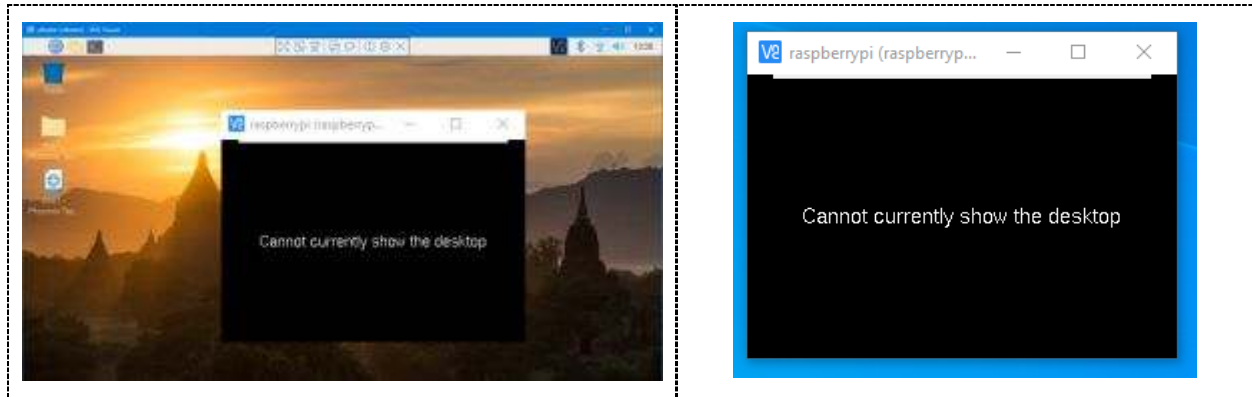
#### sudo apt-get upgrade

แล้วกด Y และ Enter เพื่อยืนยัน (ขั้นตอนนี้จะนานมาก)

### 3.7 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.

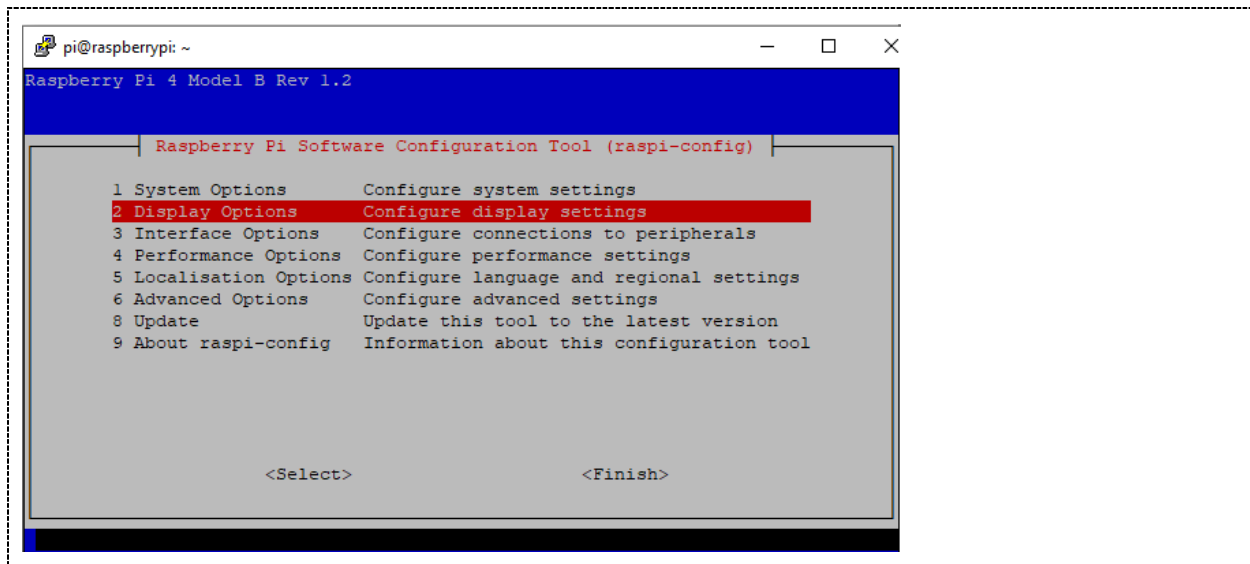
## Change Your Screen Resolution

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:

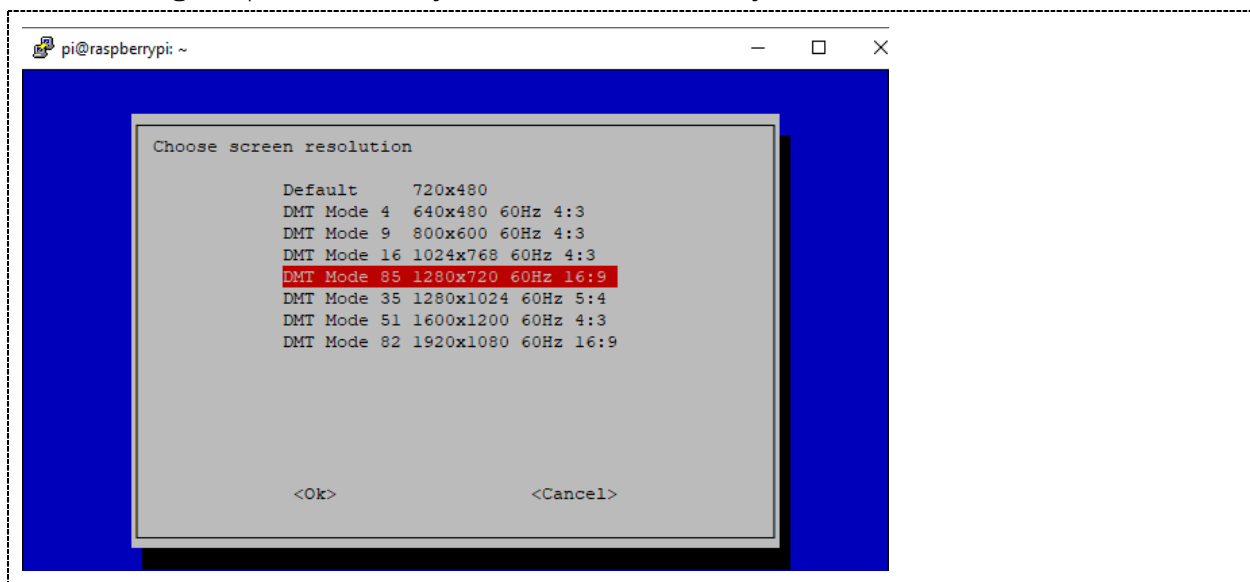
1. Launch raspi-config.

```
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.



## 3/5 -- การติดตั้ง ThingsBoard Private Server บน Raspberry Pi

<https://thingsboard.io/docs/user-guide/install/rpi/><https://thingsboard.io/docs/samples/arduino/temperature/>

## Lab302 – Installing ThingsBoard Private Server on Raspberry Pi

## 0. Raspberry Pi

**sudo apt-get update****sudo apt-get full-upgrade****node -v****npm -v**

```
pi@raspberrypi:~ $ node -v
v10.24.0
pi@raspberrypi:~ $ npm -v
5.8.0
```

**sudo npm install -g npm@latest****sudo reboot****node -v****npm -v**

```
pi@raspberrypi:~ $ node -v
v10.24.0
pi@raspberrypi:~ $ npm -v
7.21.0
```

## 1. Install Java 11 (OpenJDK)

1.1 ThingsBoard service is running on Java 11. Follow these instructions to install OpenJDK 11:

**sudo apt update****sudo apt install openjdk-11-jdk**

```
pi@raspberrypi:~ $ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-11-jdk is already the newest version (11.0.12+7-2~deb10u1).
openjdk-11-jdk set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

1.2 Please don't forget to configure your operating system to use OpenJDK 11 by default. You can configure which version is the default using the following command:

**sudo update-alternatives --config java**

1.3 You can check the installation using the following command:

**java -version**

```
pi@raspberrypi:~ $ java -version
openjdk version "11.0.12" 2021-07-20
OpenJDK Runtime Environment (build 11.0.12+7-post-Raspbian-2deb10u1)
OpenJDK Server VM (build 11.0.12+7-post-Raspbian-2deb10u1, mixed mode)
```

## 2. ThingsBoard service installation

2.1 Download installation package.

**wget** <https://github.com/thingsboard/thingsboard/releases/download/v3.3/thingsboard-3.3.deb>

```
thingsboard-3.3.deb      100%[=====] 148.35M  2.39MB/s   in 63s
2021-08-26 18:08:27 (2.37 MB/s) - 'thingsboard-3.3.deb' saved [155558024/155558024]
```

2.2 Install ThingsBoard as a service

**sudo dpkg -i thingsboard-3.3.deb**

```
pi@raspberrypi:~ $ sudo dpkg -i thingsboard-3.3.deb
Selecting previously unselected package thingsboard.
(Reading database ... 167125 files and directories currently installed.)
Preparing to unpack thingsboard-3.3.deb ...
Adding group 'thingsboard' (GID 124) ...
Done.
Unpacking thingsboard (3.3.0-1) ...
Setting up thingsboard (3.3.0-1) ...
```

## 3. Configure ThingsBoard database

3.1 PostgreSQL Installation - Instructions listed below will help you to install PostgreSQL.

*# install **wget** if not already installed:*

**sudo apt install -y wget**

*# import the repository signing key:*

**wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -**

*# add repository contents to your system:*

**RELEASE=\$(lsb\_release -cs)**

**echo "deb http://apt.postgresql.org/pub/repos/apt/ \${RELEASE}-pgdg main | sudo tee /etc/apt/sources.list.d/pgdg.list**

*# install and launch the postgresql service:*

**sudo apt update**

**sudo apt -y install postgresql**

**sudo service postgresql start**

```

update-alternatives: using /usr/share/postgresql/11/man/man1/postmaster.1.gz (postmaster.1.gz) in auto mode
Setting up postgresql (11+200+deb10u4) ...
Processing triggers for systemd (241-7~deb10u8+rpil) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rpil) ...
pi@raspberrypi:~ $ sudo service postgresql start
pi@raspberrypi:~ $ █

```

3.2 Once PostgreSQL is installed you may want to create a new user or set the password for the main user. The instructions below will help to set the password for main postgresql user

```

sudo su - postgres
psql
\password
< setpassword = myraspi >
\q

```

```

pi@raspberrypi:~ $ sudo su - postgres

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 s

postgres@raspberrypi:~$ psql
psql (11.12 (Raspbian 11.12-0+deb10u1))
Type "help" for help.

postgres=# \password
Enter new password:
Enter it again:
postgres=# \q
postgres@raspberrypi:~$ █

```

Then, press “Ctrl+D” to return to main user console and connect to the database to create thingsboard DB:

```

psql -U postgres -d postgres -h 127.0.0.1 -W
CREATE DATABASE thingsboard;
\q

```

```

postgres@raspberrypi:~$ psql -U postgres -d postgres -h 127.0.0.1 -W
Password:
psql (11.12 (Raspbian 11.12-0+deb10u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits
Type "help" for help.

postgres=# CREATE DATABASE thingsboard;
CREATE DATABASE
postgres=# \q
postgres@raspberrypi:~$ █

```

Ctrl+D for Exit

```
postgres@raspberrypi:~$ logout
pi@raspberrypi:~$
```

### 3.3 ThingsBoard Configuration - Edit ThingsBoard configuration file

`sudo nano /etc/thingsboard/conf/thingsboard.conf`

Add the following lines to the configuration file. Don't forget to replace "PUT\_YOUR\_POSTGRES\_PASSWORD\_HERE" with your real postgres user password:

```
# DB Configuration
export DATABASE_ENTITIES_TYPE=sql
export DATABASE_TS_TYPE=sql
export SPRING_JPA_DATABASE_PLATFORM=org.hibernate.dialect.PostgreSQLDialect
export SPRING_DRIVER_CLASS_NAME=org.postgresql.Driver
export SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/thingsboard
export SPRING_DATASOURCE_USERNAME=postgres
export SPRING_DATASOURCE_PASSWORD=PUT_YOUR_POSTGRES_PASSWORD_HERE
export SPRING_DATASOURCE_MAXIMUM_POOL_SIZE=5
# Specify partitioning size for timestamp key-value storage.
# Allowed values: DAYS, MONTHS, YEARS, INDEFINITE.
export SQL_POSTGRES_TS_KV_PARTITIONING=MONTHS
```

```
GNU nano 3.2 /etc/thingsboard/conf/thingsboard.conf

export JAVA_OPTS="$JAVA_OPTS -XX:+UseG1GC -XX:MaxGCPauseMillis=500 -XX:+UseStringD
export LOG_FILENAME=thingsboard.out
export LOADER_PATH=/usr/share/thingsboard/conf,/usr/share/thingsboard/extensions
export SQL_DATA_FOLDER=/usr/share/thingsboard/data/sql

# DB Configuration
export DATABASE_ENTITIES_TYPE=sql
export DATABASE_TS_TYPE=sql
export SPRING_JPA_DATABASE_PLATFORM=org.hibernate.dialect.PostgreSQLDialect
export SPRING_DRIVER_CLASS_NAME=org.postgresql.Driver
export SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/thingsboard
export SPRING_DATASOURCE_USERNAME=postgres
export SPRING_DATASOURCE_PASSWORD=myraspi
export SPRING_DATASOURCE_MAXIMUM_POOL_SIZE=5
# Specify partitioning size for timestamp key-value storage. Allowed values: DAYS,
export SQL_POSTGRES_TS_KV_PARTITIONING=MONTHS
```

Ctrl+X, Save and Exit

## 4. Choose ThingsBoard queue service

<div data-bbox="209 260 695 432" style="border: 1px solid blue; padding: 10px; text-align: center;"> <b>In Memory</b>          (built-in and default)       </div>	<ul style="list-style-type: none"> <li>• In Memory queue is built-in and enabled by default. No additional configuration steps required.</li> </ul>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------

## 5. Memory update for slow machines (1GB of RAM)

## 5.1 Edit ThingsBoard configuration file

```
sudo nano /etc/thingsboard/conf/thingsboard.conf
```

## 5.2 Add the following lines to the configuration file.

```
# Update ThingsBoard memory usage and restrict it to 256MB in
/etc/thingsboard/conf/thingsboard.conf
export JAVA_OPTS="$JAVA_OPTS -Xms256M -Xmx256M"
```

```
# Update ThingsBoard memory usage and restrict it to 256MB in /etc/thingsboa
export JAVA_OPTS="$JAVA_OPTS -Xms256M -Xmx256M"
```

Ctrl+X, Save and Exit



## 6. Run installation script

6.1 Once ThingsBoard service is installed and DB configuration is updated, you can execute the following script:

# --loadDemo option will load demo data: users, devices, assets, rules, widgets.

**sudo /usr/share/thingsboard/bin/install/install.sh --loadDemo**

```
pi@raspberrypi:~ $ sudo /usr/share/thingsboard/bin/install/install.sh --loadDemo

  _____          _
 / ____ \        / \  ( )
/ /    \ \      / \  ( )
/ /____\ \ ____/ \  ( )
/_____/_____/_____\

=====
:: ThingsBoard ::      (v3.3.0)
=====
```

```
Starting ThingsBoard Installation...
Installing DataBase schema for entities...
Installing SQL DataBase schema part: schema-entities.sql
Installing SQL DataBase schema indexes part: schema-entities-idx.sql
Installing SQL DataBase schema PostgreSQL specific indexes part: sch
Installing DataBase schema for timeseries...
Installing SQL DataBase schema part: schema-ts-psql.sql
Successfully executed query: CREATE TABLE IF NOT EXISTS ts_kv_indefi
Loading system data...
Loading demo data...
Installation finished successfully!
ThingsBoard installed successfully!
```

## 7. Start ThingsBoard service

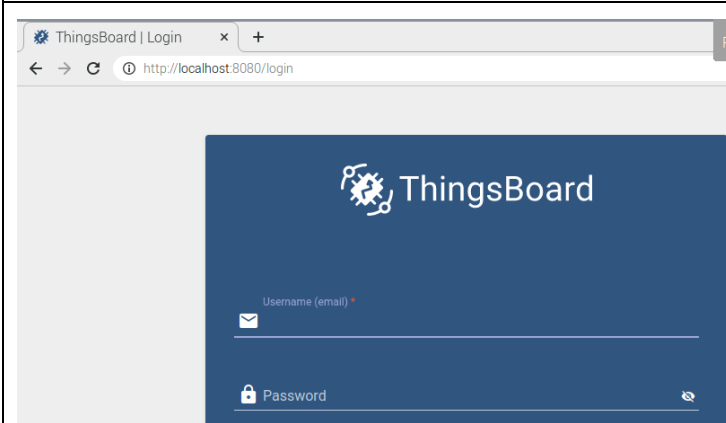
7.1 Execute the following command to start ThingsBoard:

```
sudo service thingsboard start
```

```
pi@raspberrypi:~$ sudo service thingsboard start
pi@raspberrypi:~$
```

7.2 Once started, you will be able to open Web UI using the following link:

<http://localhost:8080/>



7.3 The following default credentials are available if you have specified `--loadDemo` during execution of the installation script:

- System Administrator: `sysadmin@thingsboard.org` / `sysadmin`
- Tenant Administrator: `tenant@thingsboard.org` / `tenant`
- Customer User: `customer@thingsboard.org` / `customer`

You can always change passwords for each account in account profile page.

Please allow up to 240 seconds for the Web UI to start. This is applicable only for slow machines with 1-2 CPUs or 1-2 GB RAM.

## Troubleshooting

ThingsBoard logs are stored in the following directory:

```
/var/log/thingsboard
```

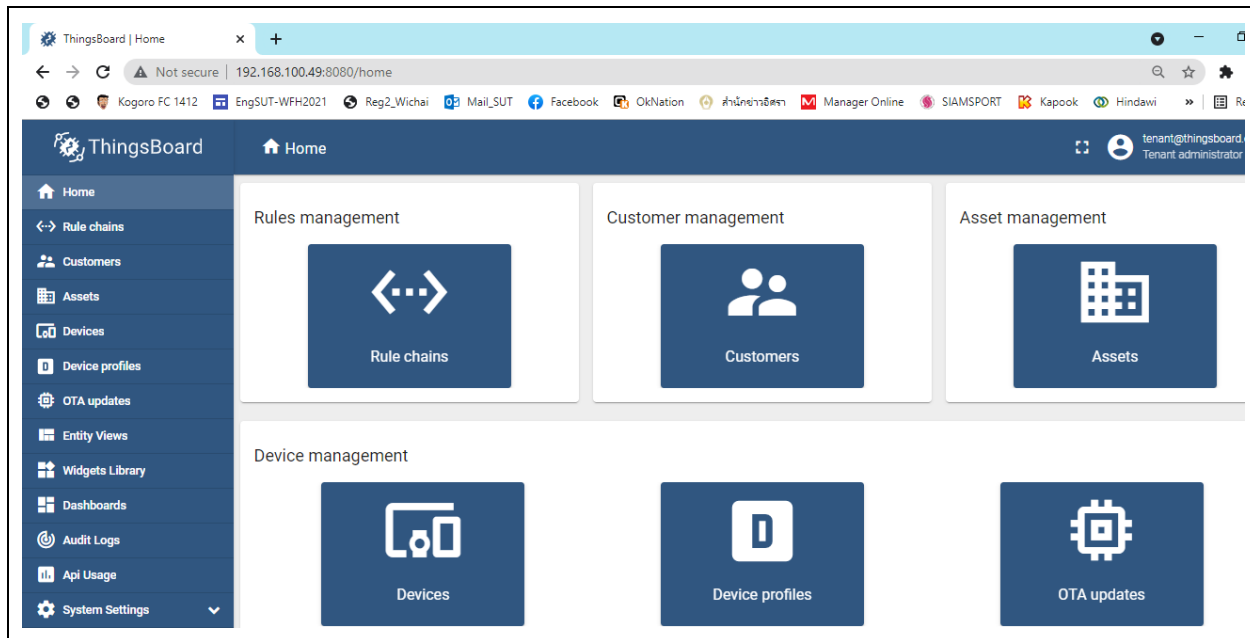
Copy to clipboard

You can issue the following command in order to check if there are any errors on the backend side:

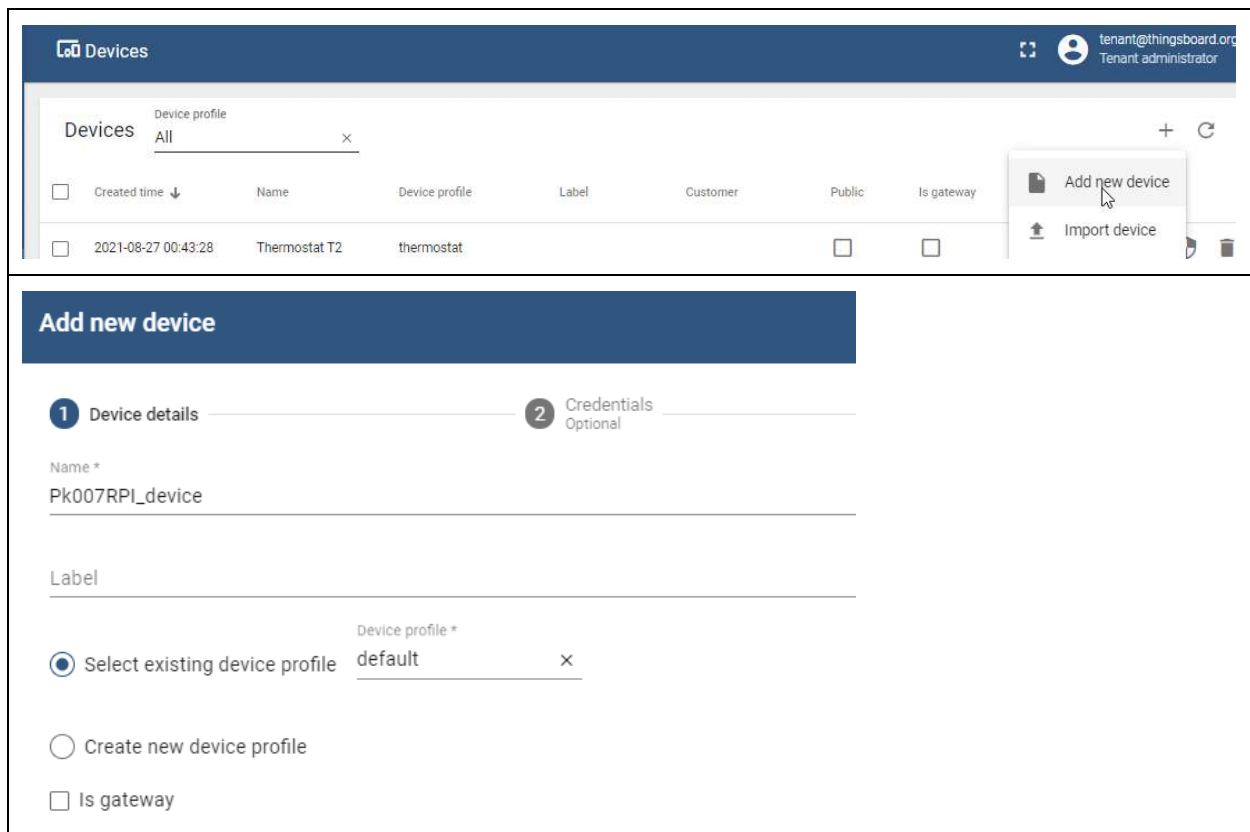
```
cat /var/log/thingsboard/thingsboard.log | grep ERROR
```

## Lab303 – Testing ThingsBoard Private Server on Raspberry Pi

1. Login ที่ IP:8080 ด้วย Tenant Administrator: tenant@thingsboard.org / tenant



2. Create Device and Get Token Key



Pk007RPI\_device

Device details

Details

Attributes

Latest telemetry

Alarms

Events

F

Make device public

Assign to customer

Manage credentials

Delete

Copy device Id

Copy access token

TSRQ5fPHm5Io4fEKLNGd

### 3. Data Simulation Test

←→ Rule chains

tenant@thingsboard.org  
Tenant administrator

Rule chains

+

↺

🔍

<input type="checkbox"/>	Created time ↓	Name
<input type="checkbox"/>	2021-08-27 00:43:24	Thermostat

Create new rule chain


Import rule chain

Add Rule Chain

Name \*

Pk007RPI\_Rule

☐ Debug mode

 generator

Details Events Help

Name \*

gen\_RPI\_AA

☐ Debug mode

Message count (0 - unlimited) \*

0

Period in seconds \*

1

Originator Type

Device

Device

▼

Pk007RPI\_device

×

Generate

function Generate(prevMsg, prevMetadata, prevMsgType) {

Tidy

```

1 var msg = {
2   temperature: (Math.random()*10+30).toFixed(2),
3   humidity: (Math.random()*10+70).toFixed(2) };
4 var metadata = { data: 40 };
5 var msgType = "POST_TELEMETRY_REQUEST";
6
7 return { msg: msg, metadata: metadata, msgType: msgType };

```

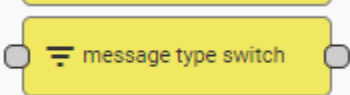
```

var msg = {
  temperature: (Math.random()*10+30).toFixed(2),
  humidity: (Math.random()*10+70).toFixed(2) };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";

return { msg: msg, metadata: metadata, msgType: msgType };

```





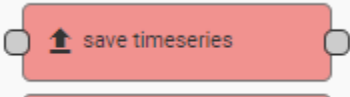
## msg\_RPI\_AA

Filter - message type switch

Details Help

Name \*  
msg\_RPI\_AA

Description

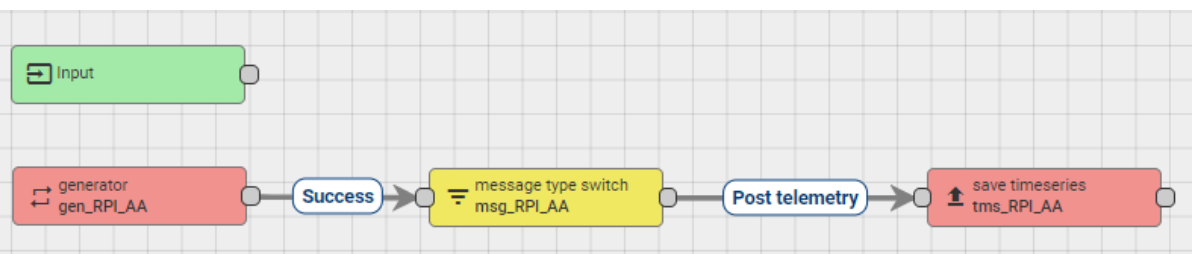


## Add rule node: save timeseries

Name \* ☐ Debug mode  
tms\_RPI\_AA

Default TTL in seconds \*  
0

Description



```
graph LR; Input[Input] --> gen_RPI_AA[generator gen_RPI_AA]; gen_RPI_AA -- Success --> msg_RPI_AA[message type switch msg_RPI_AA]; msg_RPI_AA -- Post telemetry --> tms_RPI_AA[save timeseries tms_RPI_AA];
```

## 4. ตรวจสอบข้อมูลที่ Device

The screenshot shows the ThingsBoard IoT Platform interface. On the left, the 'Devices' menu item is highlighted. The main panel displays a list of devices, with 'Pk007RPI\_device' selected. The 'Latest telemetry' tab is active, showing the following data:

Last update time	Key	Value
2021-08-27 11:52:47	humidity	77
2021-08-27 11:52:47	temperature	42

## 5. สร้าง Dashboard

The screenshot shows the ThingsBoard IoT Platform interface. On the left, the 'Dashboards' menu item is highlighted. The main panel displays a list of dashboards. The 'Create new dashboard' button is highlighted. Below, the 'Add Dashboard' form is shown with the title 'dashboard\_Pk007RP'.

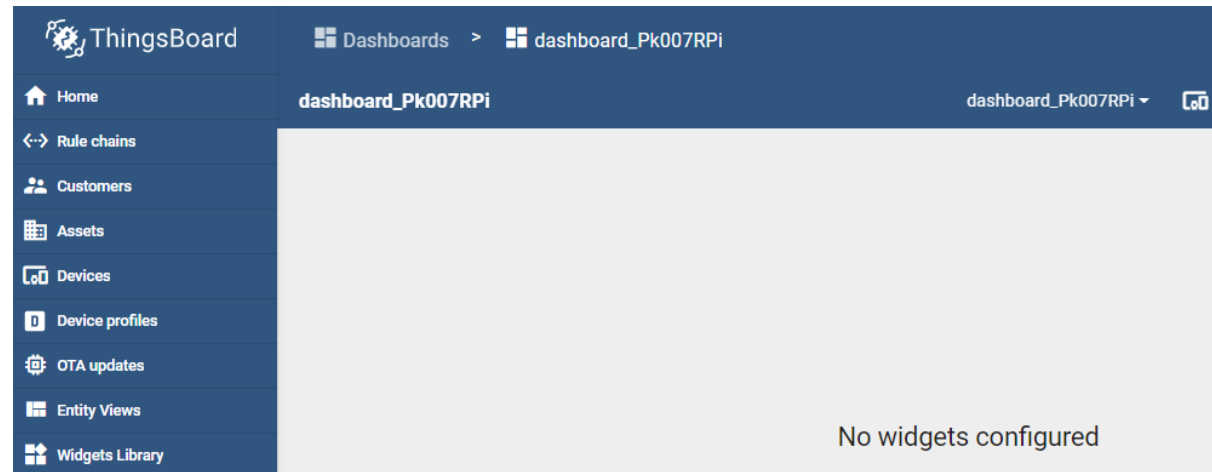
**Add Dashboard**

Title \*

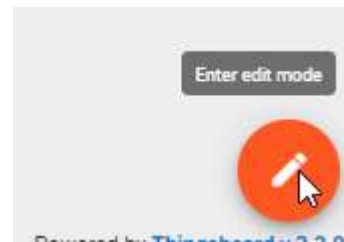
dashboard\_Pk007RP

Description

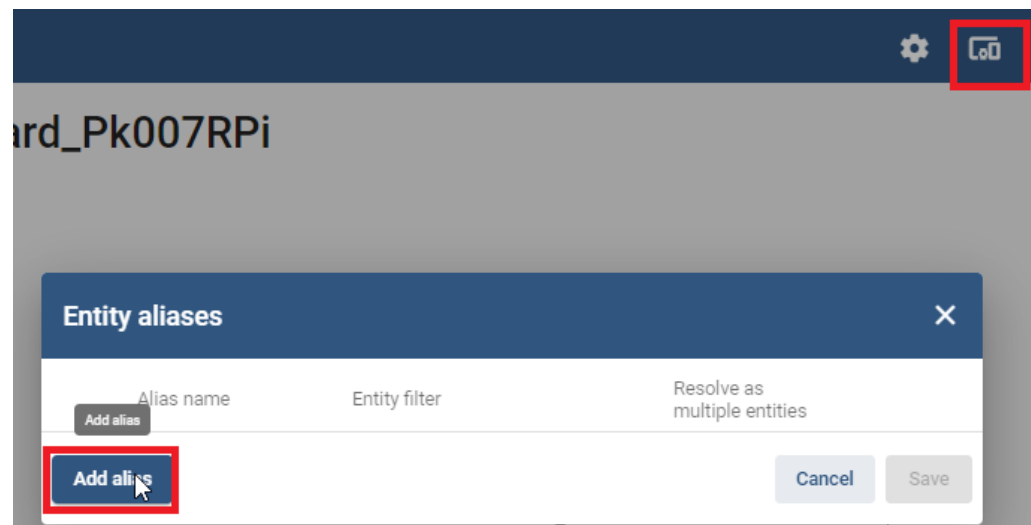
## Open Dashboard



## Edit



## Add Alias



alias\_Pk007RPI

Resolve as multiple entities

Filter type \*

Single entity

Type \*

Device

Device \*

Pk007RPI\_device

Cancel

Add

Save

✓

red by [Thingst](#)

Add Time Series widget

Type

Parameters

= 1.

Entity

Entity alias \*

alias\_Pk007RPI

Filter

humidity: humidity

temper... : tempera...

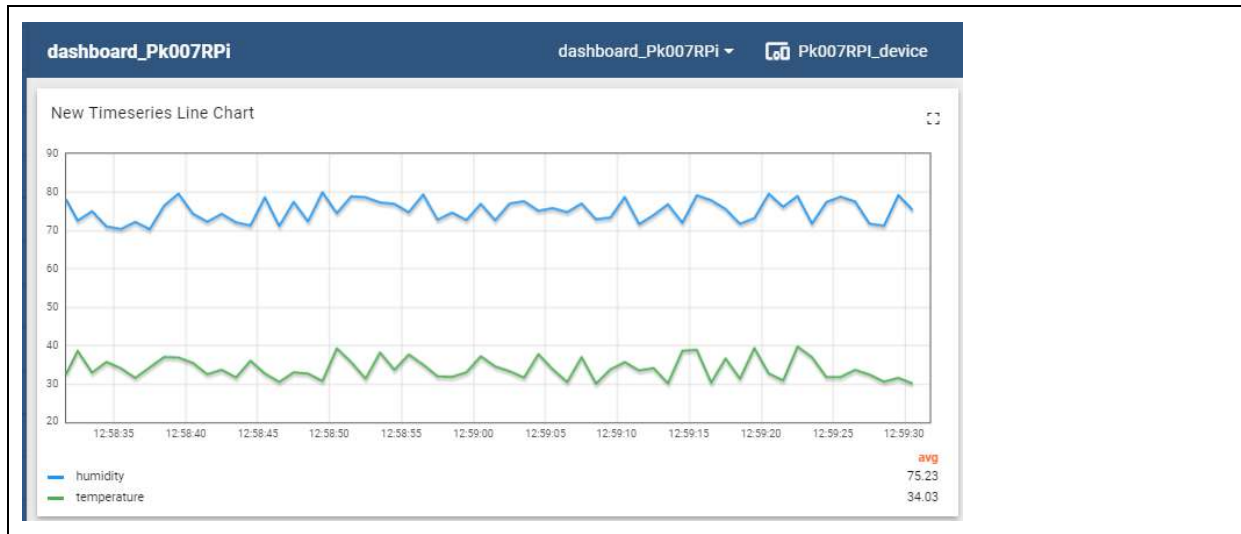
humidity

temperature

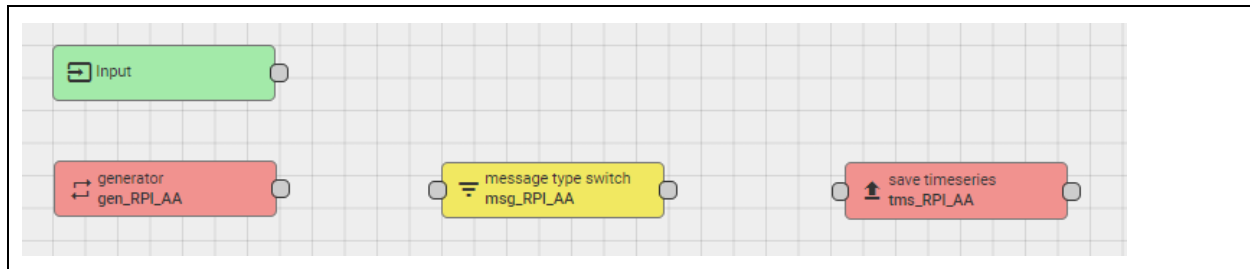
+ Add

Cancel

Add



## 6. ปิดข้อมูลทดสอบที่สร้างจาก Rule Chain



## 7. ส่งข้อมูลจาก ESP32

**ThingsBoard**

by ThingsBoard Team Version 0.4.0 **INSTALLED**

**ThingsBoard library for Arduino.** A library for connecting to the ThingsBoard IoT platform. Thin wrapper on PubSubClient.

[More info](#)

**ArduinoHttpClient**

by Arduino Version 0.4.0 **INSTALLED**

**[EXPERIMENTAL] Easily interact with web servers from Arduino, using HTTP and WebSocket's.** This library can be used for HTTP (GET, POST, PUT, DELETE) requests to a web server. It also supports exchanging messages with WebSocket servers. Based on Adrian McEwen's HttpClient library.

[More info](#)

**ArduinoJson**

by Benoit Blanchon Version 6.18.3 **INSTALLED**

**A simple and efficient JSON library for embedded C++.** ArduinoJson supports ✓ serialization, ✓ deserialization, ✓ MessagePack, ✓ fixed allocation, ✓ zero-copy, ✓ streams, ✓ filtering, and more. It is the most popular Arduino library on GitHub ♥♥♥♥♥. Check out arduinojson.org for a comprehensive documentation.

[More info](#)



```

// Add Library "ThingsBoard by ThingsBoard Team -- V 0.4.0"
// Add Library "ArduinoHttpClient by Arduino -- V 0.4.0"
// Add Library "ArduinoJson by Benoit Blanchon - V6.18.3"

#include "ThingsBoard.h"
#include <WiFi.h>
#define WIFI_AP "Test1234"
#define WIFI_PASSWORD "0816601929"
#define TOKEN "TSRQ5fPHm5Io4fEKLNGd"
#define THINGSBOARD_SERVER "192.168.1.30"
#define THINGSBOARD_PORT 1883
#define SERIAL_DEBUG_BAUD 115200

// Initialize ThingsBoard client
WiFiClient espClient;
// Initialize ThingsBoard instance
ThingsBoard tb(espClient);
// the Wifi radio's status
int status = WL_IDLE_STATUS;

void setup() {
  // initialize serial for debugging
  Serial.begin(SERIAL_DEBUG_BAUD);
  WiFi.begin(WIFI_AP, WIFI_PASSWORD);
  InitWiFi();
}

void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    reconnect();
  }

  if (!tb.connected()) {
    // Connect to the ThingsBoard
    Serial.print("Connecting to: "); Serial.print(THINGSBOARD_SERVER);
    Serial.print(" with token "); Serial.println(TOKEN);
    if (!tb.connect(THINGSBOARD_SERVER, TOKEN, THINGSBOARD_PORT)) {
      Serial.println("Failed to connect");
      return;
    }
  }

  Serial.print("Sending data...");

  // Uploads new telemetry to ThingsBoard using MQTT.
  // See https://thingsboard.io/docs/reference/mqtt-api/#telemetry-upload-api
  // for more details
  float xTempp = random(2000, 5000) / 100.0;
  float xHdmd = random(6000, 8000) / 100.0;
  Serial.print(xTempp, 2); Serial.print(",");
  Serial.print(xHdmd, 2); Serial.println();

  //tb.sendTelemetryInt("temperature", xTempp);
  //tb.sendTelemetryInt("humidity", xTempp);
  tb.sendTelemetryFloat("temperature", xTempp);
  tb.sendTelemetryFloat("humidity", xHdmd);

  tb.loop();
  delay(5000);
}

void InitWiFi()
{
  Serial.println("Connecting to AP ...");
  // attempt to connect to WiFi network

  WiFi.begin(WIFI_AP, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to AP");
}

void reconnect() {
  // Loop until we're reconnected
  status = WiFi.status();
  if (status != WL_CONNECTED) {
    WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("Connected to AP");
  }
}

```

.....Connected to AP

Connecting to: 192.168.1.30 with token TSRQ5fPHm5Io4fEKLNGd

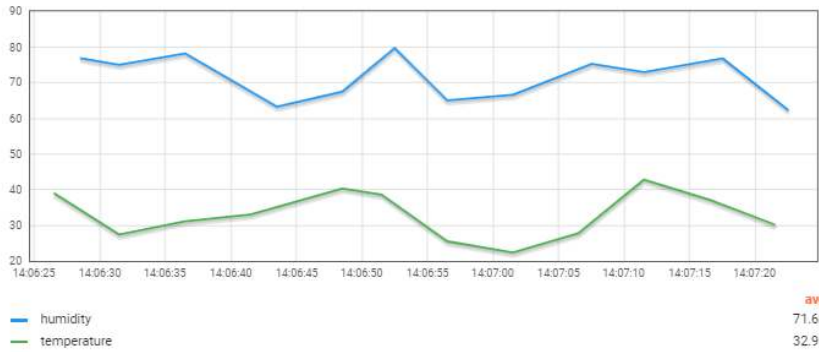
Sending data...21.45,70.98

Sending data...31.16,72.78

title "

## dashboard\_Pk007RPI

New Timeseries Line Chart



## 4/5 -- การโปรแกรมเพื่อใช้งาน Rule Chains, MAP Widget และ Multilayer dashboard

More Data

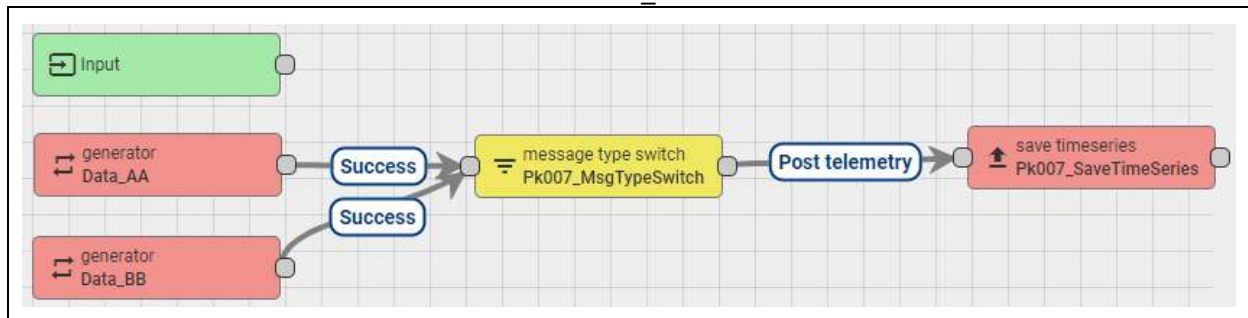
- <https://thingsboard.io/smart-metering/>
- ThingsBoard documentation - <https://thingsboard.io/docs>
- ThingsBoard Dashboard development guide. Part 1 <https://youtu.be/Oio1YnQjIwA>
- ThingsBoard Dashboard development guide. Part 2 <https://youtu.be/mqWEGs1Z2BQ>
- ThingsBoard Dashboard development guide. Part 3 <https://youtu.be/TyUKx5rNJSs>
- Emulator
  - Thermostat <https://gist.github.com/Terny22/2b380427d1a9af823dac03d14f3379ff>
  - Energy meter <https://gist.github.com/Terny22/b9e050e2c561cd7f109a2836f905085f>
  - Water meter <https://gist.github.com/Terny22/50e65c427be349f89c9c8150dfd20af4>

## Lab304 – Using Rule Chains, MAP Widget and Multilayer dashboard

## 1. Create 2 Device

Devices <small>Device profile</small>			
<input type="checkbox"/>	Created time ↓	Name	Device profile
<input type="checkbox"/>	2021-06-07 12:03:24	Pk_Station_BB	default
<input type="checkbox"/>	2021-06-07 12:03:00	Pk_Station_AA	default

## 2. Generate Data → Add Rules Chains → Pk007\_Cahins



## Test Generator Function – Data\_AA → Device → Pk\_Station\_AA

Name \*  ☐ Debug mode

Message count (0 - unlimited) \*

Period in seconds \*

Originator Type: Device Device:

Generate

```
function Generate(prevMsg, prevMetadata, prevMsgType) {
  1 var msg = {
  2   station_name : "SUT F11",
  3   latitude : 14.876488085551973,
  4   longitude : 102.01498458834561,
  5   temperature: (Math.random()*10+40).toFixed(2),
  6   humidity: (Math.random()*10+80).toFixed(2) };
  7 var metadata = { data: 40 };
  8 var msgType = "POST_TELEMETRY_REQUEST";
  9
  10 return { msg: msg, metadata: metadata, msgType: msgType };
}
```

```
var msg = {
  station_name : "SUT F11",
  latitude : 14.876488085551973,
  longitude : 102.01498458834561,
  temperature: (Math.random()*10+40).toFixed(2),
  humidity: (Math.random()*10+80).toFixed(2) };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";

return { msg: msg, metadata: metadata, msgType: msgType };
```

## Test Generator Function – Data\_BB → Device → Pk\_Station\_BB

```
var msg = {
  station_name : "SurathamPitak School",
  latitude : 14.91002308772653,
  longitude : 102.0703244094149,
  temperature: (Math.random()*10+30).toFixed(2),
  humidity: (Math.random()*10+70).toFixed(2) };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";

return { msg: msg, metadata: metadata, msgType: msgType };
```

## 3. Check Data send to Device

Pk\_Station\_AA

Device details

<

Details

Attributes

Latest telemetry

Alarms

Events

Latest telemetry

<input type="checkbox"/>	Last update time	Key ↑	Value
<input type="checkbox"/>	2021-06-07 12:06:18	humidity	85.85
<input type="checkbox"/>	2021-06-07 12:06:18	latitude	14.876488085551973
<input type="checkbox"/>	2021-06-07 12:06:18	longitude	102.01498458834561
<input type="checkbox"/>	2021-06-07 12:06:18	station_name	SUT F11




## 4. Create New Dashboard → Name= Pk007\_TB\_Part2

Pk007\_TB-Part2

Pk007\_TB-Part2 ▾

No widgets configured

## 5. Edit → Entity Alias, Add1, Add2, Add3



Entity aliases

Alias name \*

Pk\_Station\_AA

Resolve as multiple entities

☐

Filter type \*

Single entity

▼

Type \*

Device

▼

Device \*

Pk\_Station\_AA

×

Alias name \*

Pk\_Station\_BB

Resolve as multiple entities

☐

Filter type \*

Single entity

▼

Type \*

Device

▼

Device \*

Pk\_Station\_BB

×

Alias name \*

All\_Node

Resolve as multiple entities

☒

Filter type \*

Entity list

▼

Type \*

Device

▼

Pk\_Station\_AA

×







Pk\_Station\_BB

×

Entity list



### Entity aliases

	Alias name	Entity filter	Resolve as multiple entities	
1.	<input type="text" value="Pk_Station_BB"/>	One device	<input type="checkbox"/>	 
2.	<input type="text" value="Pk_Station_AA"/>	One device	<input type="checkbox"/>	 
3.	<input type="text" value="All_Node"/>	List of 2 devices	<input checked="" type="checkbox"/>	 

Add alias

CancelSave

## 6. Add MAP Widget with Data

The image shows two screenshots from the ThingsBoard IoT Platform interface.

The top screenshot displays the 'Datasources' configuration window. It has a table with columns 'Type' and 'Parameters'. The first row is selected, showing 'Entity' as the type and 'All\_Node' as the entity alias. To the right, there are five data source configurations, each with a color-coded circle, a checkmark, and a label: 'humidity: humidity' (blue), 'latitude: latitude' (green), 'longitude: longitude' (red), 'station... : station\_...' (yellow), and 'temper... : tempera...' (dark blue). Each configuration has edit and delete icons. At the bottom right are 'Cancel' and 'Add' buttons.

The bottom screenshot shows a 'New Google Maps' widget. It displays a map with two red location pins. The first pin is labeled 'Pk\_Station\_BB' and has a tooltip showing its coordinates: 'Latitude: 14.9100231' and 'Longitude: 102.0703244', along with 'Temperature: 33.18 °C'. The second pin is labeled 'Pk\_Station\_AA'. The map includes standard Google Maps controls like zoom in (+) and zoom out (-) buttons. Text overlays on the map include 'development purposes only' and 'For development purposes only'. At the bottom, it says 'Map data ©2021 Terms of Use'.

## 7. Add Entity Admin Widget – Device Admin Table with Data

Type

Parameters

= 1. Entity

Entity alias \*

All\_Node





Filter

= humidity: humidity

= temper... : tempera...

= station... : station\_...

Device admin table


Entity name ↑	Entity type	humidity	temperature	station_name	
Pk_Station_AA	Device	82.32	47.16	SUT F11	 
Pk_Station_BB	Device	79.54	30.39	SurathamPitak School	 

Items per page: 10





1 – 2 of 2

|< < > >|

New Google Maps



Device admin table

Entity name ↑	Entity type	humidity	temperature	station_name	
Pk_Station_AA	Device	86.53	48.15	SUT F11	 
Pk_Station_BB	Device	76.44	39.89	SurathamPitak School	 

Items per page: 10

1 – 2 of 2

|< < > >|

8. Edit Mode → manage dashboard state → Add 1, Add 2

Title \*

Manage dashboard states

PK007\_TB-Part2

Name \*

#{entityName}

State Id \*

SUT\_Data

☐ Root state

Cancel

Add

, #{entityName}

Name \*

#{entityName}

State Id \*

STP\_Data






☐ Root state

### Manage dashboard states

Dashboard states

+

🔍

Name ↑	State Id	Root state		
\$(entityName)	SUT_Data	<input type="checkbox"/>		
\$(entityName)	STP_Data	<input type="checkbox"/>		
Pk007_TB-Part2	default	<input checked="" type="checkbox"/>		

Items per page: 5

1 – 3 of 3

<

>

Cancel

Save

## 9. Edit Mode → Set SUT\_Data

STP\_Data

SUT\_Data

default

= 1.

Entity

Entity alias \*

Pk\_Station\_AA

Filter

= humidity: humidity

= temper... : tempera...

Maximum 1 timeseries/attribute is allowed.

New Timeseries Line Chart



Series	avg
humidity	85.15
temperature	44.63

station\_name

SUT F11



## 10. Edit Mode → Set STP\_Data

The screenshot displays the ThingsBoard IoT Platform interface in Edit Mode, showing the configuration of a new Timeseries Line Chart for STP\_Data.

**Top Panel:** A dropdown menu is open, showing the selection of **STP\_Data** (highlighted) from the options **SUT\_Data** and **default**.

**Configuration Panel:** The configuration is set to **Type: Entity** and **Parameters: Pk\_Station\_BB**. The **Entity alias \*** is set to **Pk\_Station\_BB**. The **Filter** field is empty.

**Timeseries Configuration:** Two timeseries are configured:










- humidity: humidity** (blue line)
- temper... : tempera...** (green line)

**Chart View:** A **New Timeseries Line Chart** is displayed, showing the data for **humidity** (blue line) and **temperature** (green line) over time. The chart includes a legend and a table of average values:

Attribute	avg
humidity	74.84
temperature	34.92




**Station Information:** The station name is **station\_name**, which is **SurathamPitak School**.

## 11. Set Admin Table to Switch Page → Remove 3 Action

Action cell button	Edit device		Custom action (with HTML template)	 
Action cell button	Delete device		Custom action	 
Widget header button	Add device		Custom action (with HTML template)	 





Device admin table

Entity name ↑	Entity type	humidity	temperature	station_name
Pk_Station_AA	Device	80.53	40.11	SUT F11
Pk_Station_BB	Device	79.02	30.81	SurathamPitak School

Items per page: 10

1 – 2 of 2

## 12. Set Admin Table to Switch Page → Add On Cell Click

Action source \*

Action cell button

Name \*

Cell\_AdminTable

Icon

play\_arrow

Type \*

Custom action

function (\$event, widgetContext, entityId, entityName, additionalParams, entityLabel) { Tidy }

```

1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext
  .servicesMap.get('deviceService'));
3
4 deviceService.getDevice(entityId.id).subscribe
  (function(device) {
5   if (device.name == 'Pk_Station_AA') {

```

Cancel

Save

```

var $injector = widgetContext.$scope.$injector;
let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));

deviceService.getDevice(entityId.id).subscribe(function(device) {
  if (device.name == 'Pk_Station_AA') {
    openDashboardState('SUT_Data',device)
  }
  else if(device.name == 'Pk_Station_BB') {
    openDashboardState('STP_Data',device)
  }
});

function openDashboardState(stateId,device) {
  var params = {
    entityId: entityId,
    entityName: device.name
  };
  widgetContext.stateController.openState(stateId, params, false);
}

```

```

1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'))
3
4 deviceService.getDevice(entityId.id).subscribe(function(device) {
5     if (device.name == 'Pk_Station_AA') {
6         openDashboardState('SUT_Data',device)
7     }
8     else if(device.name == 'Pk_Station_BB') {
9         openDashboardState('STP_Data',device)
10    }
11 });
12
13 function openDashboardState(stateId,device) {
14     var params = {
15         entityId: entityId,
16         entityName: device.name
17     };
18     widgetContext.stateController.openState(stateId, params, false);
19 }

```

### 13. Set Admin Table to Switch Page → Add On Row Click

Action source \*

On row click

Name \*

Row\_AdminTable

Icon

...

more\_horiz

Type \*

Update current dashboard state

Target dashboard state

☐ Open right dashboard layout (mobile view)
 ☒ Set entity from widget

State entity parameter name

By default

Cancel

Save

Actions			
Action source ↑	Name	Icon	Type
Action cell button	Cell_AdminTable	▶	Custom action
On row click	Row_AdminTable	***	Update current dashboard state

## 14. Test with Admin Click

Device admin table					🔍	☰	🗖
Entity name ↑	Entity type	humidity	temperature	station_name			
Pk_Station_AA	Device	88.72	44.82	SUT F11	Cell_AdminTable		
Pk_Station_BB	Device	79.4	33.35	SurathamPitak School	▶		

Items per page: 10 ▾ 1 – 2 of 2 |< < > >|

## 15. Edit Mode Google MAP → Advance Tab → Set Edge of node name label

## New Google Maps

Google Maps

Data Settings **Advanced** Actions

Default map type

Roadmap

Roadmap

**Satellite**

Hybrid

Terrain

☒ Show label

Label (pattern examples: '{entityName}', '{entityName}: (Text \${keyName} units.)' )



color: #fff; padding-left: 5px; padding-right: 5px; padding-top: 3px; padding-bottom: 3px;">\${entityName}<

<div style="position: relative; white-space: nowrap; text-align: center; font-size: 14px; top: 5px;"><span style="border: 2px solid #000; border-radius: 10px; color: #000; background-color: #fff; padding-left: 5px; padding-right: 5px; padding-top: 3px; padding-bottom: 3px;">\${entityName}</span> </div>

New Google Maps



## 16. Edit Mode Google MAP → Advance Tab → Set Data Display of Node

New Google Maps

Google Maps

Data Settings **Advanced** Actions

☒ Auto-close tooltips

Tooltip (for ex. 'Text \${keyName} units.' or <link-act name='my-action'>Link text</link-act>)

<b>\${entityName}</b><br/><br/>

<b>Station:</b> \${station\_name}<br/>

---

☐ Use tooltip function

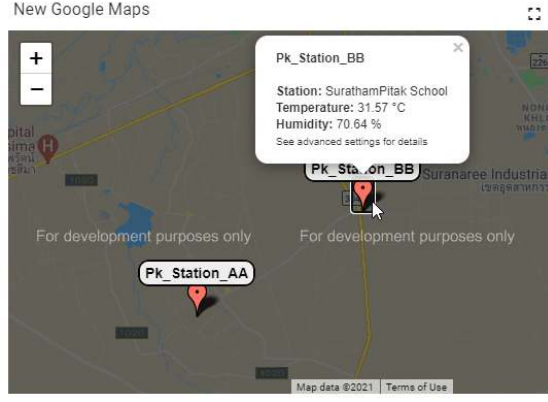
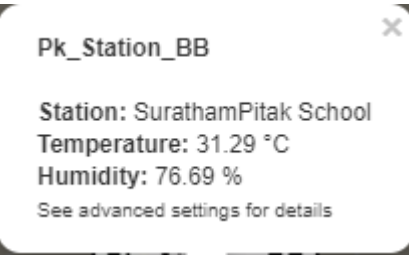
<b>\${entityName}</b><br/><br/>

<b>Station:</b> \${station\_name}<br/>

<b>Temperature:</b> \${temperature} °C<br/>

<b>Humidity:</b> \${humidity} %<br/>

<small>See advanced settings for details</small>

## 17. Edit Mode Google MAP → Advance Tab → Set Link from Node to New Dashboard

☒ Auto-close tooltips

Tooltip (for ex. 'Text \${keyName} units.' or <link-act name='my-action'>Link text</link-act>)

```

<b>${entityName}</b><br/><br/>
<b>Station:</b> ${station_name}<br/>
    
```

☐ Use tooltip function

```

<b>${entityName}</b><br/><br/>
<b>Station:</b> ${station_name}<br/>
<b>Temperature:</b> ${temperature} °C<br/>
<b>Humidity:</b> ${humidity} %<br/>
<link-act name='Sensor_Details'>Sensor Details</link-act>
    
```

## 18. Edit Mode Google MAP → Action Tab → Set Action &gt;&gt; Tool\_Action &gt; Add Update Current State

Action source \*
 

Tooltip tag action

Name \*
 

Tool\_Action

...

Icon

more\_horiz

Type \*
 

Update current dashboard state

Target dashboard state



☐ Open right dashboard layout (mobile view)

☒ Set entity from widget

State entity name

Cancel

Add

Actions					
Action source ↑	Name	Icon	Type		
Tooltip tag action	Tool_Action	...	Update current dashboard state		

19. Edit Mode Google MAP → Action Tab → Set Action >> **Sensor\_Details** > Add Open Dashboard

### Add action


Action source \*

On marker click

Name \*

Sensor\_Details

Icon

 play\_arrow

Type \*

Custom action

function (\$event, widgetContext, entityId, entityName, additionalParams, entityLabel) { Tidy }

```

1
2 var $injector = widgetContext.$scope.$injector;
3 let deviceService = $injector.get(widgetContext
  .servicesMap.get('deviceService'));

```

Cancel Add

```

var $injector = widgetContext.$scope.$injector;
let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));

deviceService.getDevice(entityId.id).subscribe(function(device) {
  if (device.name == 'Pk_Station_AA') {
    openDashboardState('SUT_Data',device)
  }
  else if(device.name == 'Pk_Station_BB') {
    openDashboardState('STP_Data',device)
  }
});

function openDashboardState(stateId,device) {
  var params = {
    entityId: entityId,
    entityName: device.name
  };
  widgetContext.stateController.openState(stateId, params, false);
}





```

```

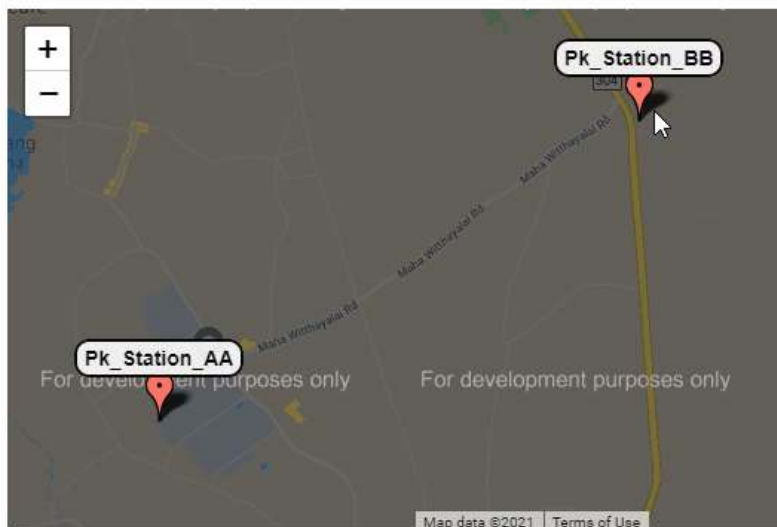
1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));
3
4 deviceService.getDevice(entityId.id).subscribe(function(device) {
5     if (device.name == 'Pk_Station_AA') {
6         openDashboardState('SUT_Data',device)
7     }
8     else if(device.name == 'Pk_Station_BB') {
9         openDashboardState('STP_Data',device)
10    }
11 });
12
13 function openDashboardState(stateId,device) {
14     var params = {
15         entityId: entityId,
16         entityName: device.name
17     };
18     widgetContext.stateController.openState(stateId, params, false);
19 }

```

### Actions

Action source ↑	Name	Icon	Type	
On marker click	Sensor_Details	▶	Custom action	 
Tooltip tag action	Tool_Action	...	Update current dashboard state	 

### New Google Maps



## 20. Create Public Data and Share Dashboard → Share 2 Device + Dashboard

Devices

Device profile

All

×

+

↺

↻

<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	Customer	Public	Is gateway	
<input type="checkbox"/>	2021-06-07 12:03:24	Pk_Station_BB	default			<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	2021-06-07 12:03:00	Pk_Station_AA	default			<input type="checkbox"/>	<input type="checkbox"/>	

Devices

Device profile

All

×

+

↺

↻

🔍

<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	Customer	Public	Is gateway	
<input type="checkbox"/>	2021-06-07 12:03:24	Pk_Station_BB	default		Public	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	2021-06-07 12:03:00	Pk_Station_AA	default		Public	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Dashboards

+

↺

↻

🔍

<input type="checkbox"/>	Created time ↓	Title	Assigned to customers	Public	
<input type="checkbox"/>	2021-06-07 23:58:42	Pk007_TB-Part2		<input type="checkbox"/>	<div>Make dashboard public</div>

Dashboard is now public

×

Your dashboard **Pk007\_TB-Part2** is now public and accessible via next public [link](#):

<https://demo.thingsboard.io/dashboard/9d1b1a30-c7b1-11eb-b36b-fb73ca05e16f?publicId=40fa9860-ba10-11eb-a311-cf80e>

Note: Do not forget to make related devices public in order to access their data.

OK

ThingsBoard Demo | Devices | ThingsBoard Demo | Dashboard

demo.thingsboard.io/dashboard/9d1b1a30-c7b1-11eb-b36b-fb73ca05e16f?publicId=40fa9860-ba10-11eb-a311-cf80d7127bfd

Kogoro FC 1412 EngSUT-WFH2021 Reg2\_Wichai Mail\_SUT Facebook OKNation Manager Online SIAMSPORT Kapook Hindawi Reading list

**Pk007\_TB-Part2** Pk007\_TB-Part2 ▾ Entities Realtime - last minute

New Google Maps

For development purposes only For development purposes only For development purposes only

Pk\_Station\_BB

Pk\_Station\_AA

Map data ©2021 Terms of Use Report a map error

Device admin table

Entity name ↑	Entity type	humidity	temperature	station_name
Pk_Station_AA	Device	86.74	40.4	SUT F11
Pk_Station_BB	Device	74.79	31.58	SurathamPitak School

Items per page: 10 1 - 2 of 2

## 21. Display on Mobile device

09:59 ThingsBoard Demo | Dashboard demo.thingsboard.io

**Pk007\_TB-Part2**

New Google Maps

For development purposes only For development purposes only For development purposes only

Pk\_Station\_BB

Pk\_Station\_AA

Map data ©2021 Terms of Use

Device admin table

Entity name ↑	Entity type	humidity	temperature
Pk_Station_AA	Device	81.17	4
Pk_Station_BB	Device	73.94	3

1 - 2 of 2

Items per page: 10

Powered by Thingsboard v.3.3.0

10:00 ThingsBoard Demo | Dashboard demo.thingsboard.io

**Pk007\_TB-Part2**

For development purposes only For development purposes only For development purposes only

Pk\_Station\_BB

Pk\_Station\_AA

Map data ©2021 Terms of Use

Device admin table

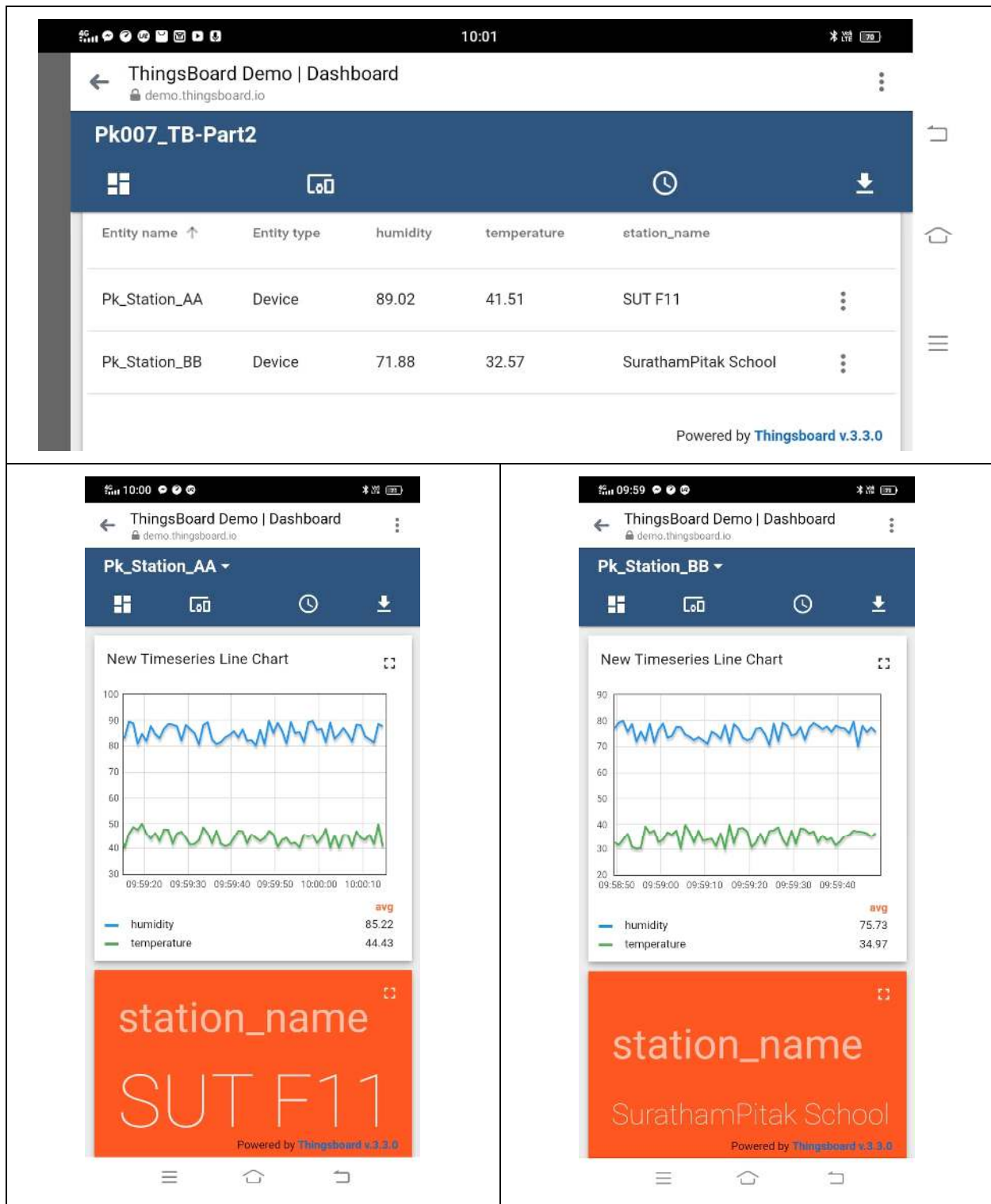
Entity name ↑	Entity type	humidity	temperature
Pk_Station_AA	Device	81.17	4
Pk_Station_BB	Device	73.94	3

1 - 2 of 2

Items per page: 10

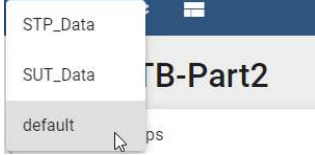
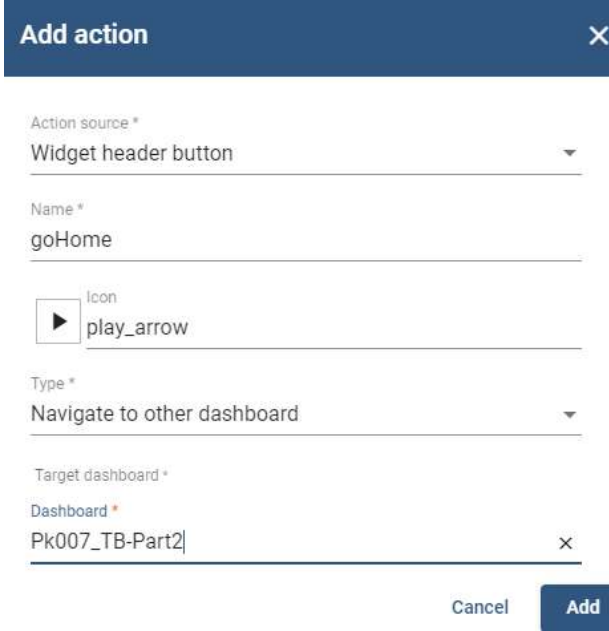
Powered by Thingsboard v.3.3.0





## 4. Question

4.1. ปรับแก้ไขให้มีปุ่ม Home ที่ SUT\_Data และ STP\_Data dashboard เพื่อกลับไปที่หน้า default dash board

	Edit Mode → dashboard
	Add Widget < MAP >
	Set Action

การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัจฉริยะ ThingsBoard IoTs Platform for smart system
ชื่อ-สกุล :
5/5 -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

### Quiz\_301 – MAP Widgets and Multilayer Dashboard – 2 Point Data

- แสดงรูป โปรแกรม ของผลการทำงานตามหัวข้อ Lab304 – Using Rule Chains, MAP Widget and Multilayer dashboard

รูปการทดสอบ 1: Dashboard
รูปการทดสอบ 2:
รูปการทดสอบ 3
รูปการทดสอบ 4
รูปการทดสอบ 5

### Quiz\_302 – MAP Widgets and Multilayer Dashboard – 4 Point Data

- จากหัวข้อ Quiz\_301 ลองปรับเพิ่มจุดข้อมูลจากสองจุด(STP,SUT) เป็นสี่จุดข้อมูล ตามแต่ผู้เรียนกำหนด

รูปการทดสอบ 1: Dashboard
รูปการทดสอบ 2:
รูปการทดสอบ 3
รูปการทดสอบ 4
รูปการทดสอบ 5