

## Instalação do WSL:

- Abrir o powershell como administrador
- Comando: wsl --install

```
PS C:\WINDOWS\system32> wsl --install
Installing: Ubuntu
[= 2.0%
```

- Definir o nome e a password

---

```
tiago@LAPTOP-FQE23UDF: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: tiago
New password:
Retype new password:
passwd: password updated successfully
The operation completed successfully.
Installation successful!
```

- Instalar o python3 e o pip:

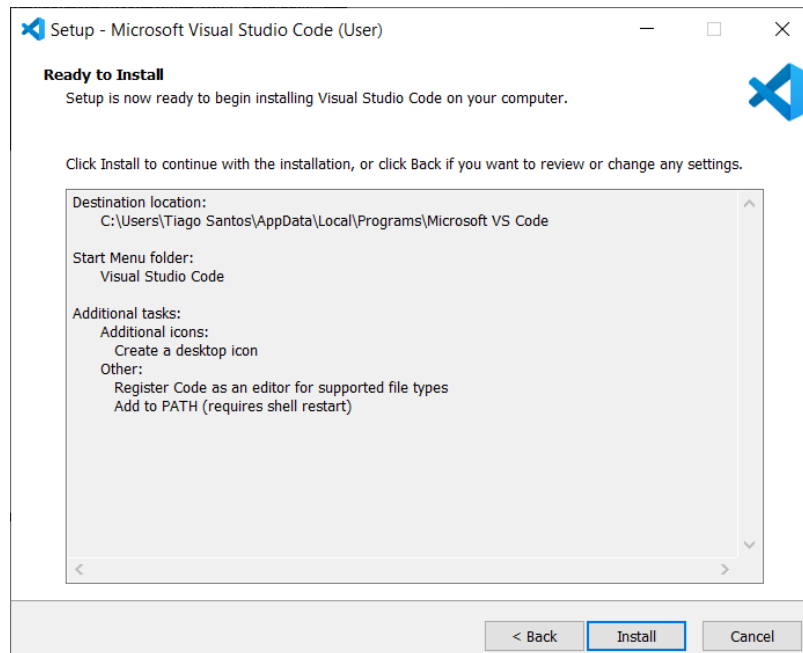
```
sudo apt update
```

```
sudo apt install python3 -y
```

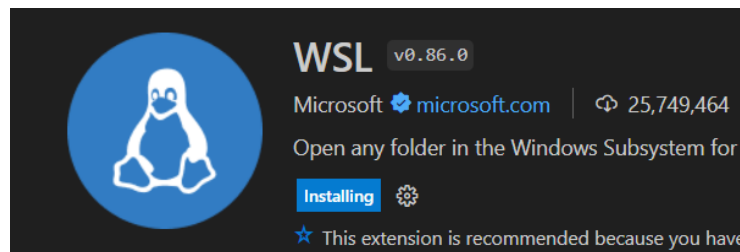
```
sudo apt install python3-pip -y
```

## Instalação Visual Studio Code

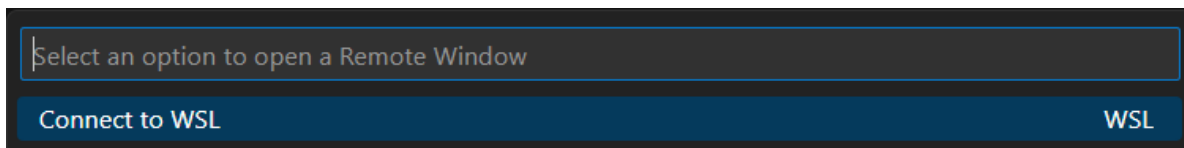
- Extrair o .exe da internet conforme o respetivo sistema operativo que utiliza



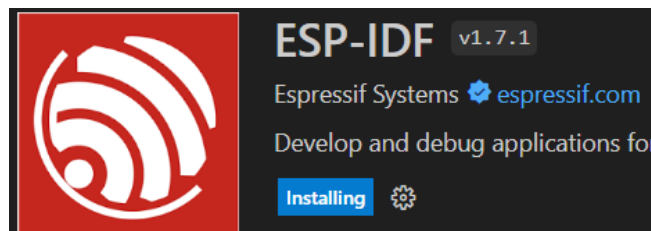
- Instalar o wsl nas extensões do vscode



- Selecionar a opção “Connect to wsl”



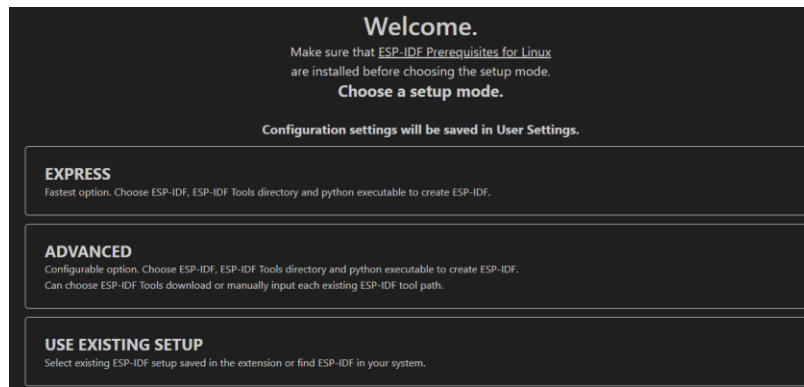
- Instalar a extensão esp-idf



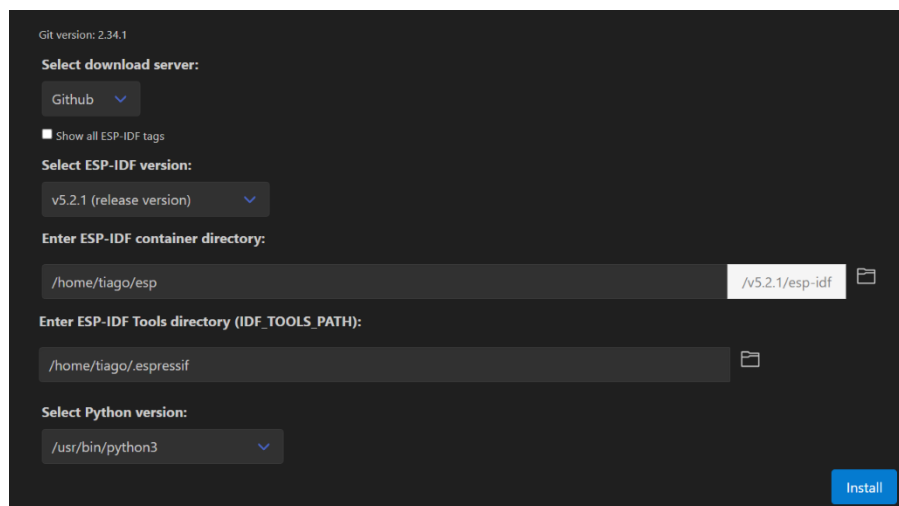
- Instalar a extensão esp-matter após ter o esp-idf instalado

Instalação esp-idf:

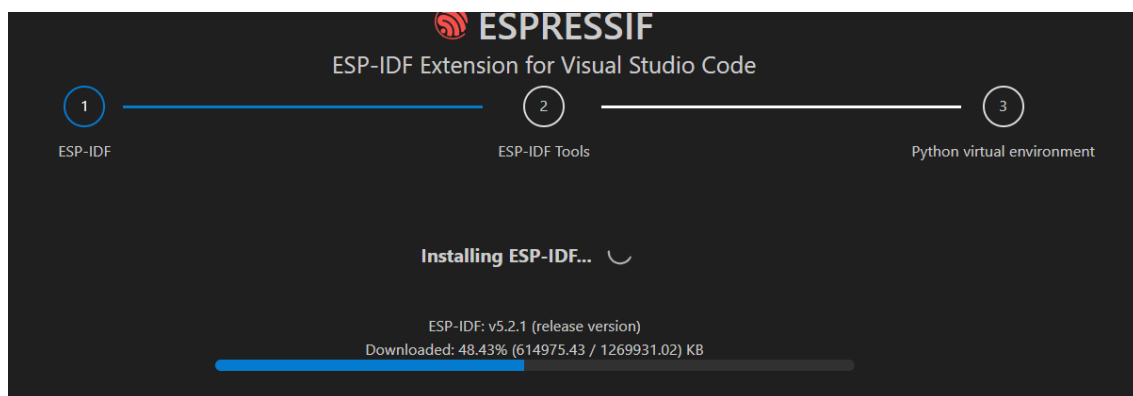
- Selecionar o esp-idf na barra lateral do visual studio code
- Escolher a opção express



- Definir o path onde irá ficar instalado e a versão que irá ser instalada



- Clicar em instalar



## Compartilhar USB ports para o WSL:

- Instalar o usbipd com o comando “winget install usbipd”

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> winget install usbipd
The 'msstore' source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires the current machine's 2-letter geographic region to be sent to the backend serv
rily (ex. "US").

Do you agree to all the source agreements terms?
[Y] Yes [N] No: Y
Found usbipd-win [dorssel.usbipd-win] Version 4.1.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/dorssel/usbipd-win/releases/download/v4.1.0/usbipd-win_4.1.0.msi
0.00 B / 4.57 MB
```

- Ver a lista de portas usb através do comando “usbipd list”

```
PS C:\Windows\system32> usbipd list
Connected:
BUSID  VID:PID  DEVICE                                STATE
-----
2-1    17ef:60be USB Input Device                      Not shared
2-4    1a86:55d3 USB-Enhanced-SERIAL CH343 (COM4)      Not shared
2-6    13d3:56ff Integrated Camera                    Not shared
2-9    048d:c100 USB Input Device                      Not shared
2-14   8087:0026 Intel(R) Wireless Bluetooth(R)       Not shared
```

- Dar bind ao BUSID que está associado à porta COM através do comando “usb bind --busid <BUSID>”

```
PS C:\Windows\system32> usbipd bind --busid 2-4
PS C:\Windows\system32> usbipd list
Connected:
BUSID  VID:PID  DEVICE                                STATE
-----
2-1    17ef:60be USB Input Device                      Not shared
2-4    1a86:55d3 USB-Enhanced-SERIAL CH343 (COM4)      Shared
2-6    13d3:56ff Integrated Camera                    Not shared
2-9    048d:c100 USB Input Device                      Not shared
2-14   8087:0026 Intel(R) Wireless Bluetooth(R)       Not shared
```

- Dar attach ao mesmo BUSID que se deu bind no comando anterior através do comando “usbipd attach --wsl --busid <BUSID>”
- Ver lista novamente e verificar se a porta está attached

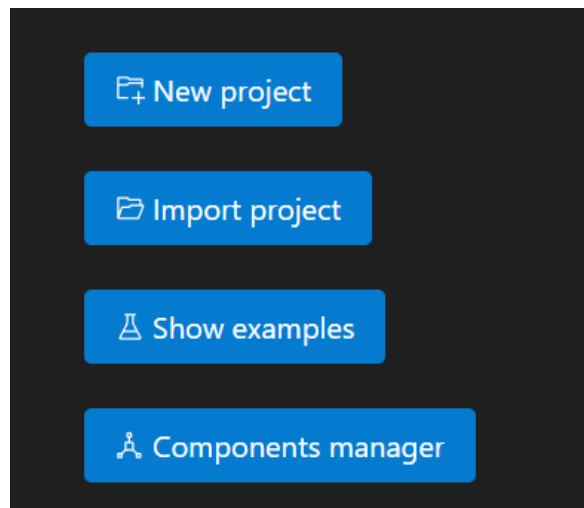
```
PS C:\Windows\system32> usbipd list
Connected:
BUSID  VID:PID  DEVICE                                STATE
-----
2-1    17ef:60be USB Input Device                      Not shared
2-4    1a86:55d3 USB-Enhanced-SERIAL CH343 (COM4)      Attached
2-6    13d3:56ff Integrated Camera                    Not shared
2-9    048d:c100 USB Input Device                      Not shared
2-14   8087:0026 Intel(R) Wireless Bluetooth(R)       Not shared
```

- Verificar que a porta está disponível no ambiente linux de modo a que seja utilizada no visual studio code através do comando “lsusb”

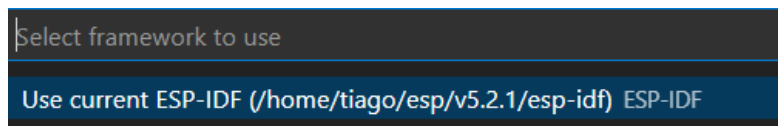
```
pedro@DESKTOP-A7H409F:~/all_device_types_app/all_device_types_app/.vscode/station$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 1a86:55d3 QinHeng Electronics USB Single Serial
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

## Testar se o esp-idf foi bem instalado

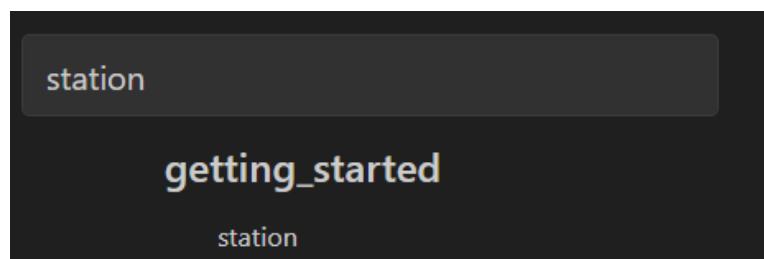
- Clicar em Show examples



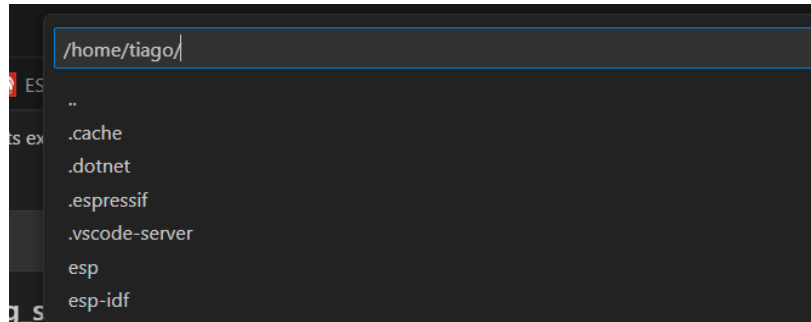
- Escolher o idf instalado



- Selecionar o projeto Station



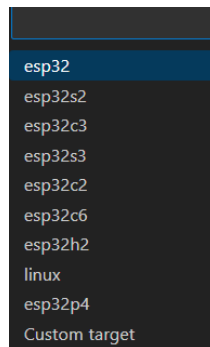
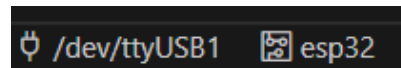
- Selecionar o caminho onde irá ser instalado o station



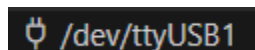
- No ficheiro station\_examples\_main.c definir o SSID e Password do WI-FI

```
#define EXAMPLE_ESP_WIFI_SSID      "SSID"
#define EXAMPLE_ESP_WIFI_PASS      "PASS"
#define EXAMPLE_ESP_MAXIMUM_RETRY  CONFIG_ESP_MAXIMUM_RETRY
```

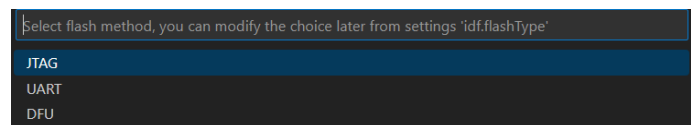
- Selecionar a placa esp32 que está a ser utilizada clicando no esp32 na barra de baixo do visual studio code



- Selecionar a porta onde está inserido o esp32 clicando no /dev/ttyUSB1

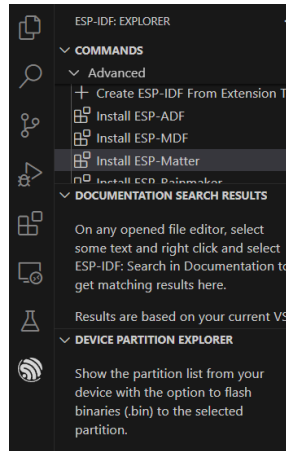


- Clicar na chama na barra de baixo do visual studio code
- Após o build ser feito selecionar "UART"

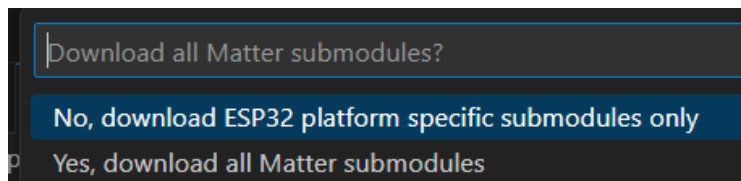


## Instalação do esp-matter

- Abrir a extensão esp-idf, selecionar Advanced e clicar em Install ESP-Matter

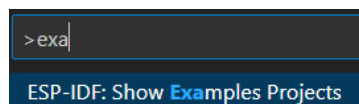


- Selecionar se pretende o repositório todos ou apenas sub-módulos, qual o caminho onde irá ser instalado, e irá começar a instalar automaticamente

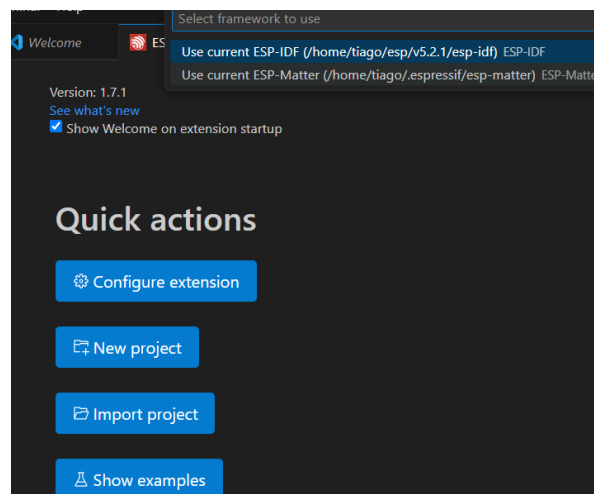


## Testar se o esp-matter foi bem instalado

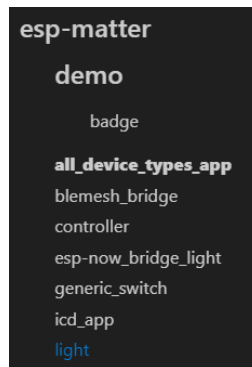
- Na barra de pesquisa do visual studio code escrever ">exa" e escolher "ESP-IDF: Show Example Projects"



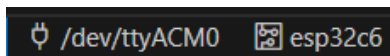
- Clicar em "Show Examples" e escolher os exemplos do esp-matter



- Escolher o projeto “light”



- Seleciona a placa do esp-32 e a porta utilizada



- Clicar na chama para dar build e flash ao projeto
- Após o flash do projeto tem que se conectar á rede wifi inserindo o seguinte comando “matter esp wifi connect <SSID> <Password>”

```
W (322316) wifi:Haven't to connect to a suitable AP now!
E (322316) chip[DL]: Failed to get configured network when updating network status: Error ESP32:0x0500300F

> matter esp wifi connect ME0-4592F0 640a87e3ab

W (325066) wifi:(itwt)itwt_stop_process!
```

## Instalação do Kernel no WSL para aceder ao Bluetooth

- Executar o comando “`git clone --depth 1 --branch linux-msft-wsl-6.1.21.2 https://github.com/microsoft/WSL2-Linux-Kernel.git`” para clonar o kernel do GitHub

```
tiago@LAPTOP-FQE23UDF:~$ git clone --depth 1 --branch linux-msft-wsl-6.1.21.2 https://github.com/microsoft/WSL2-Linux-Kernel.git
Cloning into 'WSL2-Linux-Kernel'...
remote: Enumerating objects: 83412, done.
remote: Counting objects: 100% (83412/83412), done.
remote: Compressing objects: 79% (62100/78607)
```

- Após isto executar os seguintes comandos:

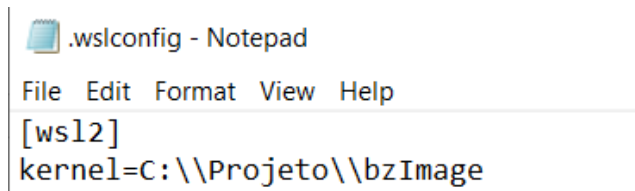
```
cd WSL2-Linux-Kernel
git checkout linux-msft-wsl-6.1.21.2
cp /proc/config.gz config.gz
gunzip config.gz
mv config .config
sudo make menuconfig
```

- Para ativar o bluetooth tem que seleccionar duas opções e após seleccioná-las guardar a configuração:
  1. Enable Networking support ->Bluetooth subsystem support.



## 2. Enable Networking Support ->Bluetooth Subsystem Support ->Bluetooth device drivers -> HCI USB driver.

- Após guardar a configuração tem que correr a configuração através do comando “`sudo make -j$(getconf _NPROCESSORS_ONLN) && sudo make modules_install -j$(getconf _NPROCESSORS_ONLN) && sudo make install -j$(getconf _NPROCESSORS_ONLN)`”
- Copiar a bzImage criada se o comando acima for executado com sucesso e inserir a mesma no disco “C:”
- Criar o ficheiro .wslconfig para indicar o caminho que o WSL tem que utilizar através do comando “`notepad.exe %UserProfile%\.wslconfig`”
- Indicar o caminho conforme onde foi colocado o bzImage no disco através da configuração demonstrada na imagem



- Verificar através do powershell se o adaptador bluetooth já está “attached” ao WSL através do comando “`usbipd list`”
- Se não estiver tem que se dar “attach” através do comando “`usbipd attach --wsl --busid=<BUSID>`”
- Instalar o bluez no WSL através do comando “`sudo apt install bluez`”
- Iniciar o bluetooth através do comando “`sudo systemctl start bluetooth`” e verificar se o mesmo está a funcionar através do comando “`sudo systemctl status bluetooth`”

```
tiago@LAPTOP-FQE23UDF:~$ sudo systemctl status bluetooth
● bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-04-02 16:41:53 WEST; 1h 24min ago
```

## Instalação do Home Assistant no VirtualBox

- Instalar da internet o .vdi do home assistant através do link <https://www.home-assistant.io/installation/alternative/>
- Criar uma máquina virtual no VirtualBox:
  - Definir o nome da máquina
  - O tipo de sistema operativo é “Linux” e a versão “Other Linux (64-bit)”

**Create Virtual Machine**

### Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name:  ✓

Folder:

ISO Image:

Edition:

Type:  64

Version:

☐ Skip Unattended Installation

*No ISO image is selected, the guest OS will need to be installed manually.*

Help Expert Mode Back **Next** Cancel

- Definir 2048MB de memória e 2 processadores
- Selecionar a opção “Enable EFI”

### Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory:  4 MB 16384 MB

Processors:  1 CPU 12 CPUs

☒ Enable EFI (special OSes only)

- Adicionar o disco instalado anteriormente à máquina virtual

### Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☐ Create a Virtual Hard Disk Now

Disk Size:  4.00 MB 2.00 TB

☐ Pre-allocate Full Size

☒ Use an Existing Virtual Hard Disk File

☐ Do Not Add a Virtual Hard Disk

- Na parte da rede selecionar o mode Bridge

### Network

Adapter 1 Adapter 2 Adapter 3 Adapter 4

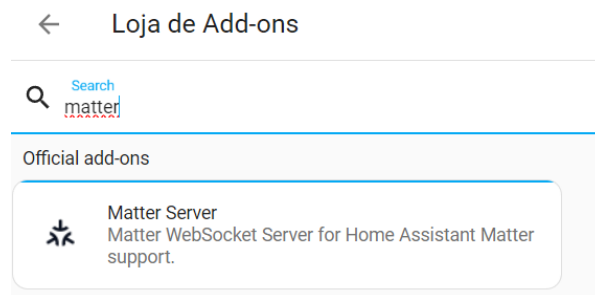
☒ Enable Network Adapter

Attached to:

Name:

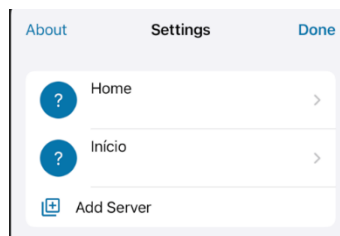
- Inicializar o Home Assistant

- Inserir o IP atribuído no browser seguido pelo porto 8123 “IP:8123”
- Configurar as credenciais do utilizador
- Ir a configurações, “Add-ons”, “Loja de ADD-ONS” e adicionar a biblioteca Matter Server”



## Instalação e configuração do Home Assistant em iOS

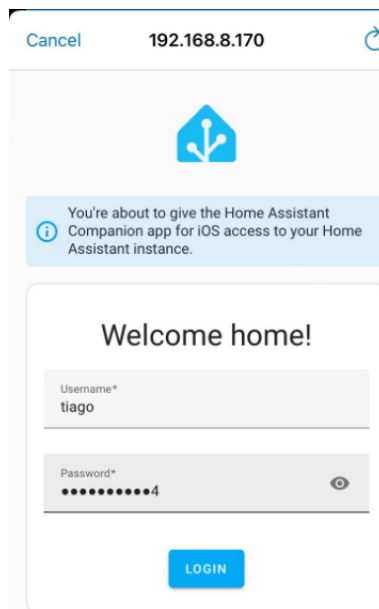
- No “AppStore” procurar por “HomeAssistant” e instalar a aplicação
- Selecionar “Add Server”



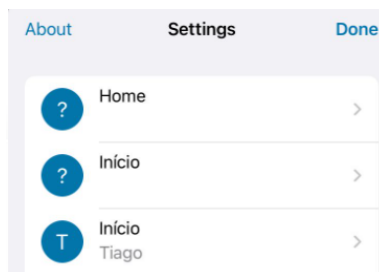
- Selecionar o servidor que aparece por omissão que tem o mesmo IP da máquina virtual



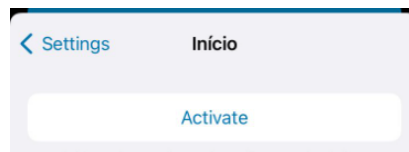
- Inserir as credenciais definidas na configuração do Home Assistant da máquina virtual e clicar em “Login”



- Após isto já deverá de aparecer o servidor adicionado, e tem que seleccioná-lo

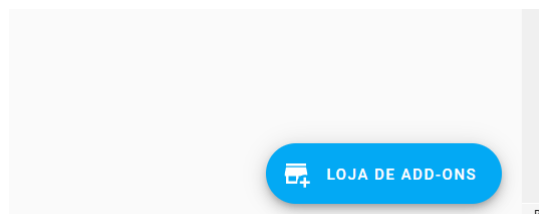


- Clicar em “Activate” para tivar o servidor

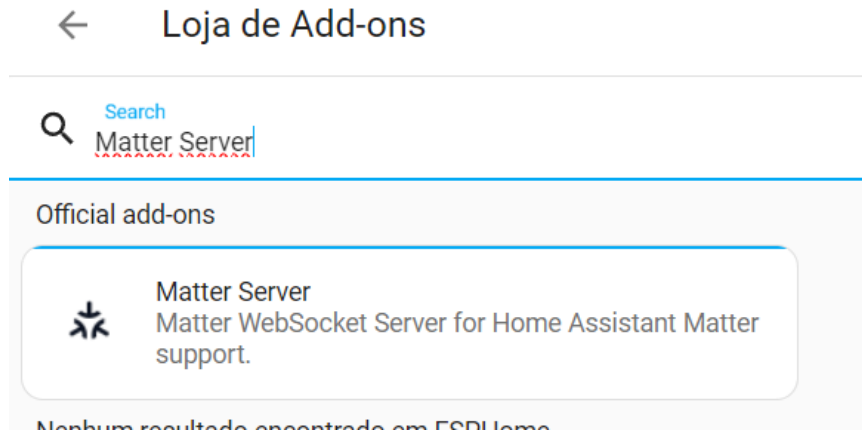


## Instalar a extensão Matter Server

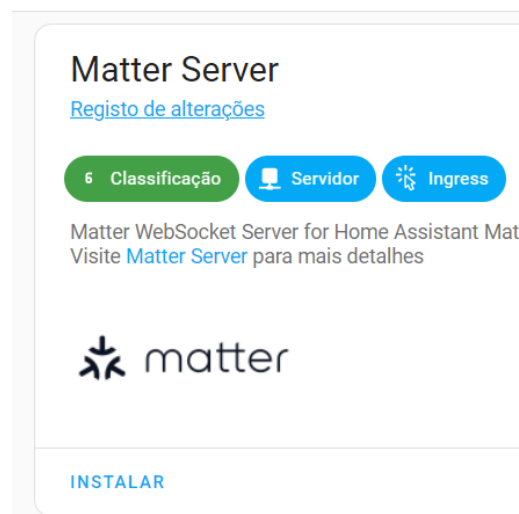
- Aceder às configurações e seleccionar a opção “Add-ons”
- Clicar no botão “LOJA DE ADD-ONS”



- Procurar por “Matter Server” na barra de pesquisa



- Clicar na extensão “Matter Server” e clicar em “Instalar”



## Adicionar o ESP32-C6 ao Home Assistant

- Na consola do esp-matter tem que criar um qrcode para o esp32-c6 através do comando “matter onboardingcodes ble” e aceder ao link criado com o qrcode

```
> matter onboardingcodes ble

QRCode:          MT:Y.K9042C00KA0648G00
QRCodeUrl:       https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3AY.K9042C00KA0648G00
ManualPairingCode: 34970112332
```

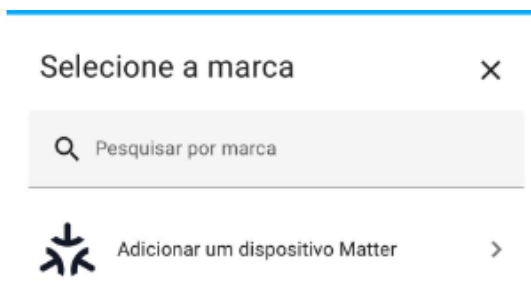
- Aceder às configurações e seleccionar a opção “Dispositivos & serviços”



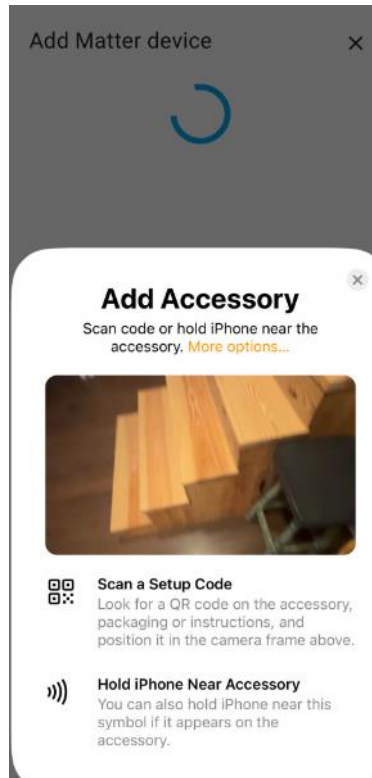
- Clicar em “Adicionar Integração”



- Selecionar “Adicionar um dispositivo Matter”

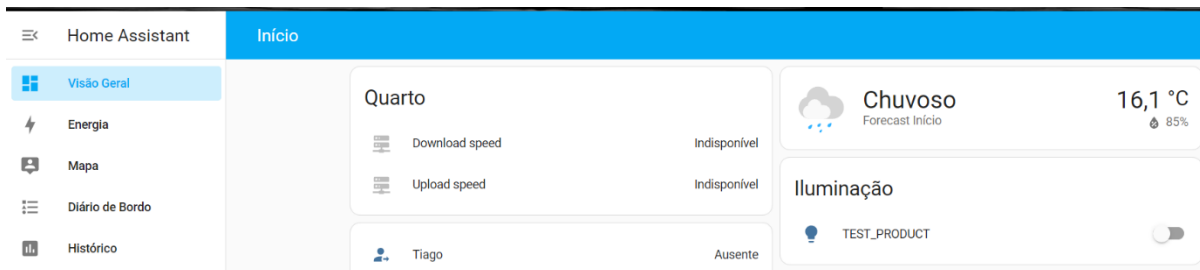


- Fazer scan com a camara do telemóvel ao qrcode criado anteriormente e automaticamente o dispositivo fica adicionado ao servidor

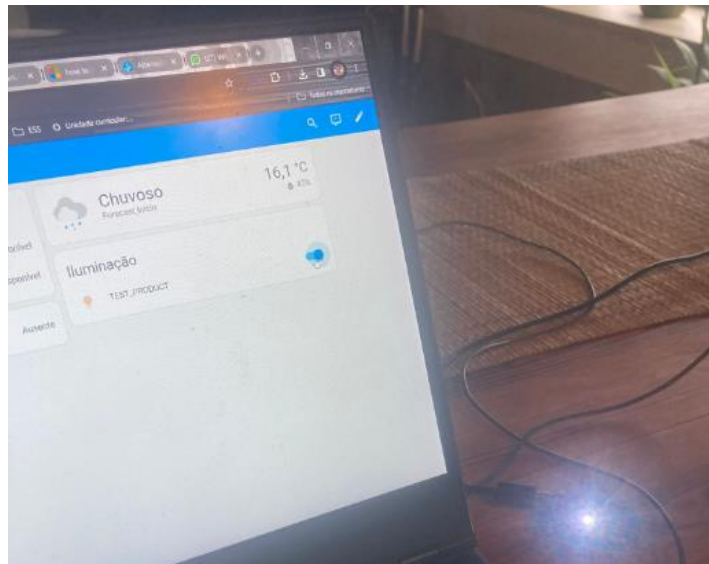
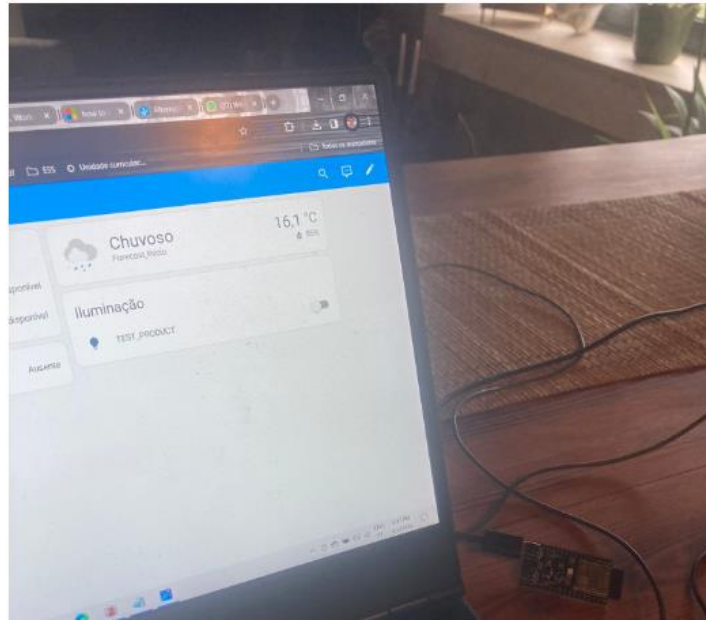


## Testar a integração do esp32-c6

- Aceder à “Visão Geral” do Home Assistant na máquina virtual e verificar se a integração foi bem-sucedida pelo telemóvel



- Dar “toggle” ao botão no dashboard do dispositivo de iluminação e verificar se a luz do ESP32-C6 está a ligar e a desligar conforme o “On” e “Off” da máquina virtual





## Instalação do firmware na sonoff USB dongle

- Aceder ao [website](#) , e na secção “Flashing the Sonoff ZBDongle-E” clicar em “Web Flasher”

- Open **darkxst**'s [Web Flasher](#)

- Ir à secção ZBDongle-E e clicar em “CONNECT”

### ZBDongle-E

Sonoff Zigbee 3.0 USB Dongle Plus V2 - No hardware flow control



CONNECT

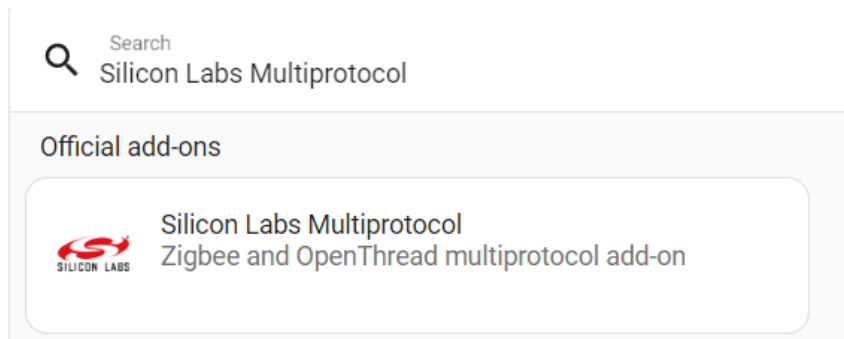
- Selecionar a sonoff e clicar em “Connect”
- Clicar em “CHANGE FIRMWARE”
- Selecionar o firmware e clicar em “INSTALL”
- Se for bem sucedido, irá aparecer uma mensagem a dizer “Installation success”

## Instalação da extensão Silicon Labs Multiprotocol

- Aceder às configurações e seleccionar a opção “Add-ons”
- Clicar no botão “LOJA DE ADD-ONS”

 LOJA DE ADD-ONS

- Procurar por “Silicon Labs Multiprotocol” na barra de pesquisa

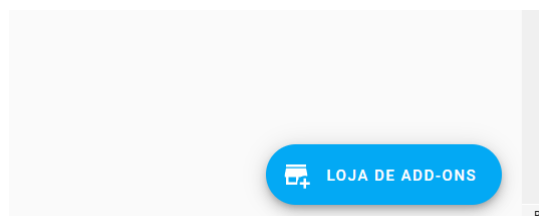


- Clicar na extensão “Silicon Labs Multiprotocol” e clicar em “Instalar”

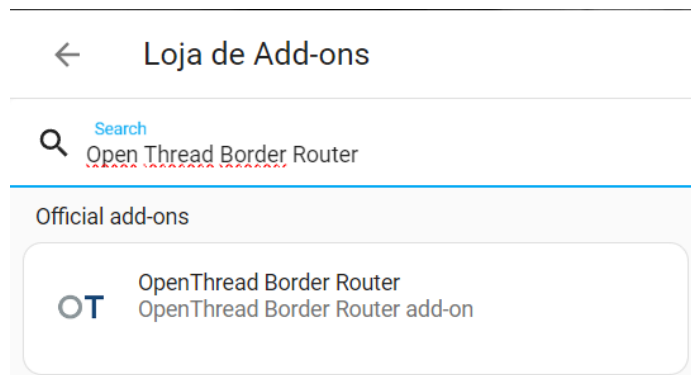


## Instalação da extensão Open Thread Border Router

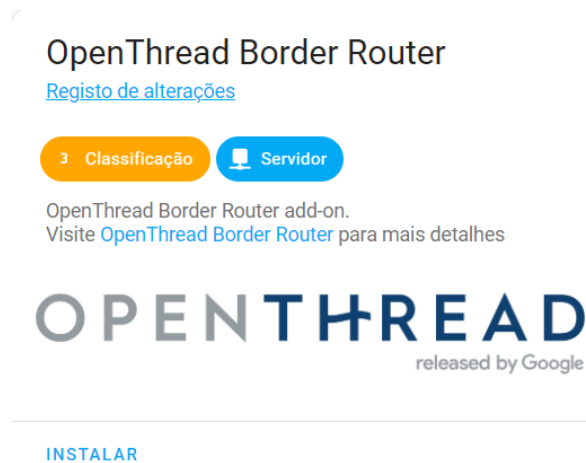
- Aceder às configurações e seleccionar a opção “Add-ons”
- Clicar no botão “LOJA DE ADD-ONS”



- Procurar por “Open Thread Border Router” na barra de pesquisa

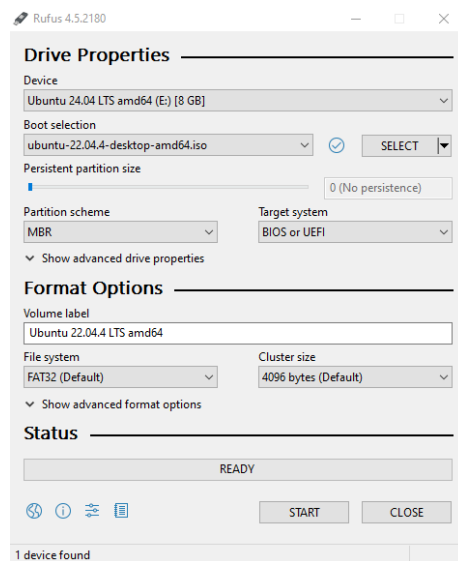


- Clicar na extensão “Open Thread Border Router” e clicar em “Instalar”

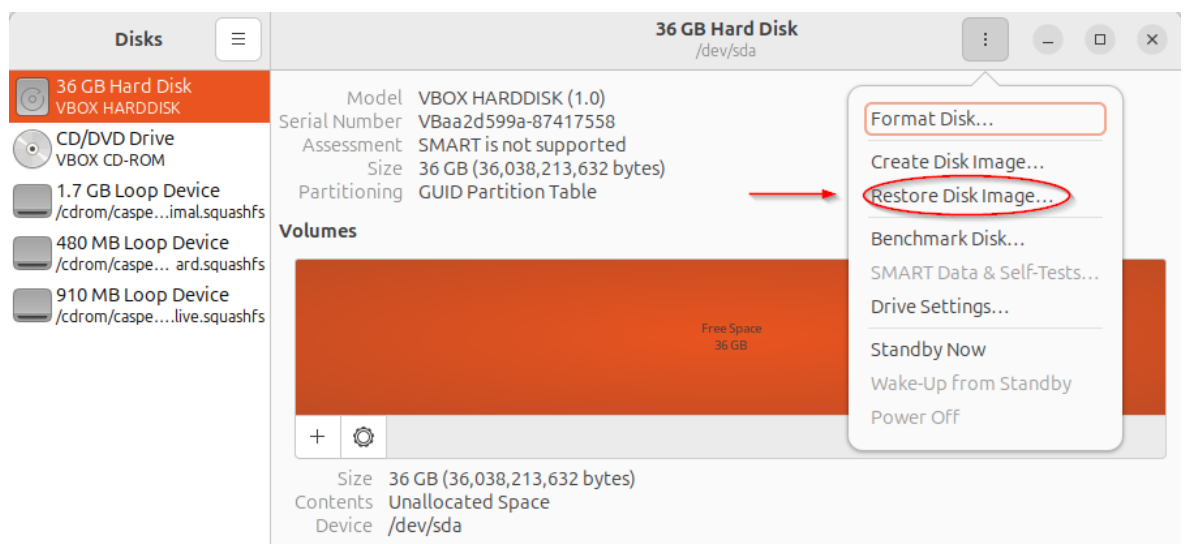


## Instalar o Home Assistant no mini PC

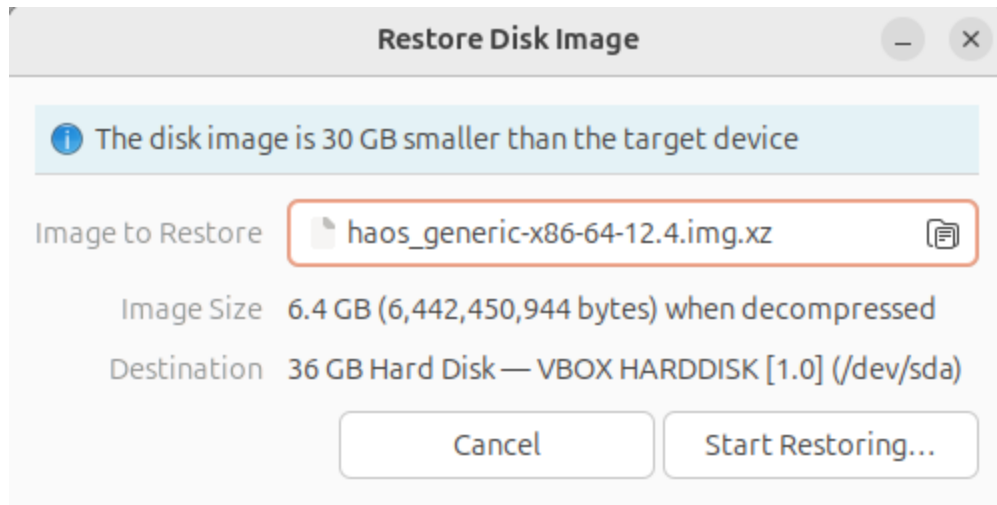
- Começar por criar uma *pen USB bootable* através do programa *rufus*.
- Fazer download de uma imagem do Sistema Operativo *Ubuntu*
- Selecionar a *pen* no campo *Device* e a imagem do *Ubunto* no campo *Boot Selection*



- Clicar em *START* e esperar que o processo termine.
- Quando o processo terminar, inserir a pen no mini PC e entrar na BIOS, alterando a ordem de boot de forma que a pen USB fique em primeiro na lista
- Quando o mini PC iniciar, clicar em “Try Ubuntu”
- Após o Sistema Operativo iniciar, fazer download da imagem do Home Assistant através do link <https://github.com/home-assistant/operating-system/releases/>
- Iniciar a aplicação “Disks” já instalada por predefinição
- Selecionar o disco em que se pretende instalar o Home Assistant e clicar em “Restore Disk Image”



- Selecionar a imagem previamente transferida do Home Assistant e clicar em “Start Restoring”



- Quando o processo de restore terminar, pode-se desligar o mini PC e retirar a pen USB
- Quando se ligar novamente o mini PC, o Home Assistant já estará instalado e pode-se proceder à instalação e configuração como previamente realizado na máquina virtual

## Configuração do Mikrotik

- Começar por definir pools de endereços *ipv4* e *ipv6*, estas pools serão utilizadas para os servidores DHCP

1 item				
		▲ Name	Addresses	Next Pool
-		+ dhcp	192.168.88.10-192.168.88.254	none

1 item					
		▲ Name	Prefix	Prefix Length	Expire Time
-		ipv6_pool	2001:db8::/64	64	

- Após as pools serem criadas, deve-se criar um servidor DHCP para ipv4 e outro para ipv6, utilizando as pools previamente criadas.

1 item							
		▲ Name	Interface	Relay	Lease Time	Address Pool	Add ARP For Leases
-	D	defconf	bridge		00:10:00	dhcp	no

1 item

		▲ Name	Interface	Address Pool6	Lease Time	
-	D	ipv6_dhcp	bridge	ipv6_pool	3d 00:00:00	

- Adicionar servidores DNS para a comunicação ipv6, utilizando os IPs do router e dos servidores da Google

RouterOS v6.49.10 (long-term)

Apply Static Cache

**Servers**

- 8.8.8.8
- 2001:db8::c6ad:34ff:fe9f:56b
- 2001:4860:4860::8844
- 2001:4860:4860::8888

**Dynamic Servers** 192.168.9.1  
192.168.9.1

**Use DoH Server** ▼

**Verify DoH Certificate** ☐

**Allow Remote Requests** ☒

**Max UDP Packet Size** 4096

**Query Server Timeout** 2.000 s

**Query Total Timeout** 10.000 s

**Max. Concurrent Queries** 100

**Max. Concurrent TCP Sessions** 20

**Cache Size** 2048 KiB

**Cache Max TTL** 7d 00:00:00

**Cache Used** 100 KiB

- Como os pings não eram realizados com sucesso, criámos rotas estáticas para garantir a comunicação na rede em ipv6

2 items

		▲ Dst. Address	Gateway	Distance	
- D	AS	▶ ::/0	bridge reachable	1	
-	DAC	▶ 2001:db8::/64	bridge reachable	0	

- Após seguirmos estes passos, conectámos um PC ao Wi-Fi do router e conseguimos obter com sucesso endereços ipv4 e ipv6

Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix . :  
IPv6 Address. . . . . : 2001:db8::7c46:fc81:d9a8:9b29  
Temporary IPv6 Address. . . . . : 2001:db8::881e:f246:db34:9655  
Link-local IPv6 Address . . . . . : fe80::fa28:6867:f7bb:91aa%23  
IPv4 Address. . . . . : 192.168.88.252  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : fe80::c6ad:34ff:fe9f:56ba%23  
                            192.168.88.1
```