W5100S-EVB-Pico

Getting Started Guide

for AWS IoT Core

Version 1.0.0

Table of Contents

1 Document Information

1.1 Revision History (Version, Date, Description of change)

Version	Date	Description of change
V1.0.0	2021-11-25	Initial Release

2 Overview

The W5100S-EVB-Pico is a microcontroller evaluation board based on the Raspberry Pi RP2040 microcontroller chip and full hardwired TCP/IP controller W5100S chip. The W5100S-EVB-Pico has the same role as the Raspberry Pi Pico platform and includes W5100S, so the Ethernet function is basically included.

- Raspberry Pi Pico Clone
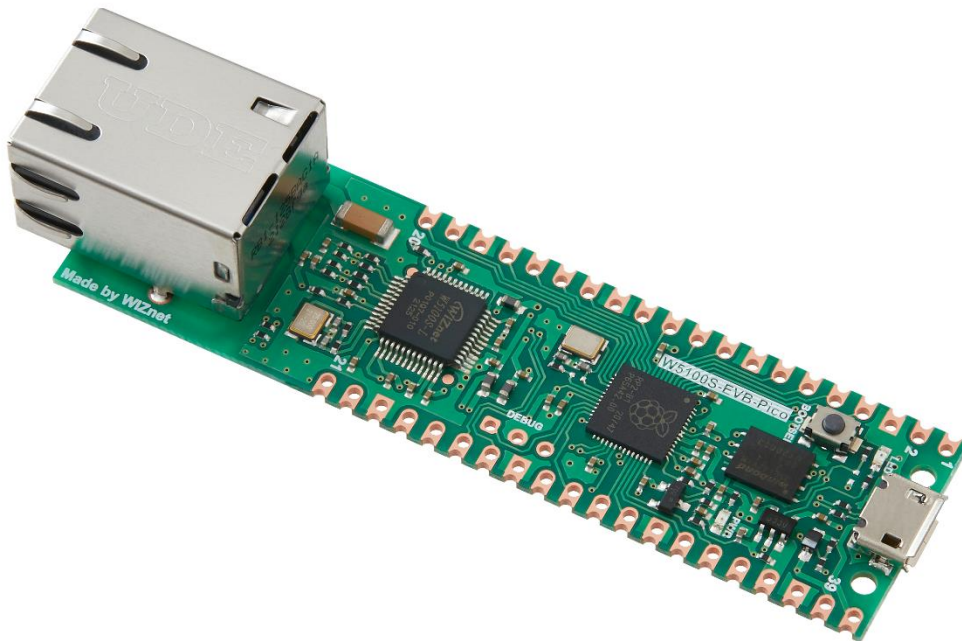- Ethernet (W5100S Hardwired TCP/IP CHIP)

Figure 1. W5100S-EVB-Pico

3 Hardware Description

3.1 DataSheet

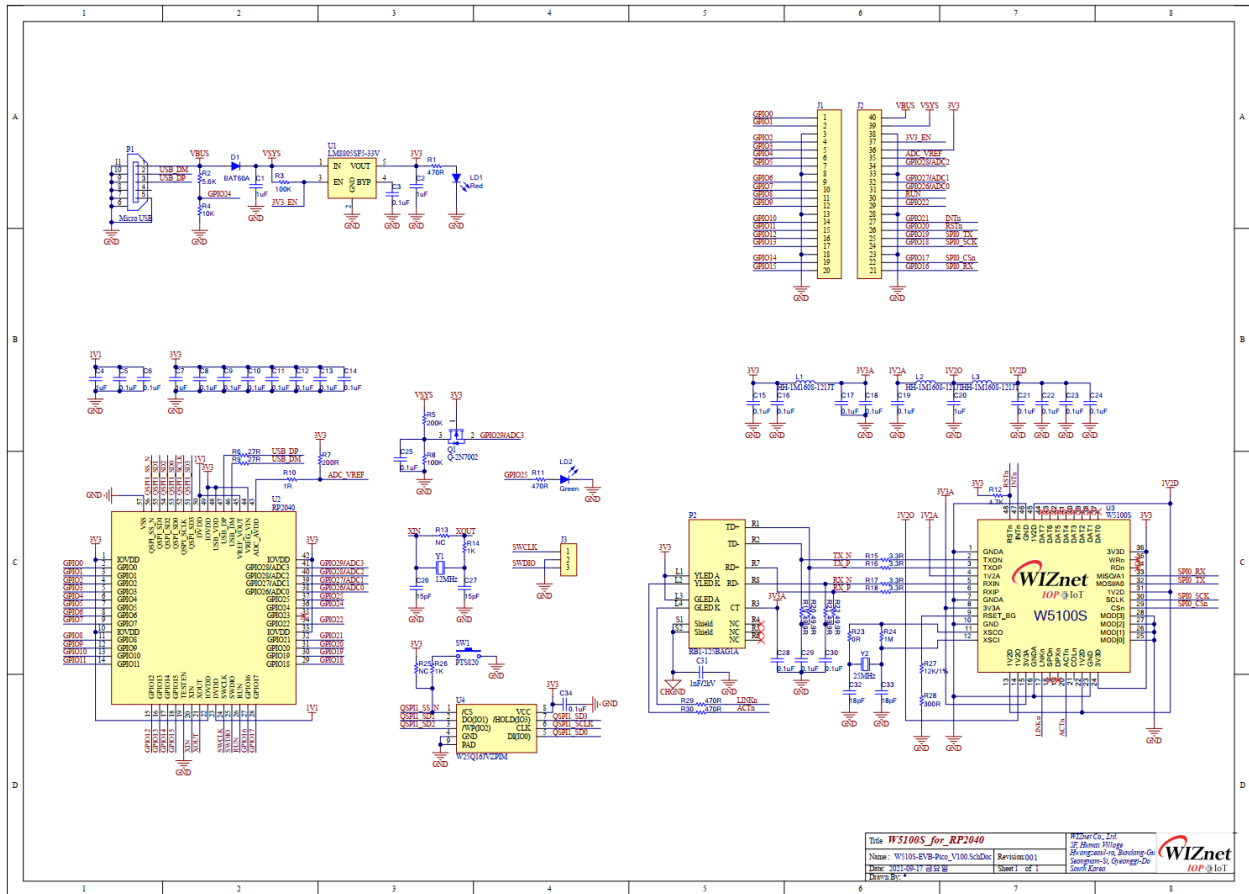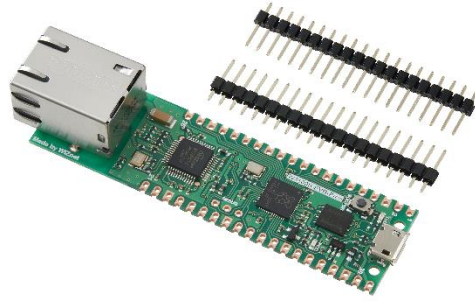https://docs.wiznet.io/assets/images/w5100S-evb_pico_schematic-52444497b86d1ef0550d99aba4c0c164.png

Figure 2. W5100S-EVB-Pico schematic

3.2 Standard Kit Contents

· W5100S-EVB-Pico : 1EA
· 1 x 20 2.54mm pitch pin header : 2EA

Figure 3. W5100S-EVB-Pico package

3.3 User Provided items

· USB type A to USB micro B cable

Figure 4. USB type A to USB micro B cable

· Ethernet cable
· Desktop or laptop

3.4 3rd Party purchasable items

None

3.5 Additional Hardware References

4 Set up your Development Environment

4.1 Tools Installation (IDEs, Toolchains, SDKs)

Windows 10 was used during preparation of this guide document. Linux and MacOS user should use compatible software, hardware-wise there is no difference. Please refer to the guide in section 4.4 to find instructions for installing toolchain on Linux and MacOS.

1) Install the Toolchain

To build you will need to install extra tools below.

- ARM GCC compiler
- CMake
- Build Tools for Visual Studio
- Python 3.9
- Git
- Visual Studio Code

Download the executable installer for each of these from the links above, and then carefully follow the instructions in the following sections to install all six packages on to your Windows computer.
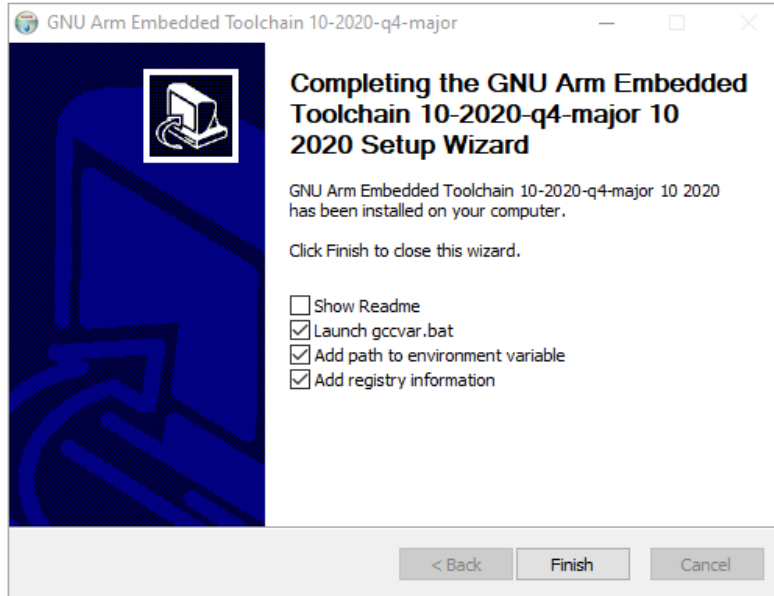
① Install ARM GCC compiler

Figure 5. Install ARM GCC compiler

During installation you should check the box to register the path to the ARM compiler as an environment variable in the Windows shell when prompted to do so.
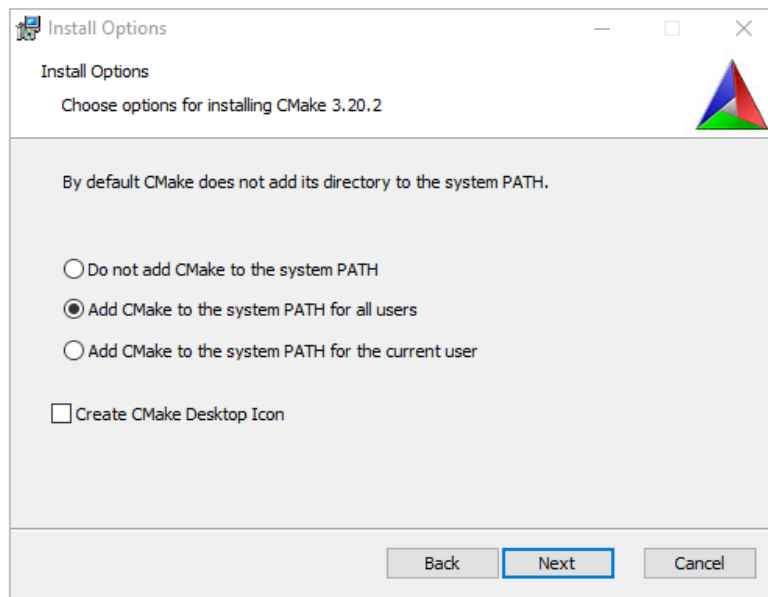
② Install CMake

Figure 6. Install CMake

During the installation add CMake to the system PATH for all users when prompted by the installer.

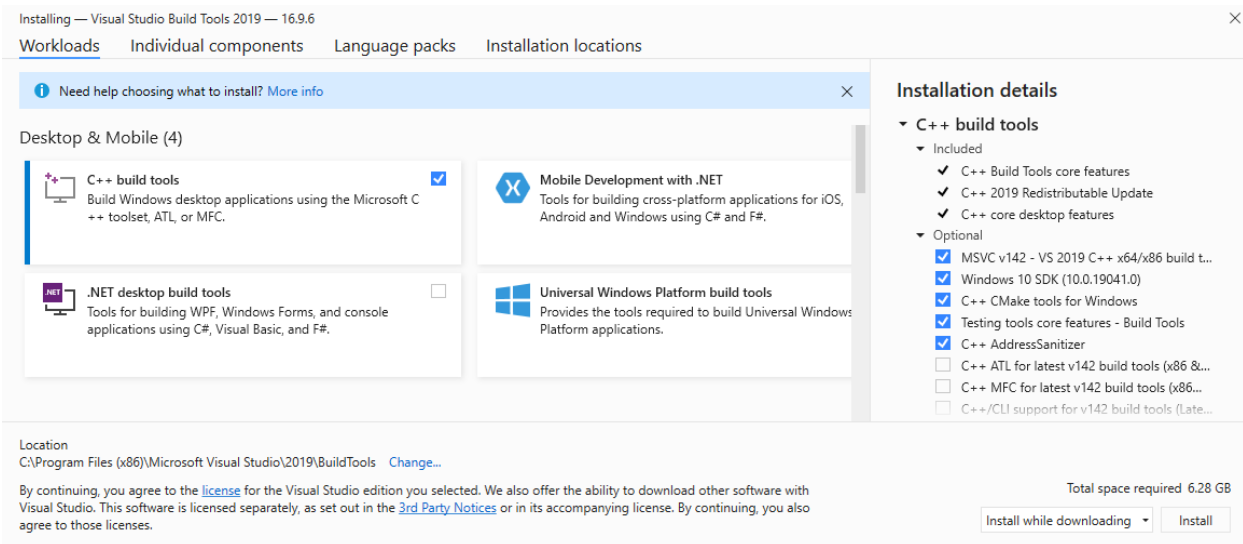③ Install Build Tools for Visual Studio

Figure 7. Install Build Tools for Visual Studio

When prompted by the Build Tools for Visual Studio installer you need to install the C++ build tools only.

④ Install Python 3.9

During the installation, ensure that it's installed 'for all users' and add Python 3.9 to the system PATH when prompted by the installer. You should additionally disable the MAX_PATH length limit when prompted at the end of the Python installation.

Figure 8. Install Python

⑤ Install Git

When installing Git you should ensure that you change the default editor away from vim.
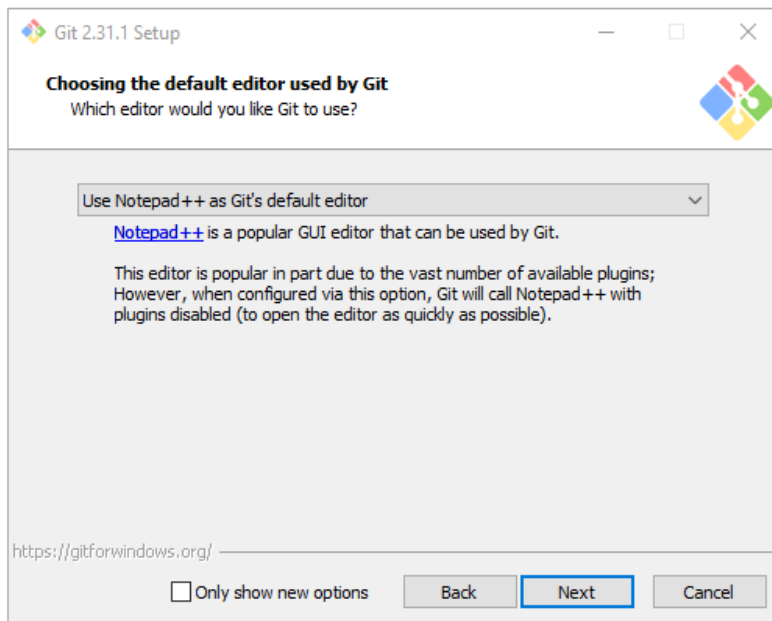

Figure 9. Install Git

⑥ Install Visual Studio Code

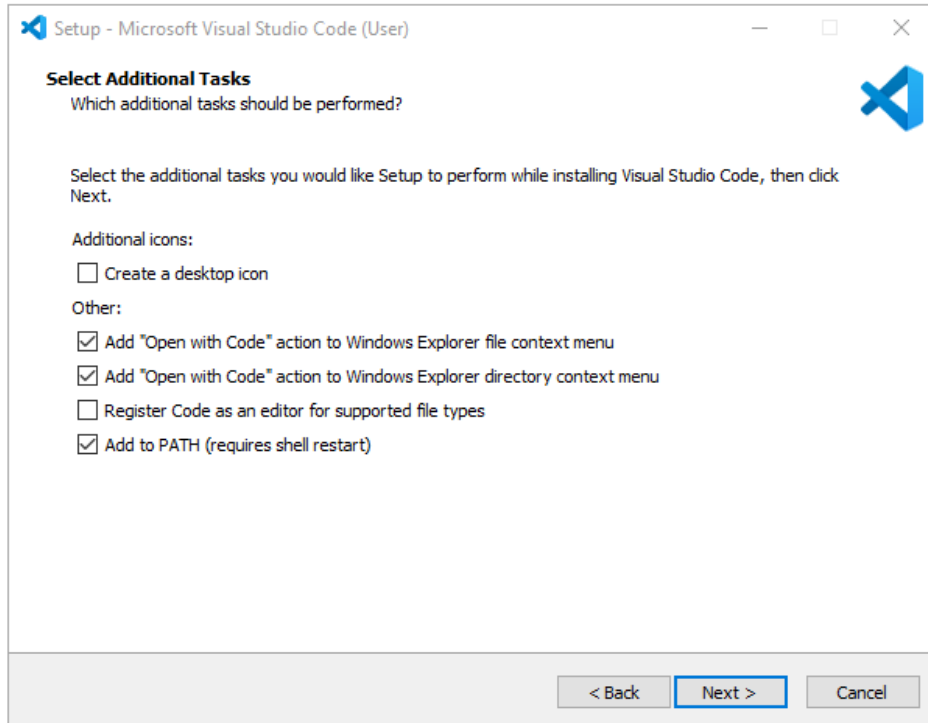During the installation add Visual Studio Code to the system PATH.

Figure 10. Install Visual Studio Code

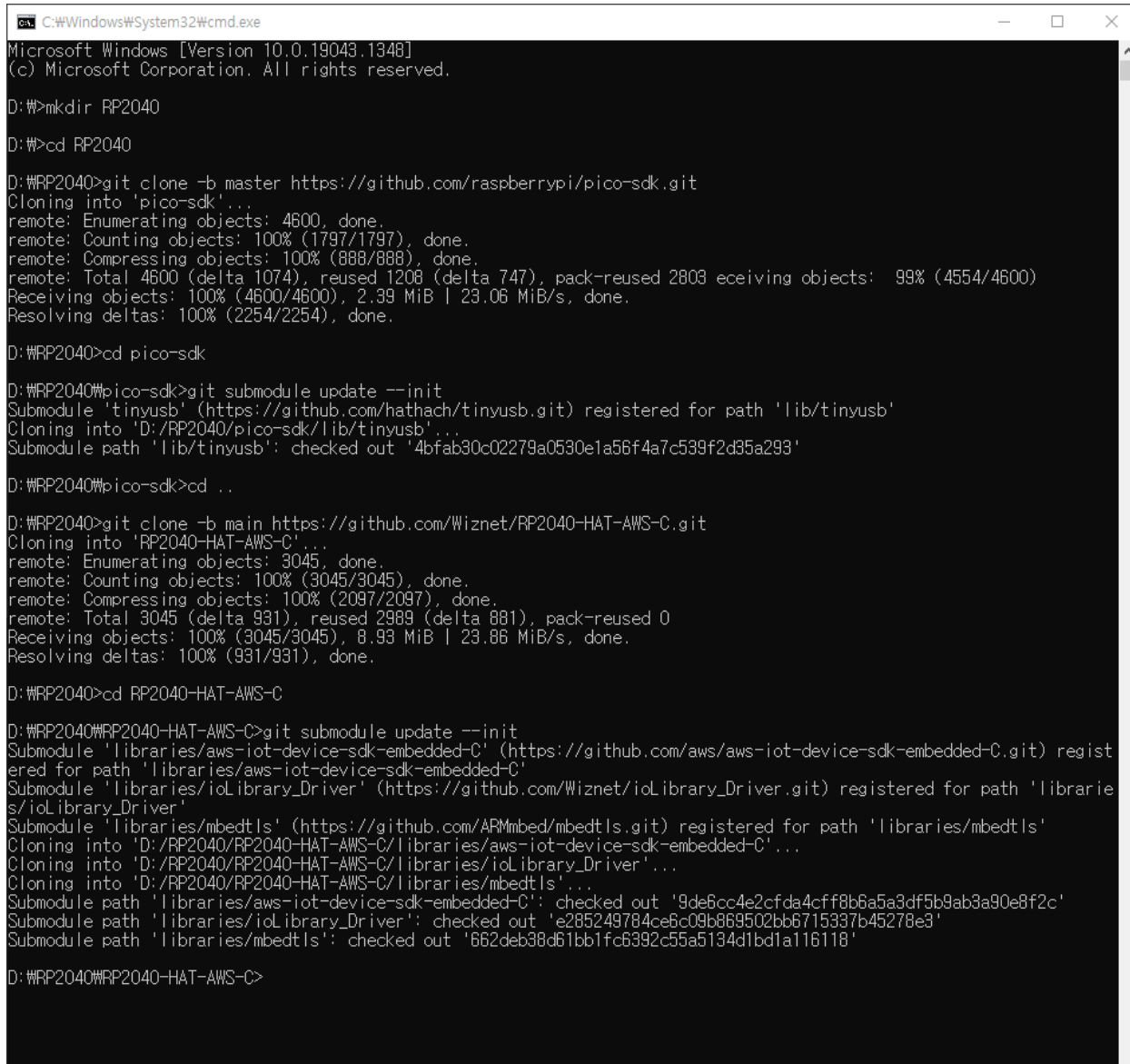2) Clone the Raspberry Pi Pico SDK and WIZnet example repository using below commands

- SDK : https://github.com/raspberrypi/pico-sdk
- Example : https://github.com/Wiznet/RP2040-HAT-AWS-C

```
// create a project directory
D:\>mkdir RP2040
D:\>cd RP2040

// get the SDK
D:\RP2040> git clone -b master https://github.com/raspberrypi/pico-sdk.git
D:\RP2040> cd pico-sdk
D:\RP2040\pico-sdk> git submodule update --init

// get the example
D:\RP2040\pico-sdk> cd ..
```

```
D:\RP2040> git clone -b main https://github.com/Wiznet/RP2040-HAT-AWS-C.git
D:\RP2040> cd RP2040-HAT-AWS-C
D:\RP2040\RP2040-HAT-AWS-C> git submodule update --init
```


Figure 11. Get the SDK and example

3) Set up Visual Studio Code

① Open a new Visual Studio 2019 Developer Command Prompt
② Run the below command to open Visual Studio Code

```
D:> code
```

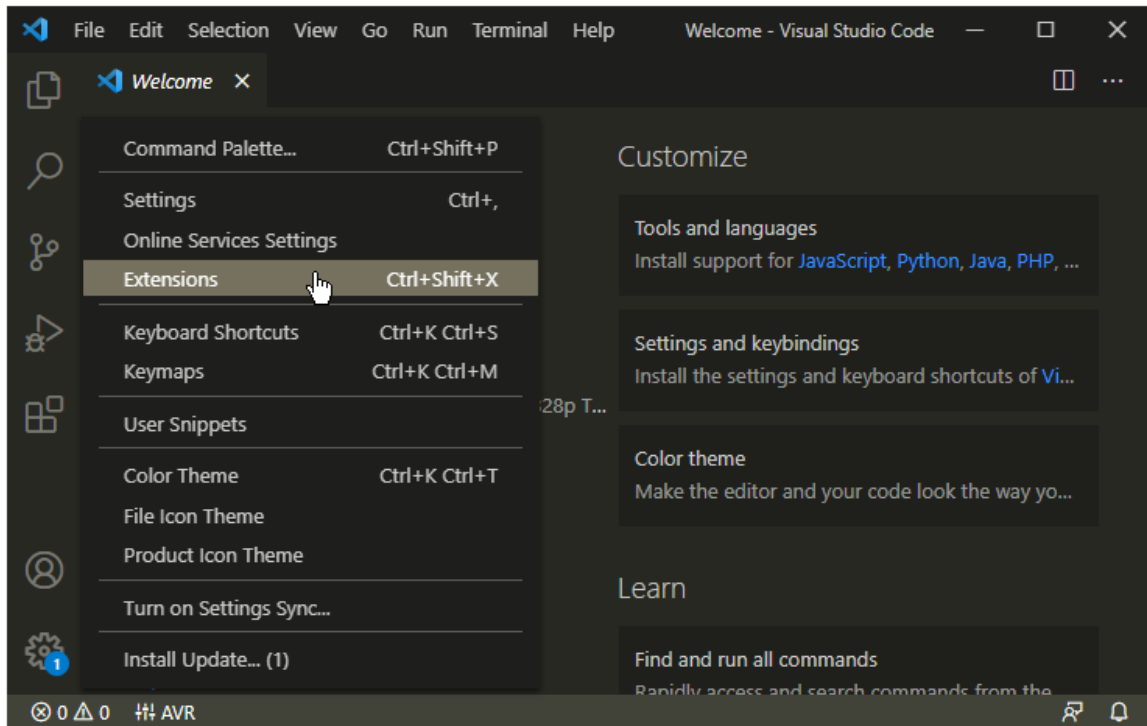③ Opening Visual Studio Code from Developer Command Prompt

④ Open Extensions

Figure 12. Install Extensions in Visual Studio Code

⑤ Install some tools
- CMake Tools
- C/C++

⑥ Open CMake Tools Extension Settings

⑦ Scroll down and set up some items
- Add Cmake: Configure Environment Item as PICO_SDK_PATH
- Add Cmake: Configure Environment Value as D:\RP2040\pico-sdk
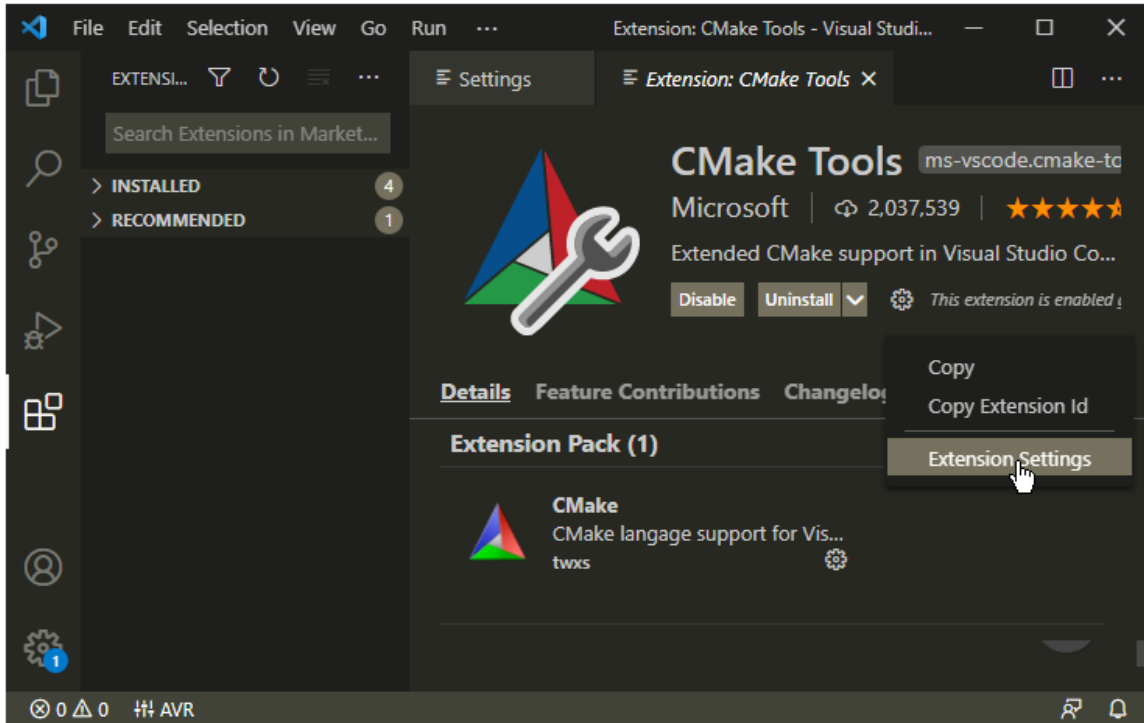- Add Cmake: Generator as NMake Makefiles

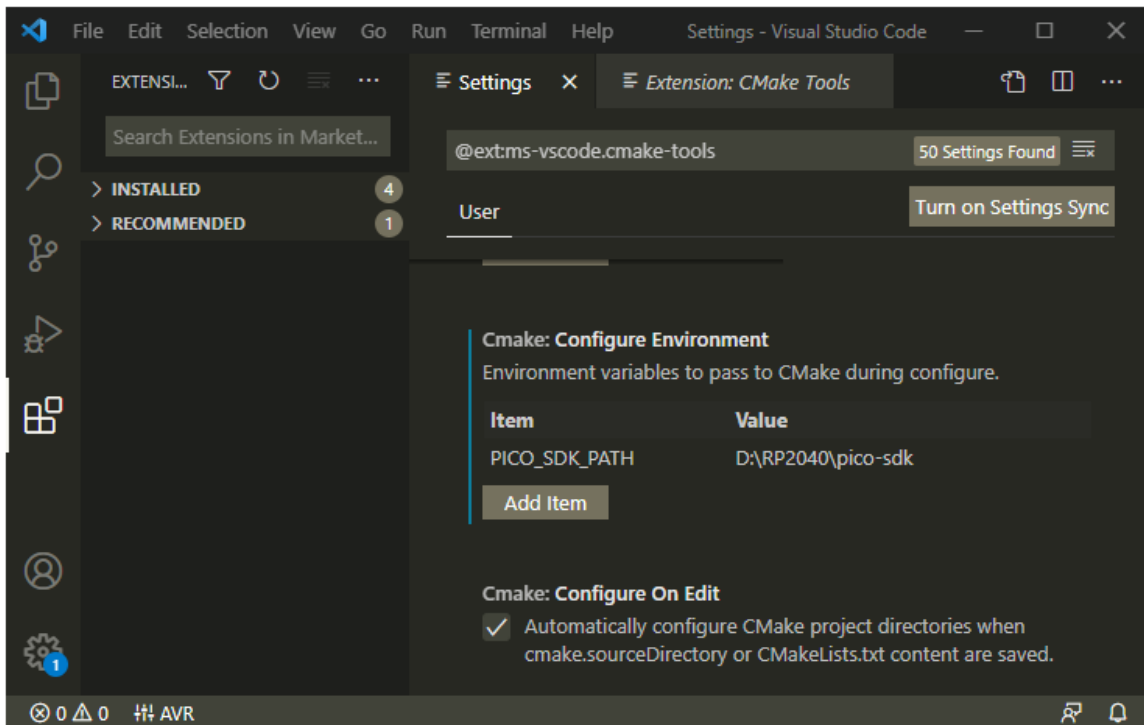*Figure 13. CMake Tools Extension Settings in Visual Studio Code*

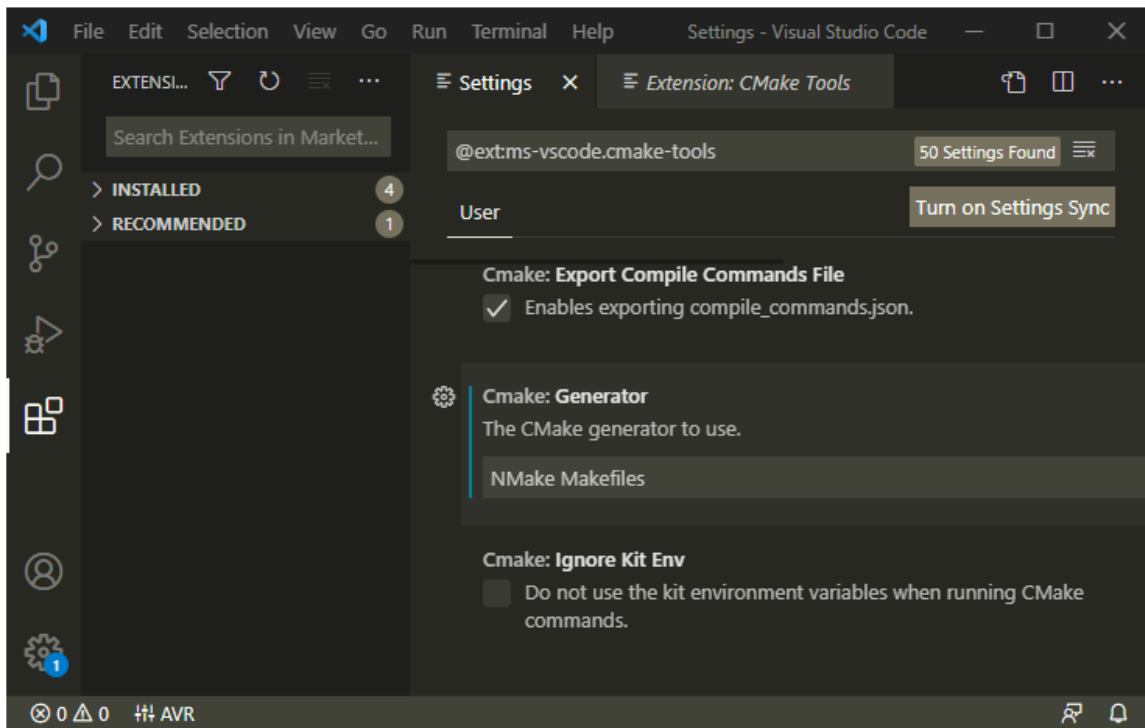*Figure 14. Add CMake Configure Environment path*

Figure 15. Add CMake Generator name

⑧ Add folder <u>RP2040-HAT-AWS-C</u> to Visual Studio Code
⑨ Visual Studio Code will scan for kits
⑩ Select 'Yes' when asked: Would you like to configure project RP2040-HAT-AWS-C?
⑪ Select 'Yes' if you like to configure the project upon opening
⑫ Click CMake in the bottom bar to select the kit RP2040-HAT-AWS-C
⑬ Select Debug / Release based on your preference
⑭ Visual Studio Code will save all file and configure the project
⑮ Click on Build to build all examples, if no error Build will finish with exit code 0

4.2   Other software required to develop and debug applications for the device

Serial terminal program is required for operation check and debugging.

· [Tera Term](#)

You may use your preferred serial terminal program.

4.3 Other pre-requisites

None

4.4 Additional Software References

Refer to the '9.2. Building on MS Windows' section of 'Getting started with Raspberry Pi Pico' document below to set up the development environment.

·   [Getting started with Raspberry Pi Pico](#)

And refer to the links below for instructions how to setup environment and proceed with tests.

·   [Getting Started with AWS IoT SDK Examples](#)
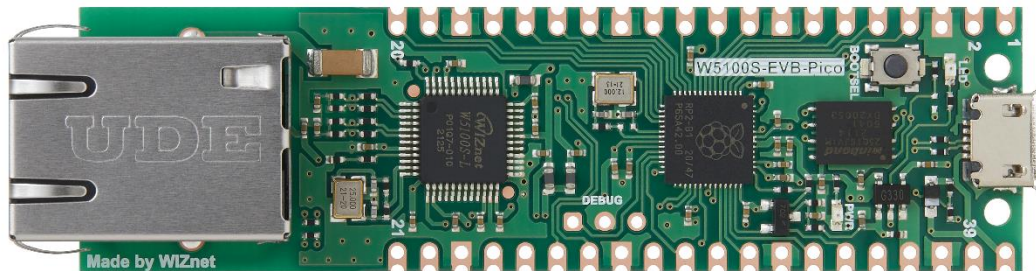·   [Connect AWS IoT through MQTT](#)

5   Set up your hardware

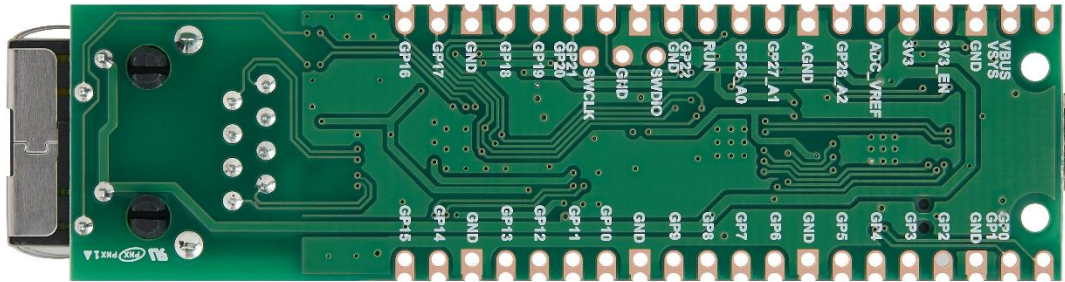*Figure 16. W5100S-EVB-Pico front*

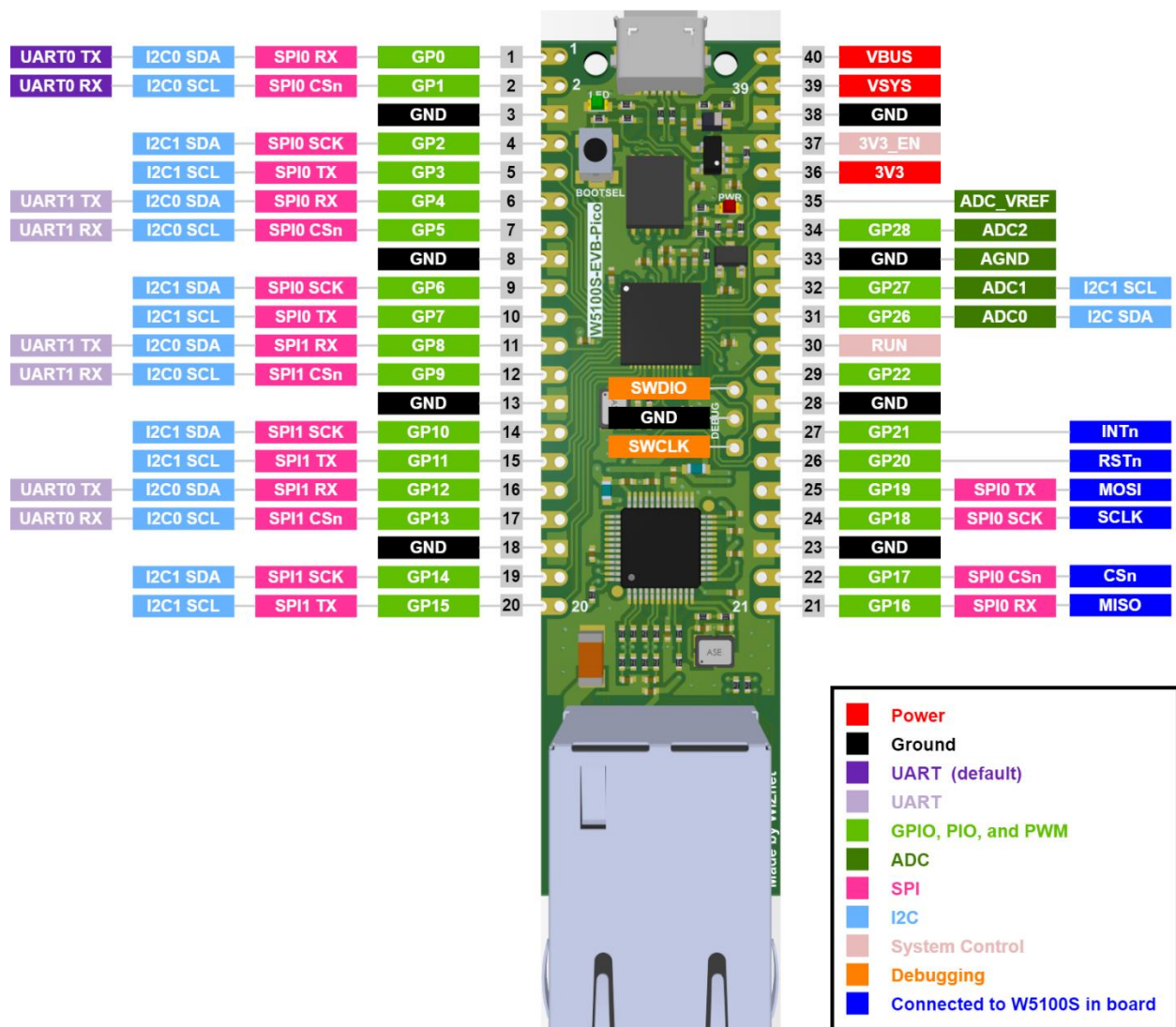*Figure 17. W5100S-EVB-Pico back*

Figure 18. W5100S-EVB-pico pin-out

W5100S-EVB-Pico pin-out is directly connected to the GPIO of RP2040 as shown in the picture above. It has the same pinout as the Raspberry Pi Pico board. However, GPIO16, GPIO17, GPIO18, GPIO19, GPIO20, GPIO21 are connected to W5100S inside the board. These pins enable SPI communication with W5100S to use Ethernet function. If you are using the Ethernet function, these pins cannot be used for any other purpose.

The RP2040 GPIO used inside W5100S-EVB-Pico is as follows.

I/O	Pin name	Description
I	GPIO16	Connected to MISO on W5100S
O	GPIO17	Connected to CSn on W5100S
O	GPIO18	Connected to SCLK on W5100S
O	GPIO19	Connected to MOSI on W5100S
O	GPIO20	Connected to RSTn on W5100S
I	GPIO21	Connected to INTn on W5100S
I	GPIO24	VBUS sense - high if VBUS is present, else low
O	GPIO25	Connected to user LED
I	GPIO29	Used in ADC mode (ADC3) to measure VSYS/3

Apart from GPIO and ground pins, there are 7 other pins on the main 40-pin interface.

Pin no.	Pin name	Description
PIN40	VBUS	Micro-USB input voltage, connected to micro-USB port pin 1. Nominally 5V.
PIN39	VSYS	Main system input voltage, which can vary in the allowed range 4.3V to 5.5V, and is used by the on-board LDO to generate the 3.3V.
PIN37	3V3_EN	Connects to the on-board LDO enable pin. To disable the 3.3V (which also de-powers the RP2040 and W5100S), short this pin low.
PIN36	3V3	Main 3.3V supply to RP2040 and W5100S, generated by the on-board LDO.
PIN35	ADC_VREF	ADC power supply (and reference) voltage, and is generated on W5100S-EVB-Pici by filtering the 3.3V supply.
PIN33	AGND	Ground reference for GPIO26-29.
PIN30	RUN	RP2040 enable pin, To reset RP2040, short this pin low.

6   Setup your AWS account and Permissions

Refer to the instructions at Set up your AWS Account. Follow the steps outlined in these sections to create your account and a user and get started:
- Sign up for an AWS account and
- Create a user and grant permissions.
- Open the AWS IoT console

Pay special attention to the Notes.

7   Create Resources in AWS IoT

Refer to the instructions at <u>Create AWS IoT Resources</u>. Follow the steps outlined in these sections to provision resources for your device:

- Create an AWS IoT Policy
- Create a thing object

Pay special attention to the Notes.

8   Provision the Device with credentials

You need to enter the root certificate, client certificate and private key that were downloaded earlier.

Root certificate uses the RSA 2048 bit key, Amazon Root CA 1, and does not use the public key.

Device certificate and key can be set in 'mqtt_certificate.h' in 'RP2040-HAT-AWS-C/Interface/' directory.

```
uint8_t mqtt_root_ca[] =
"-----BEGIN CERTIFICATE-----\r\n"
"...\r\n"
"-----END CERTIFICATE-----\r\n";

uint8_t mqtt_client_cert[] =
"-----BEGIN CERTIFICATE-----\r\n"
"...\r\n"
"-----END CERTIFICATE-----\r\n";

uint8_t mqtt_private_key[] =
"-----BEGIN RSA PRIVATE KEY-----\r\n"
"...\r\n"
"-----END RSA PRIVATE KEY-----\r\n";
```

Figure 19. Set up certificates

9 Build the demo

① Click 'build' in the status bar at the bottom of Visual Studio Code or press the 'F7' button on the keyboard to build.

② When the build is completed, 'aws_iot_mqtt.uf2' is generated in 'RP2040-HAT-AWS-C/build/examples/aws_iot_mqtt/' directory.

10 Run the demo

① While pressing the BOOTSEL button of Raspberry Pi Pico or W5100S-EVB-Pico power on the board, the USB mass storage 'RPI-RP2' is automatically mounted.
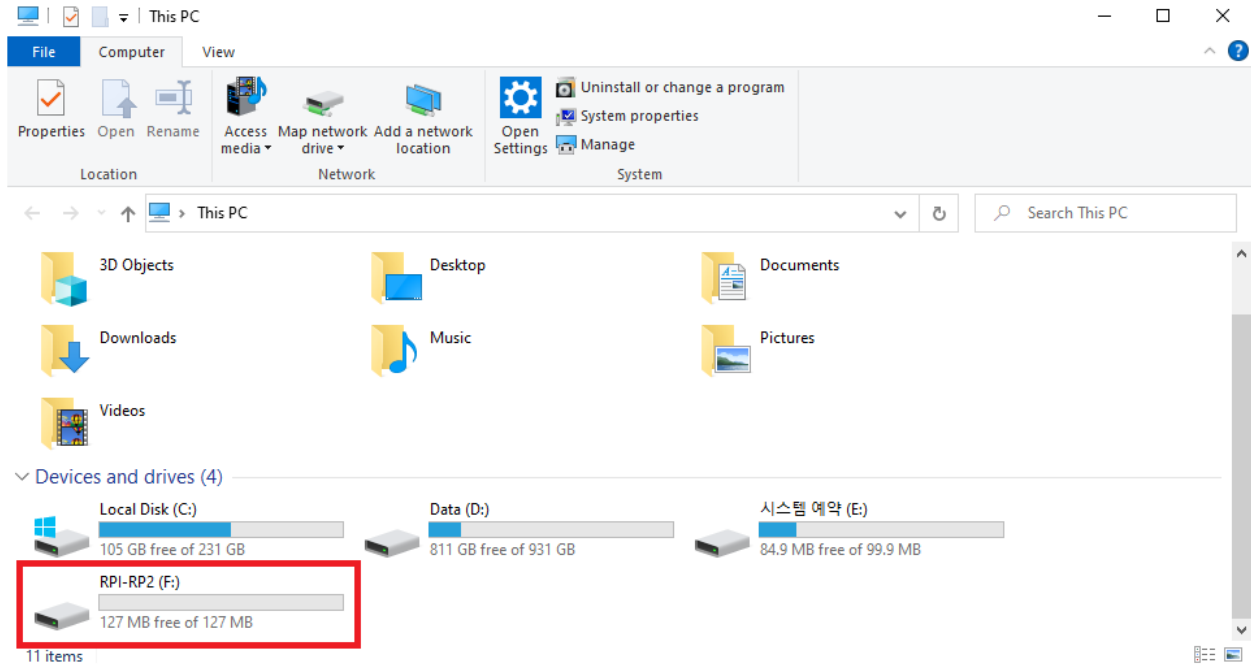
Figure 20. Automatically mounted USB mass storage 'RPI-RP2'

② Drag and drop 'aws_iot_mqtt.uf2' onto the USB mass storage device 'RPI-RP2'.

③ Connect to the serial COM port of Raspberry Pi Pico or W5100S-EVB-Pico with Tera Term.
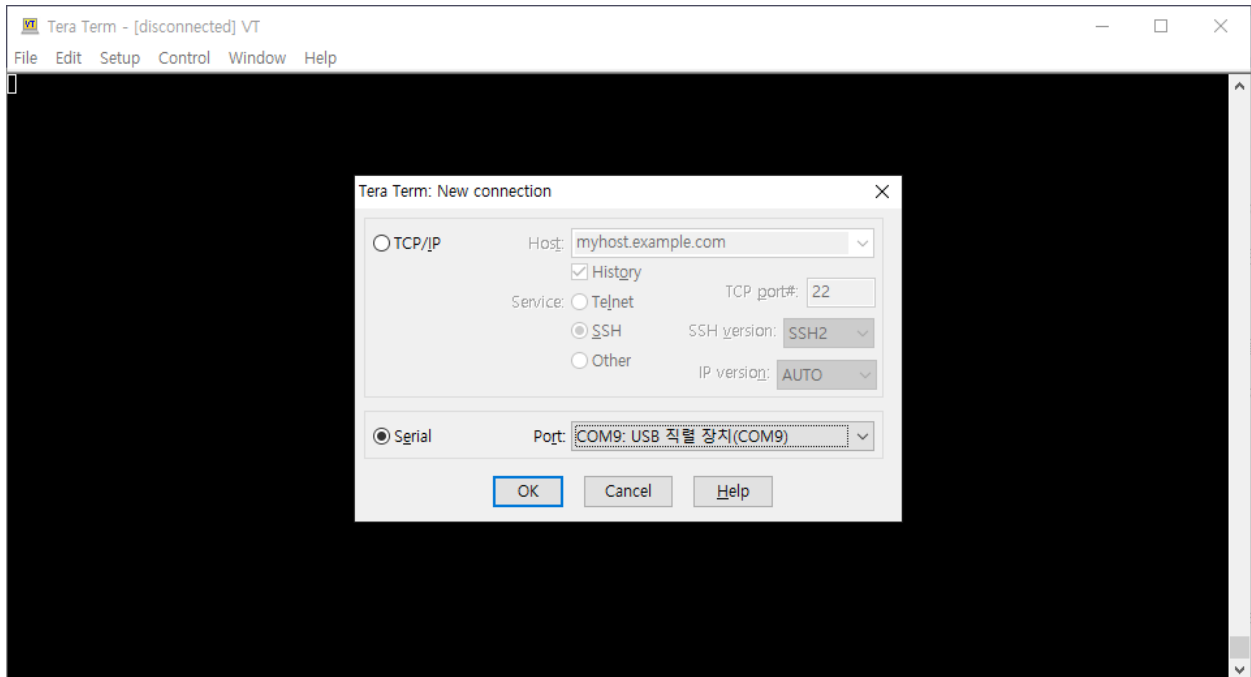
Figure 21. Connect to the serial COM port of W5100S-EVB-Pico

④ When connecting to the serial COM port of W5100S-EVB-Pico, use following settings to set up the serial port.

- Baud rate : 115,200
- Data bit : 8
- Parity bit : none
- Stop bit : 1
- Flow control : none
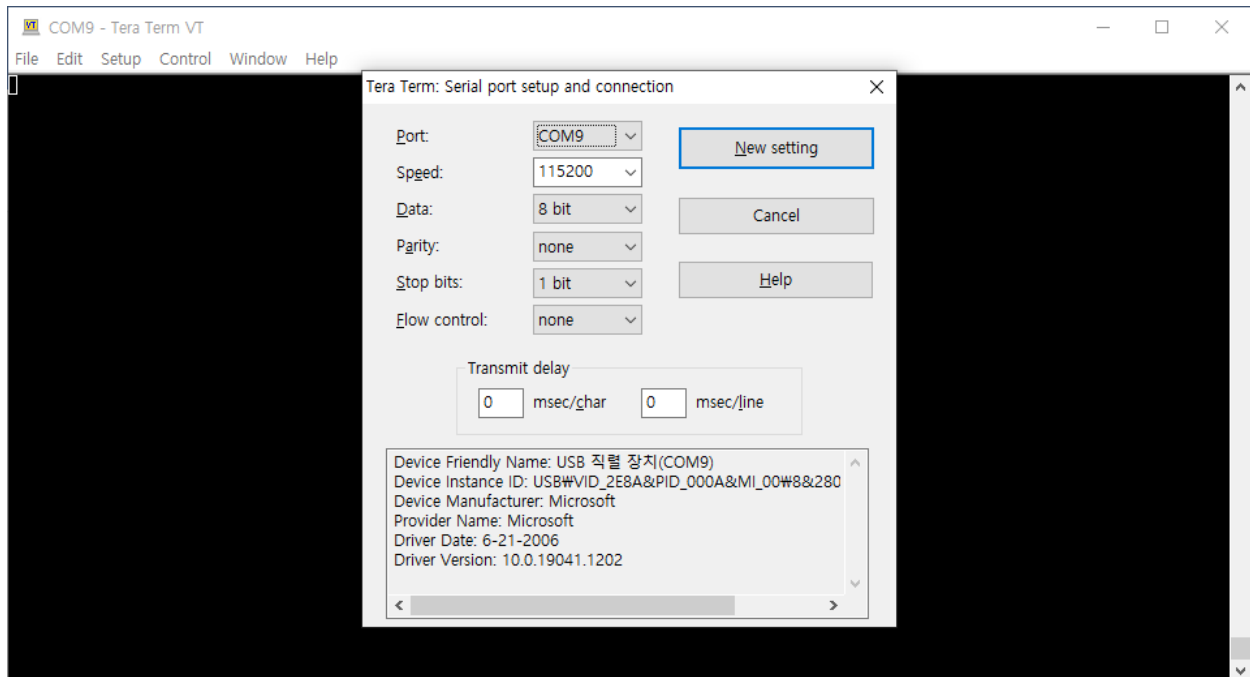
Figure 22. Set up serial port

⑤ Reset your board.

⑥ If the Connect AWS IoT through MQTT example works normally on W5100S-EVB-Pico, you can see the network information of W5100S-EVB-Pico, connecting to the AWS IoT and publishing the message.

Figure 23. Publish the message from W5100S-EVB-Pico

Figure 24. Receive the message about subscribed topic from IoT Core

Copyright 2021 WIZnet Co., Ltd. All rights reserved.

⑦ If you publish the message through the test function in AWS IoT Core to the subscribe topic, you can see that the W5100S-EVB-Pico receive the message about the subscribe topic.
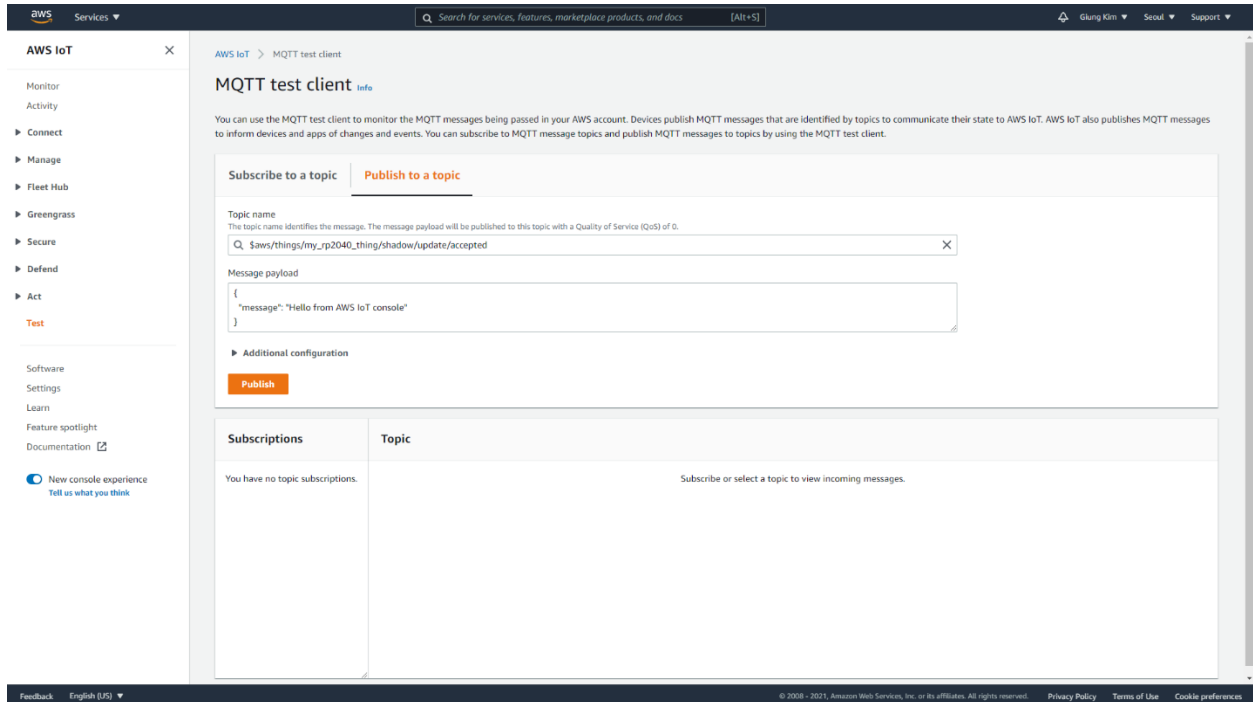
Figure 25. Publish the message from IoT Core

Figure 26. Receive the message about subscribed topic from W5100S-EVB-Pico

11 Debugging

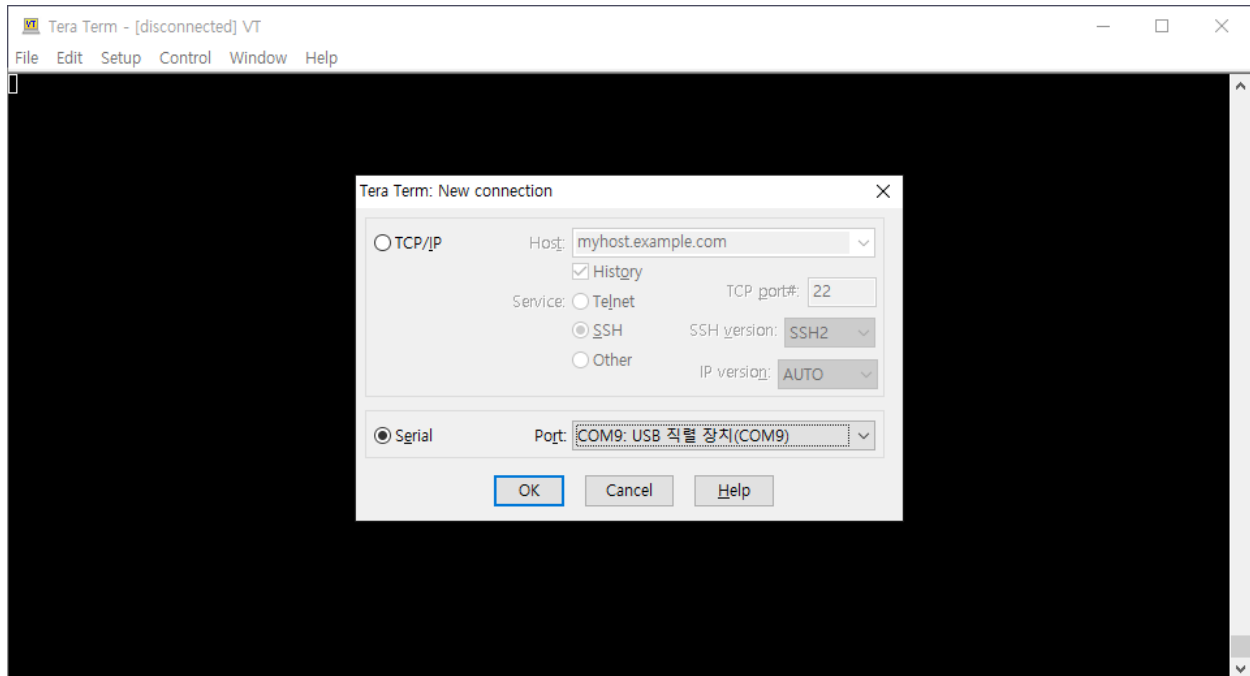① Connect to the serial COM port of W5100S-EVB-Pico with Tera Term to view logs and debugging.

Figure 27. Connect to the serial COM port of W5100S-EVB-Pico

② When connecting to the serial COM port of W5100S-EVB-Pico, use following settings to set up the serial port.

- Baud rate : 115,200
- Data bit : 8
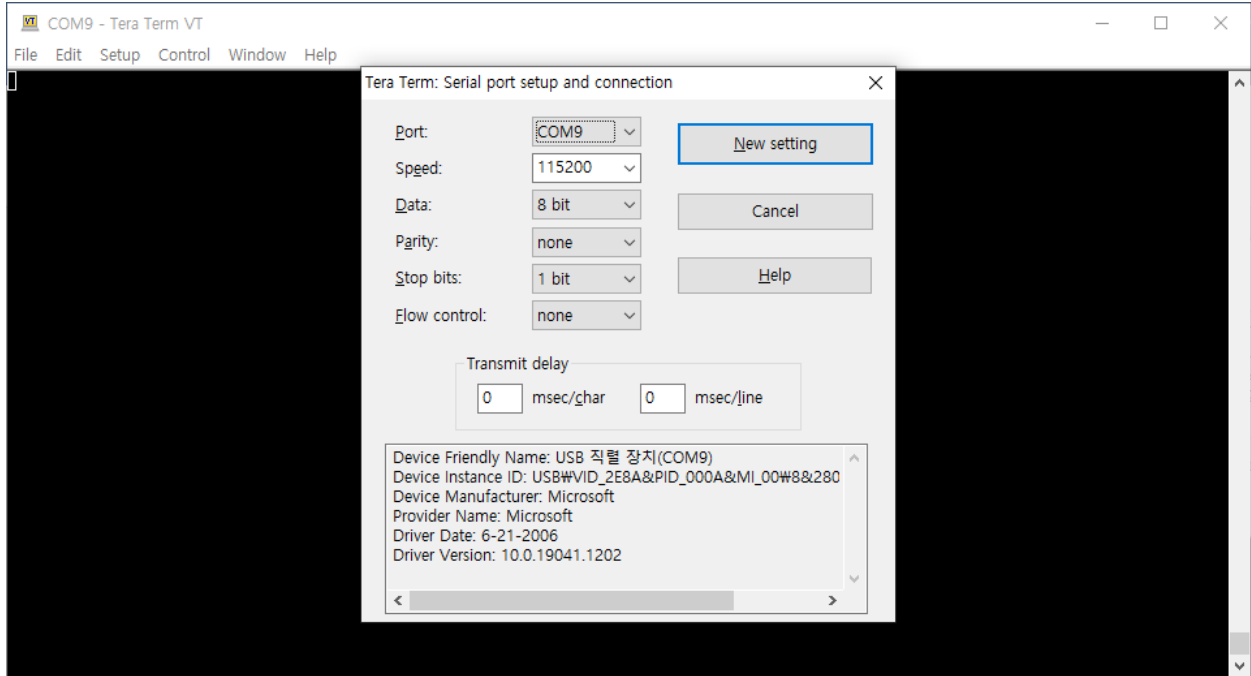- Parity bit : none
- Stop bit : 1
- Flow control : none

Figure 28. Set up serial port

③ If everything works normally, you can see the W5100S-EVB-Pico's network information, publishing the message at intervals of 10 seconds and receiving the message about the subscribe topic through logs.

Figure 29. works normally

④ If there occurs any problem, the error log is printed showing function what the problem has occurred.

Figure 30. Failed in DNS

12 Troubleshooting

If you have any questions or problems while using the W5100S-EVB-Pico examples, please leave them at the link below.

- · [WIZnet Developer Forum](#)
- · [RP2040-HAT-C Issues](#)
- · [RP2040-HAT-AWS-C Issues](#)