

RZ/A2M Group

Ethernet Sample Program

Introduction

This document describes the sample program that uses the RZ/A2M group MCU to perform Ethernet functions.

Target Device

RZ/A2M

Contents

1. Overview	3
2. Operation Confirmation Conditions	4
3. Software Description.....	6
4. How to confirm works of sample program	7
4.1 Echo Server Settings.....	7
4.1.1 Modifying IP Address	7
4.1.2 Installing Simple TCPIP services	10
4.1.3 Firewall settings.....	12
4.1.4 Run "Simple TCPIP services".....	21
4.2 HOST PC Settings.....	23
4.3 Output of the sample program.....	24
5. Connecting Amazon Web Service	25
5.1 AWS setting.....	25
5.1.1 Open AWS IoT Core Service	25
5.1.2 Creating AWS IoT things.....	26
5.1.3 Create AWS IoT policy	30
5.1.4 Attach an AWS IoT Policy to a Device Certificate.....	35
5.2 Certification settings	36
5.2.1 Setting AWS IoT Information.....	36
5.2.2 Setting certificate of AWS IoT	38
5.3 Setting RZ/A2M project	40
5.3.1 Turn off debugging message.....	40
5.3.2 Set the MAC address	40
5.3.3 Setting the IP address	41
5.3.4 Switch the demonstration program to use.....	42
5.3.5 Import, build, and download	42
5.4 Running MQTTEcho demonstration.....	43
5.4.1 Setting for AWS IoT test.....	43
6. Reference Documents	45
Revision History.....	46

1. Overview

The Ethernet Sample Program (sample program) supports the functions to communicate between RZ/A2M Evaluation Board Kit and Echo server. RZ/A2M transmits the data to Echo server and checks the received data from Echo server are same as sent data. This Sample Program uses FreeRTOS-TCP. Application program is the sample code provided by FreeRTOS. For the details of this program, refer to following URL:

https://www.freertos.org/FreeRTOS-Plus/FreeRTOS_Plus_TCP/TCP_Echo_Clients.html

Table 1-1 Peripheral devices and Usage

Peripheral devices	Usage
Ethernet Controller (ETHERC)	Controls Ethernet.
DMA Controller for Ethernet Controller (EDMACa)	Controls Ethernet.
SCIF	Outputs communication status using Terminal Application.

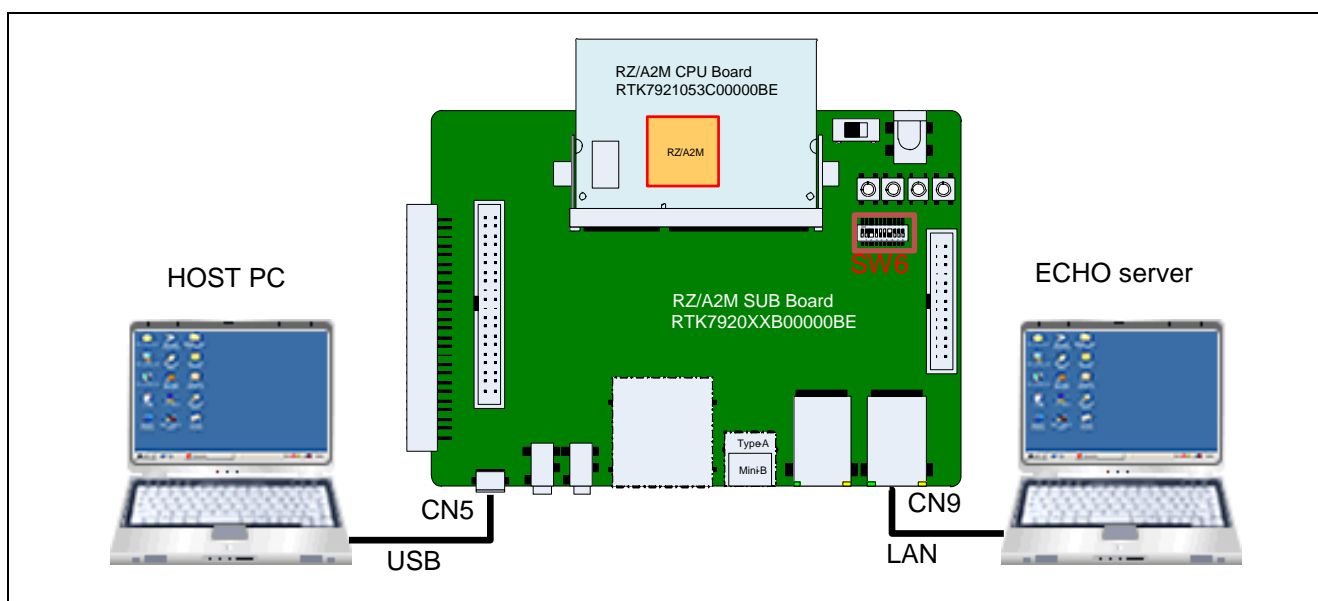


Figure 1-1 Operation check conditions

2. Operation Confirmation Conditions

The sample code of this application has been verified by following conditions.

Table 2-1 Peripheral device used(1/2)

Peripheral device	Usage
MCU used	<ul style="list-style-type: none"> RZ/A2M
Operating frequency[MHz] (Note)	CPU Clock (Iφ) : 528MHz Image processing clock (Gφ) : 264MHz Internal Bus Clock (Bφ) : 132MHz Peripheral Clock 1 (P1φ) : 66MHz Peripheral Clock 0 (P0φ) : 33MHz QSPI0_SPCLK : 66MHz CKIO : 132MHz
Operating voltage	Power supply voltage (I/O): 3.3 V Power supply voltage (either 1.8V or 3.3V I/O (PVcc SPI)) : 3.3V Power supply voltage (internal): 1.2 V
Integrated development environment	e2 studio V7.4.0
C compiler	“GNU Arm Embedded Tool chain 6-2017-q2-update” compiler options(except directory path) Release: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0

Note: The operating frequency used in clock mode 1 (Clock input of 24MHz from EXTAL pin)

Table 2-2 Peripheral device used(2/2)

Operation mode	Boot mode 3 (Serial Flash boot 3.3V)
Terminal software communication settings	Communication speed: 115200bps Data length: 8 bits Parity: None Stop bits: 1 bit Flow control: None
Board to be used	RZ/A2M CPU board RTK7921053C00000BE RZ/A2M SUB board RTK79210XXB00000BE
Device (functionality to be used on the board)	Serial flash memory allocated to SPI multi-I/O bus space (channel 0) Manufacturer : Macronix Inc. Model Name : MX25L51245GXD RL78/G1C (This device communications the host PC by convert USB Communication and Serial Communication.) Ethernet PHY RTL8201FL-VB-CG (Realtek)

Note: The operating frequency used in clock mode 1 (Clock input of 24MHz from EXTAL pin)

3. Software Description

Figure 3-1 shows the system block diagram of sample program. In this sample program, Echo Client Task is created by FreeRTOS-standard timer task. The communication application in Echo Client Task communicates to Echo Server.

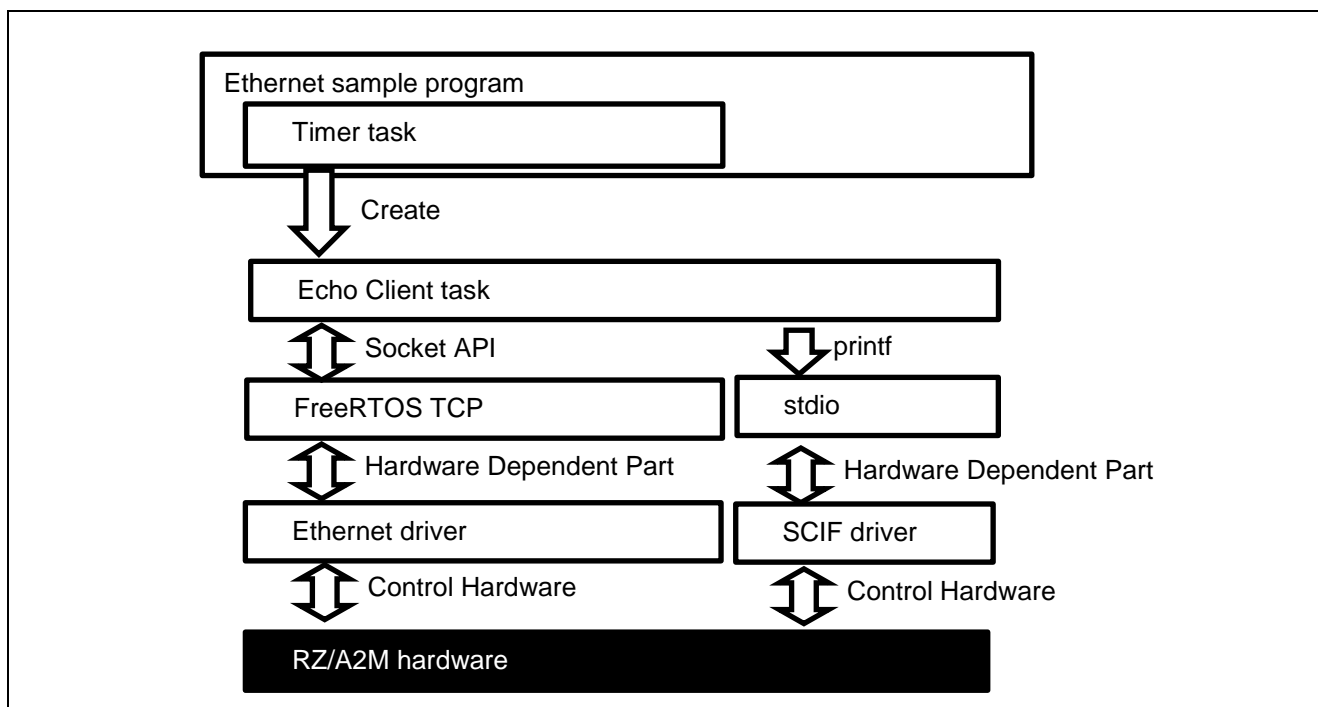


Figure 3-1 Sample program System Block Diagram

4. How to confirm works of sample program

To use this sample program, you need to prepare the echo server. This section explains the settings needed to use this sample program.

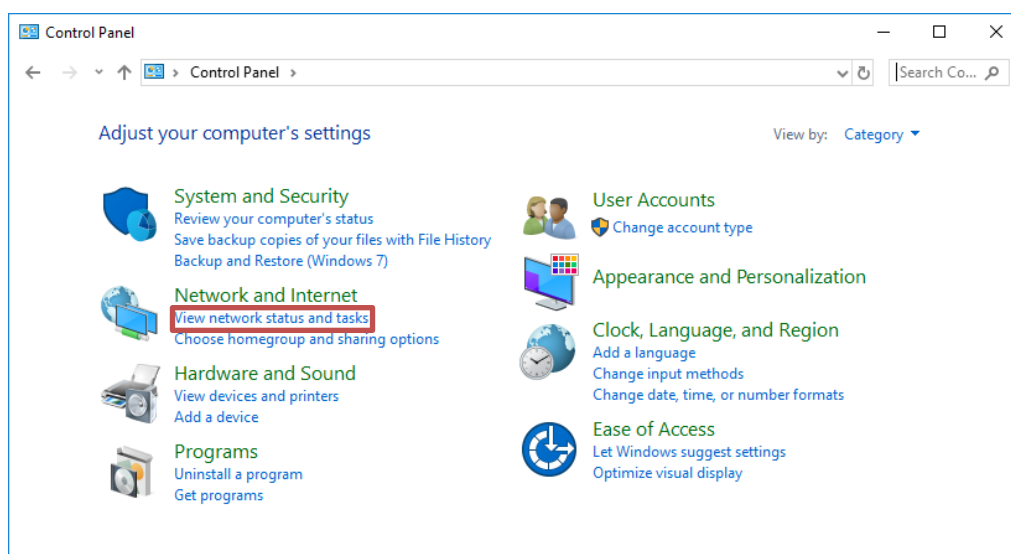
4.1 Echo Server Settings

This section explains the settings to use Windows 10 PC as an echo server.

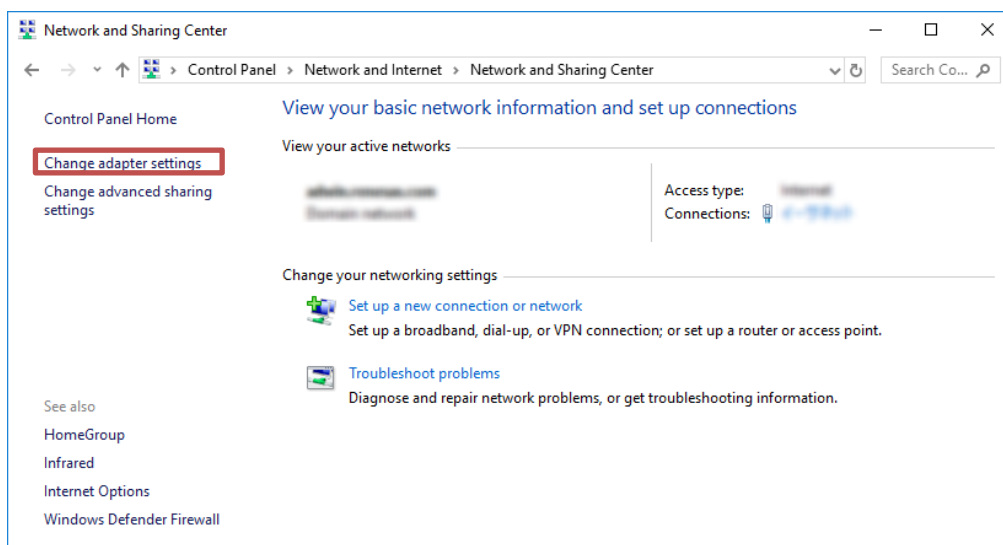
4.1.1 Modifying IP Address

Open the “Control Panel” by selecting “Windows System” of Windows start menu.

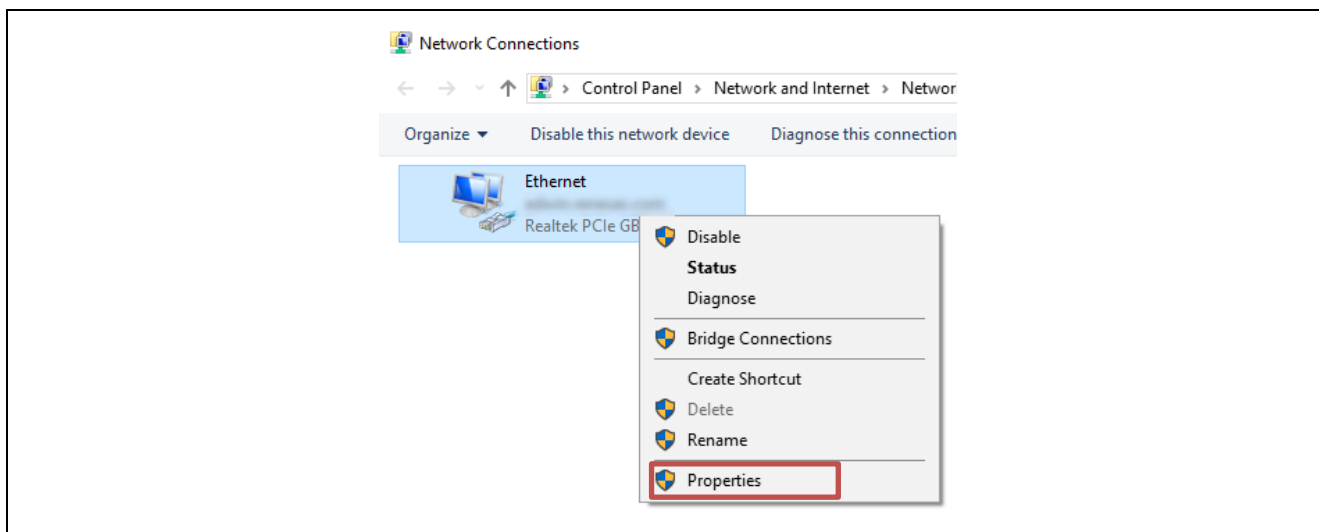
Select “View network status and tasks” in “Network and Internet”.



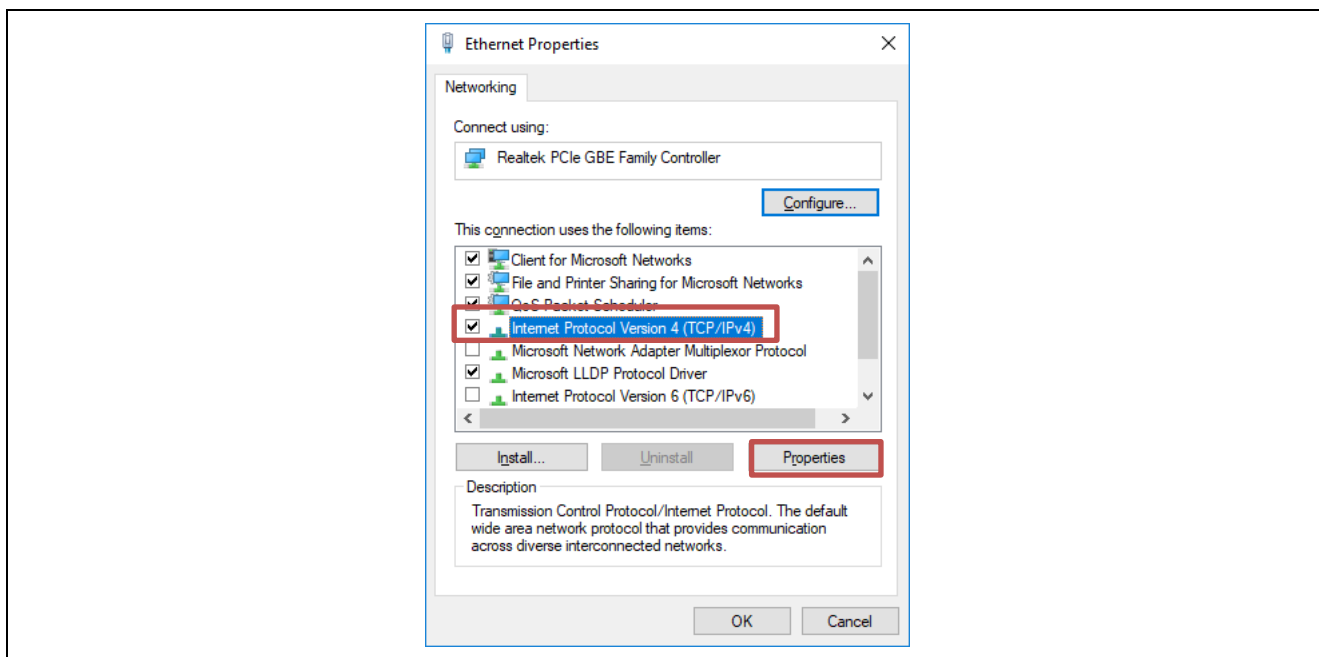
Press “Change adapter Settings”.



Select “Properties” after right-clicking “Ethernet”.



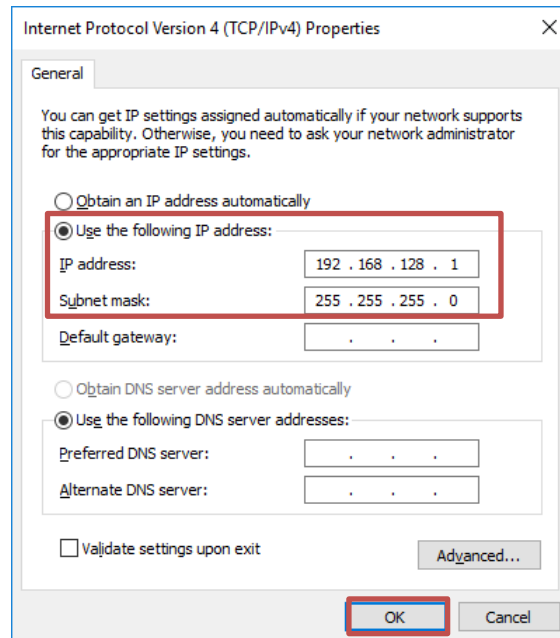
Select “Internet Protocol Version4(TCP/IPv4)” and press “Properties”.



Select "Use the following IP address:" and set following addresses:

- IP address: 192.168.128.1
- Subnet mask: 255.255.255.0

Then, click "OK".



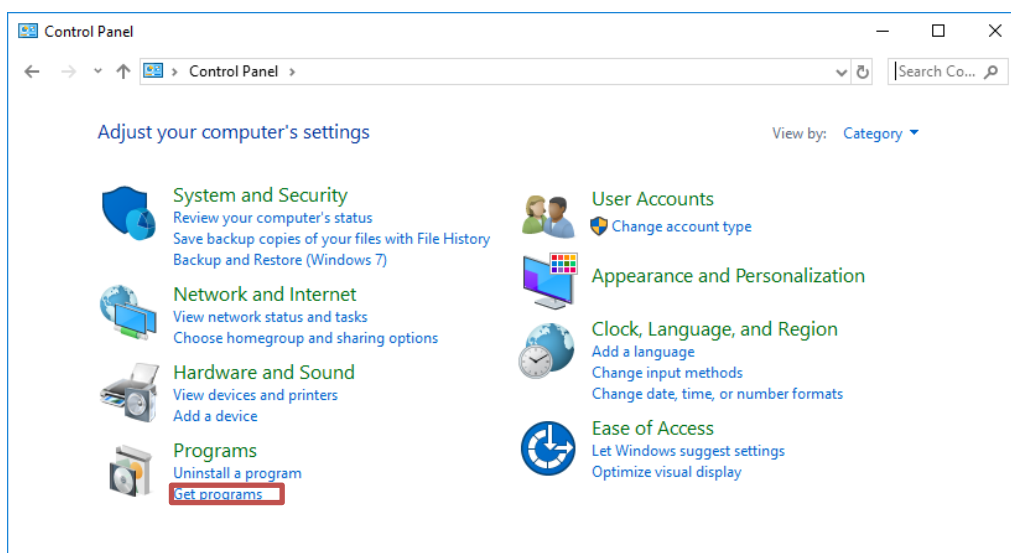
4.1.2 Installing Simple TCPIP services

Install “Simple TCPIP services” bundled in Windows to use echo function. This service includes echo function.

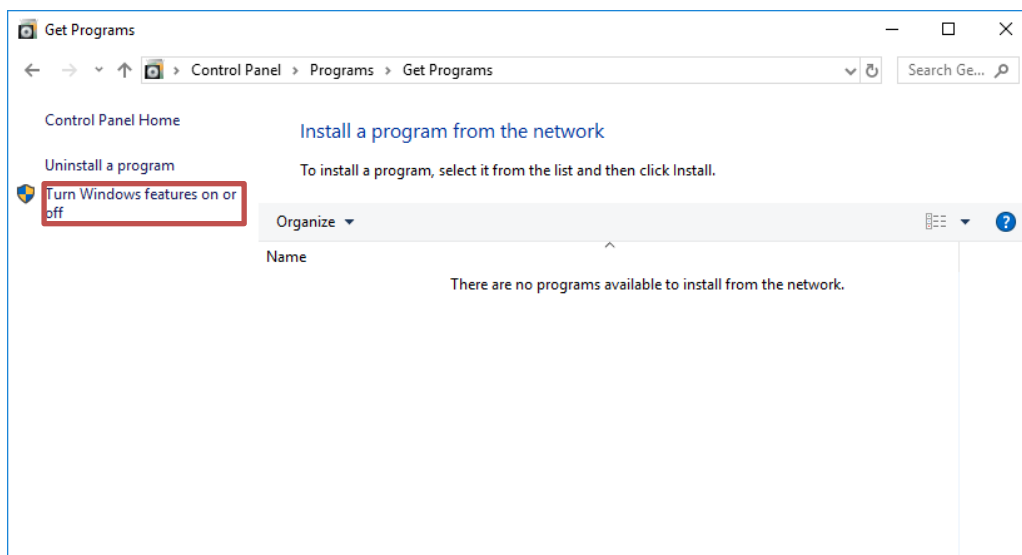
After checking this program, uninstalling echo server is recommended because Simple TCP / IP service can be a factor to create security holes.

Open the “Control Panel” by selecting “Windows System” of Windows start menu.

Select “Get Programs” in “Programs”.

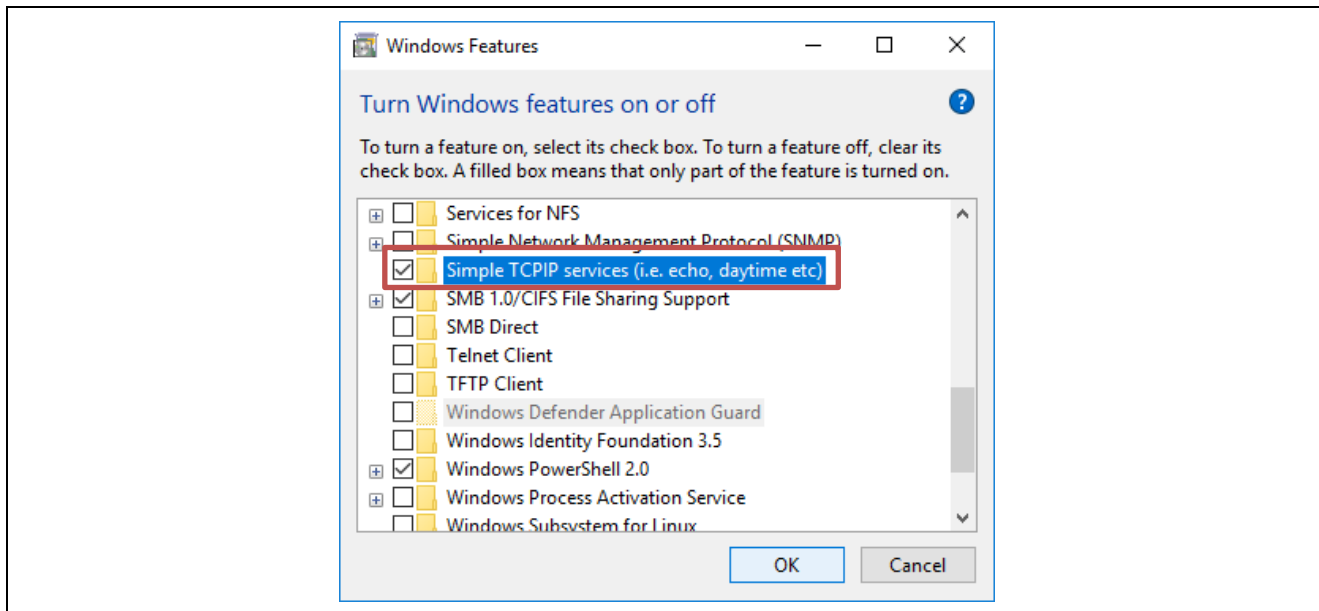


Press “Turn Windows features on or off”.



Check "Simple TCPIP services (i.e. echo, daytime etc)", and press "OK".

After checking the work of this sample program, "Simple TCPIP services (i.e. echo, daytime etc)" will be uninstalled by unchecking the check-box and press "OK".

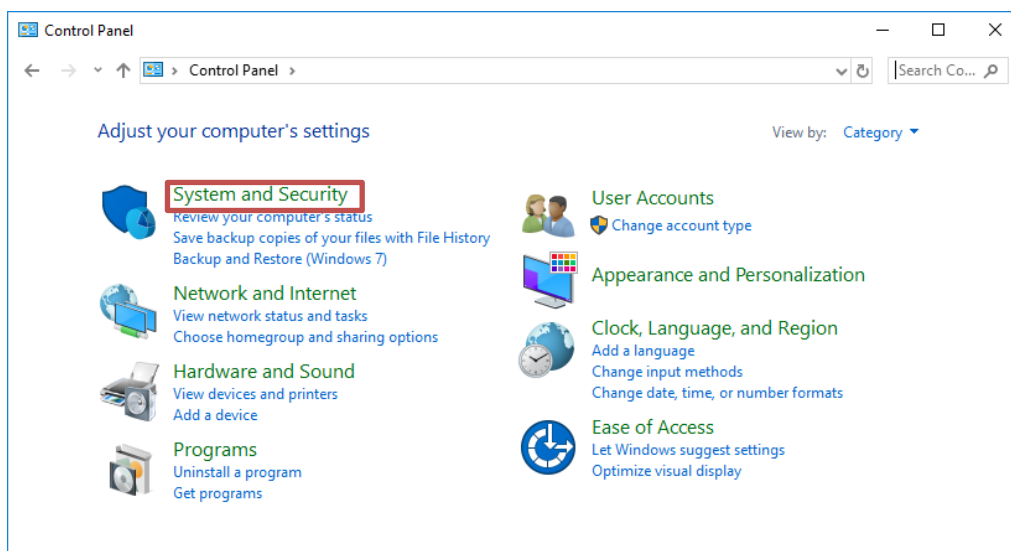


4.1.3 Firewall settings

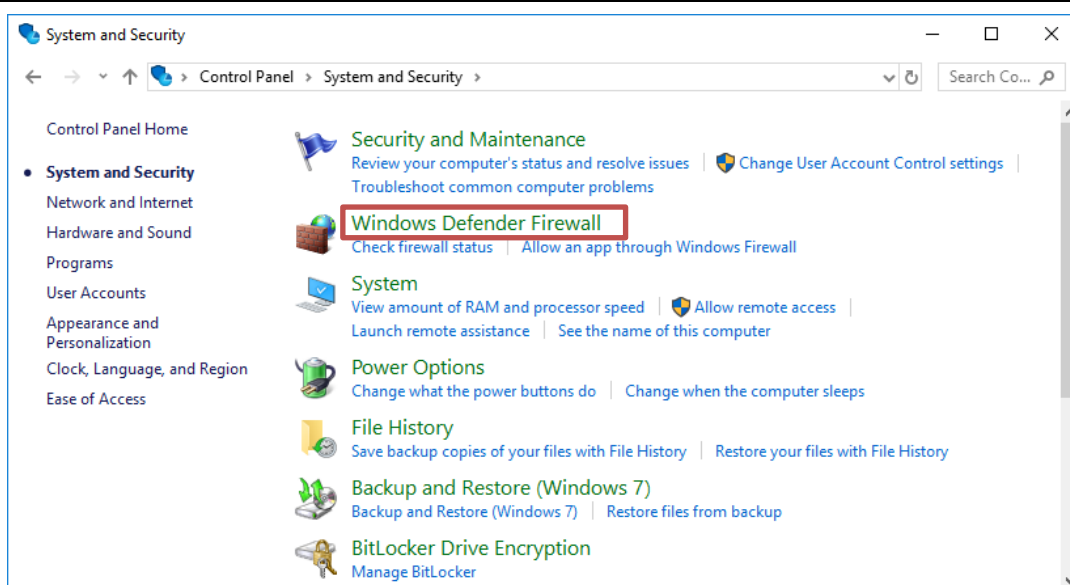
Echo function in “Simple TCPIP services” uses TCP port 7. This section explains the settings to allow connection via TCP port 7.

Open the “Control Panel” by selecting “Windows System” of Windows start menu.

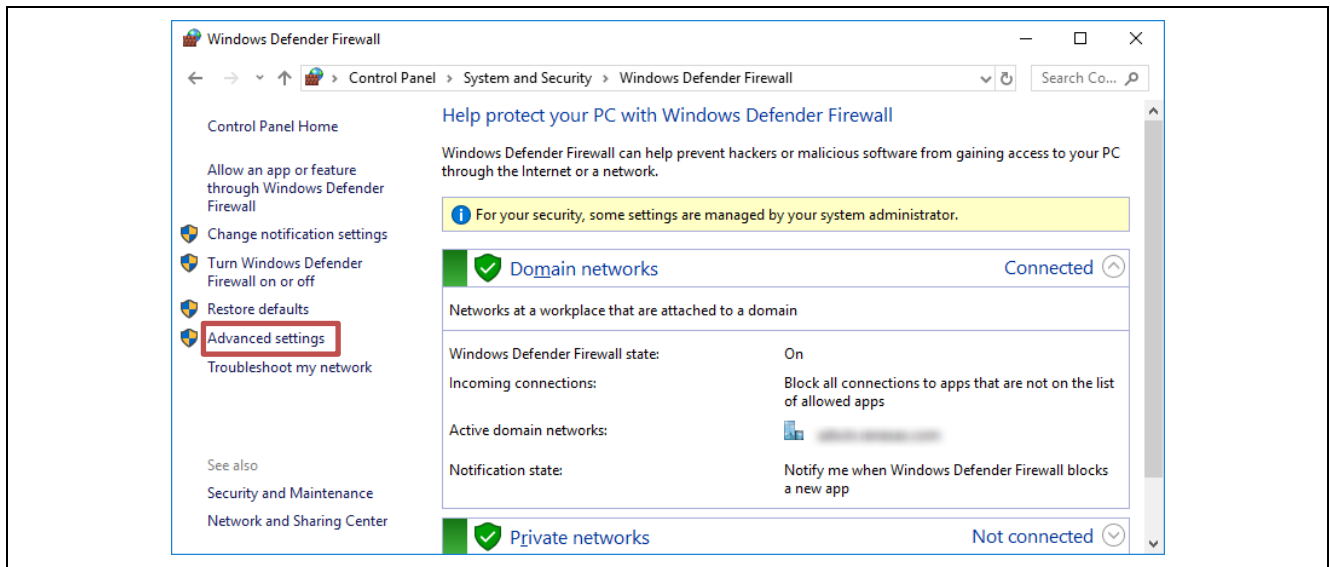
Select “System and Security”.



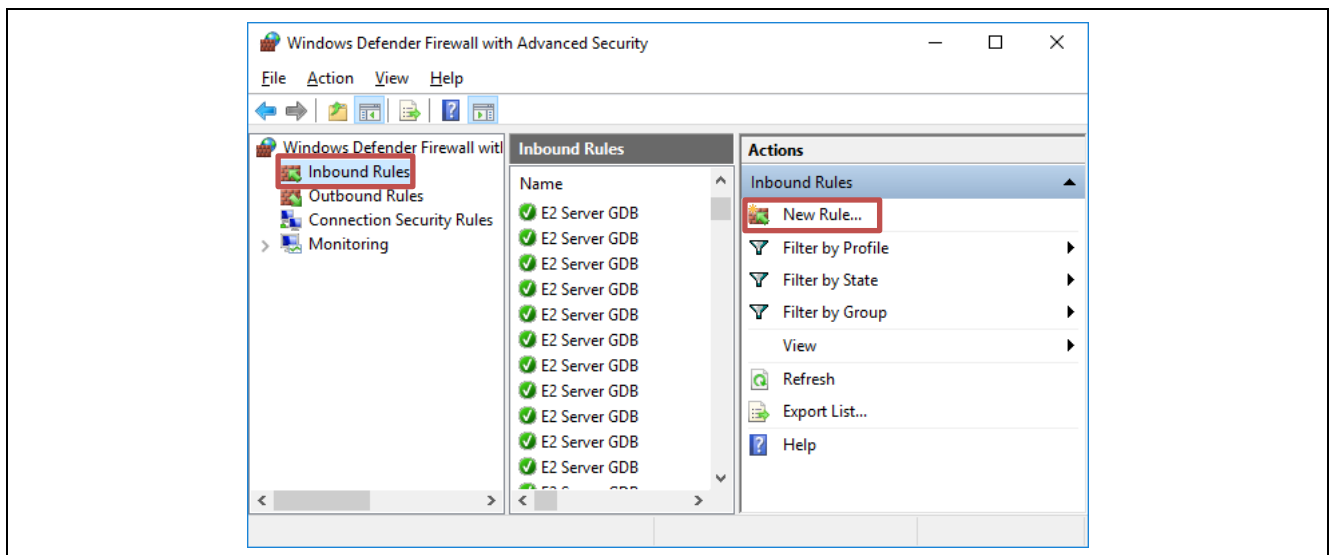
Press “Windows Defender Firewall”.



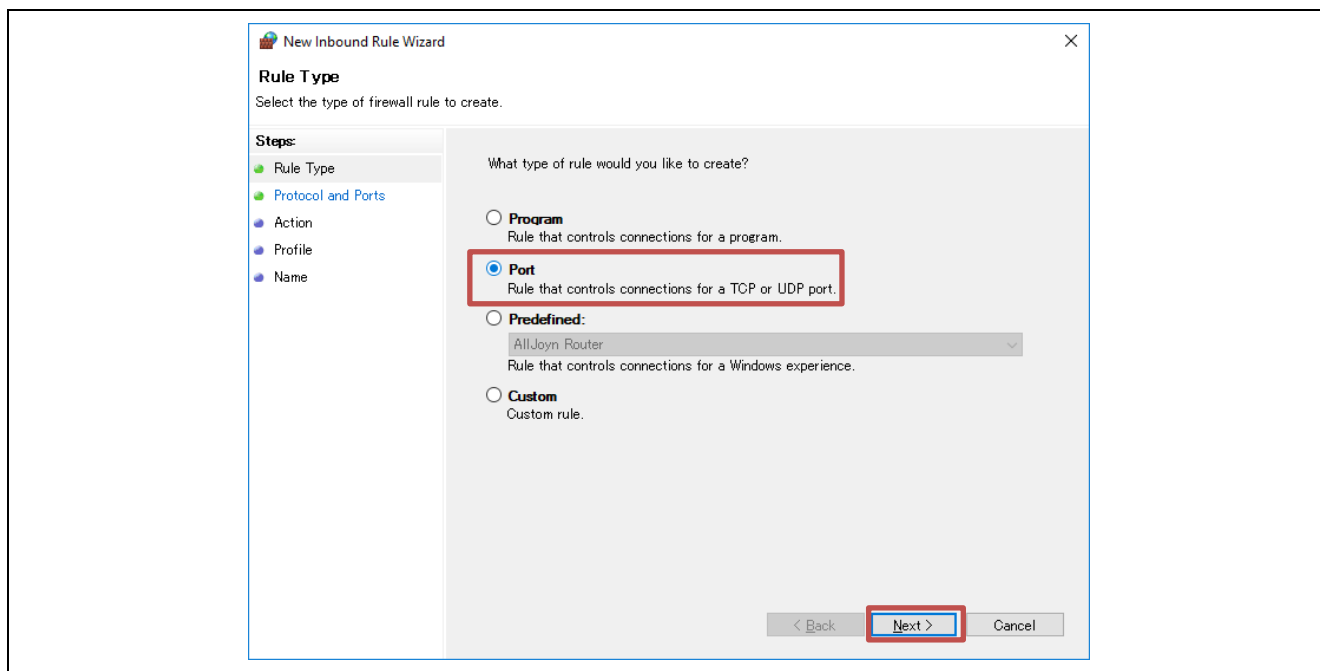
Press “Advanced settings”.



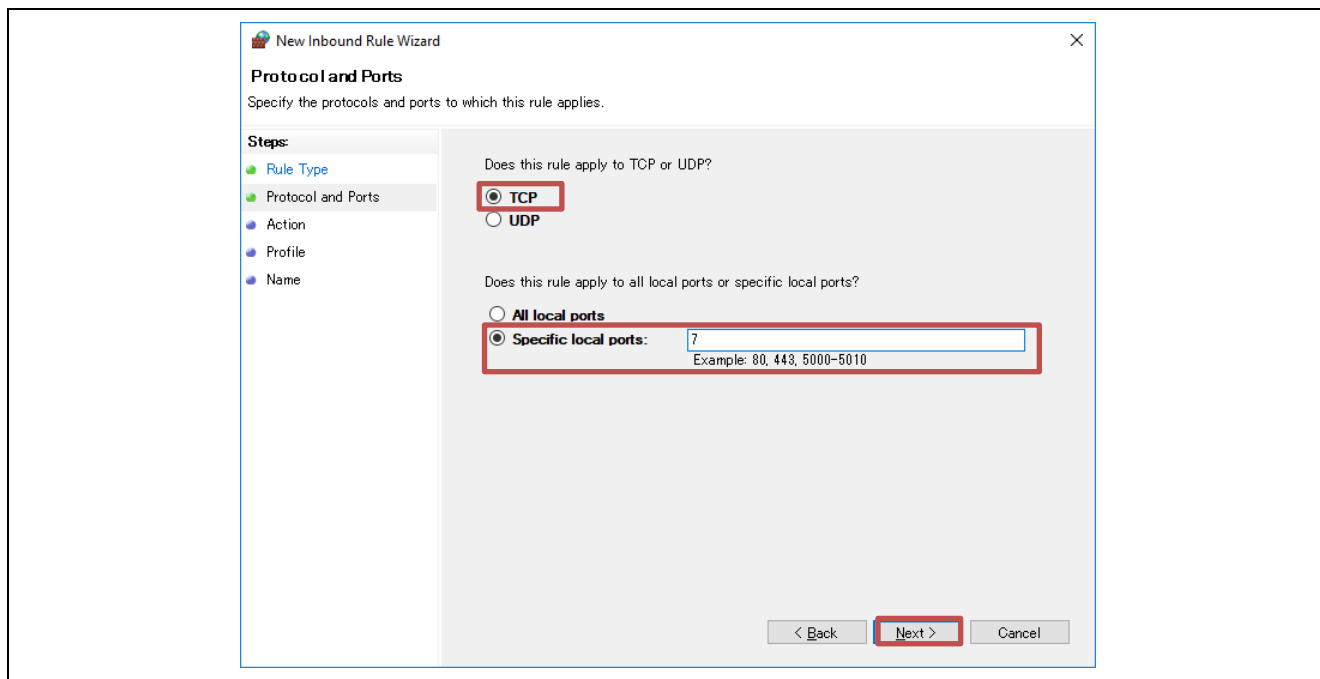
Select “Inbound Rules”, and Press “New Rule...”



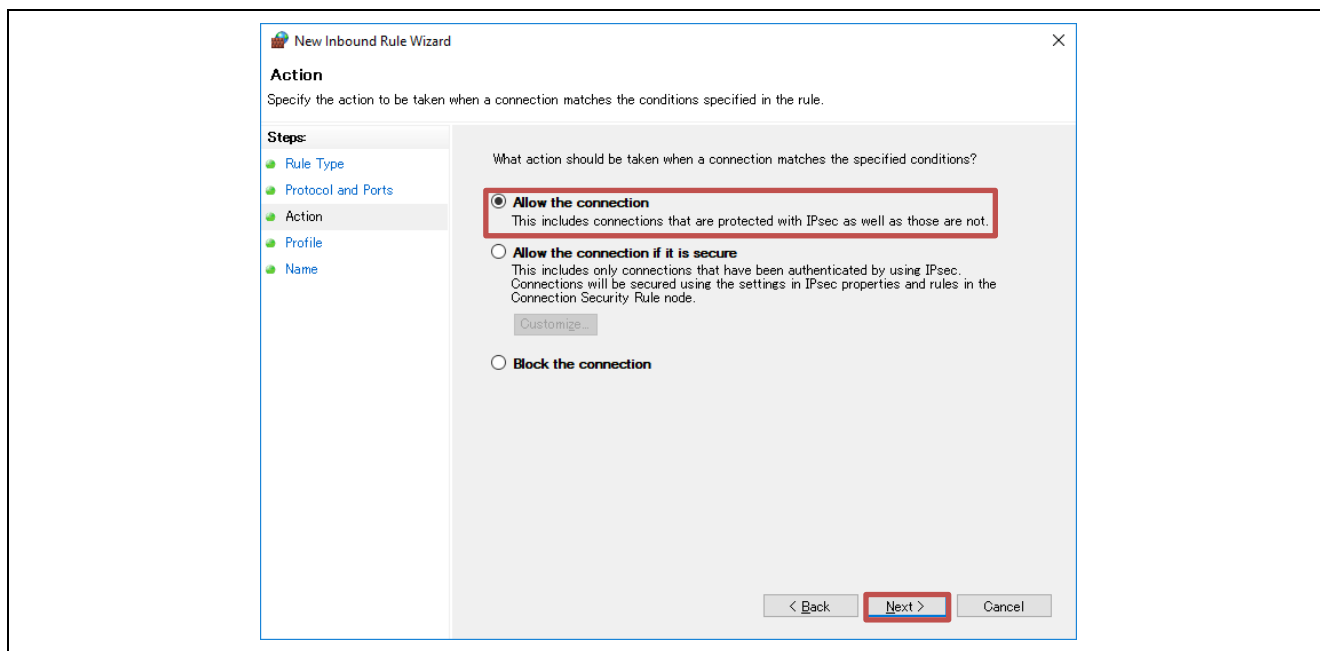
Select “Port”, and press “Next”.



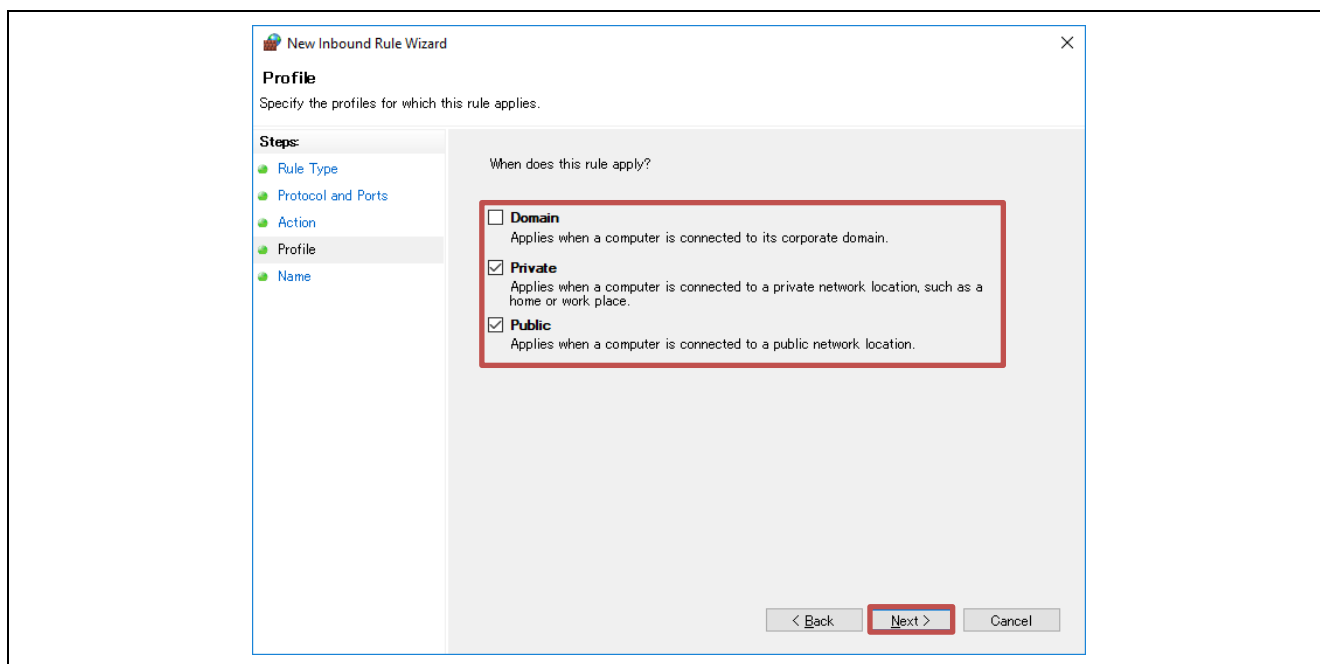
Select “TCP”, and “Specific local ports” and input “7” to text box. Then press “Next”.



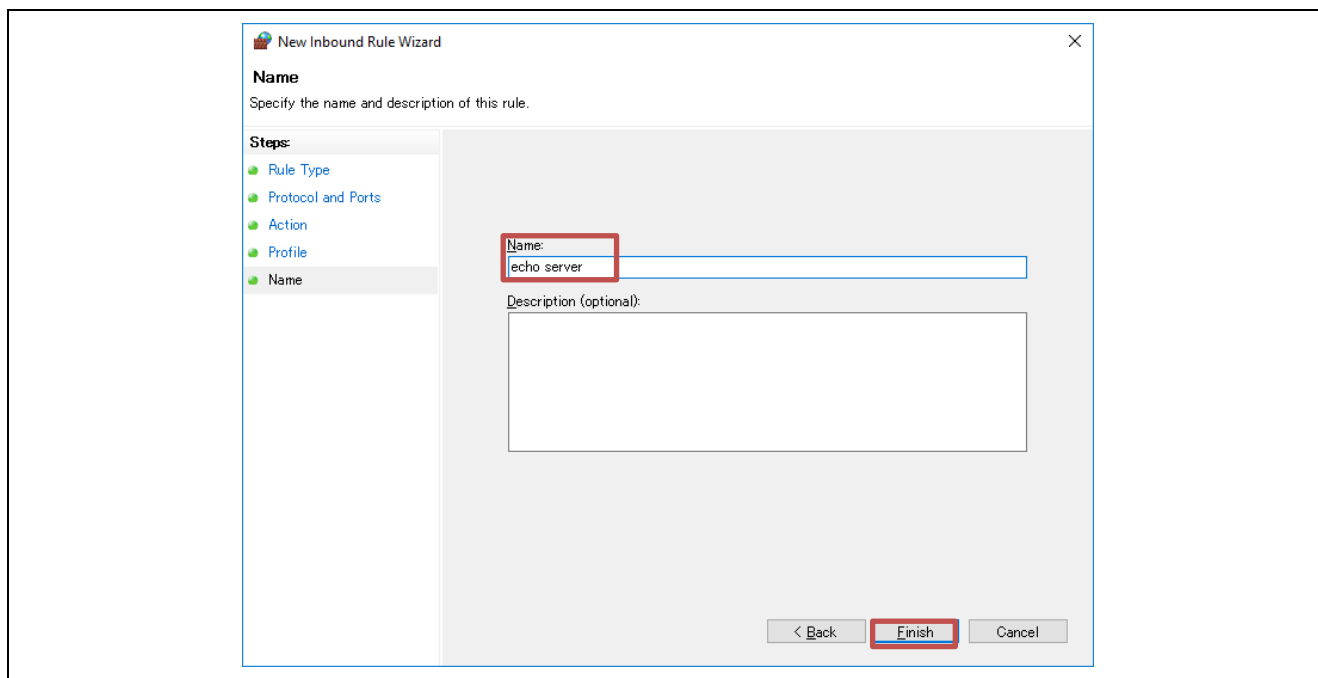
Select “Allow the connection”, and press “Next”.



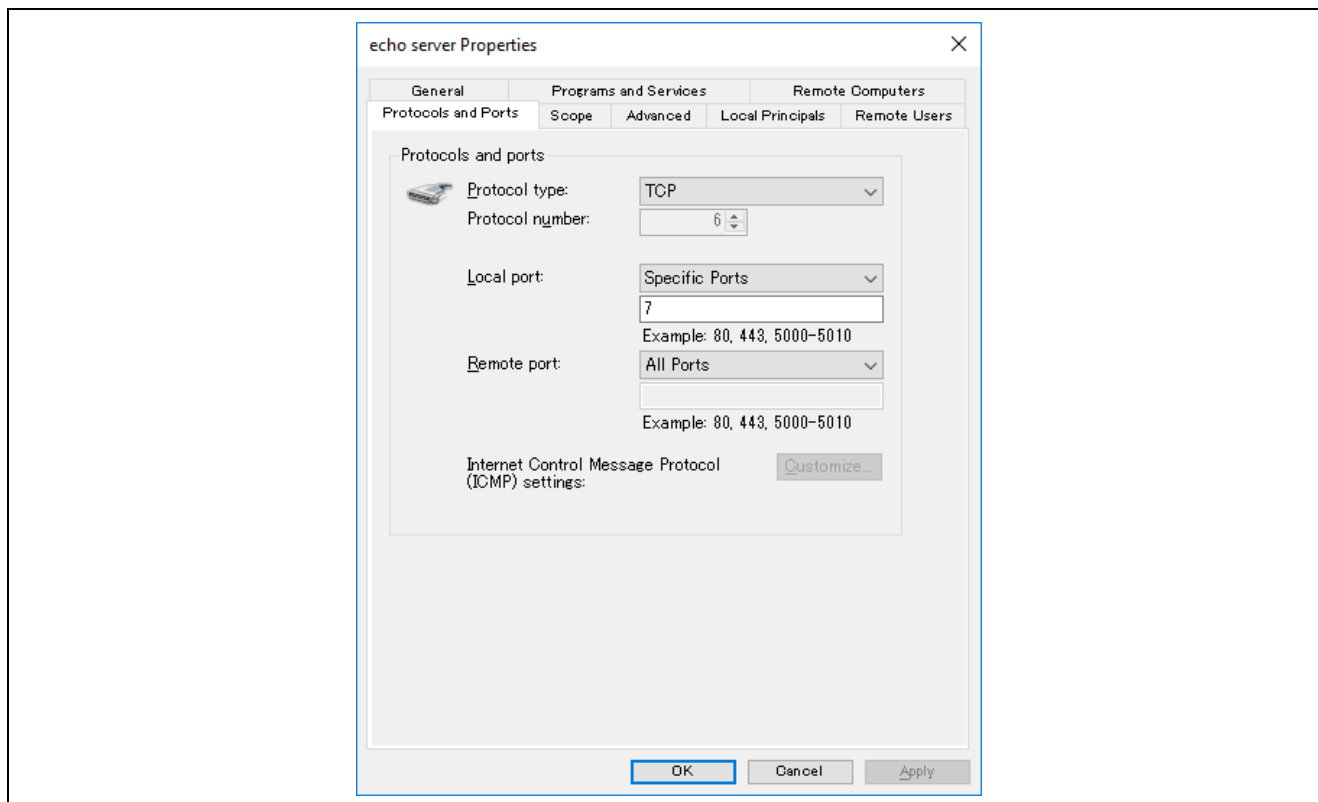
Check the “Private” and “Public”. Then press “Next”.



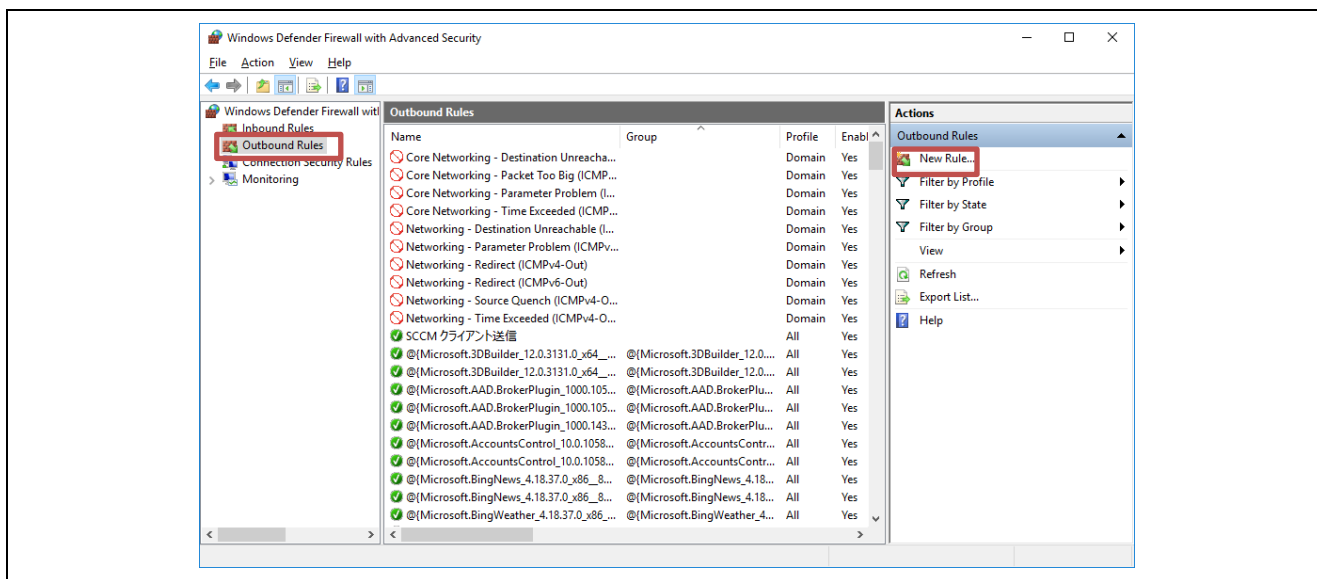
Input a unique name to “Name” input box, and press “Finish”.



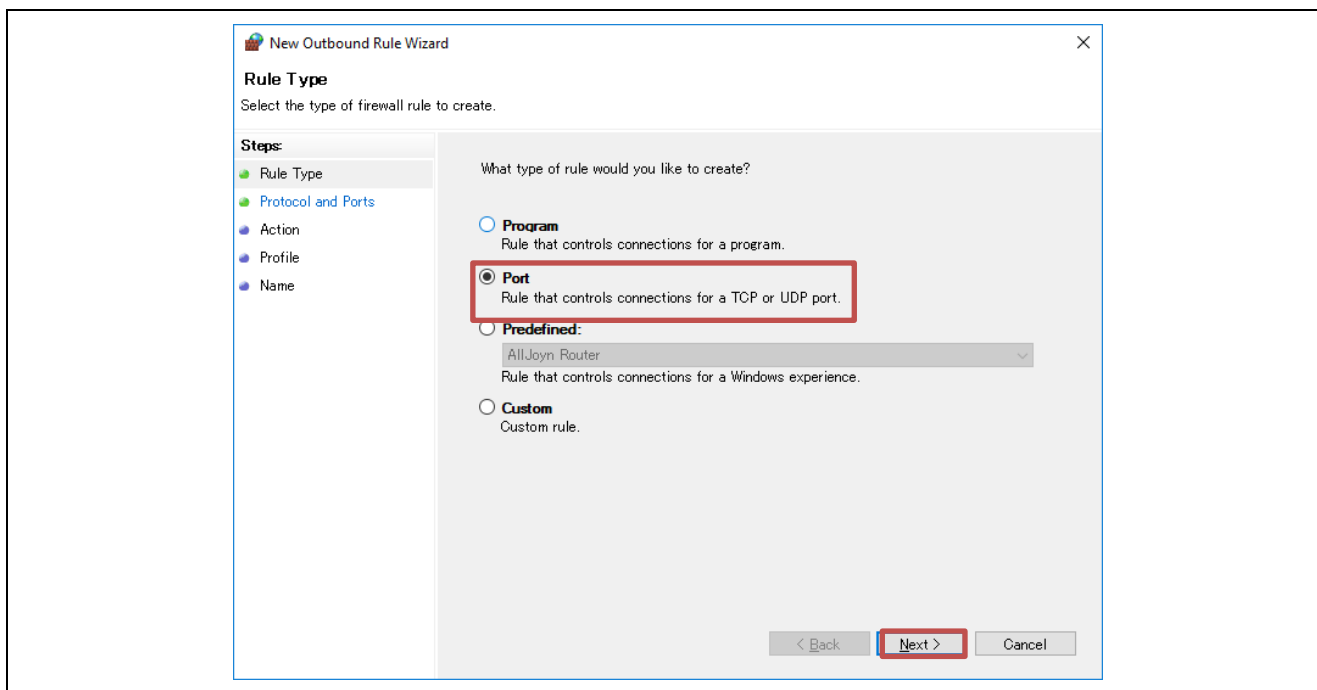
After double-clicking the generated rule, and selecting “Protocols and Ports” tab, you can see following settings.



Select “Outbound Rules”, and press “New Rule...”



Select “Port”, and press “Next”.



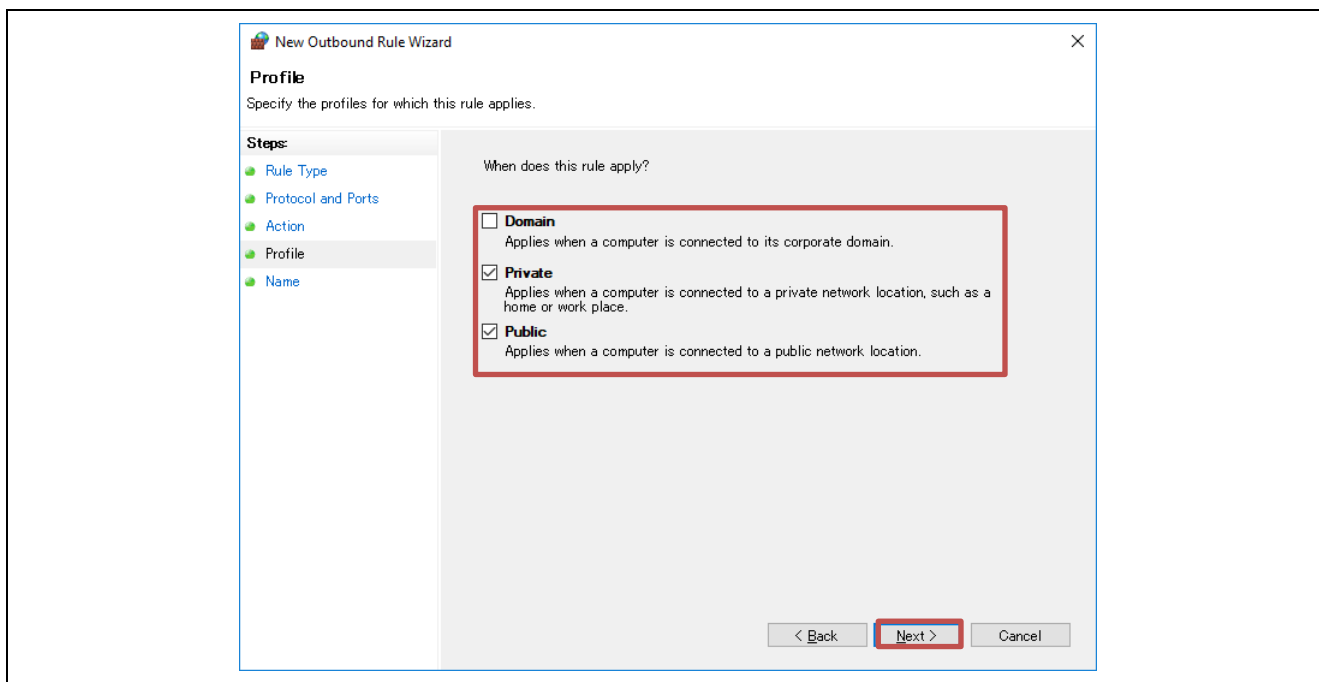
Select “TCP” and “All remote ports”, and press “Next”.

The screenshot shows the 'New Outbound Rule Wizard' dialog box, specifically the 'Protocol and Ports' step. The left sidebar lists the steps: Rule Type, Protocol and Ports (selected), Action, Profile, and Name. The main area contains two questions. The first question is 'Does this rule apply to TCP or UDP?' with radio buttons for 'TCP' (selected) and 'UDP'. The second question is 'Does this rule apply to all remote ports or specific remote ports?' with radio buttons for 'All remote ports' (selected) and 'Specific remote ports:'. Below the 'Specific remote ports' option is a text box with the example '80, 443, 5000-5010'. At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a red box), and 'Cancel'.

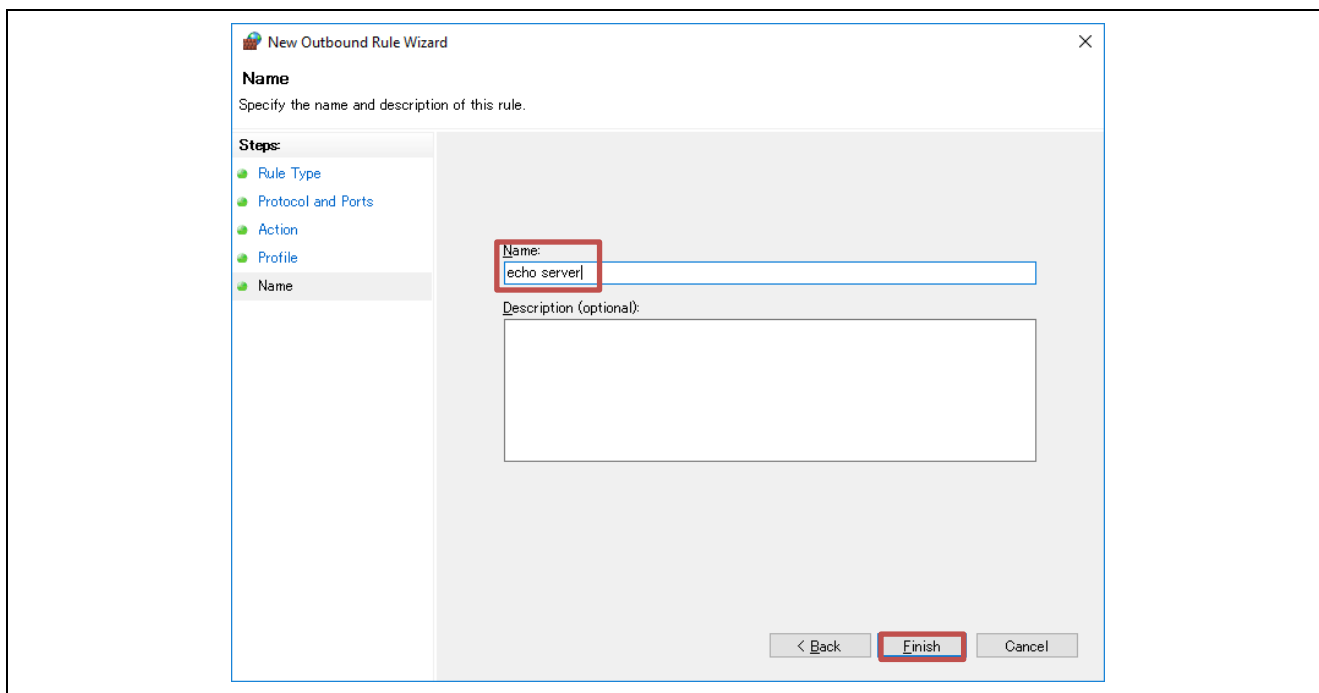
Select “Allow the connection”, and press “Next”.

The screenshot shows the 'New Outbound Rule Wizard' dialog box, specifically the 'Action' step. The left sidebar lists the steps: Rule Type, Protocol and Ports, Action (selected), Profile, and Name. The main area contains the question 'What action should be taken when a connection matches the specified conditions?'. There are three radio button options: 'Allow the connection' (selected), 'Allow the connection if it is secure', and 'Block the connection'. The 'Allow the connection' option has a description: 'This includes connections that are protected with IPsec as well as those are not.' The 'Allow the connection if it is secure' option has a description: 'This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.' Below the 'Allow the connection if it is secure' option is a 'Customize...' button. At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a red box), and 'Cancel'.

Check “Private” and “Public”, and press “Next”.

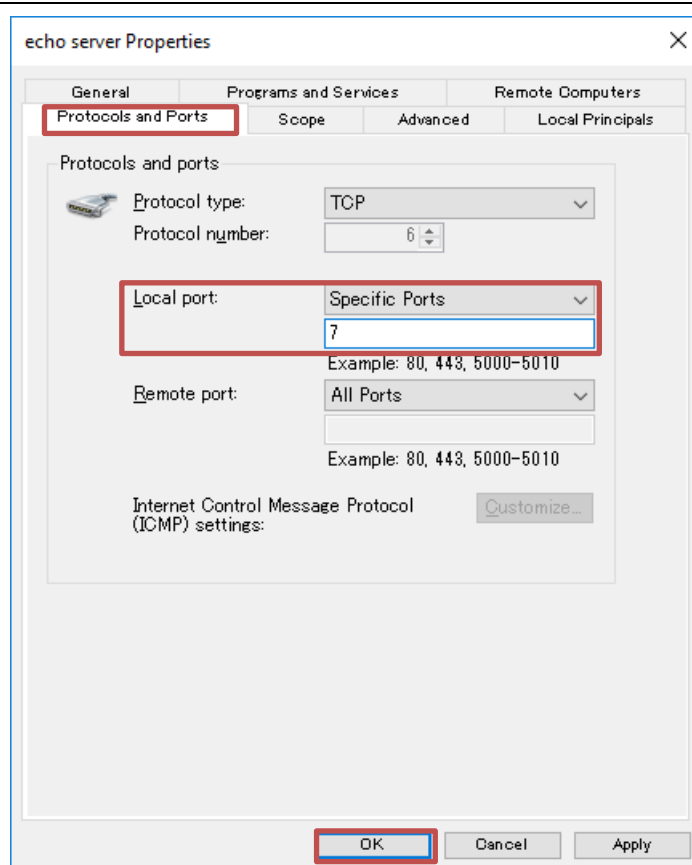


Input a unique name to “Name” input box, and press “Finish”.



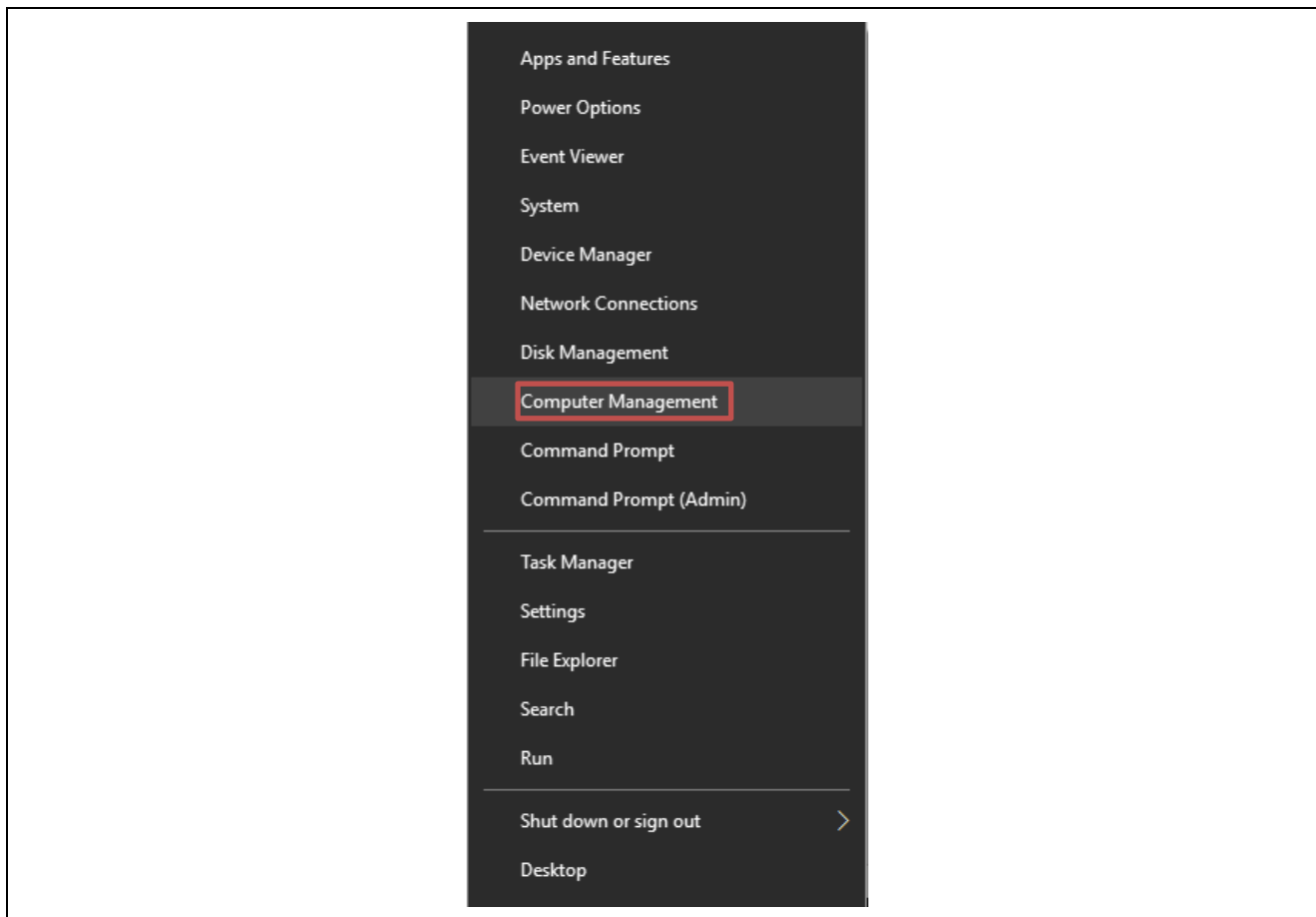
After double-clicking the generated rule, and selecting “Protocols and Ports” tab, you can see following settings.

Set “Local port” to “Specific Ports”, and input “7” to input box. Then press “OK”.



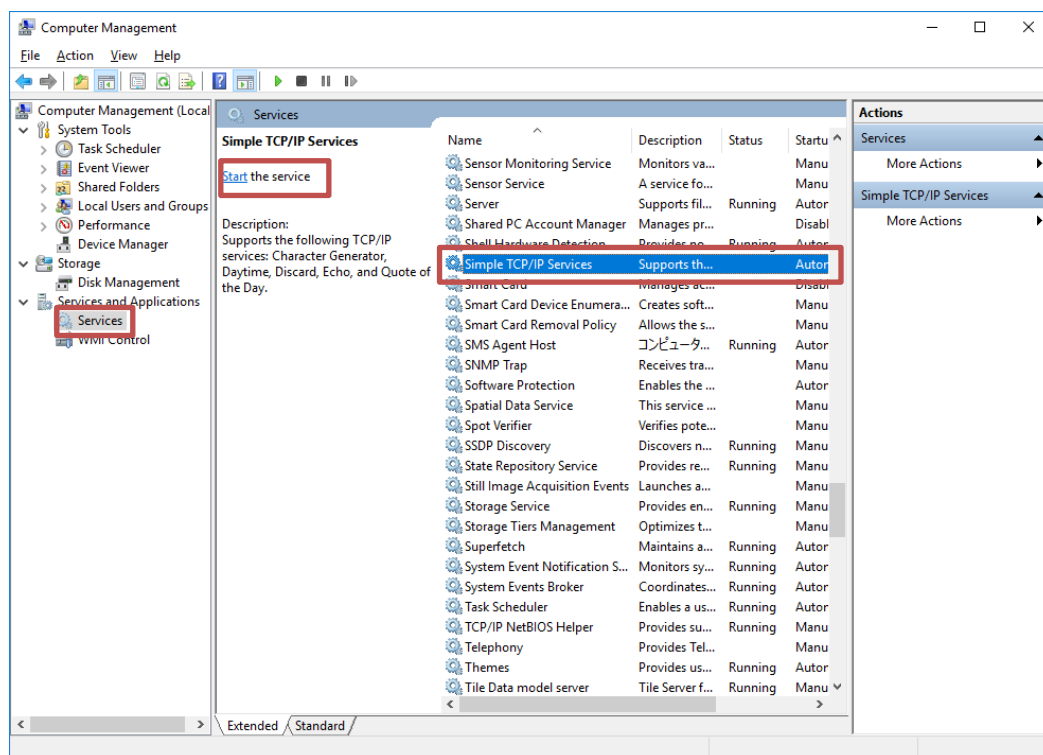
4.1.4 Run “Simple TCPIP services”

Right-click on the Windows start menu, and select “Computer Management”.



Select “Service” in the “Service and Applications”.

Select “Simple TCP/IP Services”, and press “Start the service”.



4.2 HOST PC Settings

Customize the settings of the terminal application in the HOST PC to following:

- Serial baud rate: 115,200bps
- Serial data length: 8bits
- Serial parity bit: not used
- Serial stop bit: 1bit
- Serial flow control: not used
- Line feed code: LF

4.3 Output of the sample program

Download the built sample program to RZ/A2M Evaluation Board Kit, and run the program. Then following log messages will be appeared in the terminal application: This sample program compares the messages transmitted to the echo server and the messages received from the echo server.

```
RZ/A2M network_sample for GCC Ver. 1.0
Copyright (C) 2018 Renesas Electronics Corporation. All rights reserved.
Build Info Date Dec 21 2018 at 15:01:32
FreeRTOS OS Abstraction Version 3.0
Connecting to echo server
Echo demo failed to connect to echo server 192.168.128.1.
Connecting to echo server
Connected to echo server
Sending TxRx message number 0 of length 1 to echo server
Received correct string from echo server.
Sending TxRx message number 1 of length 2 to echo server
Received correct string from echo server.
Sending TxRx message number 2 of length 3 to echo server
Received correct string from echo server.
Sending TxRx message number 3 of length 4 to echo server
Received correct string from echo server.
Sending TxRx message number 4 of length 5 to echo server
Received correct string from echo server.
Sending TxRx message number 5 of length 6 to echo server
Received correct string from echo server.
Sending TxRx message number 6 of length 7 to echo server
Received correct string from echo server.
Sending TxRx message number 7 of length 8 to echo server
Received correct string from echo server.
Sending TxRx message number 8 of length 9 to echo server
Received correct string from echo server.
Sending TxRx message number 9 of length 10 to echo server
Received correct string from echo server.
Shutting down connection to echo server.
Connecting to echo server
Connected to echo server
Sending TxRx message number 10 of length 11 to echo server
Received correct string from echo server.
:
:
:
```

Failed to connect with an echo server.

Succeed to connect with an echo server.

Succeed to transmit and receive message 0-9.

Connection is shut down after 10 messages of TxRx.

Continues re-connection every 10 messages. Message 10 or later will be used.

5. Connecting Amazon Web Service

In this section, the example of connecting Amazon Web Service (AWS) with MQTTEcho demonstration provided by Amazon.

5.1 AWS setting

To create AWS account, you need to access following URL by web browser.

<https://portal.aws.amazon.com/billing/signup#/start>

After creating AWS account, you can create things by AWS IoT Core service.

To create things, follow the steps below.

5.1.1 Open AWS IoT Core Service

1. Sign-in AWS console, and input *IoT* on the search bar of AWS Service.
2. Click IoT Core.

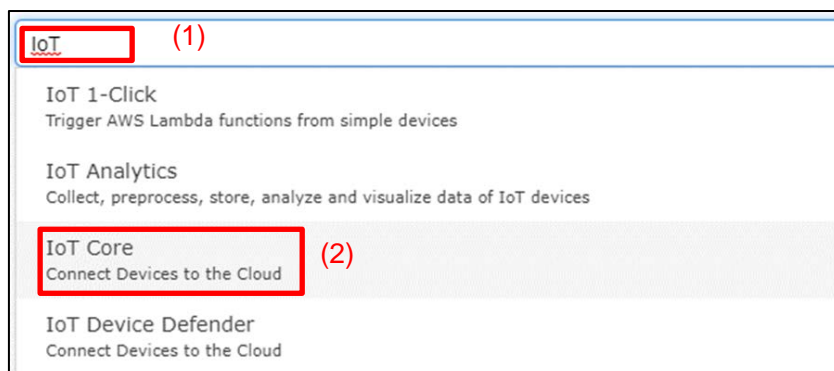


Figure 5-1 Select IoT Core Service

5.1.2 Creating AWS IoT things

1. Press **Manage** at the left of AWS IoT.
2. Select **Things**.
3. On the You don't have any things yet page, choose **Register a thing**.

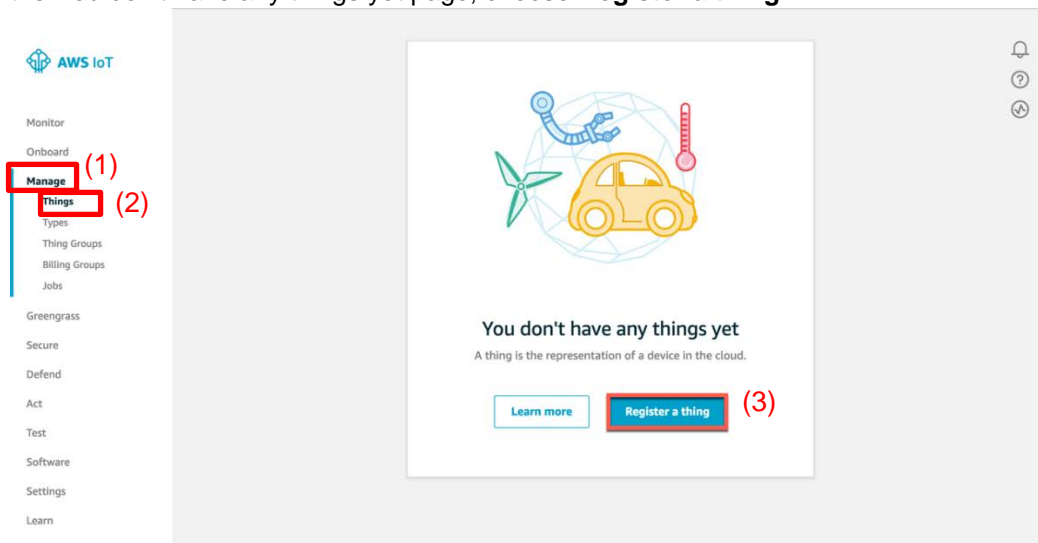


Figure 5-2 the You don't have any things yet page

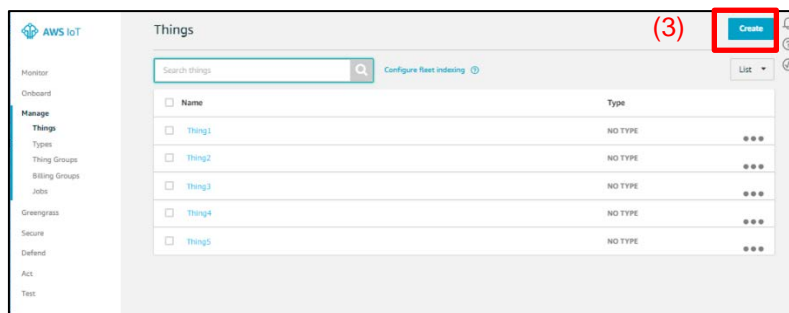


Figure 5-3 List of things page (any things exist)

4. On the **Creating AWS IoT things** page, choose **Create a single thing**.

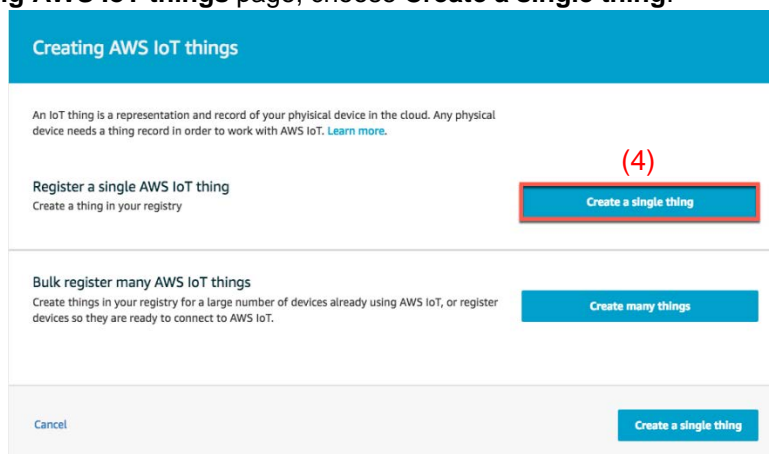


Figure 5-4 the Creating AWS IoT things page

5. On the **Create a thing** page, in the **Name** field, type a name for your thing, such as `rz_test`.
6. Choose **Next**.

CREATE A THING

STEP 1/3

Add your device to the thing registry

This step creates an entry in the thing registry and a thing shadow for your device.

Name

rz_test (5)

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected Create a type

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups / Create group Change

Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key	Value
Provide an attribute key, e.g. Manufacturer	Provide an attribute value, e.g. Acme-Corporation

Add another

Clear

Show thing shadow ▼ (6)

Cancel Back Next

Figure 5-5 Create a thing page

7. On the **Add a certificate for your thing** page, choose **Create certificate**.

CREATE A THING

Add a certificate for your thing

STEP 2/3

A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

Create certificate (7)

Create with CSR
Upload your own certificate signing request (CSR) based on a private key you own.

Create with CSR

Use my certificate
Register your CA certificate and use your own certificates for one or many devices.

Get started

Skip certificate and create thing
You will need to add a certificate to your thing later before your device can connect to AWS IoT.

Create thing without certificate

Figure 5-6 Add a certificate for your thing page

8. Choose **Download** for your certificate.
9. Choose **Download** for the Amazon root CA.
10. Choose **Download** for your private key.
11. Choose **Activate**.
12. choose **Attach a policy**.

Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	0488f55a11.cert.pem	Download (8)
A public key	0488f55a11.public.key	Download (9)
A private key	0488f55a11.private.key	Download (10)

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT [Download](#)

Activate (11)

Cancel Done **Attach a policy** (12)

Figure 5-7 Certificate created! page

13. On the **Add a policy for your thing** page, choose **Register Thing**.

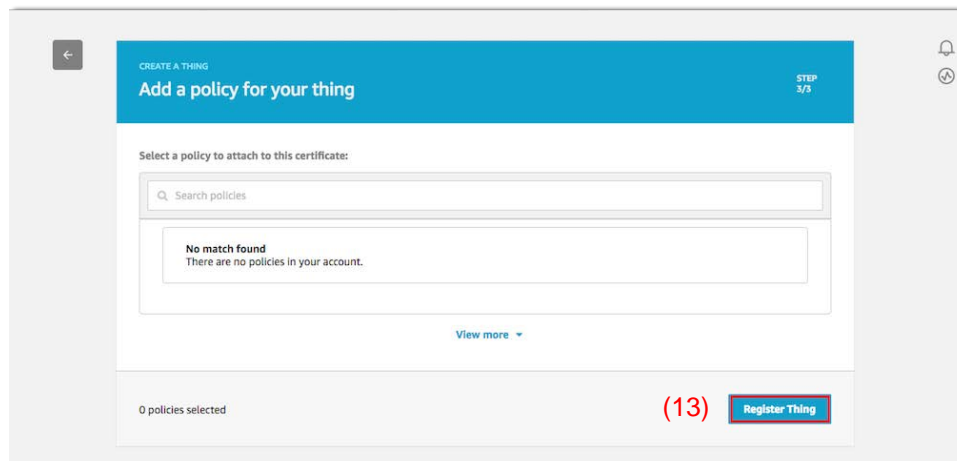


Figure 5-8 Add a policy for your thing page

5.1.3 Create AWS IoT policy

1. On the AWS IoT console, in the navigation pane, choose **Secure**.
2. Choose **Policies**.
3. Choose **Create a policy**.

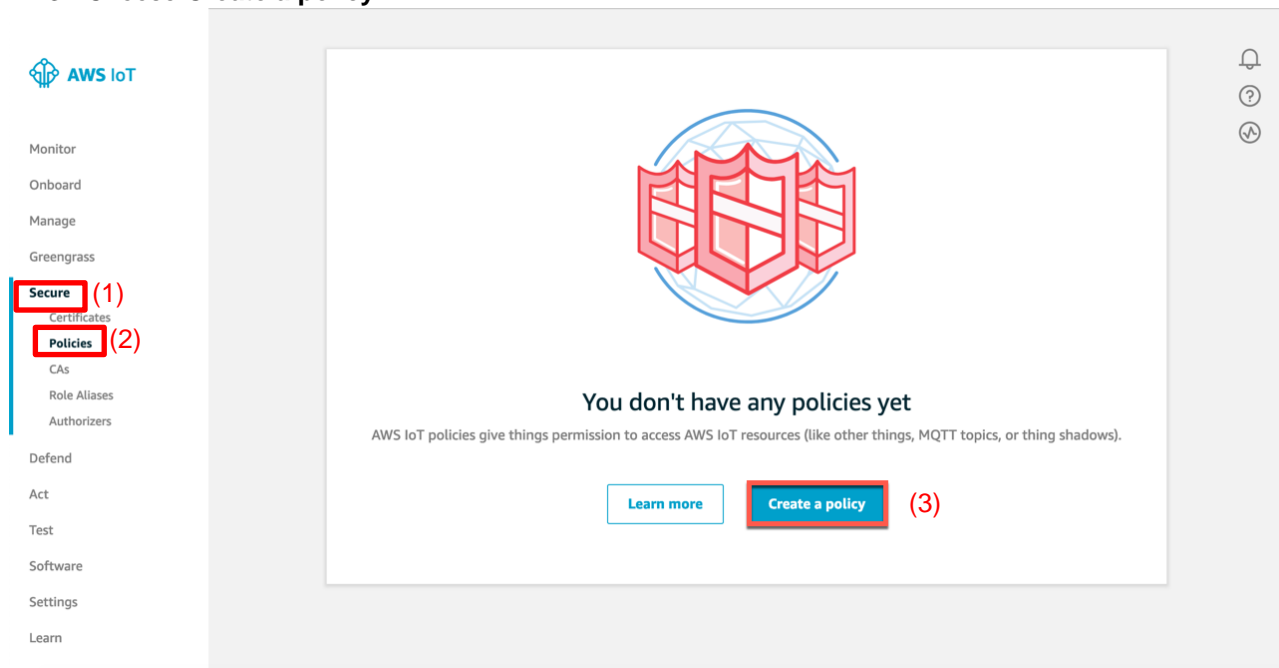


Figure 5-9 You don't have any policies yet page

4. Enter a **Name** for the policy, such as `rz_echo_test`.

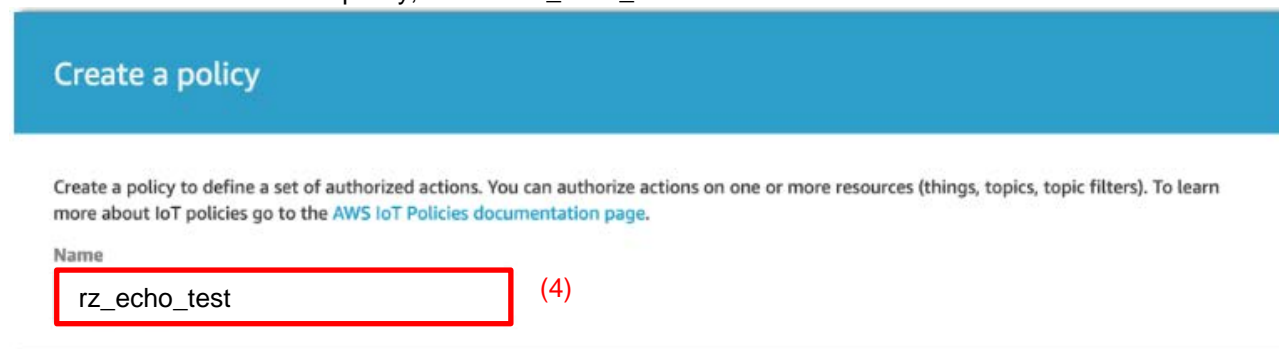


Figure 5-10 Create a policy - Name

5. For **Action**, enter `iot:Connect`.
6. For **Resource ARN**, replace `replaceWithAClientId` to the string defined in `#define echoCLIENT_ID` described in `<root>/demos/common/mqtt/aws_hello_world.c`.
7. Under **Effect**, choose **Allow**.

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action

`iot:Connect` (5)

Resource ARN

`arn:aws:iot:us-east-1: xxxxxxxxxx : client/replaceWithAClientId`

Effect

☒ Allow ☐ Deny (7)

Remove

Add statement

```
aws_hello_world.c
77 #include "aws_demo_config.h"
78 #include "aws_hello_world.h"
79
81 * @brief MQTT client ID.
85 #define echoCLIENT_ID ( ( const uint8_t * "MQTTEcho" ) )
86
```

Figure 5-11 Create a policy – Add statement (before replacement)

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action

`iot:Connect` (5)

Resource ARN

`arn:aws:iot:us-east-1: xxxxxxxxxx : client/MQTTEcho` (6)

Effect

☒ Allow ☐ Deny (7)

Remove

Add statement

Figure 5-12 Create a policy – Add statement (after replacement)

8. Choose **Add Statement**.

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name

Add statements

Policy statements define the types of actions that can be performed by a resource. Advanced mode

Action

Resource ARN

Effect

☒ Allow ☐ Deny

Remove

Add statement

 (8)

Figure 5-13 Create a policy – Add statement

9. For **Action**, enter `iot:Publish`
10. For **Resource ARN**, replace **replaceWithATopic** to the string defined in `#define echoTOPIC_NAME` described in `<root>/demos/common/mqtt/aws_hello_world.c`.
11. Under **Effect**, choose **Allow**.

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action

iot:Publish (9)

Resource ARN

arn:aws:iot:us-east-1: xxxxxxxxxx : topic/freertos/demos/echo

Effect

☒ Allow ☐ Deny (11)

Remove

Add statement

```
86
88
89  * @brief The topic that the MQTT client both subscribes and publishes to.
90  #define echoTOPIC_NAME    ( ( const uint8_t * ) freertos/demos/echo )
91
```

Figure 5-14 Create a policy – Add statement

12. Add **iot:Subscribe**, **iot:Receive** statements by following step 8 to step11.

13. Choose **Create**.

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action	<input type="text" value="iot:Subscribe"/> (12)
Resource ARN	<input type="text" value="arn:aws:iot:us-east-1: xxxxxxxxx : topic/freertos/demos/echo"/>
Effect	<input checked="" type="checkbox"/> Allow <input type="checkbox"/> Deny Remove

Add statement

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action	<input type="text" value="iot:Receive"/> (12)
Resource ARN	<input type="text" value="arn:aws:iot:us-east-1: xxxxxxxxx : topic/freertos/demos/echo"/>
Effect	<input checked="" type="checkbox"/> Allow <input type="checkbox"/> Deny Remove

Add statement

Figure 5-15 Create a policy – Add statements

(13)	<input type="button" value="Create"/>
-------------------	---------------------------------------

Figure 5-16 Create a policy – Create

5.1.4 Attach an AWS IoT Policy to a Device Certificate

1. In the left navigation pane, choose **Secure**.
2. Choose **Certificates**.
3. In the box for the certificate you created, choose ... to open a drop-down menu.
4. Choose **Attach policy**.

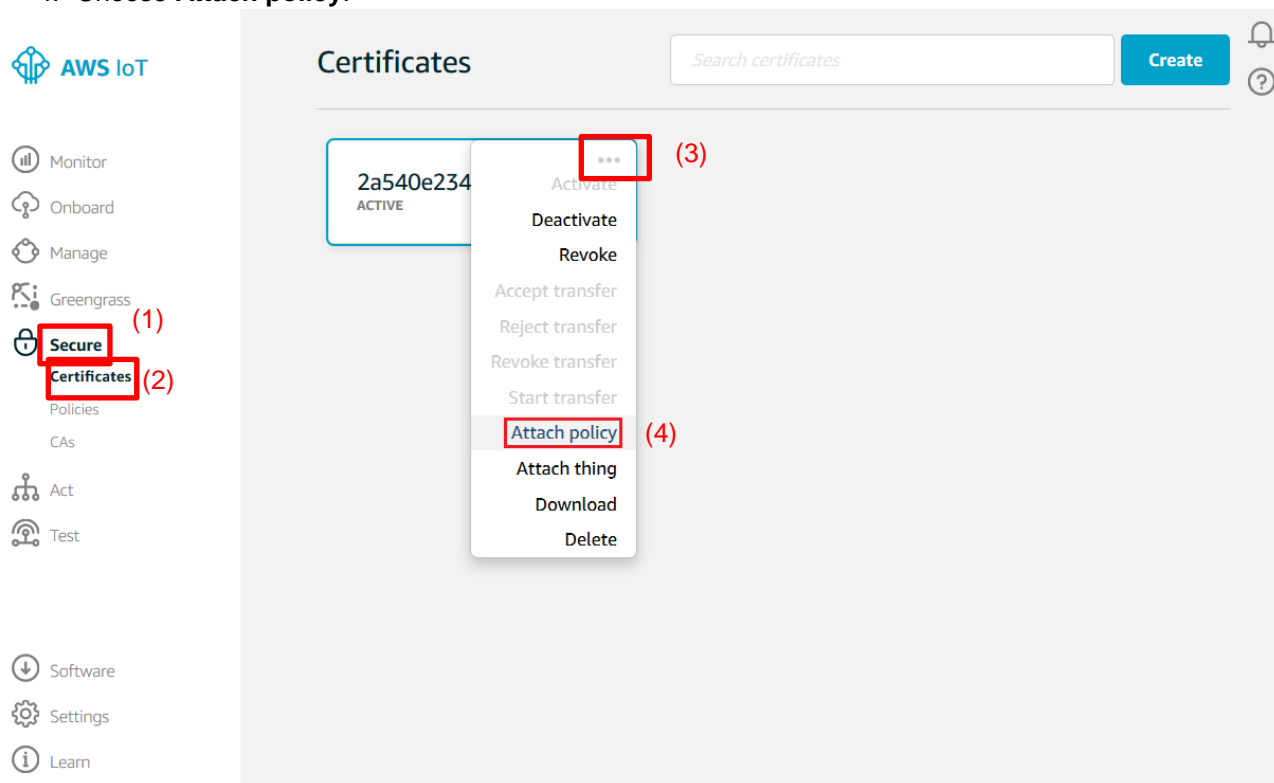


Figure 5-17 Certificates

5. In the Attach policies to certificate(s) dialog box, select the check box next to the policy you created in the previous step
6. Choose Attach.



Figure 5-18 Attach policies to Certificates(s)

5.2 Certification settings

Follow this section to set AWS IoT information of Amazon FreeRTOS sample program for RZ/A2M.

5.2.1 Setting AWS IoT Information

1. In the left navigation pane, choose **Settings**.
2. Confirm **Endpoint**.
3. Copy endpoint to `clientcredentialMQTT_BROKER_ENDPOINT[]` described in `<root>/demos/common/include/aws_clientcredential.h`.

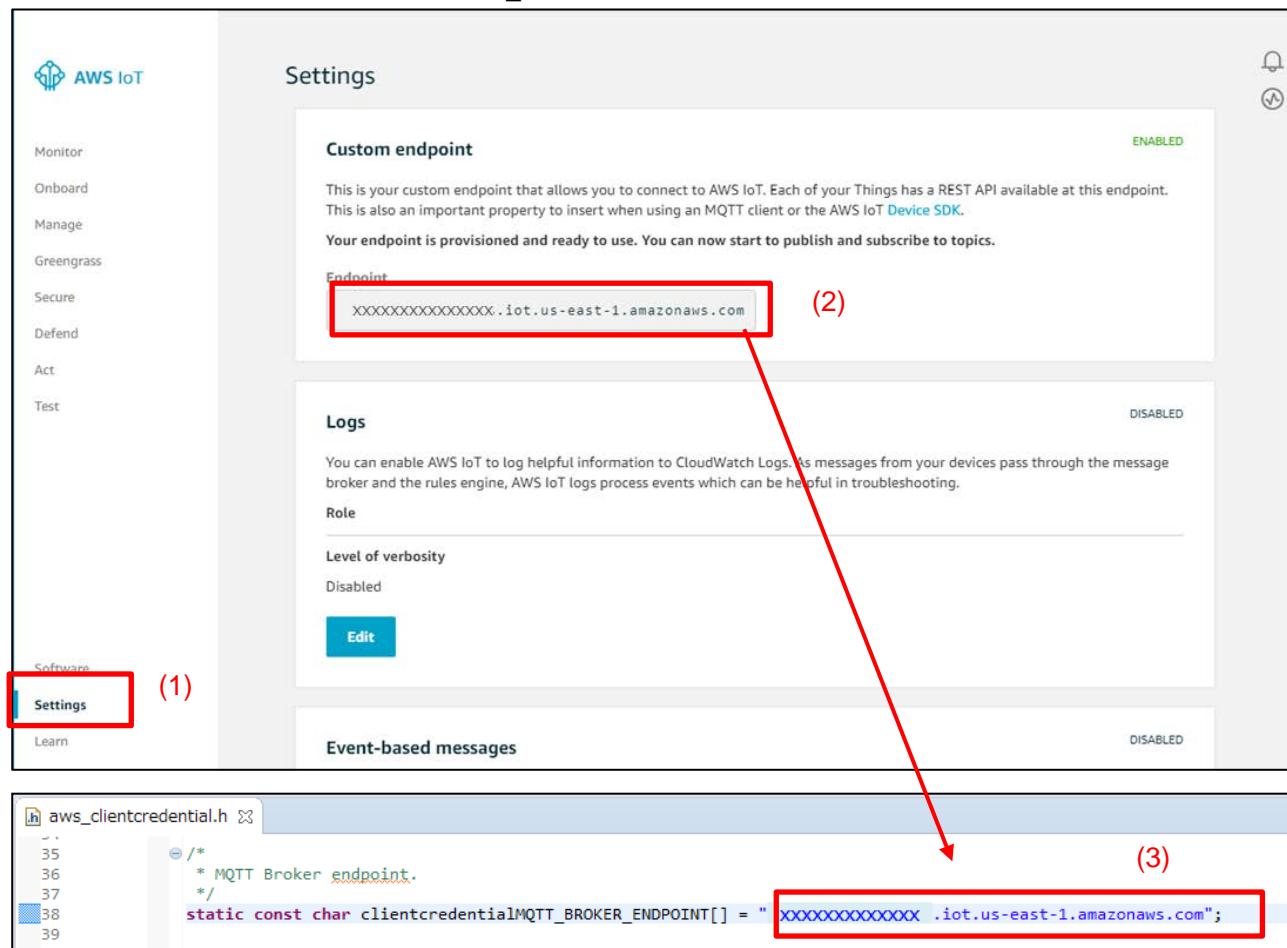


Figure 5-19 Setting Endpoint

4. In the left navigation pane, choose **Manage**.
5. Select **Things**.
6. Confirm the name of Things.
7. Copy the name of Things to **clientcredentialIOT_THING_NAME** described in `<root>/demos/common/include/aws_clientcredential.h`.

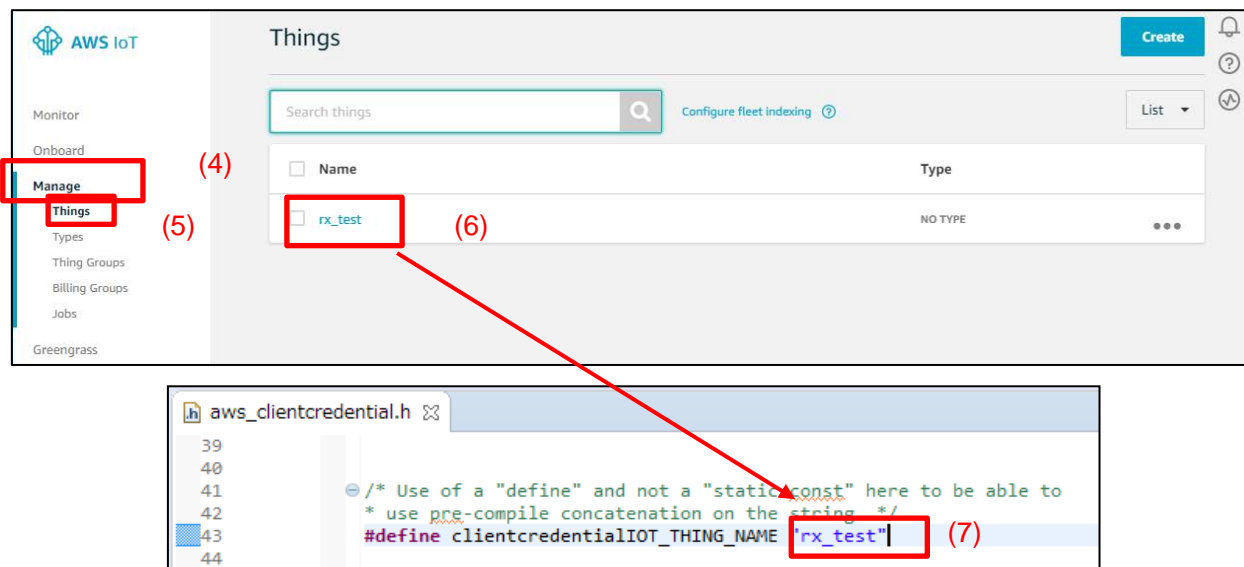


Figure 5-20 Setting the name of Things

5.2.2 Setting certificate of AWS IoT

1. Copy the content of **A certificate for this thing** downloaded in section 5.1.2 to **clientcredentialCLIENT_CERTIFICATE_PEM[]** described in `<root>/demos/common/include/aws_clientcredential_keys.h`. Add `"` to the top of each line, and add `;\n"` to the end of each line.

Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	09eb9ae91d.cert.pem	Download
A public key	09eb9ae91d.public.key	Download
A private key	09eb9ae91d.private.key	Download

(1) confirm the file downloaded from here.

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

Activate

Cancel

Done

Attach a policy

[illegible]

Figure 5-21 Setting a certificate information

- Copy the content of **A privatekey** downloaded in section 5.1.2 to **keyCLIENT_PRIVATE_KEY_PEM[]** described in `<root>/demos/common/include/aws_clientcredential_keys.h`.
Add `"` to the top of each line, and add `;\n\"` to the end of each line.

Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	09eb9ae91d.cert.pem	Download
A public key	09eb9ae91d.public.key	Download
A private key	09eb9ae91d.private.key	Download

(2) confirm the file downloaded from here.

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

[Activate](#)

[Cancel](#)

[Done](#)

[Attach a policy](#)

```
aws_clientcredential_keys.h
55
56
57 /*
58  * PEM-encoded client private key.
59  * Must include the PEM header and footer:
60  * "-----BEGIN RSA PRIVATE KEY-----\n"
61  * "...base64 data...\n"
62  * "-----END RSA PRIVATE KEY-----\n"
63  */
64 #define keyCLIENT_PRIVATE_KEY_PEM "-----BEGIN RSA PRIVATE KEY-----\n"
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
data of cc1xxxxxxx-private.pem.key file
-----END RSA PRIVATE KEY-----\n"
```

Figure 5-22 Setting a private key information

5.3 Setting RZ/A2M project

5.3.1 Turn off debugging message

Change the value of #define ipconfigHAS_DEBUG_PRINTF described in *FreeRTOSIPConfig.h* to 0 from 1.

#define ipconfigHAS_DEBUG_PRINTF	0
----------------------------------	---

5.3.2 Set the MAC address

Set the values of #define configMAC_ADDR0, ..., and #define configMAC_ADDR5 described in *FreeRTOSConfig.h* with the MAC address of the board.

#define configMAC_ADDR0	0x74
#define configMAC_ADDR1	0x90
#define configMAC_ADDR2	0x50
#define configMAC_ADDR3	0x00
#define configMAC_ADDR4	0x79
#define configMAC_ADDR5	0x03

5.3.3 Setting the IP address

- In the case to use DHCP server

Change the value of #define ipconfigUSE_DHCP described in *FreeRTOSIPConfig.h* to 1 from 0.

#define ipconfigUSE_DHCP	1
--------------------------	---

- In the case not to use DHCP server

Confirm that the value of #define ipconfigUSE_DHCP described in *FreeRTOSIPConfig.h* is 0.

#define ipconfigUSE_DHCP	0
--------------------------	---

Set the values of #define configIP_ADDR0, ..., and #define configIP_ADDR3 described in *FreeRTOSConfig.h* with the IP address to set.

#define configIP_ADDR0	192
#define configIP_ADDR1	168
#define configIP_ADDR2	128
#define configIP_ADDR3	2

Set the values of #define configGATEWAY_ADDR0, ..., and #define configGATEWAY_ADDR3 described in *FreeRTOSConfig.h* with the IP address of a Gateway server to use.

#define configGATEWAY_ADDR0	192
#define configGATEWAY_ADDR1	168
#define configGATEWAY_ADDR2	128
#define configGATEWAY_ADDR3	100

Set the values of #define configDNS_ADDR0, ..., and #define configDNS_ADDR3 described in *FreeRTOSConfig.h* with the IP address of a DNS server to use.

#define configDNS_SERVER_ADDR0	192
#define configDNS_SERVER_ADDR1	168
#define configDNS_SERVER_ADDR2	128
#define configDNS_SERVER_ADDR3	200

Set the values of #define configNET_MASK0, ..., and #define configNET_MASK3 described in *FreeRTOSConfig.h* with the subnet mask to use.

#define configNET_MASK0	255
#define configNET_MASK1	255
#define configNET_MASK2	255
#define configNET_MASK3	0

5.3.4 Switch the demonstration program to use.

Modify `demos\common\demo_runner\aws_demo_runner.c` to enable `vStartMQTTEchoDemo()` and disable `vStartTCPEchoClientTasks_SingleTasks()`.

```
/* Demo declarations. */
/* extern void vStartDeviceDefenderDemo( void ); */
/* extern void vStartGreenGrassDiscoveryTask( void ); */
extern void vStartMQTTEchoDemo( void );
/* extern void vStartOTAUpdateDemoTask( void ); */
/* extern void vStartShadowDemoTasks( void ); */
/* extern void vStartSimpleTCPServerTasks( void ); */
/* extern void vStartSubpubDemoTasks( void ); */
/* extern void vStartTCPEchoClientTasks_SeparateTasks( void ); */
/* extern void vStartTCPEchoClientTasks_SingleTasks( void ); */

/*-----*/

/**
 * @brief Runs demos in the system.
 */
void DEMO_RUNNER_RunDemos( void )
{
    /* vStartDeviceDefenderDemo(); */
    /* vStartGreenGrassDiscoveryTask(); */
    vStartMQTTEchoDemo();
    /* vStartOTAUpdateDemoTask(); */
    /* vStartShadowDemoTasks(); */
    /* vStartSimpleTCPServerTasks(); */
    /* vStartSubpubDemoTasks(); */
    /* vStartTCPEchoClientTasks_SeparateTasks(); */
    /* vStartTCPEchoClientTasks_SingleTasks(); */
}
```

5.3.5 Import, build, and download

Import the project to e2studio, and build the project, and download the generated file to the board.

5.4 Running MQTTEcho demonstration

5.4.1 Setting for AWS IoT test

1. Sign in AWS console, and open AWS IoT.
2. In the left navigation pane, choose **Test**.
3. Copy the string defined as `#define echoTOPIC_NAME` described in `<root>/demos/common/mqtt/aws_hello_world.c` to the **Subscription topic**.
4. Choose **MQTT payload display** to **Display payloads as strings (more accurate)**.
5. Choose **Subscribe to topic**.

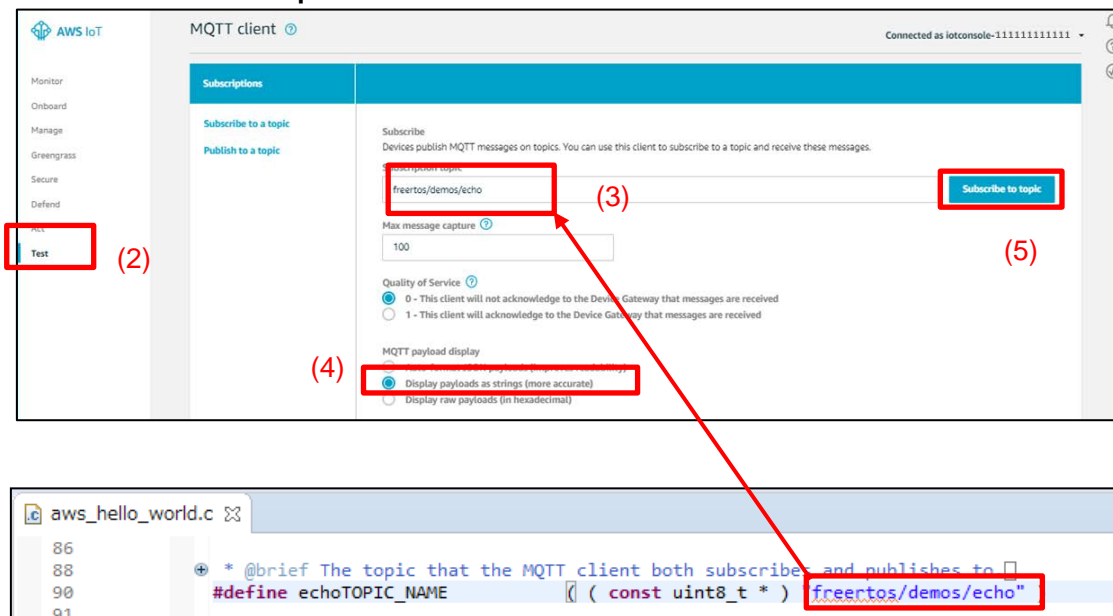


Figure 5-23 MQTTEcho settings

6. Run MQTTEcho demonstration program on RZ/A2M.
7. Confirm MQTT messages

The screenshot displays the MQTT Echo interface. On the left, under 'Subscriptions', the topic 'freertos/demos/echo' is listed. The main panel shows a 'Publish' section with a text input containing 'freertos/demos/echo' and a 'Publish to topic' button. Below this, a code editor shows a JSON message:

```
{
  "message": "Hello from AWS IoT console"
}
```

. A red box highlights a list of received messages. The first message is 'freertos/demos/echo' received on May 14, 2019, at 8:29:35 PM +0900, with the payload 'Hello World 3 ACK'. Subsequent messages are 'freertos/demos/echo' with payloads 'Hello World 3', 'Hello World 2 ACK', 'Hello World 2', and 'freertos/demos/echo'.

Topic	Time	Message	Export	Hide
freertos/demos/echo	May 14, 2019 8:29:35 PM +0900	Hello World 3 ACK	Export	Hide
freertos/demos/echo	May 14, 2019 8:29:35 PM +0900	Hello World 3	Export	Hide
freertos/demos/echo	May 14, 2019 8:29:31 PM +0900	Hello World 2 ACK	Export	Hide
freertos/demos/echo	May 14, 2019 8:29:31 PM +0900	Hello World 2	Export	Hide
freertos/demos/echo	May 14, 2019 8:29:27 PM +0900	freertos/demos/echo	Export	Hide

Figure 5-24 A result of the MQTTEcho demonstration.

6. Reference Documents

User's Manual: Hardware

RZ/A2M Group User's Manual: Hardware

The latest version can be downloaded from the Renesas Electronics website.

RTK7921053C00000BE (RZ/A2M CPU board) User's Manual

The latest version can be downloaded from the Renesas Electronics website.

RTK79210XXB00000BE (RZ/A2M SUB board) User's Manual

The latest version can be downloaded from the Renesas Electronics website.

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

The latest version can be downloaded from the ARM website.

ARM Cortex™-A9 Technical Reference Manual Revision: r4p1

The latest version can be downloaded from the ARM website.

ARM Generic Interrupt Controller Architecture Specification - Architecture version 2.0

The latest version can be downloaded from the ARM website.

ARM CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3

The latest version can be downloaded from the ARM website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User's Manual: Development Tools

Integrated development environment e2studio User's Manual can be downloaded from the Renesas Electronics website.

The latest version can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec 28, 2018	-	First edition issued
1.01	Apr 15, 2019	4	Switched confirmation version of e ² studio to 7.4.0
1.10	May 17, 2019	3	Table 2-1 Peripheral device used (1/2) Remove compiler option "-mthumb-interwork"
		25	Added section 5, Connecting Amazon Web Service.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/