RZ/A2M Group

RZ/A2M Azure RTOS Package for GR-MANGO Quick Start Guide

1. Introduction

This is the Quick Start Guide for the RZ/A2M Software Package for GR-MANGO which works on GR-MANGO board by CORE CORPORATION and the operation of Renesas e^2 studio.

This document describes the procedure for executing the sample projects that are available on the following Web site.

https://github.com/renesas-rz/rza2_gcc_azure_rtos_bsp

2. Preparation

2.1 Tool

RZ/A2M Software Package for GR-MANGO can be used on the following environment. Please check your environment before continuing.

Tools:

- IDE: e^2 studio 2021-04 Windows 64-bit product version or later
- Tool Chain: GNU ARM Embedded Toolchain 6-2017-q2-update

This tool in bundled in the IDE above. Available from
https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads/6-2017-q2-update

Refer to the following document for the details of installing e^2 studio.

e^2 studio Integrated Development Environment User's Manual: Getting Started

Target Board:

GR-MANGO

ICE (In-circuit emulator):

On GR-MANGO environment, user do not need to prepare an ICE for programming and debugging.

GR-MANGO supports Arm Mbed DAPLink feature for rapid prototyping. DAPLink provides features below.

drag-and-drop programming (MSC)
a virtual serial port (CDC)
CMSIS-DAP based debugging (HID)

For more detail of DAPLink specification, please refer following page.

https://os.mbed.com/handbook/DAPLink

Bootloader:

This package includes the bootloader as table data. If user would like to get the source code, please access following website.

https://www.mxic.com.tw/en-us/support/technical-documentation/Pages/Serial-NOR-Flash.aspx

2.2 Virtual Serial Port Connection

Connect CN1 on the GR-MANGO board to a Windows™ PC, this provides a USB virtual serial port.

When the RZ/A2M SUB board is first connected, the PC will look for a suitable driver. This driver is installed during the installation process and the PC should automatically find and install it. The PC will report it is installing a driver and report a driver has been installed successfully. The COMx port number allocated to the virtual serial port can be confirmed in Windows™ Device Manager.

2.3 Serial Terminal

Start a serial terminal program (such as PuTTY, HyperTerminal or Tera Term) using the following configuration:

Baud Rate: 115200
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None
COM Port: As shown in Windows™ Device Manager.

3. Trying sample application

3.1 Importing Software Package into IDE

This package is distributed as an archive file. Build project of this package can be imported into e² studio from the Project Import Menu. User can import the project to e² studio by the following procedure in this section.

— Obtain the package to use.
— Extract the contents of the package.
— Extract the individual projects to a short path.
— Launch e² studio from the start menu.
— Set the top directory which has each sample project sub-directory for the workspace directory. These 2 steps are shown in Figure 3-1.

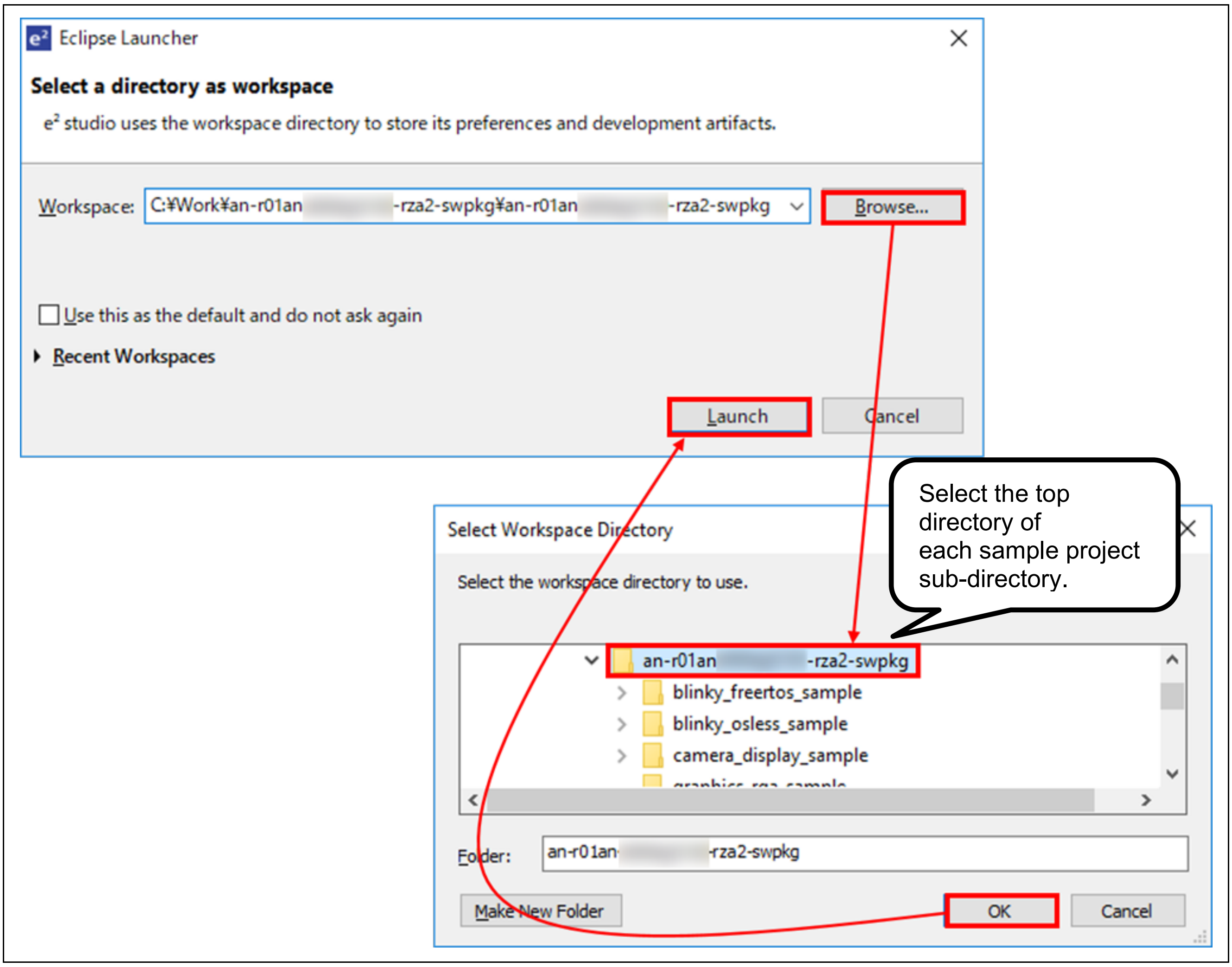

Figure 3-1 e² studio launching

— In the e² studio welcome screen, click 'Workbench'. This is shown in Figure 3-2.

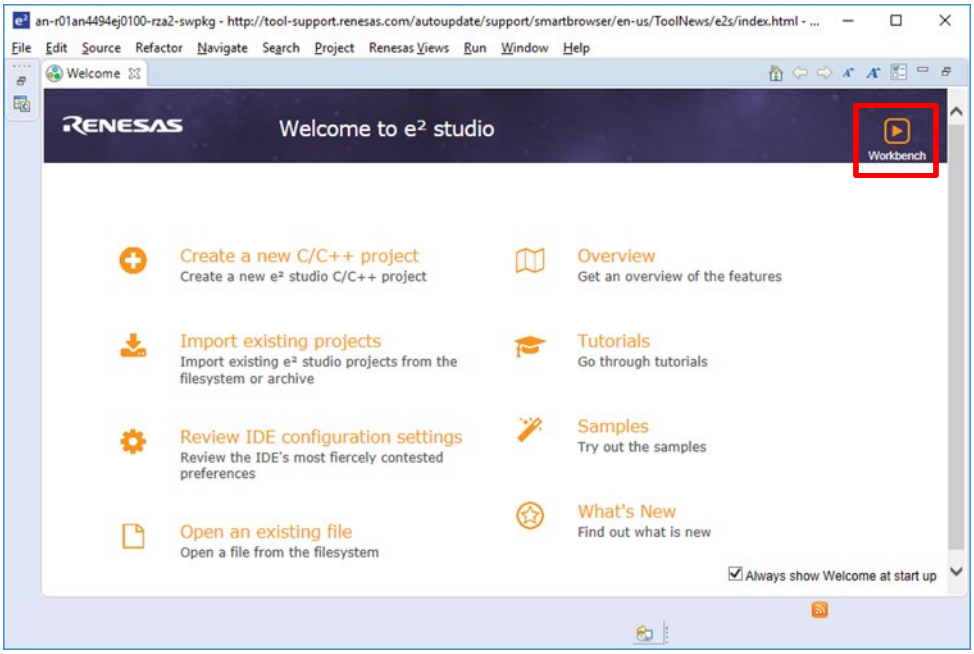

Figure 3-2 Position of 'Workbench' switch

— Right-click in the Project Explorer window and select 'Import'. This is shown in Figure 3-3.

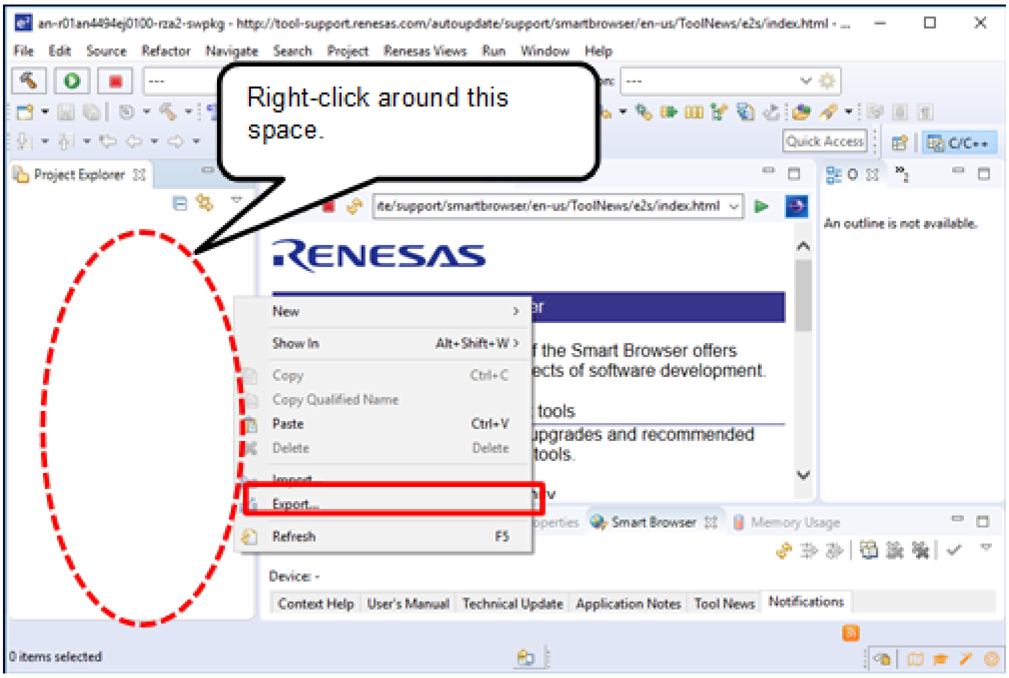

Figure 3-3 Selecting 'Import'

— Under 'Import' window, select General > Existing Projects into Workspace and click 'Next'. This is shown in Figure 3-4.

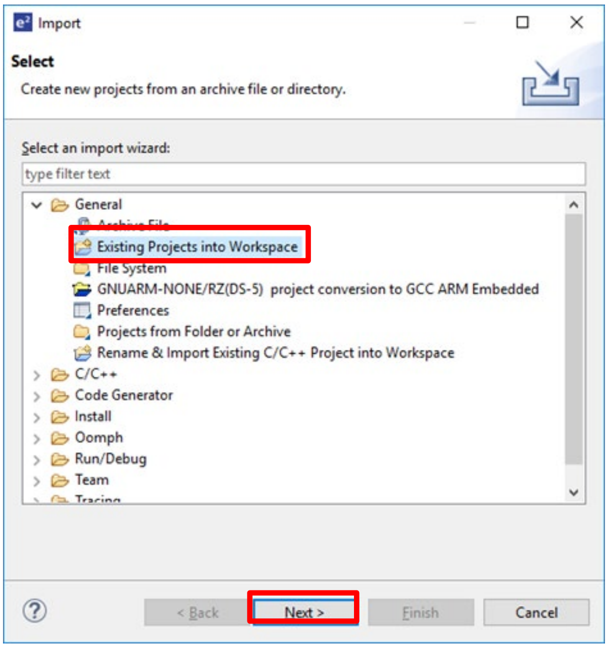

Figure 3-4 Menu of 'Import' window

— Select "Browse" at the right of "Select root directory:", and "Browse for Folder" dialog box will be appeared. Click "OK". These 2 steps are shown in Figure 3-5.

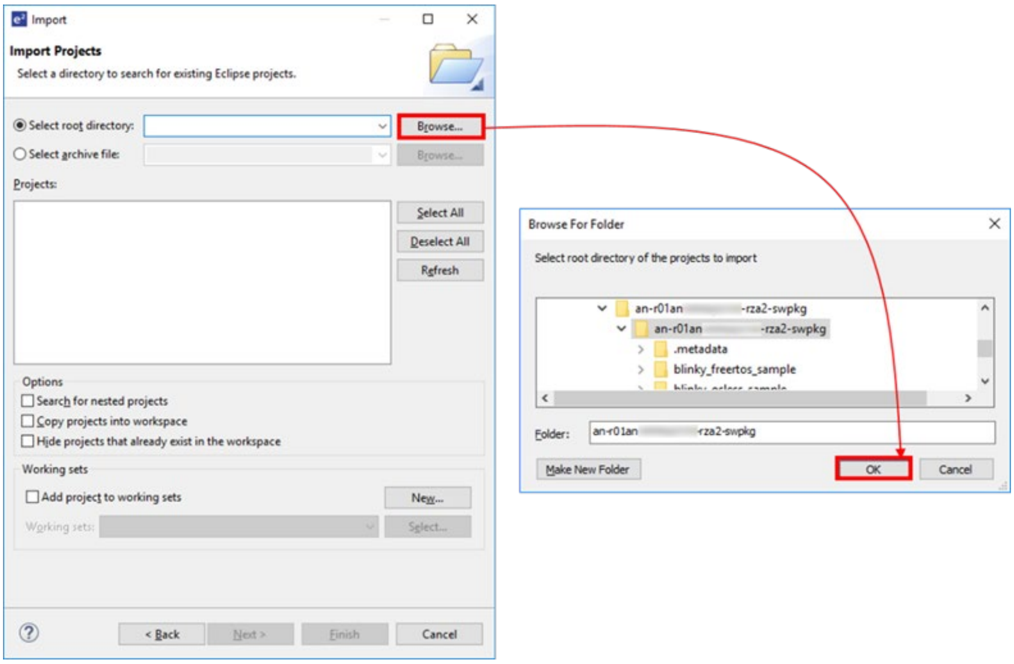

Figure 3-5 Select root directory

— Confirm your target project is checked, then click ‘Finish’. This is shown in Figure 3-6.
(Note: Projects in Figure 3-6 are just sample. From here, please read the project name as your target project name.)

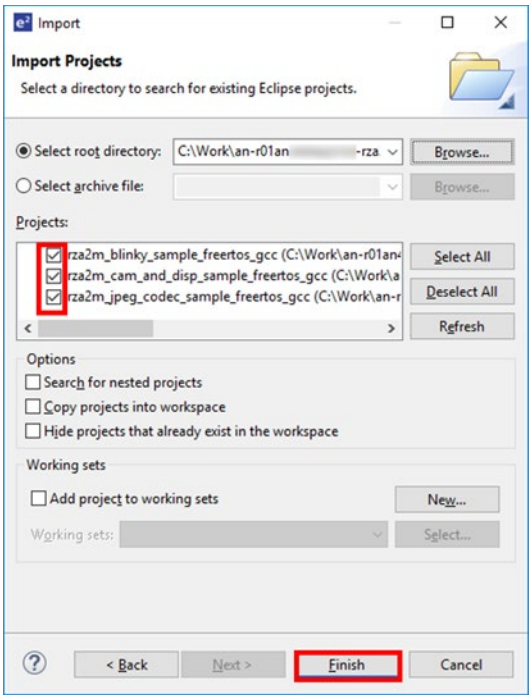

Figure 3-6 Import target project

— Now, target projects are imported, and user can see them in the Project Explorer window. This is shown in Figure 3-7.

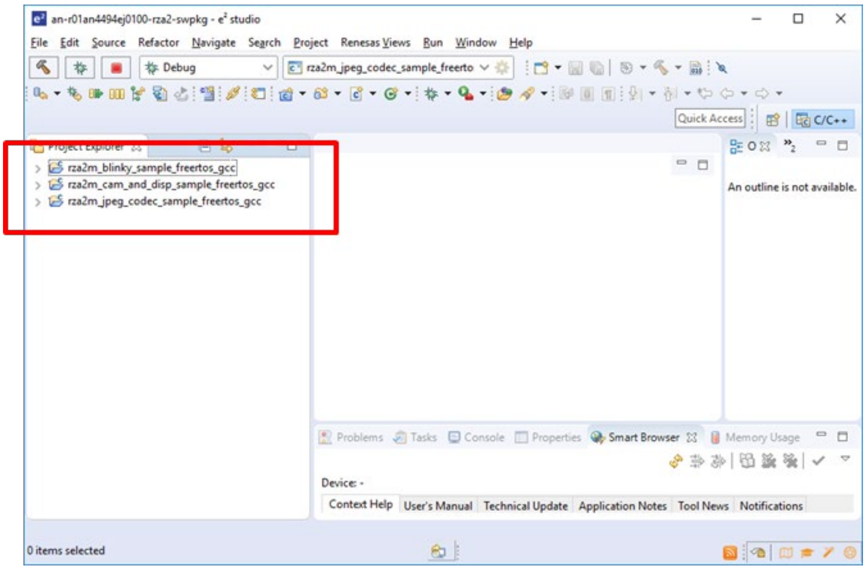

Figure 3-7 Confirmation on ‘Project Explorer’ window

3.2 Build and Download to target board

— Select your target project by left clicking on it, then click the arrow next to build button (hammer icon) and select ‘Hardware Debug’ from the drop-down menu. From next time, user can build the project by this Build button (hammer icon). This is shown in Figure 3-8

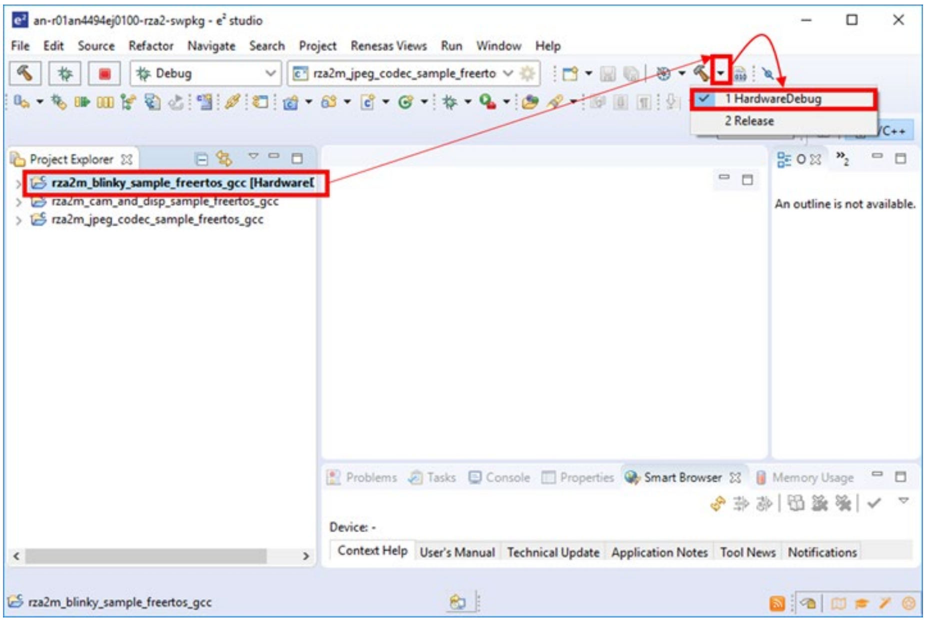

Figure 3-8 Build the target project

— e² studio tool build the project, and the build status can be confirmed in Console window (Note: Please mind the length of your workspace path. If the path is too long, there is a possibility of build error.) This is shown in Figure 3-9.

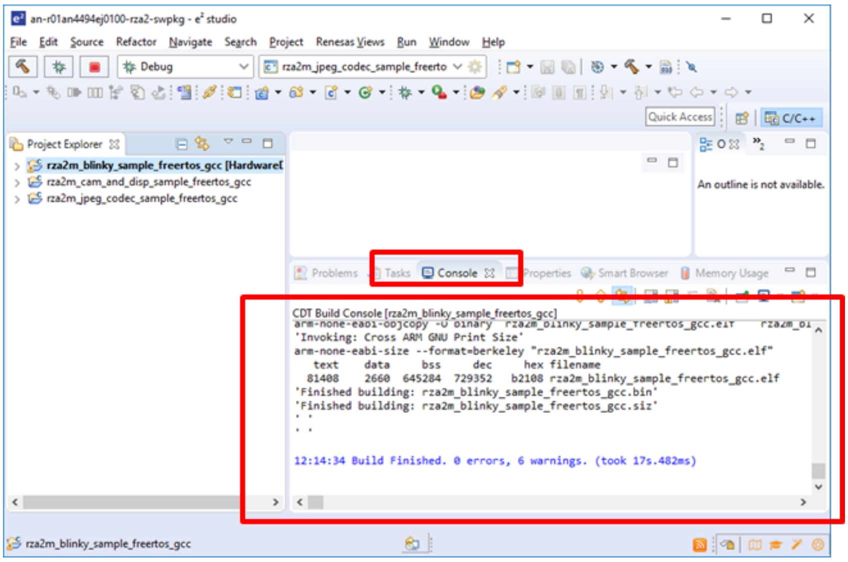

Figure 3-9 Confirmation build status

— After the build is completed, the binary format file is generated in the “Hardware Debug” directory of the target project. When connect the PC and GR-MANGO via USB cable, PC detects GR-MANGO as MBED drive. User can download the program by drag-and-drop the program binary file to MBED drive.

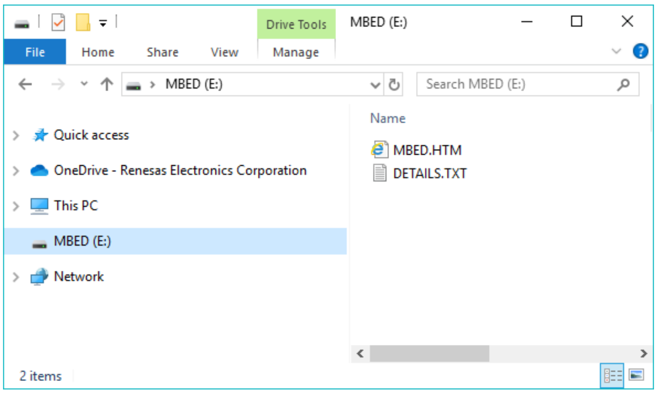

Figure 3-10 Download the program

— Execute the program by push the reset button of GR-MANGO after download.

3.3 CMSIS-DAP based debugging

GR-MANGO supports ARM Mbed DAPLink feature and user can debug with OpenOCD.

Please refer page below to make out how to debug GR-MANGO with OpenOCD.

[https://os.mbed.com/teams/Renesas/wiki/How-to-debug-with-e^2-studio](https://os.mbed.com/teams/Renesas/wiki/How-to-debug-with-e2-studio)

4. Adding drivers and middleware

This section describes how to integrate drivers, middleware into the project which included in this package. In RZ/A2M Software Package for GR-MANGO, the drivers and middleware are managed as components, and the components can be added by e² studio.

Refer the sample projects bundled in RZ/A2M Simple Applications Package for GR-MANGO(R01AN5595) for examples of usage of each driver and each middleware.

Refer [RZ/A2M Smart Configurator User's Guide: e² studio](R20AN0583) for the usage of Smart Configurator. e.g.) how to install drivers and middleware to e² studio.

4.1 Importing Software Package into IDE

- Open the project tree of target project in the Project Explorer window of e² studio, and double click .scfg file in the project.
- Select 'Components' tub and add the target component from 'Add component' button.
- After adding the target component, click 'Generate Code' button.
- The steps above are shown in Figure 4-1.
- With this, the component is added to the target project's folder, "(Project Folder)\generate\sc_drivers" and "(Project Folder)\.settings\smartconfigurator". After adding the component, re-build the target project according to section 3.2.

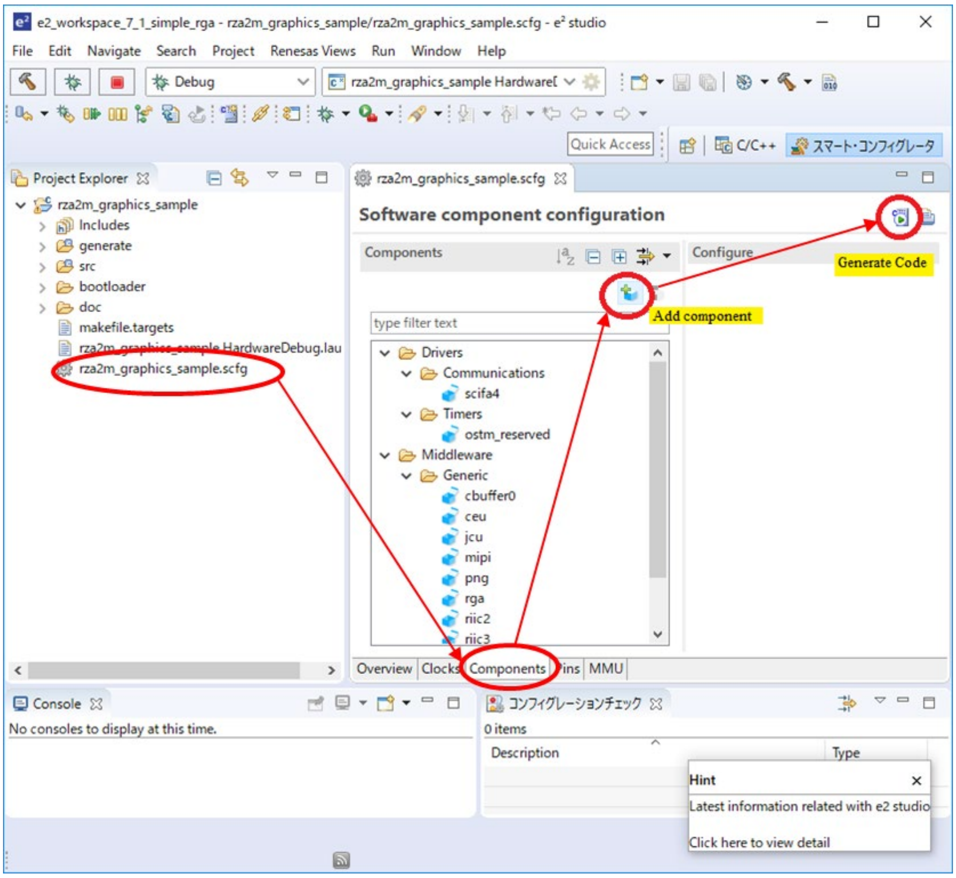

Figure 4-1 How to add components

4.2 How to integrate the new component

- New component is integrated to a project by the following step
- Add component by the procedure shown in section 4.1.
- Configure the component by Smart Configurator.
- Generate the source code of the component.
- Confirm API functions or the name of header file declaring API functions from the document of the component.
- Confirm the project which uses the component and use the API function name or the header file name to find where the component is used.
- Implement the source code into a project by referring the source file using the component.

5. ThreadX Awareness function

In this section it is shown that ThreadX Awareness function of e[2] studio.

This function supports displaying the list and the status of generated threads, queues, timers and other kernel resource during break.

5.1 How to launch ThreadX Awareness function

1. Download the program using ThreadX to your board.
2. Run the downloaded program.
3. Suspend (break) the running program.
4. Select "Renesas Views" -"Partner OS"-"RTOS Resources".

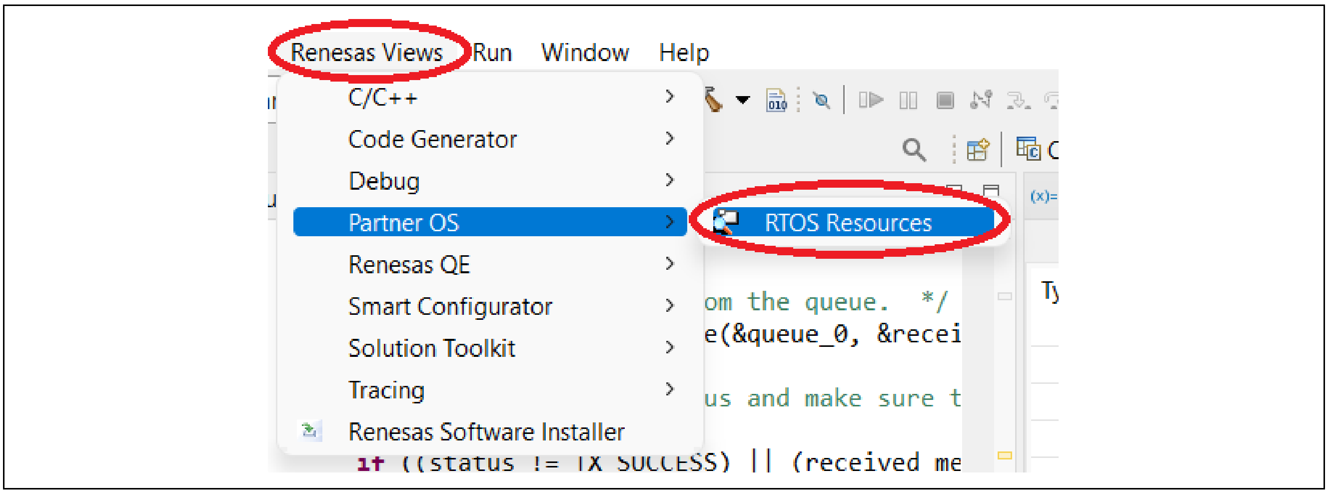

Figure 5-1 Launching ThreadX Awareness function

- 5. The RTOS resource tab will be displayed. Select "ThreadX" from [Select OS] [OS:] and click the [OK] button.

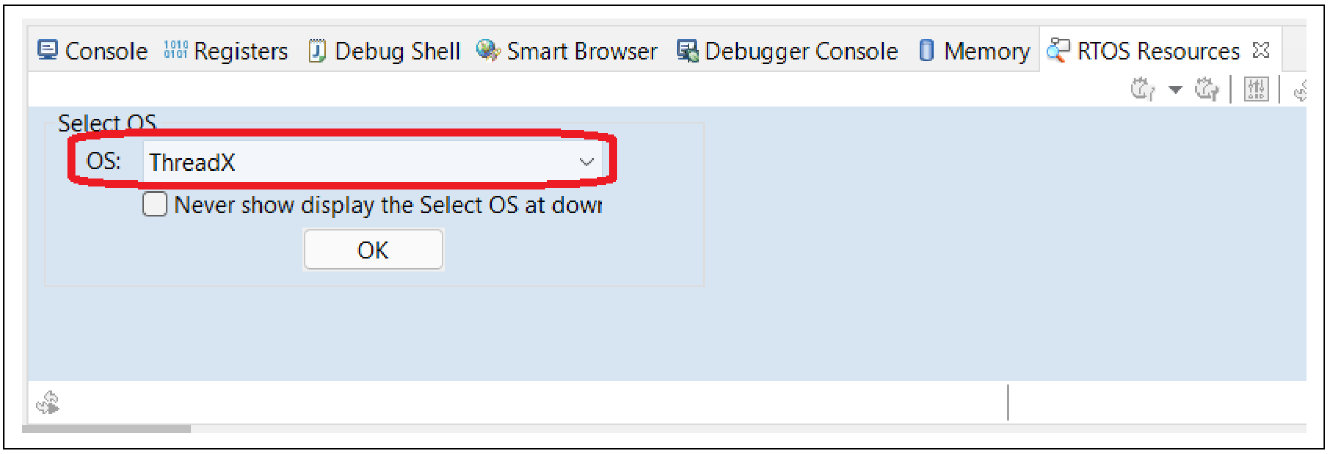

Figure 5-2 RTOS resource(Select OS)

- 6. The resources for the kernel object are displayed. The display of each resource can be switched by clicking the resource name tab inside the RTOS resource tab.

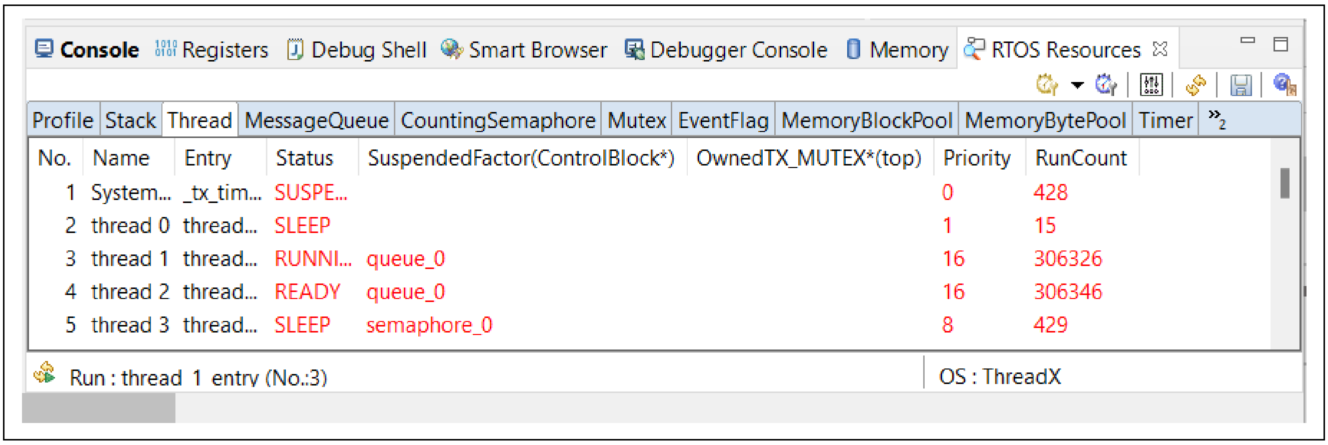

Figure 5-3 RTOS resource(Kernel objects)

6. Azure RTOS Sample Projects

Each sample project included in this package works on the GR-MANGO board manufactured by CORE CORPORATION and the operation of the Renesas e² studio. This chapter describes the procedure for executing each sample project.

6.1 How to build Azure RTOS components

This package contains projects for all components of the Azure RTOS except NetX.

Note: The Next is an Azure RTOS component of TCP/IP stack of IPv4 only. The sample project for the package uses NetX Duo with IPv4/IPv6 dual stack, but it is possible to replace NetX Duo with NetX.

All Azure RTOS components must be imported into the e²studio environment, built and created library files before running each sample project.

Follow the steps below to build each component projects and create library files.

1. Select the component project to be built on the [Project Explorer] of e² studio.
2. Make sure the selected project has the [Debug] build configuration.

   Note: The figure below shows the case where the build configuration of the "filex" component project is [Debug]. Select either Debug or Release for the build configuration of the Azure RTOS component. If you want to create a sample project with the [Release] build configuration, set the build configuration of all Azure RTOS components to [Release] and run the build.

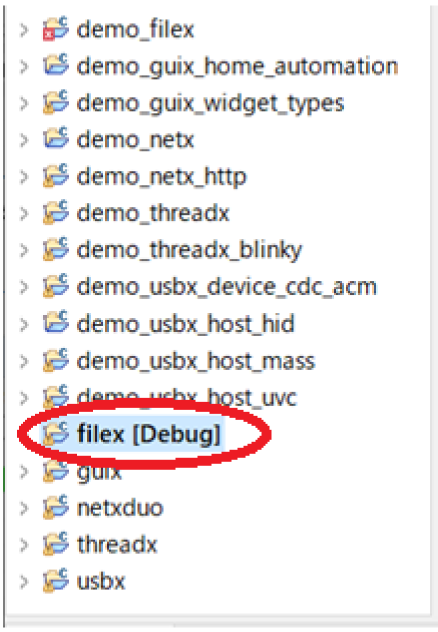

Figure 6-1 Select the "filex" component

3. Right-click on the component project you want to build. Select "Build Project" from the menu that appears to build the project.

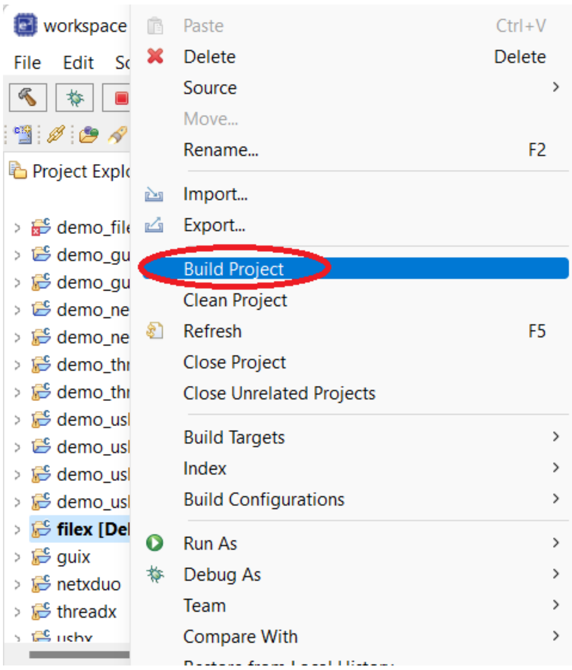

Figure 6-2 Select [Build Project] for "filex" component

4. Confirm that the project build is completed with "0 errors" (normal completion).

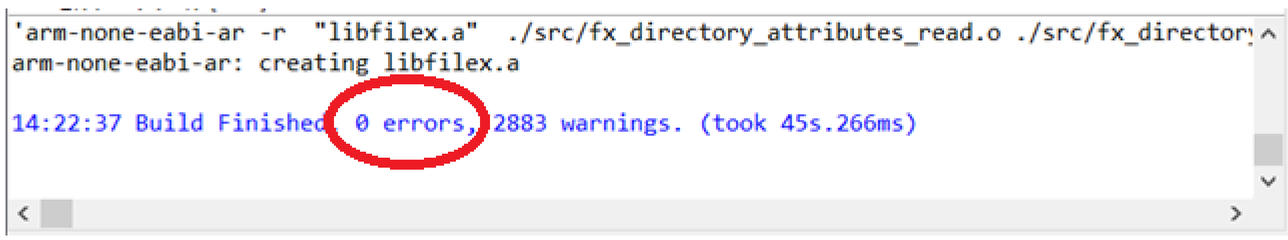

Figure 6-3 Confirm that the "filex" component finished normally

5. Please repeat steps 1 through 4 for all Azure RTOS components, filex, guix, netxduo, threadx, usbx.

6.2 Sample projects

This package provides sample projects for each component of Azure RTOS and some peripheral devices of RZ/A2M. The sample project can be used as a sample of how to use each component and peripheral devices.

The following sections describe each sample project.

6.2.1 demo_filex

The "demo_filex" is a sample project to check the function of FileX.

This sample project outputs the status to the console while writing a file called "TEST.TXT" to the media when the media is inserted into the Micro SD Card Slot of GR-MANGO.

6.2.1.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- FileX

6.2.1.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.

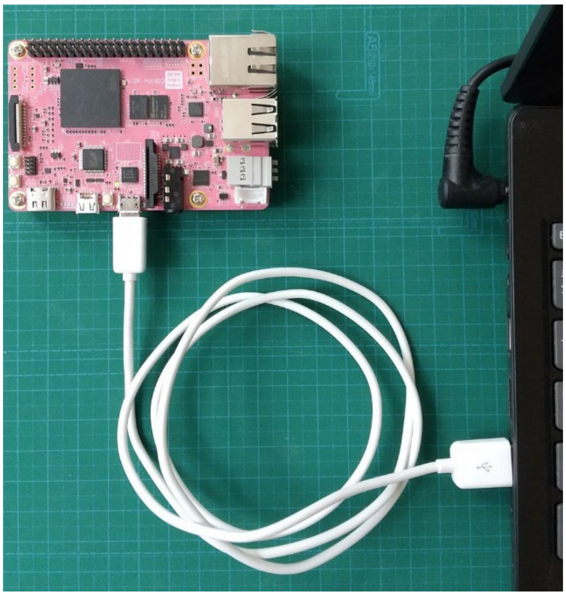

Figure 6-4 Connecting target board and PC

6.2.1.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_filex" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.

   Note: The figure below shows the case where the build configuration of the "demo_filex" sample project is [Hardware Debug]. Select either [Hardware Debug] or [Release] for the build configuration of the sample project. If you want to create a sample project with [Release] build configuration, change the build configuration to [Release] and execute the build.

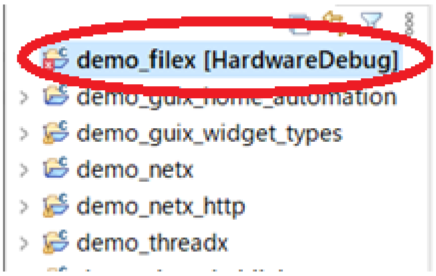

Figure 6-5 Build configuration selection

3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.1.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.

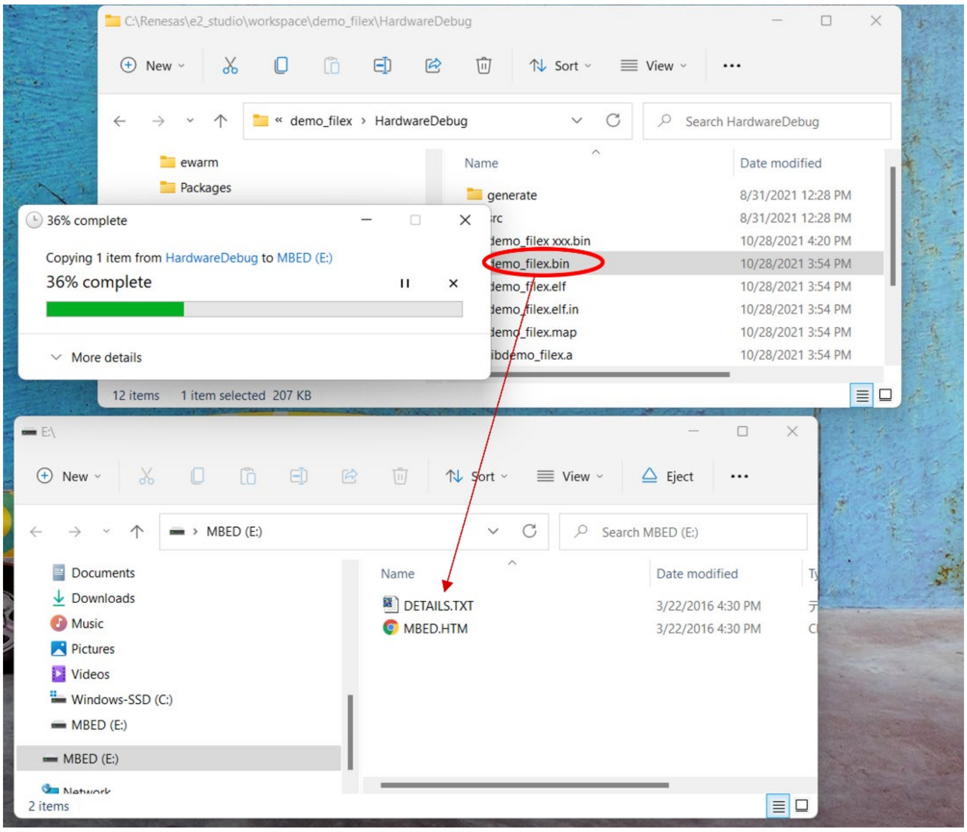

Figure 6-6 Copy the .bin file

4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Start the terminal software (e.g., Tera term) on the PC and connect to the COM port recognized when the board is connected.
7. Unplug the USB cable that connects your PC to the board and reconnect it.
8. Confirm that the following message is displayed on the terminal software.

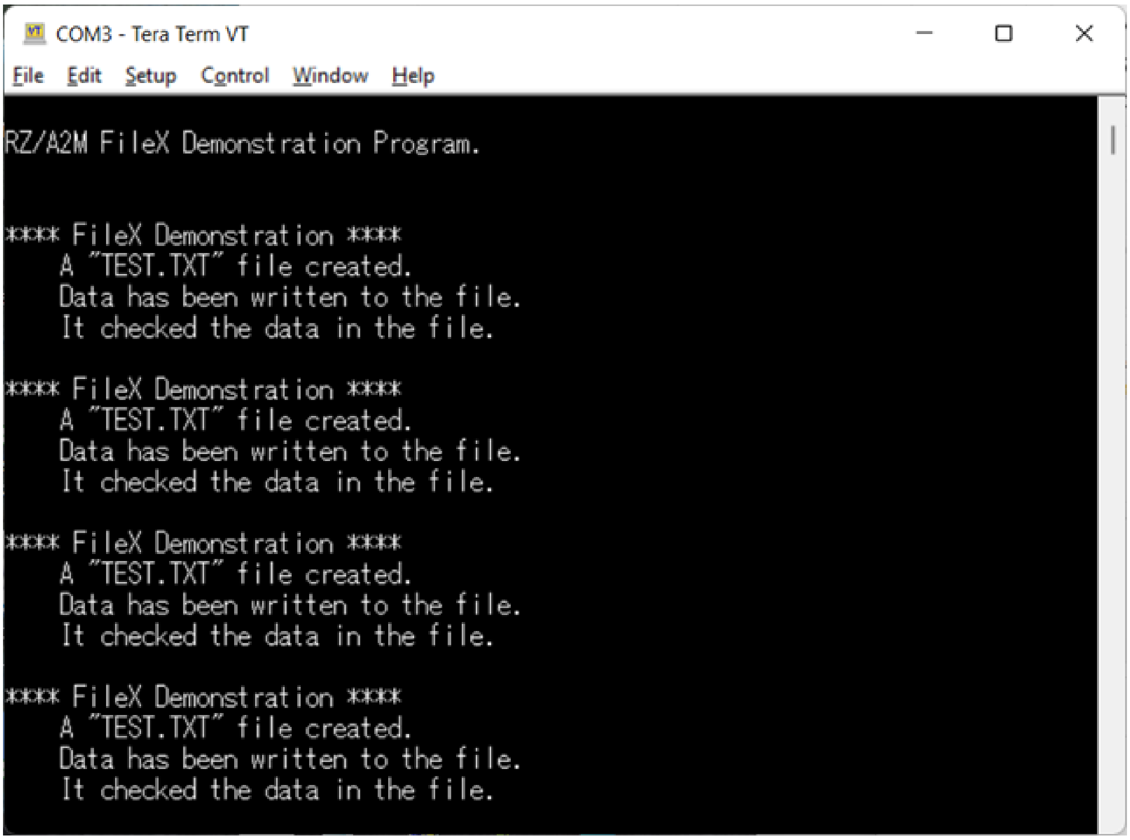

Figure 6-7 Message from "demo_filex"

6.2.2 demo_guix_home_automation

The "demo_guix_home_automation" is a sample project of GUIX and USBX.

Please use the lower part of the GR-MANGO USB connector Type-A port (CN5). Also, connect the HDMI port of GR-MANGO to the monitor and check the display.

6.2.2.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- GUIX
- USBX

6.2.2.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and mouse in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.

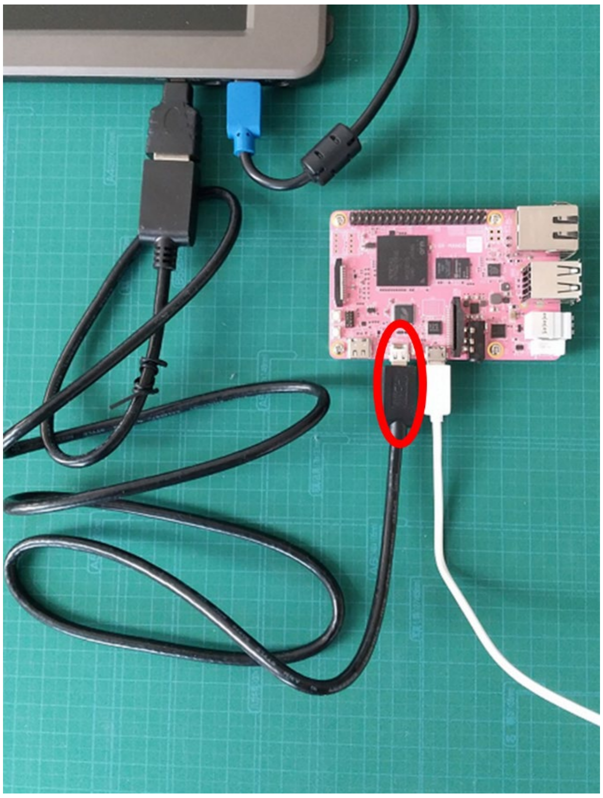

Figure 6-8 Connecting of HDMI monitor

3. Connect the USB mouse to the bottom of the CN5 connector on the board.

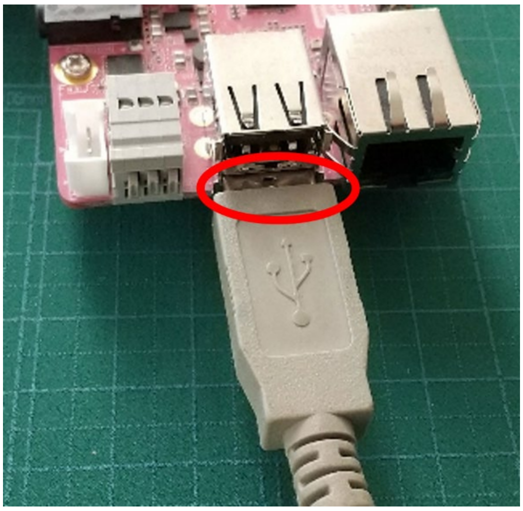

Figure 6-9 Connecting of USB mouse

6.2.2.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_guix_home_automation" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.2.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the following screen is displayed on the HDMI monitor.

Figure 6-10 Display screen of "demo_guix_home_automation"

8. Please make sure that you can operate the buttons on the screen with the mouse.

6.2.3 demo_guix_washing_machine

The "demo_guix_washing_machine" is a sample project of GUIX and USBX.

Please use the lower part of the GR-MANGO USB connector Type-A port (CN5). Also, connect the HDMI port of GR-MANGO to the monitor and check the display.

6.2.3.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- GUIX
- USBX

6.2.3.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and mouse in the following steps.

Note: Please also refer to "demo_guix_home_automation" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the USB mouse to the bottom of the CN5 connector on the board.

6.2.3.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_guix_washing_machine" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.3.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the following screen is displayed on the HDMI monitor.

Figure 6-11 Display screen of "demo_guix_washing_machine"

8. Please make sure that you can operate the buttons on the screen with the mouse.

6.2.4 demo_guix_widget_types

The "demo_guix_widget_types" is a sample project of GUIX and USBX.

Please use the lower part of the GR-MANGO USB connector Type-A port (CN5). Also, connect the HDMI port of GR-MANGO to the monitor and check the display.

6.2.4.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- GUIX
- USBX

6.2.4.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and mouse in the following steps.

Note: Please also refer to "demo_guix_home_automation" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the USB mouse to the bottom of the CN5 connector on the board.

6.2.4.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_guix_widget_types" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.4.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the following screen is displayed on the HDMI monitor.

Figure 6-12 Display screen of "demo_guix_widget_types"

8. Please make sure that you can operate the buttons on the screen with the mouse.

6.2.5 demo_netx

The "demo_netx" is a sample project of NetXDuo. Returns a response to a ping request from ping command.

Note:

The IP address of the board is set to 192.168.2.120 and the netmask is set to 255.255.255.0. If the operating network environment does not match the set IP address, change the IP address set with nx_ip_create () or change the IP address on your PC side.

6.2.5.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- NetXDuo

6.2.5.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the Ethernet connector of the PC and the CN8 connector of the board with the Ethernet cable.

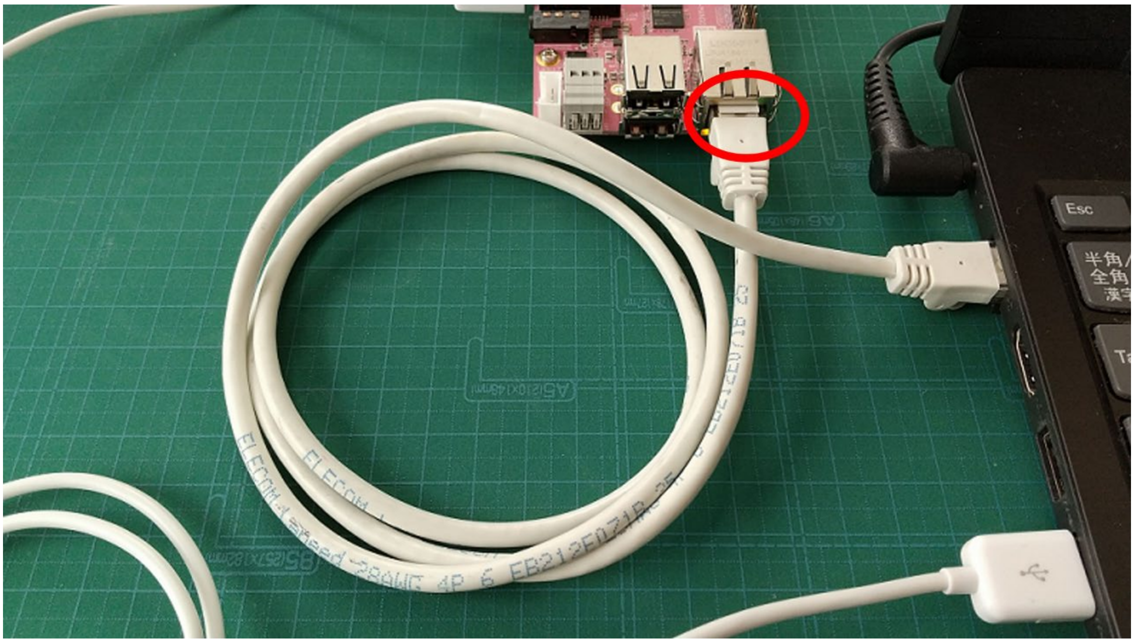

Figure 6-13 Connect Ethernet cable

6.2.5.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_netx" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.5.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Start the command prompt of the PC and execute the ping command to the IP address set on the board.
8. Confirm that the ping command responds as shown below.

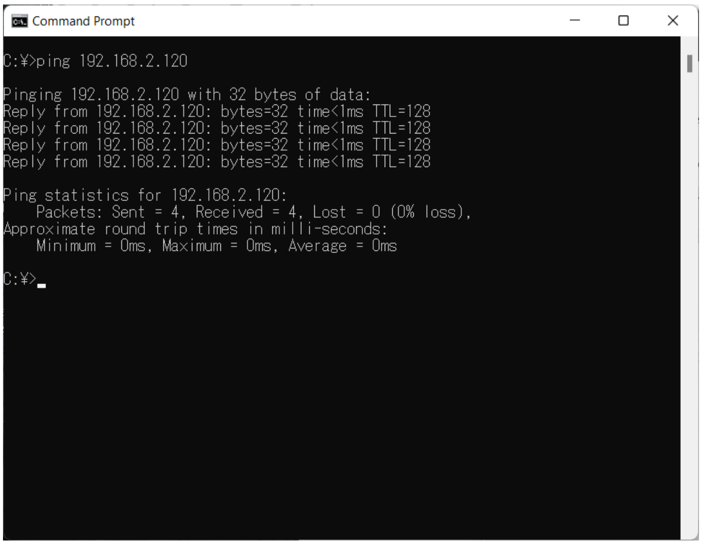

Figure 6-14 "Demo_netx" ping response

6.2.5.5 Changes to NetX

This sample project uses NetXDuo. If your application is IPv4-only, you can use NetX instead of NetXDuo.

When using NetX, create a separate library project for NetX, and create an application by linking the NetX library together with the application instead of the NetX Duo library.

6.2.6 demo_netx_http

The "demo_netx_http" is a sample project of NetX Duo HTTP server. Responds to HTTP requests from the browser.

Note:

The IP address of the board is set to 192.168.2.120 and the netmask is set to 255.255.255.0. If the operating network environment does not match the set IP address, change the IP address set with nx_ip_create () or change the IP address on your PC side.

6.2.6.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- NetXDuo

6.2.6.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the Ethernet connector of the PC and the CN8 connector of the board with the Ethernet cable.

6.2.6.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_netx_http" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.6.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Start the browser on your PC and access the IP address set on the board.
8. Confirm that the following browser is displayed in the browser.

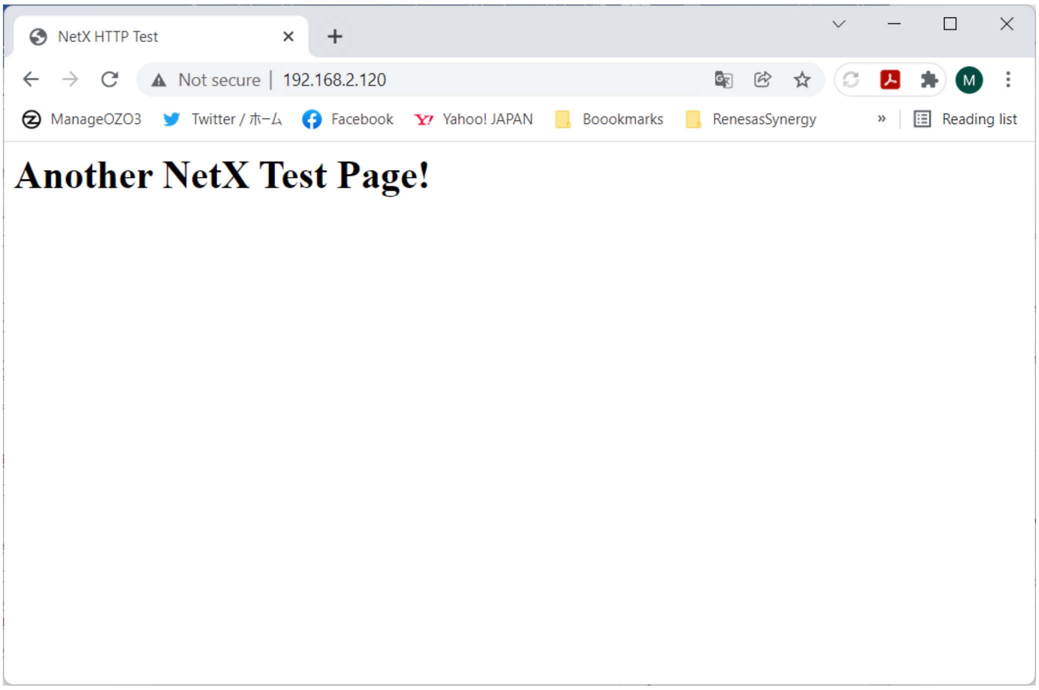

Figure 6-15 Browser display of "demo_netx_http"

6.2.6.5 Changes to NetX

This sample project uses NetXDuo. If your application is IPv4-only, you can use NetX instead of NetXDuo.

When using NetX, create a separate library project for NetX, and create an application by linking the NetX library together with the application instead of the NetX Duo library.

6.2.7 demo_threadx

The "demo_threadx" is a sample project to check the operation of ThreadX.

This sample project blinks the LEDs on the board at 0.5 second intervals.

6.2.7.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX

6.2.7.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.

6.2.7.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_threadx" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.7.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the following message is displayed on the terminal software.

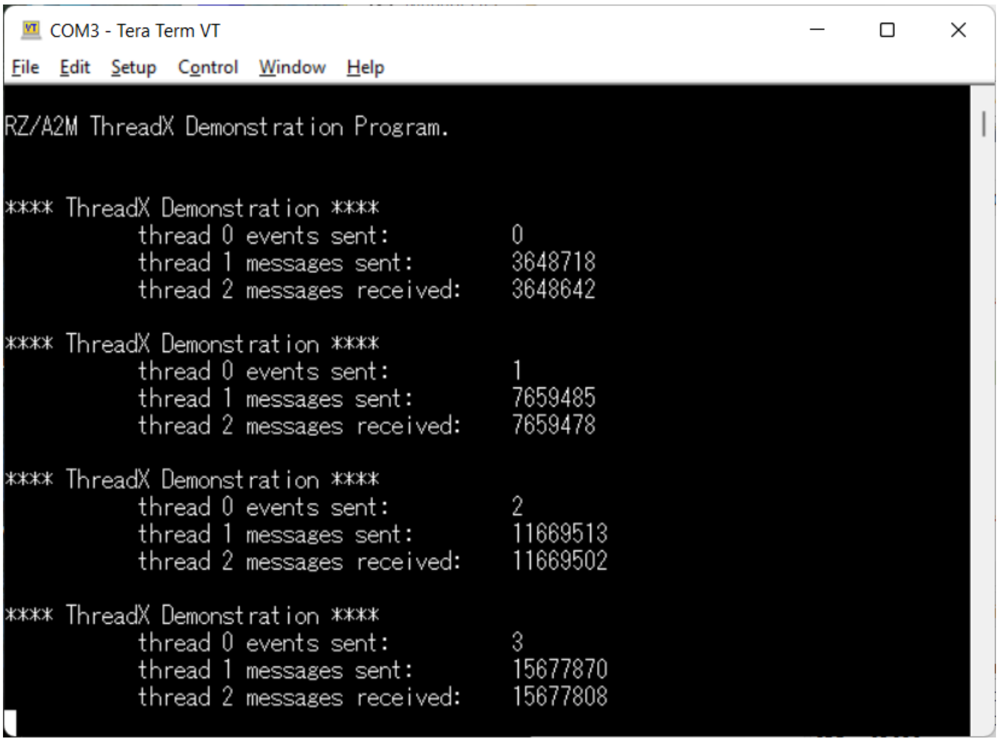

Figure 6-16 Message from "demo_threadx"

6.2.8 demo_threadx_blinky

The "demo_threadx_blinky" is a sample project to check the operation of ThreadX.

This sample project blinks the LEDs on the board at 0.5 second intervals.

6.2.8.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX

6.2.8.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.

6.2.8.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_threadx" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.8.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the LED on the board flashes every 0.5 seconds.

6.2.9 demo_usbx_device_cdc_acm

The "demo_usbx_device_cdc_acm" is a sample project to confirm the CDC ACM function of USBX device.

This sample project works as a CDC ACM function of USB, is recognized as a serial port by Windows, and echoes back the input characters. If the entered character is a line feed code, a message will be output to the console.

6.2.9.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- USBX

6.2.9.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.
2. Connect the CN6 connector (USB Type-C connector) of the board to the PC with a USB cable.

6.2.9.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_usbx_device_cdc_acm" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.9.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Start the terminal software (e.g., Teraterm) on the PC and connect to the COM port recognized when the board is connected.
7. Unplug the USB cable that connects your PC to the board and reconnect it.
8. When the board boots again, it will be recognized by the PC as a separate UART (CDC ACM) from the console.
9. Start the terminal software on the PC and connect to a UART different from the console.
10. Enter any characters into the terminal software connected to another UART and check that the entered characters are echoed back.
11. Confirm that a message is output to the console if the entered character has a line feed code.

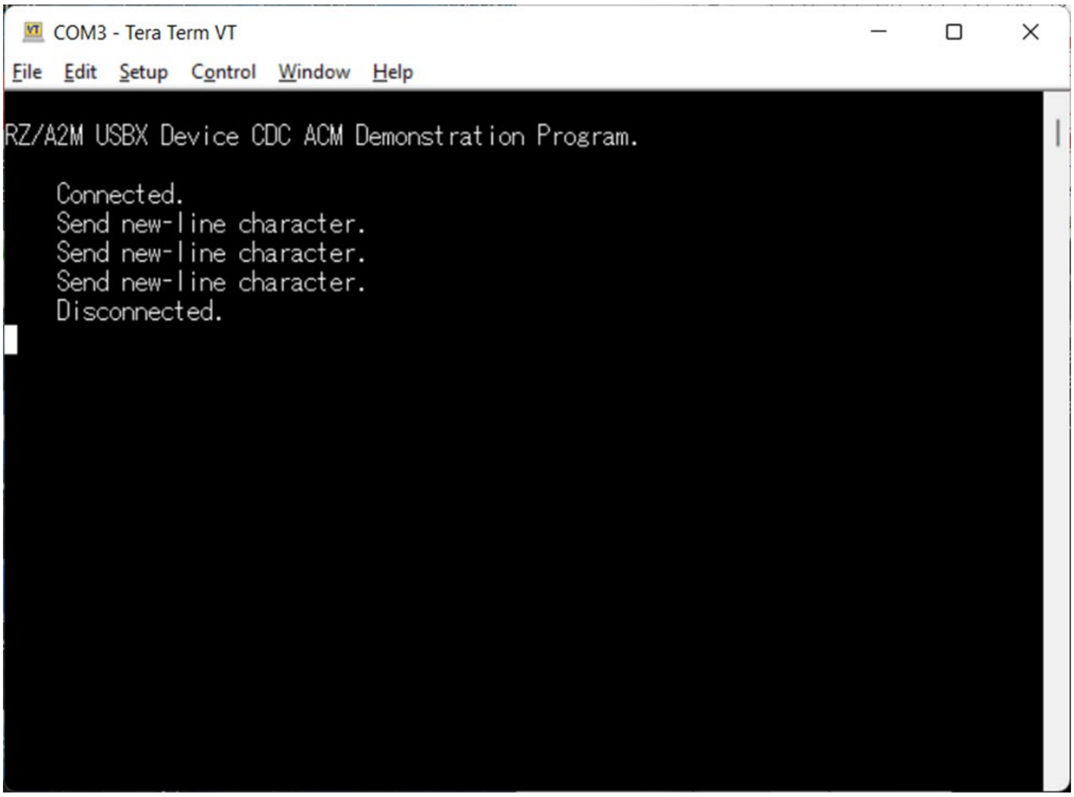

Figure 6-17 Message from "demo_usbx_cdc_acm"

6.2.10 demo_usbx_host_hid

The "demo_usbx_host_hid" is a sample project of HID function of USBX host.

Please use the lower part of the GR-MANGO USB connector Type-A port (CN5).

6.2.10.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- USBX

6.2.10.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.
2. Connect the USB mouse to the bottom of the CN5 connector on the board.

6.2.10.3 How to build sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_usbx_host_hid" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.10.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Start the terminal software (e.g., Tera term) on the PC and connect to the COM port recognized when the board is connected.
7. Unplug the USB cable that connects your PC to the board and reconnect it.
8. Confirm that the following mouse position information is displayed on the console.

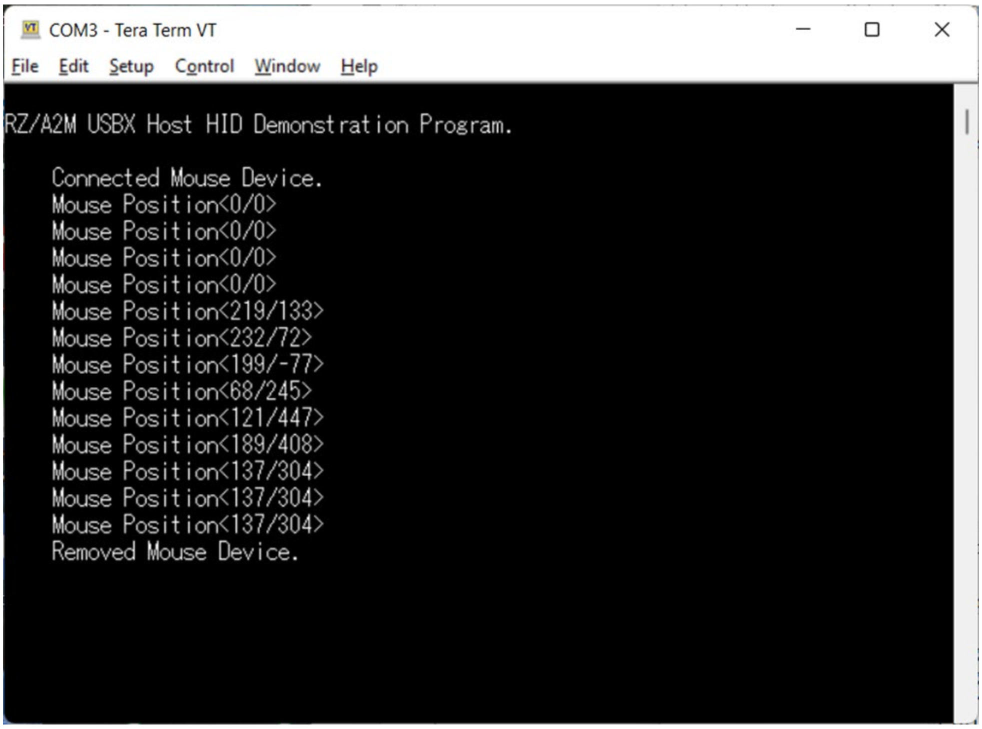

Figure 6-18 Display mouse position information of "demo_usbx_hid"

6.2.11 demo_usbx_host_mass

The "demo_usbx_host_mass" is a sample project of MSC function of USBX host.

Please use the lower part of the GR-MANGO USB connector Type-A port (CN5).

6.2.11.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- USBX

6.2.11.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.
2. Connect the USB memory to the bottom of the CN5 connector on the board.

6.2.11.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_usbx_host_mass" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.11.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT". Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Start the terminal software (e.g., Tera term) on the PC and connect to the COM port recognized when the board is connected.
7. Unplug the USB cable that connects your PC to the board and reconnect it.
8. Confirm that the information of the file in the USB memory is displayed on the console as shown below.

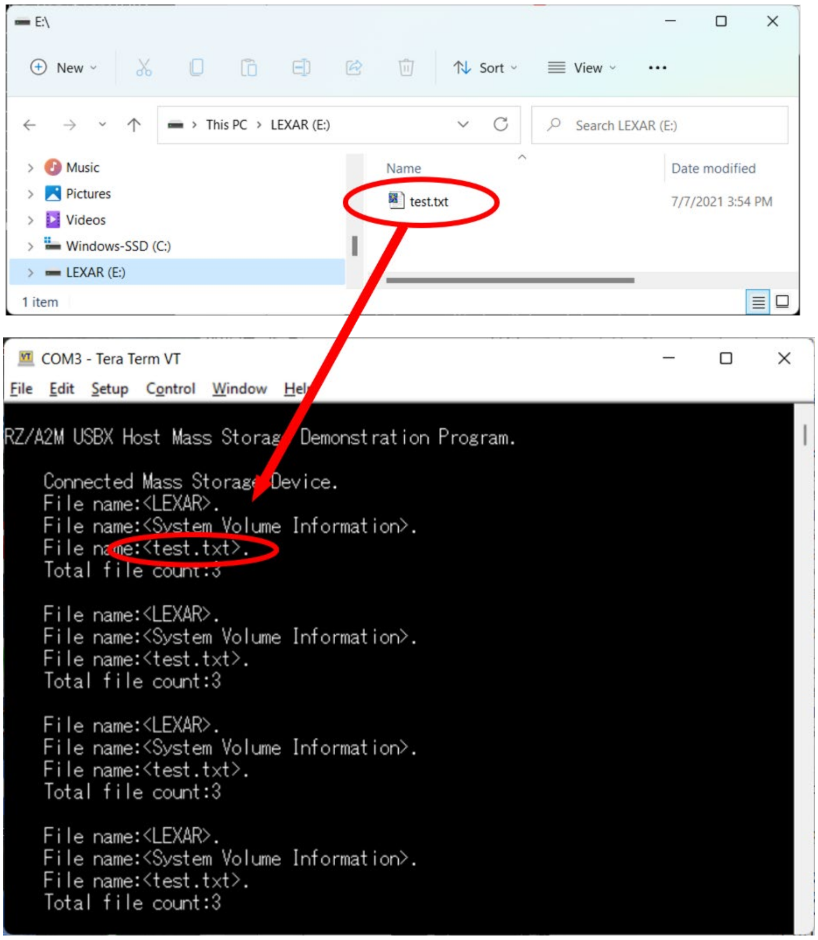

Figure 6-19 Displaying the contents of the USB memory of "demo_usbx_host_mass"

6.2.12 demo_usbx_host_uvc

The "demo_usbx_host_uvc" is a USB Video Class (UVC) sample project for a USB X host.

Please use the upper part of the GR-MANGO USB connector Type-A port (CN5). You can check the camera image with the browser on your PC.

Note:

The IP address of the board is set to 192.168.2.120 and the netmask is set to 255.255.255.0. If the operating network environment does not match the set IP address, change the IP address set with nx_ip_create () or change the IP address on your PC side.

6.2.12.1 Azure RTOS components

This sample project uses the following Azure RTOS components.

- ThreadX
- USBX
- NetXDuo

6.2.12.2 How to connect to the target board

Connect the target board and the PC in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.
2. Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console. Connect the Ethernet connector of the PC and the CN8 connector of the board with the Ethernet cable.
3. Connect the USB camera to the upper of the CN5 connector on the board.

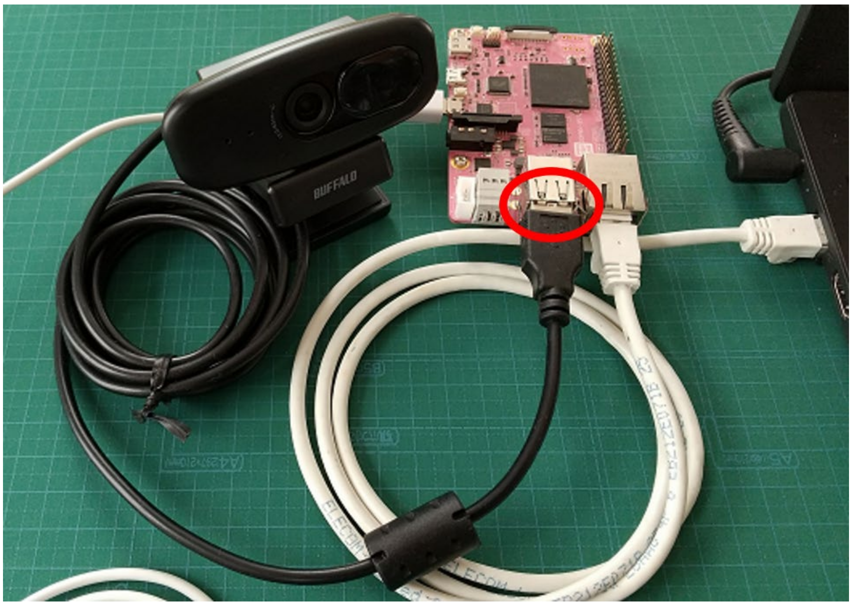

Figure 6-20 Connecting USB camera

6.2.12.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_usbx_host_uvc" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.12.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Start the terminal software (e.g., Tera term) on the PC and connect to the COM port recognized when the board is connected.
7. Unplug the USB cable that connects your PC to the board and reconnect it.
8. Start a browser (e.g., Google Chrome) that supports motion JPEG on your PC and access the IP address set on the board. Confirm that the image from the camera shown below is displayed in the browser.

Figure 6-21 Displaying the camera image of "demo_usbx_host_uvc"

6.2.13 demo_cam_and_disp

The “demo_cam_and_disp” is a sample project to capture an image from the MIPI camera and display the captured image. Also, this sample project adjusts an image from the camera as follows by the CUI (Character User Interface) tool.

Table 6-1 Image adjustments supported by this sample project

Peripheral device	Adjustments	Description
VDC	Brightness (Note 1)	Adjusts brightness by changing the luminance components.
	Contrast (Note 1)	Adjusts contrast by changing the color components.
	Sharpness	Sharpens the outline of an edge by adjusting overshoot and undershoot.
	Gamma adjustment	Makes a gamma adjustment with a preset value (four types of preset gamma adjustments are enabled with preset values).
	Dithering process	Performs dithering with random patterns.
	Rotation and Horizontal Mirroring (Note 2)	Rotate an image 180 degree or flip horizontally.

Note 1. The adjustment timing is limited for camera input through the MIPI and CEU.
Note 2. Unusable in the case of camera input with MIPI and CEU.

6.2.13.1 Peripheral devices

This sample project uses the following peripheral devices.

- VDC6
- MIPI・VIN

6.2.13.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

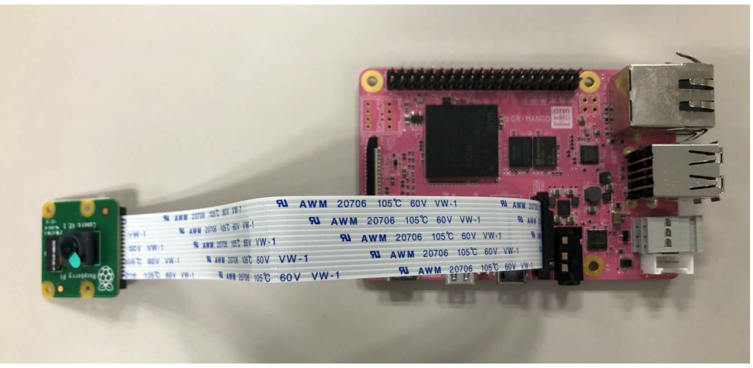

Figure 6-22 Connecting target board and MIPI Camera

6.2.13.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_cam_and_disp" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.13.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Start the terminal software (e.g., Tera term) on the PC and connect to the COM port recognized when the board is connected.
7. Unplug the USB cable that connects your PC to the board and reconnect it.
8. Confirm that the image from the camera shown below is displayed on the HDMI monitor.

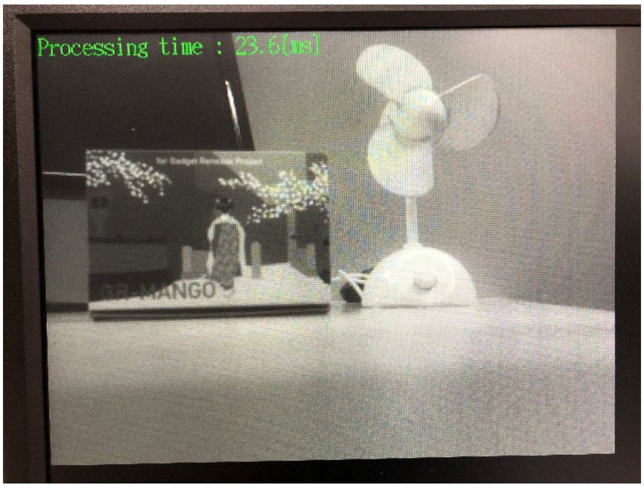

Figure 6-23 Displayed screen of "demo_cam_and_disp"

9. Confirm that the following message is displayed on the terminal software.

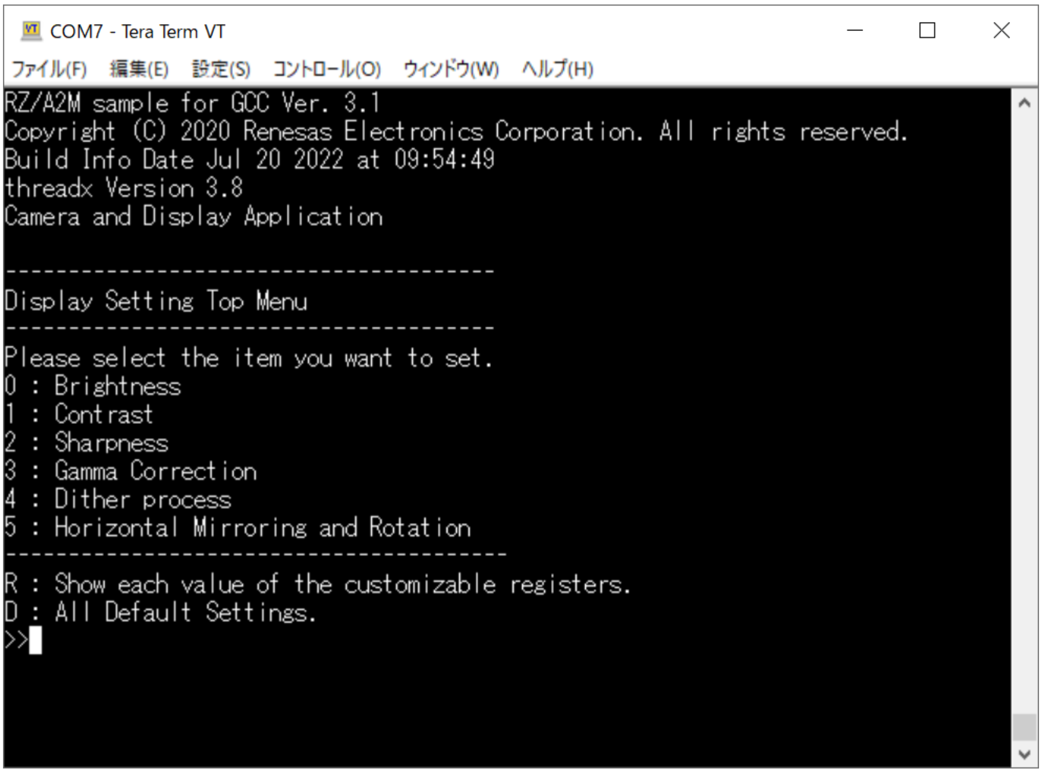

Figure 6-24 Message from "demo_cam_and_disp"

10. Confirm that the displayed image is adjusted by executing following commands on the terminal software.

Table 6-2 CUI operation commands on terminal application

Command	Operation
Numerical value	Operations in each menu - Selection of image adjustment content - Selection of image adjustment position (selection of H/W block for image adjustment) - Selection of presets - Input custom value
C, c	Custom setting selection (Selected when user wants to set a preset other than the various adjustment items)
D, d	Set image adjustment to default (Default value of each register described in H/W manual)
B, b	Return to the previous menu
R, r	Output current image adjustment value
T, t	Return to the Top menu
Enter	Determine contents inputted
Delete/ Back space	Delete one character from the input character

6.2.14 demo_drp_basic_sample

The “demo_drp_basic_sample” is a sample project to check the basic function of DRP.

This sample project converts the input image from MIPI camera to grayscale image using DRP Library and outputs to display.

6.2.14.1 Peripheral devices

This sample project uses the following peripheral devices.

- DRP

6.2.14.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" and "demo_cam_and_disp" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

6.2.14.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_drp_basic_sample " to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.14.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the image from the camera is displayed on the HDMI monitor.

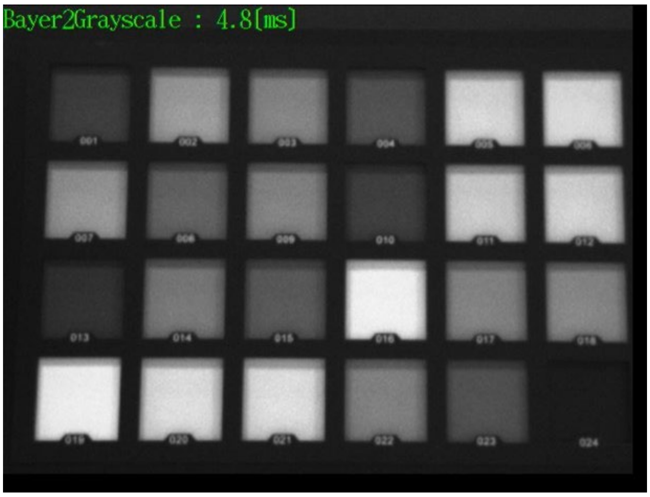

Figure 6-25 Displayed screen of "demo_drp_basic_sample"

6.2.15 demo_drp_dynamic_sample1

The “demo_drp_dynamic_sample1” is a sample project to check the dynamic loading of DRP.

This sample project detects the edges of the input image from MIPI camera by Canny method using DRP Library and outputs to display.

6.2.15.1 Peripheral devices

This sample project uses the following peripheral devices.

- DRP

6.2.15.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" and "demo_cam_and_disp" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

6.2.15.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_drp_dynamic_sample1" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.15.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the image from the camera is displayed on the HDMI monitor and that edges of the image are detected.

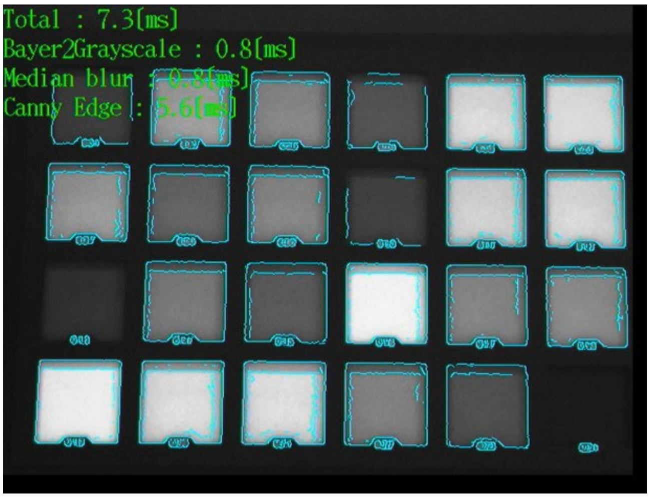

Figure 6-26 Displayed screen of "demo_drp_dynamic_sample1"

6.2.16 demo_drp_dynamic_sample2

The “demo_drp_dynamic_sample2” is a sample project to check the dynamic loading of DRP.

This sample project detects the corners of the input image from MIPI camera by the Harris corner detector using DRP Library and outputs to display.

6.2.16.1 Peripheral devices

This sample project uses the following peripheral devices.

- DRP

6.2.16.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" and "demo_cam_and_disp" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

6.2.16.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_drp_dynamic_sample2" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.16.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the image from the camera is displayed on the HDMI monitor and that corners of the image are detected.

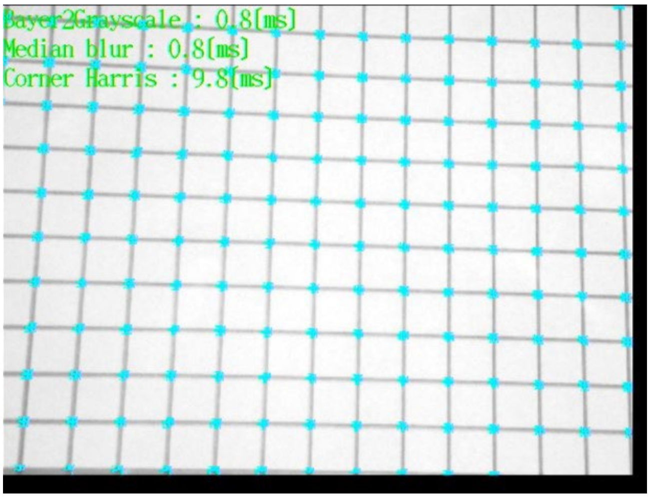

Figure 6-27 Displayed screen of "demo_drp_dynamic_sample2"

6.2.17 demo_drp_dynamic_sample3

The “demo_drp_dynamic_sample3” is a sample project to check the dynamic loading of DRP.

This sample project performs contour detection using find contours application after detecting the edges of the input image from MIPI camera by Canny method using DRP Library. The results are output to display.

6.2.17.1 Peripheral devices

This sample project uses the following peripheral devices.

- DRP

6.2.17.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" and "demo_cam_and_disp" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

6.2.17.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_drp_dynamic_sample3" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.17.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the image from the camera is displayed on the HDMI monitor and that contours of the image are detected. In addition, you can control the Auto Exposure Correction (AE) ON/OFF with SW2 on the board.

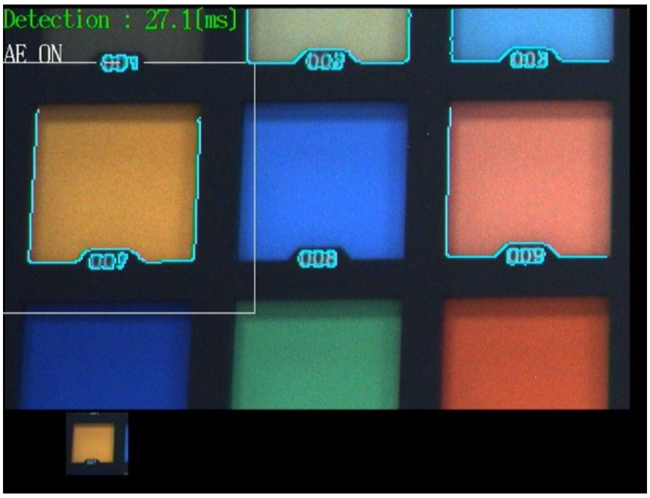

Figure 6-28 Displayed screen of "demo_drp_dynamic_sample3"

6.2.18 demo_drp_parallel_sample

The “demo_drp_parallel_sample” is a sample project to check the parallel operation of DRP.

This sample project converts the input image from MIPI camera to grayscale image at high-speed using parallel operation function of DRP Library and outputs to display.

6.2.18.1 Peripheral devices

This sample project uses the following peripheral devices.

- DRP

6.2.18.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" and "demo_cam_and_disp" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

6.2.18.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_drp_parallel_sample" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.18.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the image from the camera is displayed on the HDMI monitor.

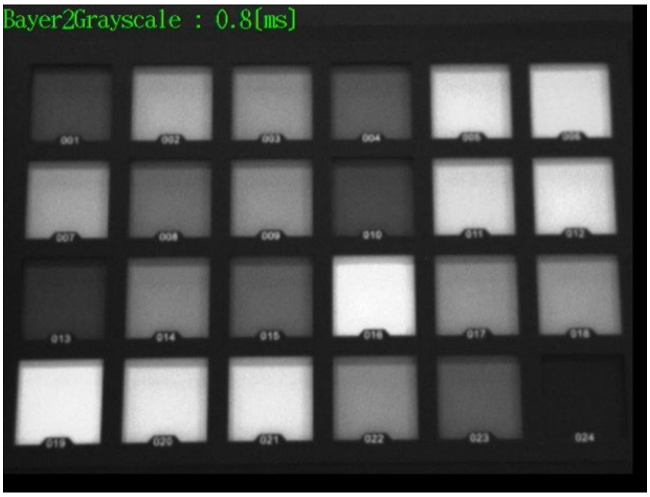

Figure 6-29 Displayed screen of "demo_drp_parallel_sample"

6.2.19 demo_drp_simple_isp_sample1

The “demo_drp_simple_isp_sample1” is a sample project to check the ISP (Image Signal Processor) of DRP.

This sample project corrects the color and noise of the input image from MIPI camera using DRP Library, and the image with high color reproducibility is output to the Display.

6.2.19.1 Peripheral devices

This sample project uses the following peripheral devices.

- DRP

6.2.19.2 How to connect to the target board

Connect the target board to your PC, HDMI monitor, and MIPI camera in the following steps.

Note: Please also refer to "demo_guix_home_automation" and "demo_cam_and_disp" for the connecting.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.
3. Connect the MIPI camera to the CN13 connector on the board.

6.2.19.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_drp_simple_isp_sample1" to be built with in [Project Explorer] of e² studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.19.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the image from the camera is displayed in color on the HDMI monitor. In addition, you can control the color correction mode and the noise reduction ON/OFF with SW2 and SW3 on the board respectively.

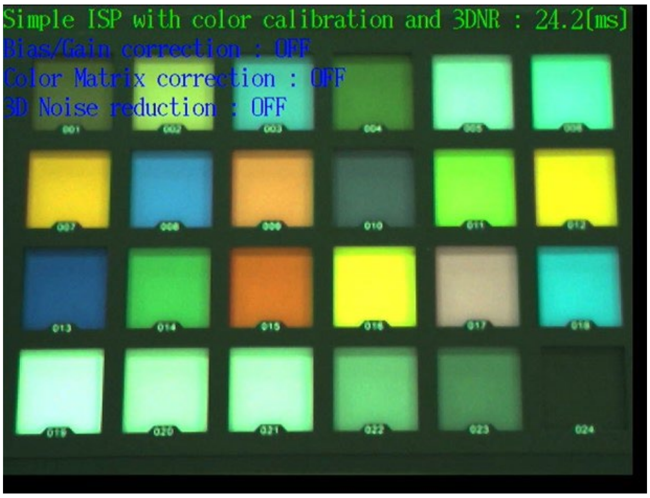

Figure 6-30 Displayed screen of "demo_drp_simple_isp_sample1"

6.2.20 demo_jpeg_codec_sample

The "demo_jpeg_codec_sample" is a sample project that decodes JPEG compressed image and encodes to JPEG compressed image using JCU (JPEG codec unit).

6.2.20.1 Peripheral devices

This sample project uses the following peripheral devices.

- JCU

6.2.20.2 How to connect to the target board

Connect the target board to your PC and HDMI monitor in the following steps.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). This sample project does not use the console.
2. Connect the CN9 connector on the board to the HDMI monitor with an HDMI cable, and also connect the power supply of the HDMI monitor.

6.2.20.3 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "demo_jpeg_codec_sample " to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

6.2.20.4 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that the following screen is displayed on the HDMI monitor.

Figure 6-31 Displayed screen of "demo_jpeg_codec_sample "

7. Azure RTOS IoT Embedded SDK Samples

This package provides sample projects based on Azure RTOS IoT Embedded SDK, which can connect to Azure IoT.

GR-MANGO received "Azure Certified Device" and "IoT Plug and Play" certification.

The following sections describe each sample project.

7.1 sample_azure_iot_embedded_sdk

The "sample_azure_iot_embedded_sdk" is a sample project to connect to Azure IoT Hub using Azure IoT Middleware for Azure RTOS.

7.1.1 How to prepare Azure Resource

Follow the steps below to prepare Azure Resource.

1. Create Azure IoT Hub and register your device (GR-MANGO). Refer to the following document for the details.

   Use the Azure portal to create an IoT Hub | Microsoft Learn

   Note 1: When you create Azure IoT Hub, choose "Free" or "Standard" as "Tier"

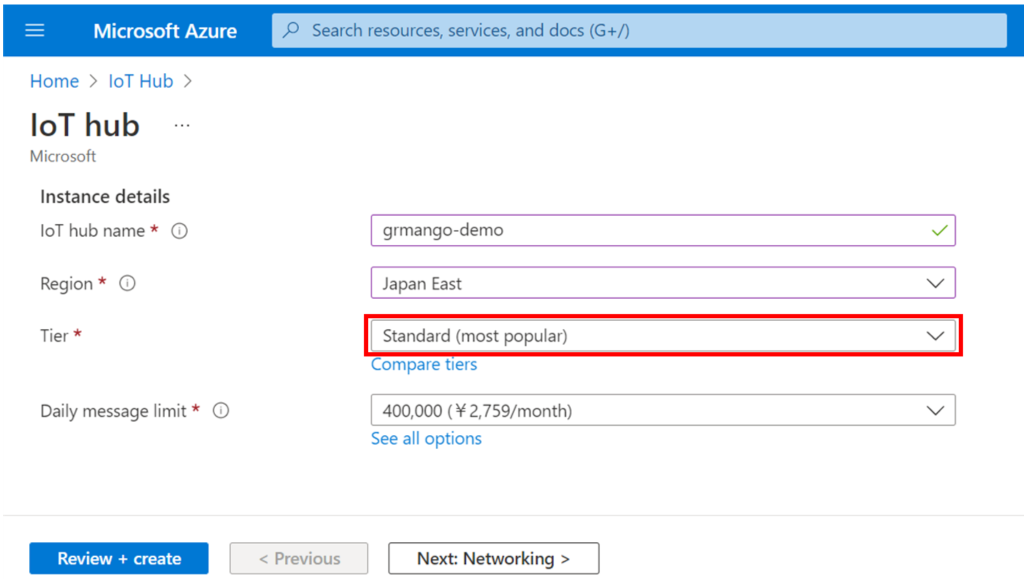

Figure 7-1 Choose "Free tier" or "Standard tier" as "Pricing and scale tier"

Note 2: When you register your device in the IoT Hub, choose “Symmetric key” as “Authentication”.

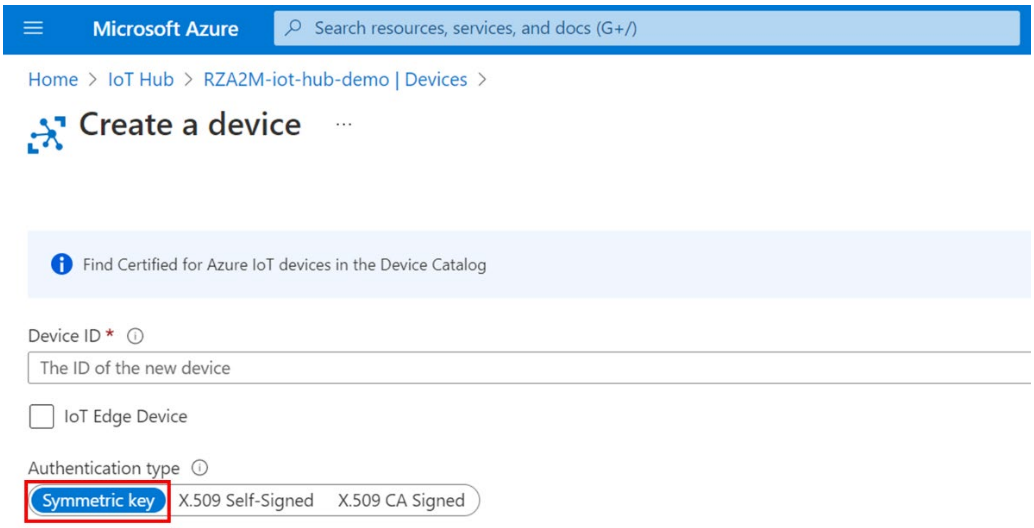

Figure 7-2 Choose “Symmetric key” as “Authentication”

2. Save the following values corresponding to newly registered device. These values are used to connect GR-MANGO to IoT Hub.
   - IoT Hub Hostname
   - Device ID
   - Primary Key

3. Install Azure IoT Explore. Then, get the iothubowner connection string for IoT Hub and register it with IoT Explore. Refer to the following document for the details.

   [Install and use Azure IoT explorer | Microsoft Learn]

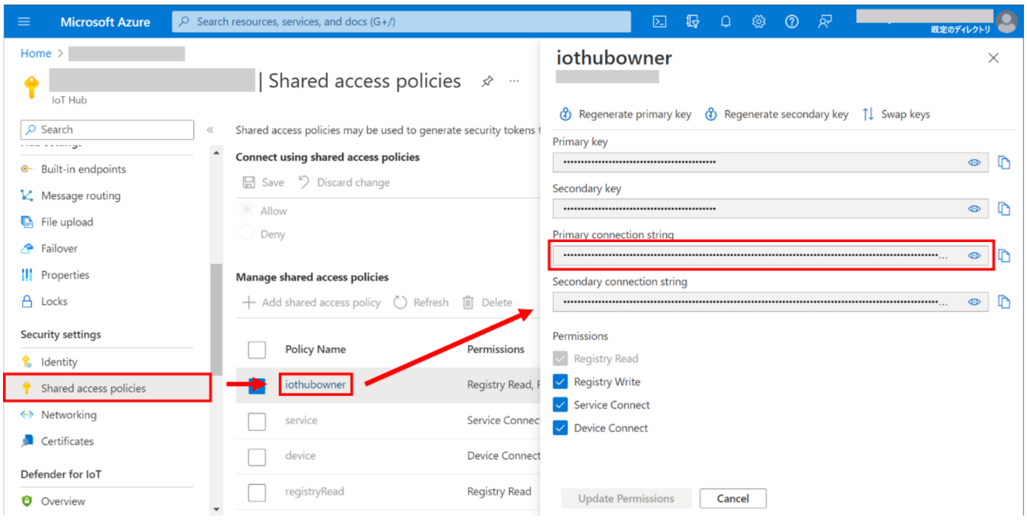

Figure 7-3 iothubowner connection string

7.1.2 How to connect to the target board

Connect the target board to your PC and connect to the internet.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.

2. Connect an Ethernet port to the CN8 connector of the board and connect to the internet.

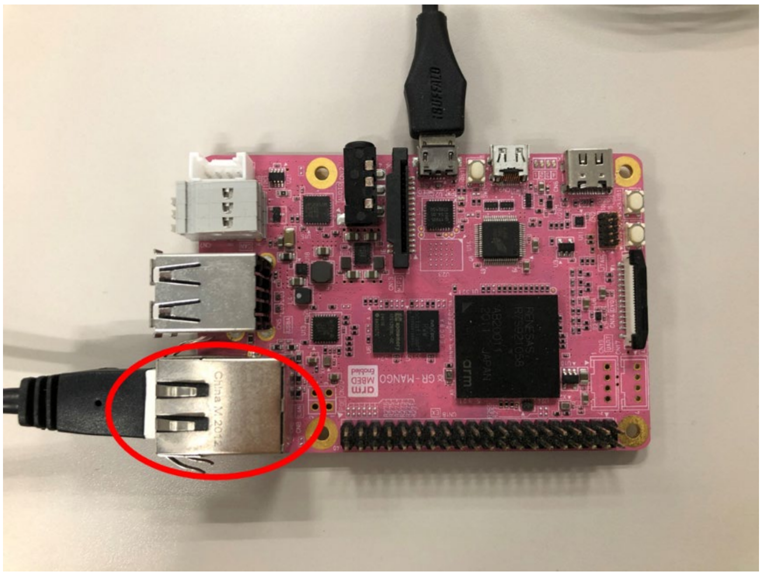

Figure 7-4 Connect to ethernet

7.1.3 How to modify a configuration file

Modify a configuration file to connect the device to Azure IoT.

1. Open the configuration file “sample_config.h” of the sample project “sample_azure_iot_embedded_sdk”.
2. Set the following constants to the values that you saved after you created Azure resources.

Table 7-1 Azure IoT device information constants to be set

Constant name	Value
HOST_NAME	{Your IoT Hub Hostname value}
DEVICE_ID	{Your Device ID value}
DEVICE_SYMMETRIC_KEY	{Your Primary Key value}

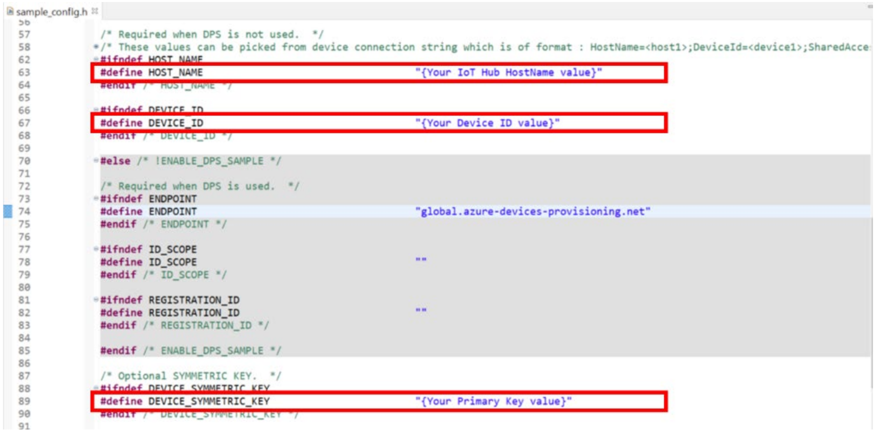

Figure 7-5 Set the values to “sample_config.h”

7.1.4 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "sample_azure_iot_embedded_sdk " to be built with in [Project Explorer] of e[2] studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

7.1.5 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that “Connected to IoT Hub.” is displayed on the terminal software.

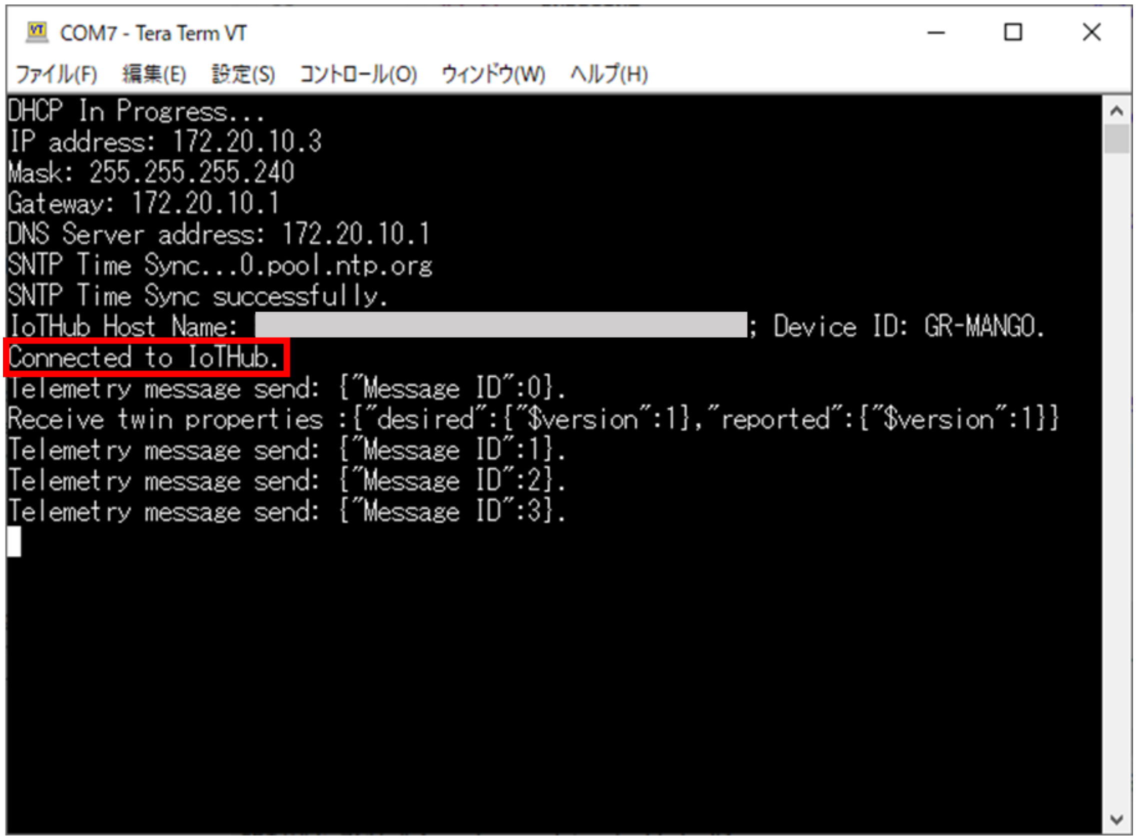

Figure 7-6 Message from "sample_azure_iot_embedded_sdk "

8. Confirm that you can view and manage the properties of your devices using Azure IoT Explorer. This sample project includes following functions.
   - On the “Telemetry” tab, you can view the flow of telemetry from your device to the cloud.

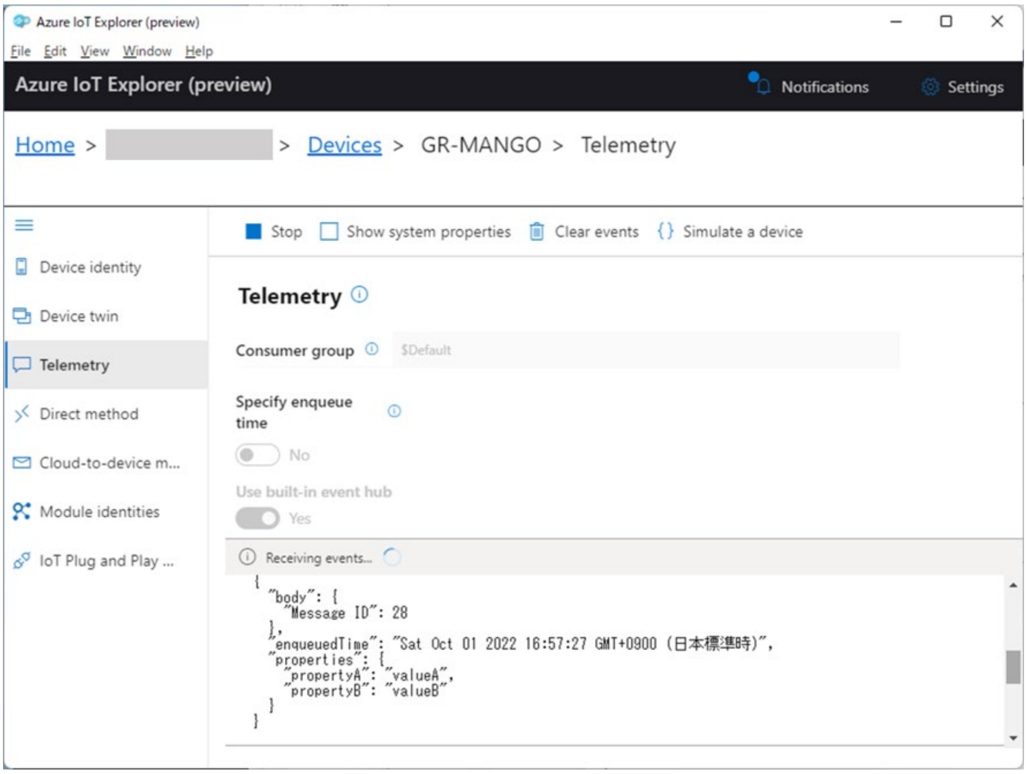

Figure 7-7 View device telemetry

- On the “Device twin” tab, you can update the device property by modifying “desired” section of the device twin. In this sample project, following property can be updated.

```
"weather": {
  "temperature": "25"
},
```

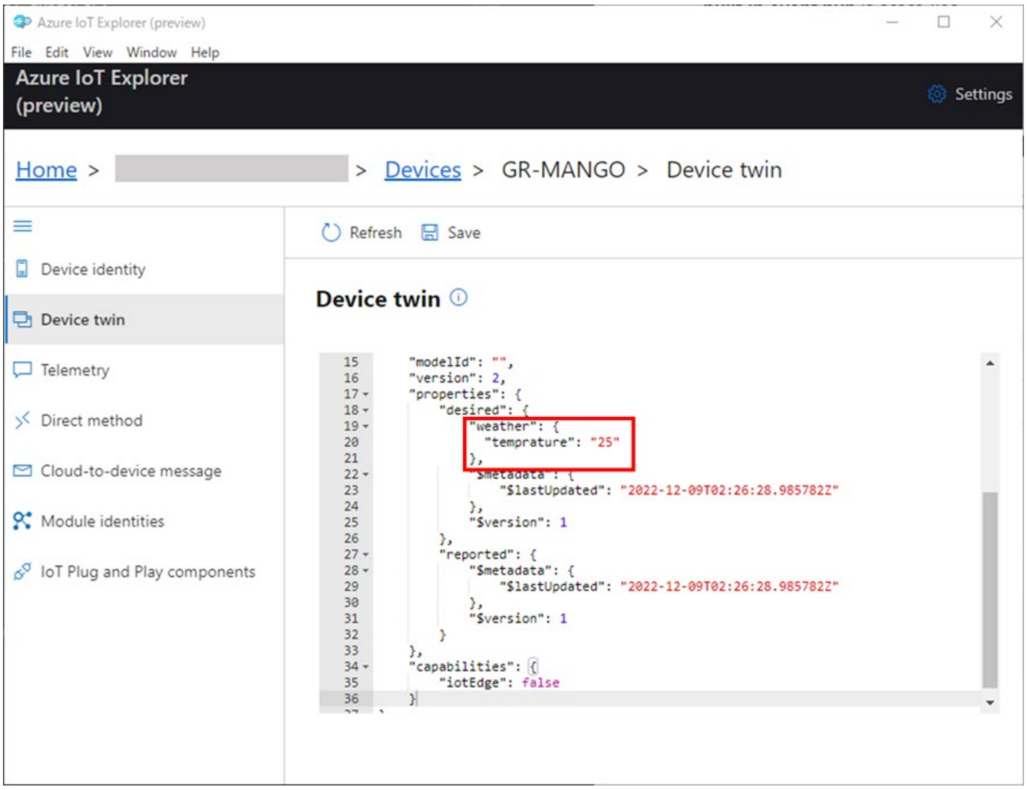

Figure 7-8 Modify device twin

- On the “Direct method” tab, you can call a direct method implemented on the device. In this sample project, following method can be invoked.

  Method name:

  ```
  reboot
  ```

  Payload:

  ```
  {"timeout": 500}
  ```

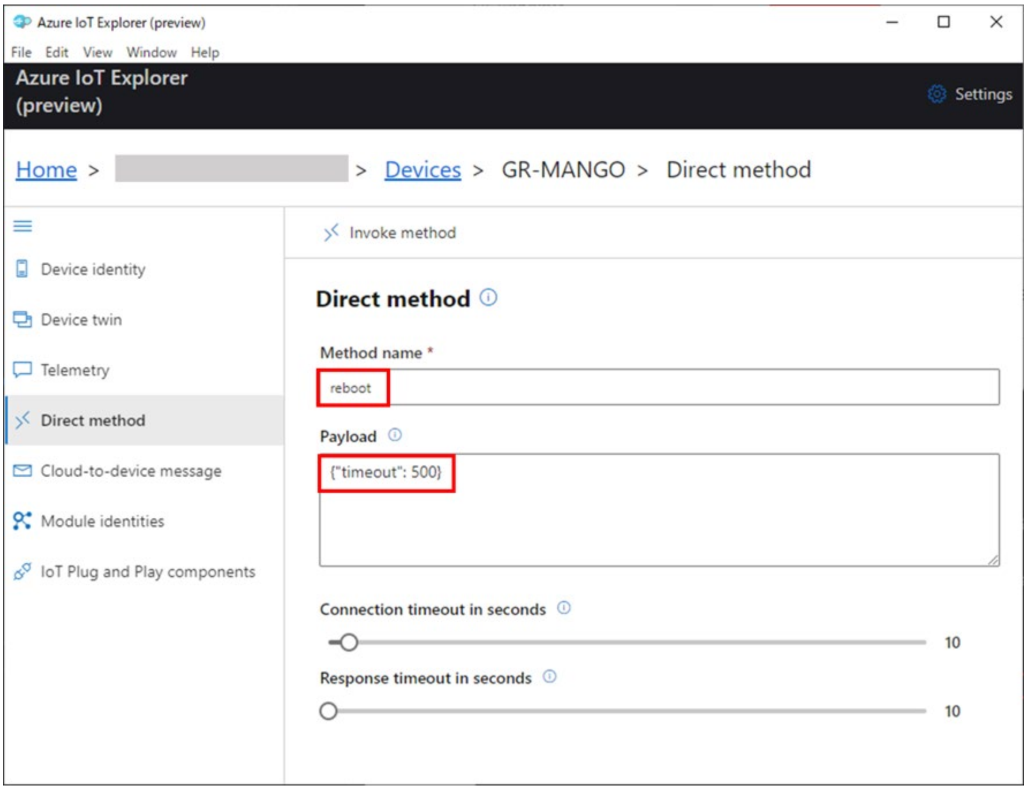

Figure 7-9 Call direct method

- On the “Cloud to device message” tab, you can send cloud to device message.

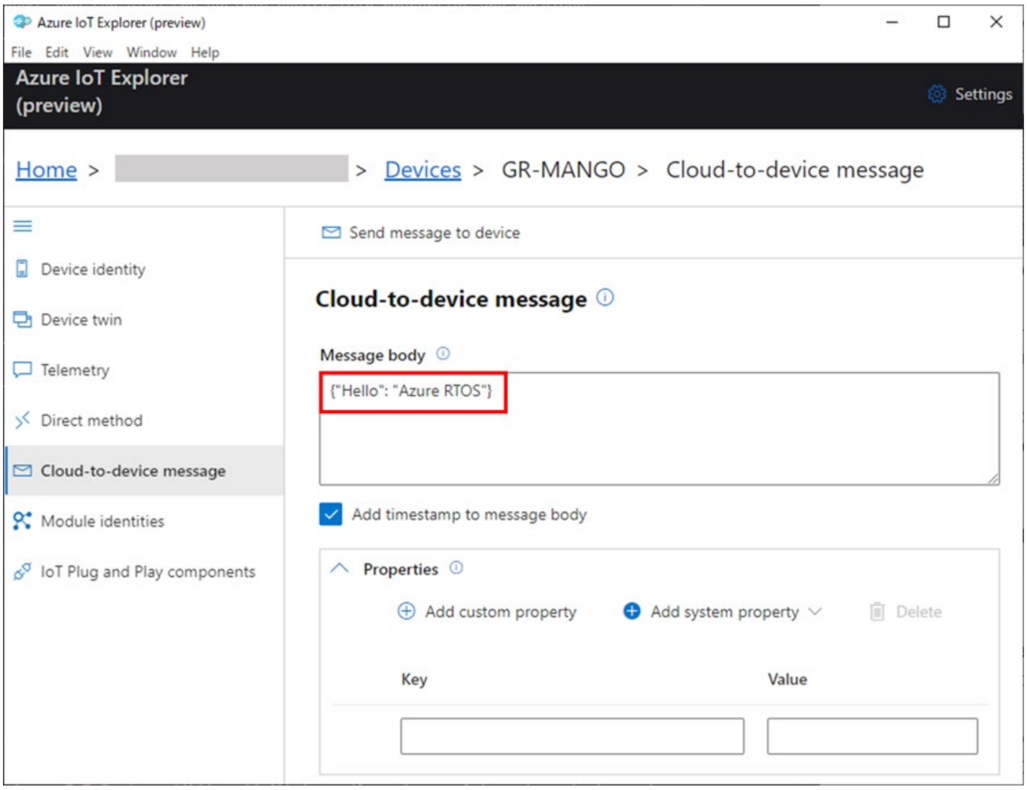

Figure 7-10 Send cloud to device message

7.2 sample_azure_iot_embedded_sdk_pnp

The "sample_azure_iot_embedded_sdk_pnp" is a sample project to connect to Azure IoT Hub using Azure IoT Middleware for Azure RTOS via IoT Plug and Play. Refer to the following document for the details of Plug and Play.

Introduction to IoT Plug and Play | Microsoft Learn

7.2.1 How to prepare Azure Resource

Follow the steps below to prepare Azure Resource.

1. Create an IoT Central application and register your device (GR-MANGO). Refer to the following document for the details.

   Quickstart - Connect a device to an Azure IoT Central application | Microsoft Learn

2. Save the following values generated for newly registered device. These values are used to connect GR-MANGO to IoT Central application.
   - ID Scope
   - Device ID
   - Primary Key

7.2.2 How to connect to the target board

Connect the target board to your PC and connect to the internet.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.
2. Connect an Ethernet port to the CN8 connector of the board and connect to the internet.

7.2.3 How to modify a configuration file

Modify a configuration file to connect the device to Azure IoT.

1. Open the configuration file “sample_config.h” of the sample project “sample_azure_iot_embedded_sdk_pnp”.
2. Set the following constants to the values that you saved after you created Azure resources.

Table 7-2 Azure IoT device information constants to be set

Constant name	Value
ID_SCOPE	{Your ID Scope value}
REGISTRATION_ID	{Your Device ID value}
DEVICE_SYMMETRIC_KEY	{Your Primary Key value}

Figure 7-11 Set the values to “sample_config.h”

7.2.4 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "sample_azure_iot_embedded_sdk_pnp " to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

7.2.5 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that “Connected to IoT Hub.” is displayed on the terminal software.

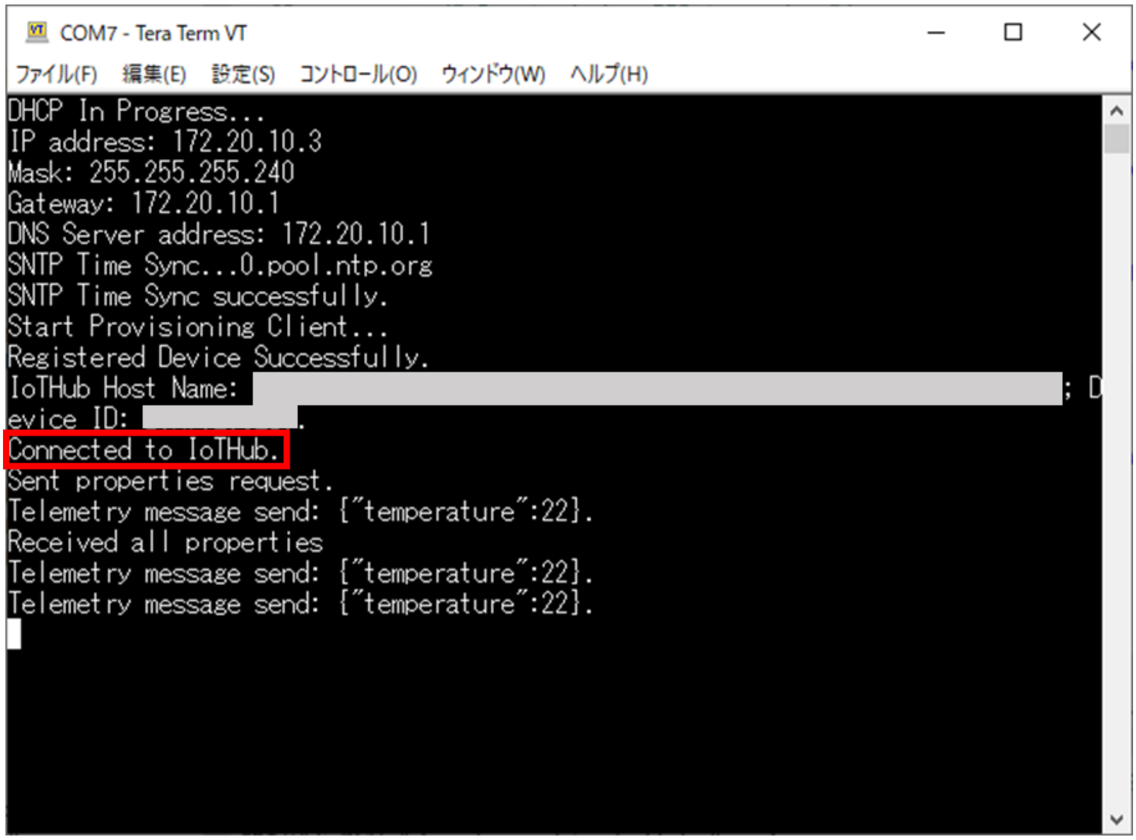

Figure 7-12 Message from "sample_azure_iot_embedded_sdk_pnp "

8. Confirm that you can use Azure IoT Central to interact with IoT Plug and Play components.
   - On the “Overview" tab, you can view a graph of the received data. In this sample project, {“temperature”: 22} is sent as dummy data.

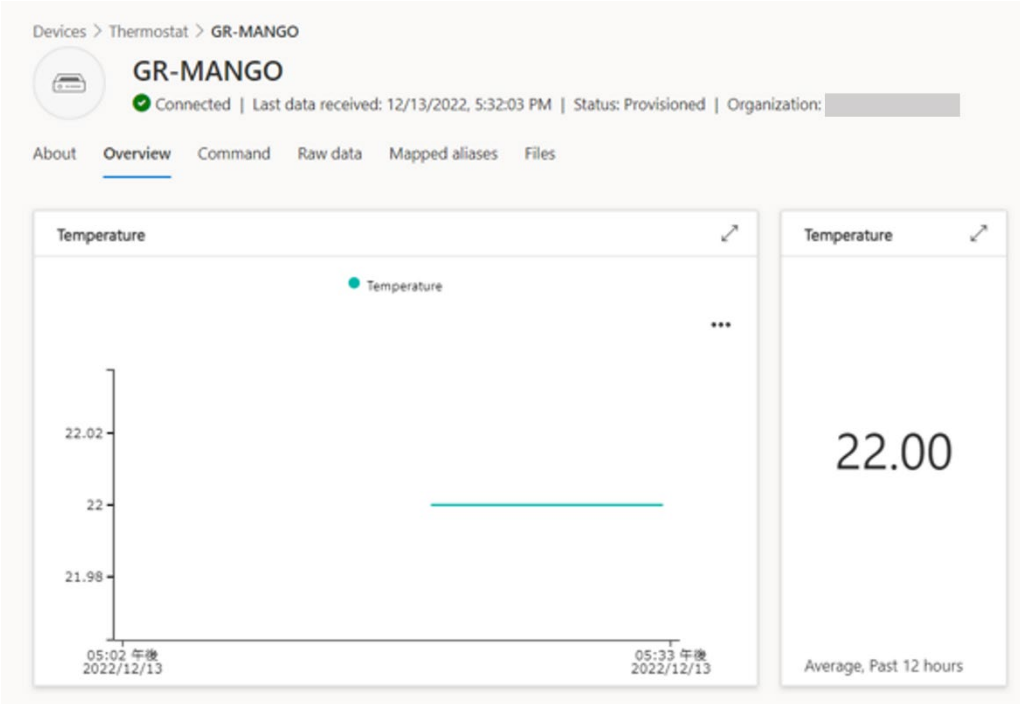

Figure 7-13 Graph of the received data

   - On the “Command” tab, you can send IoT Plug and Play commands to the device.

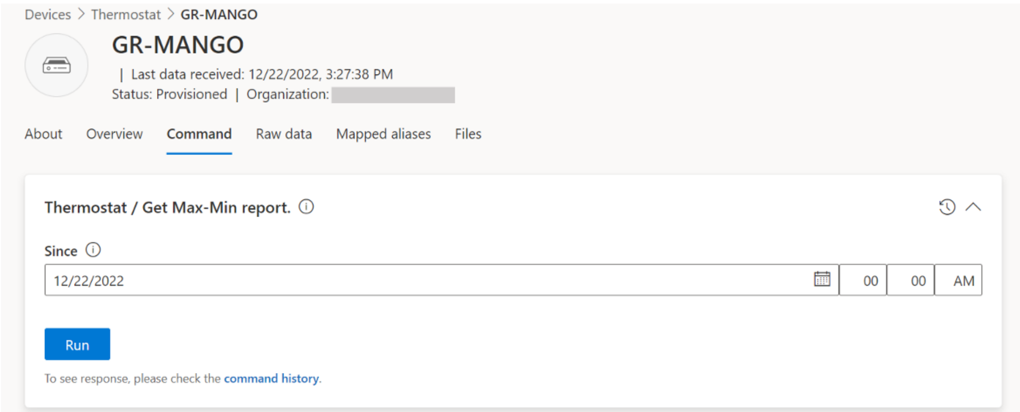

Figure 7-14 Send commands

- On the “Raw data” tab, you can view the raw data such as “Telemetry” or “Command response”.

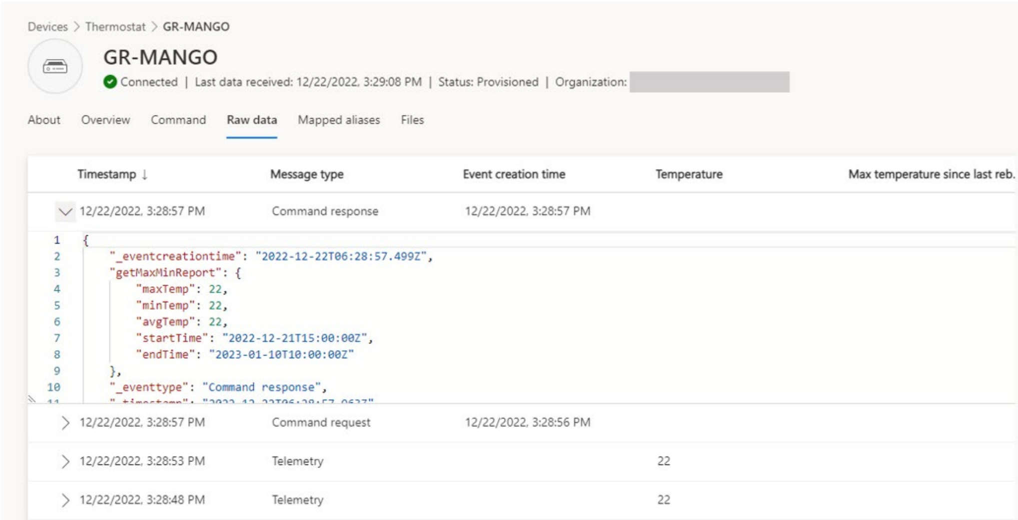

Figure 7-15 View the raw data

7.3 sample_pnp_temperature_controller

The "sample_pnp_temperature_controller" is a sample project with IoT Plug and Play using multiple components. Refer to the following document for the details of Plug and Play.

Introduction to IoT Plug and Play | Microsoft Learn

7.3.1 How to prepare Azure Resource

Follow the steps below to prepare Azure Resource.

1. Create an IoT Central application and register your device (GR-MANGO). Refer to the following document for the details.

   Quickstart - Connect a device to an Azure IoT Central application | Microsoft Learn

2. Save the following values generated for newly registered device. These values are used to connect GR-MANGO to IoT Central application.
   - ID Scope
   - Device ID
   - Primary Key

7.3.2 How to connect to the target board

Connect the target board to your PC and connect to the internet.

1. Connect the CN1 connector of the board and the PC with a USB cable to supply power to the board.

   Note: By connecting to CN1 via USB, it will be recognized by the PC as a USB memory (Mass Storage Class) and UART (CDC ACM). Use the UART as a console.
2. Connect an Ethernet port to the CN8 connector of the board and connect to the internet.

7.3.3 How to modify a configuration file

Modify a configuration file to connect the device to Azure IoT.

1. Open the configuration file “sample_config.h” of the sample project “sample_pnp_temperature_controller”.
2. Set the constants to the values that you saved after you created Azure resources.

   Note: Please refer to "sample_azure_iot_embedded_sdk_pnp" for the modification.

7.3.4 How to build a sample project

Follow the steps below to build the sample project and create an executable file.

1. Select the sample project "sample_pnp_temperature_controller" to be built with in [Project Explorer] of e^2 studio.
2. Make sure the selected project has the [Hardware Debug] build configuration.
3. Right-click on the component project you want to build. Select Build Project from the menu that appears to build the project.
4. Confirm that the project build is completed with "0 errors" (normal completion).

7.3.5 How to execute a sample project

Follow the steps below to write the executable file to the board and execute the program.

1. Open the folder where the executable file is saved in Explorer of the Windows.
2. In another explorer, open the drive that you recognize when you connected the board to your PC.
3. Copy the file with the .bin extension in the folder where the executable file is saved to the drive recognized when you connect the board to your PC.
4. When the copy is completed, the drive recognized by the PC will be closed and recognized as a drive again.
5. Make sure that the re-recognized drive has only two files, "MBED.HTM" and "DETAILS.TXT".

   Note: If the copy fails, the drive will have a "FAIL.TXT" file. Check the Mbed documentation for more details.
6. Unplug the USB cable that connects your PC to the board and reconnect it.
7. Confirm that “Connected to IoT Hub.” is displayed on the terminal software.

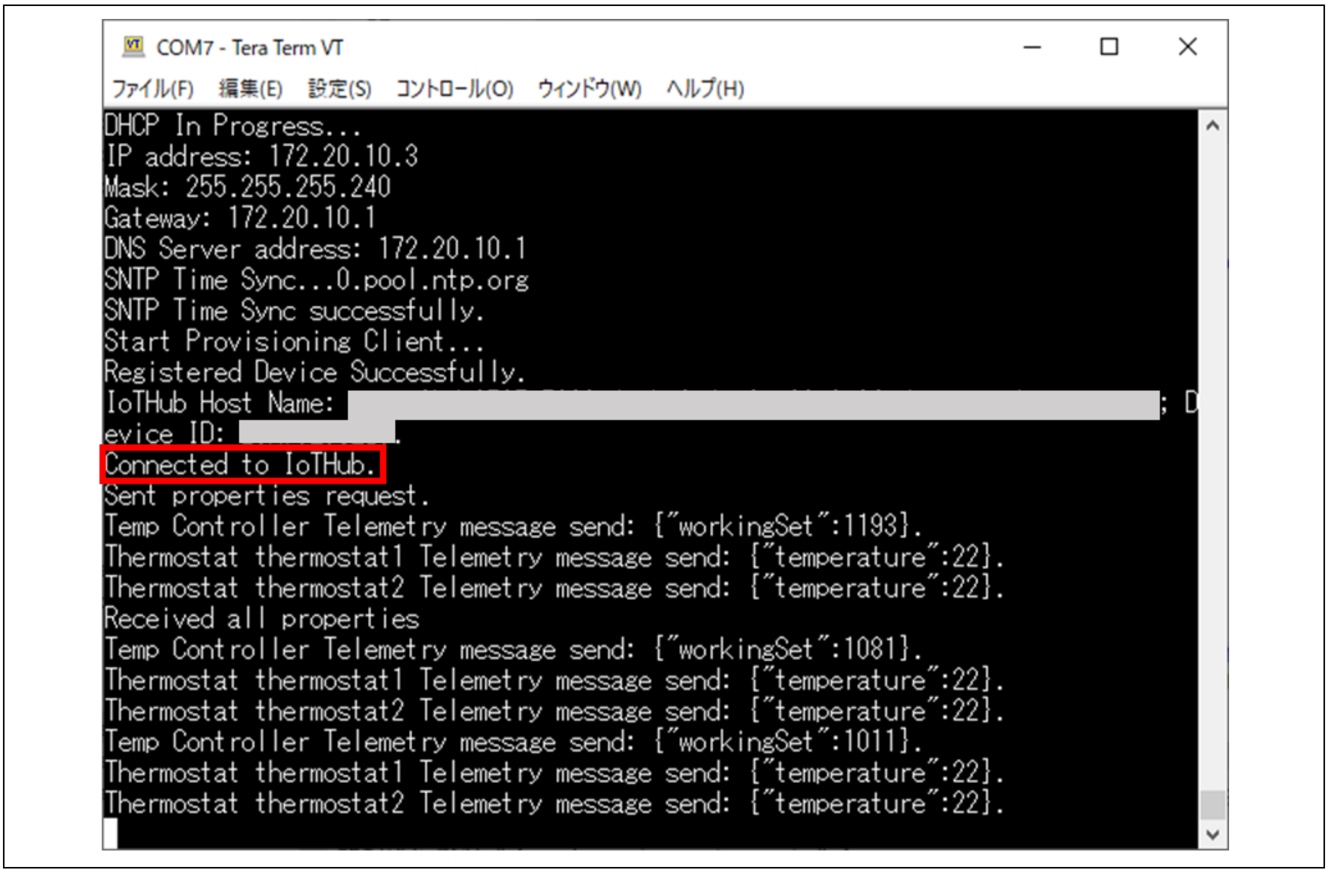

Figure 7-16 Message from "sample_pnp_temperature_controller "

8. Confirm that you can use Azure IoT Central to interact with IoT Plug and Play components.

   Note: Please refer to "sample_azure_iot_embedded_sdk_pnp " for the detail.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 14, 2021	—	First edition issued
1.01	Oct. 4, 2022	—	Add some samples
1.02	Jan. 23, 2023	—	Add “Azure RTOS IoT Embedded SDK Samples” section

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

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/.