RX72M Group

CPU Card EtherNet/IP Startup Manual

Introduction

This application note is a quick start guide for EtherNet/IP communication with the RX72M CPU Card for industrial network evaluation.

This stack runs on E-Force's real-time OS "µC3 (micro-C-cube)" and TCP/IP protocol stack "µNet 3 (micro-net-cube)".

Target Device

RX72M Group

Sample program

This sample program uses the evaluation version of EtherNet/IP stack of our partner: JSL Technology Co., Ltd.

Please contact JSL Technology Corporation for the purchase of the official version and the specifications of the sample program.

JSL Technology Co., Ltd .: https://jslt.co.jp/

【Restrictions】

The evaluation version of the EtherNet/IPIP stack is a library file built with the following functional restrictions, and can only be used for evaluation. It cannot used for integration into the product. In that case, please consider purchasing the official version.

Contents

1.   Operating Environment

The sample program in this manual assumes the following environment.

Table 1.1   Operating Environment

Item	Description
Board	Renesas Electronics RX72M CPU Card
	Model name: RTK0EMXDE0C00000BJ
CPU	RX CPU (RXv3)
	R5F572MNDxBD
Operating frequency	CPU clock (CPUCLK): 240 MHz
Operating voltage	3.3 V
Operating modes	Single chip mode
Device requirements	R5F572MNDDBD
	• Code flash memory
	Capacity: 4 Mbytes
	ROM cache: 8 Kbytes
	• Data flash memory
	Capacity: 32 Kbytes
	• RAM / extended RAM
	Capacity: 512 Kbytes / 512 Kbytes
Communications protocol	EtherNet/IP (Version 3.6.1.0, Tested with ODVA CT19)
Available port	CN8, CN9
Integrated development environment	e2Studio V7.5.0 or later
Tool chain	C/C ++ compiler package V3.01.00 or later for RX family
Emulator (ICE)	Renesas Electronics
	On-board emulator E2 On-Board (E2OB)

1.1   Stack compatible specifications

The supported stack specifications are as follows.

・CIP Networks Library, CIP, Volume 1 (Edition 3.15, November 2022)

・CIP Networks Library, EtherNet/IP, Volume 2 (Edition 1.16, November 2022)

1.2 Supported CIP Object

The following CIP objects are supported.

CIP Objects

0x01 : Identity

0x02 : Message Router

0x04 : Assembly

0x06 : Connection Manager

0xF5 : TCP/IP Interface

0xF6 : Ethernet Link

1.2.1 Identity Object, Class 0x01

The support status of "Identity Object, Class 0x01" is described below.

■Class Attributes

Id	Description	Get	Data Type	Value
01h	Revision	○	-	1
02h	Max Instance	○	-	1
03h	Number of instances	○	-	1

■Class Services

Id	Service
01h	Get_Attributes_All
0Eh	Get_Attribute_Single

■Instance Attributes

Id	Description	Get	Data Type	Value
01h	Vendor Id	○	UINT	0x451(1105) "Renesas Electronics"
02h	Device Type	○	UINT	0x2b (43) "Generic Device"
03h	Product Code	○	UINT	0x65（101）
04h	Revision	○	UINT	3
		○	UINT	5
05h	Status	○	WORD	Bit 0: Owned Bit 1: Reserved, set to 0 Bit 2: Configured Bit 3: Reserved, set to 0 Bit 4-7: See Extended Device Status Bit 8: Minor Recoverable fault Bit 9: Minor Unrecoverable fault Bit 10: Major Recoverable fault Bit 11: Major Unrecoverable fault

					Bit 12-15: Reserved, set to 0 Extended device status (Bit 4-7) 0000 = Self-Testing or Unknown 0001 = Firmware Update in Progress 0010 = At least one faulted I/O connection 0011 = No I/O connections established 0100 = Non-Volatile Configuration bad 0101 = Major Fault 0110 = At least one I/O connection in run mode 0111 = At least one I/O connection established, all in idle mode
06h	Serial Number	○		UDINT	0x00000001 （1）
07h	Product Name	○		SHORT_STRING	0F 45 49 50 53 2d 44 45 4d 4f 2d 52 58 37 32 4d Size: 15Byte "EIPS-DEMO-RX72M"

■Common Services

Id	Implemented		Service
	Class	Instance	
01h	○	○	Get_Attributes_All
05h	-	○	Reset
0Eh	○	○	Get_Attribute_Single

■Reset Services

Parameter	Service
0 (Power Cycle)	Save the current value in non-volatile memory and reset the power supply
1 (Return to Factory Defaults)	Save the default value to non-volatile memory and reset the power supply

1.2.2 Message Router Object, Class 0x02

The support status of "Message Router Object, Class 0x02" is described below.

* Access to Class Attributes and Instance Attributes for this object is not supported.

* Message Router Object is used only to route Explicit messages to other objects.

1.2.3 Assembly Object, Class 0x04

The support status of "Assembly Object, Class 0x04" is described below.

■Class Attributes

Id	Description	Get	Data Type	Value
01h	Revision	○	UINT	2
02h	Max Instance	○	UINT	199
03h	Number of instances	○	UINT	8

■Instance Attributes

Id	Access Rule	Name	Data Type	Value
03h	Set by instance	Data	ARRAY of BYTE	Set by instance

■Instance 101, 103, 104, 106

Id	Access Rule	Name	Data Type	Size	Value
03h	Get	Data	ARRAY of BYTE	8	00 00 00 00 00 00 00 00

■Instance 102, 105

Id	Access Rule	Name	Data Type	Size	Value
03h	Get / Set	Data	ARRAY of BYTE	8	00 00 00 00 00 00 00 00

■Instance 198, 199

Id	Access Rule	Name	Data Type	Size	Value
03h	Get / Set	Data	ARRAY of BYTE	0	

■Common Services

Id	Implemented		Service
	Class	Instance	
0Eh	○	○	Get_Attributes_Single
10h	-	○	Set_Attribute_Single

1.2.4 Connection Manager Object, Class 0x06

The support status of "Connection Manager Object, Class 0x06" is described below.

■Class Attributes

Id	Description	Get	Data Type	Value
01h	Revision	○	UINT	1
02h	Max Instance	○	UINT	1
03h	Number of instances	○	UINT	1

■Instance Attributes

Id	Description	Get	Data Type	Value
01h	Open Requests	○	UINT	0
02h	Open Format Rejects	○	UINT	0
03h	Open Resource Rejects	○	UINT	0
04h	Open Other Rejects	○	UINT	0
05h	Close Requests	○	UINT	0
06h	Close Format Requests	○	UINT	0
07h	Close Other Requests	○	UINT	0
08h	Connection Timeouts	○	UINT	0

■Common Services

Id	Implemented		Service
	Class	Instance	
01h	○	○	Get_Attributes_All
0Eh	○	○	Get_Attribute_Single

■Object-Specific Services

Id	Implemented		Service
	Class	Instance	
4Eh	-	○	Forward_Close
54h	-	○	Forward_Open
5Bh	-	○	Large_Forward_Open

1.2.5 TCP/IP Interface Object, Class 0xF5

The support status of "TCP/IP Interface Object, Class 0xF5" is described below.

■Class Attributes

Id	Description	Get	Data Type	Value
01h	Revision	○	UINT	4
02h	Max Instance	○	UINT	1
03h	Number of instances	○	UINT	1

■Instance Attributes

Id	Description	Get	Data Type	Value
01h	Status	○	UINT	2
02h	Configuration Capability	○	UINT	32
03h	Configuration Control	○	UINT	0
04h	Physical Link	○	STRUCT of:	-
	Path size		UINT	2
	Path		Padded EPATH	20 F6 24 01
05h	Interface Configuration	○	STRUCT of:	0
	IP Address		UDINT	0A 01 A8 C0 "192.168.1.10"
	Network Mask		UDINT	00 FF FF FF "255.255.255.0"
	Gateway Address		UDINT	00 00 00 00 "0.0.0.0"
	Name Server		UDINT	00 00 00 00 "0.0.0.0"
	Name Server2		UDINT	00 00 00 00 "0.0.0.0"
	Domain Name		STRING	00 00
06h	Host Name	○	STRING	0
08h	TTL Value	○	USINT	1
09h	MCast Config	○	STRUCT of:	-
	Alloc Control		USINT	0
	Reserved		USINT	0
	Mun Mcast		UINT	32
	Mcast Start Addr		UDINT	20 02 C0 EF "239.192.2.32"
0Ah	SelectAcd	Get/Set	BOOL	1

0Bh	LastConflictDetected	Get/Set	STRUCT of:	
	AcdActivity		USINT	
	RemoteMAC		Array of 6 USINT	00 00 00 00 00 00 Mac address "00:00:00:00:00:00"
	ArpPdu		ARRAY of 28 USINT	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0Dh		Get/Set		

■Common Services

Id	Implemented		Service
	Class	Instance	
01h	○	○	Get_Attributes_All
0Eh	○	○	Get_Attribute_Single
10h	-	○	Set_Attribute_Single

1.2.6 EtherNet Link Object, Class 0xF6

The support status of "EtherNet Link Object, Class 0xF6"" is described below.

■Class Attributes

Id	Description	Get	Data Type	Value
01h	Revision	○	UINT	4
02h	Max Instance	○	UINT	3
03h	Number of instances	○	UINT	3

■Instance Attributes

Id	Description	Get	Data Type	Value
01h	Interface Speed	○	UDINT	Set by instance
02h	Interface Flags	○	DWORD	Set by instance
03h	Physical Address	○	Array of 6 USINT	Set by instance
04h	Interface Counters	○	STRUCT of:	-
	In Octets		UDINT	Set by instance
	In Ucast Packets		UDINT	Set by instance
	In NUcast Packets		UDINT	Set by instance
	In Discards		UDINT	Set by instance
	In Errors		UDINT	Set by instance
	In Unknown Protos		UDINT	Set by instance
	Out Octets		UDINT	Set by instance
	Out Ucast Packets		UDINT	Set by instance
	Out NUcast Packets		UDINT	Set by instance
	Out Discards		UDINT	Set by instance
	Out Errors		UDINT	Set by instance
05h	Media Counters	○	STRUCT of:	-
	Alignment Errors		UDINT	Set by instance
	FCS Errors		UDINT	Set by instance
	Single Collisions		UDINT	Set by instance
	Multiple Collisions		UDINT	Set by instance
	SQE Test Errors		UDINT	Set by instance
	Deferred Transmissions		UDINT	Set by instance
	Late Collisions		UDINT	Set by instance
	Excessive Collisions		UDINT	Set by instance
	MAC Transmit Errors		UDINT	Set by instance
	Carrier Sense Errors		UDINT	Set by instance

RENESAS

Id	Description	Get	Data Type	Value
	Frame Too Long		UDINT	Set by instance
	MAC Receive Errors		UDINT	Set by instance
07h	Interface Type	○	USINT	Set by instance
08h	Interface State	○	USINT	Set by instance
0Ah	Interface Label	○	SHORT_STRING	Set by instance
0Bh	Interface Capability	○	STRUCT of:	-
	Capability Bits		DWORD	Set by instance
	Speed/Duplex Options		STRUCT of:	-
	Speed/Duplex Array Count		USINT	Set by instance
	Speed/Duplex Array		ARRAY of:	-
	Speed/Duplex Pair		STRUCT of:	-
	Interface Speed		UINT	Set by instance
	Interface Duplex Mode		USINT	Set by instance
	Interface Speed		UINT	Set by instance
	Interface Duplex Mode		USINT	Set by instance
	Interface Speed		UINT	Set by instance
	Interface Duplex Mode		USINT	Set by instance
	Interface Speed		UINT	Set by instance
	Interface Duplex		USINT	Set by instance

■Instance 1 (Port 1)

Id	Description	Get	Data Type	Value
01h	Interface Speed	○	UDINT	0x00000064 (100) 100Mbps
02h	Interface Flags	○	DWORD	0x0000000F (15) "Successfully negotiated speed and duplex"
03h	Physical Address	○	Array of 6 USINT	12 34 56 78 36 15 Mac address "12:34:56:78:36:15"
04h	Interface Counters	○	STRUCT of:	-
	In Octets		UDINT	0
	In Ucast Packets		UDINT	0
	In NUcast Packets		UDINT	0
	In Discards		UDINT	0
	In Errors		UDINT	0
	In Unknown Protos		UDINT	0
	Out Octets		UDINT	0
	Out Ucast Packets		UDINT	0
	Out NUcast Packets		UDINT	0

RENESAS

	Out Discards		UDINT	0
	Out Errors		UDINT	0
05h	Media Counters	◯	STRUCT of:	-
	Alignment Errors		UDINT	0
	FCS Errors		UDINT	0
	Single Collisions		UDINT	0
	Multiple Collisions		UDINT	0
	SQE Test Errors		UDINT	0
	Deferred Transmissions		UDINT	0
	Late Collisions		UDINT	0
	Excessive Collisions		UDINT	0
	MAC Transmit Errors		UDINT	0
	Carrier Sense Errors		UDINT	0
	Frame Too Long		UDINT	0
	MAC Receive Errors		UDINT	0
07h	Interface Type	◯	USINT	0x02 (2) "Twisted-pair"
08h	Interface State	◯	USINT	0x01 (1) "The interface is enabled"
0Ah	Interface Label	◯	SHORT_STRING	06 50 6f 72 74 20 31 Size: 6Byte "Port 1"
0Bh	Interface Capability	◯	STRUCT of:	-
	Capability Bits		DWORD	0x00000006 Auto-negotiate, Auto-MDIX
	Speed/Duplex Options		STRUCT of:	-
	Speed/Duplex Array Count		USINT	4
	Speed/Duplex Array		ARRAY of:	-
	Speed/Duplex Pair		STRUCT of:	-
	Interface Speed		UINT	10
	Interface Duplex Mode		USINT	0
	Interface Speed		UINT	10
	Interface Duplex Mode		USINT	1
	Interface Speed		UINT	100
	Interface Duplex Mode		USINT	0
	Interface Speed		UINT	100
	Interface Duplex		USINT	1

■Instance 2 (Port 2)

Id	Description	Get	Data Type	Value
01h	Interface Speed	○	UDINT	0x00000064 (100) 100Mbps
02h	Interface Flags	○	DWORD	0x0000000F (15) "Successfully negotiated speed and duplex"
03h	Physical Address	○	Array of 6 USINT	12 34 56 78 36 15 Mac address "12:34:56:78:36:15"
04h	Interface Counters	○	STRUCT of:	-
	In Octets		UDINT	0
	In Ucast Packets		UDINT	0
	In NUcast Packets		UDINT	0
	In Discards		UDINT	0
	In Errors		UDINT	0
	In Unknown Protos		UDINT	0
	Out Octets		UDINT	0
	Out Ucast Packets		UDINT	0
	Out NUcast Packets		UDINT	0
	Out Discards		UDINT	0
	Out Errors		UDINT	0
05h	Media Counters	○	STRUCT of:	-
	Alignment Errors		UDINT	0
	FCS Errors		UDINT	0
	Single Collisions		UDINT	0
	Multiple Collisions		UDINT	0
	SQE Test Errors		UDINT	0
	Deferred Transmissions		UDINT	0
	Late Collisions		UDINT	0
	Excessive Collisions		UDINT	0
	MAC Transmit Errors		UDINT	0
	Carrier Sense Errors		UDINT	0
	Frame Too Long		UDINT	0
	MAC Receive Errors		UDINT	0
07h	Interface Type	○	USINT	0x02 (2) "Twisted-pair"
08h	Interface State	○	USINT	0x01 (1) "The interface is enabled"
0Ah	Interface Label	○	SHORT_STRING	06 50 6f 72 74 20 32 Size: 6Byte "Port 2"
0Bh	Interface Capability		STRUCT of:	-

Capability Bits	○	DWORD	0x00000006 Auto-negotiate, Auto-MDIX	
Speed/Duplex Options		STRUCT of:	-	
Speed/Duplex Array Count		USINT	4	
Speed/Duplex Array		ARRAY of:	-	
Speed/Duplex Pair		STRUCT of:	-	
Interface Speed		UINT	10	
Interface Duplex Mode		USINT	0	
Interface Speed		UINT	10	
Interface Duplex Mode		USINT	1	
Interface Speed		UINT	100	
Interface Duplex Mode		USINT	0	
Interface Speed		UINT	100	
Interface Duplex		USINT	1	

■Instance 3 (internal)

Id	Description	Get	Data Type	Value
01h	Interface Speed	○	UDINT	0x00000000 (0) 0Mbps
02h	Interface Flags	○	DWORD	0x00000010 (16) "Successfully negotiated speed and duplex"
03h	Physical Address	○	Array of 6 USINT	12 34 56 78 36 15 Mac address "12:34:56:78:36:15"
04h	Interface Counters	○	STRUCT of:	-
	In Octets		UDINT	0
	In Ucast Packets		UDINT	0
	In NUcast Packets		UDINT	0
	In Discards		UDINT	0
	In Errors		UDINT	0
	In Unknown Protos		UDINT	0
	Out Octets		UDINT	0
	Out Ucast Packets		UDINT	0
	Out NUcast Packets		UDINT	0
	Out Discards		UDINT	0
	Out Errors		UDINT	0
05h	Media Counters	○	STRUCT of:	-
	Alignment Errors		UDINT	0
	FCS Errors		UDINT	0
	Single Collisions		UDINT	0

	Multiple Collisions		UDINT	0
	SQE Test Errors		UDINT	0
	Deferred Transmissions		UDINT	0
	Late Collisions		UDINT	0
	Excessive Collisions		UDINT	0
	MAC Transmit Errors		UDINT	0
	Carrier Sense Errors		UDINT	0
	Frame Too Long		UDINT	0
	MAC Receive Errors		UDINT	0
07h	Interface Type	○	USINT	0x01 (1) "The interface is internal to the device"
08h	Interface State	○	USINT	0x01 (1) "The interface is enabled"
0Ah	Interface Label	○	SHORT_STRING	08 69 6e 74 65 72 6e 61 6c Size: 8Byte "internal"
0Bh	Interface Capability	○	STRUCT of:	-
	Capability Bits		DWORD	0x00000006 Auto-negotiate, Auto-MDIX
	Speed/Duplex Options		STRUCT of:	-
	Speed/Duplex Array Count		USINT	4
	Speed/Duplex Array		ARRAY of:	-
	Speed/Duplex Pair		STRUCT of:	-
	Interface Speed		UINT	10
	Interface Duplex Mode		USINT	0
	Interface Speed		UINT	10
	Interface Duplex Mode		USINT	1
	Interface Speed		UINT	100
	Interface Duplex Mode		USINT	0
	Interface Speed		UINT	100
	Interface Duplex		USINT	1

■Common Services

Id	Implemented		Service
	Class	Instance	
01h	○	○	Get_Attributes_All
0Eh	○	○	Get_Attribute_Single
10h	-	○	Set_Attribute_Single

■Object-Specific Services

Id	Implemented		Service
	Class	Instance	
4Ch	-	○	Get_and_Clear

Get_and_Clear corresponds to the following Instance Attribute.

・ 0x04: Interface Counters

・ 0x05: Media Counters

The value of each attribute is reset to 0 after the service is executed.

1.3 **Sample program folder structure**

The release folder structure is described below.

[eip_adpt]

  |

  |-[lib]：library

  |

  |-[prj_rx72m_e]：e² studio folder

  |

  |-[src]：Source folder

    |-[app]

    | |-[rx72m]

    | |-[uC3]：uC3 main, driver

    | |-[odva]：EDS/STC file

    | | |-[eds]：EDS file　"Renesas RX72M EIP Adapter Sample.eds"

    | | |-[stc]：Structure file for CT19 test "Renesas EIP RX72M Sample.stc"

2.   Setting up and Connecting the Evaluation Board

For detailed information on this board, refer to the "RX72M CPU Card with RDC-IC User's Manual" (R12UZ0098EJ0100).
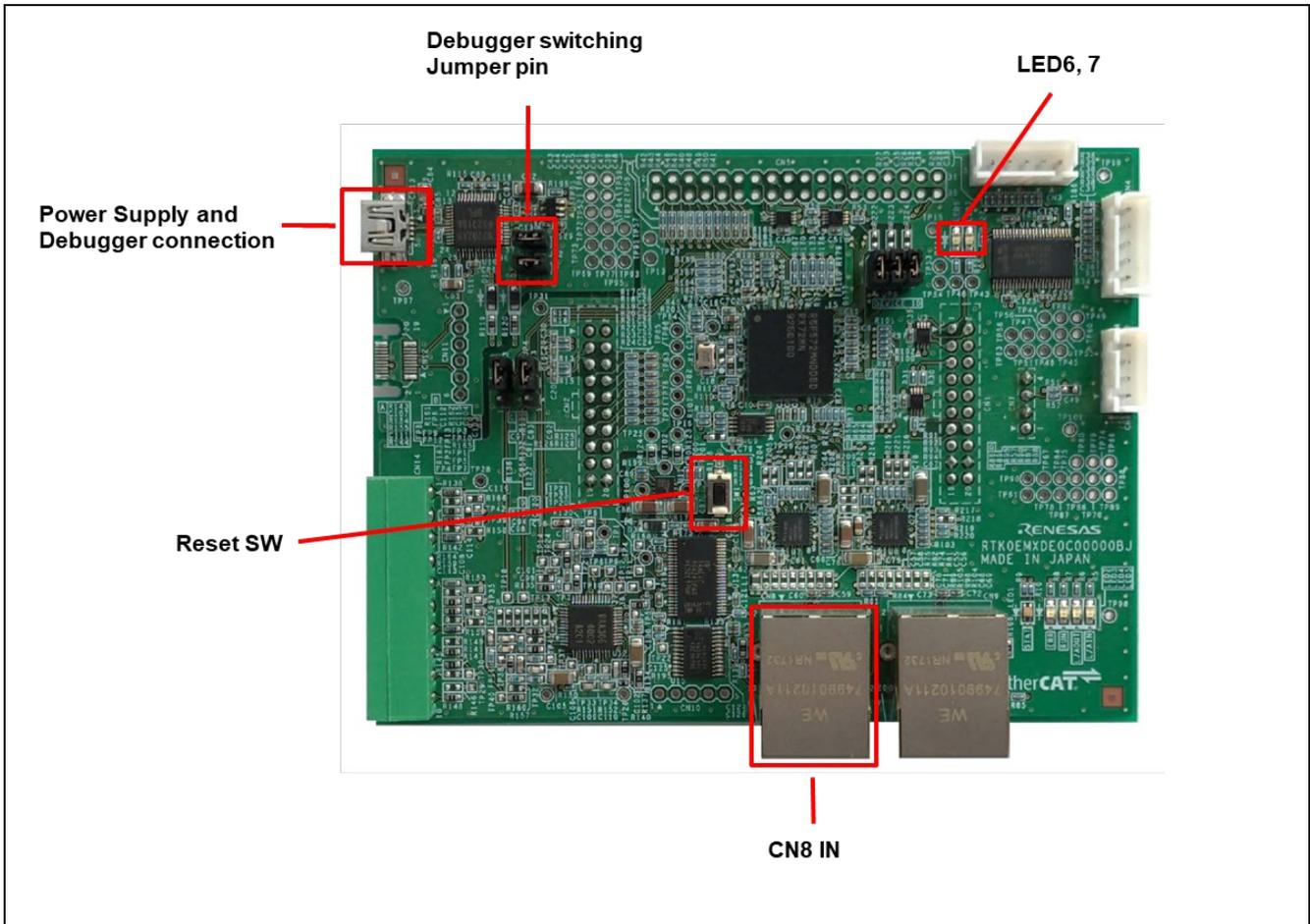
Figure 2.1   Configuration of the RX72M CPU Card

2.1   CPU Card setting

Set the jumper pins before powering on the CPU Card.

Using E2OB   : Short the Debugger switching Jumper pin.

2.2   Power supply selection

CPU Card does not have DC Jack, input DC5V from USB connector.

3.   Installing the e² studio

Download RX72M compatible e2studio (V7.5.0 or later) from the following website.

   https://www.renesas.com/e2studio_download

3.1   Installing the CC-RX Compiler V3.01.00

The compiler selection screen appears while installing e2studio. By selecting [Renesas CCRX v3.0 1.00] and selecting [Next], CC-RX V3.0 1.00 compiler compatible with RX72M will be installed together.
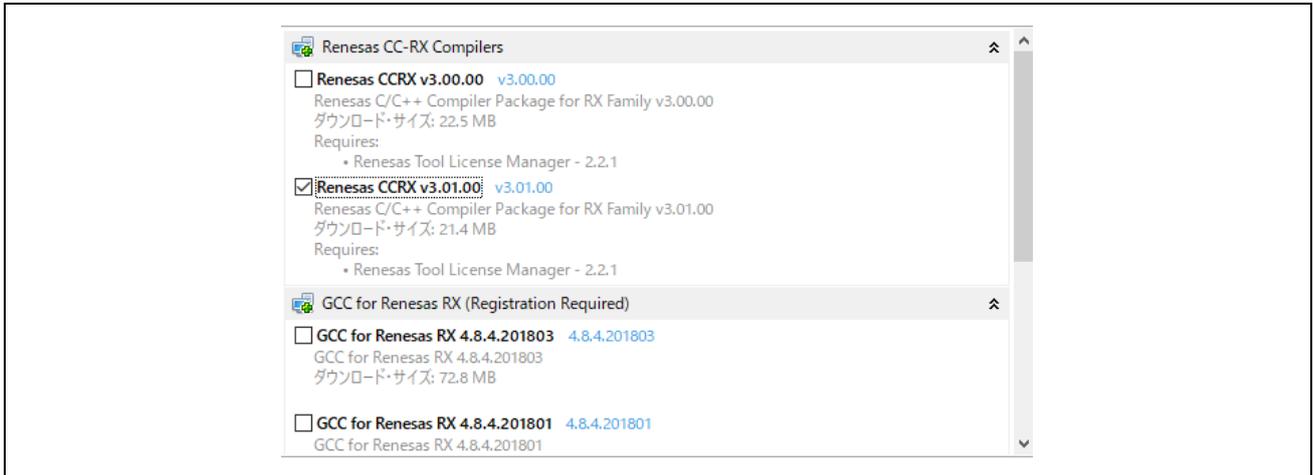
To start e2studio, please run "e2studio.exe" located in the installed folder below.

   e2_studio_rx72m\eclipse

3.2   Registering the Tool Chain

Register the CC-RX compiler v3.01.00 so that it can be used with the e² studio for RX72M.

(1)   Start the e² studio for RX72M.

(2)   Select [File] → [New] → [C/C++Project] → [Next].
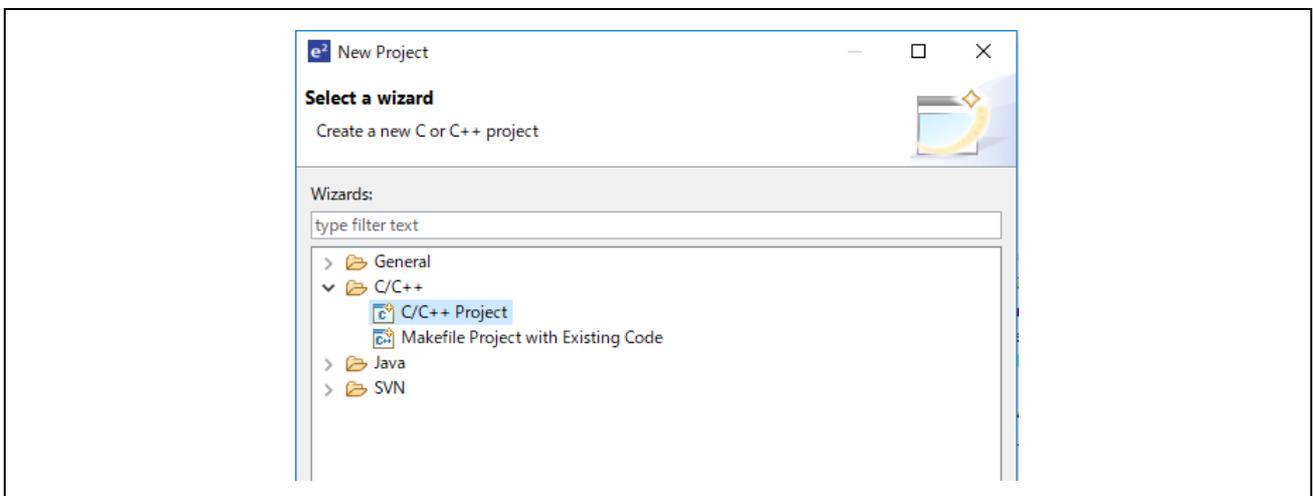
(3) In the [Templates for New C/C++ Project] dialog box, select [Renesas RX] → [Renesas CC-RX C/C++ Executable Project] → [Next].
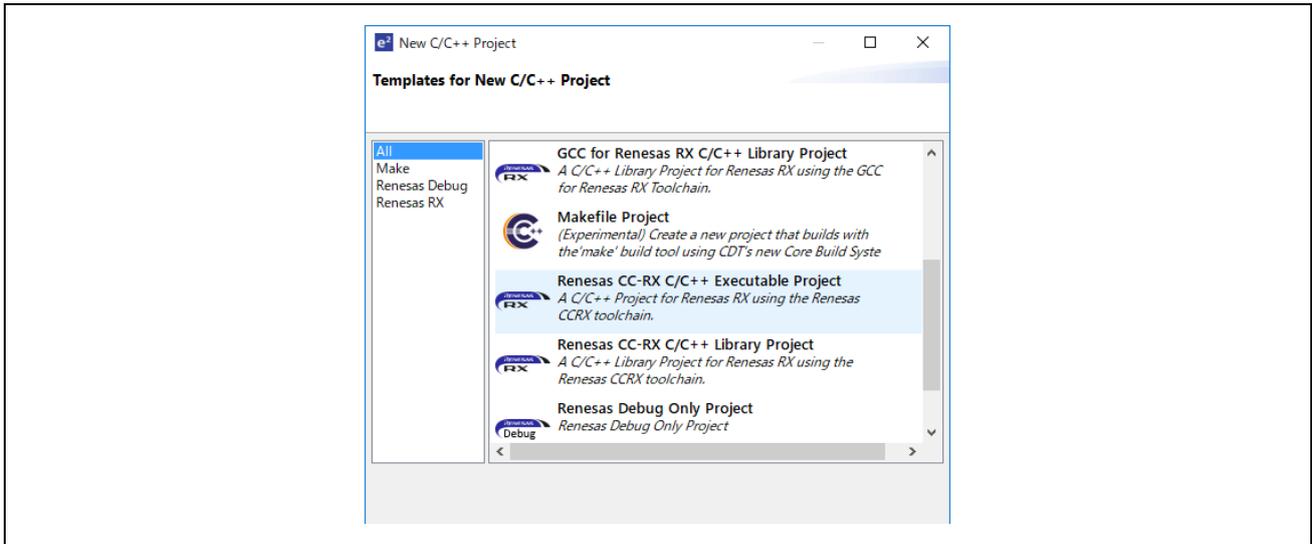
(4) In the [New Renesas CC-RX C/C++ Executable Project] dialog box, enter a desired project name and select [Next].

(5) In the [Select toolchain, device & debug settings] dialog box, select [Toolchain Management] under [Toolchain Settings].

(6) In the [Renesas Toolchain Management] dialog box, select [Add] → [Browse...] to refer to the installation folder "C:¥Renesas¥RX¥3_0_1".
The registration was successful if "v3.01.00 has been added under "Renesas CCRX".
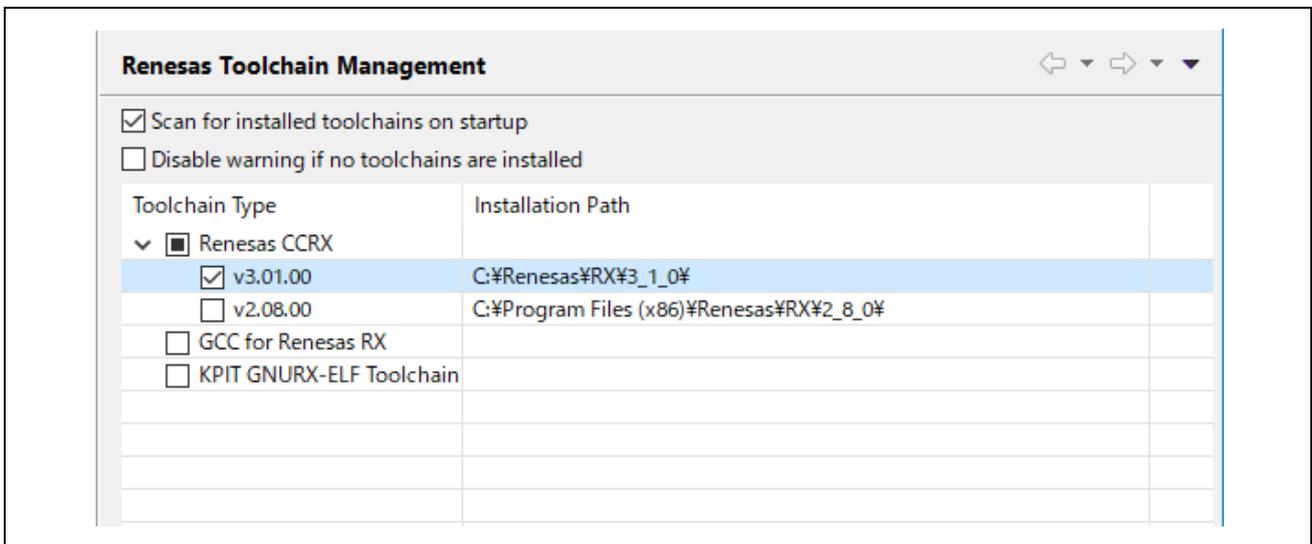
4. Evaluation environment construction method

The following describes method to build a development environment.

4.1 Preparing the evaluation environment

(1) Preparing the environment folder
    Release environment (unzid download folder) in an optional folder.

    ex：C:\proj\eip_adp

(2) EDS File
    Use the EDS file in the following folder for the EDS file.eip_adpt\src\app\rx72m\uc3\odva\eds

    ・Renesas RX72M EIP Adapter Sample.eds

4.2 Execution procedure

Describes the procedure to execute communication in the sample application.

(1) After starting e2studio, click "File"-> "Import".

(2) In the "Select" dialog, select "General" → "To an existing project to workspace" and click "Next".
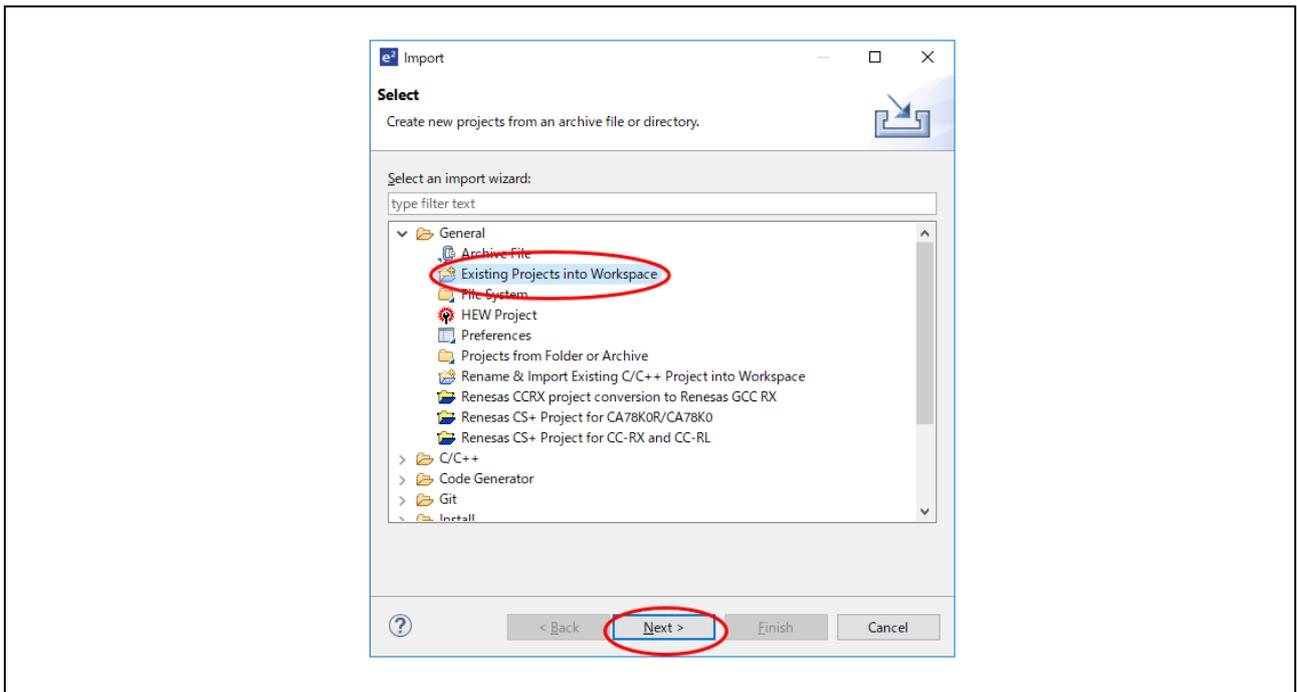
(3)   Select the "Select root directory" check box in the "Import project" dialog and click "Browse". Select
      "prj_rx72m_e" and click "Open". Click Finish to complete the project import.
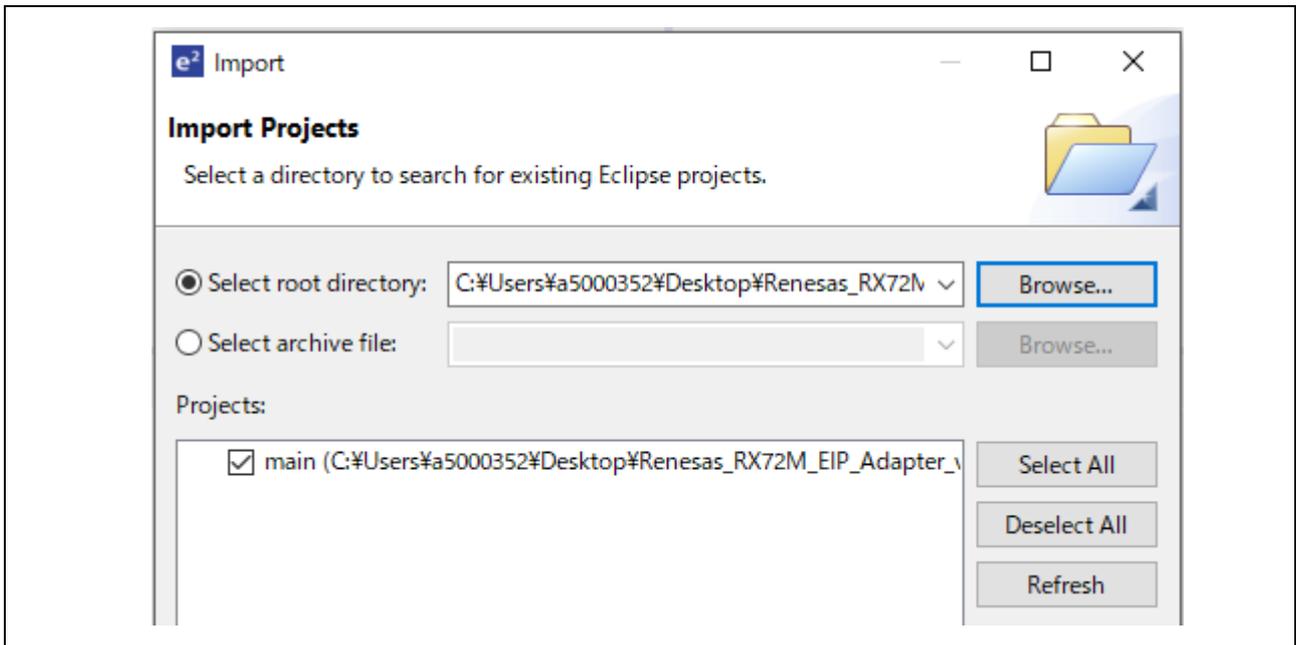
(4)   Select the "main" project in the [Project Explorer] field, select the arrow next to the [Build] button
      (hammer icon), and select [HardwareDebug] from the drop-down menu.
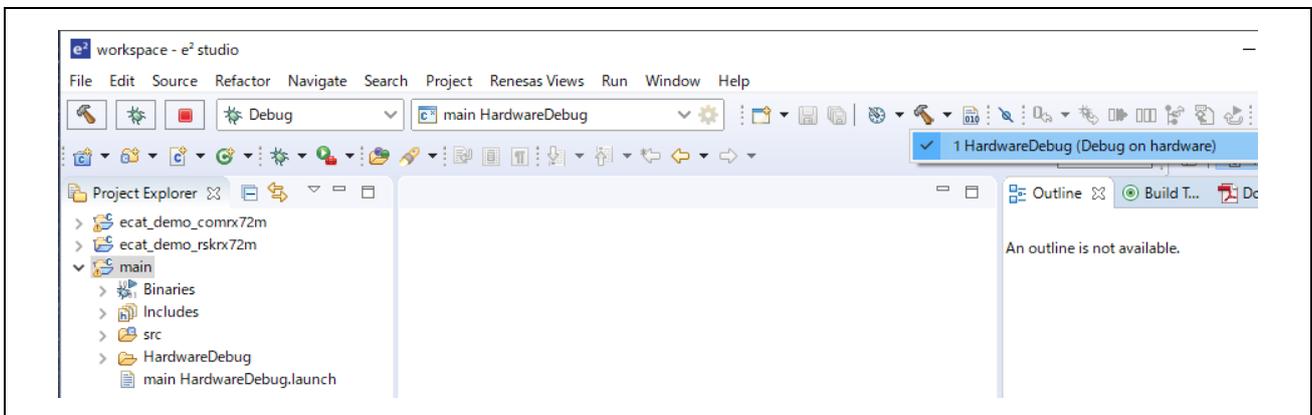
(5)   e2studio builds the project. Once the build is complete, you can start debugging by selecting the arrow
      next to the Debug button (bug icon) and selecting "Debug Configuration".
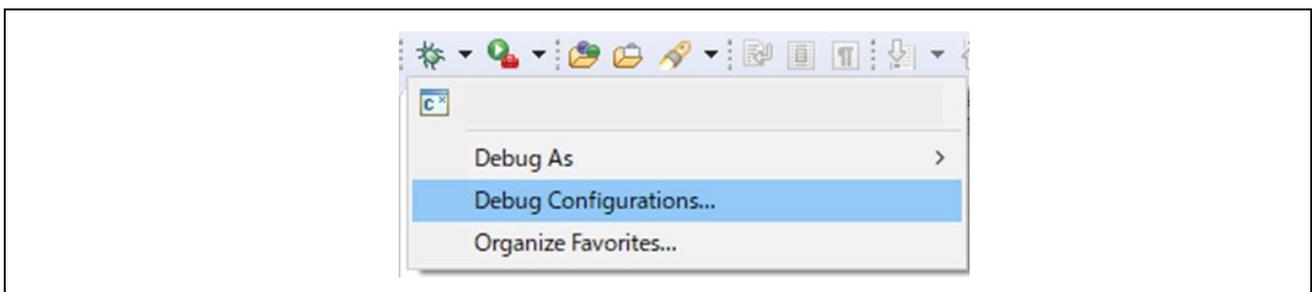
(6) Set up E2OB. E2OB functions as an emulator equivalent to E2Lite, so when connecting, set the emulator type to "E2Lite" and the communication interface to "FINE".
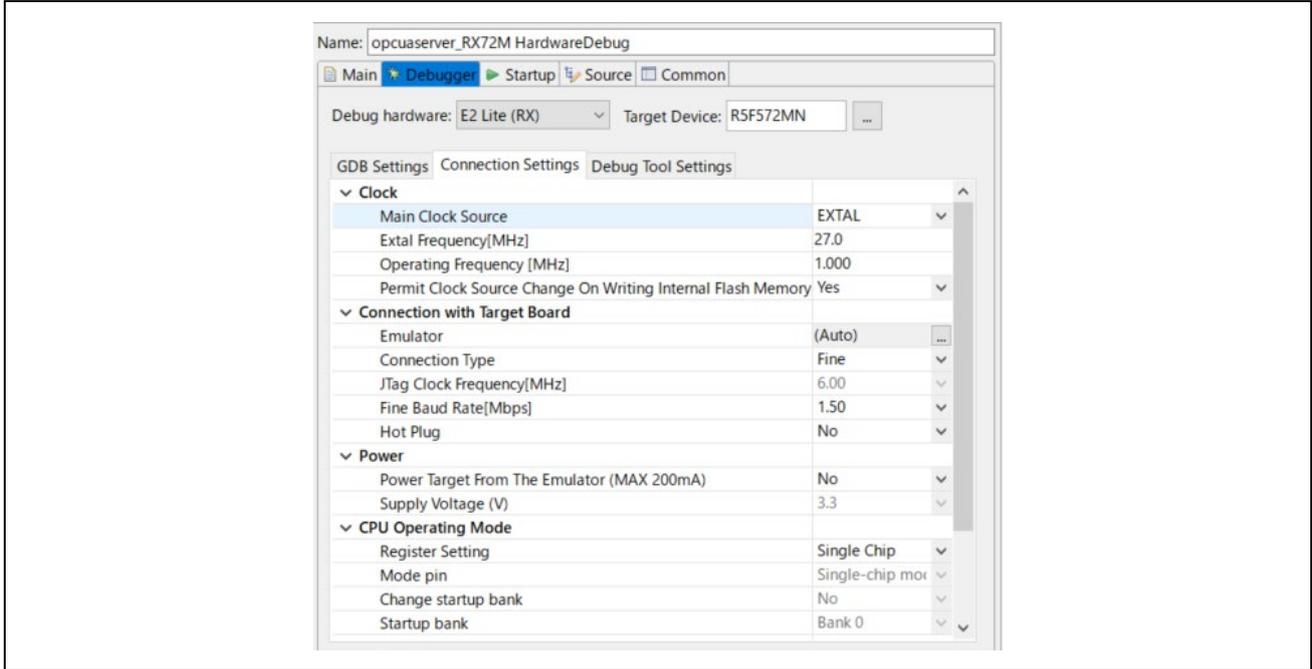
(7) Select "main HardwareDebug" and select [Debug] to download the program to the target.
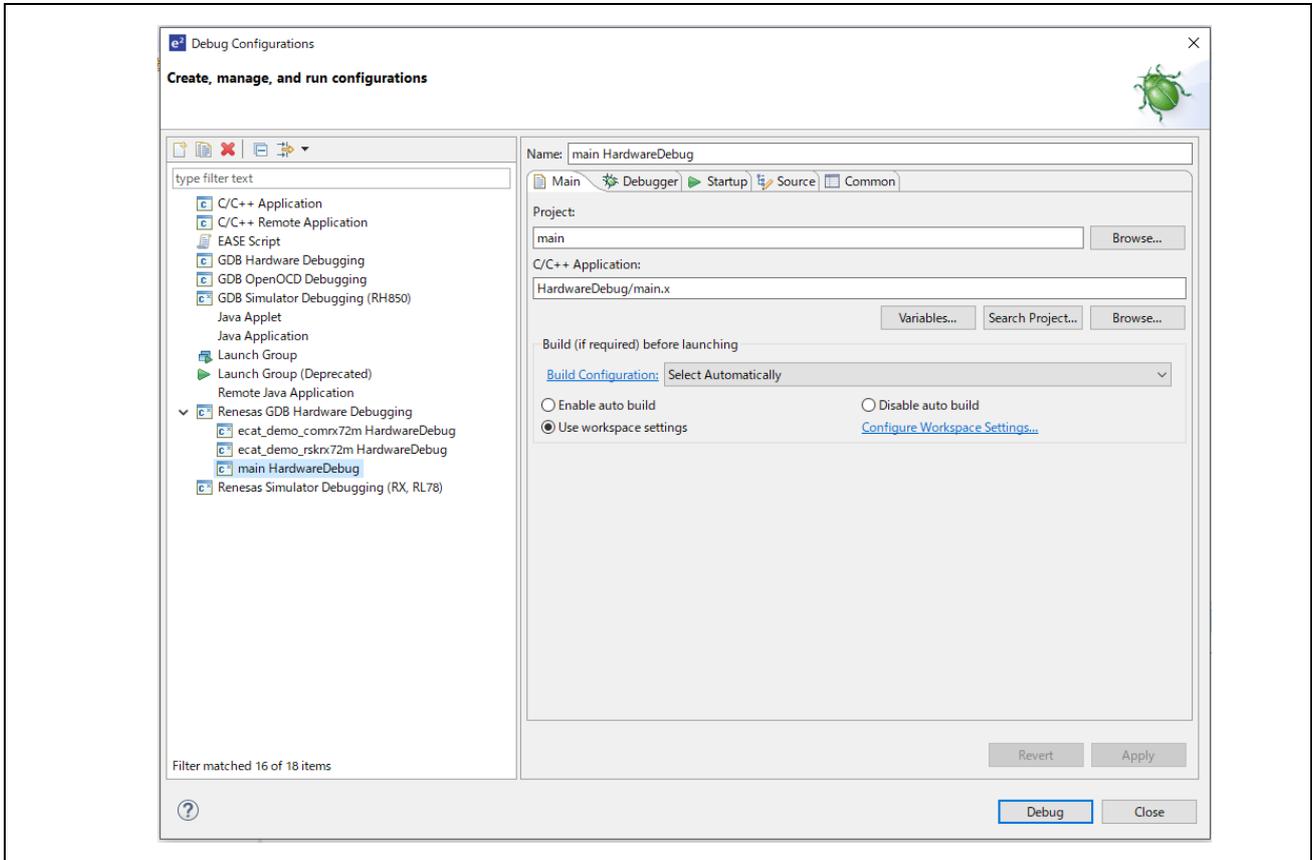
(8)   If a firewall warning is displayed for "e2-server-gdb.exe", check the checkbox for [Private networks, such as my home or work network] and select [Allow access].

(9)   The User Account Control (UAC) dialog box may appear. Enter the administrator's password and select [Yes].

(10) If the Confirm Perspective Switch dialog box appears prompting you to switch the perspective, check the checkbox for [Remember my decision] and select [Yes].

(11) The green "ACT" LED on the E2 Lite debugger will be continuously lit.

(12) After downloading the code, select the [Resume] button to run the code. The code will break at the address where the main function starts. Select the [Resume] button again to continue to run the code.
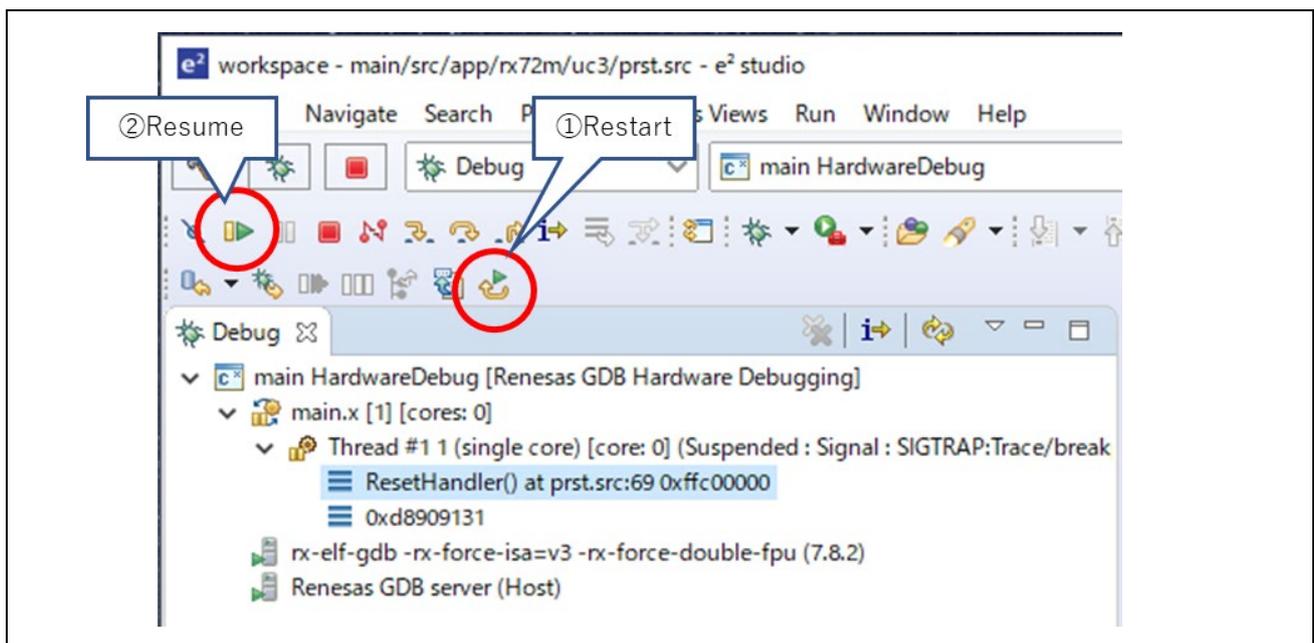
5. Confirmation of sample project operation

This chapter describes the sample project operation check using the CODESYS software PLC.

Connect the communication board and PC according to "5.1 Connection Configuration". Connecting the PC and USB Serial of the communication board turns on the power to the CPU Card.
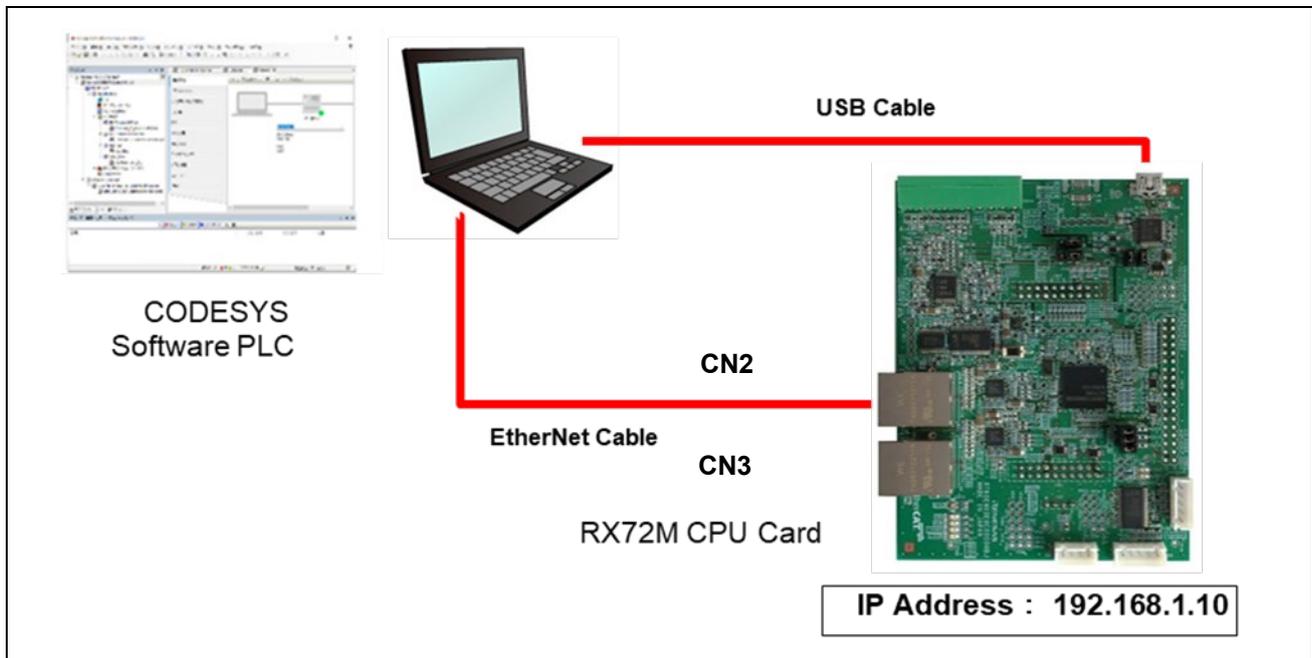
Figure 5.1   Connection configuration

5.1 IP address, MAC address confirmation and setting method

The IP address and MAC address are listed below. Modifiable in source code.

"RX72M_eip_adpt_eva_cpucard\eip_adpt\src\app\rx72m\uc3\user_main.c


```
/***************/
/* USER CUSTOM */
/***************/
char user_app_mac[6] = { 0x12, 0x34, 0x56, 0x78, 0x36, 0x15 };      /* MAC Address */
#ifdef DUAL_PORT_FUNC
char user_port1_mac[6] = { 0x12, 0x34, 0x56, 0x78, 0x36, 0x16 };    /* Port 1 MAC Address */
char user_port2_mac[6] = { 0x12, 0x34, 0x56, 0x78, 0x36, 0x17 };    /* Port 2 MAC Address */
#endif


unsigned long user_app_ipaddr          = 0xC0A8010A;       /* IP address  (192.168.  1.  10) */
unsigned long user_app_mask            = 0xFFFFFF00;       /* Subnet mask (255.255.255.   0) */
unsigned long user_app_gateway         = 0xC0A80101;       /* Gateway    (192.168.  1.  1) */
```

5.2 Starting the CODESYS and Creating a New Project

5.2.1 Starting the CODESYS

Select "All Programs" > 3S CODESYS > CODESYS > CODESYS Vx.x (x.x represents the version number) from the Windows start menu.

You can also startup the program by double-clicking on the "CODESYS" icon, which will be created on the desktop after the installation of the program.

5.2.2 Creating a New Project

Select "New Project" from the "File" menu to create a new project.
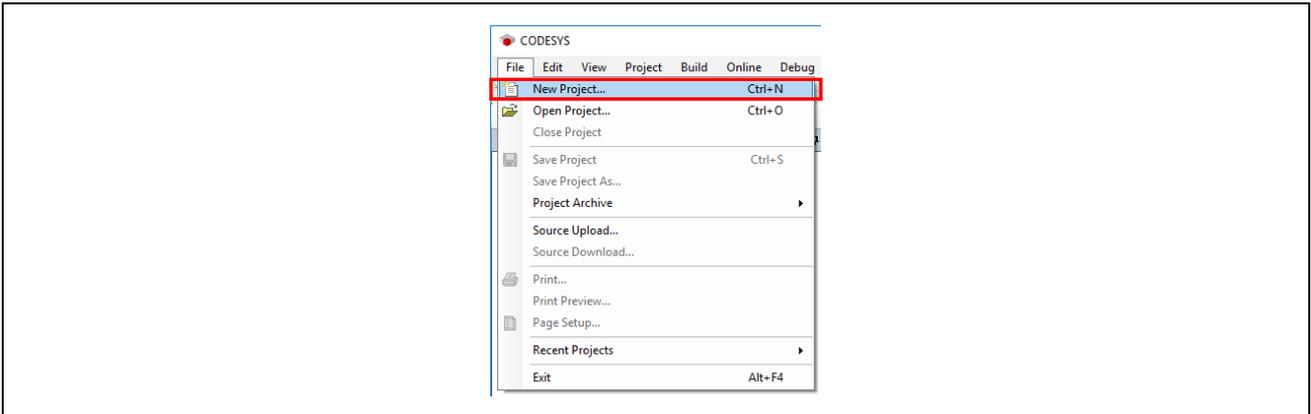
In the "New Project" window, select "Projects" from the "Categories" section and "Standard project" from the "Templates" section.
Then, specify the name of the project and its location and click on "OK".
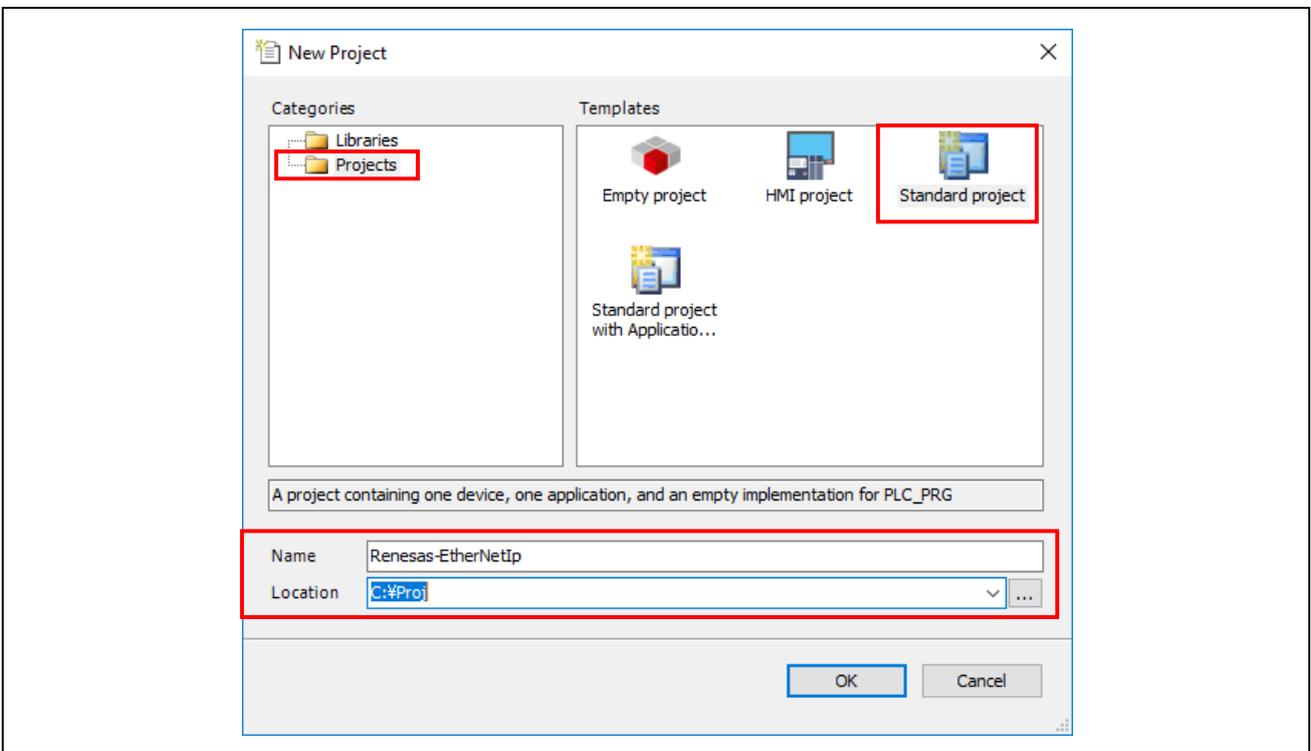
In the "Standard Project" window, select the controller and programming language you wish to use from the drop-down lists for "Device" and "PLC_PRG in". For this example, select "CODESYS Control Win V3" and "Structured Text (ST)", respectively. (If the 64-bit version is installed on the PC, select "CODESYS Control Win V3 x64") After that, click on "OK" to open the new project
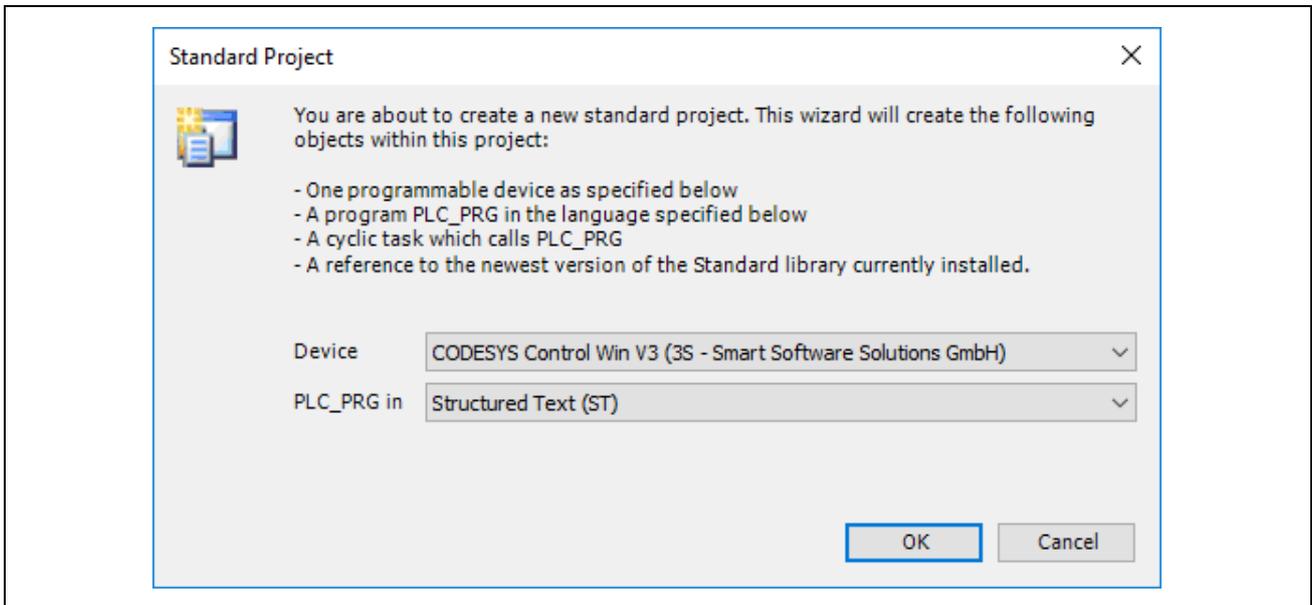
The "Device" tree for the newly created project will be displayed as shown below.

The components that belong to "Device (CODESYS Control Win V3)" are managed in a tree structure.
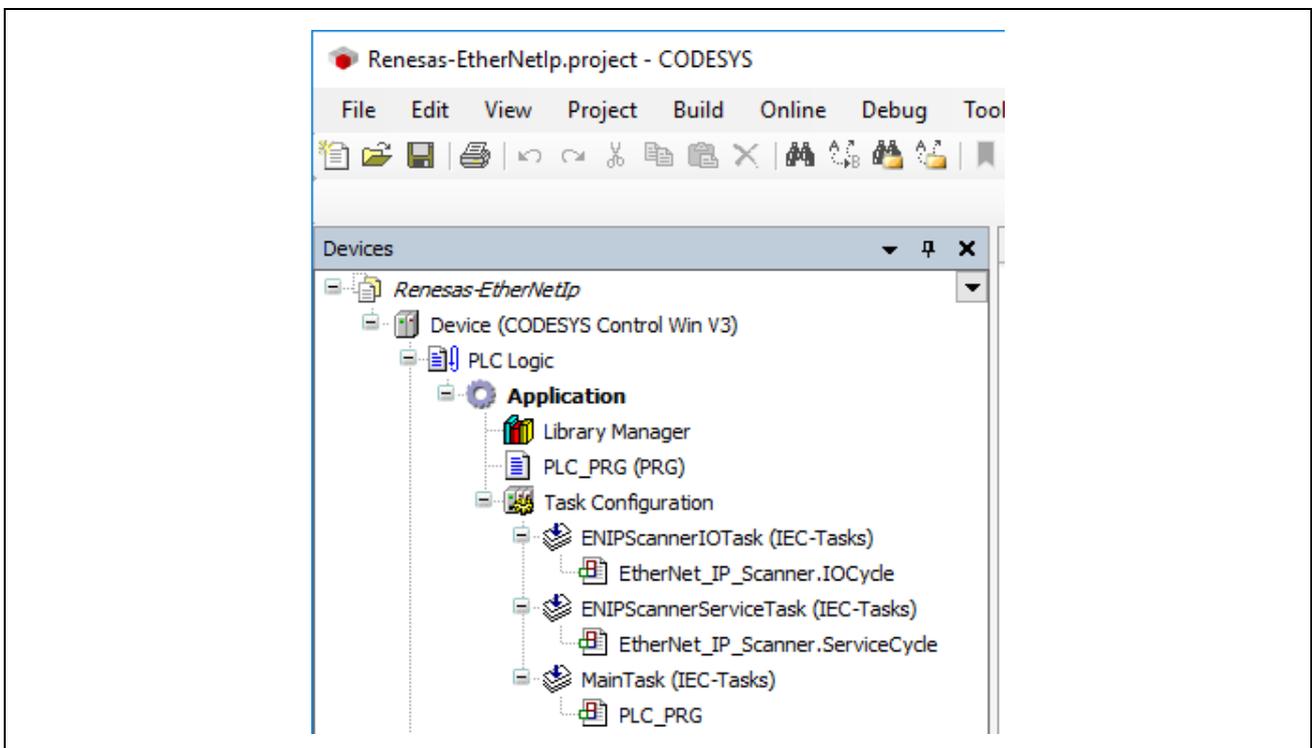
5.2.3   Starting the Gateway Server

Check the state of the gateway server on the system tray. If the server is down, click on the " 🔴 " icon and select "Start Gateway" to start the server up. Usually, the server will automatically be started as a standard service on booting of

Windows and its status are indicated in the system tray in the lower-right corner of the desktop.
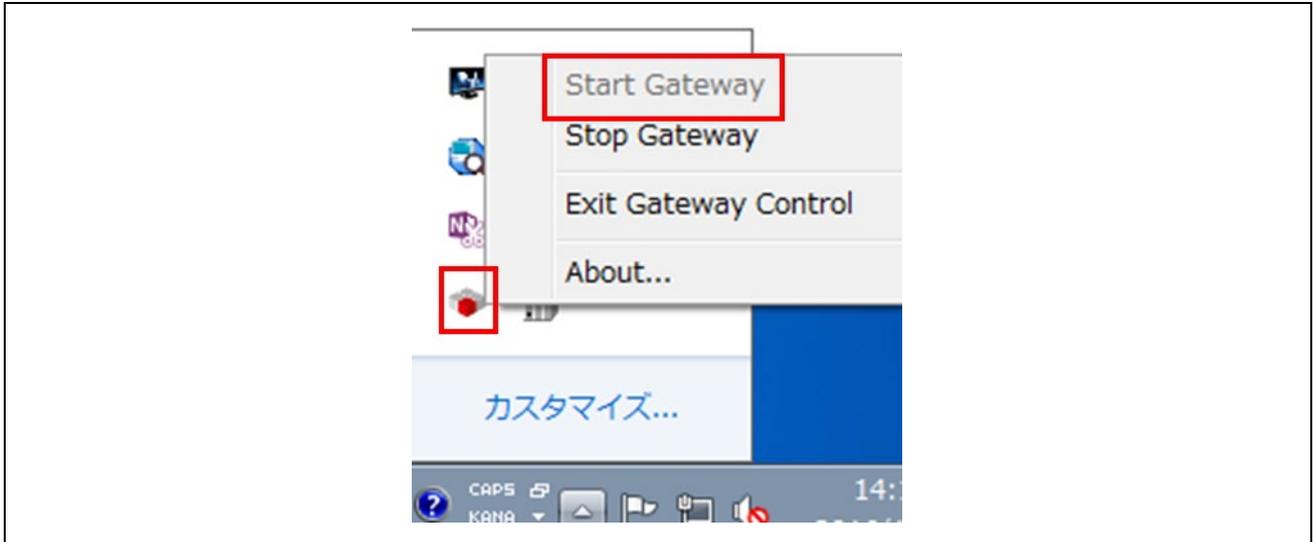
5.2.4   Starting the Software PLC

Check the state of the software PLC on the system tray. If the program is stopped, click on the " 🗔 " icon and select "Start PLC" to start the program up. Usually, the program will automatically be started as a standard service on booting of Windows and its status is indicated in the system tray in the lower-right corner of the desktop
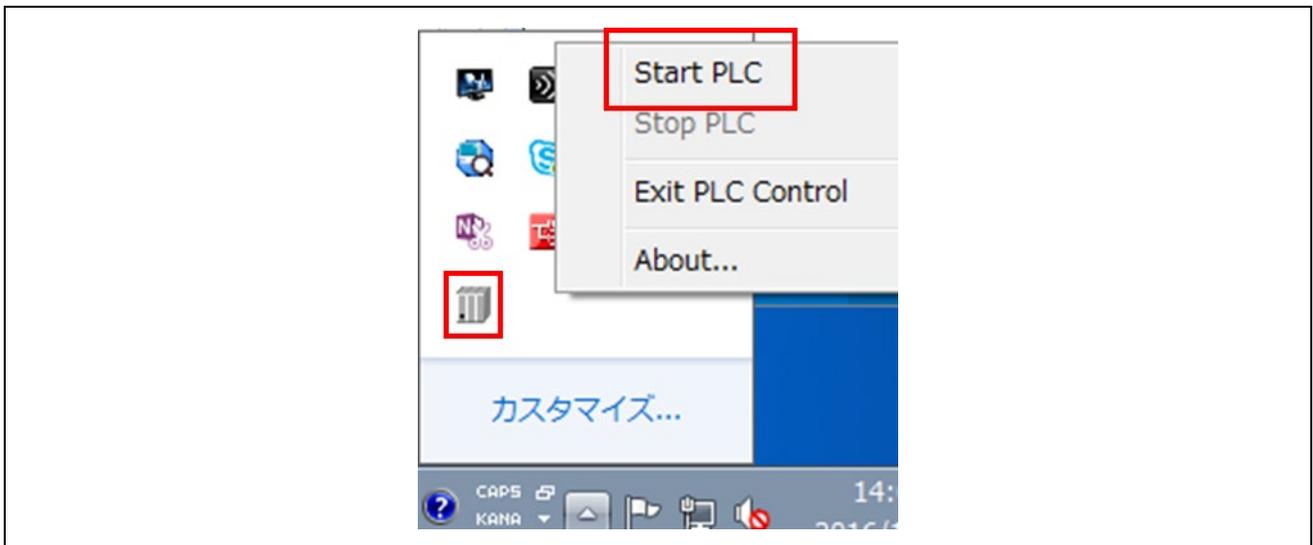
5.2.5   Configuring a Device Network

This section describes device configuration, including the setting of an IP address for use in a network of devices.

Setting the Host IP Address

Set the IP address before configuring the device.

Open "Network Connection".
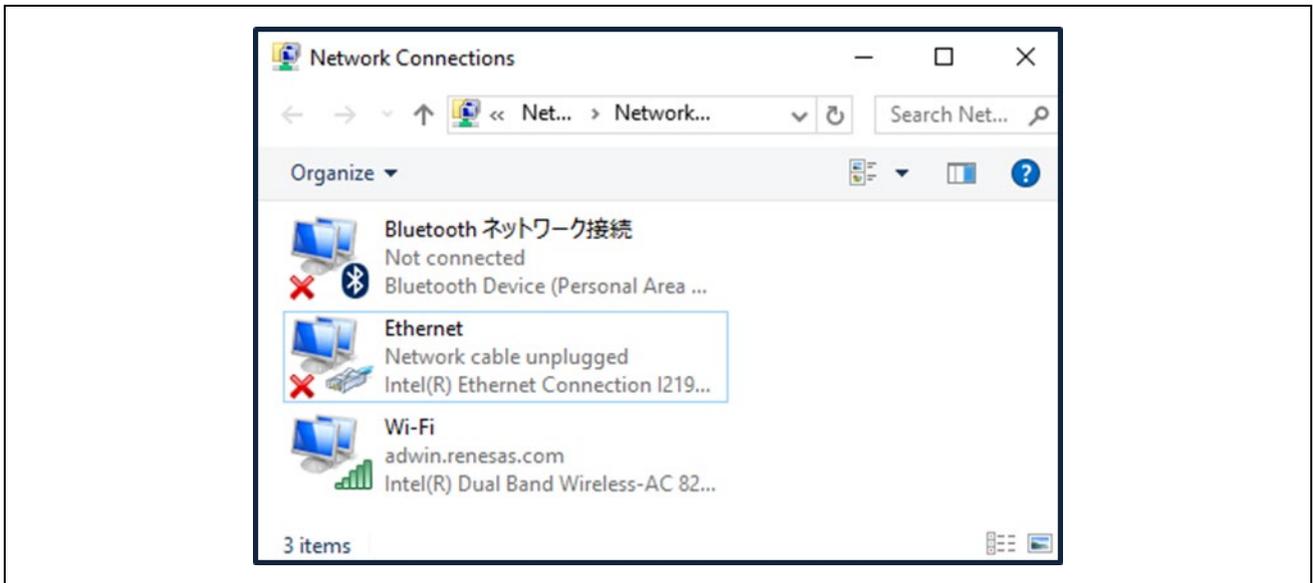
In the "Local Area Connection Status" window, select "Properties".
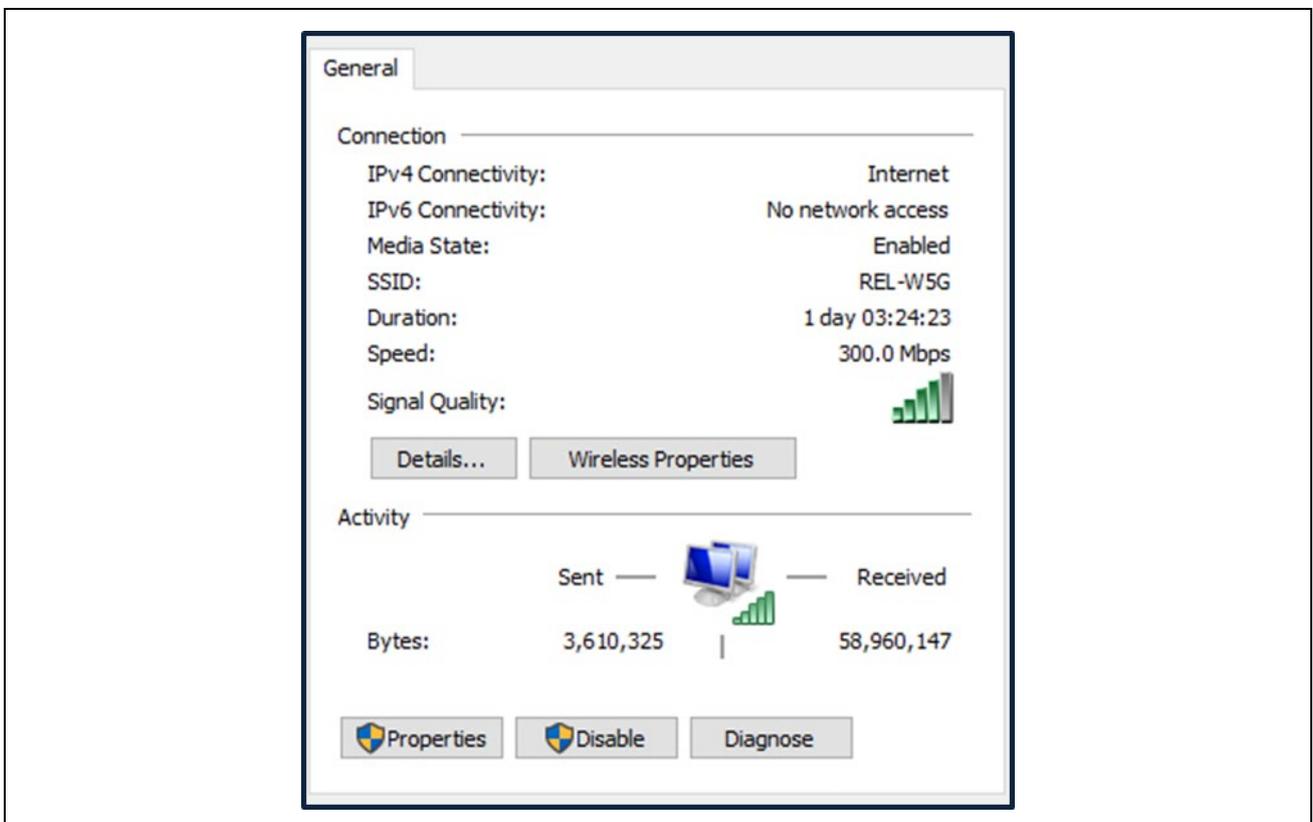
In the "Local Area Connection Properties" window, highlight "Internet Protocol Version 4 (TCP/IPv4)" then click on the "Properties" button.
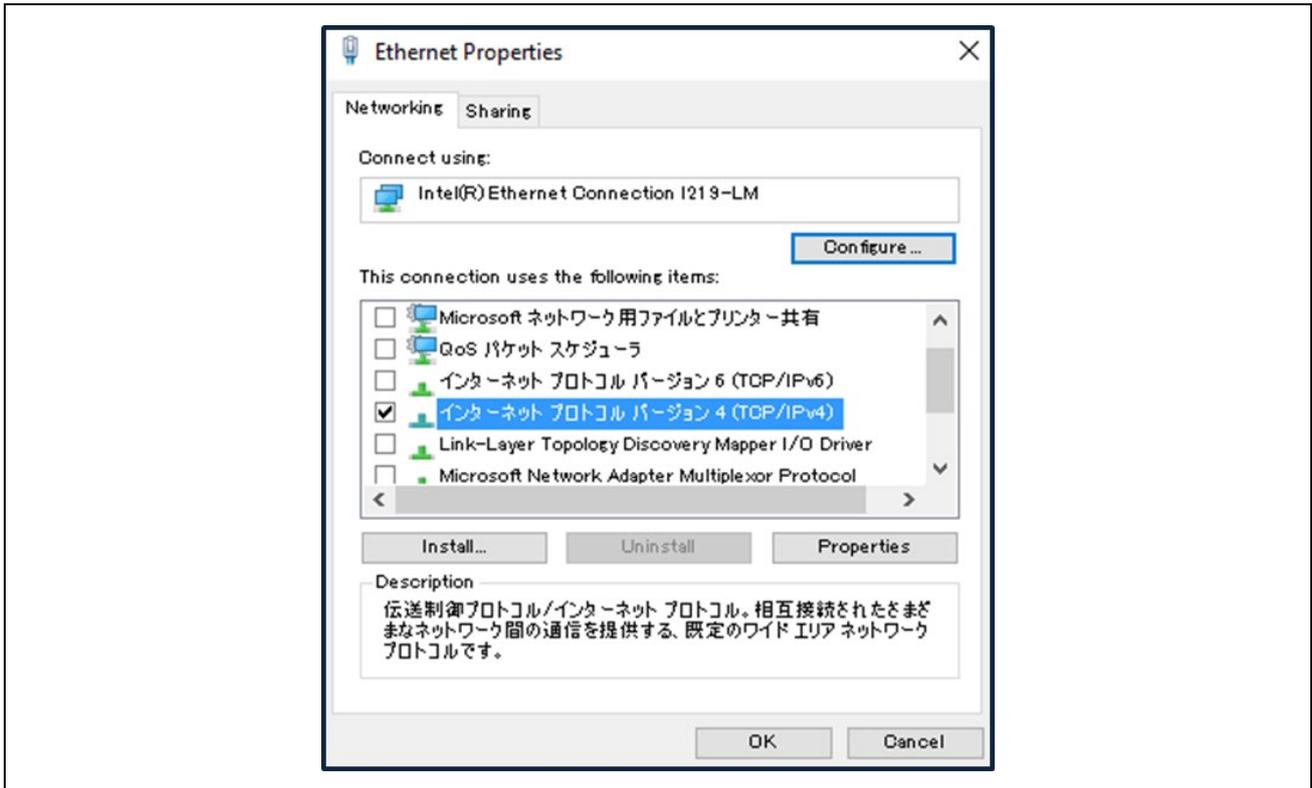
Select the radio button "Use the following IP Address" and set IP and subnet mask. However, the network IP address should match with Device IP.
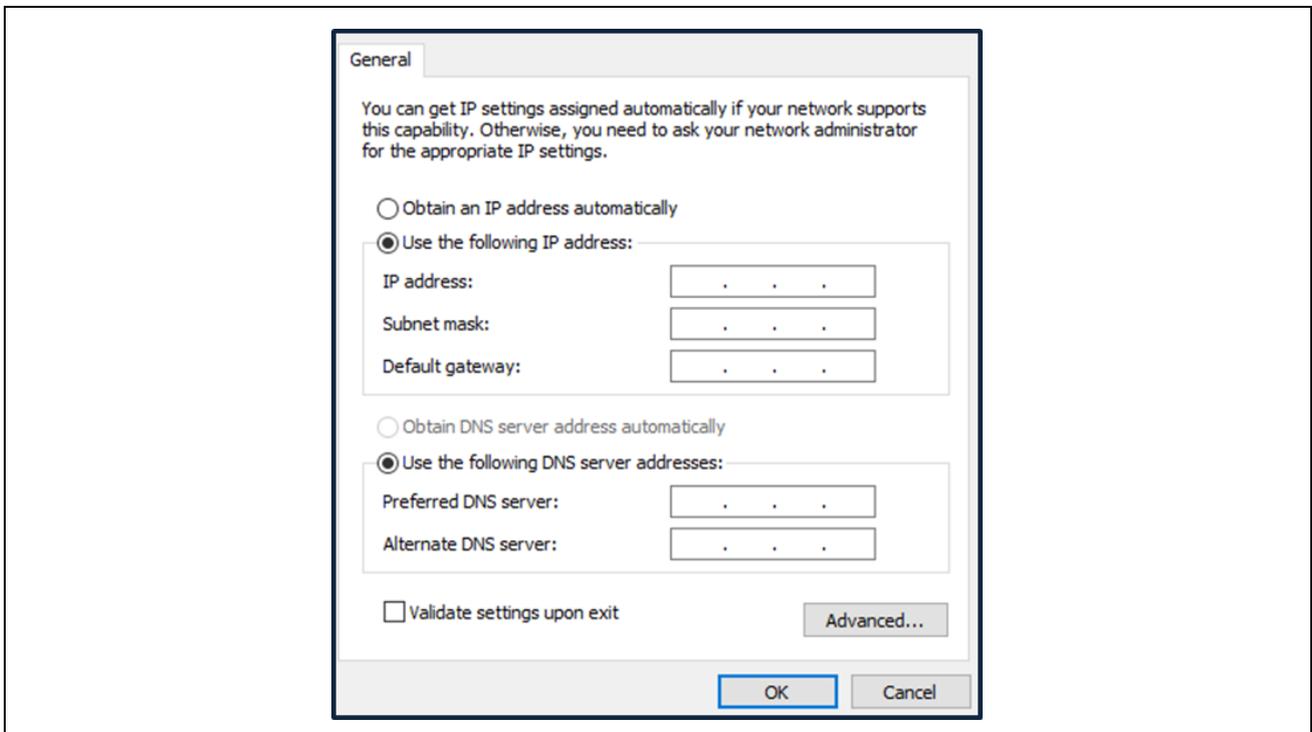
This is the end of the configuration.

5.3   Connection setting with CODESYS

5.3.1   Select device

Make connection settings for connecting the software PLC service from your development environment. Double-click on the "Device (CODESYS Control Win V3)" in the "Device" tree. On the "Device" tabbed page, select "Connection settings" and click on the "Scan network…" button.
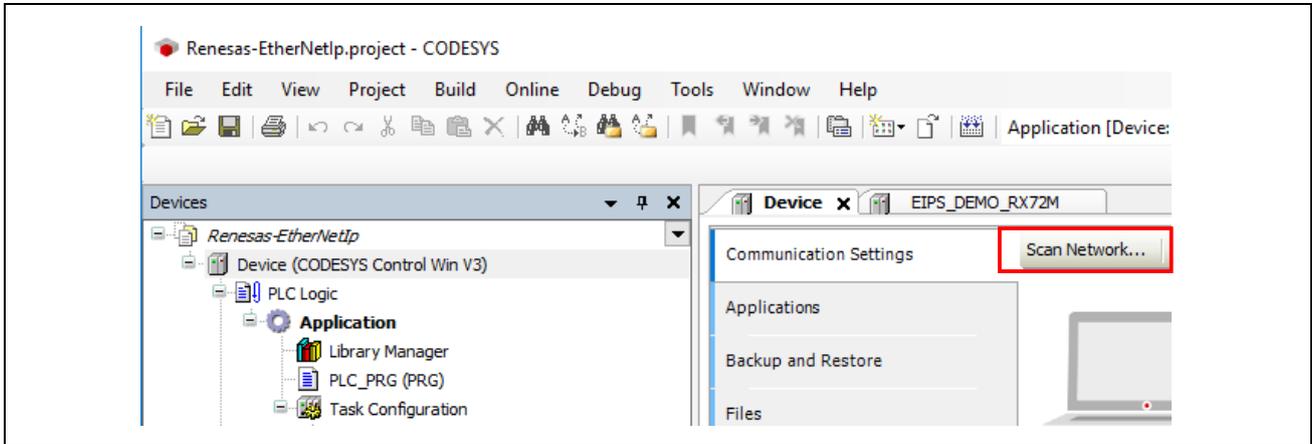
The "Select Device" window opens and a search for available devices that can use the local network automatically starts. Finding a software PLC service constitutes success and the name of the corresponding PC will be indicated. Double-click on the PC name to make a connection.
If the service will not be found, check the settings described in previous sections, 5.1.3Starting the Gateway Server and 5.1.4Starting the Software PLC.
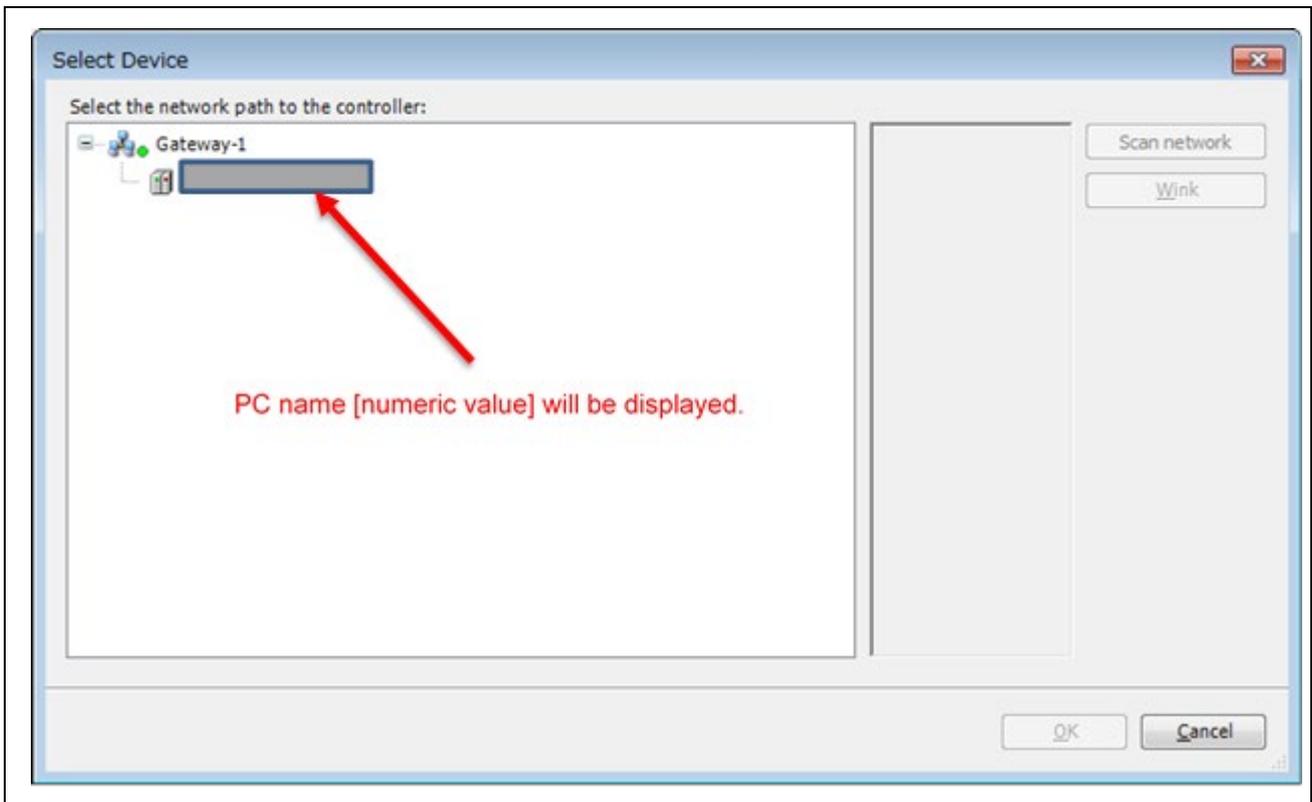
5.3.2 Installing the Device Information

Install an EDS (electronic data sheet) file which contains a description of the EtherNet/IP slave device. A file for use with EtherNet/IP is provided with the released stack.

Select "Device Repository…" from the "Tools" menu of the CODESYS program.
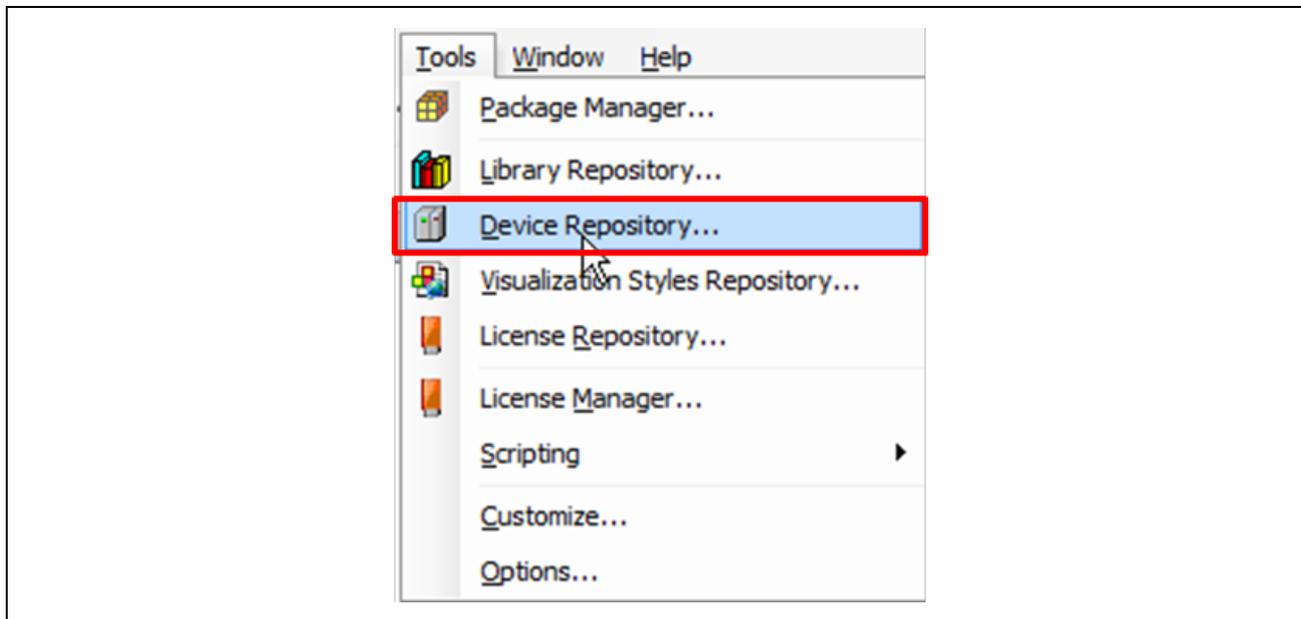
In the dialog box, click on the "Install" button to produce the dialog box where you are to enter the name of the provided EDS file. Specify " Renesas_RX72M_EtherNetIP_Adapter.eds ". The result of installation will be indicated under the file name. An icon " " appears in the case of normal installation, as is shown within the blue rectangle in the figure below
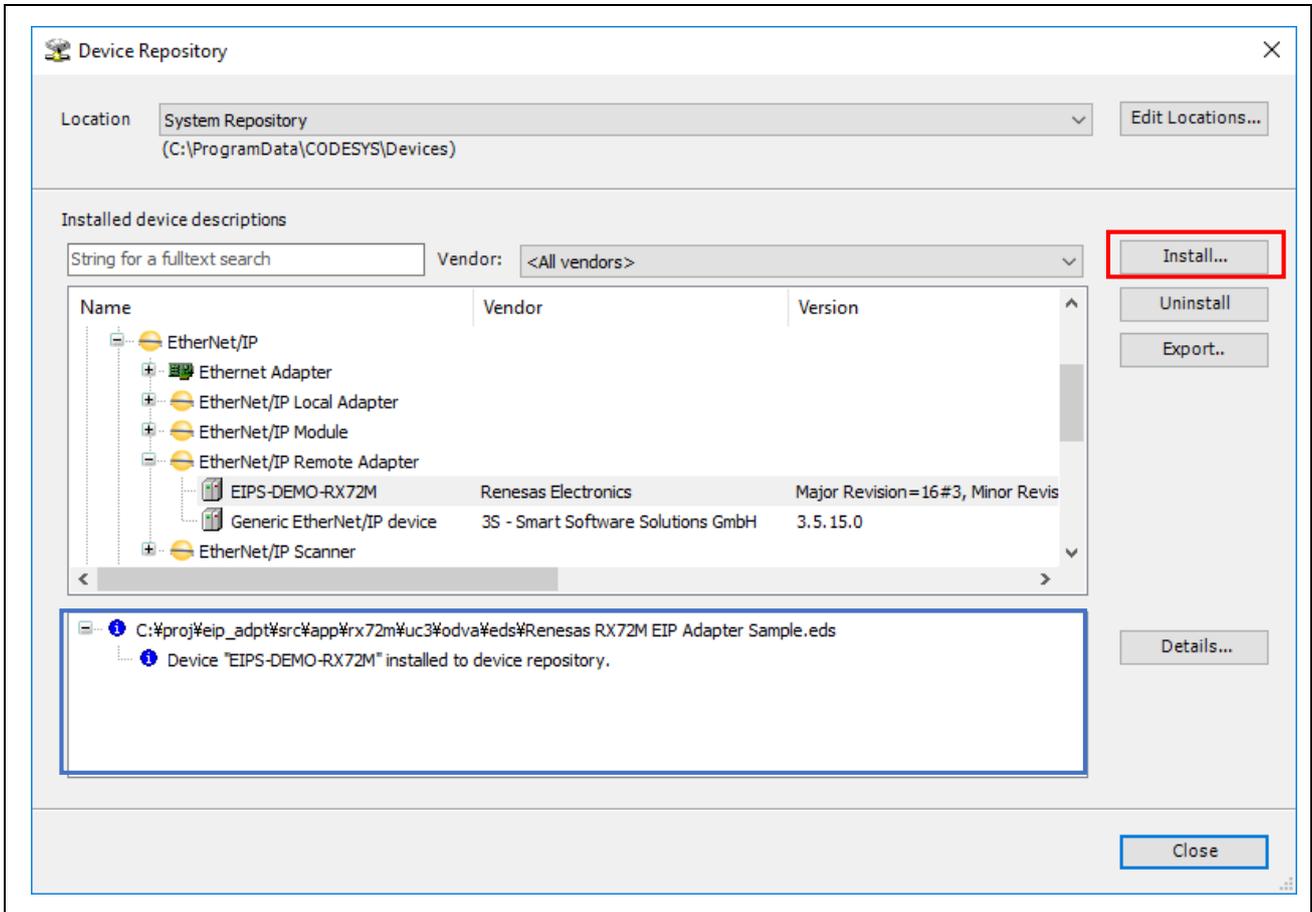
5.3.3   Adding EtherNet/IP Device

Add necessary devices to the "Device" tree.

(1)   Adding Ethernet

Right-click on "Device (CODESYS Control Win V3)" in the "Device" tree and select "Add Device".
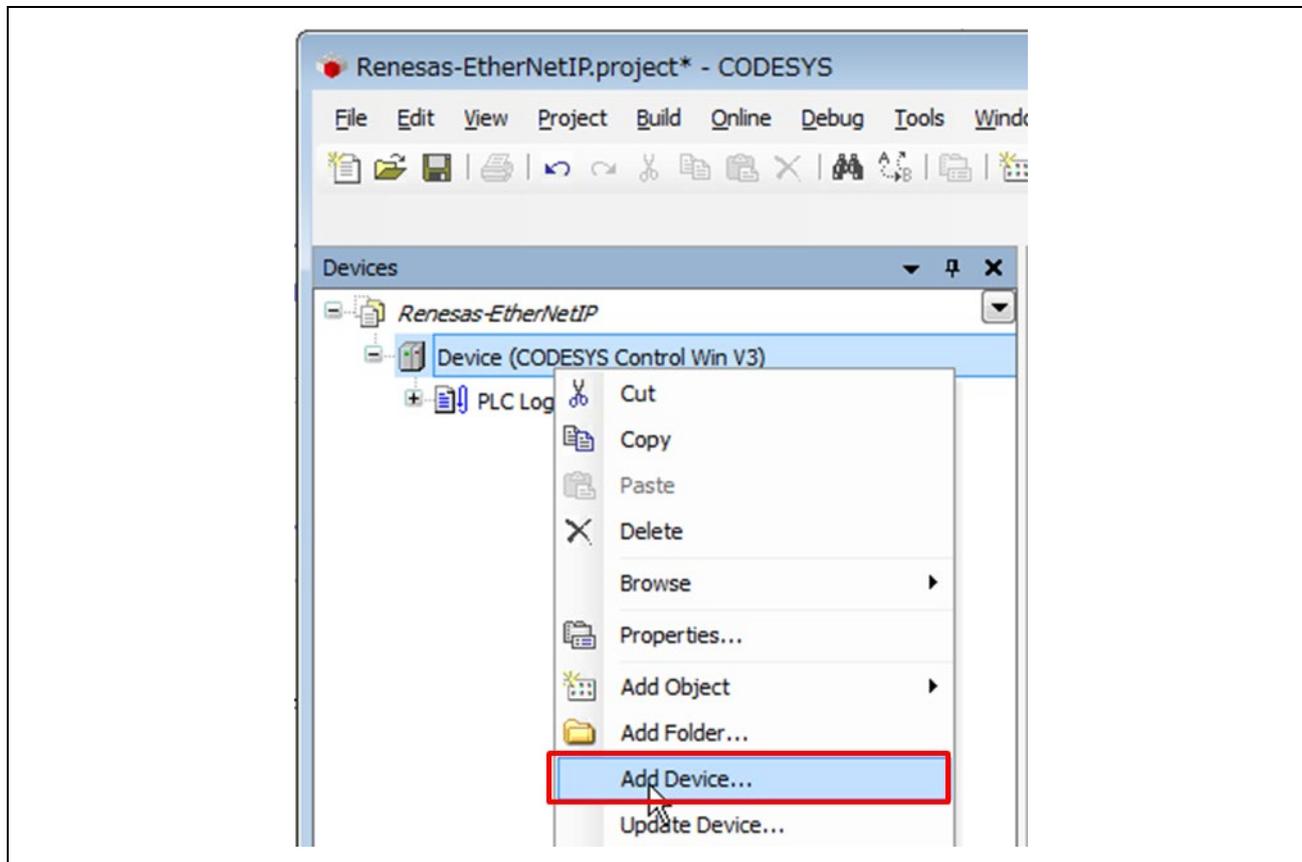
The "Add Device" dialog box opens. Select "Ethernet" under "Fieldbusses", "Profinet IO", then "Ethernet Adapter" and click on the "Add Device" button.
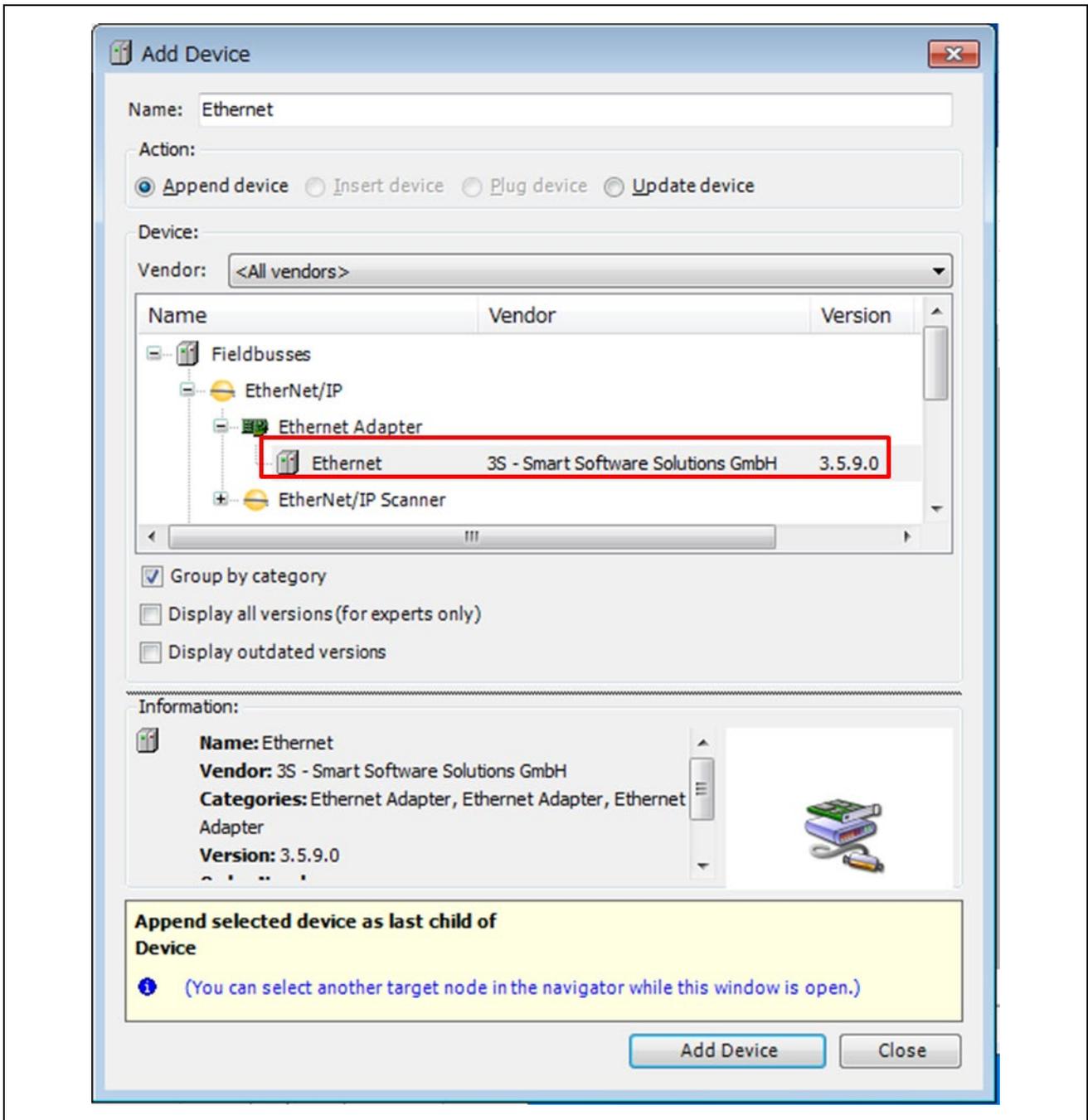
You can see that "Ethernet" has been added under "Device (CODESYS Control Win V3)" in the "Device" tree.
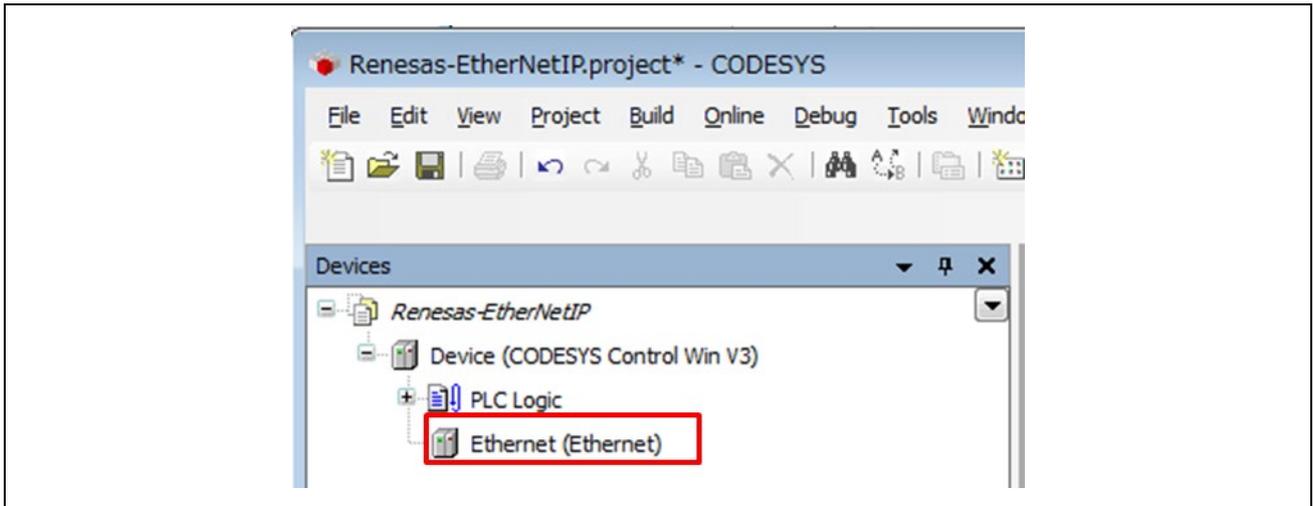
(2)   Adding EtherNet/IP Scanner

Right-click on "Ethernet (Ethernet)" in the "Device" tree and select "Add Device".
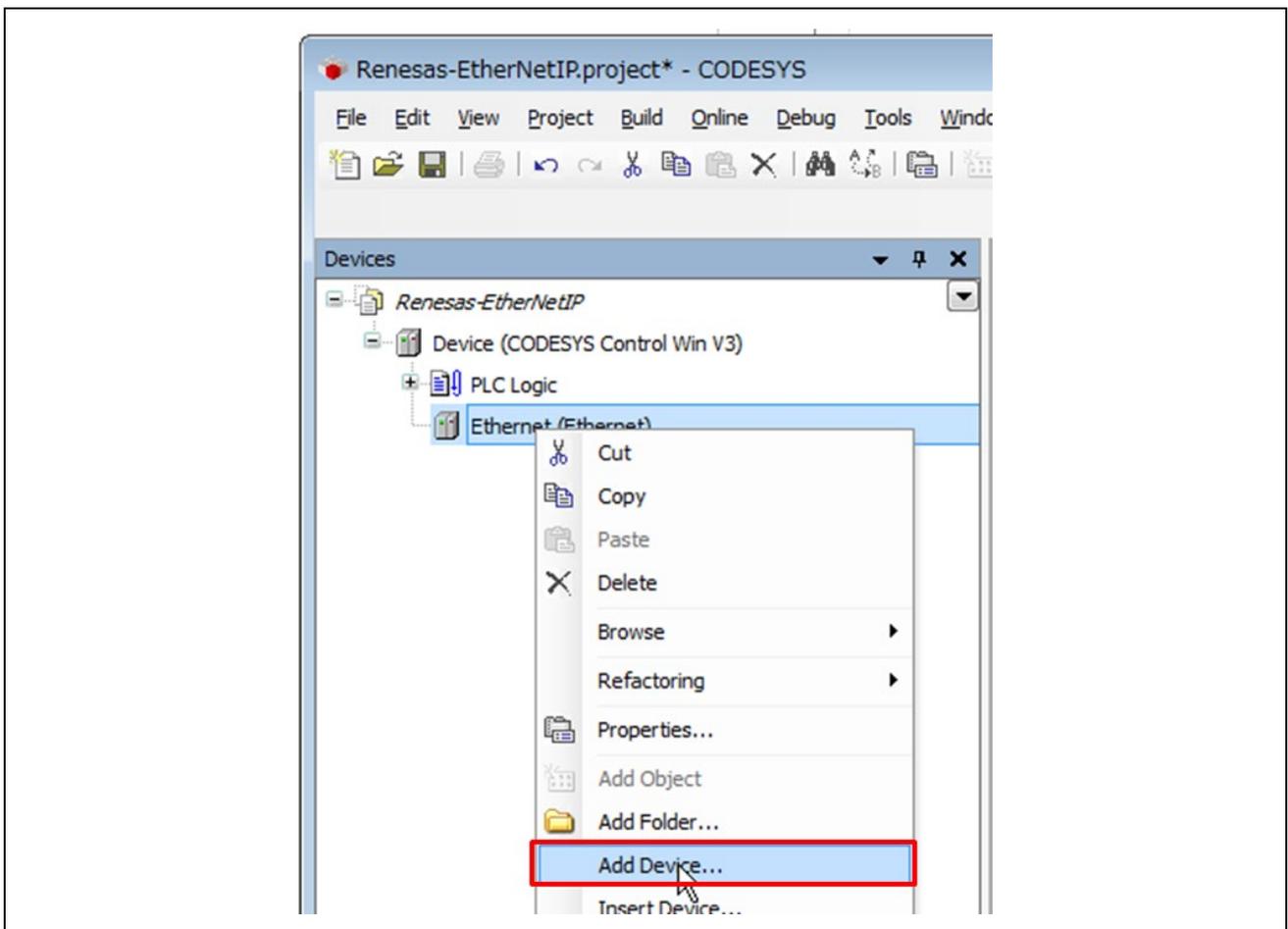
The "Add Device" dialog box opens. Select "EtherNet/IP Scanner" under "Fieldbusses", "EtherNet/IP", then "EtherNet/IP Scanner" and click on the "Add Device" button.
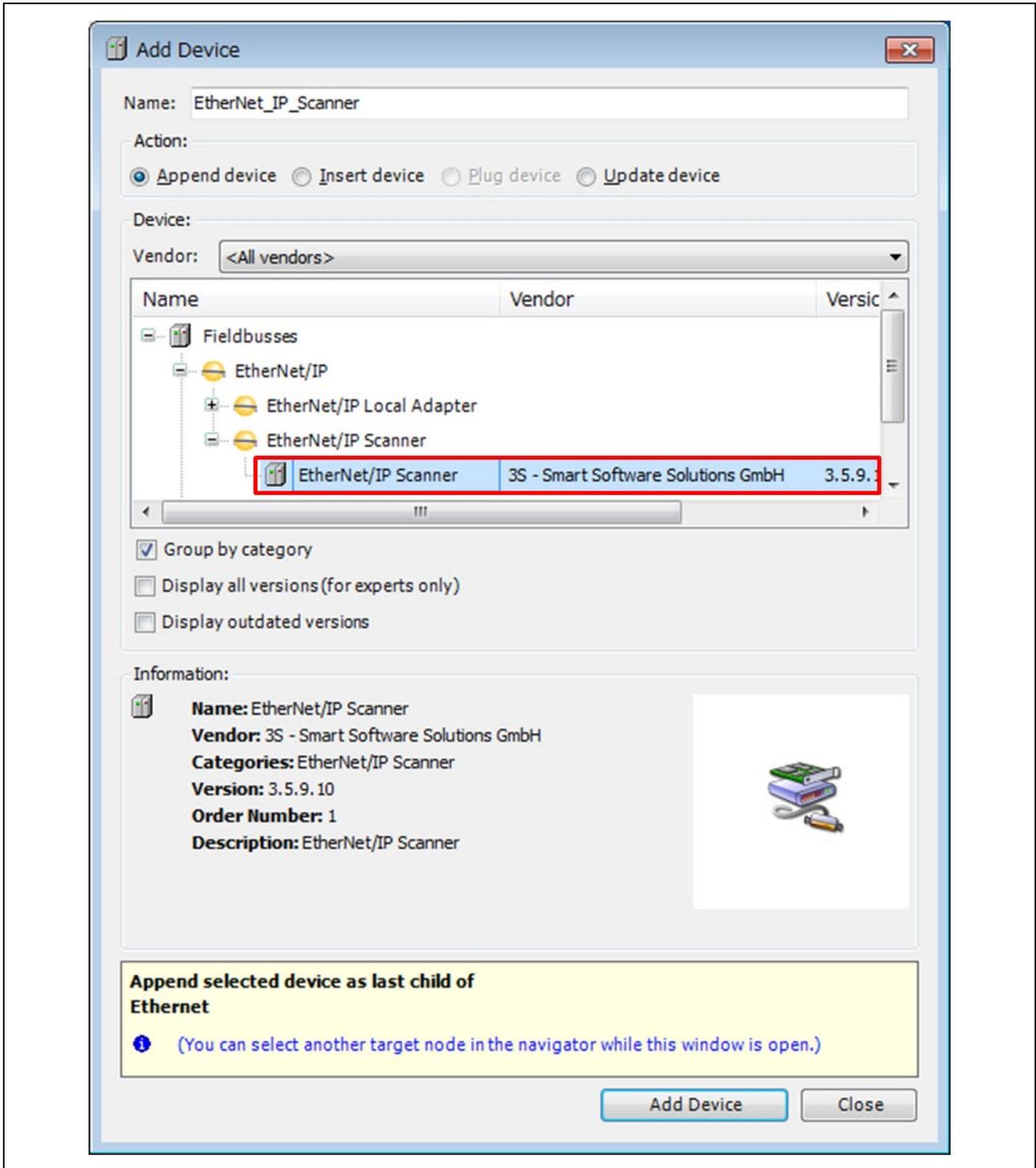
You can see that "EtherNet/IP Scanner" has been added under "Ethernet" in the "Device" tree.
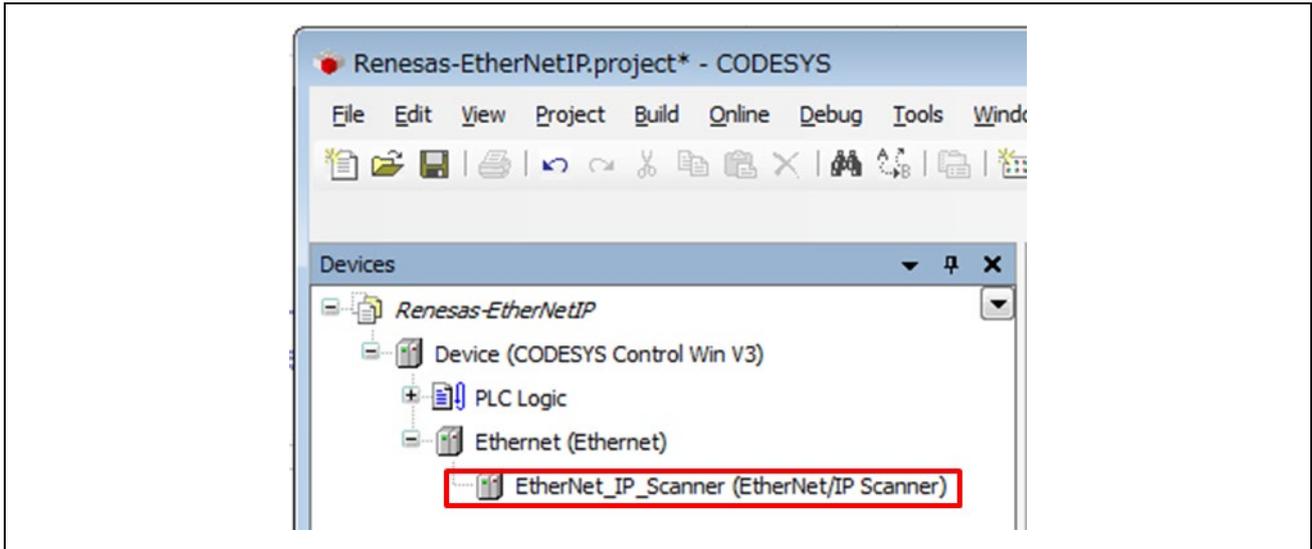
(3)   Adding Renesas_EIP_Adapter

Right-click on "EtherNet_IP_Scanner" in the "Device" tree and select "Add Device".
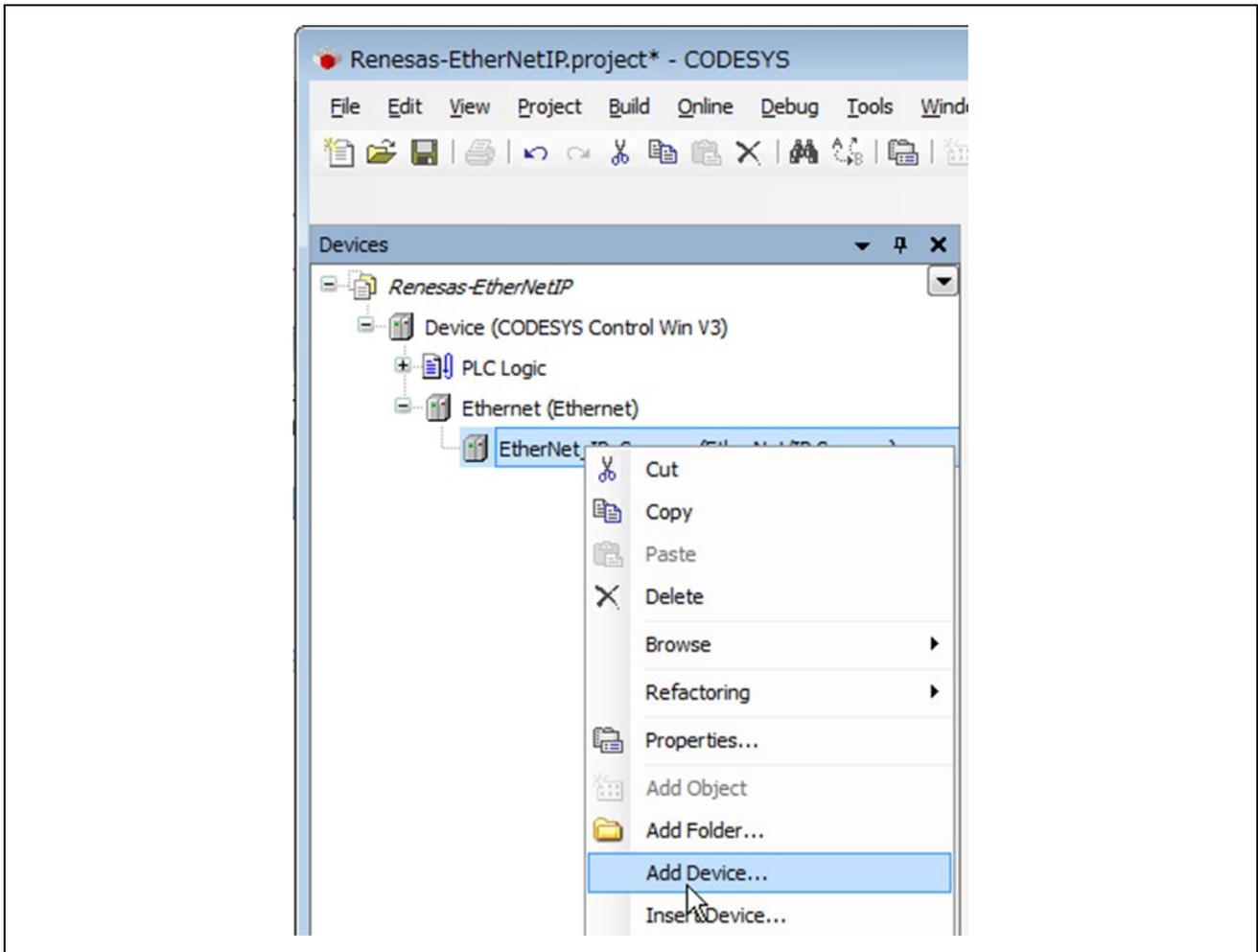
In the Add Device dialog, select "EIPS-DEMO-RX72M" under "Fieldbusses – EtherNet / IP – EtherNet / IP Remote Adapter" and click the "Add Device" button.
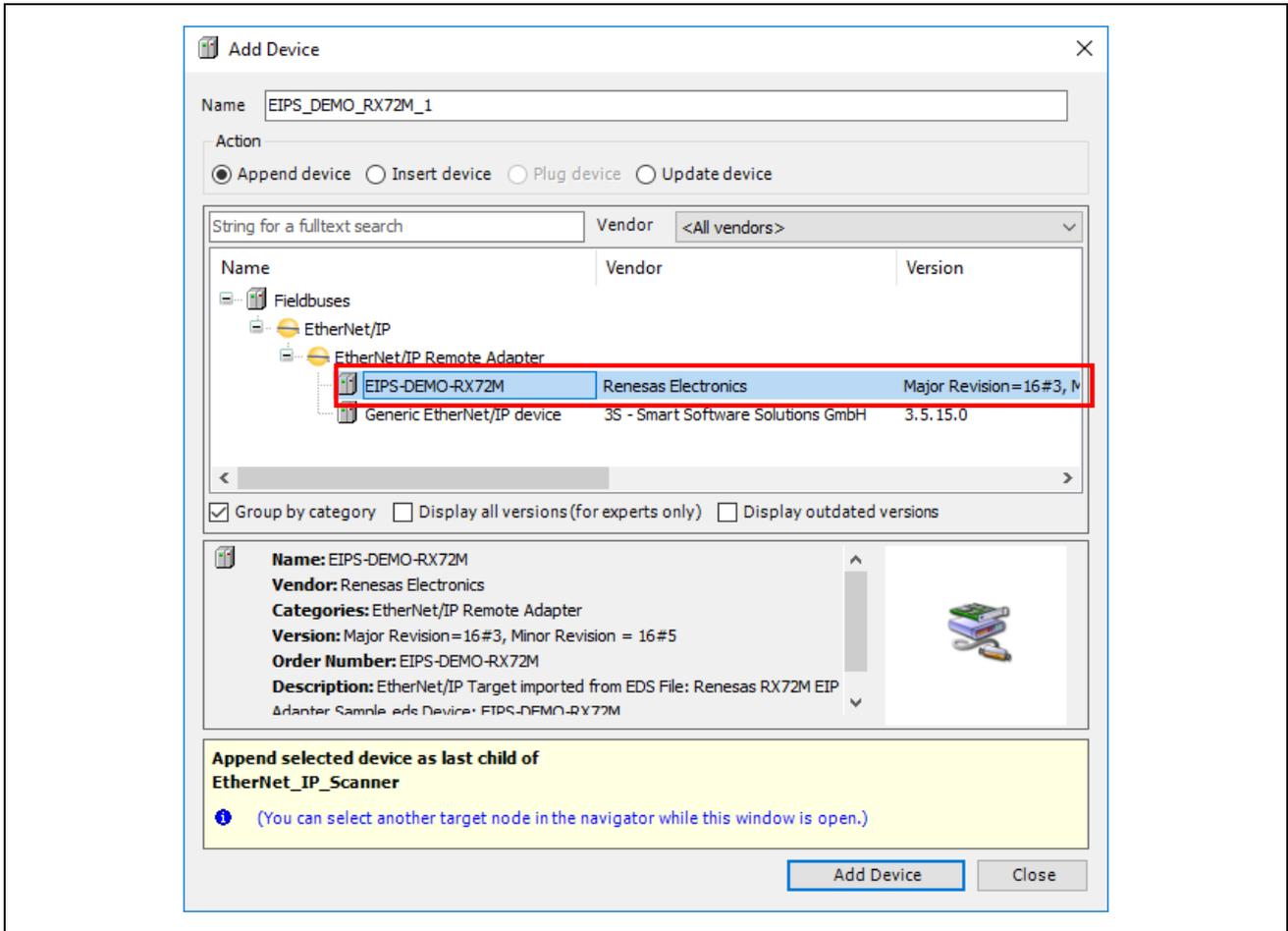
"EIPS-DEMO-RX72M" is added as a device under "EtherNet_IP_Scanner" in the device window.
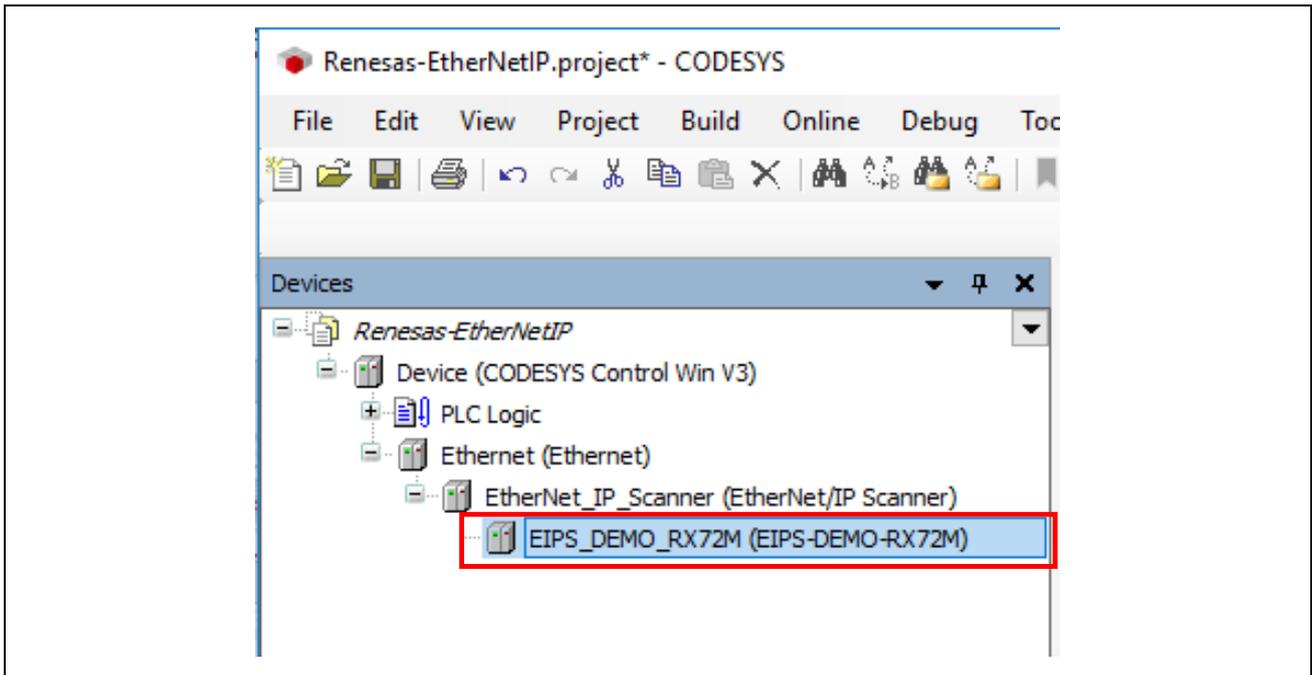
(4)   Configuring the Ethernet

Double-click on "Ethernet (Ethernet)" in the "Device" tree to open the configuration window. In the "General" tabbed page, click on the icon next to the text box for "Interface" section as shown in a red rectangle below
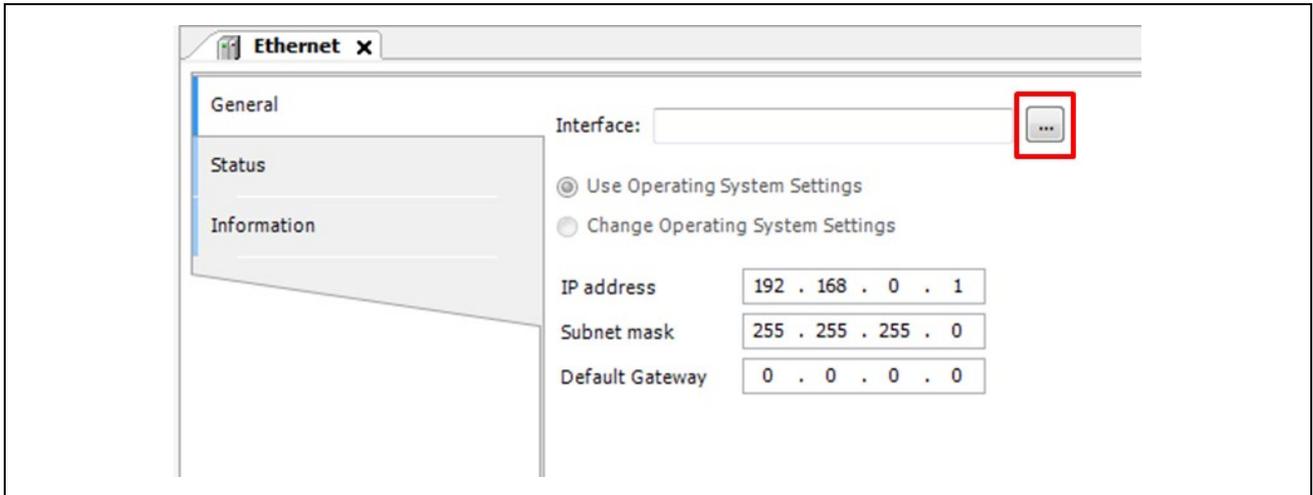
In the "Network Adapters" window, select the interface you wish to use from among the interfaces offered for connection.
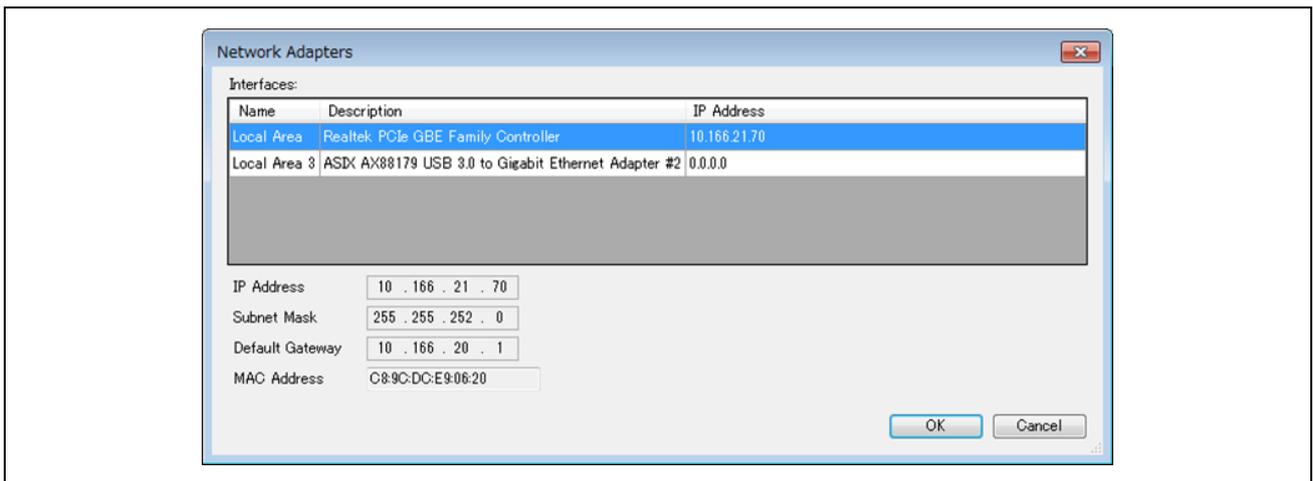
Confirm that the correct IP address is set for the interface you have selected
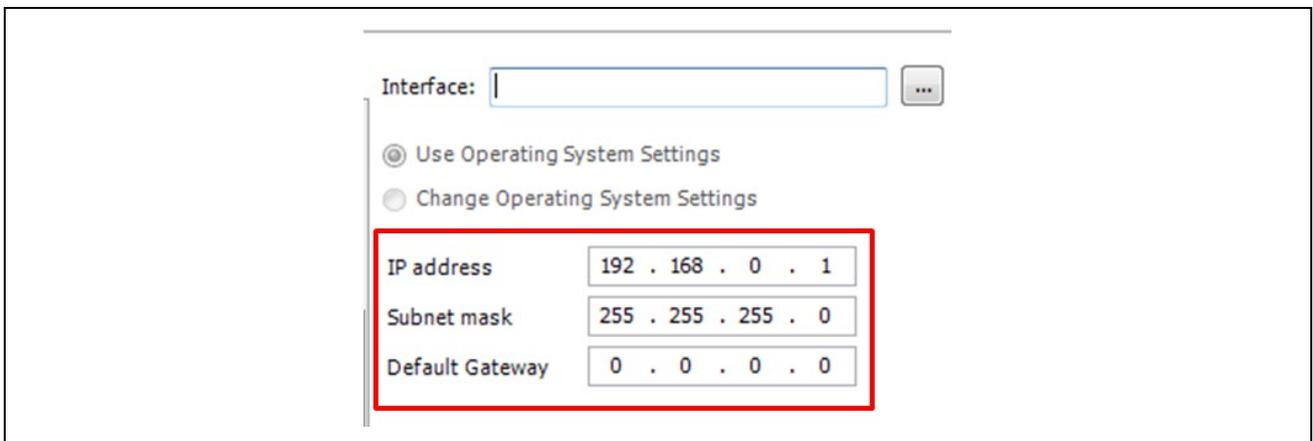
(5)   Setting of EtherNet/IP_Scanner

Users do not need to make settings for this device.

(6)   Setting of EtherNet/IP_Adapter

Double-click on "EtherNet_IP_Scanner (EtherNet_IP_Scanner)" in the "Device" tree to open the configuration window. Then, select the "General" tab.

The EtherNet/IP system recognizes slave devices by the IP address. Enter the address set within the slave device   (For setting method please refer to 6.6 IP address setting method).
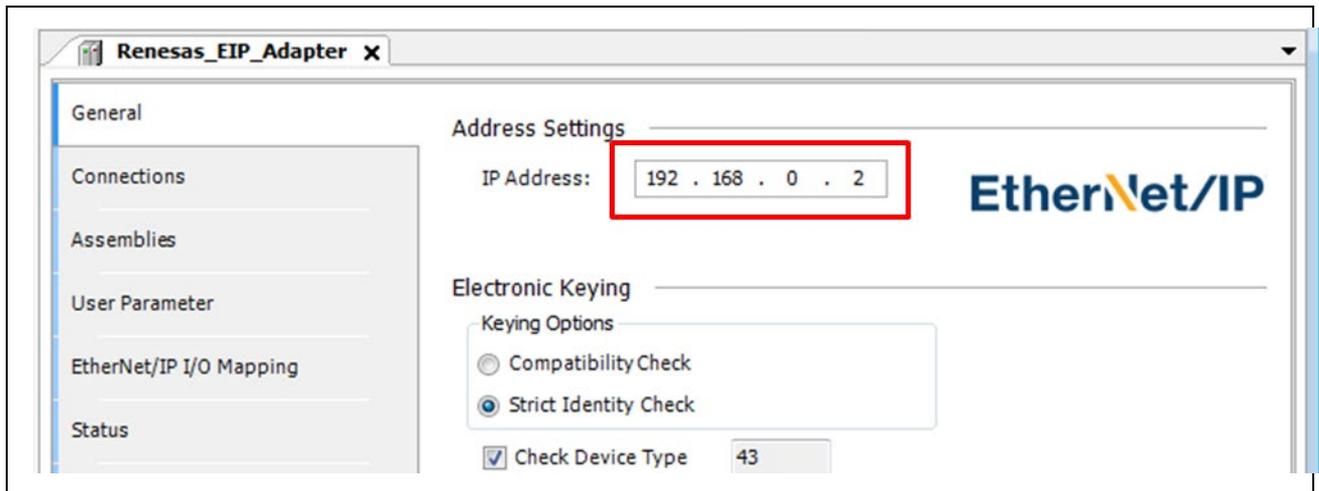
5.4　Connection confirmation with CODESYS

This section describes how to run a program on the CODESYS development environment and how to ensure connection of the device.

Select "Build" from the "Build" menu.
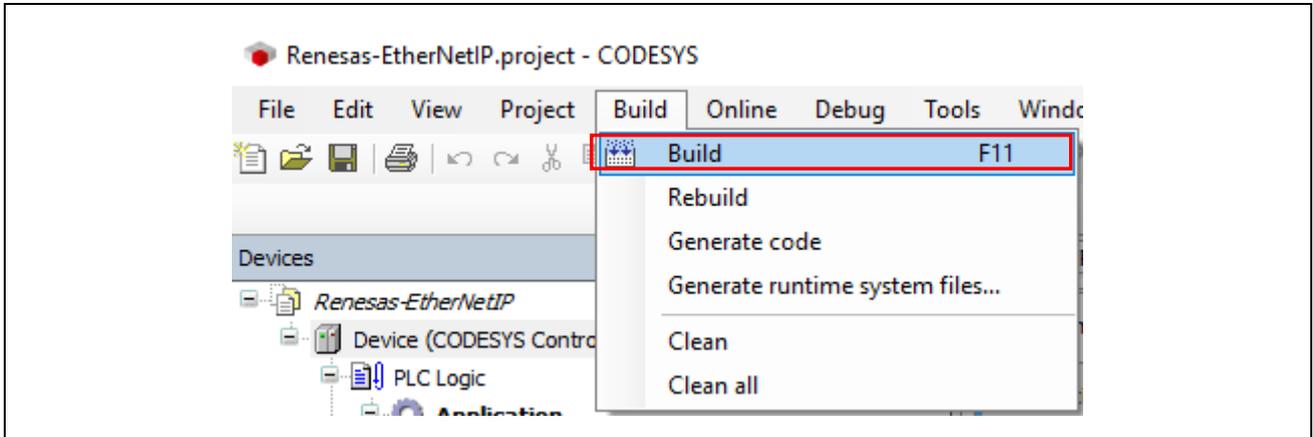
Select "Login" from the "Online" menu
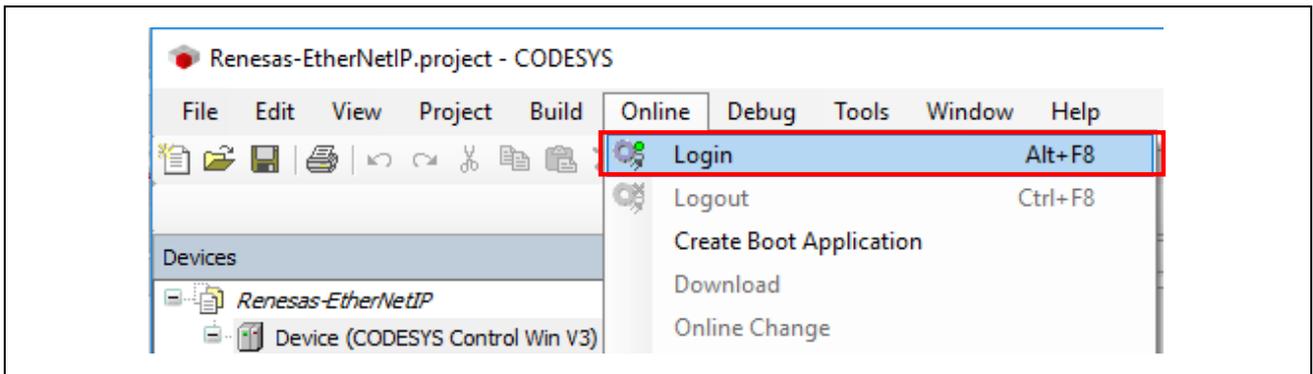
Select "Start" from the "Debug" menu. The project will automatically start on Windows
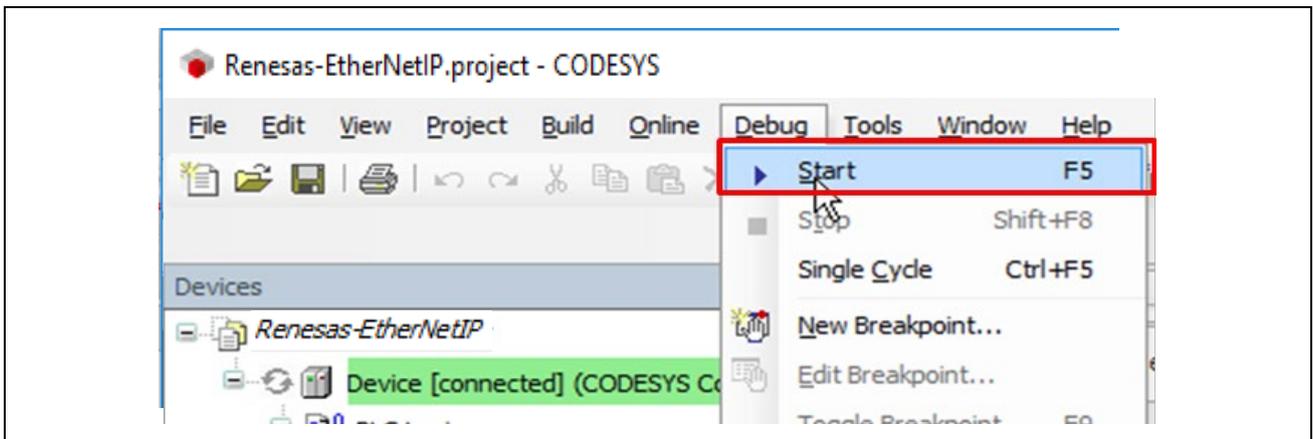
The states of connection will be indicated next to each device name. Successful connection is indicated by the
" 🔄 " icon
as shown in the tree view below
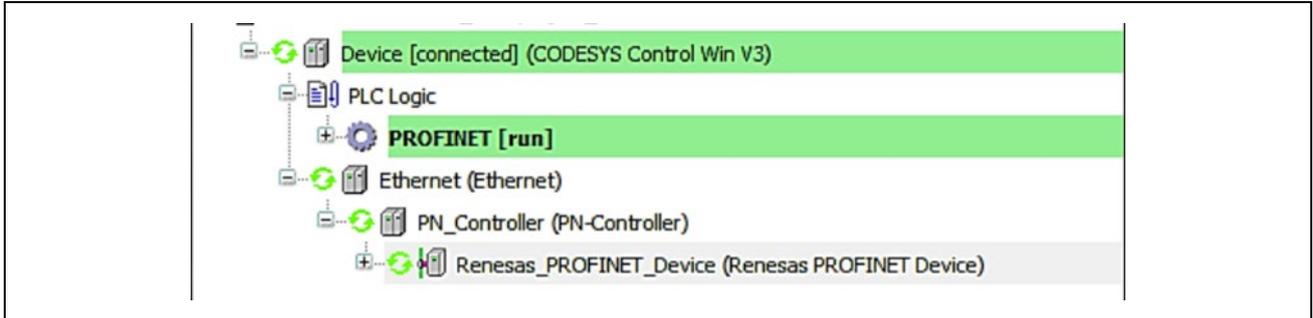
The icons indicating status of each device is listed below.

🔄   :   The application is connected to the PLC and is running.

🔄   :   The application is connected to the PLC but is not running.

⚠   :   Error. Check the error contents and the settings of the device.

❓   :   There is no device information in the device repository. Review the device information file and re-install it.

Revision History

| Rev. | Date | Description | | |
		Page	Summary	
1.00	July. 31, 2023	-	First edition issued	

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

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

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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

Contact information

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