RZ/N2L Group

BACnet Controller Sample Software

R01AN7237EJ0100
Rev.1.00
Mar.25.2024

Introduction

This document describes sample software for executing BACnet Controller profile (B-BC) of BACnet communication protocol for building automation (BA) on RZ/N2L.

Target Device

RZ/N2L Group

Contents

List of Abbreviations and Acronyms

In this document, the terms below are defined as follows:

Terms	Description
FSP	Flexible Software Package
RSK	Renesas Starter Kit
BA	Building Automation
BACnet	Building Automation and Control Networking
B-SS	BACnet Smart Sensor
B-BC	BACnet Building Controller
B-RTR	BACnet Router
B-OWS	BACnet Operator Workstation
Pmod	Peripheral module interface defined by Digilent Inc.
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
ANSI	American National Standards Institute
BIBB	BACnet Interoperability Building Blocks
API	Application Program Interface
APDU	Application Layer Protocol Data Unit
SNTP	Simple Network Time Protocol
BTL	BACnet Testing Laboratories
MS/TP	Master Slave / Token Passing
BIP	BACnet/IP
BVLCI	BACnet virtual link control information
NPCI	network protocol control information

Related documents

Document Type	Document Title	Document No.
Data Sheet	RZ/N2L Group Datasheet	R01DS0397EJ****
User's Manual	RZ/N2L Group User's Manual: Hardware	R01UH0955EJ****
User's Manual	Renesas Starter Kit+ for RZ/N2L User's Manual	R20UT4984EG****
Application Note	RZ/N2L Group TCP/IP lwIP Sample Program Package	R01AN6588EJ****
Application Note	RZ/N2L Group BACnet Sample Software	R01AN6789EJ****

1. Overview

1.1 Abstract

BACnet is the major communication protocol for building automation (BA). This document describes the configuration and usage of the sample software of BACnet controller (B-BC) with BACnet router function (B-RTR) that enables interoperation between BIP (BACnet/IP) device and MS/TP device in RZ/N2L, RZ processor for industrial network.

Fig. 1-1 RSK+ for RZ/N2L

Note that some figures in this document are reused from other BACnet application notes as long as it is not inconvenient for the reader to refer to them.

Since the sample software described in this document is BACnet Router, which is an interface connecting BACnet devices, the verification in this document uses an application on a PC as a BACnet Client and RZ/N2L BACnet Sample Software (R01AN6789xJ****) as an MS/TP slave at the other end, as shown in Fig. 1-2.

For the convenience of explanation, the BACnet Router described in this document is referred to as B-BC and the corresponding BACnet Slave is referred to as B-SS.

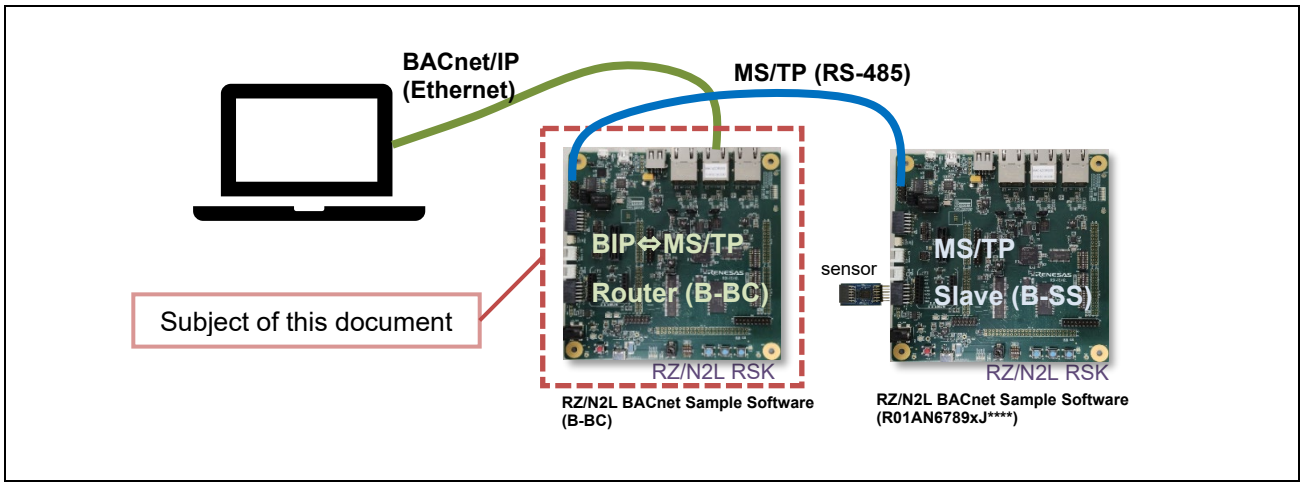

Fig. 1-2 Subject of this document and test setup

1.2 Operating Environment

1.2.1 Software Environment

The operating environment of this sample software is shown in Table 1-1

Table 1-1 Operating Environment

Category	Name	Version	Link	備考
RZ/N2L BACnet sample software	Sample Package			
IDE	e2studio	23.7.0	https://github.com/renesas/rzn-fsp/releases/download/v1.3.0/setup_rznfsp_v1_3_0_e2s_v2023-07.exe.	Included with e2studio installer
Flexible Software Package	FSP	1.3.0		Included with e2studio installer
GNU Arm Embedded Toolchain	GCC Toolchain	V9.3.1.20200408 (*1)		Included with e2studio installer
BACnet/IP Client Tool	VTS	3.6.7.0	Visual Test Shell for BACnet download | SourceForge.net	
BACnet/MSTP Master Tool	Yabe	1.3.0.0	Yet Another Bacnet Explorer download | SourceForge.net	
Packet analyzer	Wireshark	4.0.3	Wireshark · Download	
MS/TP Capture tool	mstpcap.exe		Capturing MS/TP packets – Optigo Networks (zendesk.com)	Integration with Wireshark.
Terminal Software	TeraTerm	4.108	Releases · TeraTermProject/teraterm (github.com)	

(*1). The recommended version of GCC Toolchain for FSP v1.3.0 is v12.2.1.arm-12-24, but this sample software is tested with v9.3.1.20200408.

1.2.2 Hardware Environment

This sample software is tested under the hardware environment of Table 1-2.

Table 1-2 Hardware Environment

Name	Type Name	Maker	Link	Note
Renesas Starter Kit+ for RZ/N2L	RTK9RZN2L0S00000BE	Renesas Electronics	www.renesas.com/rskrzn2l	RSK Board
Air Velocity Sensor Pmod™ Board	US082-FS3000EVZ	Renesas Electronics	US082-FS3000EVZ - Air Velocity Sensor Pmod™ Board (Renesas Quick-Connect IoT) | Renesas	Renesas Quick Connect IoT
USB/RS485 Convertor	BOB-09822	SparkFun	SparkFun USB to RS-485 Converter - BOB-09822 - SparkFun Electronics	2pcs (one for Yabe and one for Wireshark)

2. Hardware configuration

This section describes the hardware configuration of executing the sample software.

2.1 RSK Board Settings

When executing the sample software, configure the RSK board settings in Fig. 2-1

- The boot mode is xSPI0 boot mode.
- RS-485 half-duplex mode for BACnet MS/TP

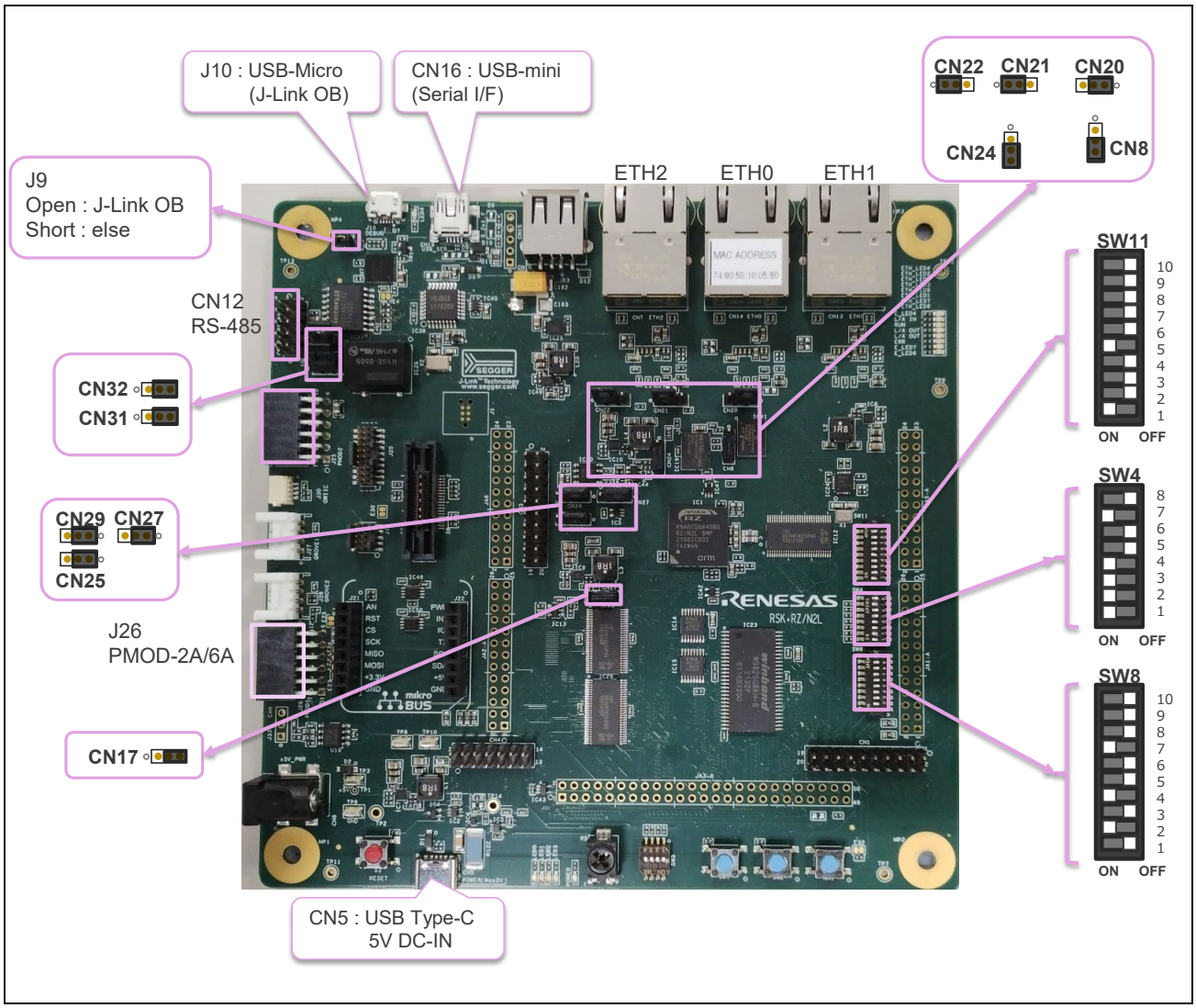

Fig. 2-1 Board Configuration

Table 2-1 DIPSW Settings

DIPSW		Setting	Default	Description
SW11	1	ON	ON	Enable LED_RED2 signal
	2	OFF	OFF	
	3	OFF	OFF	
	4	OFF	OFF	Enable RS485_RX signal
	5	ON	OFF	
	6	OFF	OFF	Disable P21_5、M2_VP、CAN_RX、ADTRG、P01_7
	7	OFF	OFF	
	8	OFF	OFF	
	9	OFF	OFF	
	10	OFF	OFF	
SW4	1	ON	ON	xSPI0 boot mode (x1 boot Serial flash)
	2	ON	ON	
	3	ON	ON	
	4	ON	ON	JTAG Authentication by Hash is disabled
	5	OFF	OFF	-
	6	OFF	OFF	Enables signals other the trace. (Motor, RS485, etc.) (TRACE_OPTION_SEL=H)
	7	ON	ON	Enables signals other than the external bus. (CAN, Emulator, I2C, etc.) (BSC_OPTION_SW=L)
	8	OFF	OFF	Enable SW3 (general purpose DIPSW)
SW8	1	OFF	OFF	Enable LED_GREEN
	2	ON	ON	
	3	OFF	OFF	
	4	ON	ON	Enable LED5
	5	OFF	OFF	
	6	OFF	OFF	Enable RS485_DE
	7	ON	OFF	
	8	OFF	ON	Disable P02_2, IRQ4, CAN_TX
	9	OFF	OFF	
	10	OFF	OFF	

Table 2-2 Jumper Settings

Jumper	Setting	Default	Description
J9	open	open	When using the J-Link® OB
	short		When using the external emulator or not using the emulator
CN31	2-3short	1-2short	RS485 Half Duplex
CN32	2-3short	1-2short	RS485 Half Duplex
CN20	1-2short	1-2short	When using 3 ports in the same PHY mode
CN21	1-2short	1-2short	When using 3 ports in the same PHY mode
CN22	1-2short	1-2short	When using 3 ports in the same PHY mode
CN24	2-3short	2-3short	Connect 1.8V Power rail to VCC1833_3. (Using XSPI0)
CN8	2-3short	2-3short	Select QSPI Serial Flash (QSPI_CS)
CN29	1-2short	1-2short	USB Serial (UART_USB_RX)

CN27	1-2short	1-2short	HyperRAM (IC41)
CN25	1-2short	1-2short	Other than the SHOST interface(Trace, SPI, external bus)
CN17	2-3short	2-3short	Select 1.8V for VCC1833_2

3. Sample Software

This chapter describes the structure and usage of the sample software.

Please note that FSP v1.1.0 in the following figures should be read as v1.3.0.

3.1 Folder structure

The folder structure of the sample software is shown below. The bolded text aim for indicating folders containing files that users will customize with this sample software.

RZN2L_BACnet_BBC_V1.0.0

├──BACNETOSS	BACnet Open Source Software
│ └──bacnet	**BACnet Protocol Stack**
├──rzn	RZ
│ ├──arm	ARM
│ │ └──CMSIS_5	CMSIS
│ ├──aws	AWS
│ │ └──amazon-freertos	FreeRTOS
│ ├──board	Board
│ │ └──rzn2l_rsk	RZ/N2L Renesas Starter Kit
│ └──fsp	Flexible Software Package
├──rzn_cfg	Configuration
│ ├──aws	AWS
│ ├──driver	r_xspi_qspi_cfg.h
│ └──fsp_cfg	FSP
│ └──bsp	Board Support Package
├──rzn_gen	Generated files
├──script	Linker Script
├──src	**User Thread Entry**
└──user	User files
├──oss	Other OSS
│ ├──amazon-freertos	AWS_OSS
│ └──lwip	lwIP_OSS
└──renesas	Renesas common files
├──application	**User lwIP application**, RTC module, Flash access for configurable data
├──module	User module
└──oss_deps	lwIP OSS dependencies

Fig.3-1 Folder Structure

3.2 Boot Sequence

Describes the boot procedure and memory allocation.

The boot mode of the sample software is xSPI0 x1 boot. The figure below shows the BSP tag in the Smart Configurator.

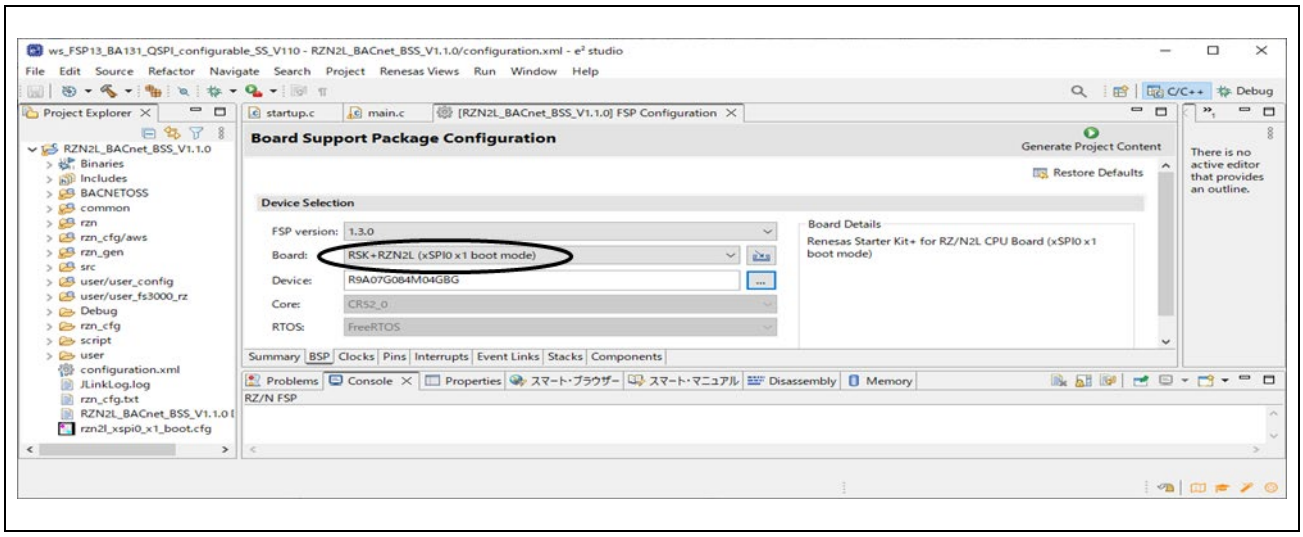

Fig.3-2 Boot mode

After downloading the program to the flash memory, the board operates independently by pressing the RESET button on the RSK board or turning the power ON without a debugger connection. You can still connect the debugger for evaluation. However, if jumper 9 (J9) of the RSK board is shorted, the debugger (J-Link OB) cannot be connected.

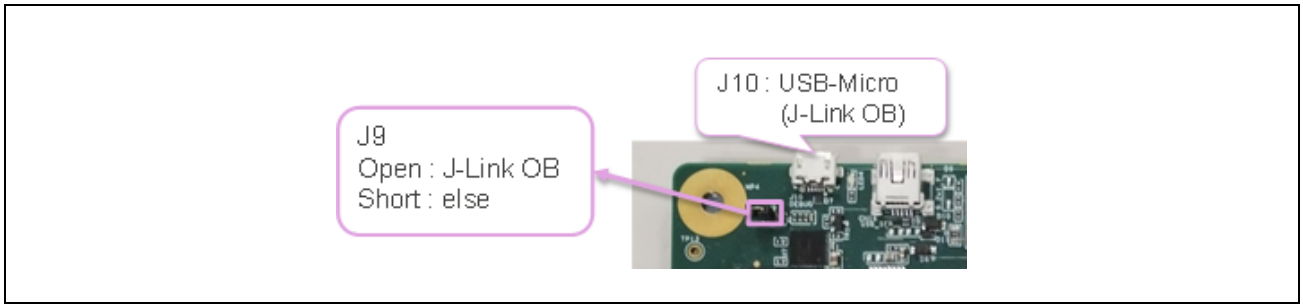

Fig.3-3 J9

This is the Smart Configurator screen showing the terminal settings (Pins tag) of the serial flash memory device. No changes are required because they have already been configured.

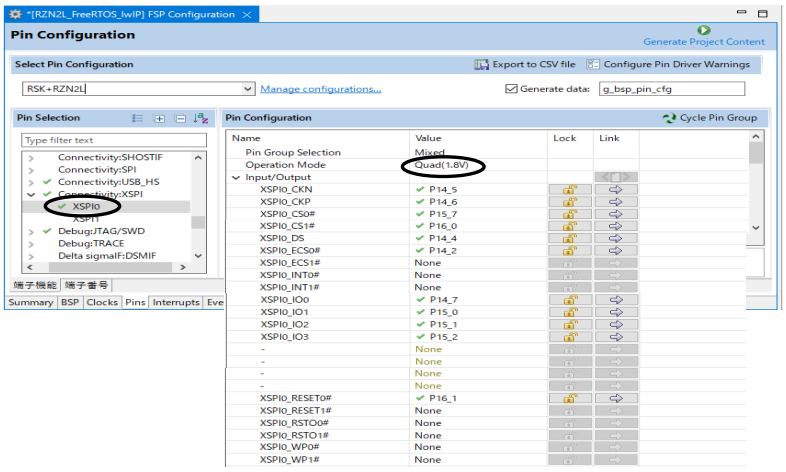

Fig.3-4 XSPI0 Pin Configuration

The order of memory writing in the boot sequence is shown in column "writing order" in the memory layout table below, and the memory is written in the order from (1) to (4). However, (5) is a storage area for data to be retained after the device is rebooted, regardless of the boot sequence. Therefore, they are written during running the system. See Chapter 5.3 for details.

Address	Memory	Content	Length		writing order	remarks
0x00000000	ATCM	intvec(64B)	0x00020000	128KB	(3)	Internal tightly coupled memory
0x00000040		Unused				
0x00000100		hal_entry,ROMdata				
0x00020000	Reserved area	-	-			
0x00100000	BTCM	Unused	0x00020000	128KB	(2)	Internal tightly coupled memory
0x00102000		Loader program(24KB)				
0x00108000		stack(60KB)				
0x00120000	Reserved area	-	-			
0x10000000	SYSTEM_RAM	Unused	0x00180000	1.5MB		
0x10180000	Reserved area	-	-			
0x30000000	SYSTEM_RAM_MIRROR	Body of program and data	0x00180000	1.5MB	(4)	Non-cached system RAM
0x30180000	Reserved area	-	-			
0x40000000	xSPI0_CS0_SPACE_MIRROR	Unused	0x04000000	64MB		
0x44000000	xSPI0_CS1_SPACE_MIRROR	Unused	0x04000000	64MB		
0x48000000	xSPI1_CS0_SPACE_MIRROR	Unused	0x04000000	64MB		
0x4C000000	xSPI1_CS1_SPACE_MIRROR	Unused	0x04000000	64MB		
0x50000000	CS0_SPACE_MIRROR	Unused	0x04000000	64MB		
0x54000000	CS2_SPACE_MIRROR	Unused	0x04000000	64MB		
0x58000000	CS3_SPACE_MIRROR	Unused	0x04000000	64MB		
0x5C000000	CS5_SPACE_MIRROR	Unused	0x04000000	64MB		
0x60000000	xSPI0_CS0_SPACE	Parameters for the loader(76B)	0x04000000	64MB	(1)	512M bits Serial Flash
0x6000004C		Loader program(24KB)				
0x6000604C		Body of program and data				
0x60FFFE00		Reserved area				
0x63FF8000		Unused				
0x63FFC000		Configurable properties			(5)	
0x63FFC092		Unused				
0x64000000	xSPI0_CS1_SPACE	Unused	0x04000000	64MB		
0x68000000	xSPI1_CS0_SPACE	Unused	0x04000000	64MB		
0x6C000000	xSPI1_CS1_SPACE	Unused	0x04000000	64MB		
0x70000000	CS0_SPACE	Unused	0x04000000	64MB		
0x74000000	CS2_SPACE	Unused	0x04000000	64MB		
0x78000000	CS3_SPACE	Unused	0x04000000	64MB		
0x7C000000	CS5_SPACE	Unused	0x04000000	64MB		

Fig.3-5 Memory layout

(1) Writing to a Serial Flash Memory Device

e2studio writes the download data to a serial flash ROM with an extended serial peripheral interface (xSPI) assigned to xSPI0_CS0_SPACE memory. The download data includes the loader parameters, the loader program and its data, the user program itself and its data.

The memory allocations are specified in the linker script file (fsp_xspi0_boot.ld), which is in the script folder.

*RZN2L_BACnet_BBC_V***\script\fsp_xspi0_boot.ld*

(2) Deploy the loader program to BTCM

The CPU automatically extracts the loader program included in the download data to the BTCM. After extraction, break in system_init() at the beginning of the initialization on the loader program.

(3) Deploy to ATCM

The initial setup of the loader program deploys the user program allocated in ATCM from flash memory to ATCM memory.

(4) Deploy to SYSTEM_RAM_MIRROR

The initial setup of the loader program deploys the user program allocated in the SYSTEM_RAM_MIRROR from the flash memory to the SYSTEM_RAM_MIRROR memory.

3.3 BACnet Stack

BACnet (Building Automation and Control Network) is the major communication protocol for Building Automation (BA) standardized in ASHRAE/ANSI Standard 135. Air conditioning, lighting, disaster prevention, access control, etc. can be integrated to control and monitor buildings.

BACnet devices are classified into different profiles according to their function and application, such as operator or controller. Major profiles include the central monitoring profile B-OWS (BACnet Operator Workstation), the controller profile B-BC (BACnet Building Controller), and the profile for various sensors B-SS (BACnet Smart Sensor).

The BACnet standard also defines standardized functional blocks (BIBBs) that should be supported for each profile, but it is permissible to support BIBBs that are not included there.

In this document, the sample software (RZN2L_BACnet_BBC_V1.0.0) is called B-BC, but it supports both B-RTR and B-BC profiles.

The function of B-RTR profile is to allow BACnet clients connected to BACnet/IP networks to access B-SS connected to MS/TP networks via B-BC (B-RTR profile). In this case, B-BC (B-RTR profile) works as a MS/TP master for B-SS.

B-BC Profile allows BACnet clients to instruct B-BC (B-BC Profile) to log sensor input values from B-SS or schedule B-SS LEDs to turn on or off at any given day and time. B-BC (B-BC Profile) also functions as a BACnet server for BACnet clients in the upper network layer such as B-OWS.

.

3.3.1 BACnet Protocol Stack

BACnet Protocol Stack (bacnet-stack) is an open-source stack for the BACnet communication protocol. This sample software is a port of BACnet Protocol Stack to RZ/N2L.

Base Version : bacnet-stack-1.3.1

Tags · bacnet-stack/bacnet-stack · GitHub

3.3.2 License

The license terms for the BACnet Protocol Stack are GPL with exception license. The original text is transcribed below for reference. Please refer BACnet Protocol Stack download | SourceForge.net for more information and comply with the license terms and conditions.

> *This BACnet protocol stack implementation is specifically designed for the embedded BACnet appliance, using a GPL with exception license (like eCos), which means that any changes to the core code that are distributed are shared, but the BACnet library can be linked to proprietary code without the proprietary code becoming GPL. Note that some of the source files are designed as skeleton or example or template files, and are not copyrighted as GPL.*
>
> *The text of the GPL exception included in each source file is as follows:*
>
> *"As a special exception, if other files instantiate templates or use macros or inline functions from this file, or you compile this file and link it with other works to produce a work based on this file, this file does not by itself cause the resulting work to be covered by the GNU General Public License. However the source code for this file must still be made available in accordance with section (3) of the GNU General Public License."*

3.3.3 Specifications

3.3.3.1 BACnet Revision

The protocol version and revision of the BACnet stack used in this sample software are as follows

・BACnet standard Protocol Version：1

・BACnet standard Protocol Revision：23

3.3.3.2 Service

The sequence of BACnet stack implemented in the sample software is service driven. Interoperability of BACnet devices is provided by the connection between users and providers via services (WhoIs, I-Am, ReadProperty, etc.).

There are two types of services: Unconfirmed and Confirmed. In the unconfirmed type, the provider does not return an Ack for the service requested by the user. On the other hand, confirmed type will return an Ack.

- **Users** of the sample software mean the following.

  In the case of BACnet devices that interconnect over BACnet/IP, it corresponds to the client.

  For BACnet MS/TP, it corresponds to the master.
- **Providers** mean the following.

  In the case of BACnet devices that interconnect over BACnet/IP, it corresponds to the server.

  For BACnet MS/TP, it corresponds to the slave.

The B-BC in this sample software is a server (provider) for BACnet client and a master (user) for B-SS slave.

Table 3-1 shows the services implemented in the sample software.(✓ : Applicable, blank : Not applicable)

Table 3-1 Implemented Services

BACnet service	Initiate[1]	Execute[2]
Who-Is	✓(Request)	✓
I-Am	✓ (Notification)	✓
Who-Has		✓
I-Have	✓ (Notification)	
ReadProperty	✓(Request)[3]	✓
WriteProperty	✓(Request)[3]	✓
DeviceCommunicationControl		✓
ReinitializeDevice		✓
AtomicReadFile		✓
AtomicWriteFile		✓
TimeSynchronization		✓
UTCTimeSynchronization		✓
SubscribeCOV		✓

BACnet service	Initiate[1]	Execute[2]
ConfirmedCOVNotification	✓ (Notification)	
UnconfirmedCOVNotification	✓ (Notification)	
ReadPropertyMultiple	✓(Request)[3]	✓
ReadPropertyConditional		
ReadRange		✓
WritePropertyMultiple	✓(Request)[3]	✓
GetAlarmSummary		✓
GetEventInformation		✓
GetEnrollmentSummary		
AcknowledgeAlarm		✓
ConfirmedEventNotification	✓（Notification）	
UnconfirmedEventNotification	✓（Notification）	
UnconfirmedTextMessage		
ConfirmedTextMessage		
AddListElement		
RemoveListElement		
CreateObject		
DeleteObject		
UnconfirmedPrivateTransfer		
ConfirmedPrivateTransfer		
VTOpen		
VTData		
VTClose		

✓ is applicable, blank is not applicable

1. Sends a BACnet service request or notification.
2. Execute the BACnet service and send a response (if a confirmed service is requested).
3. Service request to B-SS, but ReadPropertyMultiple and WritePropertyMultiple are unused.

The following is an overview of the implemented services

Table 3-2 Implemented service overview

BACnet service	Description
Who-Is	Who-Is service is used by BACnet users to know which other BACnet devices are sharing the network. Who-Is service is a broadcasted, unconfirmed (does not require an Ack) service.
I-Am	I-Am service is intended to respond to Who-Is service requests. However, I-Am service requests are broadcast transmissions that can be sent anytime. Receipt of Who-Is service request need not be preceded.
Who-Has	Who-Has service is used by BACnet users to identify BACnet devices with specific objects. Who-Has service is a broadcasted, unconfirmed type of service.
I-Have	I-Have service is available to respond to Who-Has service requests. However, I-Have service requests can be issued at any time. Receipt of Who-Has service requests need not be preceded; I-Have service is sent broadcast and is an unconfirmed type of service.
ReadProperty	ReadProperty service is used by BACnet users to request the value of one property of one BACnet object; the BACnet provider responds with Ack and returns the result.
WriteProperty	WriteProperty service is used by BACnet users to change the value of a specified property of one of the BACnet objects. BACnet provider responds with an Ack. If you want to restrict the write access to a specified property, an error with "Error Class" PROPERTY and "Error Code" WRITE_ACCESS_DENIED is returned.
WritePropertyMultiple	The WritePropertyMultiple service is used by BACnet users to set the value of one or more specified properties of one or more BACnet objects. BACnet provider responds with Ack. BACnet users can write any number of properties of any number of objects.
DeviceCommunicationControl	DeviceCommunicationControl service is used by BACnet users to instruct other BACnet devices to stop starting the BACnet service for a specified period of time. The period can be set to "indefinite". BACnet provider responds Ack. If the period is set to "indefinite", the communication must be activated again by DeviceCommunicationControl or ReinitializeDevice service.
ReinitializeDevice	ReinitializeDevice service is used by BACnet users to instruct other BACnet devices to reboot; the BACnet provider responds with Ack.
TimeSynchronization	TimeSynchronization service is used by BACnet users to broadcast or unicast the current time to other BACnet devices so that the devices' clocks can be synchronized with each other. The BACnet provider will not respond with Ack because this service is unconfirmed.
UTCTimeSynchronization	UTCTimeSynchronization service is used by BACnet users to broadcast or unicast the UTC current time relative to the meridian to other BACnet devices so that they can synchronize their clocks with each other. This service is unconfirmed, so the BACnet provider does not respond Ack. The BACnet provider receiving this service subtracts the UTC_Offset property value from the received UTC time to obtain the local time.
SubscribeCOV	SubscribeCOV service is used by BACnet users to receive notification of changes in property values for a particular object; the BACnet provider responds with Ack. ConfirmedCOVNotification and UnconfirmedCOVNotification services are used by BACnet providers to transmit change notifications. The choice of confirmed or unconfirmed type is specified by SubscribeCOV service from the BACnet user. If BACnet providers that receive SubscribeCOV service issue COV notifications, they always send unicast to the BACnet user.
ConfirmedCOVNotification	ConfirmedCOVNotification service is used by BACnet providers to notify subscribers (BACnet users) of changes in the property values of a particular object; the BACnet user responds with Ack.
UnconfirmedCOVNotification	UnconfirmedCOVNotification service is used by BACnet providers to unicast notifications to subscribers of changes in certain object property values or to broadcast notifications of certain object properties (such as outdoor temperature) to many devices.

BACnet service	Description
ReadPropertyMultiple	ReadPropertyMultiple service is used by BACnet users to request the value of one or more specified properties of one or more BACnet objects. BACnet users can read any number of properties of any number of objects. In particular, the property identifier ALL can be used to retrieve all the properties of the object and its values at once.
ReadRange	ReadRange service is used by BACnet users; the BACnet provider reads a specific range of data items in the LogBuffer property of the TrendLog object and responds with Ack.
GetAlarmSummary	GetAlarmsummary service is used by BACnet users to obtain a summary of "active alarms". The BACnet provider responds Ack with an EventState property whose object value does not equal NORMAL and a NotifyType property whose value is ALARM.
GetEventInformation	GetEventInformation service is used by BACnet users to get an overview of all "active event states". BACnet providers will respond Ack with information about the object that notified them of the event.
AcknowledgeAlarm	AcknowledgeAlarm service is used by the BACnet user to tell the BACnet provider that the BACnet user has acknowledged the confirmedEventNotification service or UnconfirmedEventNotification service notified by the BACnet provider. The BACnet provider responds with Ack.
AtomicReadFile	AtomicReadFile service is used by BACnet users to read the BACnet provider's configuration data file to keep a backup of the file. The BACnet provider responds Ack.
AtomicWriteFile	The AtomicWriteFile service is used by BACnet users. The backup file of the configuration data is transferred to the BACnet provider, who restores the configuration data; the BACnet provider responds with Ack.

3.3.3.3 Restrictions

The released version V1.0.0 of this sample software has the following restrictions.

- ✓ This sample software does not implement all the functions required for a B-BC device, and BTL testing has not been carried out.
- ✓ This sample software is intended to connect to upper devices of B-BC with BACnet/IP protocol and lower devices with BACnet MS/TP protocol (Fig.3-6). As shown in Table 3-3, the B-BC in this sample software does not work as a BACnet/IP client device and cannot connect to subordinate devices of BACnet server. It also cannot be a MS/TP slave device.

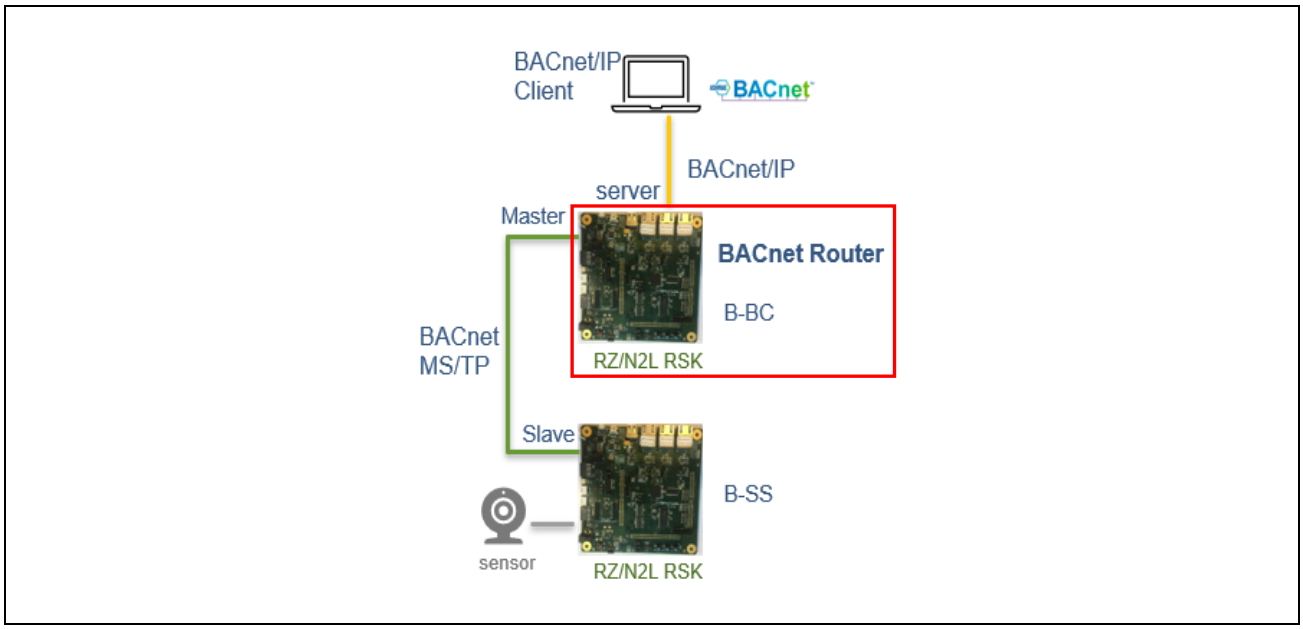

Fig.3-6 BACnet devices that can be connected to B-BC

Table 3-3 Connectable device configuration

Connect to \ The Sample Software				BACnet/IP		BACnet MS/TP	
				Client	Server	Master	Slave
Upper layer device	VTS	BACnet/IP	Client		✓		
	Yabe				✓		
		BACnet MS/TP	Master			✓(*1)	
Lower layer device	B-SS	BACnet/IP	Server				
		BACnet MS/TP	Slave			✓	

✓ is applicable, blank is not applicable

(*1) B-BC can transfer TOKEN between other MS/TP master devices, but does not support the execution of services such as ReadProperty and WriteProperty.

3.3.3.4 BIBBs

BIBBs (BACnet Interoperability Building Blocks) defines a set of services that apply to interoperating BACnet devices. "A" and "B" devices are defined, with the "A" device representing the BACnet user and the "B" device representing the BACnet provider.

BACnet standard (Annex L) defines various device profiles that describe the characteristics of each device, such as B-OWS (BACnet Operator WorkStation) and B-SS (BACnet Smart Sensor).

B-BC (BACnet Building Controller) in this sample software have both "A" and "B" characteristics.

The implemented BIBBs of the sample software is as follows. (✓ : Applicable, blank : Not applicable)

Table 3-4 Implemented BIBBs (B-BC Profile)

BIBB Class	BIBB	BACnet Service	Initiate[1]	Execute[2]	B-BC Standardized[3]
DataSharing	DS-RP-A,B	ReadProperty	✓	✓	✓
	DS-WP-A,B	WriteProperty	✓	✓	✓
	DS-RPM-A,B	ReadPropertyMultiple	✓[4]	✓	✓
	DS-WPM-A,B	WritePropertyMultiple	✓[4]	✓	✓
	DS-COV-B	SubscribeCOV		✓	
		ConfirmedCOVNotification	✓		
		UnconfirmedCOVNotification	✓		
Device & Network Management	DM-DDB-A,B	Who-Is	✓	✓	✓
		I-Am	✓	✓	✓
	DM-DOB-B	Who-Has		✓	✓
		I-Have	✓		✓
	DM-DCC-B	DeviceCommunicationControl		✓	✓
	DM-TS-B	TimeSynchronization / UTCTimeSynchronization		✓	✓
	DM-RD-B	ReinitializeDevice		✓	✓
	DM-BR-B	AtomicReadFile		✓	✓
		AtomicWriteFile		✓	✓
		ReinitializeDevice		✓	✓
Alarm & Event Management	AE-N-I-B	ConfirmedEventNotification	✓		✓
		UnconfirmedEventNotification	✓		✓
	AE-ACK-B	AcknowledgeAlarm		✓	✓
	AE-INFO-B	GetEventInformation		✓	✓
Scheduling	SCHED-I-B	ReadProperty		✓	✓
		WriteProperty		✓	✓
		TimeSynchronization / UTCTimeSynchronization		✓	✓
	SCHED-E-B	ReadProperty		✓	✓
		WriteProperty	✓	✓	✓
		TimeSynchronization / UTCTimeSynchronization		✓	✓

Trending	T-VMT-I-B	ReadRange		✓	✓
	T-ATR-B[5]	ConfirmedEventNotification	✓		✓
		UnconfirmedEventNotification	✓		✓
		ReadRange		✓	✓

✓ is applicable, blank is not applicable

1. Sends a BACnet service request or notification.
2. Execute the BACnet service and send a response (if a confirmed service is requested).
3. BIBBs which is defined as normalized for B-BC in ANNEX L.4 of BACnet standards.
4. It is used to request service to B-SS, but ReadPropertyMultiple and WritePropertyMultiple are not used.
5. BACnet service used by T-ATR-B is implemented but BUFFER_READY event algorithm is not yet supported.

Table 3-5 Implemented BIBBs (B-RTR Profile)

BIBB Class	BIBB	BACnet Service	Initiate[1]	Execute[2]	B-RTR Standardized[3]
DataSharing	DS-RP-B	ReadProperty		✓	✓
	DS-WP- B	WriteProperty		✓	✓
Device & Network Management	DM-DDB- B	Who-Is		✓	✓
		I-Am	✓		✓
	DM-DOB-B	Who-Has		✓	✓
		I-Have	✓		✓
BIBB Class	BIBB	BACnet Network Layer Message	Initiate[1]	Execute[2]	B-RTR Standardized[3]
Device & Network Management	NM-RC-B	Who-Is-Router-To-Network	✓	✓	✓
		I-Am-Router-To-Network	✓	✓	✓
		Reject-Message-To-Network	✓	✓	✓
		Router-Busy-To-Network	✓	✓	✓
		Router-Available-To-Network	✓	✓	✓
		Network-Number-Is	✓	✓	✓
		What-Is-Network-Number		✓	✓

✓ is applicable, blank is not applicable

1. Sends a BACnet service request or notification.
2. Execute BACnet service and respond (if confirmed type service is requested) or accept messages.
3. which is defined as normalized for B-RTR in ANNEX L.7 of BACnet standards.

Outlines of the implemented BIBBs in the B-SS sample software is as follows.

Table 3-6 Outlines of the implemented BIBB

BIBBs	Description
DS-RP-A	Device A is one property user from device B.
DS-RP-B	Device B returns one property value to device A.
DS-WP-A	Device A sets one property of Device B.
DS-WP-B	Device B writes value from device A to one property.
DS-RPM-A	Device A is a data user from Device B and requests multiple properties at once.

BIBBs	Description
DS-RPM-B	Device B returns multiple property values at once to device A.
DS-WPM-A	Device A sets multiple properties on Device B at once.
DS-WPM-B	Device B writes multiple values from device A to multiple properties at once.
DS-COV-B	Device B accepts COV notification subscription from Device A and sends COV notification to Device A.
DM-DDB-A	Device A makes an identification request to another device and interprets the device's announcement.
DM-DDB-B	Device B responds to the identification request from Device A.
DM-DOB-B	Device B responds to an identification request from Device A with the specified object.
DM-DCC-B	Device B responds to a request from Device A to stop communication.
DM-TS-B	Device B accepts time synchronization from Device A.
DM-RD-B	Device B responds to the reinitialization request from Device A.
DM-BR-B	Device B provides the setting file to Device A, which in turn writes the file to Device B so that its settings can be recovered in case of a failure of Device B.
AE-N-I-B	Device B generates notifications of alarms and other events.
AE-ACK-B	Device B responds to an acknowledgement from Device A for an alarm/event notification that has been sent.
AE-INFO-B	Device B provides event information to Device A.
SCHED-I-B	B devices provide a date and time schedule of the value of specified properties of certain objects in the device.
SCHED-E-B	Device B provides a date and time schedule of the values of specified properties of certain objects on other devices.
T-VMT-I-B	Device B collects trend log data records in the internal buffer.
T-ATR-B	Device B uses the BUFFER_READY event algorithm in the trend log object to notify Device A that the trend log buffer has acquired a given number of data samples.
NM-RC-B	B devices need to respond to router management commands and meet the BACnet router requirements in the Standards.

3.3.3.5 Implemented service as A-Device

The B-BC in this sample software has device A functions as defined in BIBBs. This sample software supports the following BIBBs, service request destinations, and Ack senders as B-BC A devices.

Refer to the links in the " Reference" column for details.

Table 3-7 Support services for B-BC as “A” device

BIBBs	Service	Send to	Receive from	Reference
DS-RP-A	ReadProperty	MS/TP slave		**4.6.2.1 Trending &** ReadRange
	Complex-Ack		MS/TP slave	
DS-WP-A	WriteProperty	MS/TP slave		**4.6.2.2 Scheduling**
	Simple-Ack		MS/TP slave	
DS-RPM-A	ReadPropertyMultiple	MS/TP slave		Unused(*1)
	Complex-Ack		MS/TP slave	
DS-WPM-A	WritePropertyMultiple	MS/TP slave		
	Simple-Ack		MS/TP slave	
DM-DDB-A	Who-Is	BIP client		**4.6.3 EventNotification /** GetEventInformation / AcknowledgeAlarm
	I-AM		All	

(*1) ReadPropertyMultiple and WritePropertyMultiple are not used when making service requests to B-SS.

3.3.3.6 Object

A BACnet device consists of a set of objects. An object is represented by an object type and an instance number from 0 to 4194303, which is called an object ID. However, the number 4194303 means invalid and is not used.

The device itself is also an object and is defined in Device object; the object ID of the device is called the device ID. Each BACnet device is required to have a Device object.

Furthermore, objects consist of a set of properties of various data types, and the B-SS accesses hardware to read and write these properties.

The implemented Objects of the sample software is as follows. (✓ : Applicable, blank : Not applicable)

Table 3-8 Implemented Objects in the B-SS sample software

BACnet Object Type	Object ID	Implementation
Accumulator		
Analog Input	Analog Input, 0	✓
	Analog Input, 1	✓
Analog Value	Analog Value, 0	✓
	Analog Value, 1	✓
Averaging		
Binary Output	Binary Output, 0	✓

BACnet Object Type	Object ID	Implementation
	Binary Output, 1	✓
Binary Value	Binary Value, 0	✓
	Binary Value, 1	✓
Calendar		
Command		
Device	Device, 10	✓
Event Enrollment		
File	File,0	✓
Group		
Life Safety Point		
Life Safety Zone		
Loop		
Multi state Input		
Multi state Output		
Multi state Value	Multi state Value, 0	✓
	Multi state Value, 1	✓
Notification Class	Notification Class,0	✓
Program		
Pulse Converter		
Schedule	Schedule,0	✓
Trend Log	Trend Log,0	✓
Access Door		
Event Log		
Load Control		
Structured View		
Trend Log Multiple		
Access Point		
Access Zone		
Access User		
Access Rights		
Access Credential		
Credential Data Input		
CharacterString Value		
DateTime Value		
Large Analog Value		
BitString Value		

BACnet Object Type	Object ID	Implementation
OctetString Value		
Time Value		
Integer Value		
Positive Integer Value	Positive Integer Value, 0	✓
	Positive Integer Value, 1	✓
Date Value		
DateTime Pattern Value		
Time Pattern Value		
Date Pattern Value		
Network Security		
Global Group		
Notification Forwarder		
Alert Enrollment		
Channel		
Lighting Output		
Network Port	Network Port, 1	✓
	Network Port, 2	✓
Binary Lighting Output		

✓ is applicable, blank is not applicable

Outlines of the implemented object types are as follows.

Table 3-9 Outlines of the implemented object types

BACnet Object Type	Description
Analog Input	Analog Input object has properties that represent analog inputs from hardware.
Analog Output	Analog Output object has properties that represent analog output to hardware.
Analog Value	Analog Value object has properties representing analog values that reside in the memory of the BACnet device.
Binary Input	Binary Input object is input from hardware and has property representing two states: ACTIVE or INACTIVE.
Binary Output	Binary Output object is the output to the hardware and has properties representing two states: ACTIVE or INACTIVE.
Binary Value	Binary Value object has properties that represent two states, ACTIVE or INACTIVE, resident in the memory of the BACnet device.
Device	BACnet device must have one Device object for sure: it has Object_Identifier property that is unique to the BACnet device. This is also unique to the entire network.
Multi state Value	Multi state Value object has properties that represent one or more states resident in the memory of the BACnet device.
Positive Integer Value	Positive Integer Value object has properties that allow the BACnet device to access any kind of unsigned data value.
Network Port	Network Port object has properties that represent the network configuration of the BACnet device and must contain at least one network port object.
File	File object has properties of a data file that can be accessed using the file service.

BACnet Object Type	Description
Notification Class	Notification Class objects have properties necessary for event notification within the BACnet system.
Schedule	Schedule objects have properties to link the writing of specified values to specified properties of a particular object with a recurring schedule that repeats within a specified date range, at any given time on any given date.
Trend Log	The Trend Log object monitors the properties of the referenced object and saves the property value and timestamp to an internal buffer represented in the LogBuffer property when the defined conditions are met. Reading the LogBuffer property requires the ReadRange service.

3.3.3.7 Property

BACnet objects have various data elements called "Property," and each property is accessed through services. Properties defined as Required (R) in the Conformance Code are properties that must be supported when an object is supported. Supporting of optional properties (O) is arbitrary, but depending on the conditions, several properties have to be supported (or unsupported).

Table 3-10 to **Table 3-22** show the supporting objects and its properties included in this sample software. The legends in the tables are shown below.

1. CC : Conformance Code

R: Required, O: Optional, W: Writable

2. Configurable

x(*1) Configurable by both Initial Configuration Command and WriteProperty Service

Once Configurable property values are written to Flash memory by using Initial Configuration Command in section 5.3, they will be reflected as initial values in the properties when the board is reset. When WriteProperty or WritePropertyMultiple services are executed, the values are also written to Flash memory.

x(*2) Configurable by Initial Configuration Command

Once Configurable property values are written to Flash memory by using Initial Configuration Command in section 5.3, they will be reflected as initial values in the properties when the board is reset.

x(*3) Configurable by WriteProperty Service

When WriteProperty or WritePropertyMultiple services are executed, the property values are also written to Flash memory. They will be reflected as initial values in the properties when the board is reset.

3. Access

R: Readable using ReadProperty or ReadPropertyMultiple services

W: Writeable using WriteProperty or WritePropertyMultiple services

Table 3-10 Analog Input Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Present_Value	R		R/W
Description	O		R
Status_Flags	R		R
Event_State	R		R
Reliability	O		R
Out_Of_Service	R	✓(*1)	R/W
Units	R		R/W
COV_Increment	O		R/W
Time_Delay	O		R/W
Notification_Class	O		R/W
High_Limit	O		R/W
Low_Limit	O		R/W
Deadband	O		R/W
Limit_Enable	O		R/W
Event_Enable	O		R/W
Acked_Transitions	O		R
Notify_Type	O		R/W
Event_Time_Stamps	O		R
Property_List	R		R

Table 3-11 Analog Value Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Present_Value	R		R/W
Description	O		R
Status_Flags	R		R
Event_State	R		R
Out_Of_Service	R	✓(*1)	R/W
Units	R		R/W
COV_Increment	O		R/W
Time_Delay	O		R/W
Notification_Class	O		R/W
High_Limit	O		R/W
Low_Limit	O		R/W
Deadband	O		R/W
Limit_Enable	O		R/W
Event_Enable	O		R/W
Acked_Transitions	O		R
Notify_Type	O		R/W
Event_Time_Stamps	O		R
Property_List	R		R

Table 3-12 Binary Output Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Present_Value	W		R/W
Description	O		R
Status_Flags	R		R
Event_State	R		R
Reliability	O		R
Out_Of_Service	R	✓(*1)	R/W
Polarity	R		R/W
Inactive_Text	O		R
Active_Text	O		R
Priority_Array	R		R
Relinquish_Default	R		R
Current_Command_Priority	R		R
Property_List	R		R

Table 3-13 Binary Value Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W

Object_Type	R		R
Present_Value	R		R/W
Description	O		R
Status_Flags	R		R
Event_State	R		R
Reliability	O		R
Out_Of_Service	R	✓(*1)	R/W
Priority_Array	O		R
Relinquish_Default	O		R
Current_Command_Priority	O		R
Property_List	R		R

Table 3-14 File Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
File_Type	R		R
File_Size	R		R/W
Modification_Date	R		R
Archive	W		R/W
Read_Only	R		R
File_Access_Method	R		R
Description	O		R
Status_Flags	R		R

Table 3-15 Notification Class Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Notification_Class	R		R
Priority	R		R/W
Ack_Required	R		R/W
Recipient_List	R		R/W
Description	O		R

Table 3-16 Schedule Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Present_Value	R		R/W
Effective_Period	R		R/W
Schedule_Default	R		R/W
List_Of_Object_Property_References	R		R/W
Priority_For_Writing	R		R

Status_Flags	R		R
Reliability	R		R
Out_Of_Service	R	✓(*1)	R/W
Weekly_Schedule	R		R/W
Description	O		R

Table 3-17 Multi-state Value Object Type

Property Identifier	CC[1]	Configurable	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Present_Value	R		R/W
Description	O		R
Status_Flags	R		R
Event_State	R		R
Out_Of_Service	R	✓(*1)	R/W
Number_Of_States	R		R
State_Text	O		R
Property_List	R		R

Table 3-18 Trend Log Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Enable	W		R/W
Start_Time	O		R/W
Stop_Time	O		R/W
Log_DeviceObjectProperty	O		R/W
Log_Interval	O		R/W
Stop_When_Full	R		R/W
Buffer_Size	R		R
Log_Buffer	R		R
Record_Count	W		R/W
Total_Record_Count	R		R
Logging_Type	R		R/W
Align_Intervals	O		R/W
Interval_Offset	O		R/W
Trigger	O		R/W
Status_Flags	R		R
Event_State	R		R
Description	O		R
Property_List	R		R

Table 3-19 Positive Integer Value Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R

Object_Name	R		R/W
Object_Type	R		R
Present_Value	R		R/W
Status_Flags	R		R
Out_Of_Service	O	✓(*1)	R/W
Units	R		R
Event_State	O		R
Description	O		R
Property_List	R		R

Table 3-20 Network Port Object Type(for BIP)

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Status_Flags	R		R
Reliability	R		R
Out_Of_Service	R		R
Network_Type	R		R
Protocol_Level	R		R
Changes_Pending	R		R
Description	O		R
MAC_Address	O	✓(*2)	R
BACnet_IP_Mode	O	✓(*3)	R/W
IP_Address	O	✓(*2)	R
BACnet_IP_UDP_Port	O	✓(*2)	R
IP_Subnet_Mask	O		R
IP_Default_Gateway	O	✓(*2)	R
IP_DNS_Server	O		R
FD_BBMD_Address	O	✓(*3)	R/W
FD_Subscription_Lifetime	O	✓(*3)	R/W
Property_List	R		R

Table 3-21 Network Port Object Type(for MSTP)

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R		R
Object_Name	R		R/W
Object_Type	R		R
Status_Flags	R		R
Reliability	R		R
Out_Of_Service	R		R
Network_Type	R		R
Protocol_Level	R		R
Network_Number	O	✓(*3)	R/W
Network_Number_Quality	O		R
Changes_Pending	R		R
Apdu_Length	O		R

Link_Speed	R		R
Description	O		R
MAC_Address	O	✓(*2)	R
Max_Master	O		R/W
Max_Info_Frames	O		R/W
Property_List	R		R

Table 3-22 Device Object Type

Property Identifier	CC[1]	Configurable[2]	Access[3]
Object_Identifier	R	✓(*2)	R
Object_Name	R	✓(*1)	R/W
Object_Type	R		R
System_Status	R		R
Vendor_Name	R		R
Vendor_Identifier	R		R
Model_Name	R		R
Firmware_Revision	R		R
Application_Software_Version	R		R
Location	O		R/W
Description	O		R/W
Protocol_Version	R		R
Protocol_Revision	R		R
Protocol_Services_Supported	R		R
Protocol_Object_Types_Supported	R		R
Object_List	R		R
Max_APDU_Length_Accepted	R		R
Segmentation_Supported	R		R
Local_Time	O		R
Local_Date	O		R
UTC_Offset	O	✓(*2)	R/W
Daylight_Savings_Status	O		R
APDU_Timeout	R		R/W
Number_Of_APDU_Retries	R		R/W
Device_Address_Binding	R		R
Database_Revision	R		R
Active_COV_Subscriptions	O		R
Max_Master	O		R/W
Max_Info_Frames	O		R/W
Property_List	R		R

3.4 Installation of Development Environment

3.4.1 e2studio

3.4.1.1 Install

Download the version listed in Table 1-1 and install it on your PC. The latest version has a downloadable installer that includes FSP, e2studio, and the GCC toolchain as a single package.

・Double-click the downloaded “setup_rznfsp_v1_3_0_e2s_v2023-07.exe”.

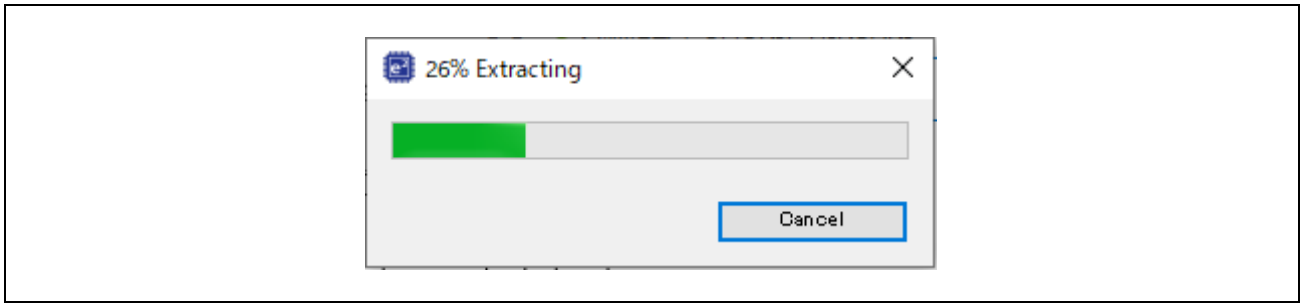

Fig.3-7 e2studio Install (1)

・Select Users

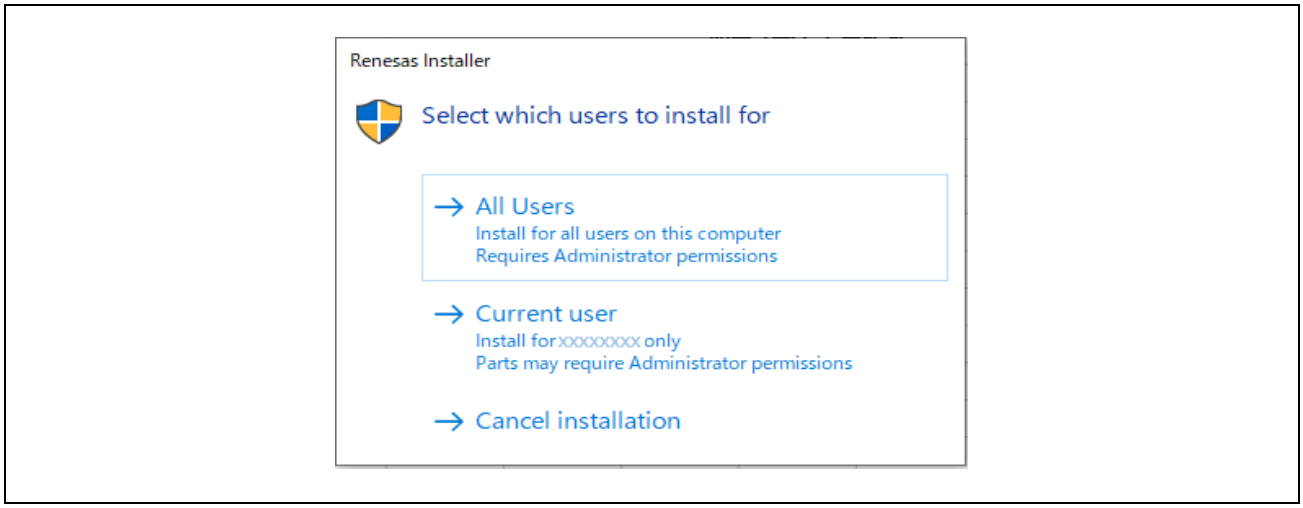

Fig.3-8 e2studio Install (2)

・Select “Install”

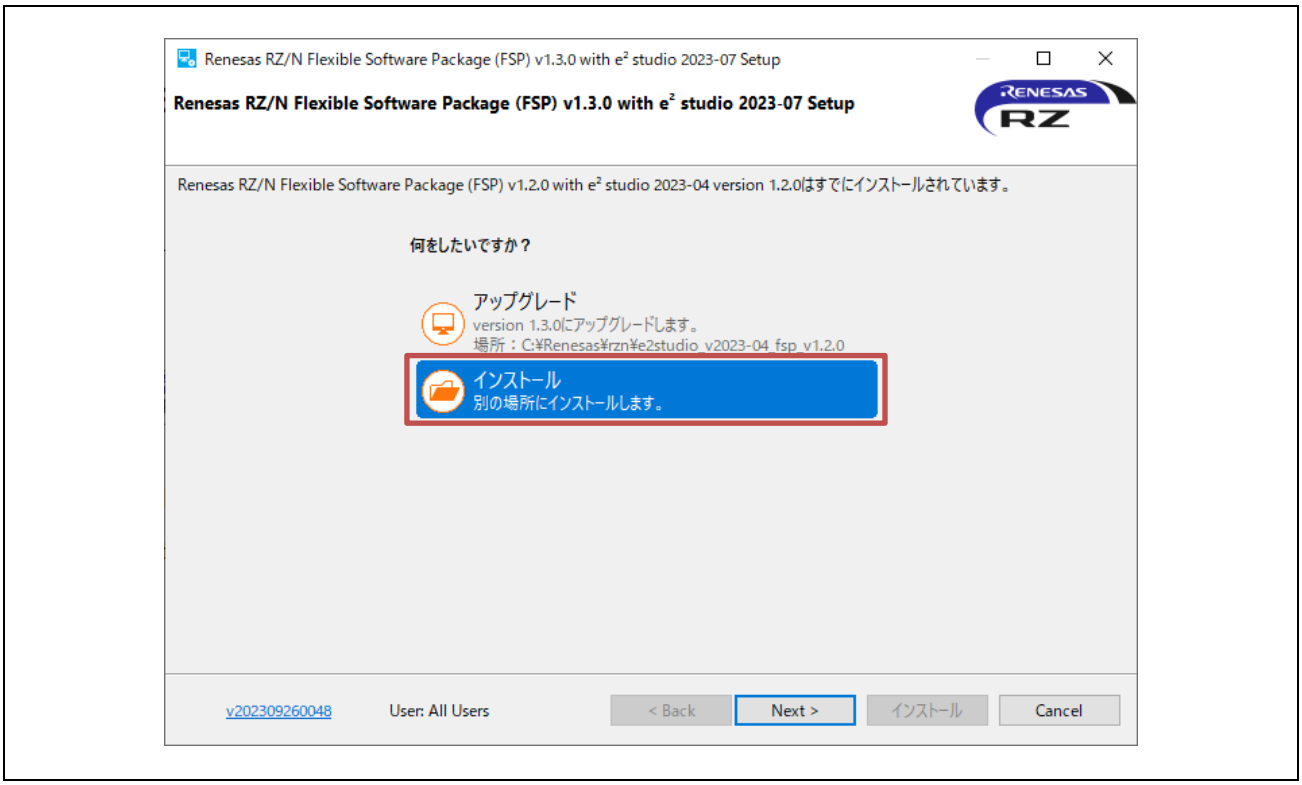

Fig.3-9 e2studio Install (3)

・Select Install Type

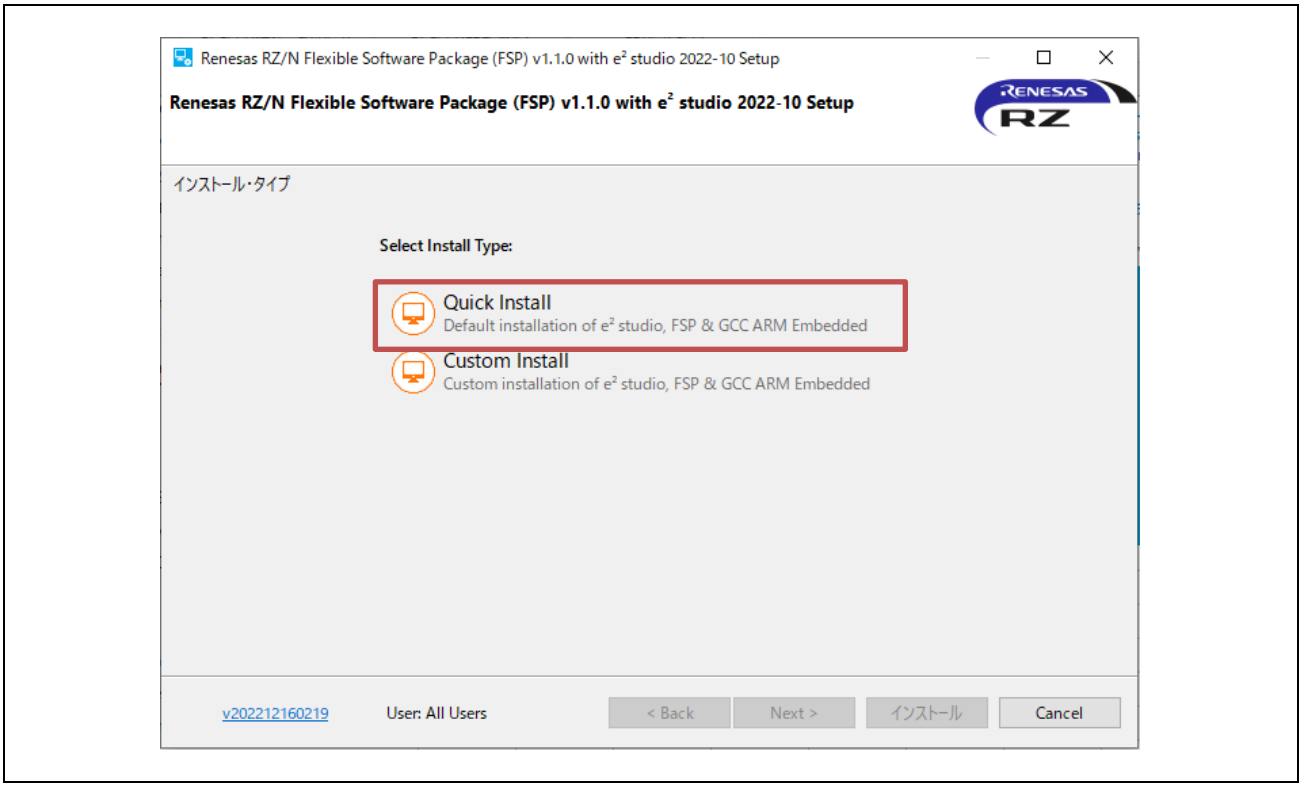

Fig.3-10 e2studio Install (4)

・Select Install folder

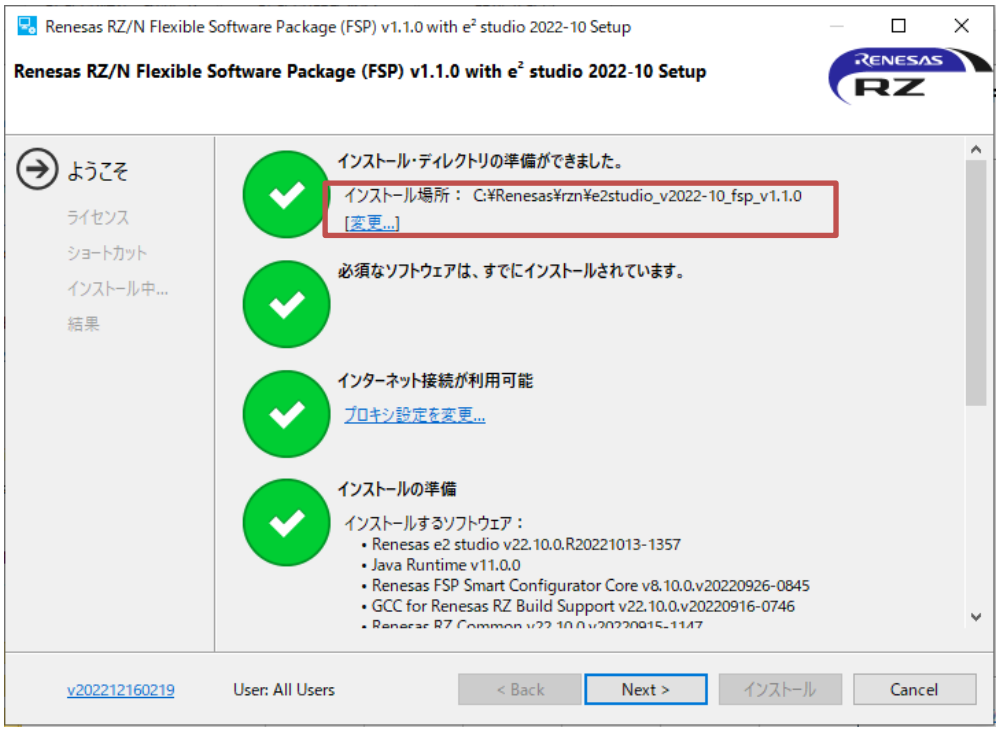

Fig.3-11 e2studio Install (5)

・Check and Click “Next”

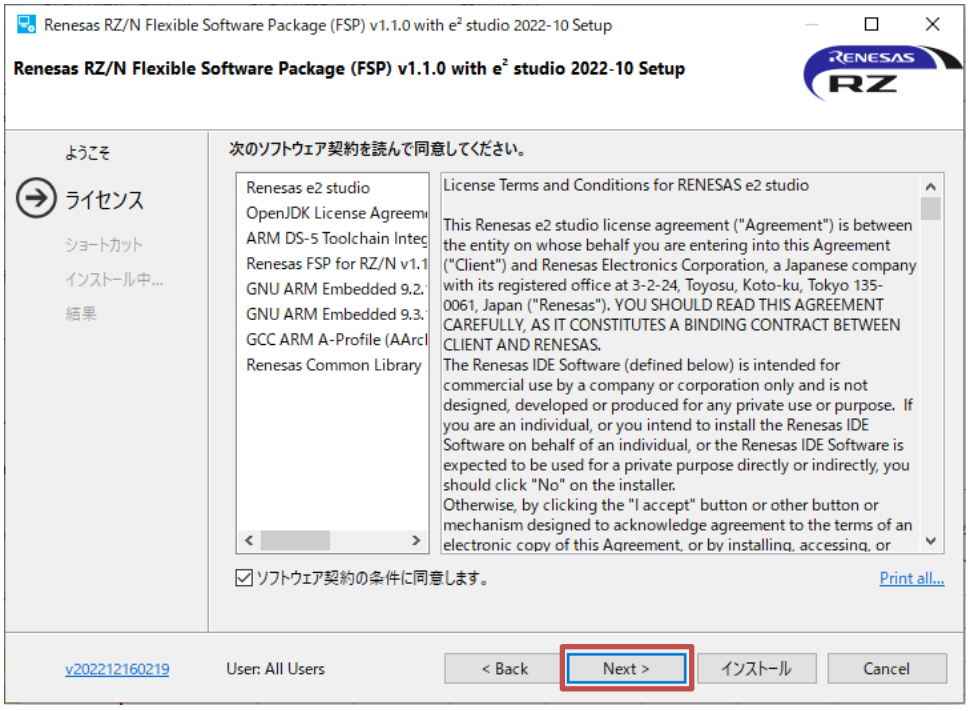

Fig.3-12 e2studio Install (6)

・Click “Install”

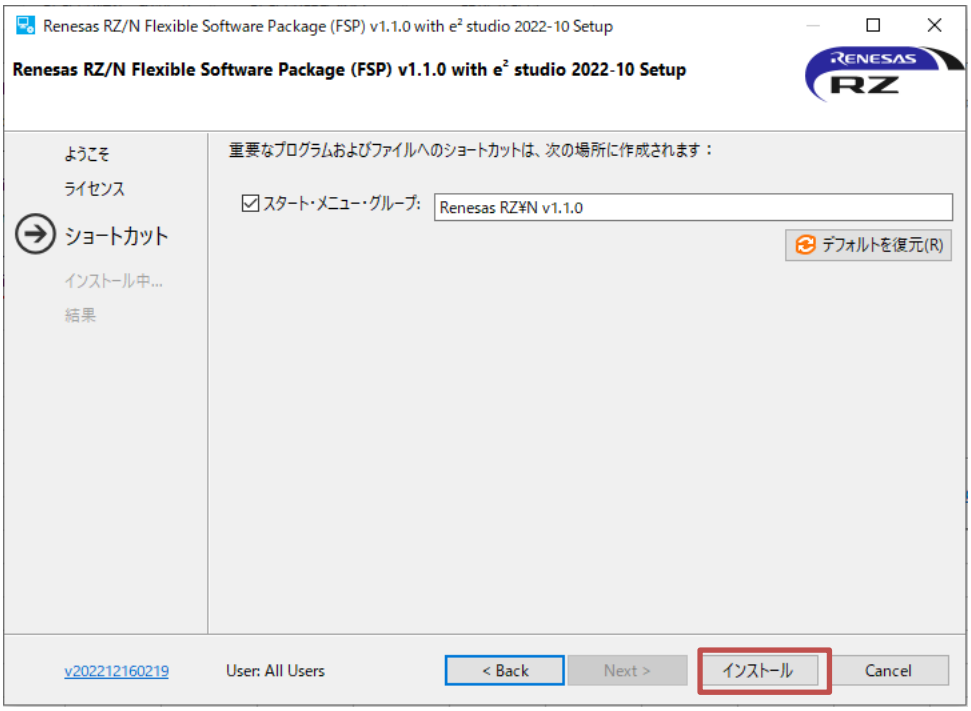

Fig.3-13 e2studio Install (7)

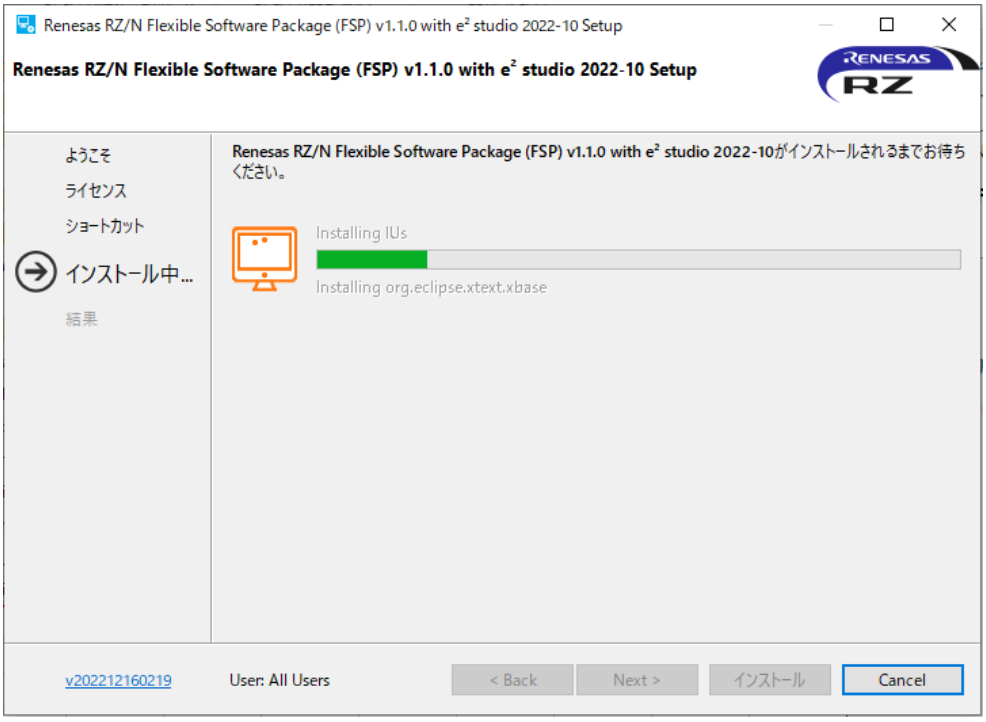

Fig.3-14 e2studio Install (8)

・Click “OK”

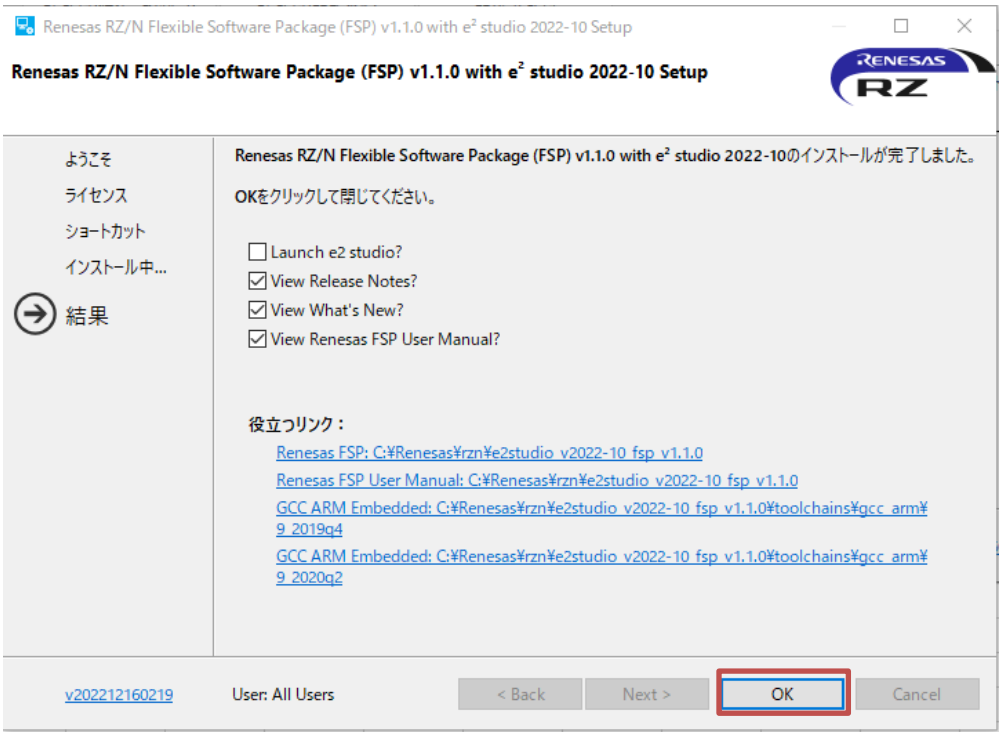

Fig.3-15 e2studio Install (9)

3.4.1.2 Project start-up

(1) Unzip package

First, unzip the archived package of this sample software (RZN2L_BACnet_BBC_V***.zip) and store it in arbitrary folder. Because e2studio cannot recognize project properly if file path is too long in the folder hierarchy, place it in shorter path. Also, do not use multi-byte character, such as Japanese, in the folder path.

(2) Execute e2studio

Execute "e2studio.exe" to start e2studio in the following folder (default case) installed:

\Renesas\rzn\e2studio_v2023-07_fsp_v1.3.0\eclipse¥e2studio.exe

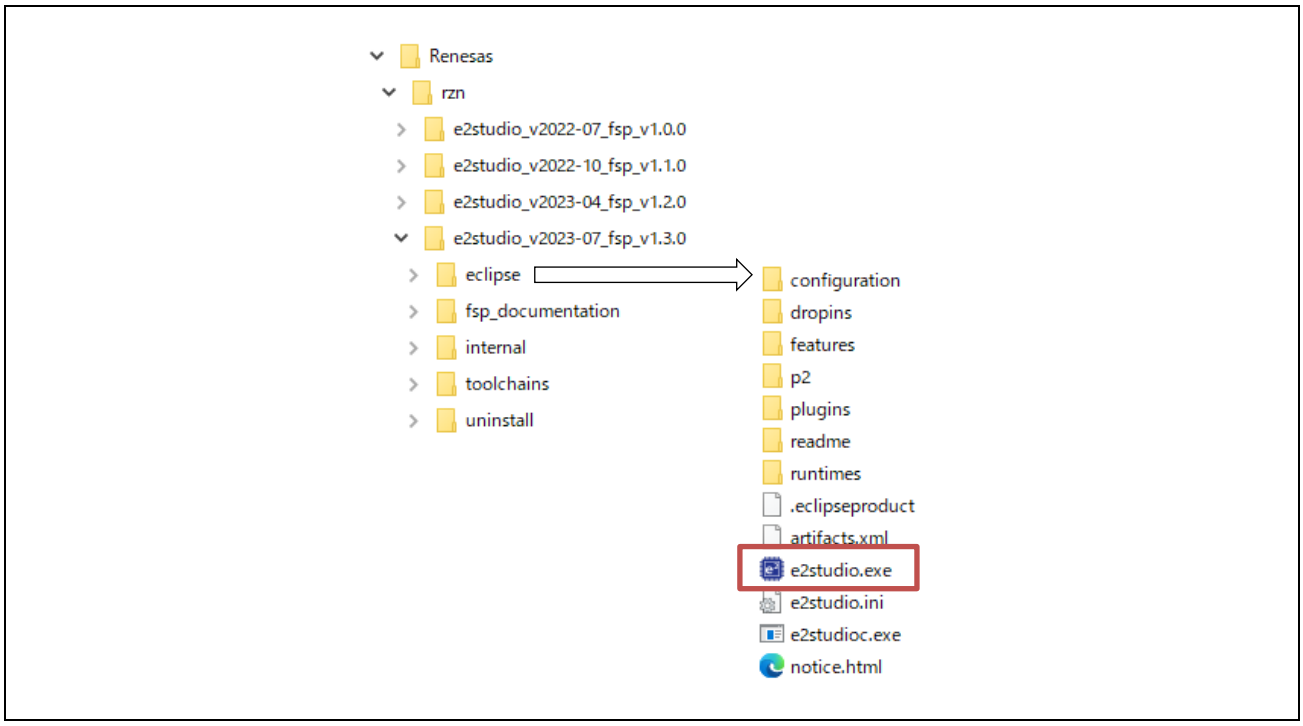

Fig.3-16 Launch project (1)

(3) Import Project

Enter any workspace directory and click “Launch”.

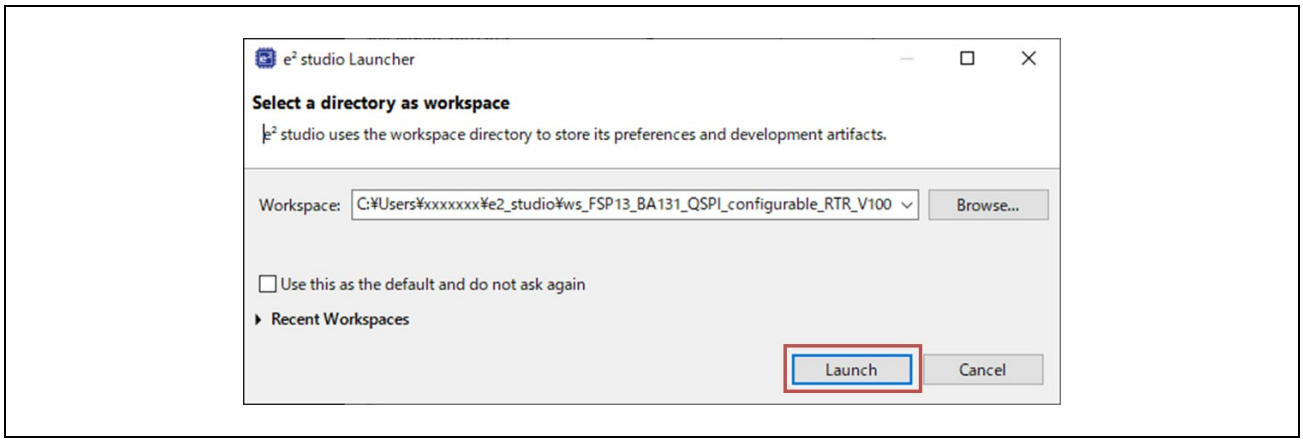

Fig.3-17 Launch project (2)

・Select “Import existing projects”

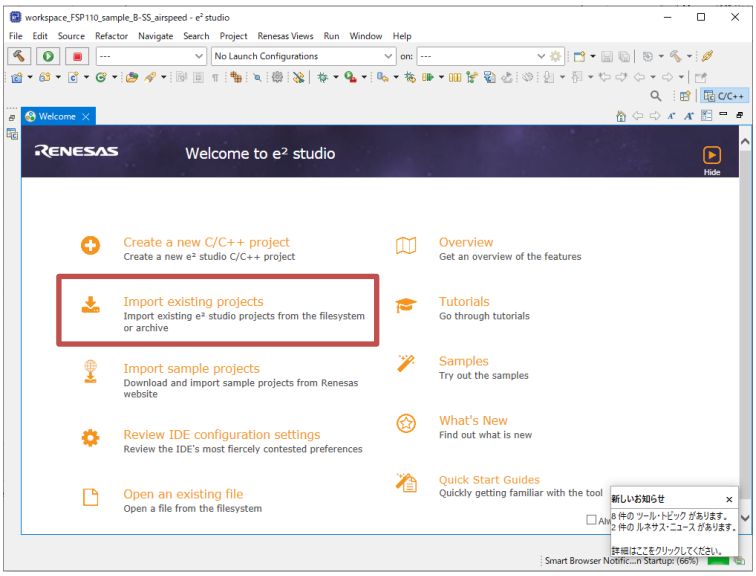

Fig.3-18 Launch project (3)

Click ”Browse” at “Select root directory” and enter the project folder to be imported.

Check the “Copy projects into workspace” checkbox to copy the import project.

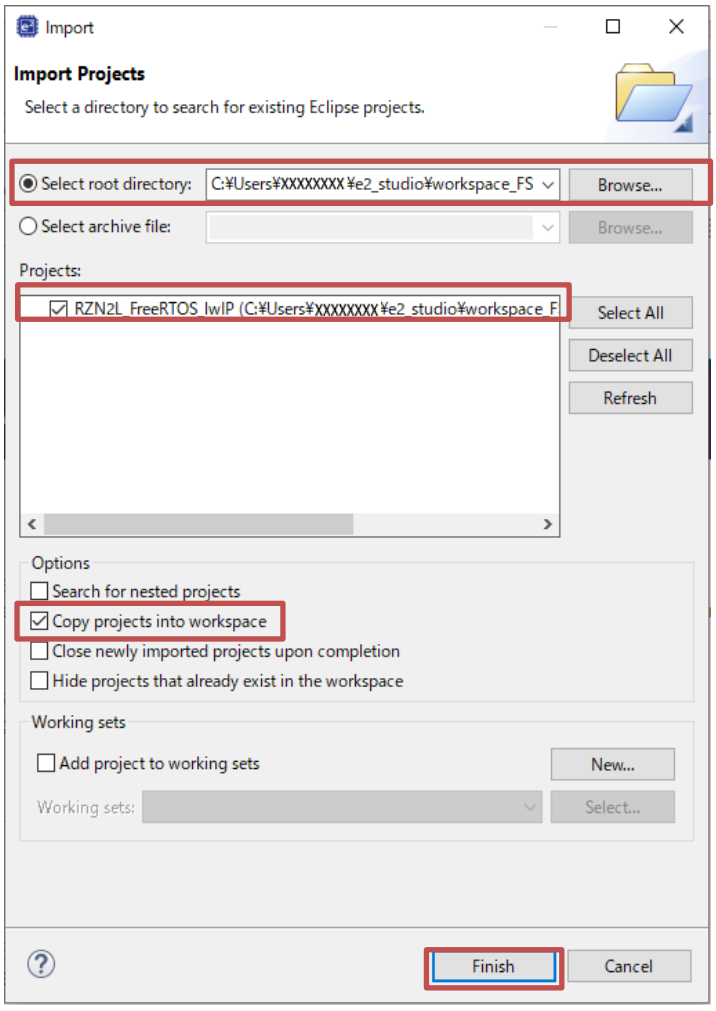

Fig.3-19 Launch project (4)

Click “Finish” in Fig.3-19 to display the following and click “Yes To All”.

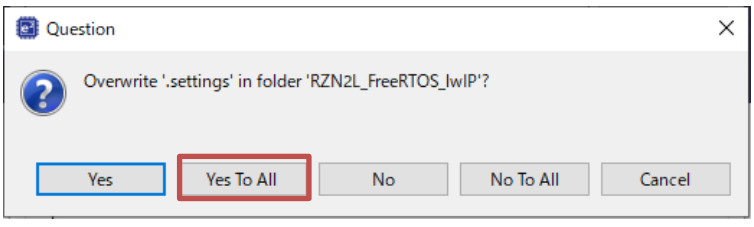

Fig.3-20 Launch project (5)

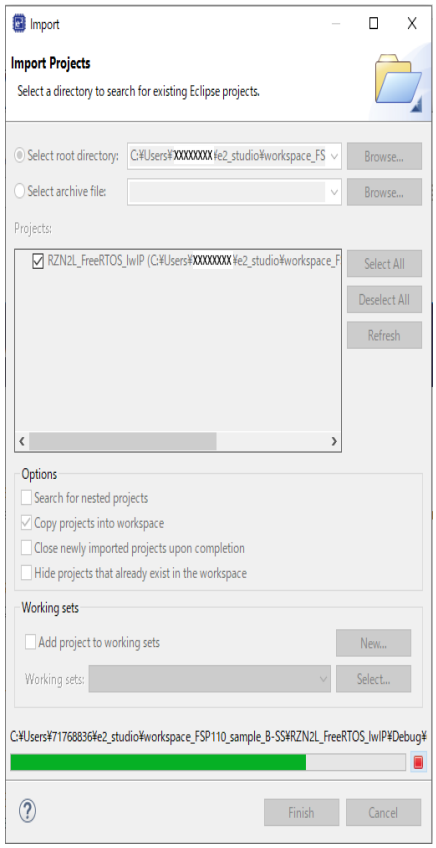

Fig.3-21 Launch project (6)

・When the project import is complete, the following will be displayed. The subsequent sections will be explained in chapter 4.

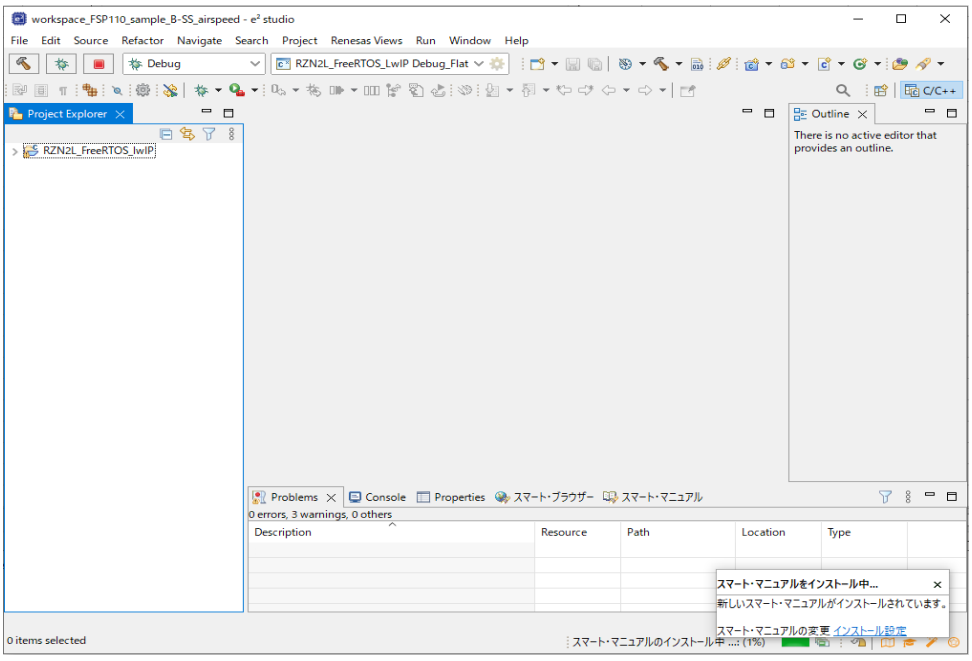

Fig.3-22 Launch project (7)

3.4.2 VTS

Visual Test Shell (VTS) is an application for testing BACnet functionality in systems that use the BACnet/IP protocol. Download the relevant version listed in Table 1-1 from the website and install it on your PC.

For the installation procedure, unzip the downloaded file, open QuickStart.html in the \Docs folder, and refer to the Quick Start Guide.

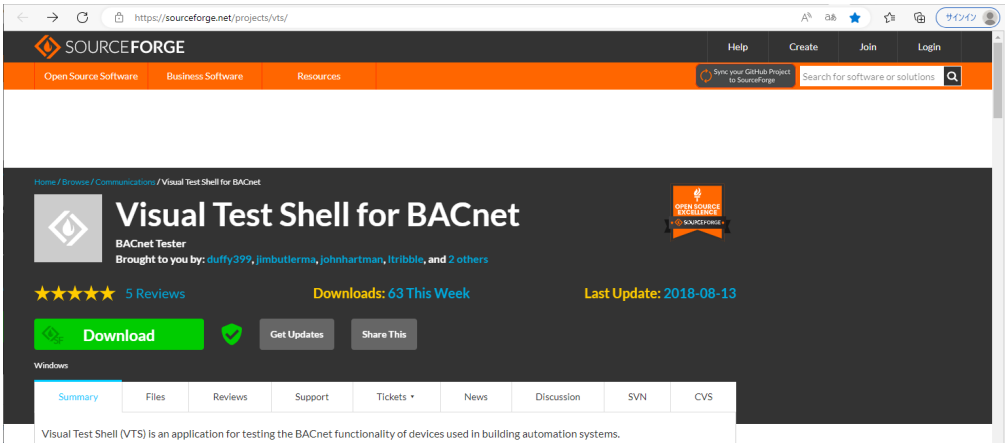

Fig.3-23 VTS

3.4.3 Yabe

Yet Another Bacnet Explorer (YABE) is a graphical window program for exploring and navigating BACnet devices. It does not have a single service output interface like VTS, but it can test systems running on the BACnet MS/TP and BACnet/IP protocols with ease.

Download the relevant version listed in Table 1-1 from the website and install it on your PC.

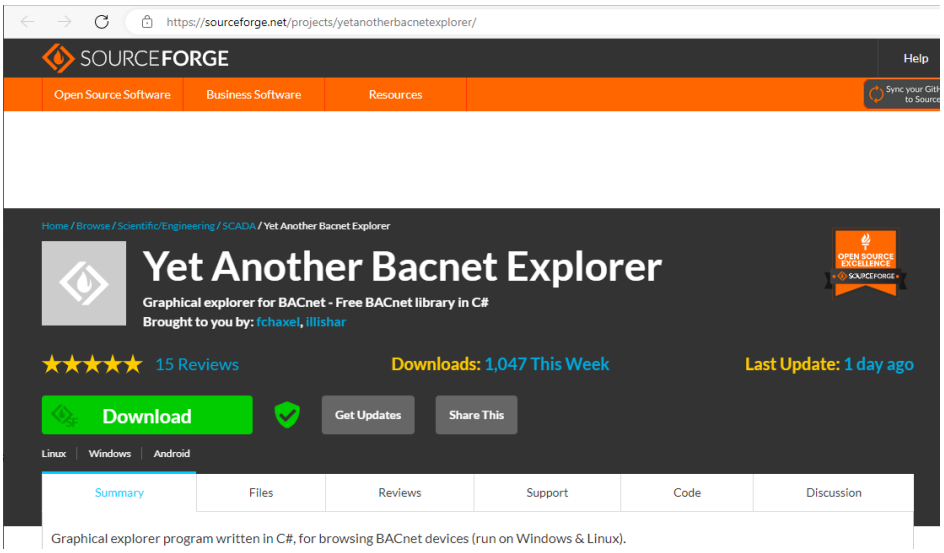

Fig.3-24 Yabe

3.4.4 Wireshark

Wireshark is a free network protocol analyzer. Download and install Wireshark from the link in Table 1-1.

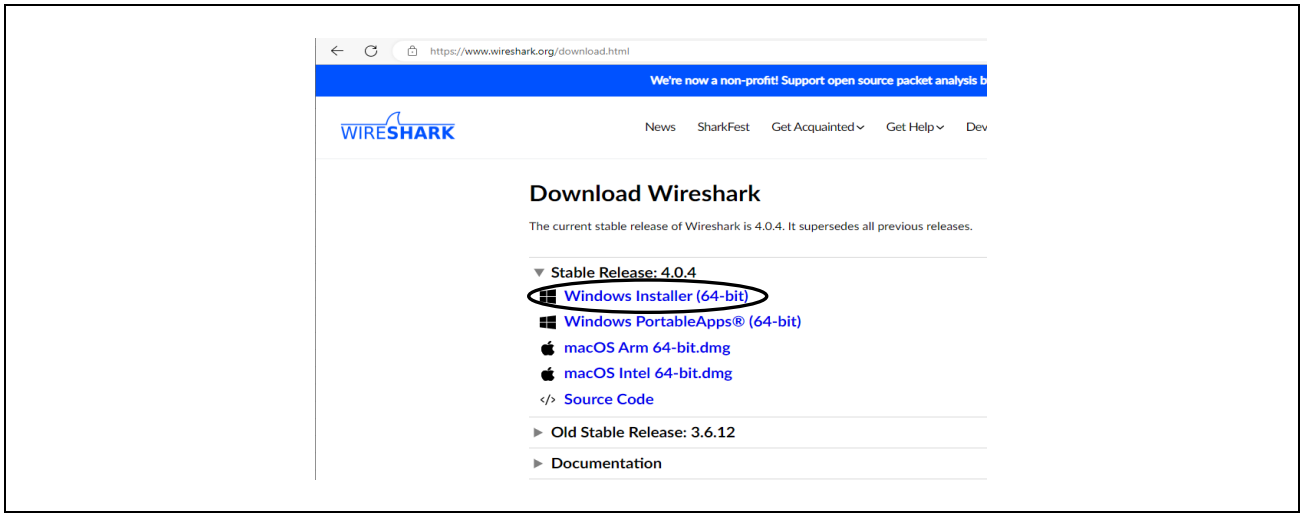

Fig.3-25 download Wireshark

3.4.5 Terminal software

Download and install free terminal software such as TeraTerm.

The terminal software is used to execute initial configuration commands to store configurable property values (configurable properties) in Flash memory. See chapter 5.3 for details.

4. Operation Verification

4.1 Connection

Fig.4-1 shows a connection diagram for this sample software. Connect the Ethernet cable, J-Link OB debugger, 5V DC cables to the RZ/N2L RSK board. When connecting the board for B-SS , connect the air velocity sensor to the J26 on it. In the board settings in chapter 2.1, it is possible to connect an Ethernet cable to any of the Ethernet connectors of ETH0, ETH1, and ETH2. When using the on-board debugger J-Link OB on the RSK board, leave J9 open and connect the USB Micro cable to J10.

RS-485 is a 2-wire half-duplex communication, connecting the positive line of CN12-6 pin (RS485_A) and the negative line of CN12-3 pin (RS485_B) between the B-BC and B-SS boards. Sends service requests from the PC tool to the B-SS via the B-BC as the master device and sends B-SS responses to the PC tool.

To monitor packets of BACnet MS/TP communication with Wireshark, it is necessary to separate the USB port of the PC from that for MS/TP communication, so prepare an RS485/USB converter and connect it by splitting RS485_A /B signals. (Fig.4-1 1 dotted line)

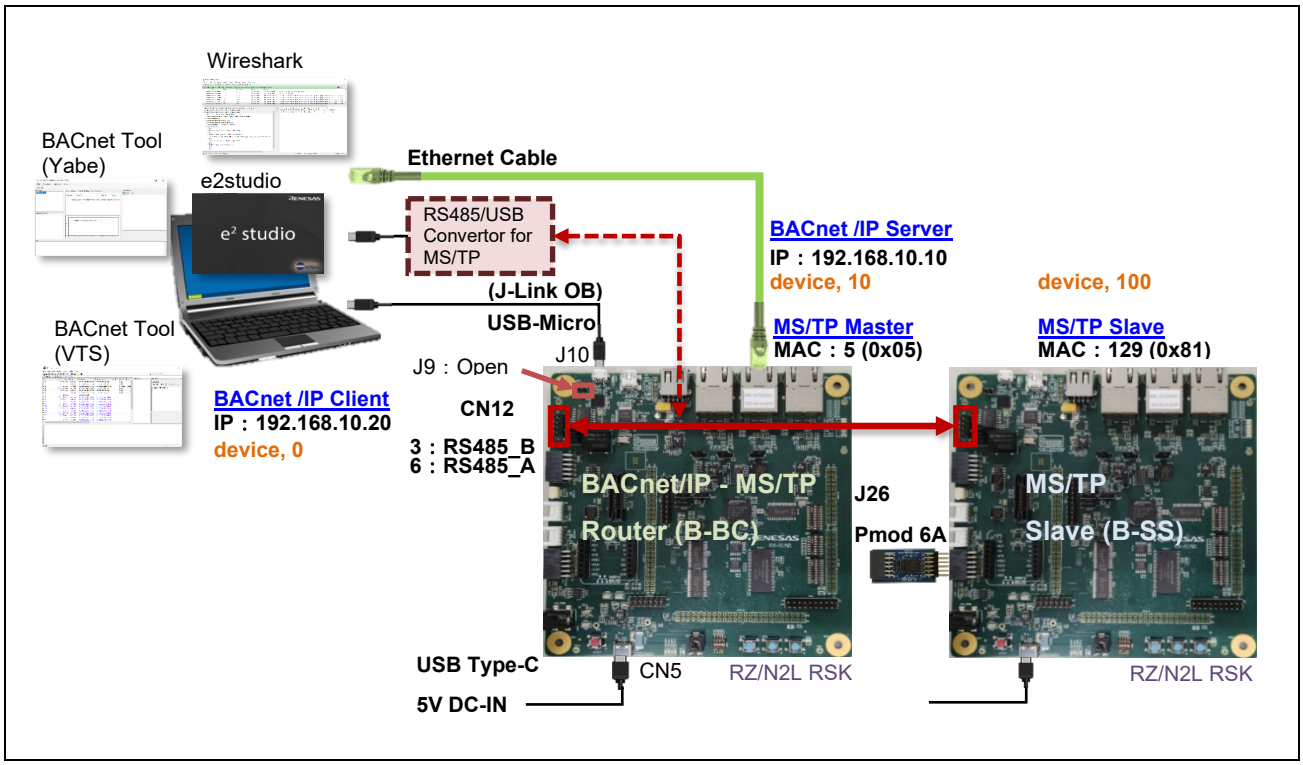

Fig.4-1 BACnet/IP-MS/TP Hardware Diagram

4.2 IP Address Setting for BACnet Client

Configure the Ethernet IP address settings for the PC.

Click on settings in Windows Start.. Configure the IP address as follows.

Settings > Network and Internet > Change adapter options > Ethernet

＞Properties > Internet Protocol Version 4 (TCP/IPv4) > Properties

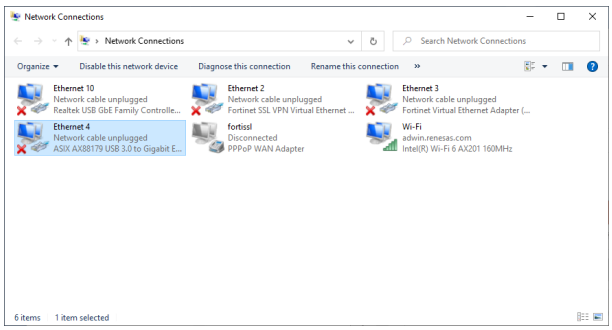

Fig.4-2 network connection

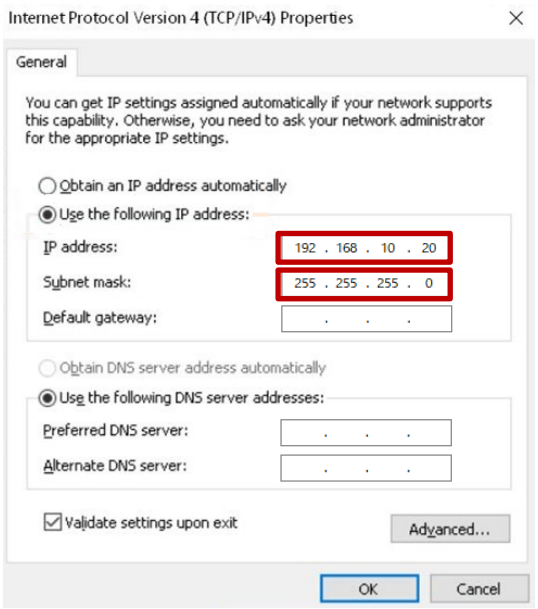

Fig.4-3 TCP/IPv4 properties

The IP address of the RSK board set in the B-BC sample software is 192.168.10.10. The IP address of the PC needs to be set to 192.168.10.XXX. In this document, 192.168.10.20 is used.

4.3 Setup Wireshark

It is possible to capture BACnet/IP communication packets between a PC and B-BC with Wireshark.

Also, to capture the MS/TP protocol packet with Wireshark, download mstpcap.exe from the link in Table 1-1.

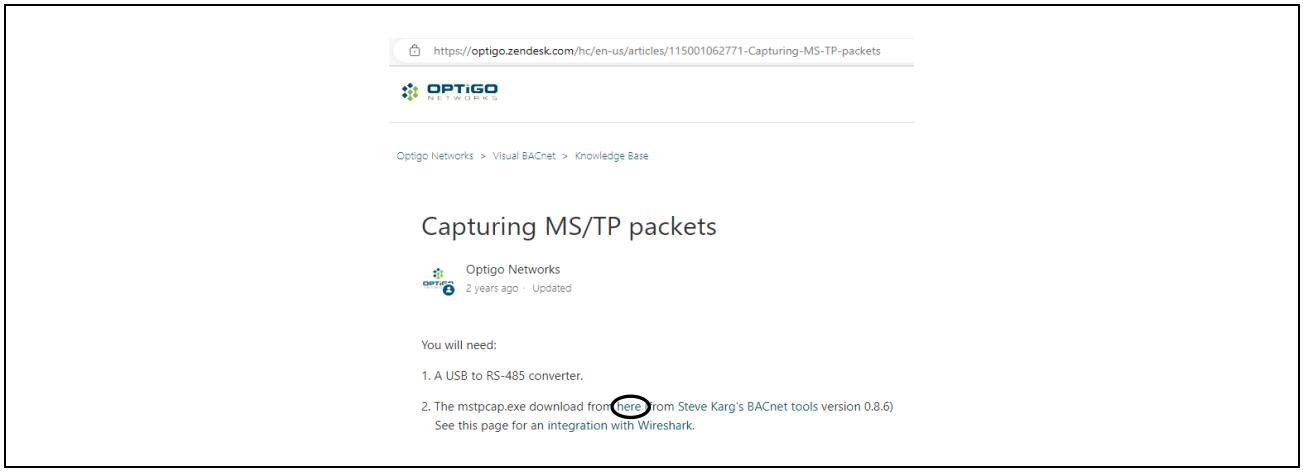

Fig.4-4 Download mstpcap.exe

Paste mstpcap.exe under /Program Files/Wireshark/extcap folder.

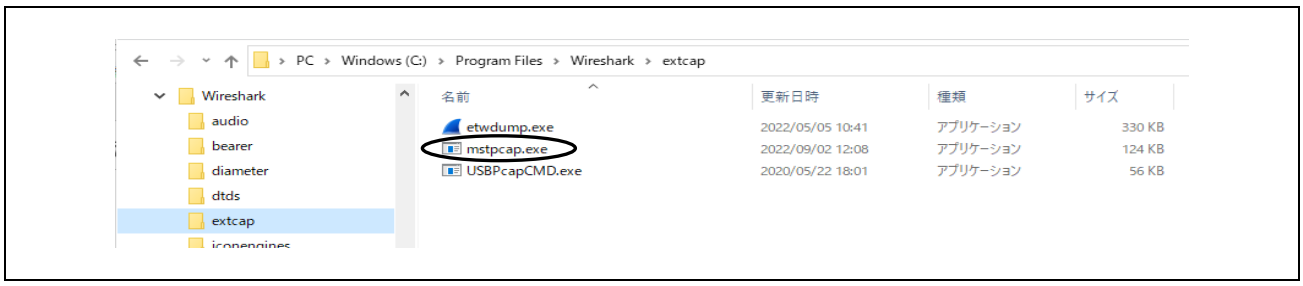

Fig.4-5 Paste mstpcap.exe into extcap folder

Launch Wireshark and click COM Port Settings 

Select Baud Rate 115200 in the pop-up dialog and Save. Click Start Packet Capture. 

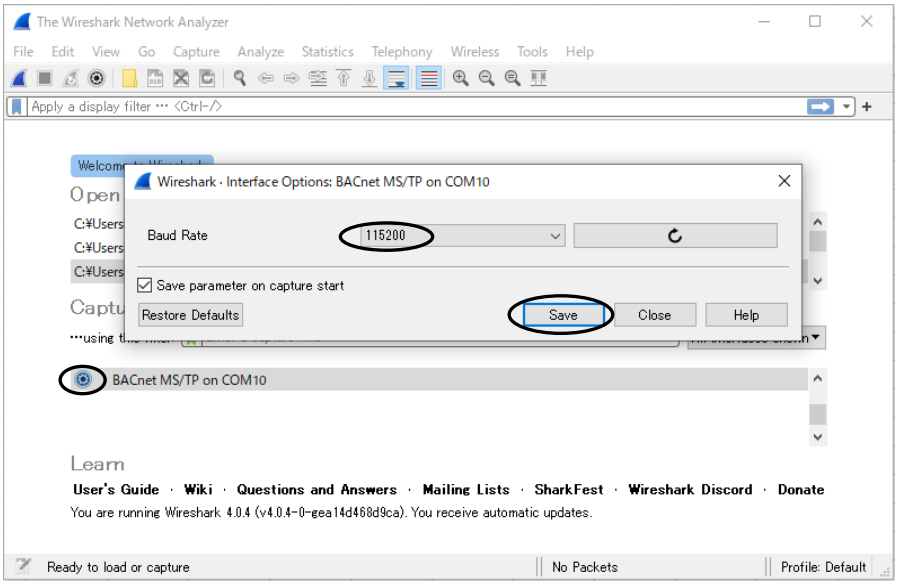

Fig.4-6 Baud rate selection

MS/TP capture screen of Wireshark appears.

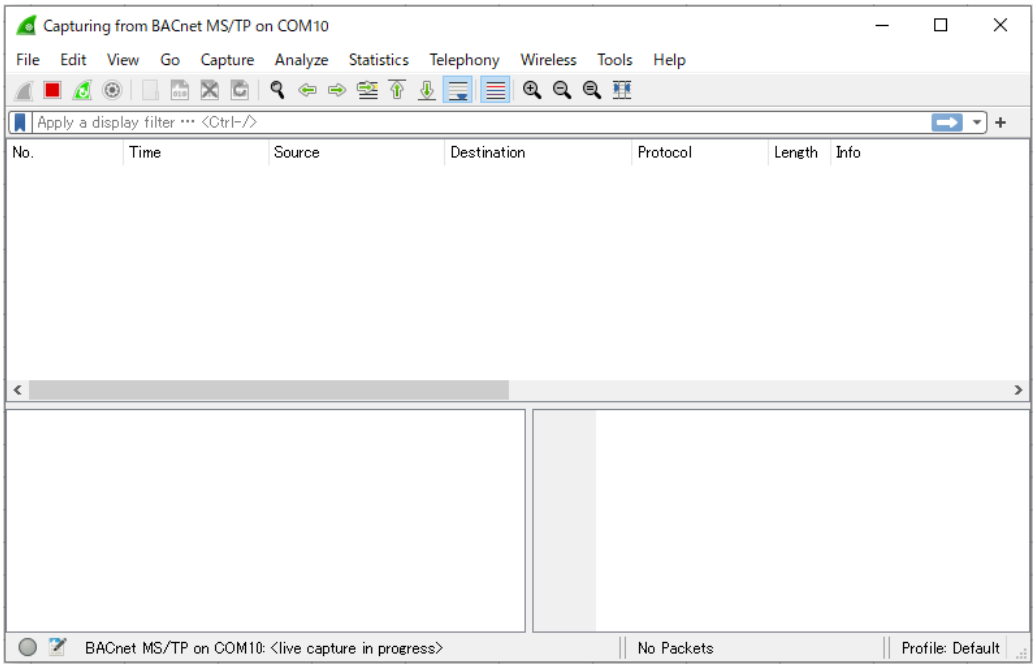

Fig.4-7 MS/TP protocol packet capture screen

4.4 Start Project

First, import the project as described in section 3.4.1.2.

4.4.1 Build Configuration Notes

Various Symbol definitions are referenced for building. See chapters 5.2.45.1.4 and 5.2.6 for details.

4.4.1.1 Change Prohibited Symbols

Changing values of some symbols will result in building errors.

Select the project name in the Project Explorer window, then open Properties in the Project menu.

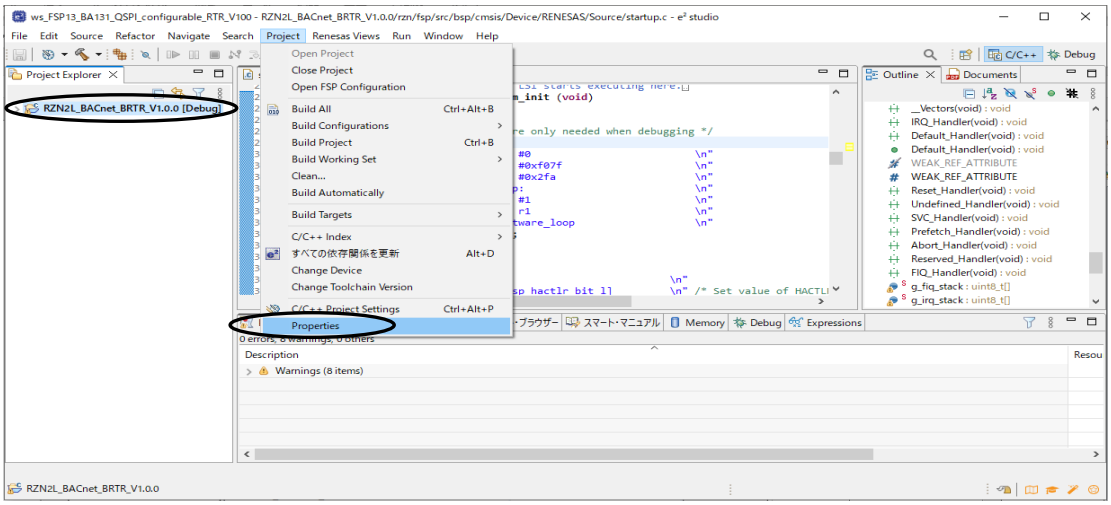

Fig.4-8 Open project properties

Select GNU C in Languages from the #Symbols tag in C/C++General > Paths and Symbols.

Do not change #BACDL_ALL, #BACDL_BIP, or #BACDL_MSTP in Symbol. If changed, B-BC sample software will not be able to be built.

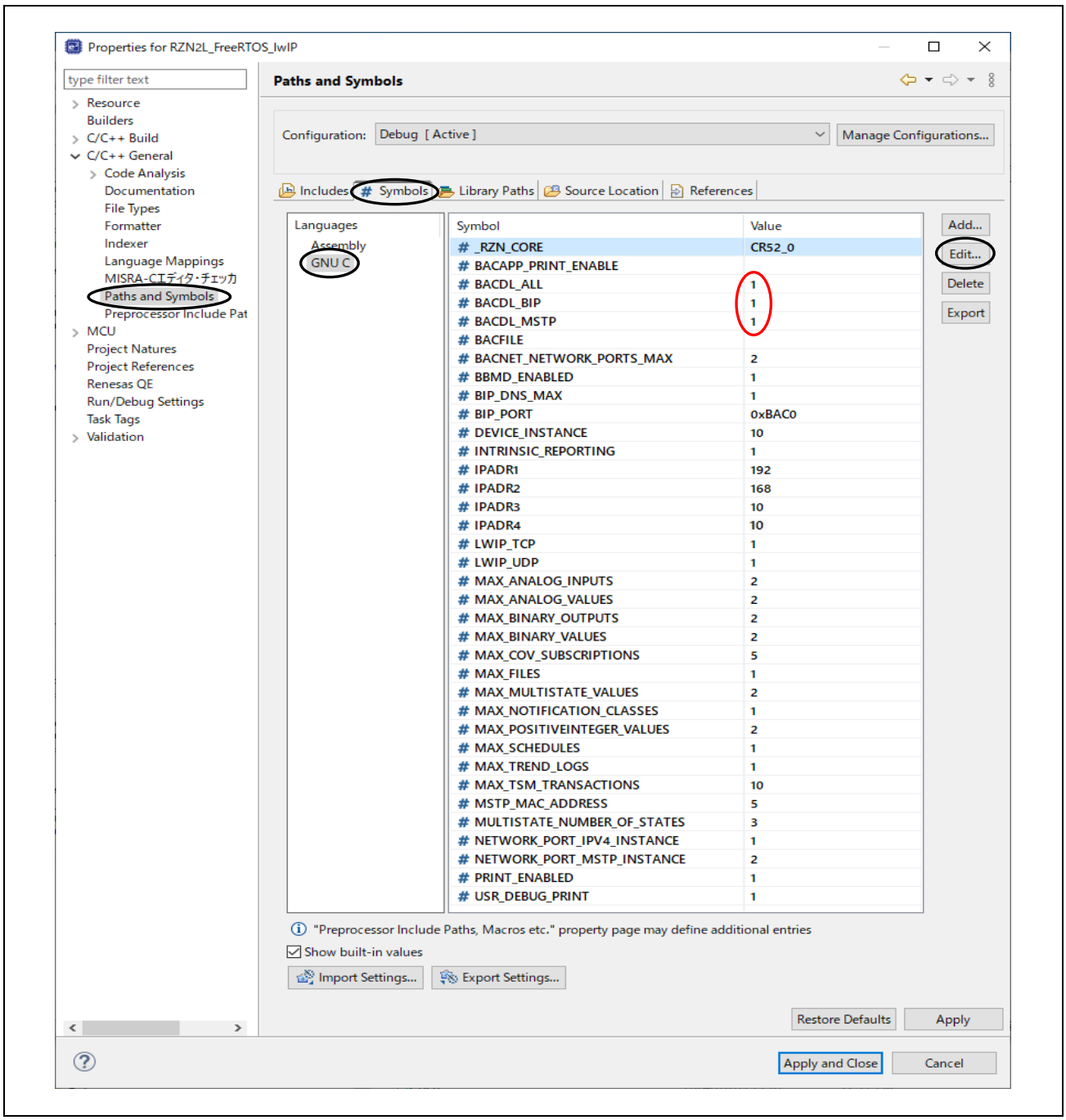

Fig.4-9 Change BACDL_MSTP to 0 and BACDL_BIP to 1

4.4.1.2 Settings for NTP Server

NTP client functionality is implemented in the B-BC sample software to get the current time automatically from NTP (Network Time Protocol) server with "SNTP" (Simple Network Time Protocol) from an open-source lwIP.

However, it is assumed that the connecting PC is the NTP server, and B-BC identifies the NTP server by the IP address; identification by the NTP server name is not supported. If the connecting PC does not have NTP server function, the B-BC can get the current time from VTS (or Yabe), see chapter 4.5.3 and 4.6.1 for Time Synchronization Service.

Change the following code if the PC connecting to B-BC via BACnet/IP has NTP server. This IP address must be the same as the one configured in chapter 4.2. 192.168.10.20 is the default value.

user\renesas\application\lwip_port_main.c

```
/*****************************************************************************
 * @brief SNTP initial settings
 *****************************************************************************/
/*****************************************************************************
* Function Name: user_sntp_init
* Description  : 8.3a and 8.3c valid function (8.3a REQUIRED)
* Arguments    : void
* Return Value : None
*****************************************************************************/
void user_sntp_init(void);

void user_sntp_init(void)
{
    ip_addr_t sntpIPADDR;

    sntp_setoperatingmode(SNTP_OPMODE_POLL);
#if SNTP_SERVER_DNS
    sntp_setservername(0, "pool.ntp.org"); /* DNS not supported */
#else
    IP_ADDR4(&sntpIPADDR, 192, 168, 10, 20);
    sntp_setserver(0, &sntpIPADDR);
#endif
    sntp_init();
}
```

Fig.4-10 Setting NTP server address

The following Wireshark capture shows NTP protocol packets (filtered by "ntp"). The time request cycle from B-BC to the NTP server is one hour.

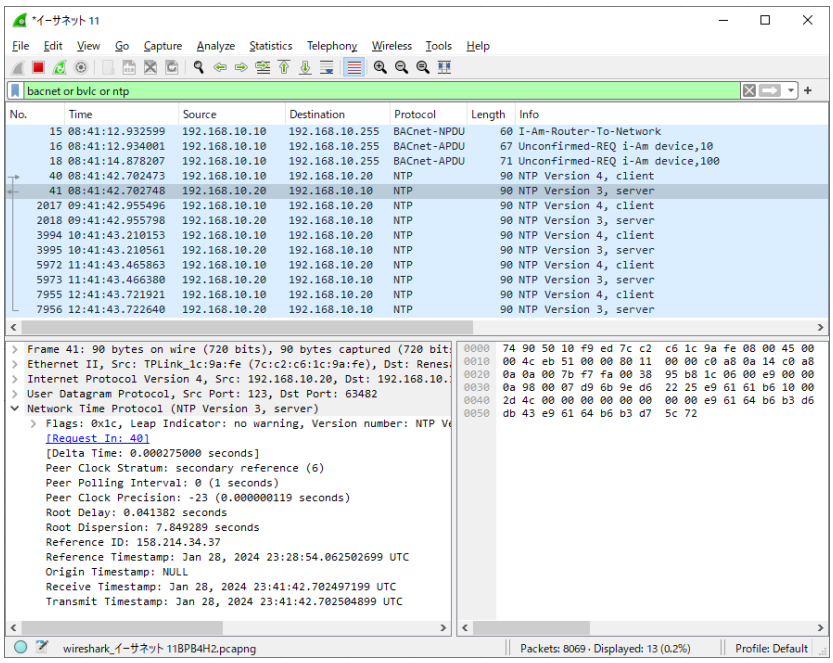

Fig.4-11 Capture image of NTP protocol packet

4.4.2 Build

Select the project name in the Project Explorer window and click Clean... in the Project menu.

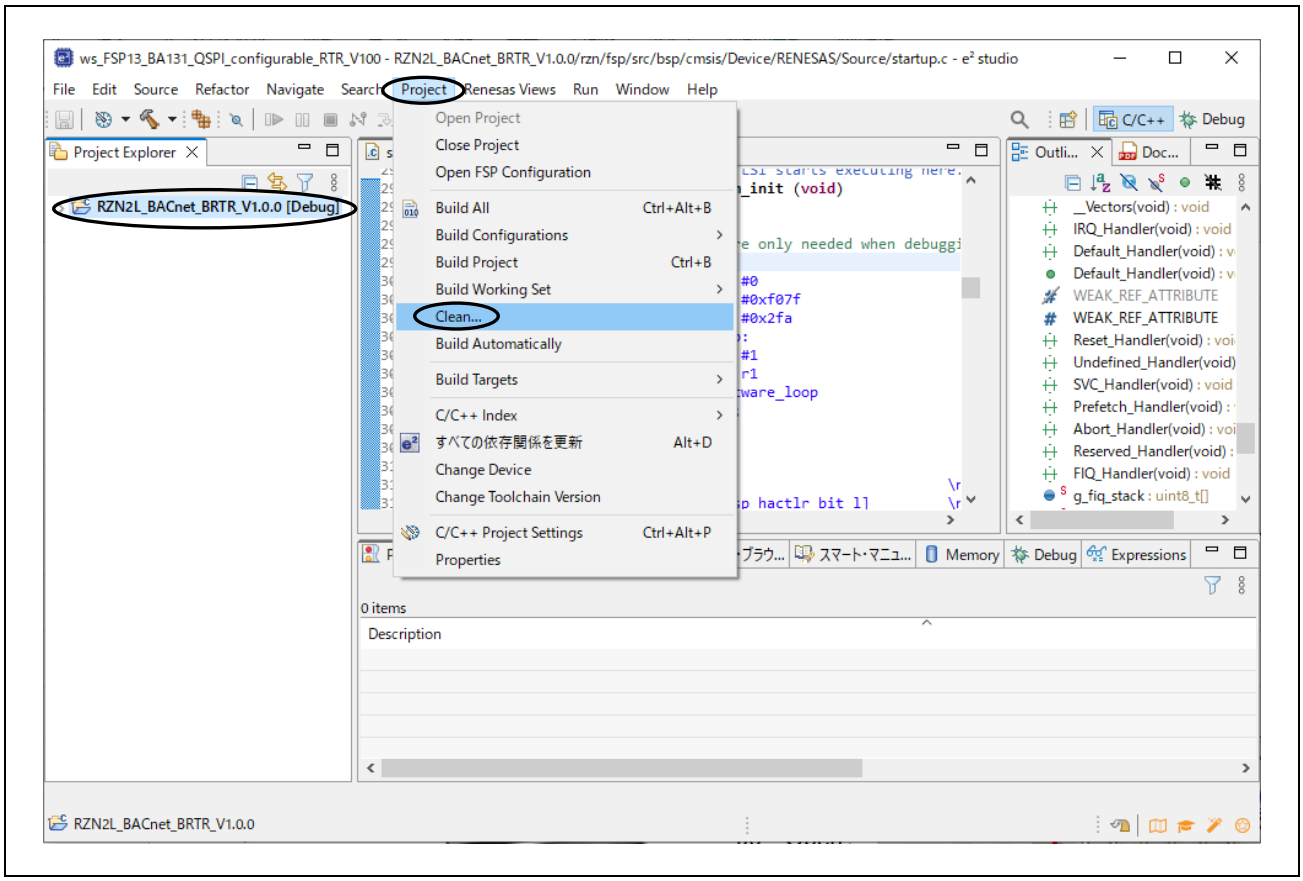

Fig.4-12 Open project Clean…

Enable the following in the pop-up dialog and click Clean to start all builds.

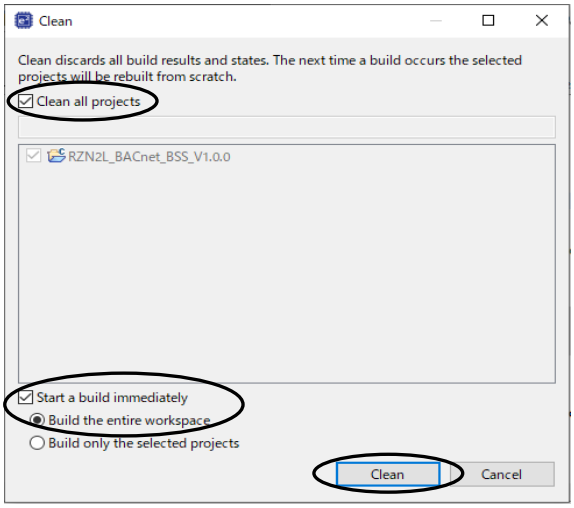

Fig.4-13 clean and rebuild

4.4.3 Debug Configurations

After confirming that the build result is 0 errors, select the project name in the Project Explorer window and click Debug Configurations... in the Run menu. Ignore the warning message generated for OSS code.

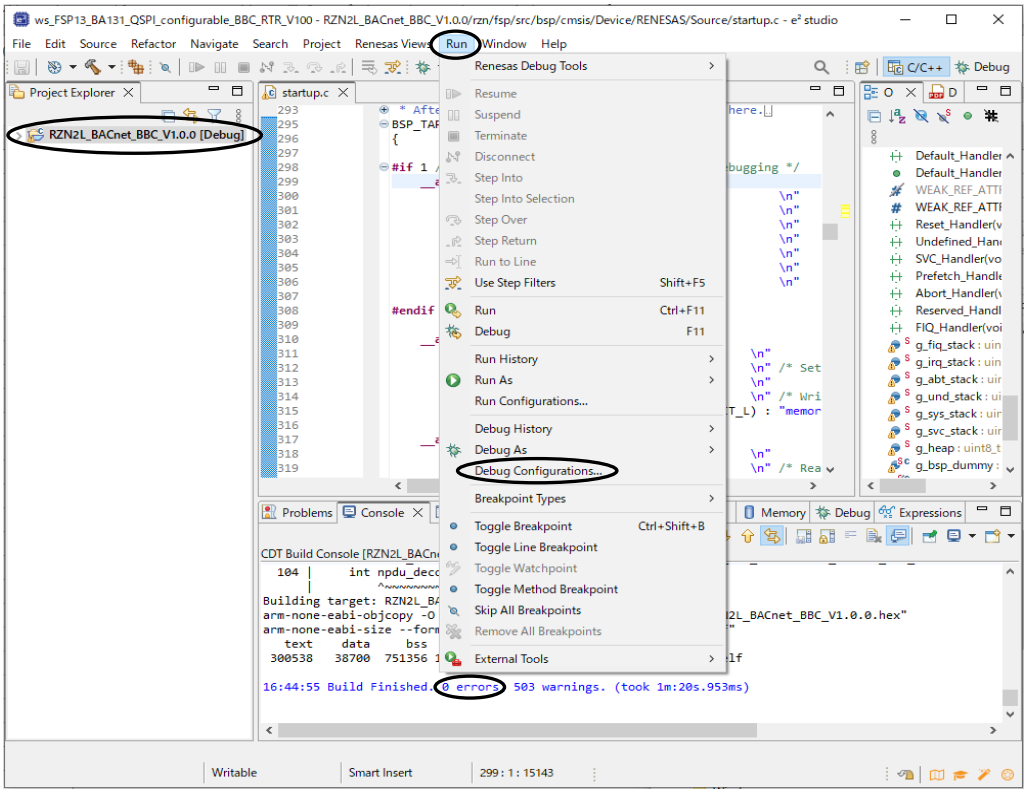

Fig.4-14 Open Debug Configurations…

Operations when starting the debugger for the first time after importing a project

Only when importing a project and launching the debugger for the first time, the following operations should be performed.

a. Create RZN2L_BACnet_BBC_V*** Debug[local]
b. Select Target Device
c. Debut Tool Settings
d. Macro Registration

See the following explanation of the above.

a. Create RZN2L_BACnet_BBC_V*** Debug[local]

Double click on Renesas GDB Hardware Debugging to generate RZN2L_BACnet_BBC_V*** Debug[local]

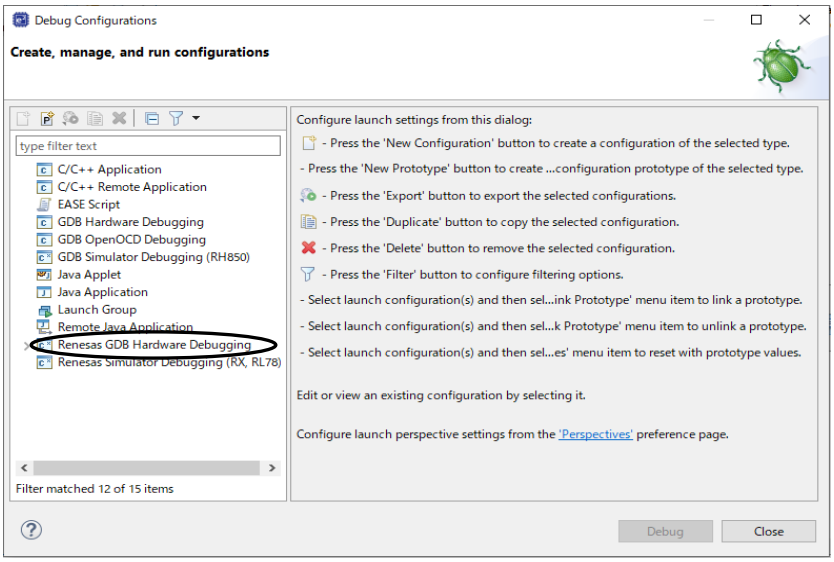

Fig.4-15 Debug Configurations(1)

b. Select Target Device

Click on the Debugger tag in the displayed dialog and select Target Device.

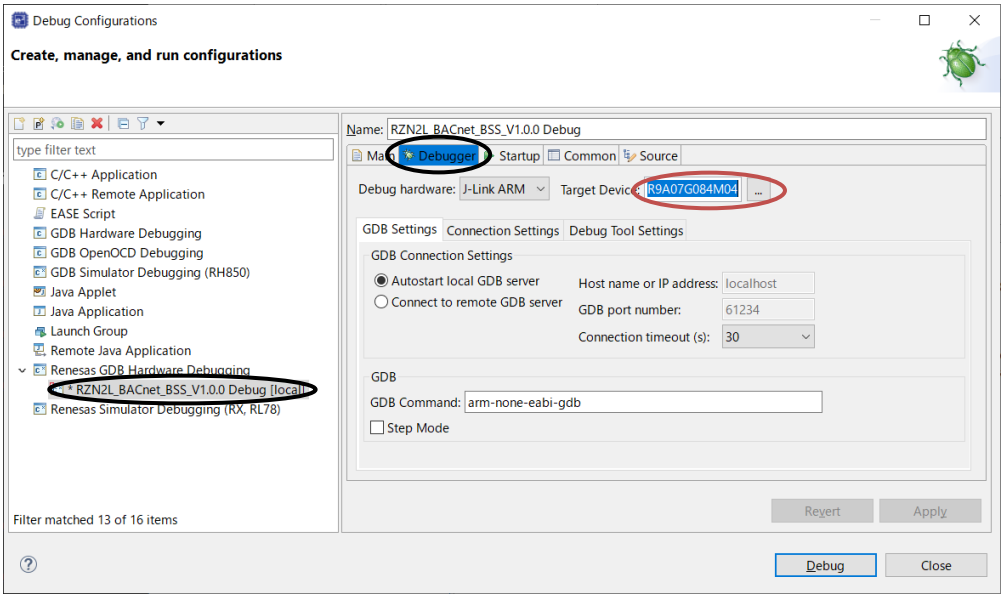

Fig.4-16 Debug Configurations(2)

Select **R9A07G084M04** and click OK.

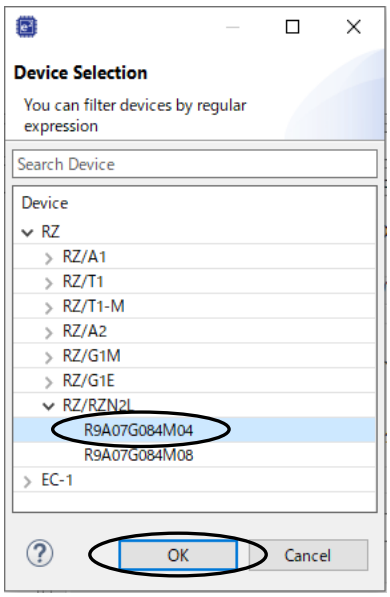

Fig.4-17 Debug Configurations(3)

c. Debug Tool Settings

Click the Debut Tool Settings tag and write **400** at Operating Frequency [MHz]

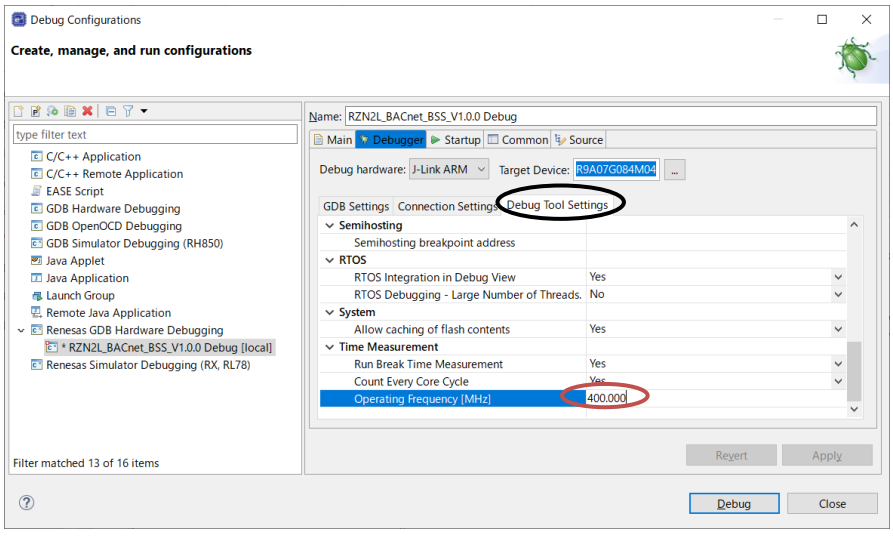

Fig.4-18 Debug Configurations(4)

d. Macro Registration

Click on the Startup tag, input *"source rzn2l_xspi0_x1_boot.cfg"* in Run Commands, and click Apply.

Click on "Debug" to start downloading. Continue with the procedure in Fig.4-21

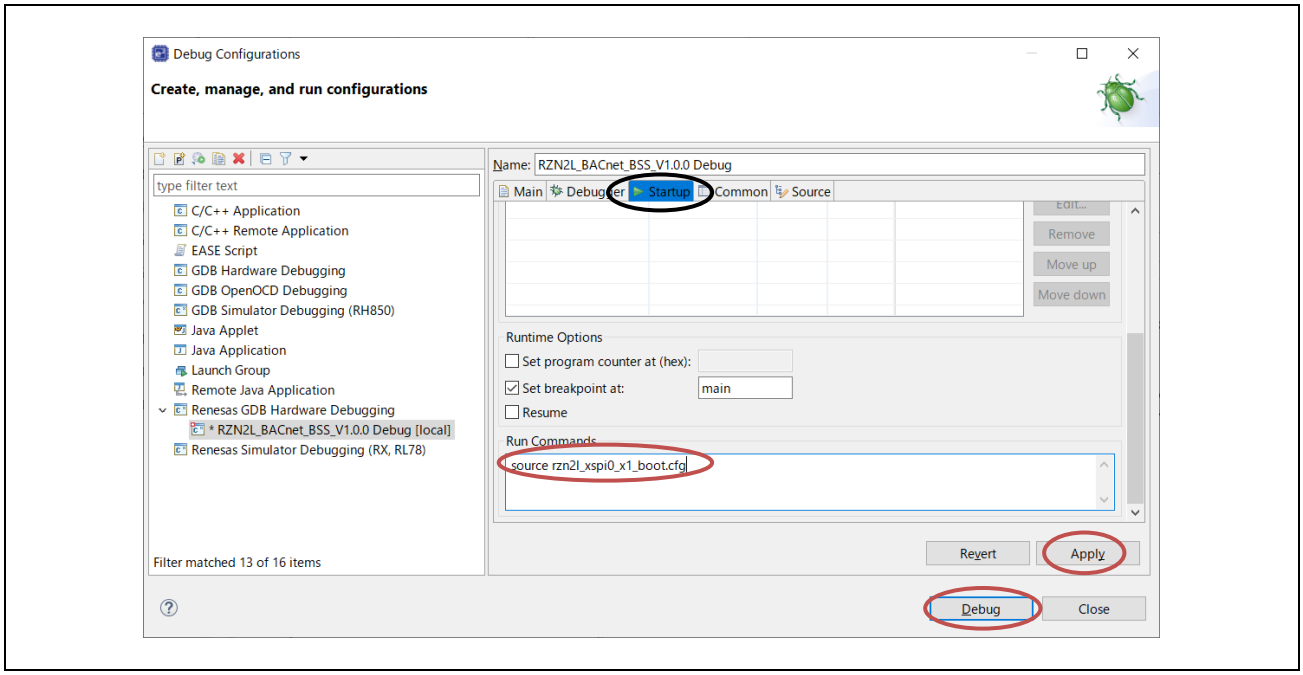

Fig.4-19 Debug Configurations(5)

4.4.4 Debug

The download procedure after completing the build is shown below.
At the second and subsequent debugger launches, click the Run menu with the project name selected in the C/C++ view.

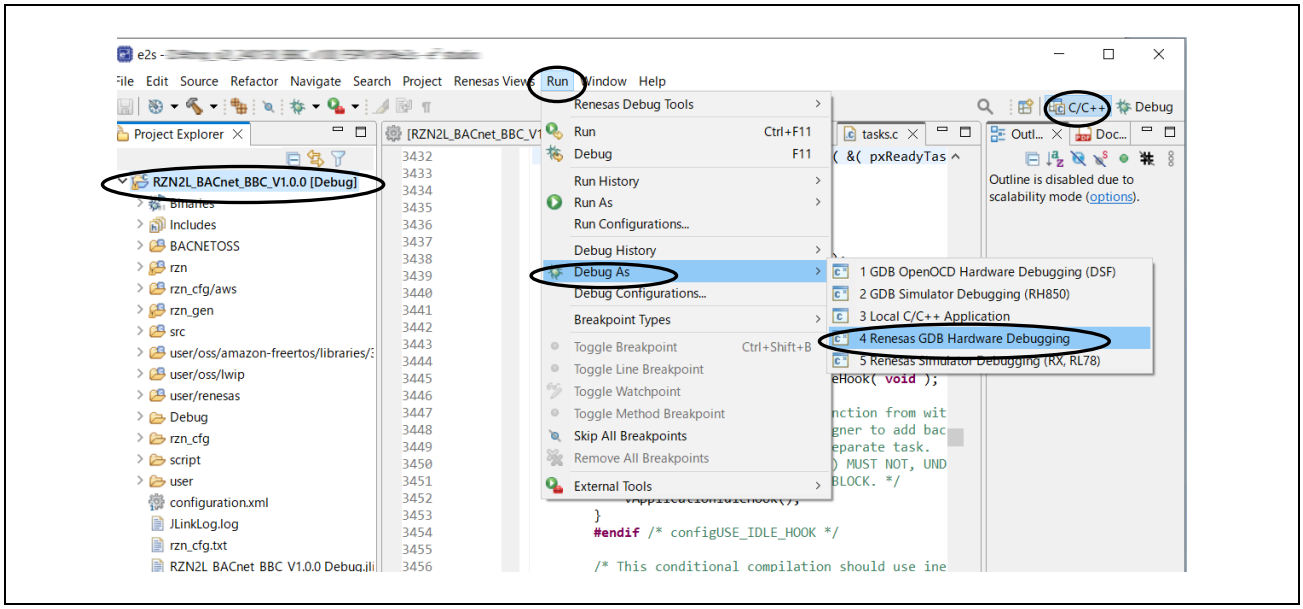

Fig.4-20 Run menu Debug As

Downloading the program to serial flash ROM.

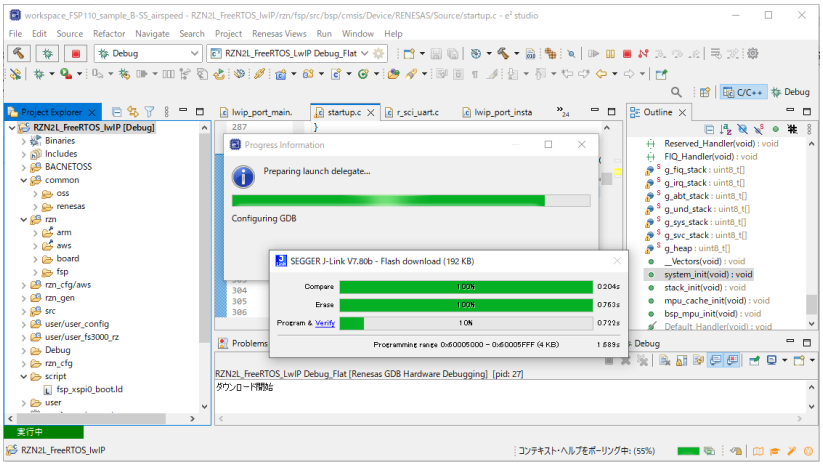

Fig.4-21 Download

Click Switch to change to debug view.

Fig.4-22 Perspective Switch

The CPU automatically extracts the loader program included in the download data to the BTCM. After extraction, it breaks in system_init() at the beginning of the initialization on the loader program.

- In case of operating the RSK board alone without using the debugger, turn off the board power supply, disconnect the debugger cable, and then turn on the board power supply again.

When using the debugger, **click the "reset" icon and then "resume"** after switching to the Debug screen.

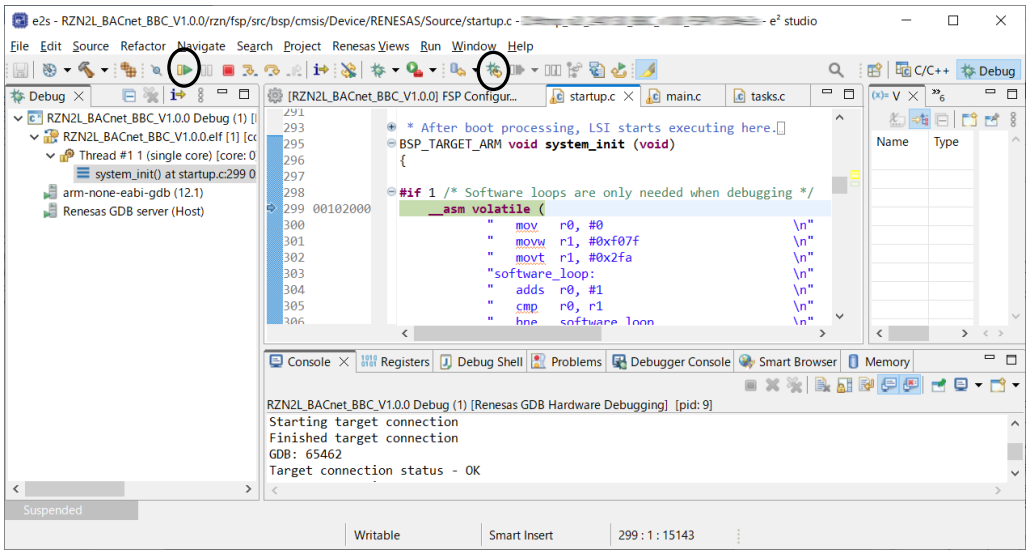

Fig.4-23 Break at system_init()

After completing initialization, the loader program jumps and stops at the beginning of main(), Then, click "resume" to return to the running state.

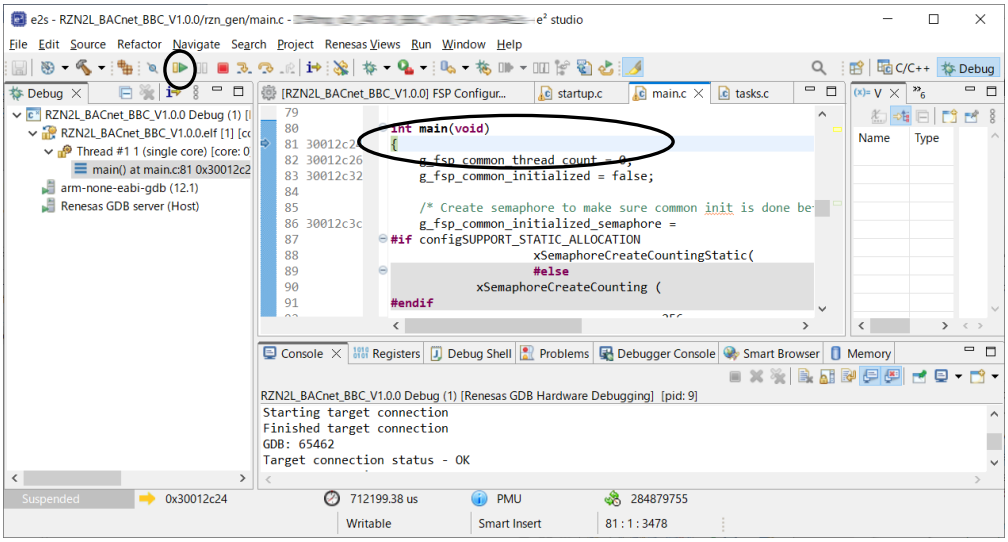

Fig.4-24 Break at main()

4.5 BACnet Communication with VTS

The following description assumes that Wireshark(Ethernet) is running before starting VTS.

B-BC broadcasts I-Am-Router-To-Network and I-Am service frames to the BACnet/IP network to which it belongs after initial configuration. The following shows these packets captured with Wireshark.

192.168.10.10 in Source is the B-BC server address.

192.168.10.255 in Destination means a broadcast address.

BACnet/IP uses UDP packets for communication and the default port is 47808 (0xBAC0). You can filter "bacnet" with Wireshark.(Filtered by "bacnet or bvlc or ntp" below)

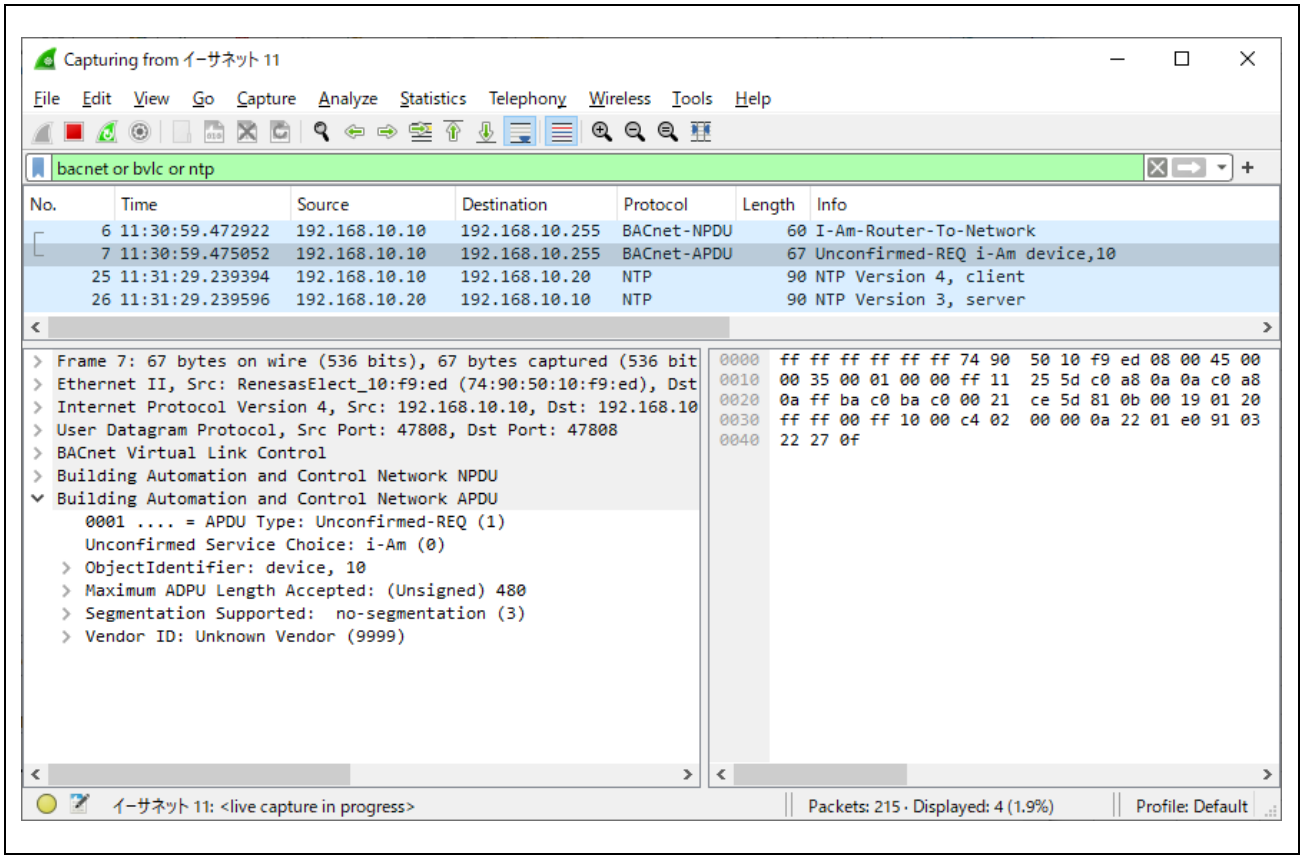

Fig.4-25 Capture I-Am service packet multicast from B-BC

・Launch VTS

Double-click VTS.exe in the folder where VTS was installed.

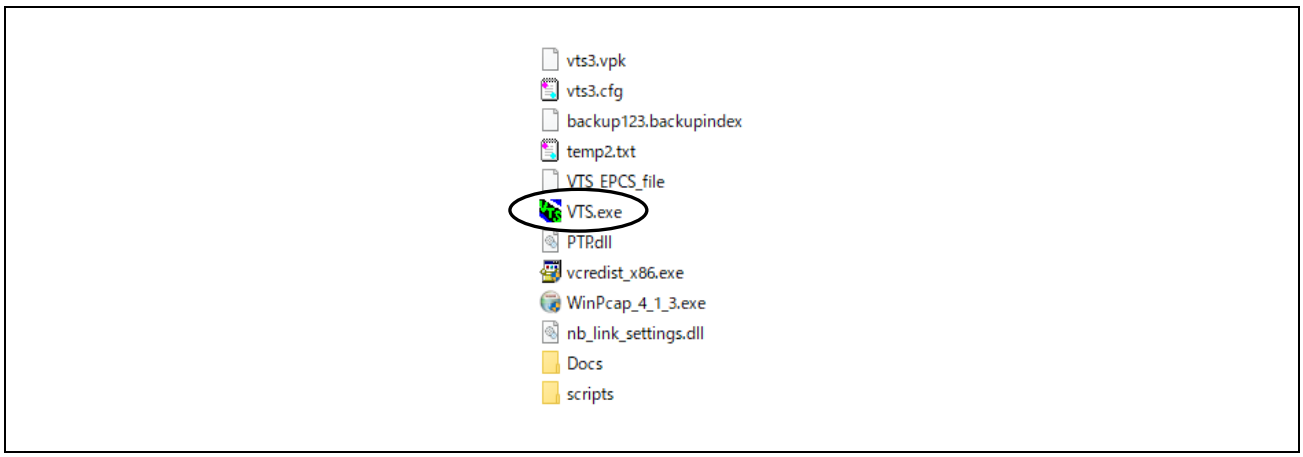

Fig.4-26 Launch VTS

The following descriptions are the settings used in the evaluation of the B-SS sample software. Please modify them according to your operating environment.

Click on Devices... in the Edit menu.

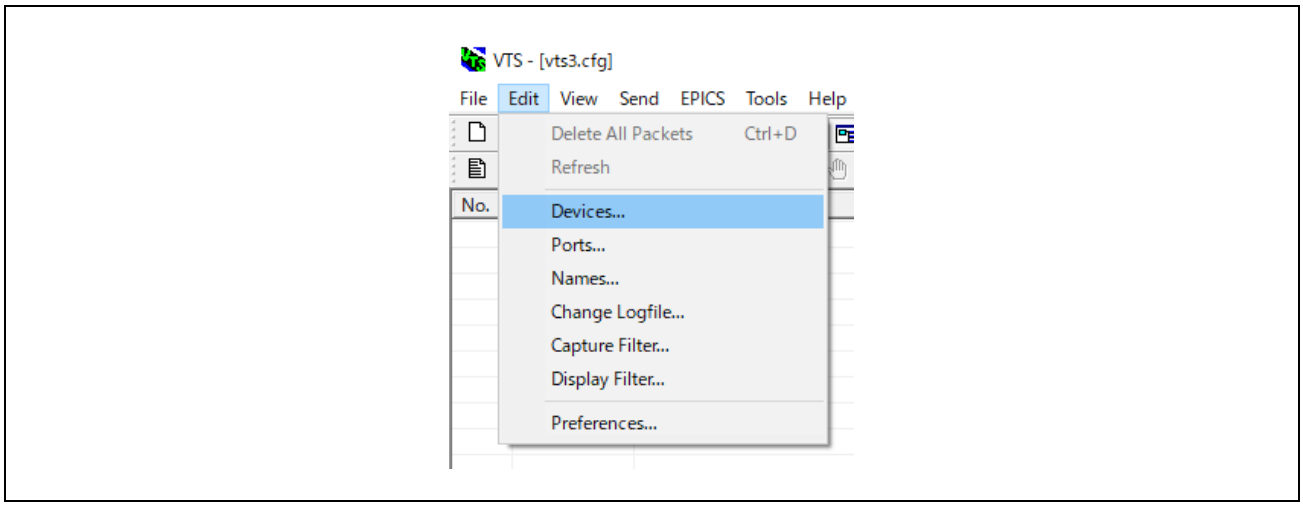

Fig.4-27 Edit menu Devices(1)

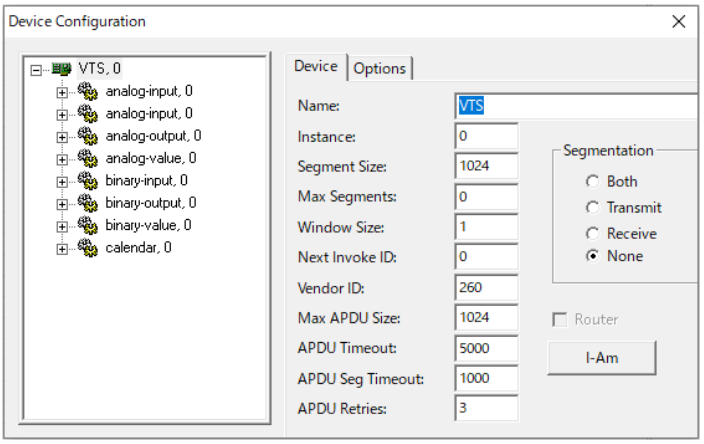

Fig.4-28 Edit menu Devices(2)

Click Ports... in the Edit menu.

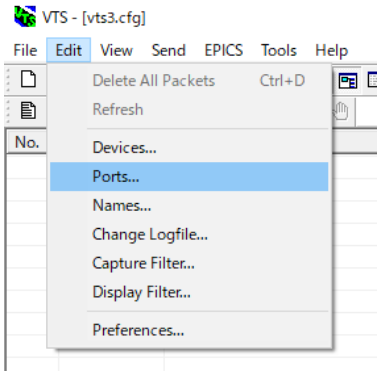

Fig.4-29 Edit menu Ports(1)

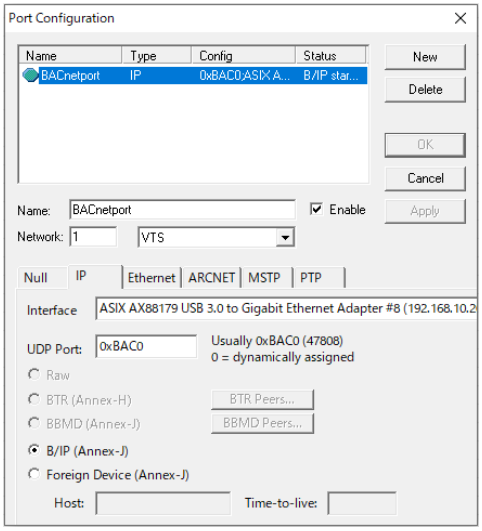

Fig.4-30 Edit menu Ports(2)

Click Names... in the Edit menu.

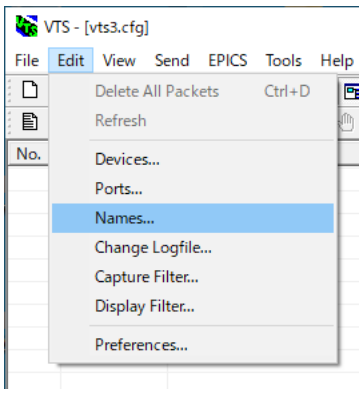

Fig.4-31 Edit menu Names(1)

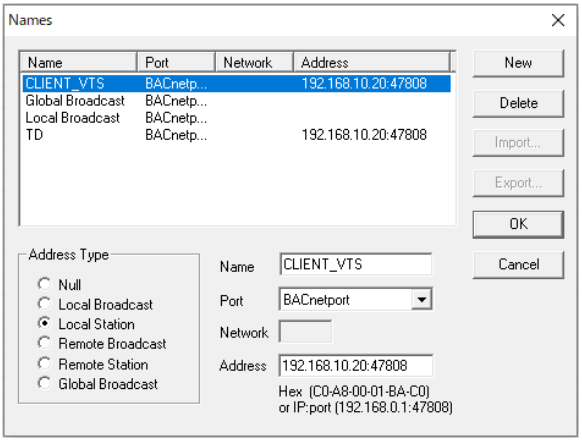

Fig.4-32 Edit menu Names(2)

4.5.1 Who-Is / I-Am

(1) Discover Devices

Click Discover Devices... in the Tools menu of the VTS.

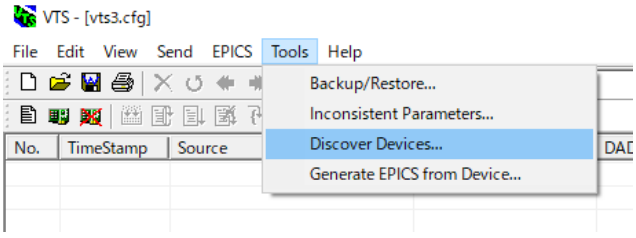

Fig.4-33 Tools menu Discover Devices(1)

Select Device: as Global Broadcast in the displayed dialog then click OK.

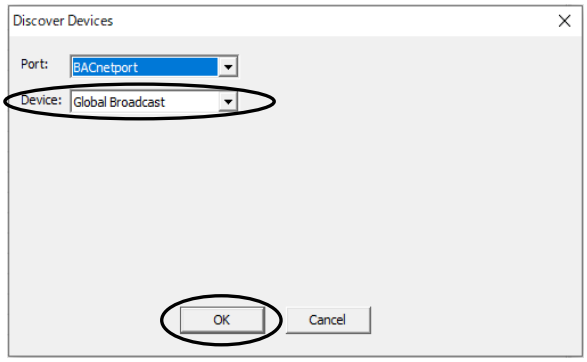

Fig.4-34 Tools menu Discover Devices(2)

The following Wireshark capture shows that the B-BC server returned I-Am device,10 and I-Am-Router-To-Network responses to the Who-Is service request and Who-Is-Router-To-Network network layer messages from the VTS client. The first Who-Is from the VTS is sent to all device IDs (0-4194303), and three devices respond I-Am. I-Am device,0 indicates the VTS itself, and I-Am device,100 is the B-SS connected to the MS/TP network; the source IP address of I-Am device,100 was routed through the IP address of the B-BC.

The VTS client is requesting four different property values to B-BC and B-SS with ReadProperty service; B-BC and B-SS respond with a Complex-Ack containing the results.

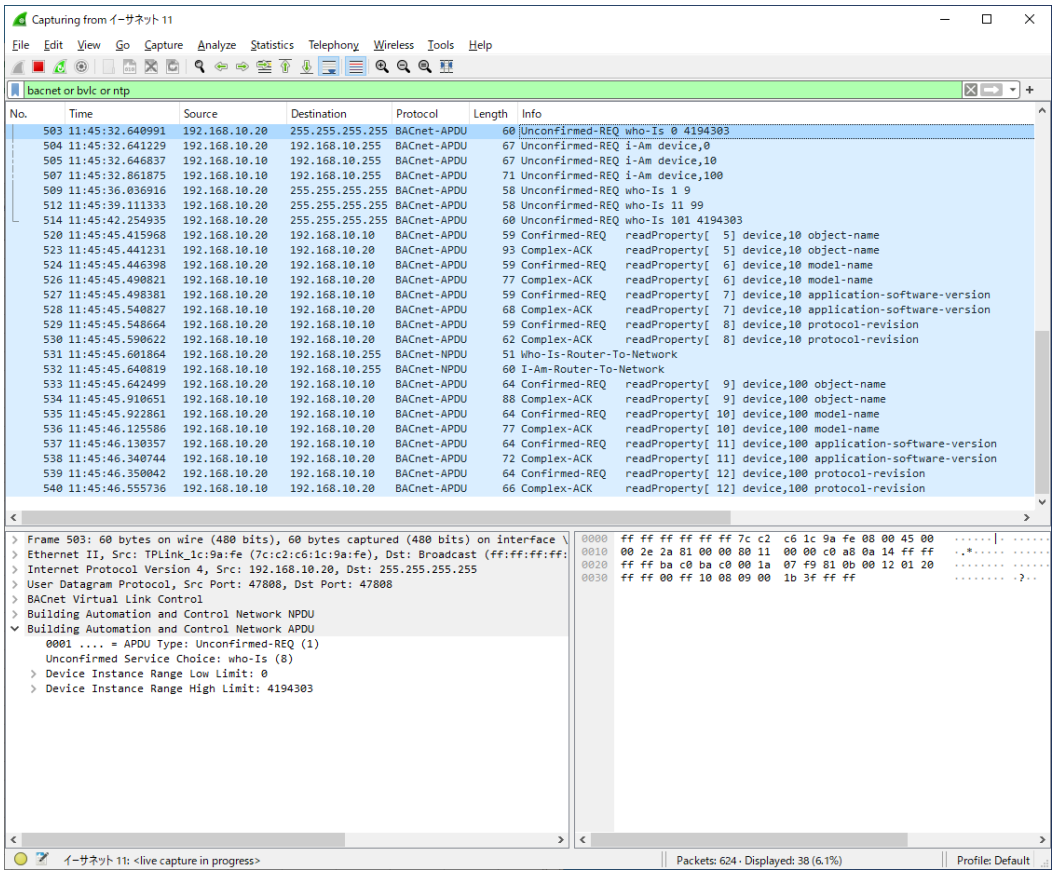

Fig.4-35 Tools menu Discover Devices(2)

VTS displays transmitted and received frames as follows. The contents of the display are not much different from those of Wireshark.

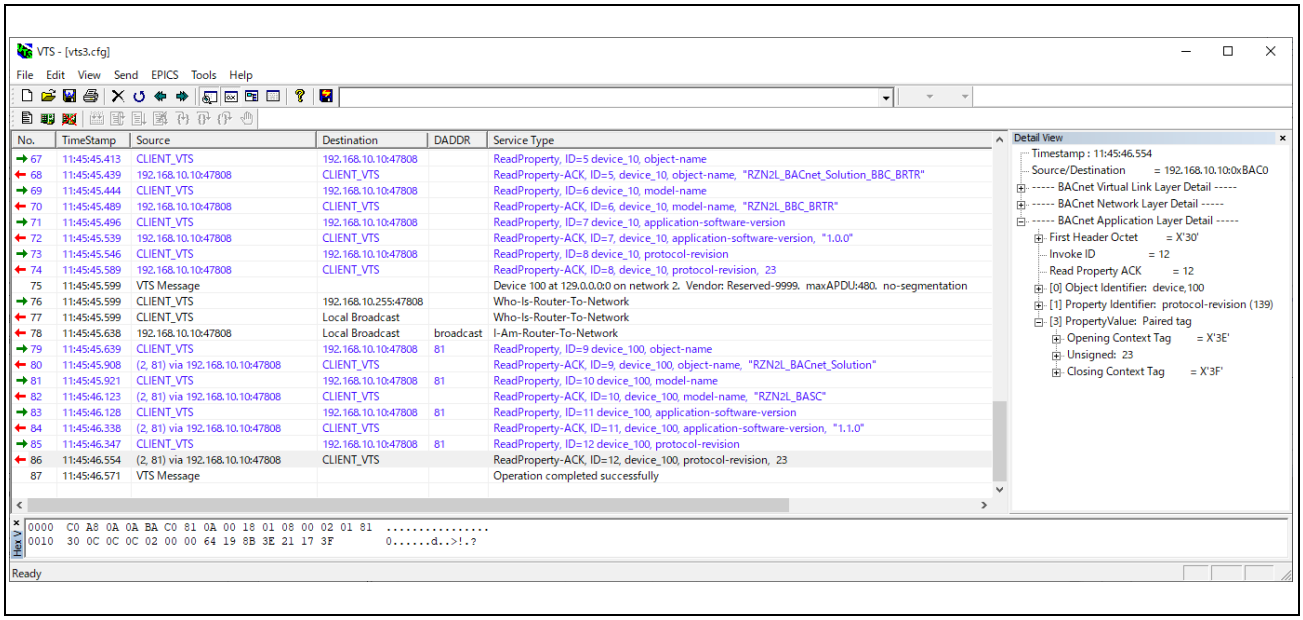

Fig.4-36 VTS log screen

(2) Remote Device Management

Who-Is can also be sent from the Send menu in addition to sending it from the Tools menu. Normal service requests can be sent from the Send menu. Click Remote Device Management > Who-Is.

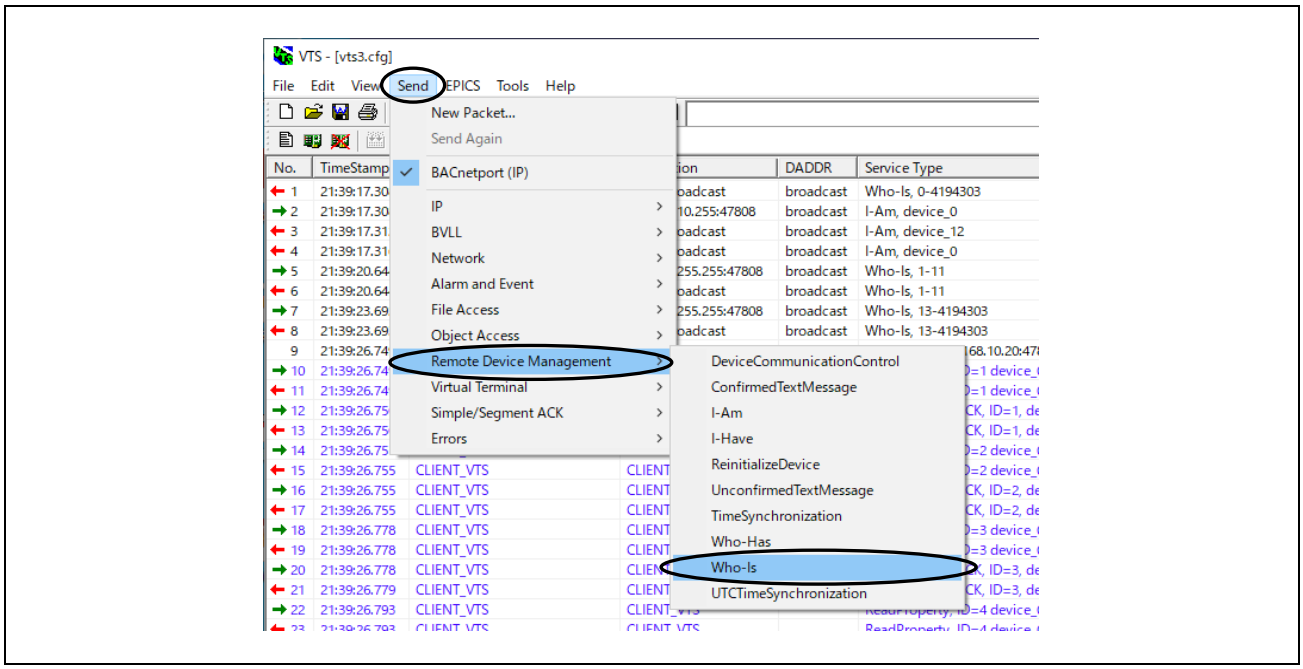

Fig.4-37 Send menu Who-Is

Click on the IP tag in the Who-Is dialog and select Global Broadcast or Local Broadcast then the address is automatically filled in.

Global Broadcast:255.255.255.255:47808

Local Broadcast:192.168.10.255:47808

You can also enter the address directly.

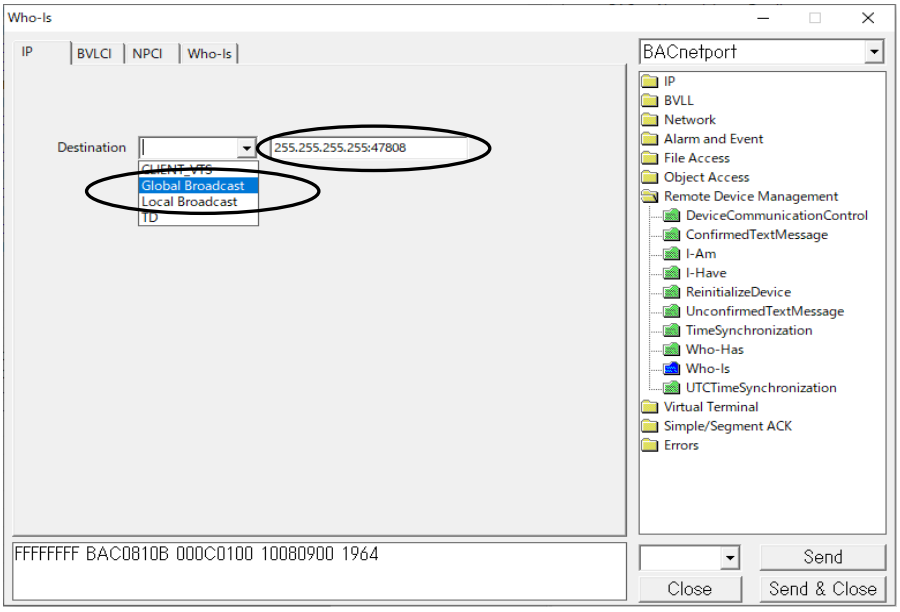

Fig.4-38 Who-Is parameters(1)

Click on the BVLCI tag in the Who-Is dialog and make sure that either Original Uncast or Original Broadcast is selected.

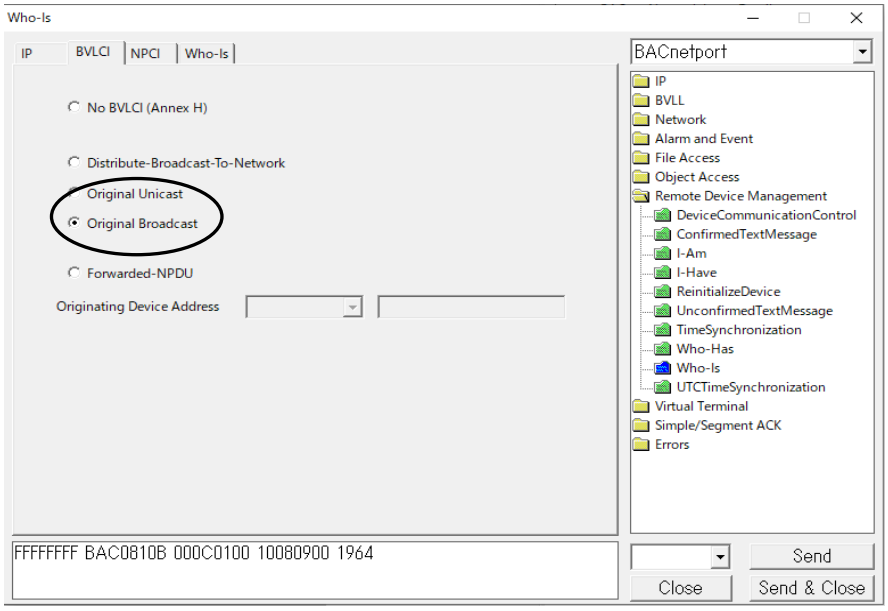

Fig.4-39 Who-Is parameters(2)

Click on the NPCI tag in the Who-Is dialog and make sure “DNET/DLEN/DADR Present” is selected.

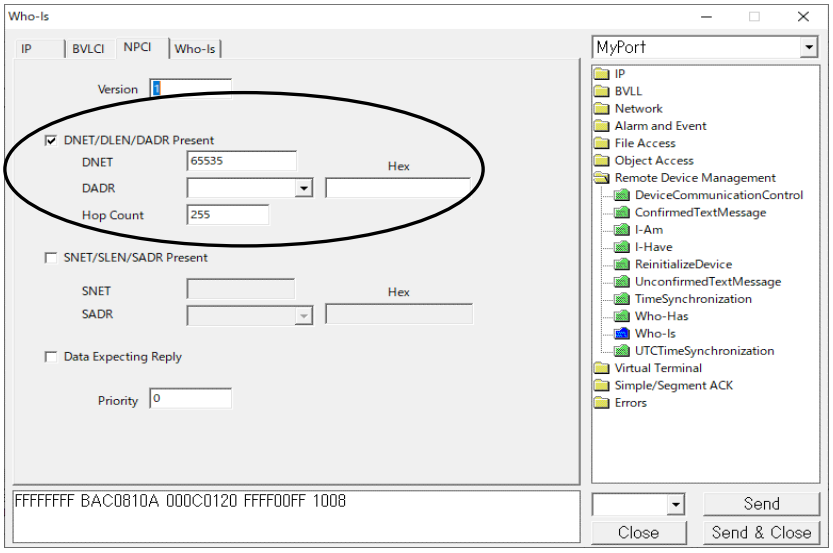

Fig.4-40 Who-Is parameters(3)

Click on the Who-Is tag in the Who-Is dialog and enter the ID range of the device to be searched. If blank, the range will be 0 to 4194303, which is the full range. Lastly, click on Send.

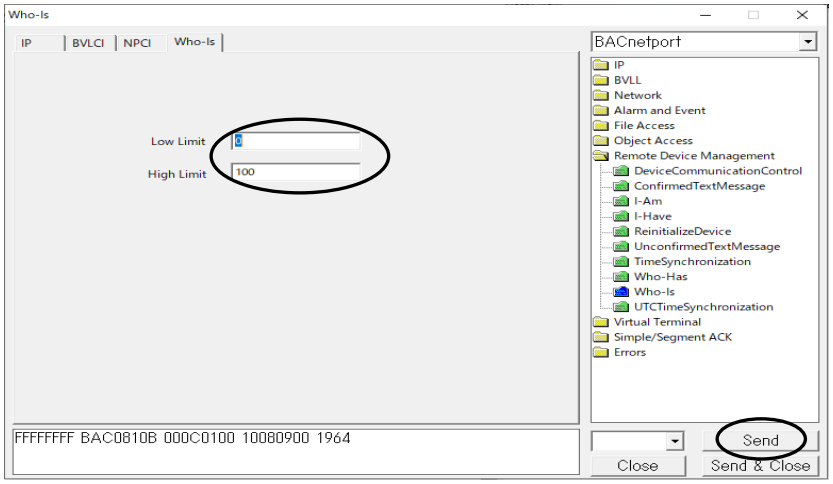

Fig.4-41 Who-Is parameters(4)

The following is a capture of I-Am responses from B-BC and B-SS to a Who-Is service request.

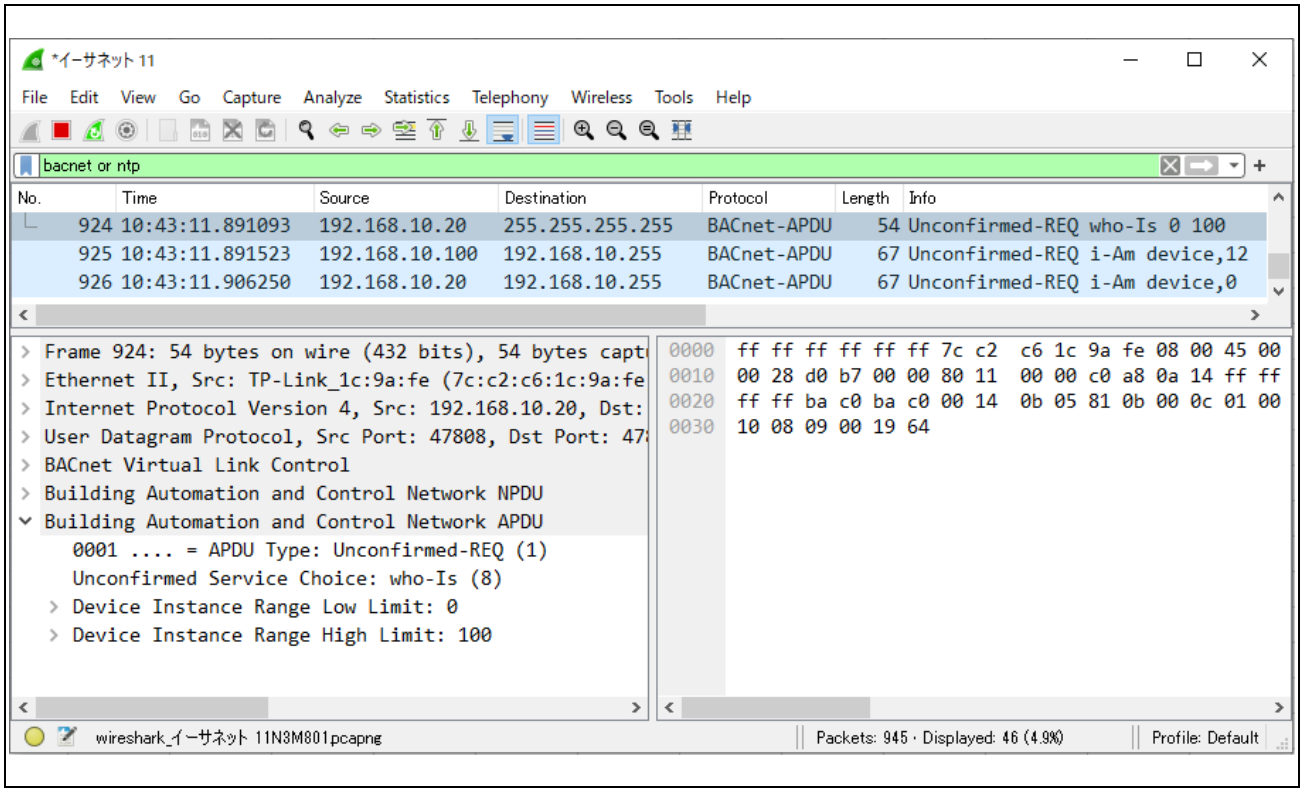

Fig.4-42 Capture Who-Is and I-Am

4.5.2 ReadProperty

Click Send menu > Object Access > ReadProperty.

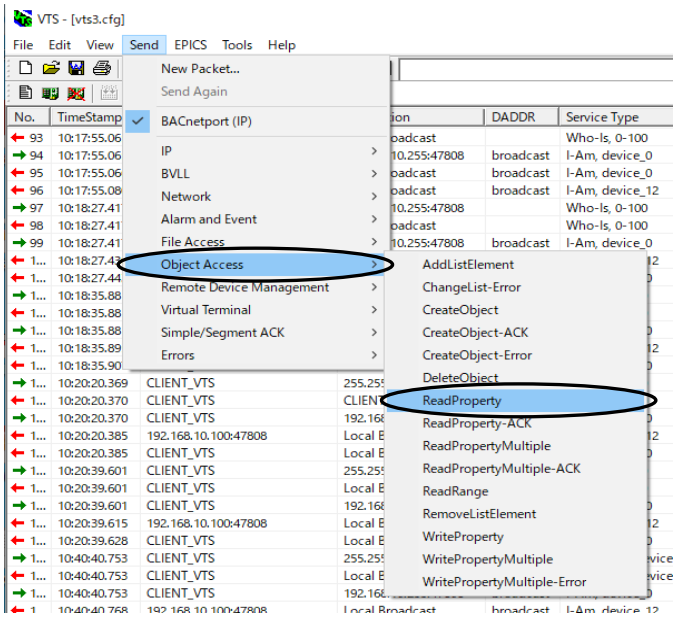

Fig.4-43 Send menu ReadProperty

Click on the IP tag in the ReadProperty dialog and fill in the B-BC server address directly.

192.168.10.10:47808

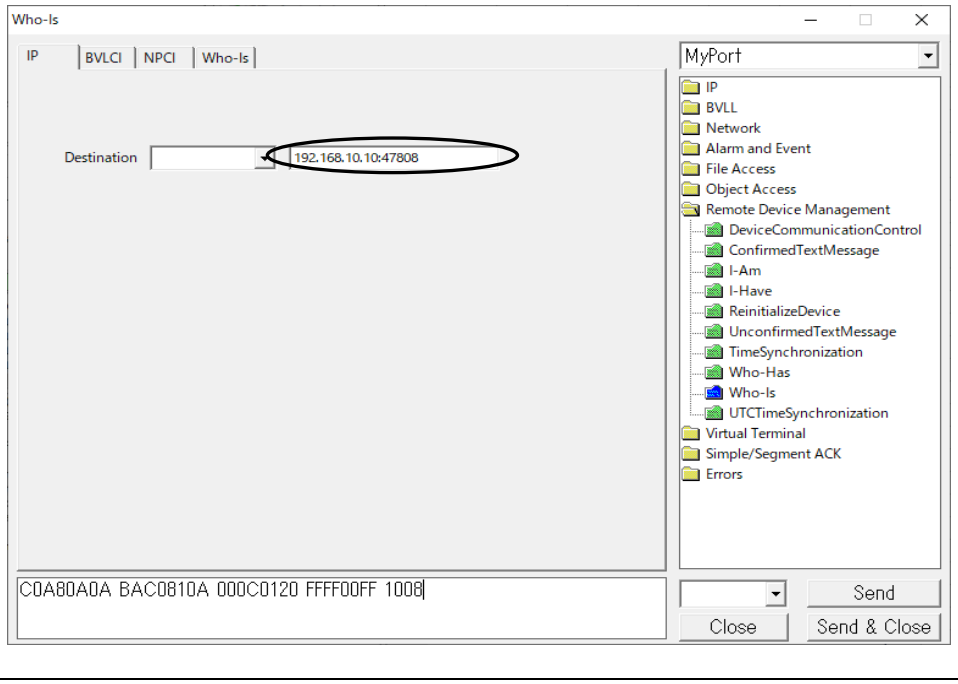

Fig.4-44 IP tag parameters

Click on the NPCI tag in the ReadProperty dialog and select B-BC or B-SS in this dialog.

The case of selecting B-BC is shown in Fig.4-45.

Verify that Data Expecting Reply is marked ✓.

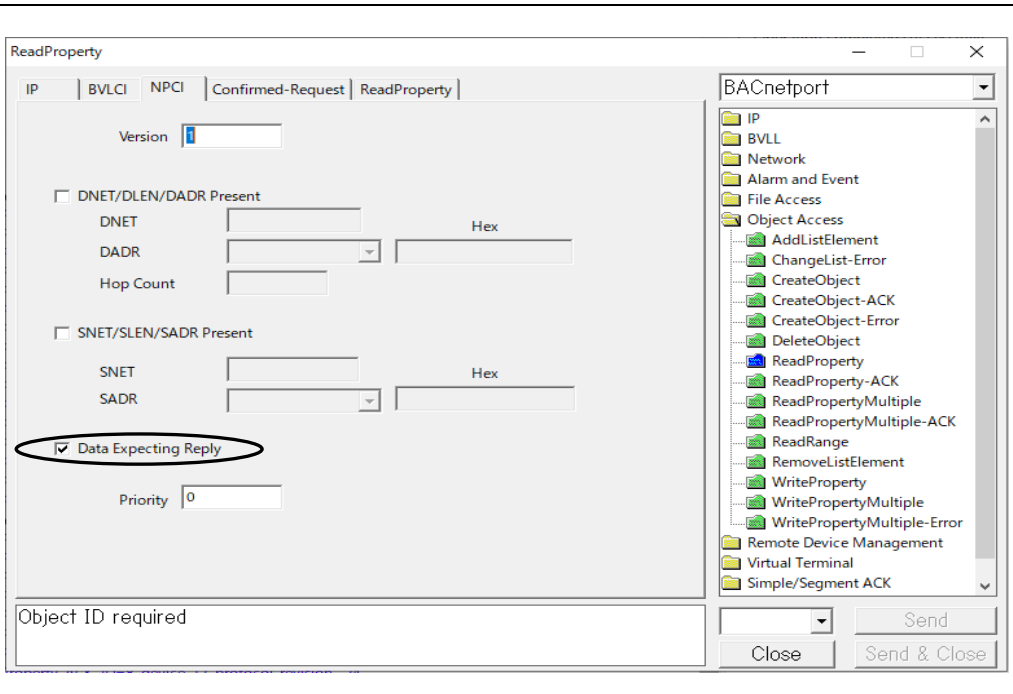

Fig.4-45 NPCI tag parameters to B-BC

The case of selecting B-BC is shown in Fig.4-46

Enter 2 of the Network_Number property value for B-SS in DNET. Enter (81) as Hex of 129 of the B-SS's MAC address in DADR. Enter 255 for Hop Count.

The settings in the NPCI dialog are the same when requesting each service other than ReadProperty to B-SS.

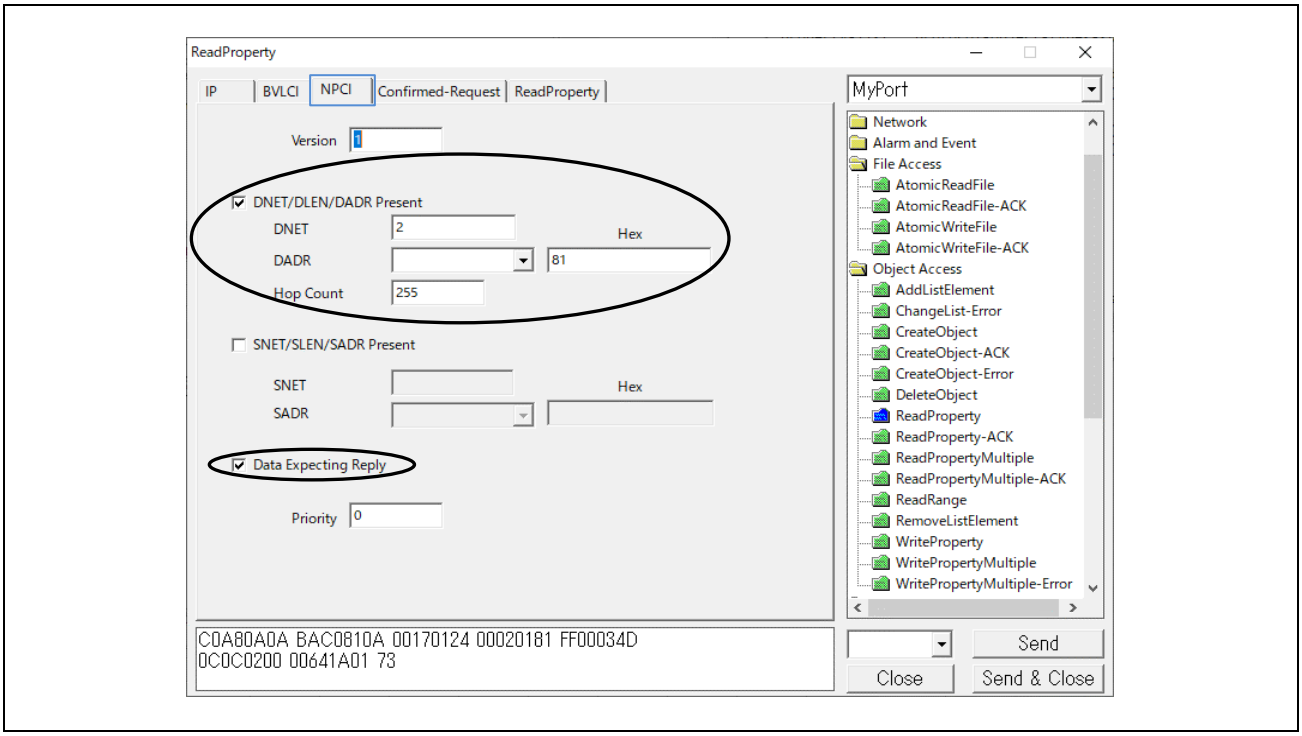

Fig.4-46 NPCI tag parameters to B-SS

Click on the Confirmed-Request tag in the ReadProperty dialog and select **480** for Max APDU length accepted.

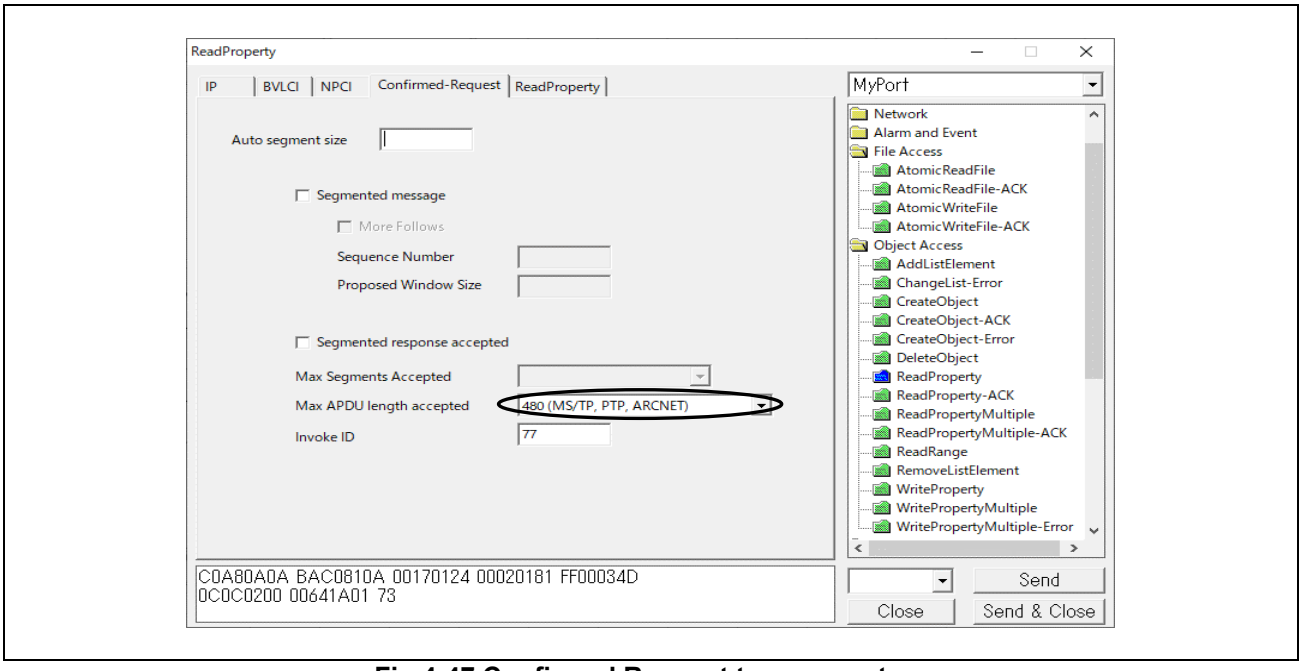

Fig.4-47 Confirmed Request tag parameters

Click on the ReadProperty tag in the ReadProperty dialog, select ID... > Object Type and enter that Instance.

In the example below, device,100 is input for B-SS, but if B-BC is selected, device,10 should be input.

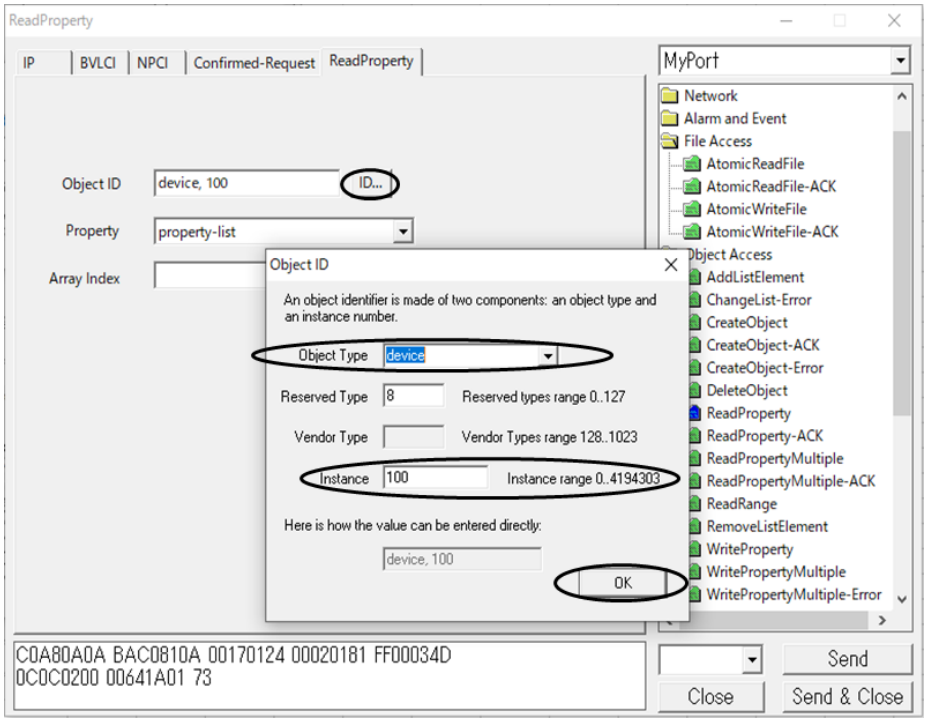

Fig.4-48 ReadProperty parameters

Then select Property. In the example, property-list is selected. Lastly, click Send.

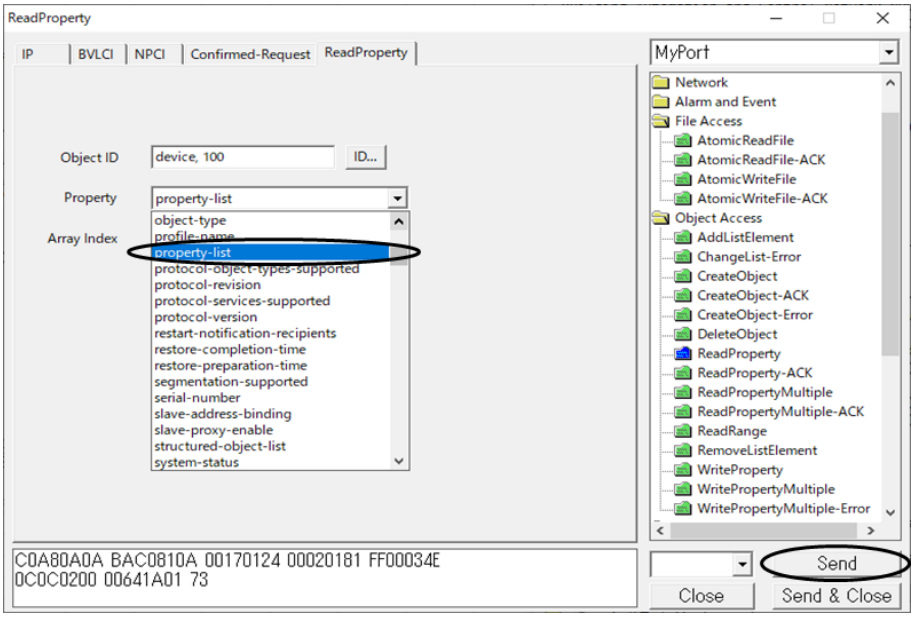

Fig.4-49 ReadProperty parameters(2)

B-SS responds with Complex-Ack for the property-list property of the devive,100 object.

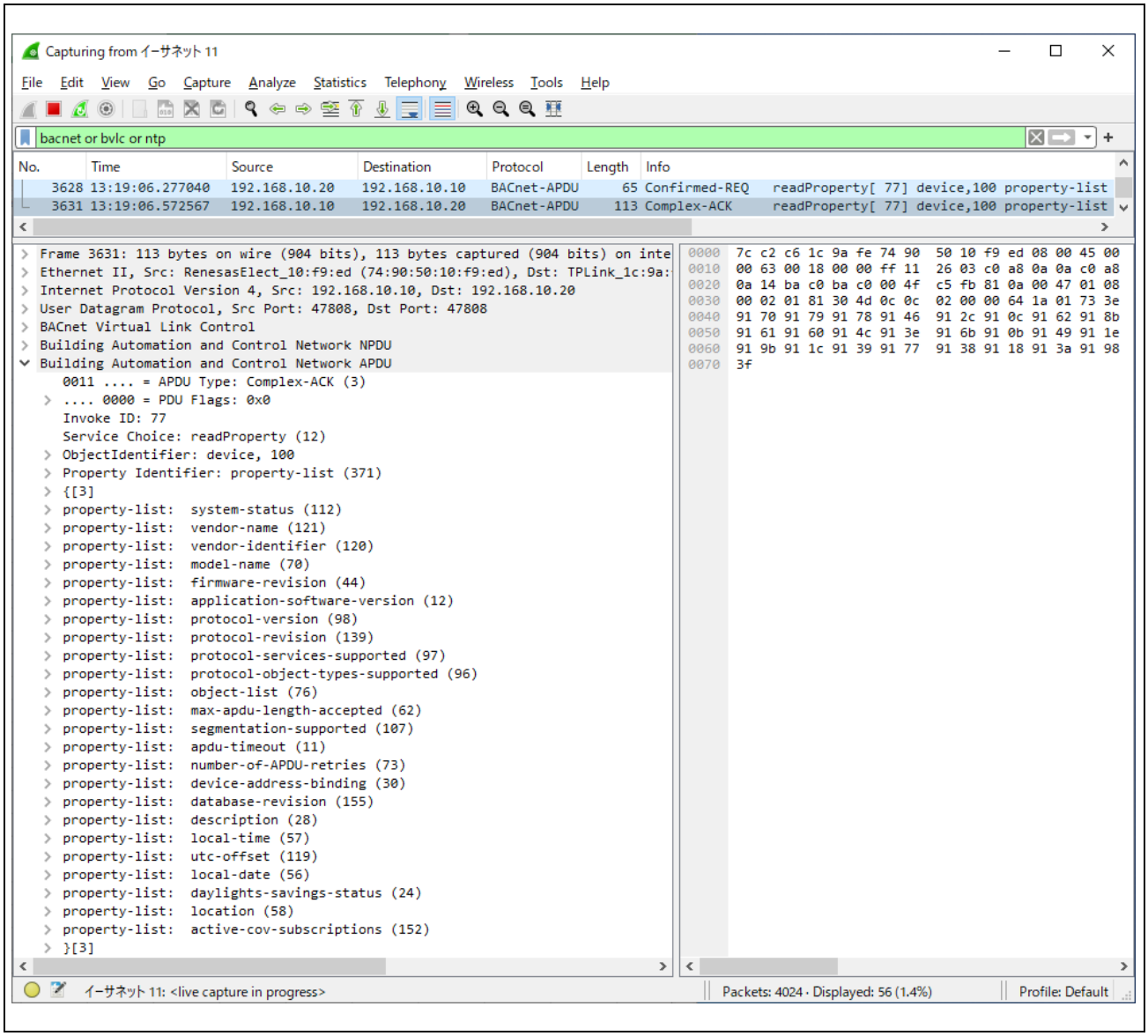

Fig.4-50 Capture ReadProperty device,100 property-list

4.5.3 TimeSynchronization / UTCTimeSynchronization

Double-click Remote Device Management from the service tree displayed in the previously used dialog.

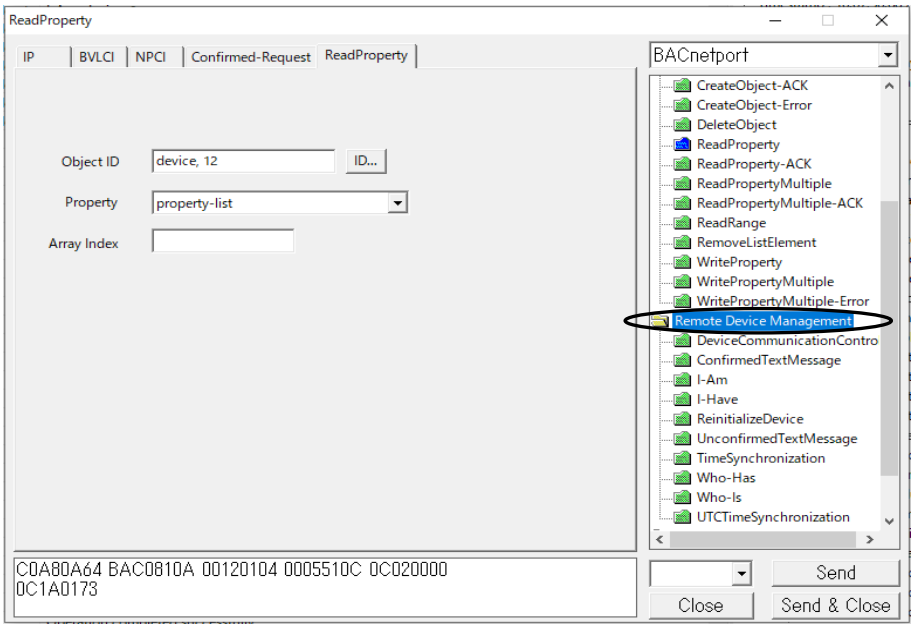

Fig.4-51 TimeSynchronization parameters(1)

Click on TimeSynchronization or UTCTimeSynchronization in the expanded Remote Device Management. Mark ✓ at Synchronize with VTS in the TimeSynchronization tag and click Send. If you want to enter a time other than the current time, remove the ✓ from Synchronize with VTS and enter the Date and Time directly.

For the destination IP address, the previous value is applied. To change the destination IP address, enter the IP address from the IP tag. The same applies hereafter.

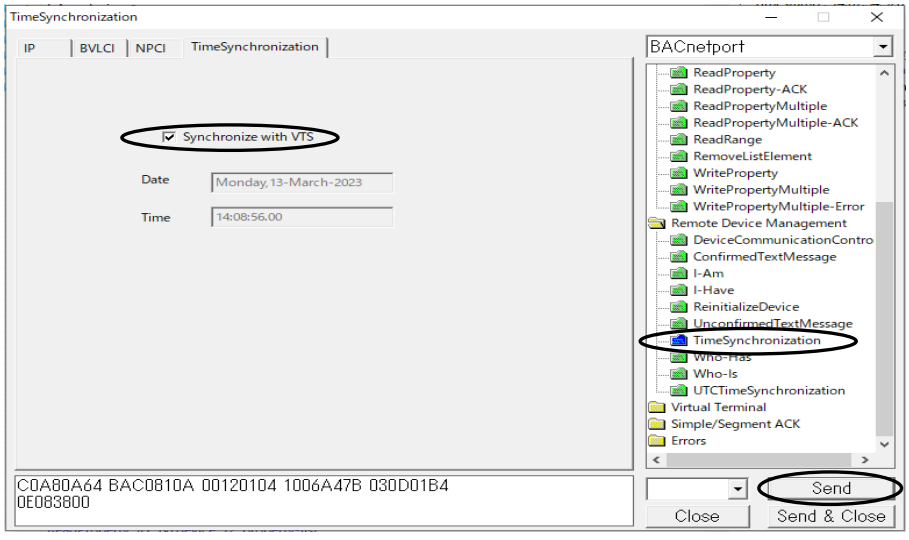

Fig.4-52 TimeSynchronization parameters(2)

TimeSynchronization is an unconfirmed service. However, the destination IP address can be unicast. The following captured screen shows a unicast from a VTS client to a B-BC server.

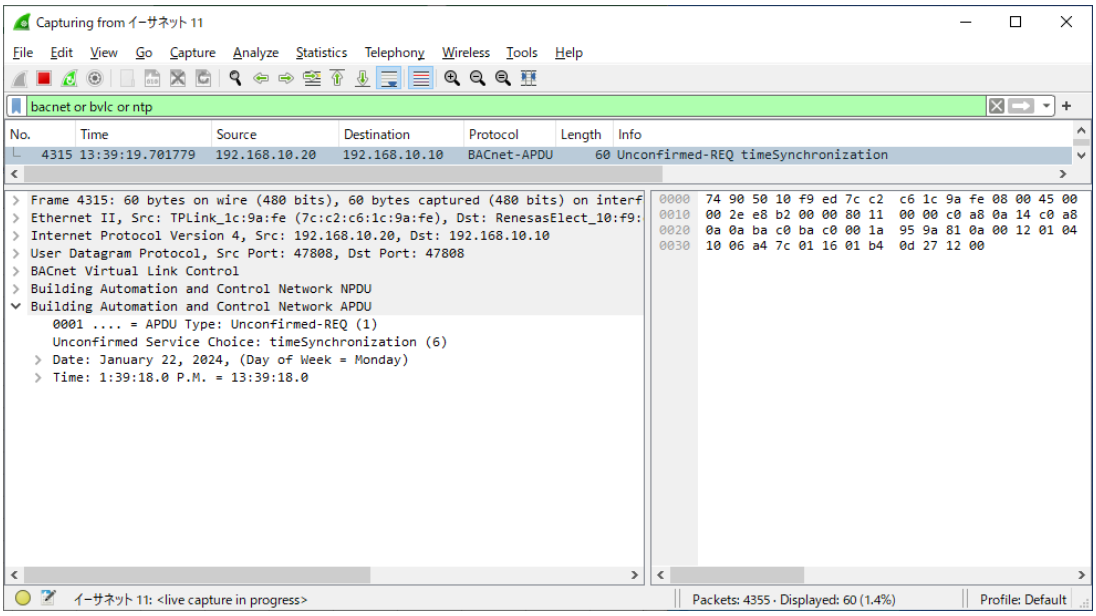

Fig.4-53 Capture TimeSynchronization

The same applies for sending the UTCTimeSynchronization service instead of the TimeSynchronization service.

4.5.4 Who-Has / I-Have

Click Remote Device Management > Who-Has in the Service Tree displayed in the previously used dialog. Enter the Low Limit and High Limit for the search ID ranges. Select the object type from the Object ID pull-down menu, enter the instance number, then click OK.

In the example, the analog-input,0 object is selected; either Object ID or Object Name can be selected.

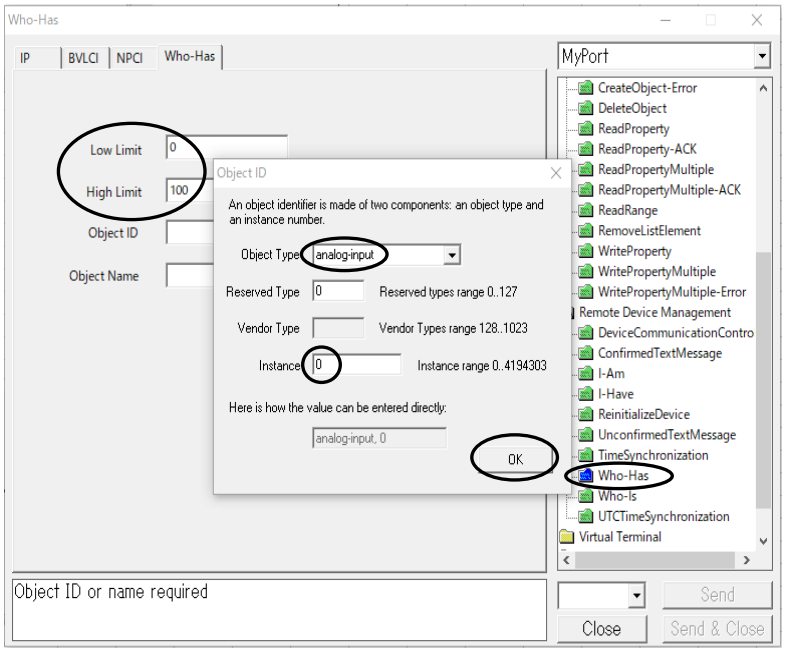

Fig.4-54 Who-Has parameters(1)

The following is an example of entering an Object Name.

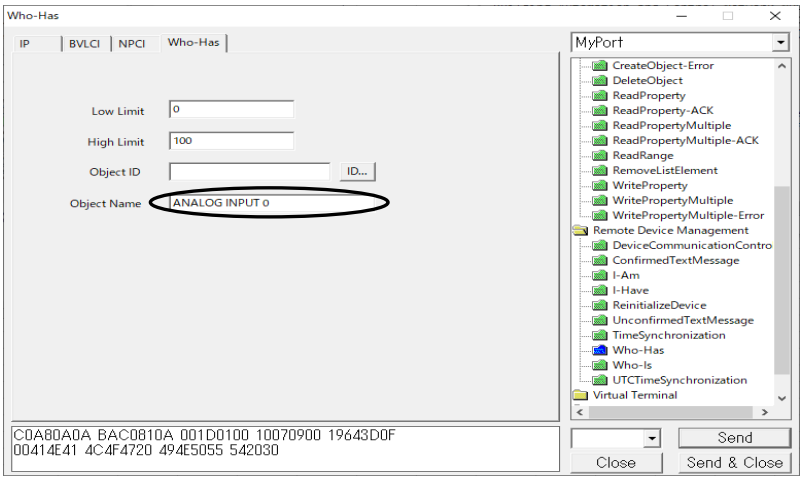

Fig.4-55 Who-Has parameters(2)

Next click on the IP tag and select Global Broadcast or Local Broadcast, then click Send.

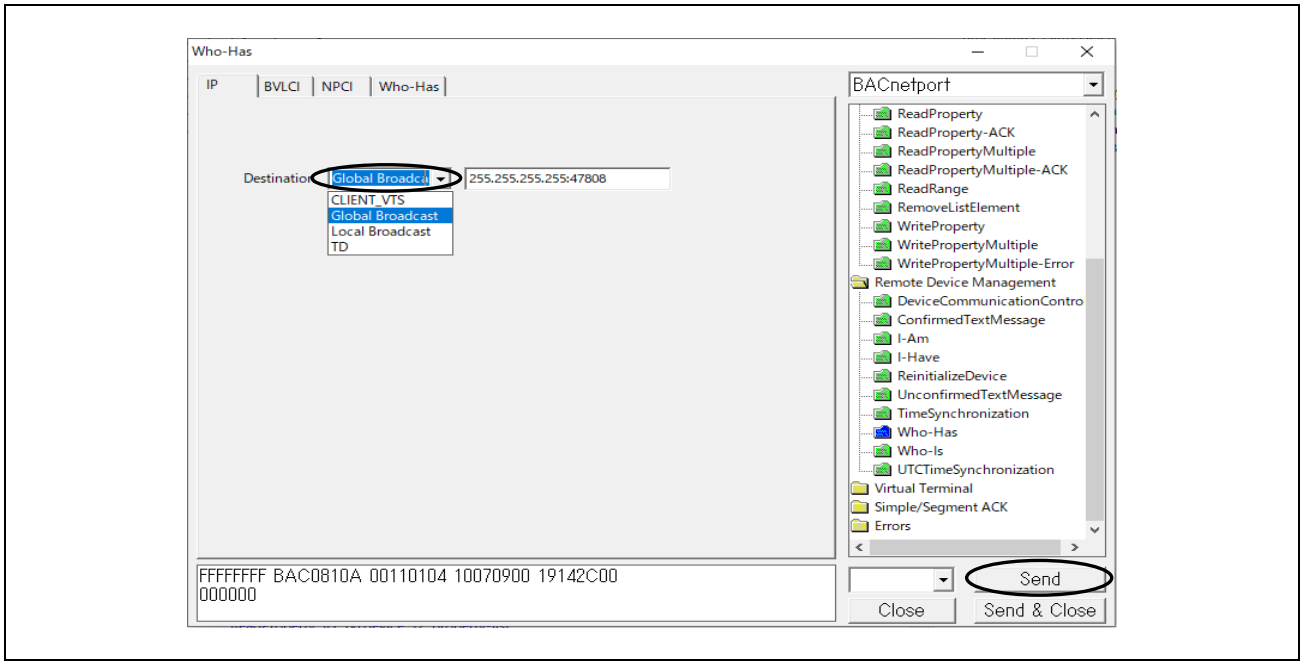

Fig.4-56 Who-Has parameters(3)

In the following example, the B-BC server locally broadcasts an I-Have response to a device search broadcast with an analog-input,0 object by a Who-Has service request.

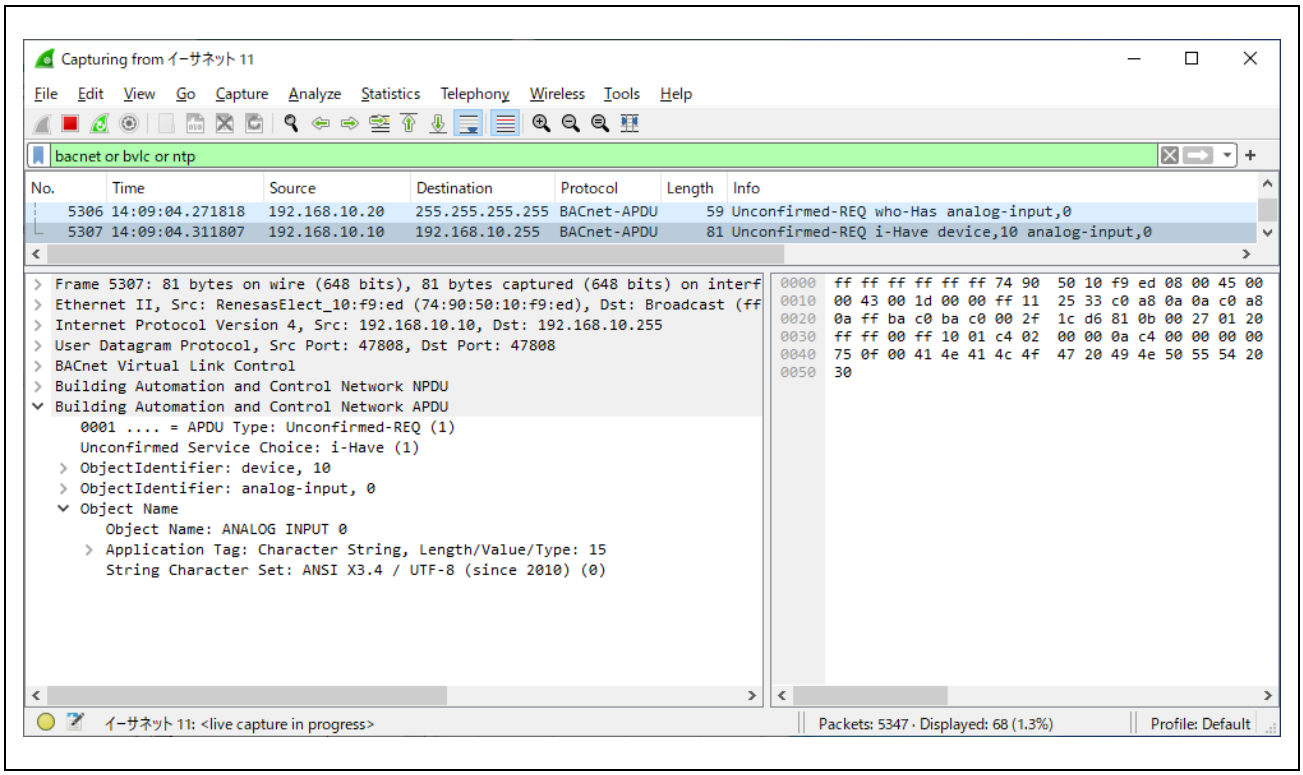

Fig.4-57 Capture Who-Has and I-Have

4.5.5 ReadPropertyMultiple

Click Object Access > ReadPropertyMultiple from the service tree displayed in the previously used dialog.

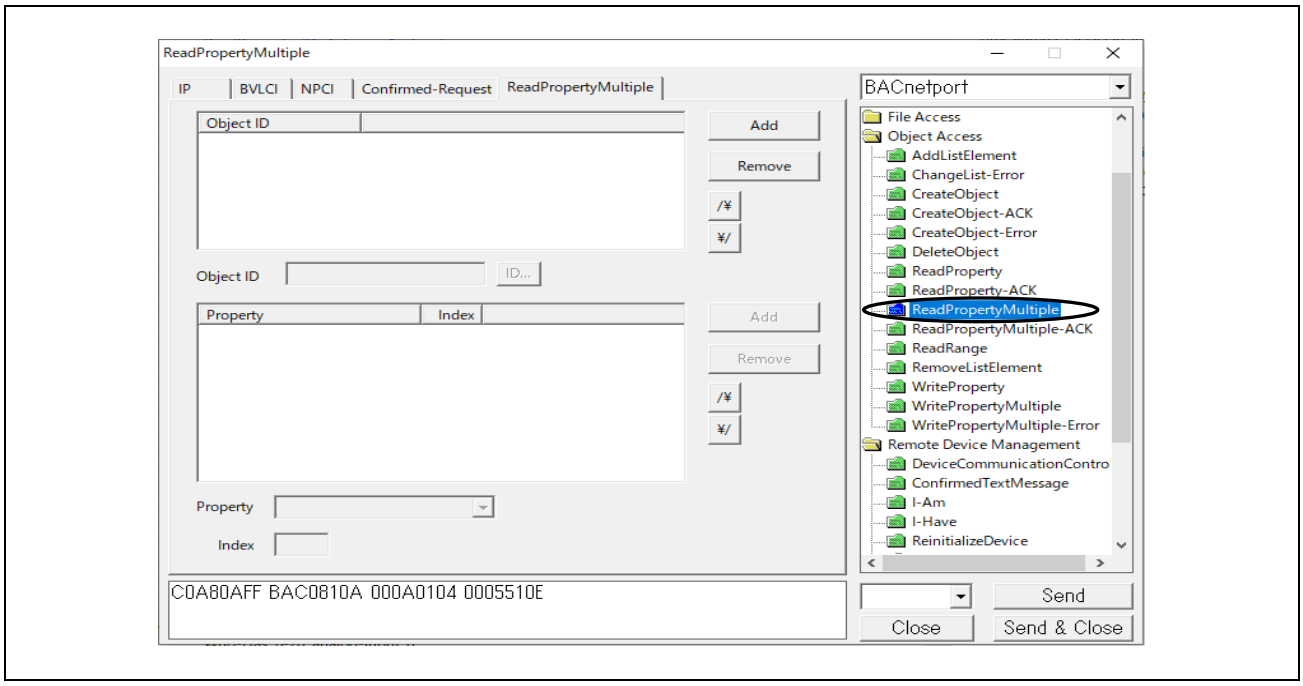

Fig.4-58 ReadPropertyMultiple parameters(1)

Click Add at the upper part of the ReadPropertyMultiple dialog displayed. Click ID..., select an Object ID, and then click OK. In the example, the device,10 object is selected.

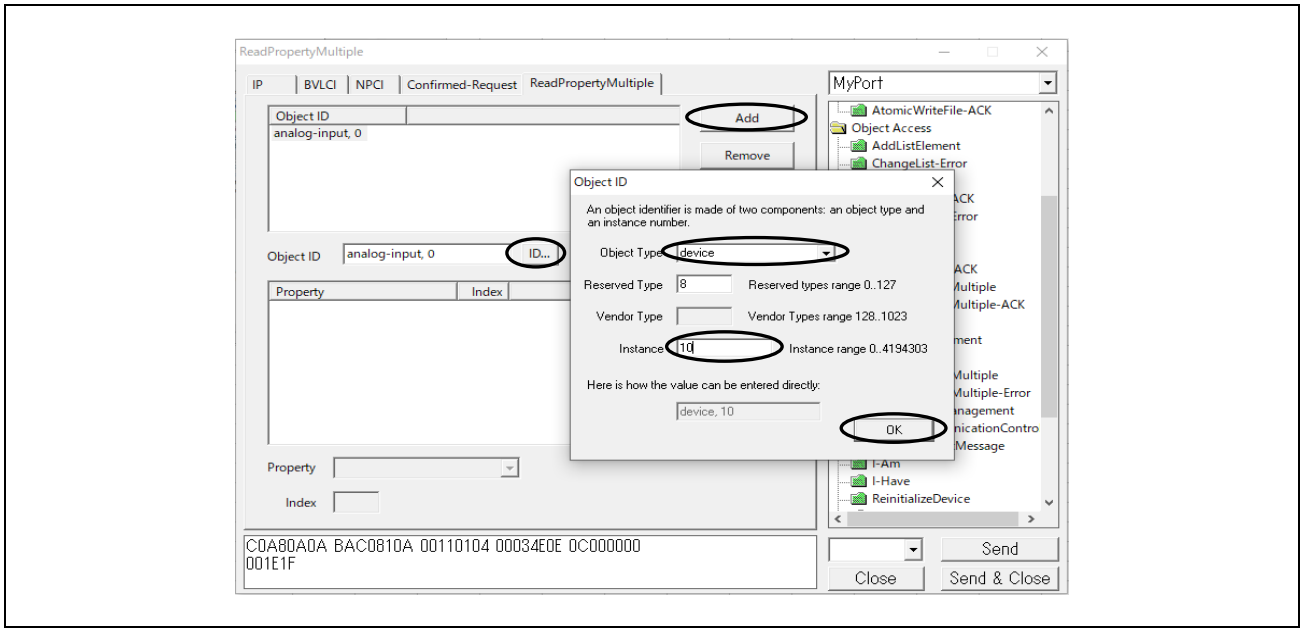

Fig.4-59 ReadPropertyMultiple parameters(2)

Click Add in the center of the dialog; select a property from the Property pull-down menu. In the example "all" property is selected; then click on Send.

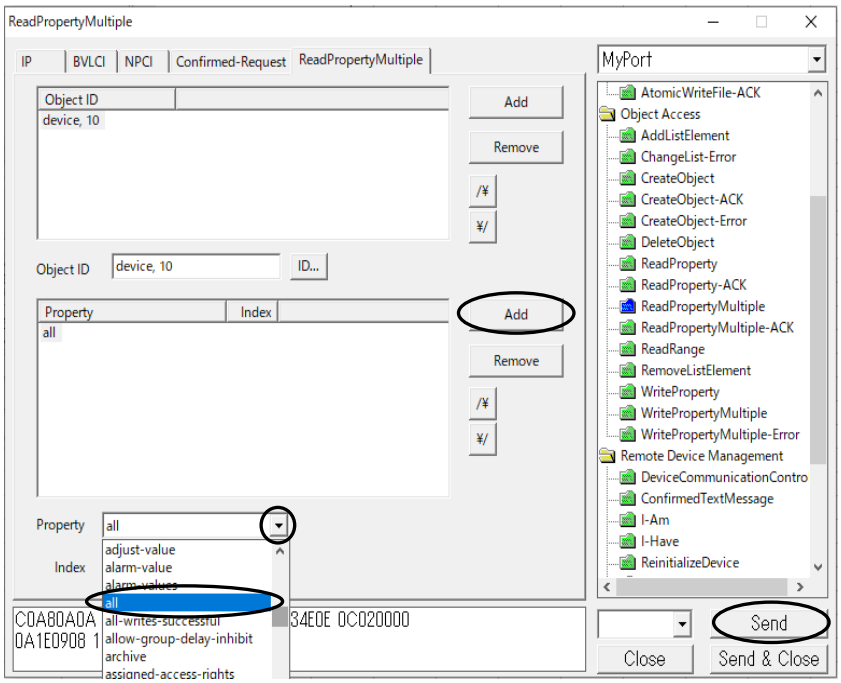

Fig.4-60 ReadPropertyMultiple parameters(3)

The following example shows a "all" property request of a device,10 object by the ReadPropertyMultiple service from a VTS client and a Complex-Ack response with results from the B-BC server.

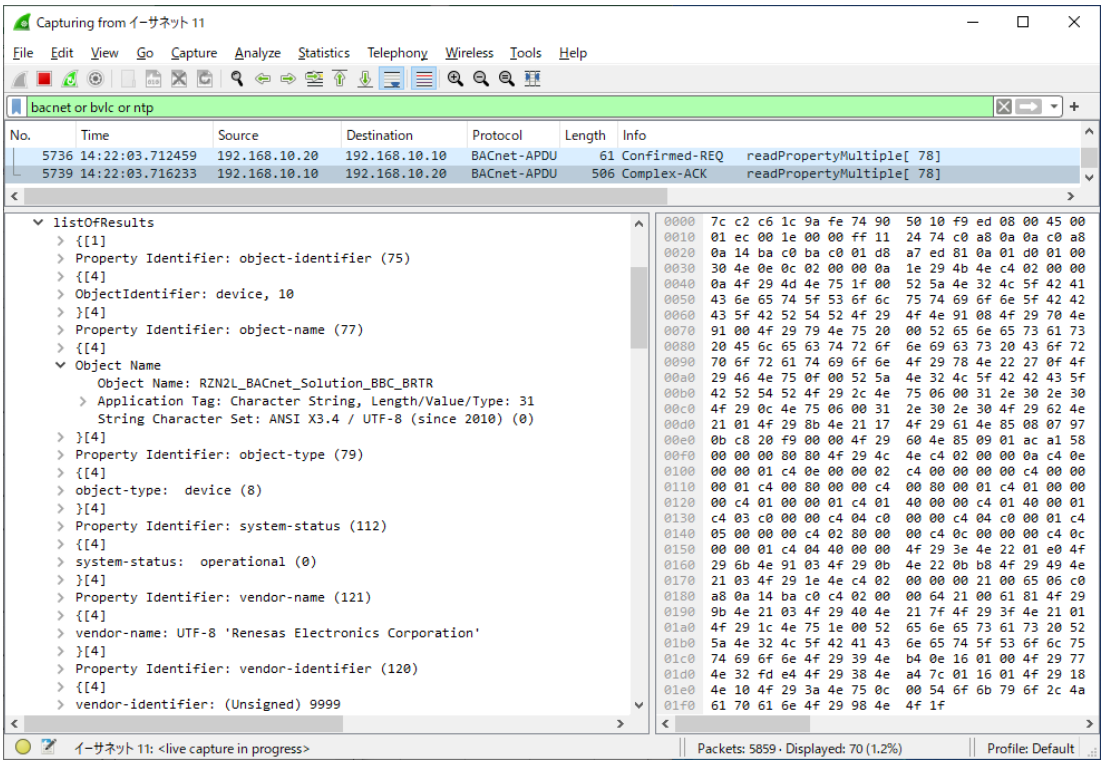

Fig.4-61 Capture ReadPropertyMultiple device,10 all

4.5.6 WriteProperty

Click Object Access > WriteProperty from the service tree displayed in the previously used dialog. Select the Object ID then click OK. In the example, the multi-state-value,0 object is selected.

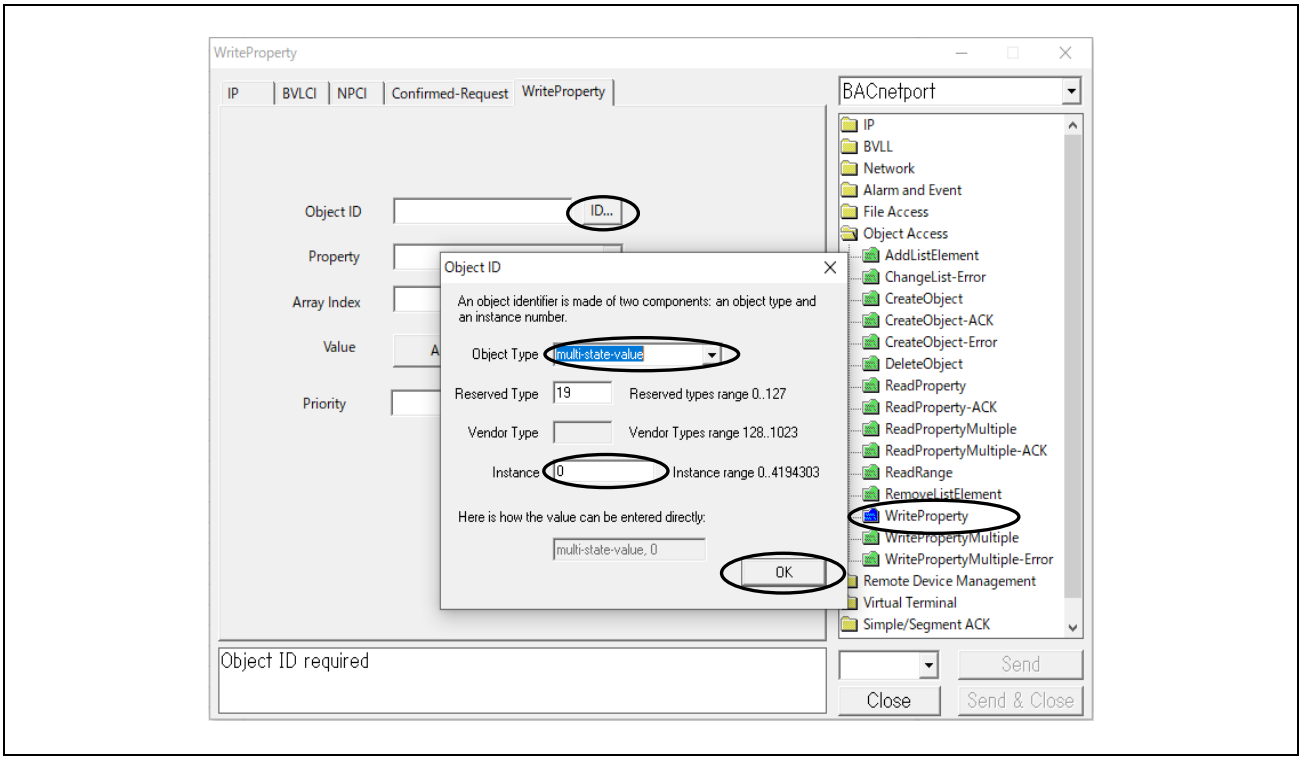

Fig.4-62 WriteProperty parameters(1)

Next, select present-value from Property pull-down list.

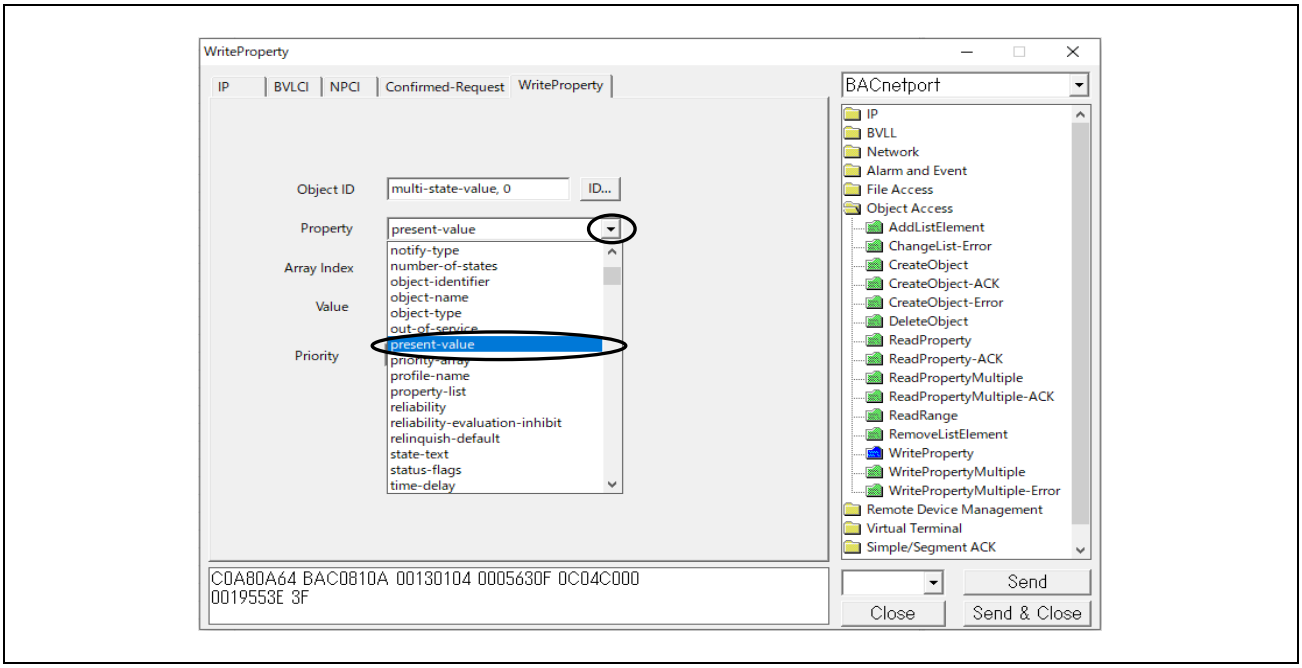

Fig.4-63 WriteProperty parameters(2)

Next click Any....By clicking Add in the pop-up dialog, Null is displayed. With this Null selected, choose the data type from the Type pull-down menu. In the example, Unsigned, which is a multi-state-value data type, is selected.

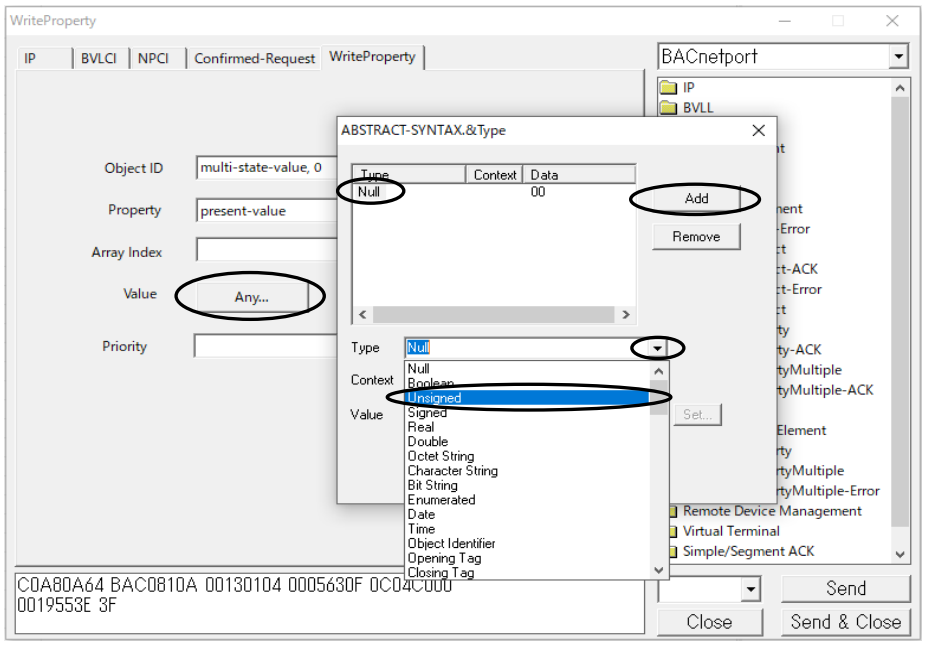

Fig.4-64 WriteProperty parameters(3)

Input the Value of the popup dialog. For this sample software, the setting range of multi-state-value is 1 to 3. Also, multi-state-value must be greater than 0 at all times. Furthermore, the data type of each property is strictly defined one by one in the standard, so the data type displayed in the Type pull-down menu must be appropriately selected according to the standard. Refer to [12 MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS] in the standard.

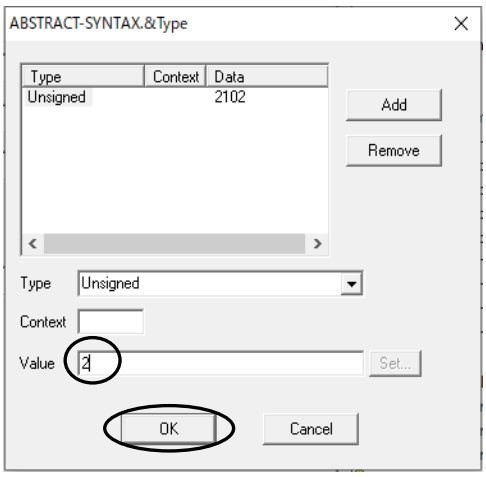

Fig.4-65 WriteProperty parameters(4)

Then click Send.

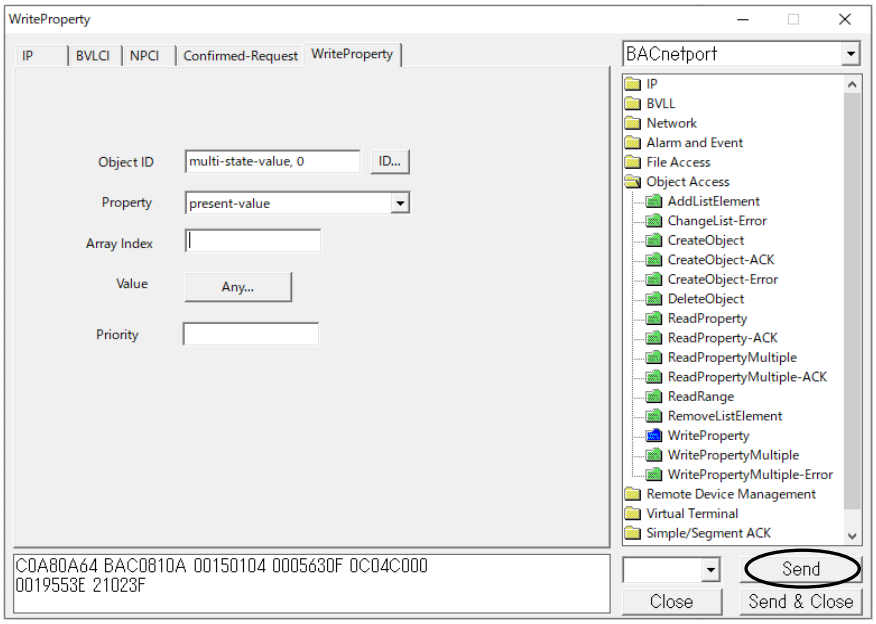

Fig.4-66 WriteProperty parameters(5)

The following example shows a request from a VTS client to change the present-value property of a multi-state-value,0 object by the WriteProperty service and the Simple-Ack response from the B-SS server.

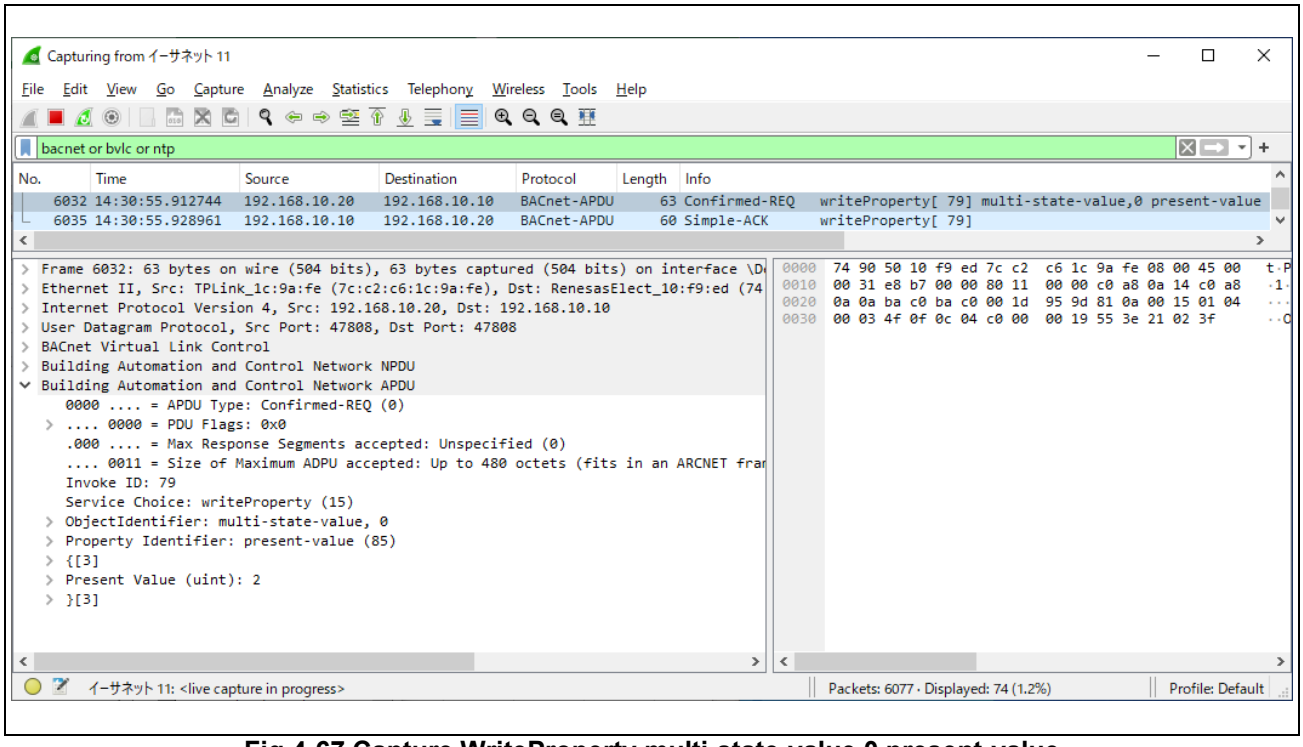

Fig.4-67 Capture WriteProperty multi-state-value,0 present-value

4.5.7 WritePropertyMultiple

Click Object Access > WritePropertyMultiple from the service tree displayed in the previously used dialog. Select an object by clicking Add, ID... in Object ID, and then click OK.

In the example, the procedure for verifying the operation of ConfirmedEventNotification from B-BC to VTS clients by using the WritePropertyMultiple service is provided. The notification-class,0 object is selected in Fig.4-68.

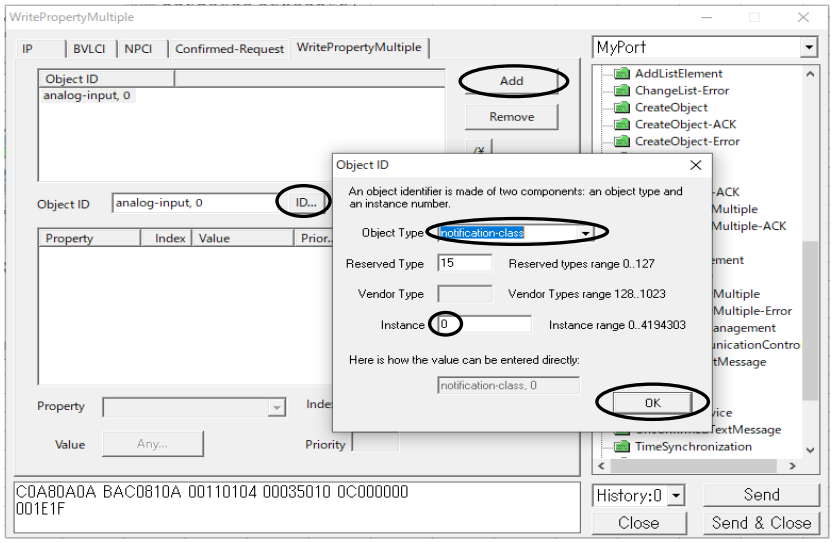

Fig.4-68 WritePropertyMultiple parameters(1)

Click Add on Property and select ack-required. Clicking on "Any..." will display a dialog box for selecting the data type. Click Add and select a data type from Type. Click "Set..." to display the corresponding dialog and select a value. Click OK to close the respective dialog.

In the example, the data type EventTransitionBits is selected, and all three event notification conditions are selected.

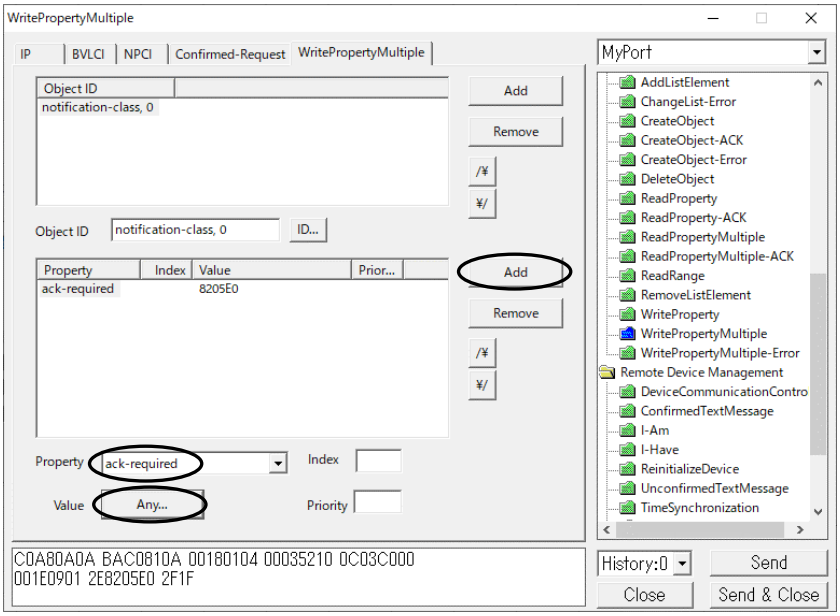

Fig.4-69 WritePropertyMultiple parameters(2)

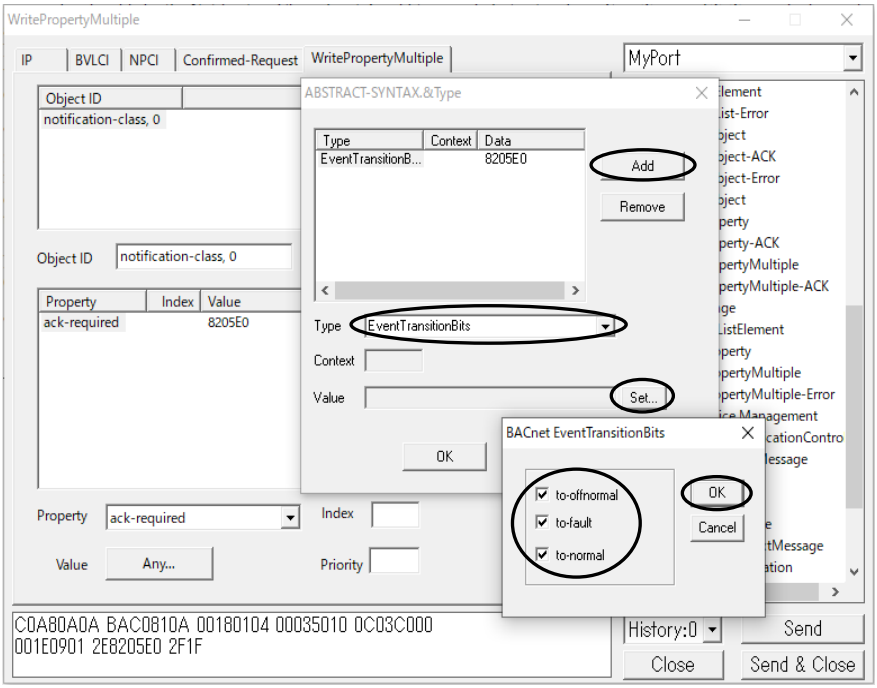

Fig.4-70 WritePropertyMultiple parameters(3)

As well, click on Add in Property and select recipient-list.

In the example, the data type Destination is selected, the event recipient is VTS(device,0), Ack from VTS is available, all three event notification conditions are selected, and the start and end of the validity period are set.

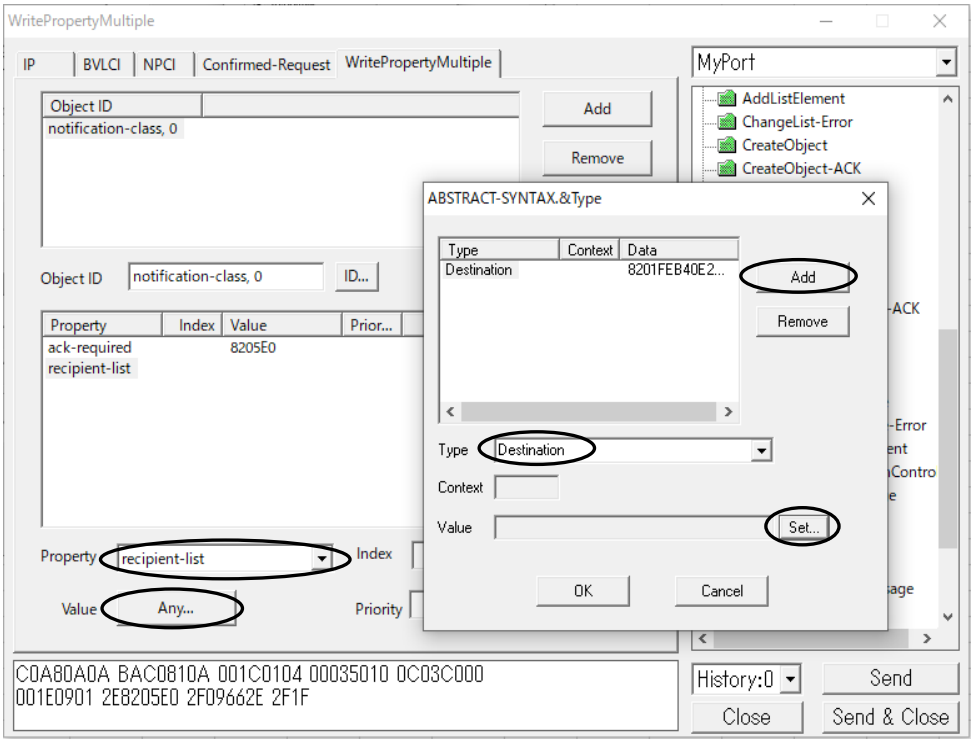

Fig.4-71 WritePropertyMultiple parameters(4)

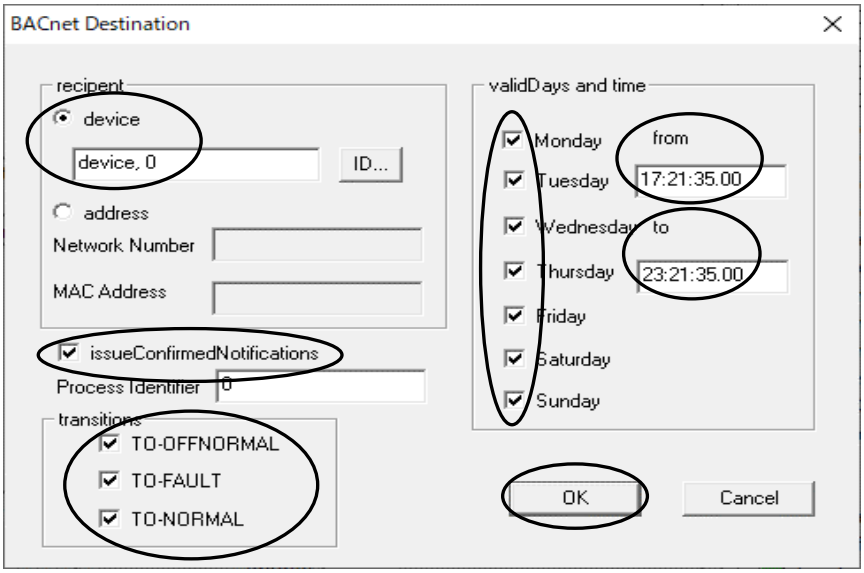

Fig.4-72 WritePropertyMultiple parameters(5)

Furthermore, click on Add in Property and select priority. Finally, click Send.

In the example, the data type PriorityArray is selected and the priorities (255: lowest) corresponding to the three notification conditions are set.

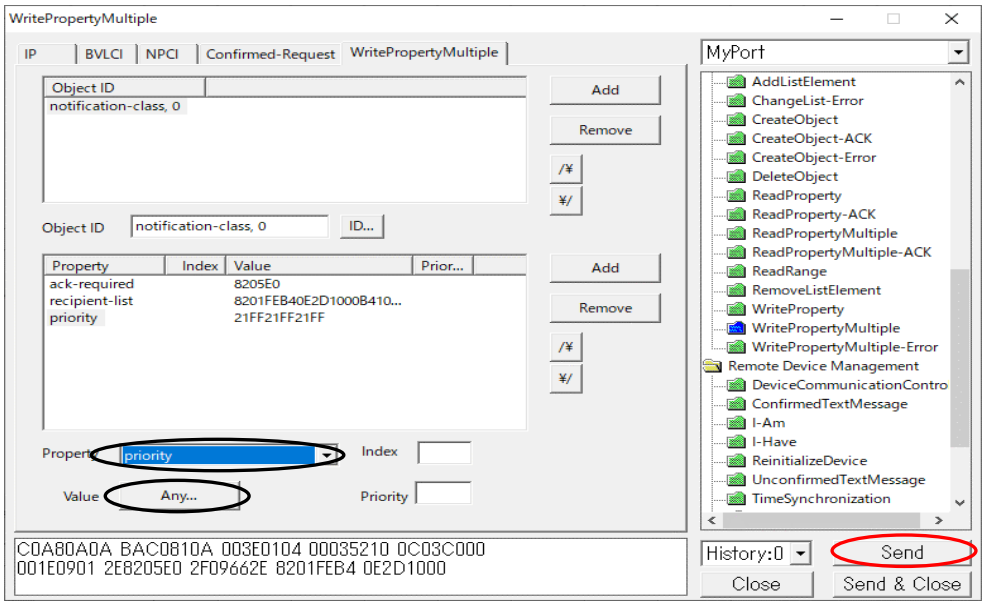

Fig.4-73 WritePropertyMultiple parameters(6)

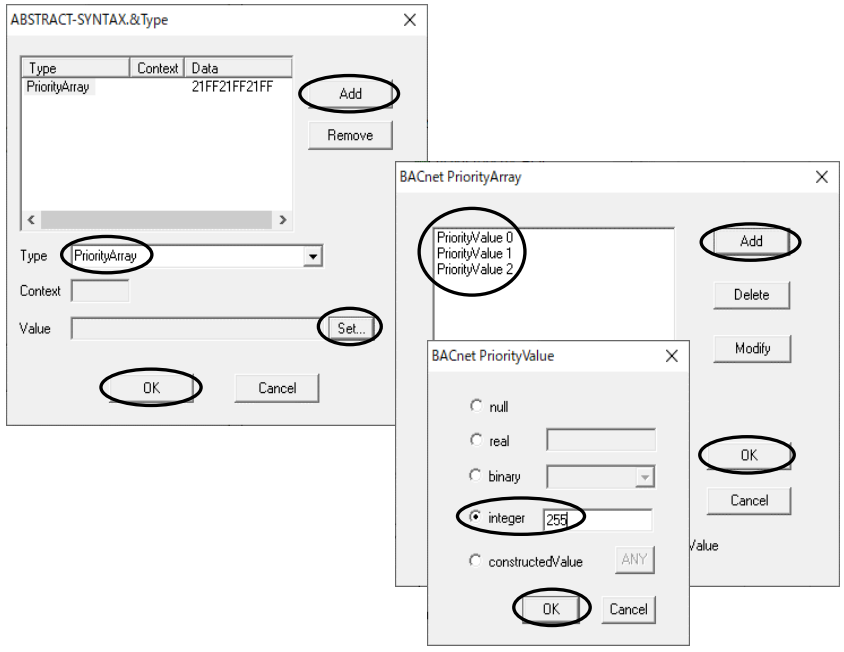

Fig.4-74 WritePropertyMultiple parameters(7)

Next, select the old value (notification-class,0) once set for the Object ID and click Remove to delete the old setting. Select the object by clicking on the new Object ID Add, ID..., in that order, and click OK.
In the example, the analog-input,0 object is selected (Fig.4-82). Click Add in Property and set out-of-service=true and present-value=-0.1 (Fig.4-76) With this setting, you can confirm that the event notification occurs just below the LowLimit property value (0.0) of the OutOfRange event algorithm. Incidentally, an event notification will also occur if the HighLimit property value (100.0) is exceeded. Finally, click Send.

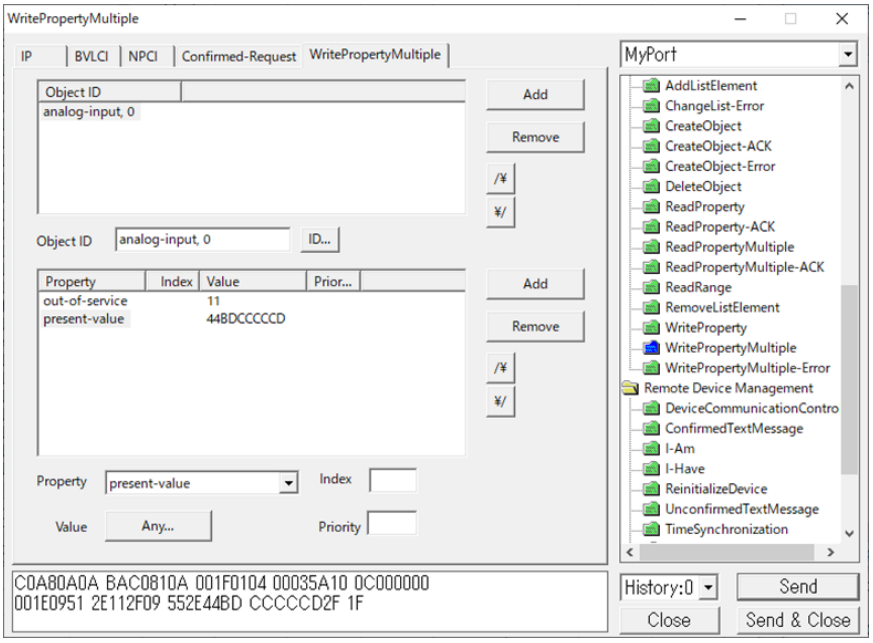

Fig.4-75 WritePropertyMultiple parameters(8)

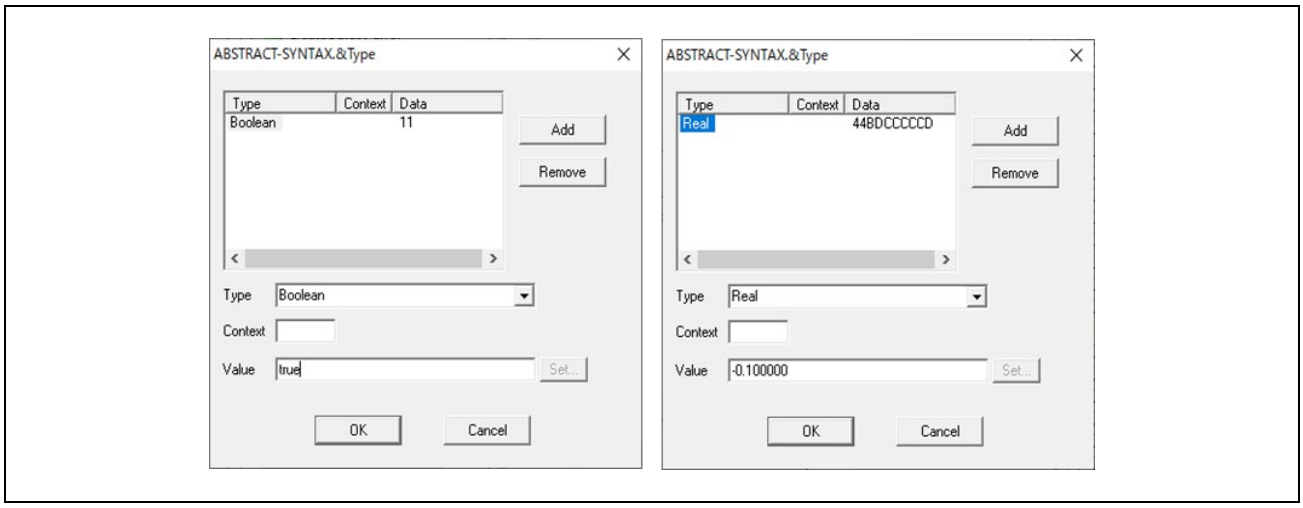

Fig.4-76 WritePropertyMultiple parameters(9)

Below is the packet capture for this. The notification-class,0 object and analog-input,0 object are set by the WritePropertyMultiple service from the VTS client. Then, ConfirmedEventNotification service is requested from the B-BC server and the VTS client returns an Ack.

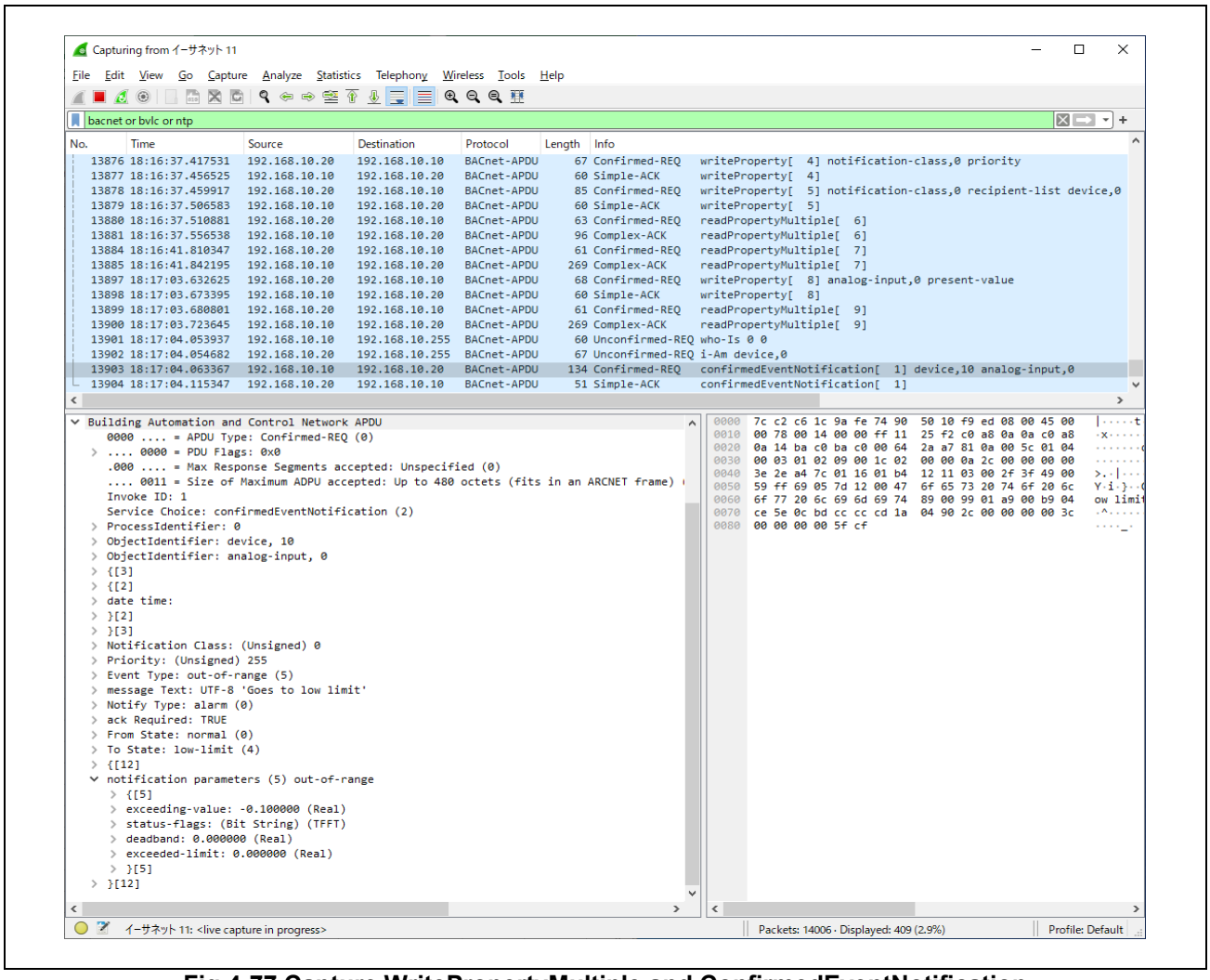

Fig.4-77 Capture WritePropertyMultiple and ConfirmedEventNotification

4.5.8 SubscribeCOV

Click Alarm and Event > SubscribeCOV from the service tree displayed in the previously used dialog.

- The Subscribe Process ID is used to identify the COV client, i.e., the process in the VTS. This ID is used by the B-SS server to identify which COV client it is when the COV notification is made or when the subscribe is canceled. The value 0 is reserved and is used for the COV notification to be made when there is no subscriber. The COV server uses it to broadcast the outdoor air temperature etc.
- The Monitored Object ID is used to specify the object that holds the property to detect changes in value.
- Issue Confirmed Notifications are specified as True/False. If True, the COV client specify a ConfirmedCOVNotification to the COV server. When the COV client receives a COV notification, it returns Ack response to the COV server. The COV server does not send the next COV notification before receiving the Ack. If False, UnconfirmedCOVNotification is specified.
- Lifetime is the subscribe period and the unit is minutes. The COV server will not notify the COV client corresponding to the Subscribe Process ID after the Lifetime has elapsed. If this parameter is left blank, it indicates an indefinite period of time. If the COV client cancels the subscribe, both Issue Confirmed Notifications and Lifetime should be left blank.

Click Send to send a SubscribeCOV service request. then click Send.
In the example, an analog-input,0 object is selected that is assigned the input value of the air velocity sensor, and an unconfirmed COV notification is specified for an indefinite period of time. Fig.4-78 selects B-SS as the destination for the SubscribeCOV service request, the same as in Fig.4-46.

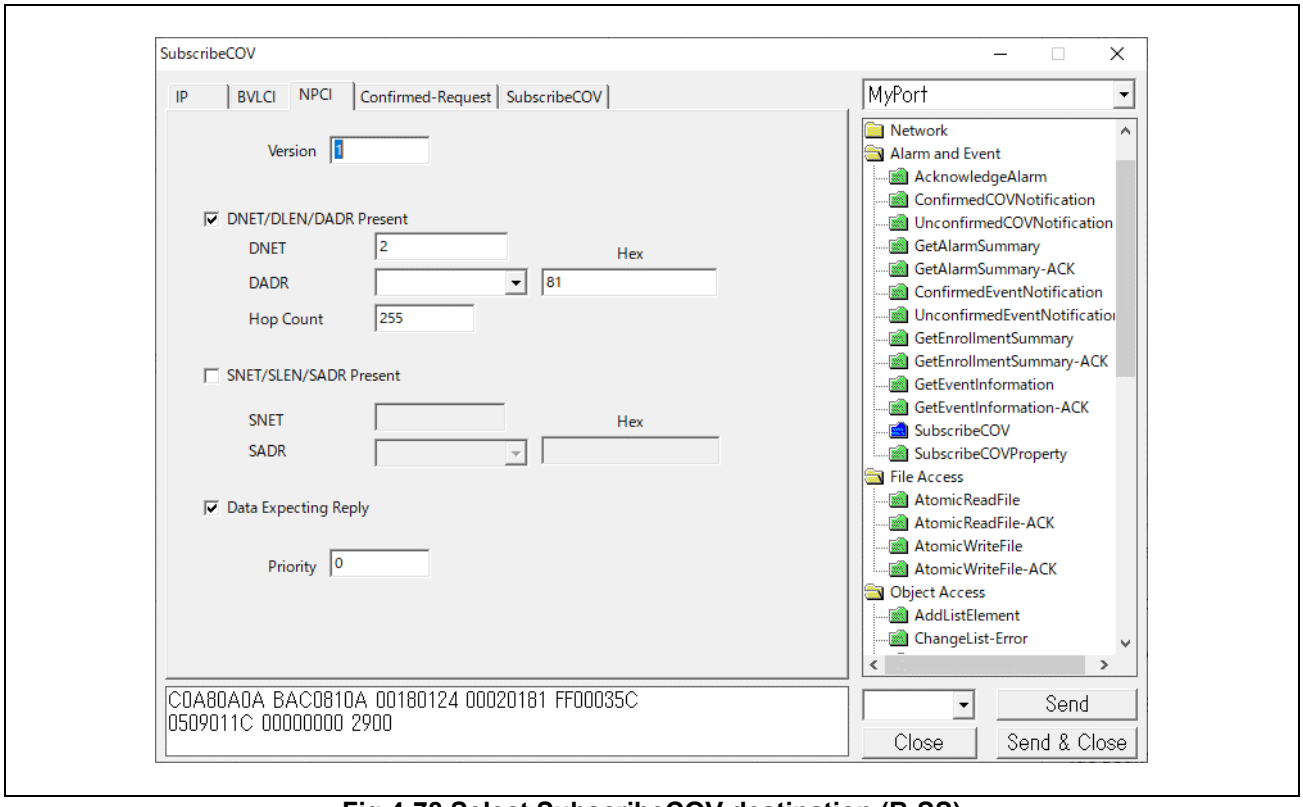

Fig.4-78 Select SubscribeCOV destination (B-SS)

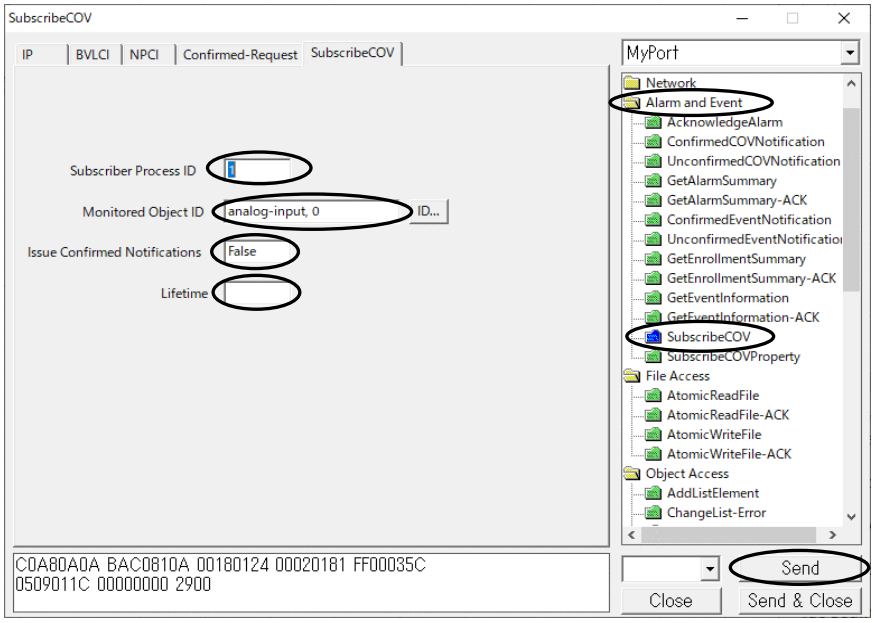

Fig.4-79 SubscribeCOV parameters

The following shows a request from a VTS client to detect changes in the property value of an analog-input,0 object by the SubscribeCOV service and a Simple-Ack response from the B-SS slave. The present-value and status-flags property values are notified by the UnconfirmedCOVNotification service from the B-SS slave that detected the air speed change. Time remaining indicates the remaining time of the subscribe period, but it returns 0.00.00 because an indefinite period is requested.

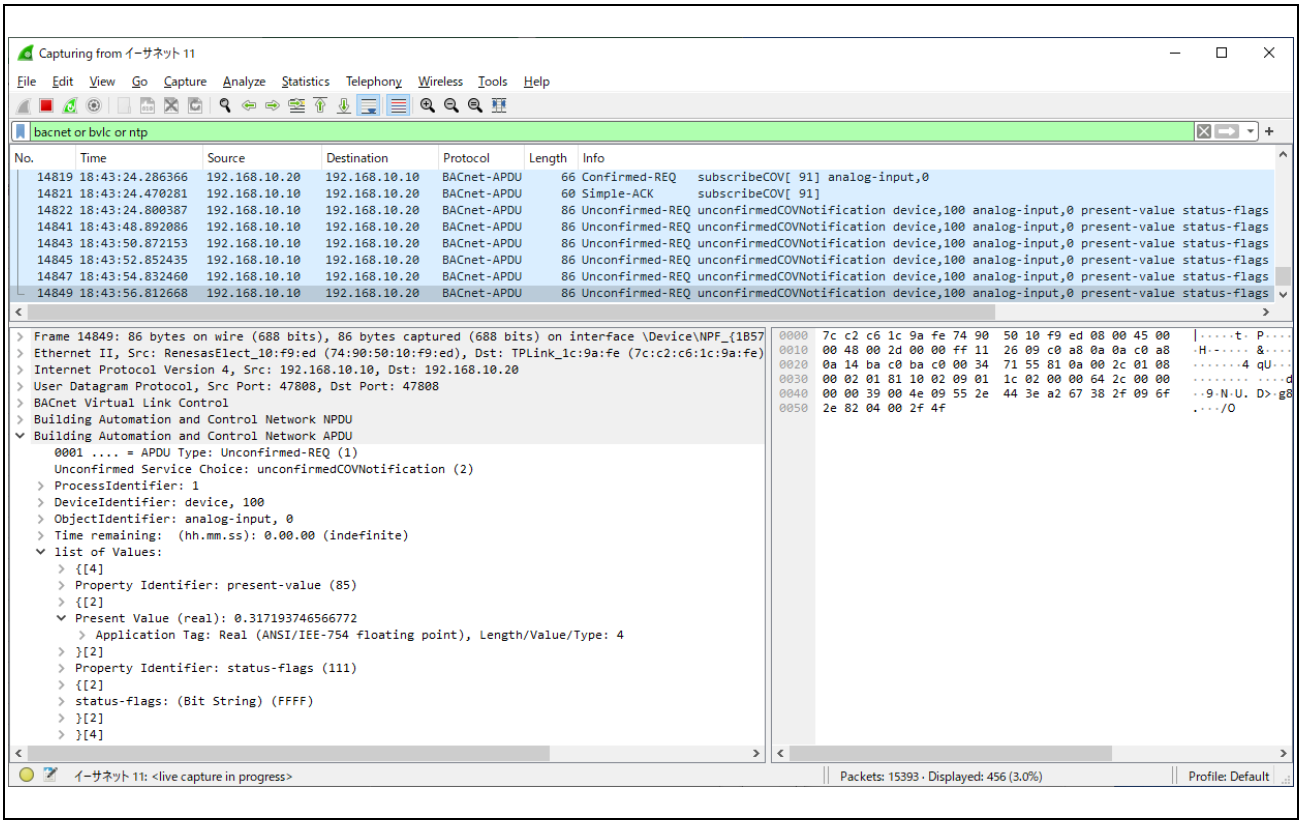

Fig.4-80 Capture SubscribeCOV and UnconfirmedCOVNotification

The following shows a SubscribeCOV service request from a VTS client with Issue Confirmed Notifications set to True and a Simple-Ack response from the B-SS slave. ConfirmedCOVNortification service notification is sent from the B-SS slave that detected the air velocity change, and the COV client responds with a Simple-Ack.

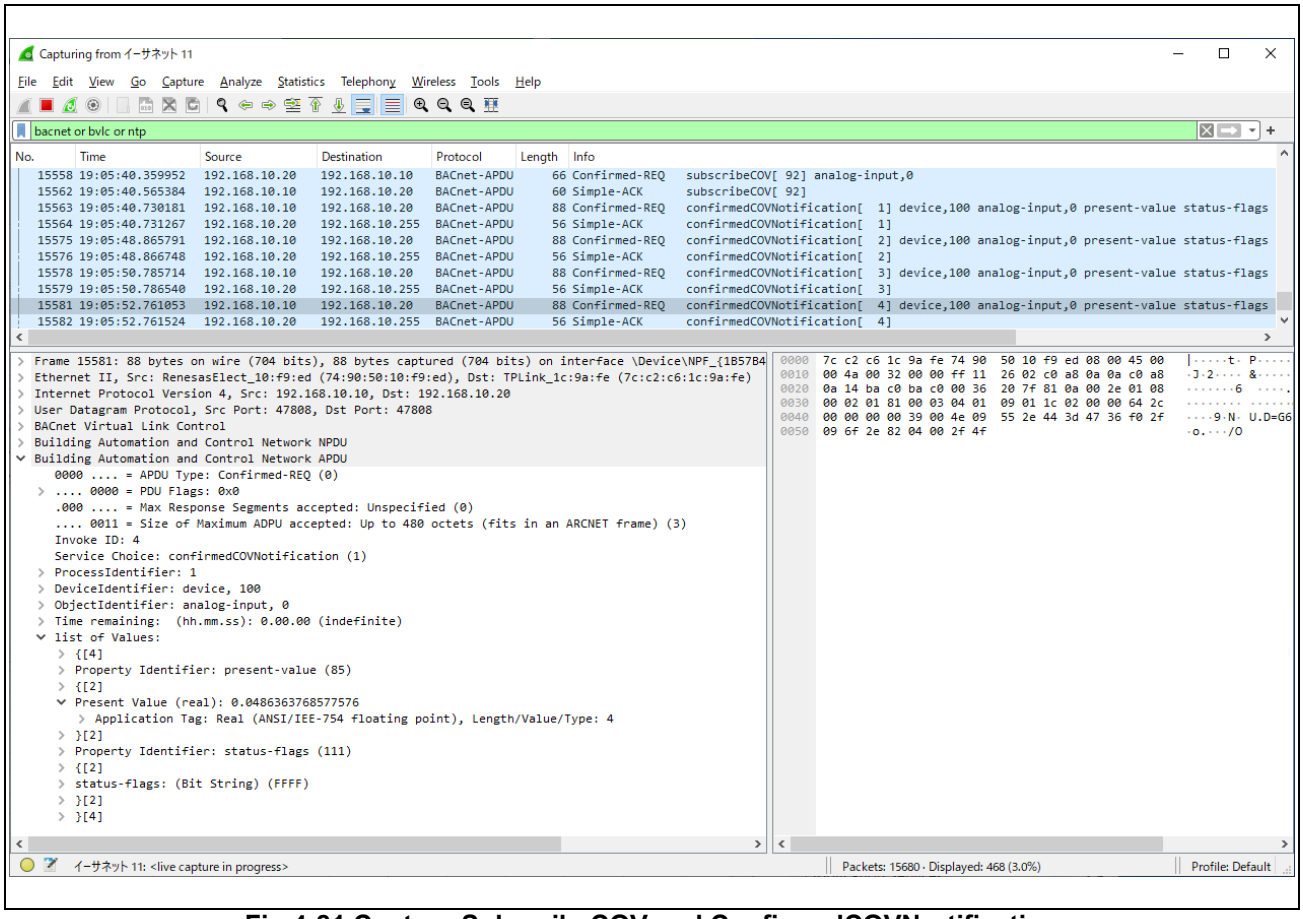

Fig.4-81 Capture SubscribeCOV and ConfirmedCOVNortification

4.5.9 ReinitializeDevice

Note) The ReinitializeDevice service resets the target device. When the debugger is connected, push the S3 RESET switch (red) on the RSK board to disconnect the debugger connection. If this service is executed while the debugger is still connected, Ethernet communication will not be established after the B-SS reboots.

Restrictions) The following State parameters of ReinitializeDevice are not yet supported by the B-BC sample software.

STARTBACKUP

ENDBACKUP

STARTRESTORE

ENDRESTORE

ABORTRESTORE

Click Remote Device Management > ReinitializeDevice from the service tree displayed in the previously used dialog. Select Cold Start or Warm Start for State. Other options are selectable, but with the exception of Activate Changes, the BACnet stack is not yet supported. (However, Activate Changes cannot be selected from the VTS.)

Enter “**filister**” in Password and click Send.

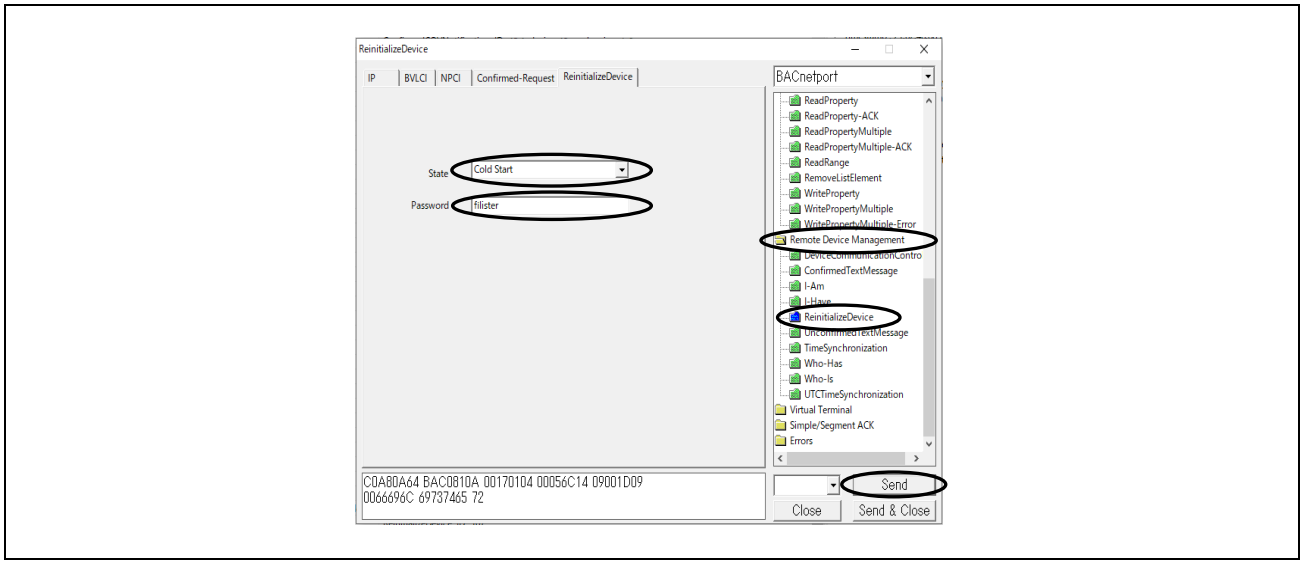

Fig.4-82 ReinitializeDevice parameters

The following shows a VTS client requesting the ReinitializeDevice service and a Simple-Ack response from the B-BC or B-SS. The B-BC or B-SS locally broadcasts I-Am services when it reboots.

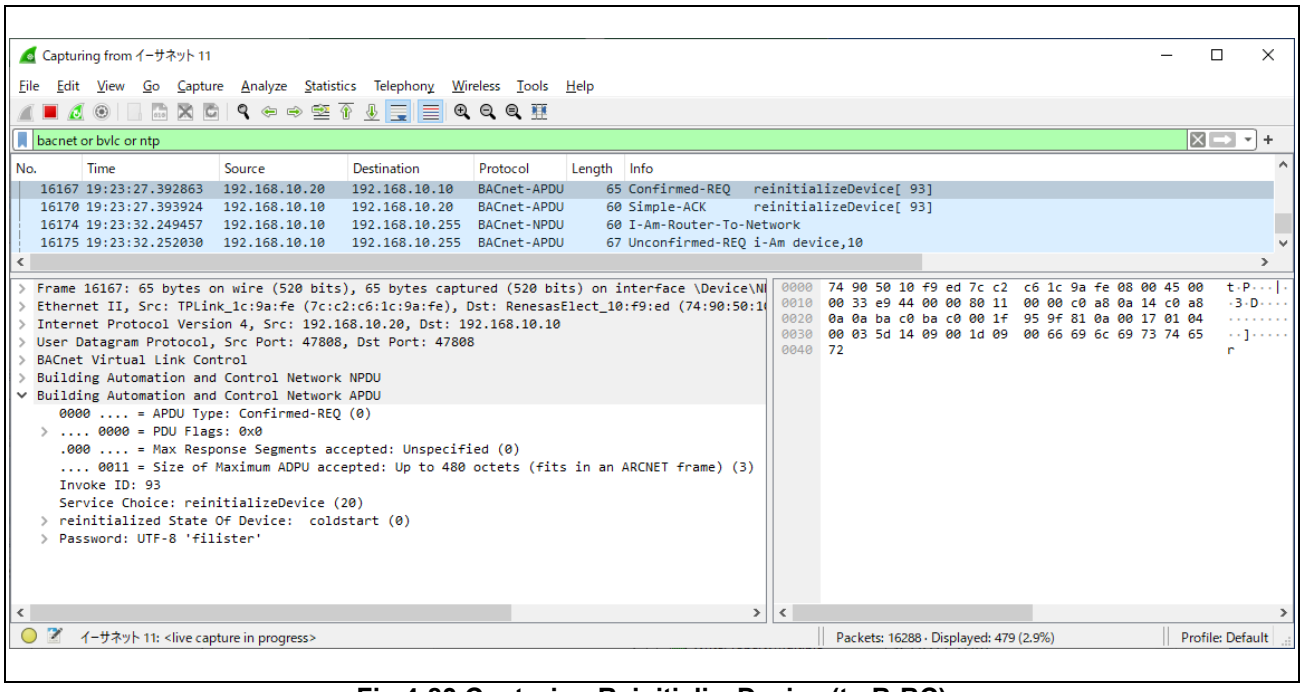

Fig.4-83 Capturing ReinitializeDevice (to B-BC)

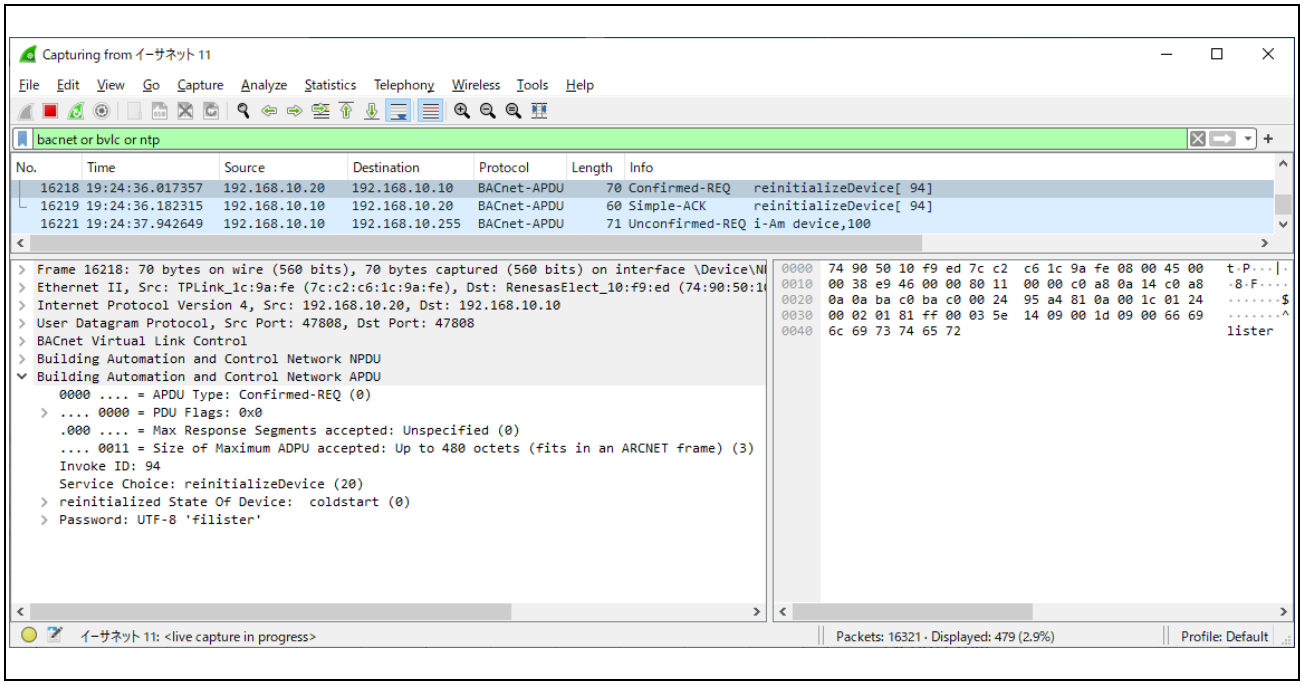

Fig.4-84 Capturing ReinitializeDevice (to B-SS)

4.5.10 DeviceCommunicationControl

Click Remote Device Management > DeviceCommunicationControl from the service tree displayed in the previously used dialog.

- In Time duration, enter the communication halt period in minutes.
- Note: If Disable is selected, it requests to stop communication, but does not accept BACnet Protocol Revision 20 or later. The server ignores the Disable request and sends an error PDU with ErrorClass = SERVICES and ErrorCode = SERVICE_REQUEST_DENIED.
- Selecting Enable requests the release of communication halt. Time duration is ignored.
- Selecting Disable Initiation will stop notifications from the server, except for I-Am service. Ack responses for service requests from clients are not stopped.

Enter “**filister**” in Password and click Send. In the example, Disable Initiation for an infinite period is selected.

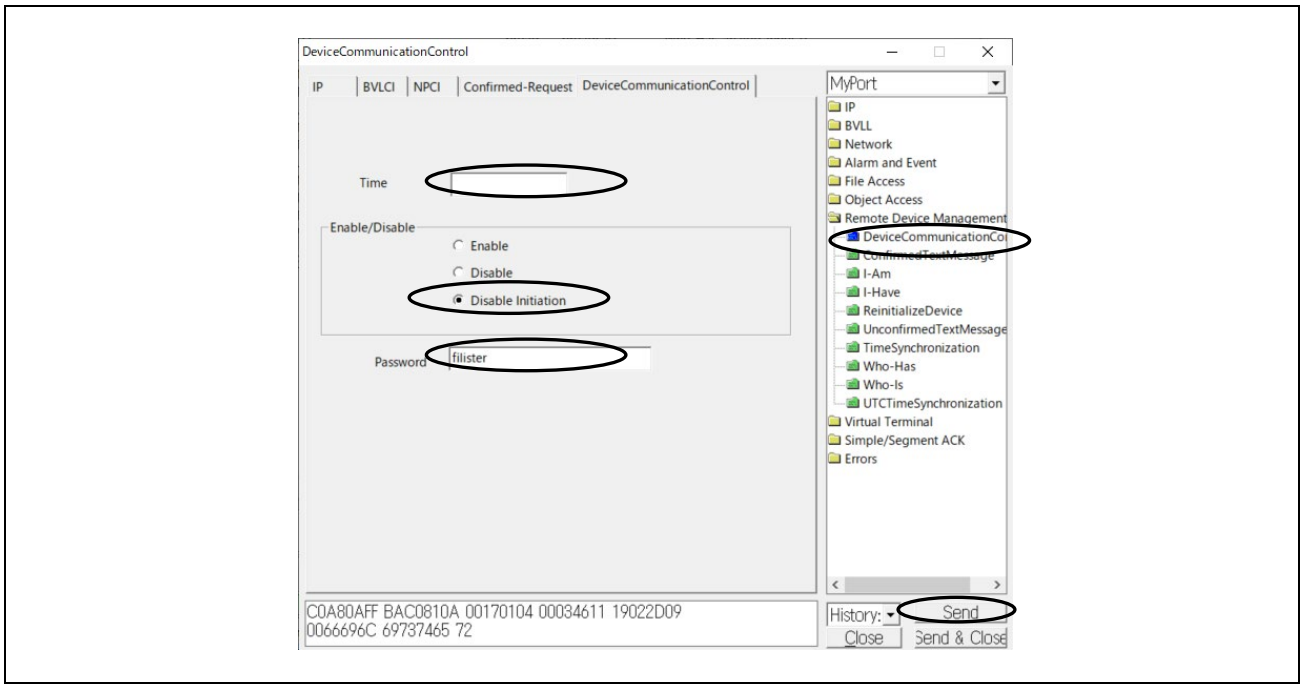

Fig. 4-85 DeviceCommunicationControl parameters(Disable Initiation)

The following shows a request from a VTS client for the DeviceCommunicationControl service and notification service from B-SS slave is stopped

(No.565) notify the UnconfirmedCOVNotification service.
(No.566) request Disable initiation.
(No.574) After Simple-Ack response, UnconfirmedCOVNotification service notification has stopped.
(No.623) request Who-Has service but not returned any I-Have service response.
(No.650) request Who-Is service
(No.653) returned I-Am service response.
(No.683) request Enable
(No.685) resume UnconfirmedCOVNotification service notification.

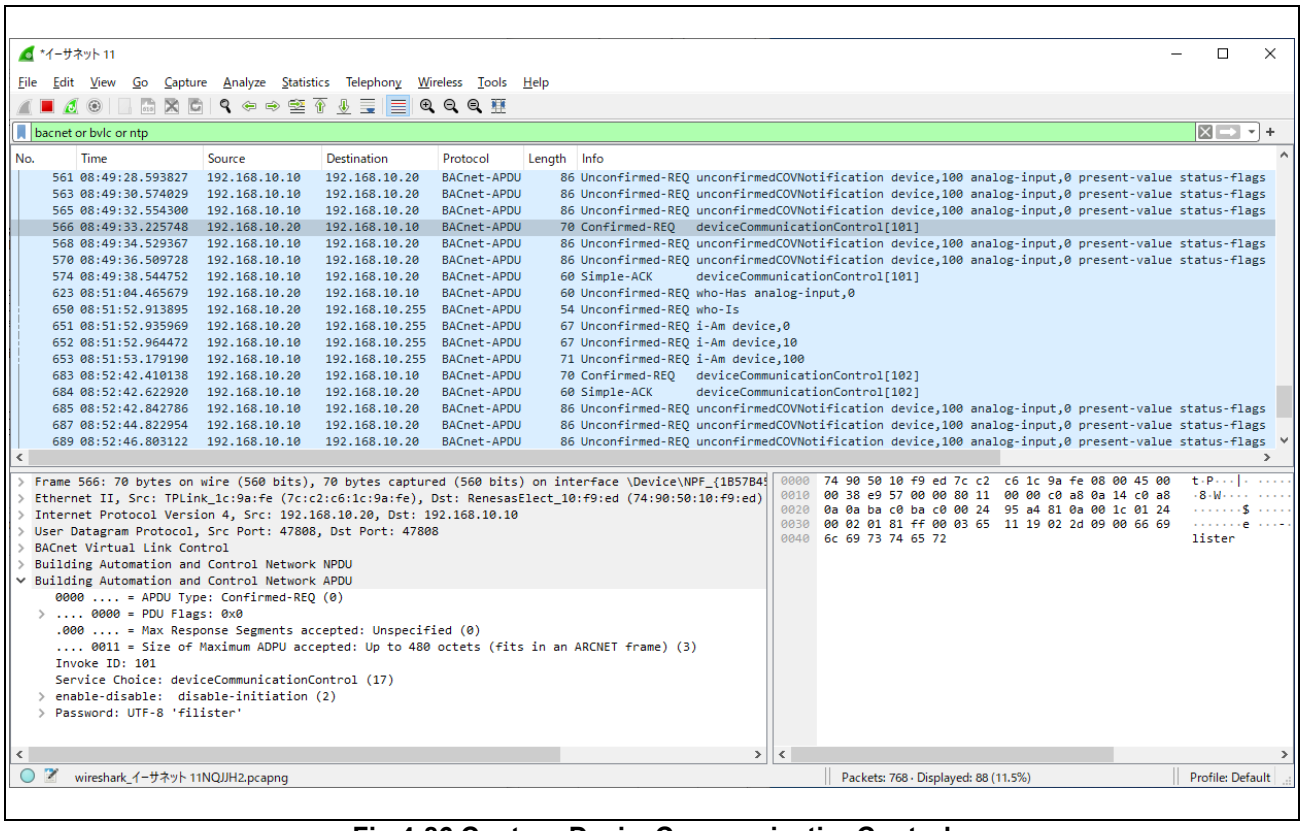

Fig.4-86 Capture DeviceCommunicationControl

4.5.11 AtomicReadFile

Many BACnet devices contain configuration data set by vendor-specific configuration tools.

The AtomicReadFile service reads configuration data inside B-BC. The client saves the read data as a backup file. Also, the saved backup file is transferred to B-BC by the AtomicWriteFile service, and B-BC restores the received file data as internal configuration data. File contents and format are vendor-specific.

In the B-BC sample software, the configuration data (variable name: FlashData) is defined by the following structure (FLASH_DATA_STRUCT) aligned with 4 bytes.

```
#pragma pack(4)
typedef struct
{
    uint32_t CheckSum;
    FLASH_DATA_ETHER_MAC emac;
    FLASH_DATA_BBMD_MAC bmac;
    FLASH_DATA_FD_LIFETIME fdsubscription;
    FLASH_DATA_BIP_MODE mode;
    FLASH_DATA_BIP_MAC mac_bip;
    FLASH_DATA_MSTP_MAC mac_mstp;
    FLASH_DATA_NW_NUMBER mstp;
    FLASH_DATA_NAME devn;
    FLASH_DATA_INSTANCE devi;
    FLASH_DATA_UTC utc;
    FLASH_DATA_OOS obj[OBJ_MAX_OOS];
}FLASH_DATA_STRUCT;
#pragma pack()
```

Fig.4-87 FLASH_DATA_STRUCT FlashData

FlashData is allocated on the System RAM but is updated during operation and stored in the QSPI flash memory by the xSPI0 driver. Upon reset, the configuration data is read from the QSPI flash memory and extracted into FlashData on the system RAM; see below for details on FlashData.

Fig.3-5 Memory layout
5.3 Initial Configuration Command
user\renesas\application\configurable_property.c

user\renesas\application\configurable_property.h

From the service tree in VTS, click File Access > AtomicReadFile.

- File ID is the File object type and instance number.

Stream Access

- Start Position is the number of octets to start reading from the beginning of the file.
  0 means the beginning of the file.
- Octet Count is the number of octets to be read from the file, starting from Start Position.
  As shown in the example below, if a larger value is set, the actual file size is read.

Record Access

- Record-oriented file access is not supported.

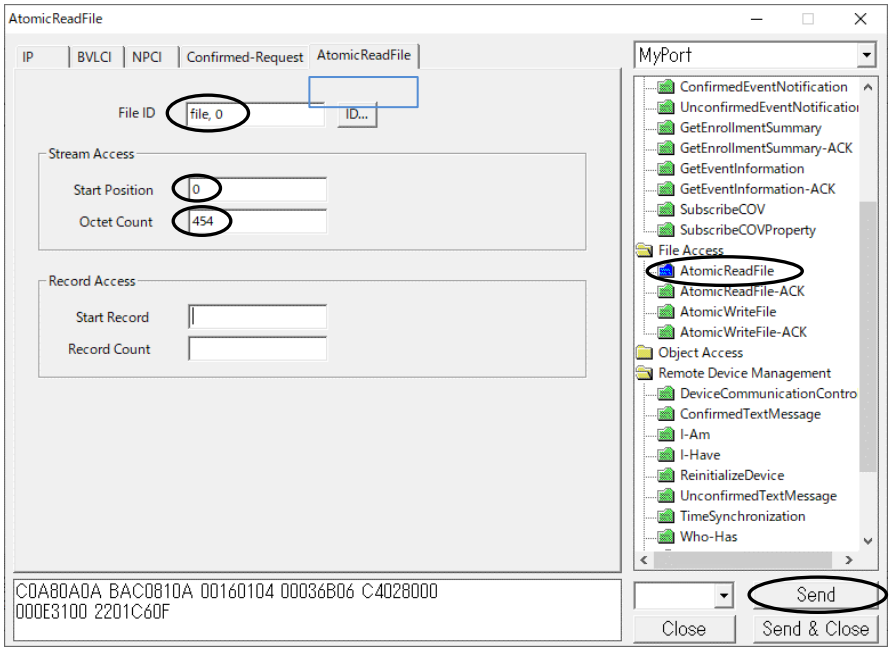

Fig.4-88 AtomicReadFile parameters

The following capture shows a VTS client requesting AtomicReadFile service from B-BC and B-BC responding with an Ack containing file data.

Length: 148 octets, End Of File: TRUE is indicated.

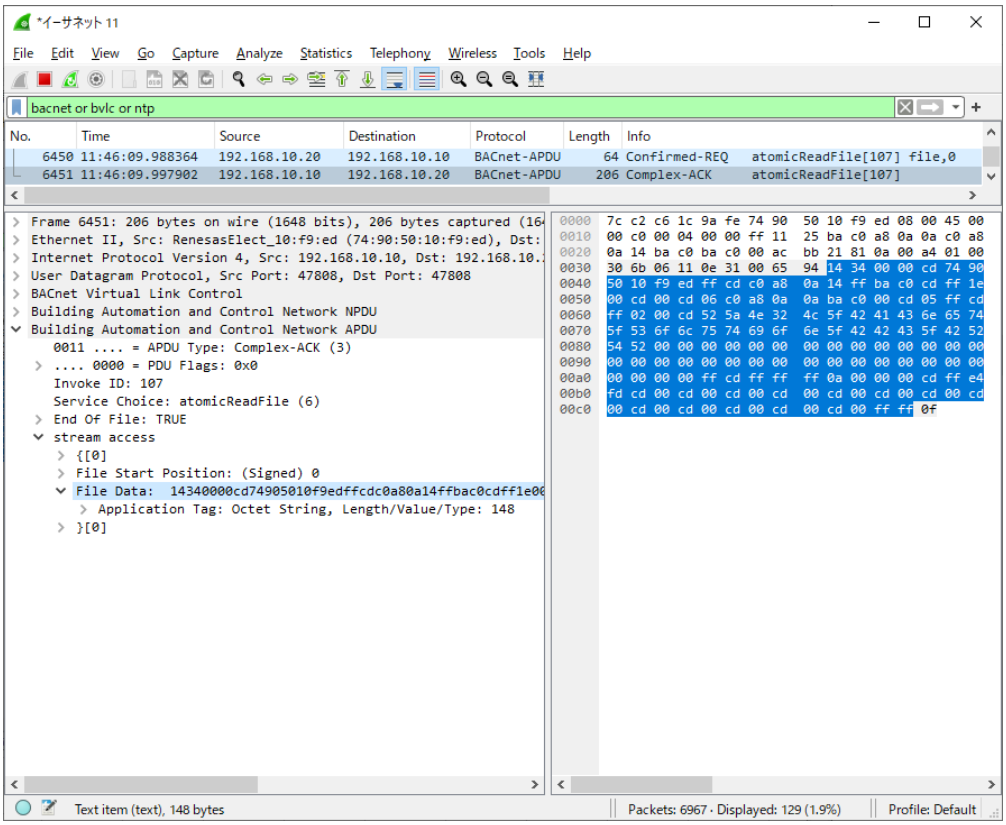

Fig.4-89 Capturing AtomicReadFile

4.5.12 AtomicWriteFile

The AtomicWriteFile service overwrites configuration data stored in B-BC. The client transfers the previously read backup file to B-BC by AtomicWriteFile service, and B-BC restores the received file data by overwriting it with the internal configuration data. File contents and format are vendor-specific. Refer to section 4.5.11

From the service tree in VTS, click File Access > AtomicWriteFile.

- File ID is the File object type and instance number.

Stream Access

- Start Position is the number of octets to start reading from the beginning of the file.
  0 means the beginning of the file.
  If Start Position is set to -1, it indicates an operation to be added from the end of the current file.
- Data consists of OCTET STRINGs that are written to a file.
  Set up a Hex data stream such as 14340000cd74....

Record Access

- Record-oriented file access is not supported.

The following shows how to make Data for the parameters above.

Select the Complex-ACK line from B-BC for the AtomicReadFile service (No. 7902 below) in Wireshark. In the lower left window, place the cursor on > File Data: and right-click, then click Copy>...as a Hex Stream in the menu that appears. Paste to ”Data” in the AtomicWriteFile parameter dialog of VTS and click Send.

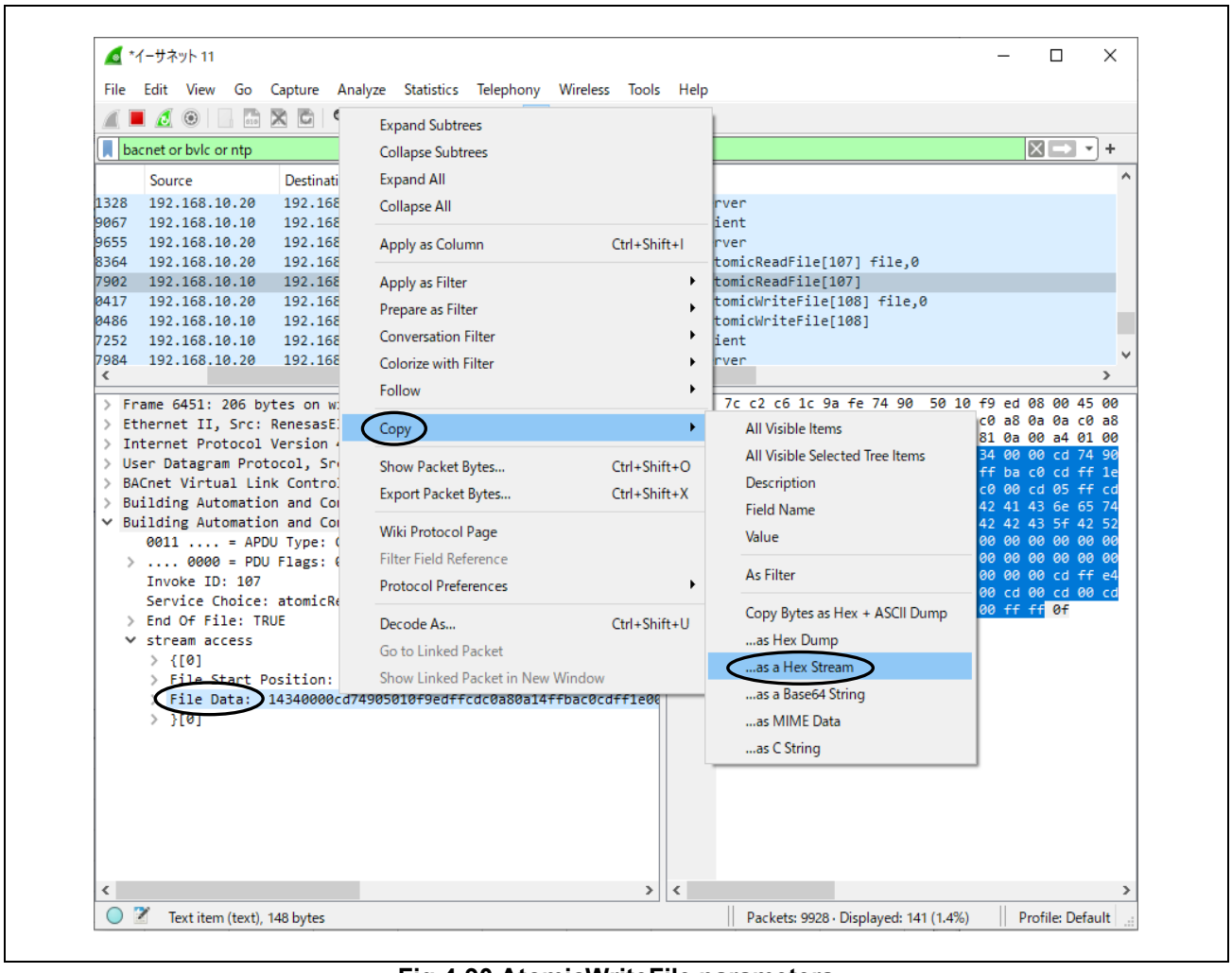

Fig.4-90 AtomicWriteFile parameters

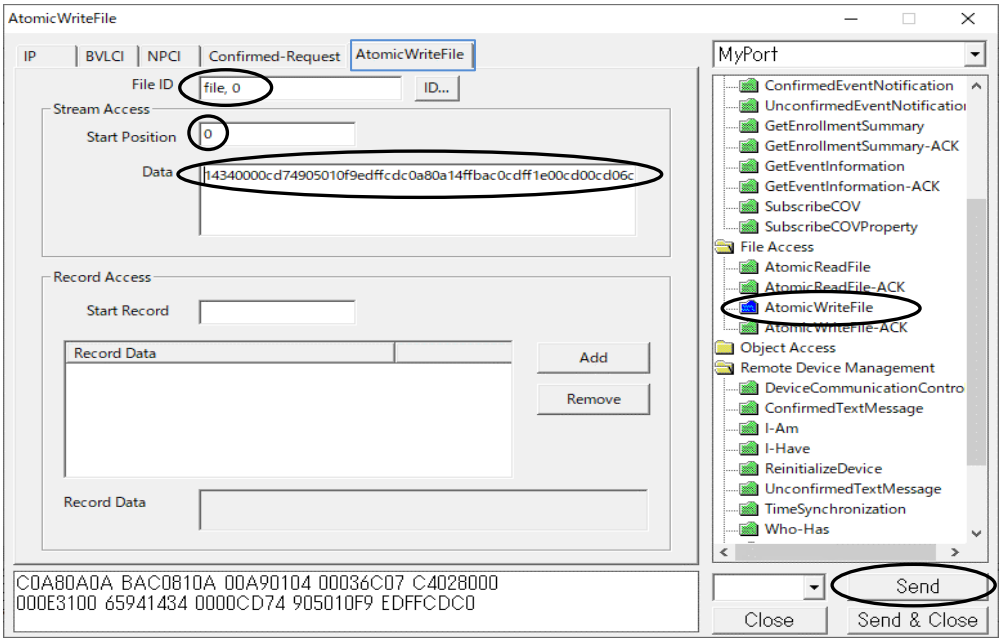

Fig.4-91 AtomicWriteFile parameters

The following capture shows a VTS client requesting AtomicWriteFile service from B-BC and receiving an Ack from B-BC.

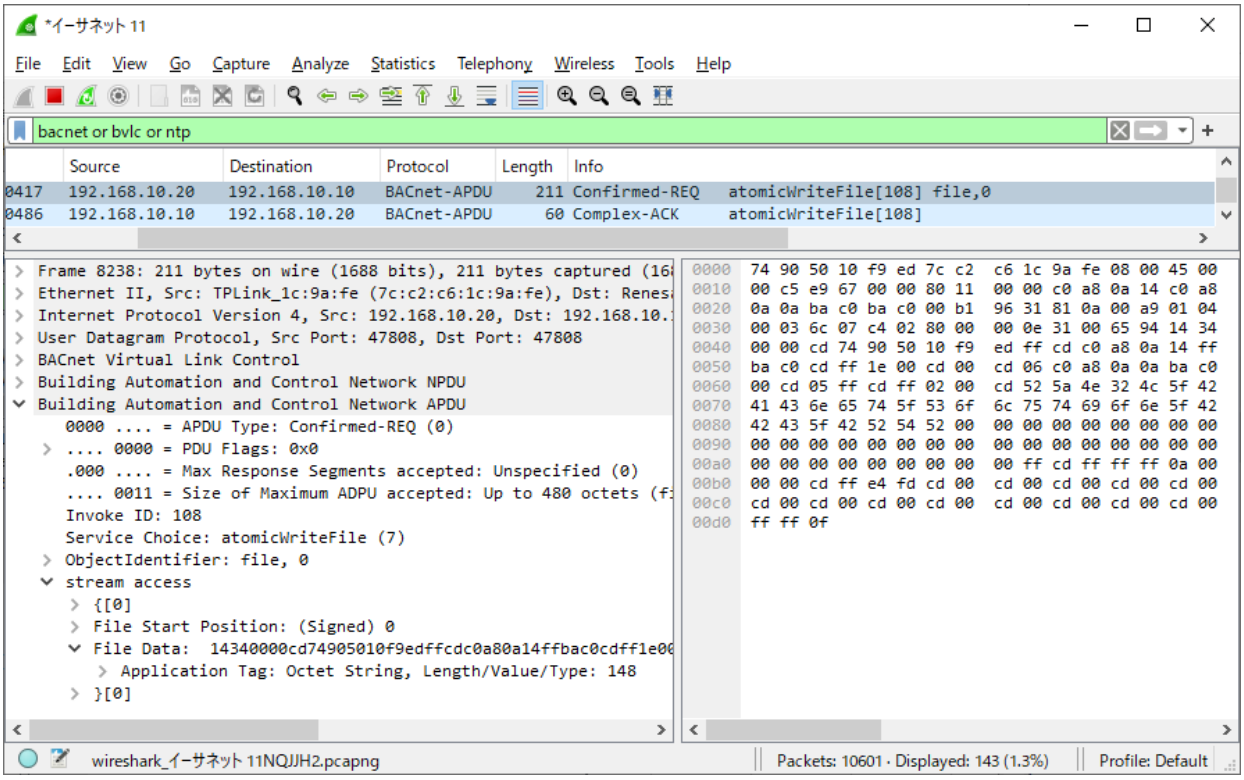

Fig.4-92 Capturing AtomicWriteFile

4.6 BACnet Communication with Yabe

The following operation can be verified from VTS, but will be explained using Yabe, which is easier to operate. Open Windows Start and click on Yabe to launch Yabe.

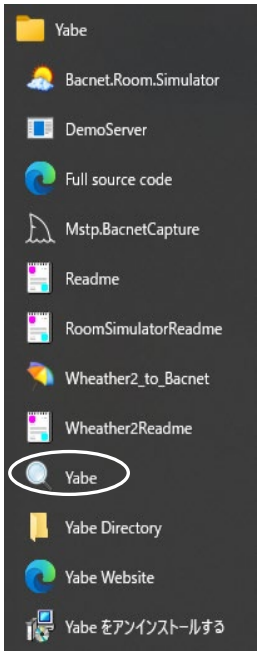

Fig.4-93 Launch Yabe

This section describes BACnet/IP communication with Yabe. After starting up Yabe, add devices.

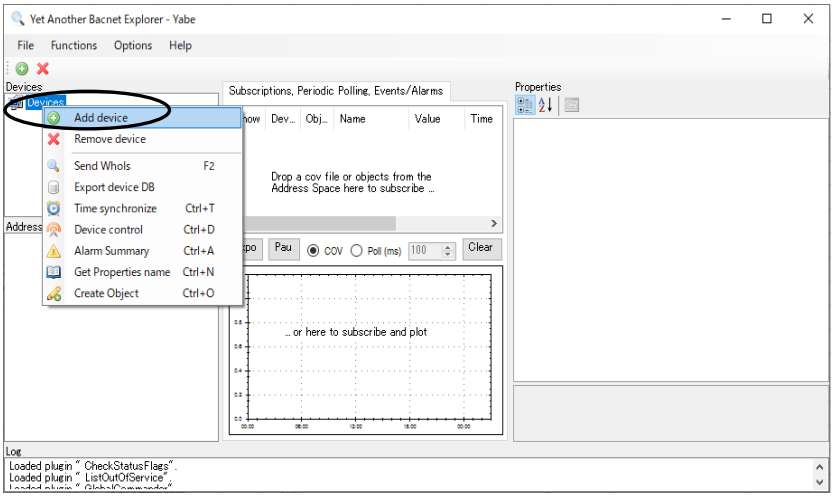

Fig.4-94 Yabe Add device(1)

Enter BAC0 (47808) for Port in the dialog displayed and select the IP address of the PC client. Click Start to initiate communication.

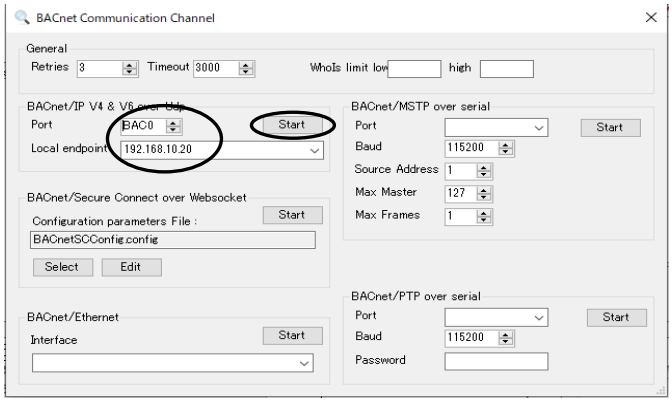

Fig.4-95 Yabe Add device(2)

Device 10 in the Devices window is B-BC, Device 100 is B-SS, and Device 2 is Yabe. Here, the YabeDeviceID is set to 2 from Yabe's Options > Settings window. (see chapter 4.6.1)

Click on Device 10.

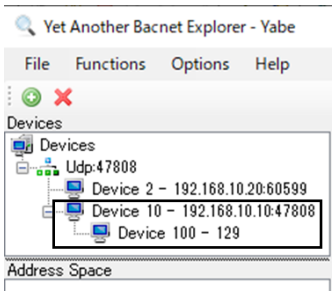

Fig.4-96 Yabe Add device(3)

Object list for Device 10 appears in the Address Space window.

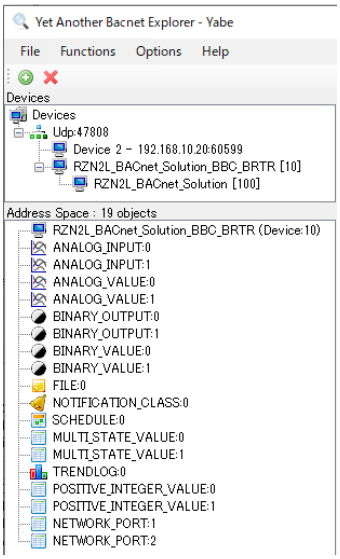

Fig.4-97 Yabe Add device(4)

Click on Device 100 in the Devices window. The Address Space window switches to a display of the object list for Device 100.

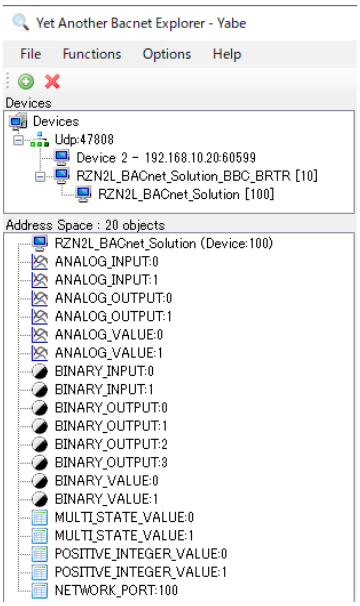

Fig.4-98 Yabe Add device(5)

4.6.1 TimeSynchronization / UTCTimeSynchronization

This section describes the procedure for setting up TimeSynchronization / UTCTimeSynchronization from Yabe, as described in section 4.5.3 from VTS.

For time synchronization service, select TimeSynchronization or UTCTimeSynchronization through Options.

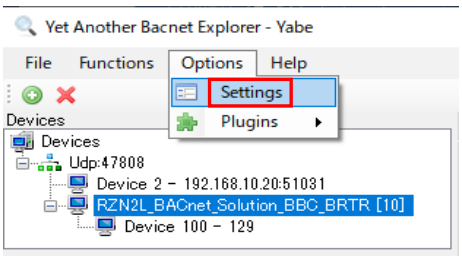

Fig.4-99 Selecting a time synchronization service(1)

Setting TimeSynchronize_UTC to False in the Settings screen below selects TimeSynchronization, while setting it to True selects UTCTimeSynchronization.

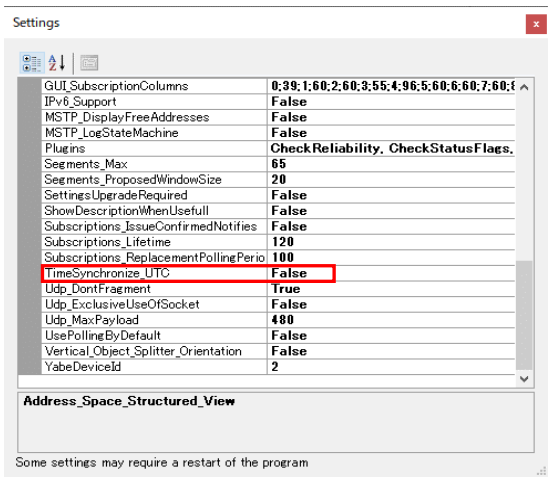

Fig.4-100 Selecting a time synchronization service(2)

Select B-BC in the Devices window and right-click to select Time_synchronize from the list that appears.

Click OK on the pop-up window.

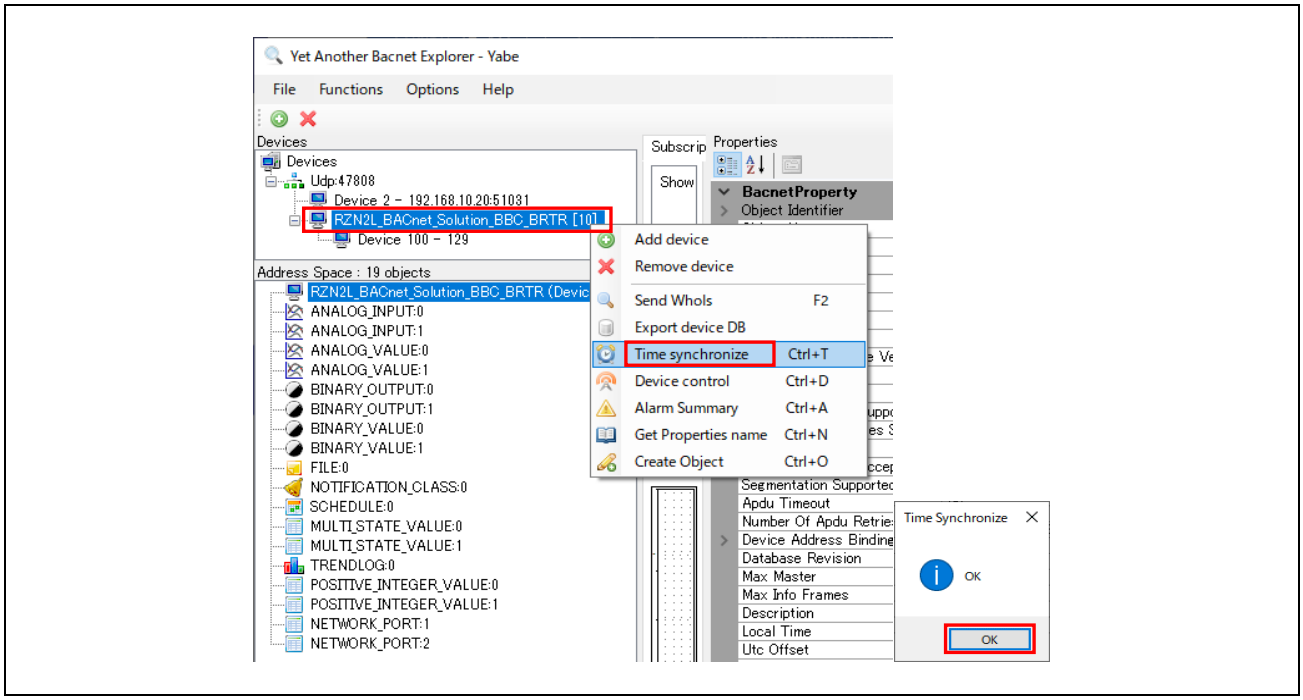

Fig.4-101 Selecting a time synchronization service(2)

Similarly, Time_synchronize should be performed for B-SS.

4.6.2 Controlling B-SS from B-BC

B-BC devices support DS-RP-A and DS-WP-A profiles and can Initiate the ReadProperty and WriteProperty services on their own. This chapter describes controlling an external B-SS device from B-BC with DS-RP-A and DS-WP-A.

The TrendLog and Schedule objects handled in this chapter use timestamps. Please execute TimeSynchronization or UTCTimeSynchronization service request to B-BC and B-SS in advance.(Chapter 4.6.1)

4.6.2.1 Trending & ReadRange

B-BC requests ReadProperty service (DS-RP-A) to B-SS to collect the PresentValue property of the AnalogInput,0 object of B-SS input from the sensor as a log record in the LogBuffer property of the TrendLog,0 object of B-BC.

Click RZN2L_BACnet_Solution_BBC_BRTR [10] in the Devices window.

Click TREND LOG 0 (Trendlog:0) in the Address Space window.

Change the following properties in the Properties window

- Set today's date in “Start Time”.
- Set tomorrow's date in “Stop Time”.
- Change the Instance under DeviceID to (100), which is the instance number of the B-SS device.

  If the instance number is 10, the PresentValue of B-BC's own AnalogInput,0 object is the logging target. Note that it is necessary to set OutOfService=True for the AnalogInput,0 object before changing the PresentValue of the B-BC's own AnalogInput,0 object.
- Log Interval is displayed in units of 10[ms], but the actual setting unit is in seconds.

  If 1~99 is input, it will be rounded to 100 (100 x 10[ms]=1000[ms]).

  When setting 1 second, set 100 (100 x 10[ms]=1000[ms]=1[s]).

  The default is 90000 (90000 x 10[ms] = 900000[ms] = 900[s] = 15[min]).

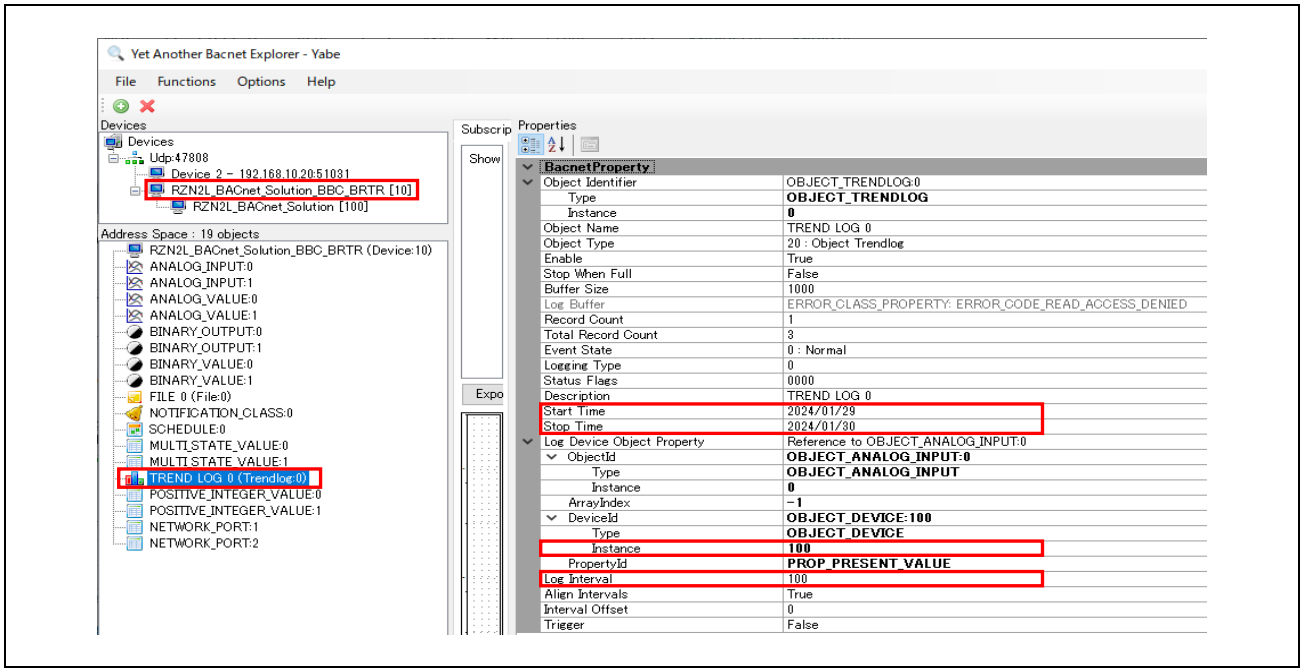

Fig.4-102 Trend Log object(1)

To refresh TREND LOG 0 (Trendlog:0) in the Address Space window, click on another object and then click on TREND LOG 0 (Trendlog:0) again. Confirm that the “Record Count” property value has increased enough.

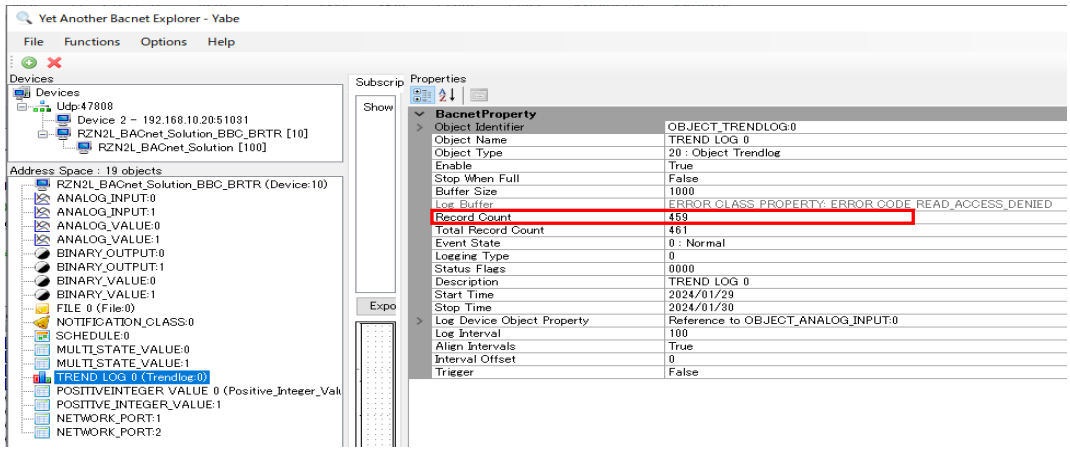

Fig.4-103 Trend Log object(2)

Right click on TREND LOG 0 (Trendlog:0) and select “Show TrendLog”. Then Yabe makes a ReadRange service request to B-BC.

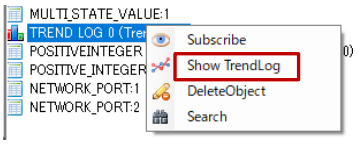

Fig.4-104 Trend Log object(3)

Logs with time stamps and graphs of B-SS sensor input values are displayed. Make sure that the sensor detects wind while logging is in progress.

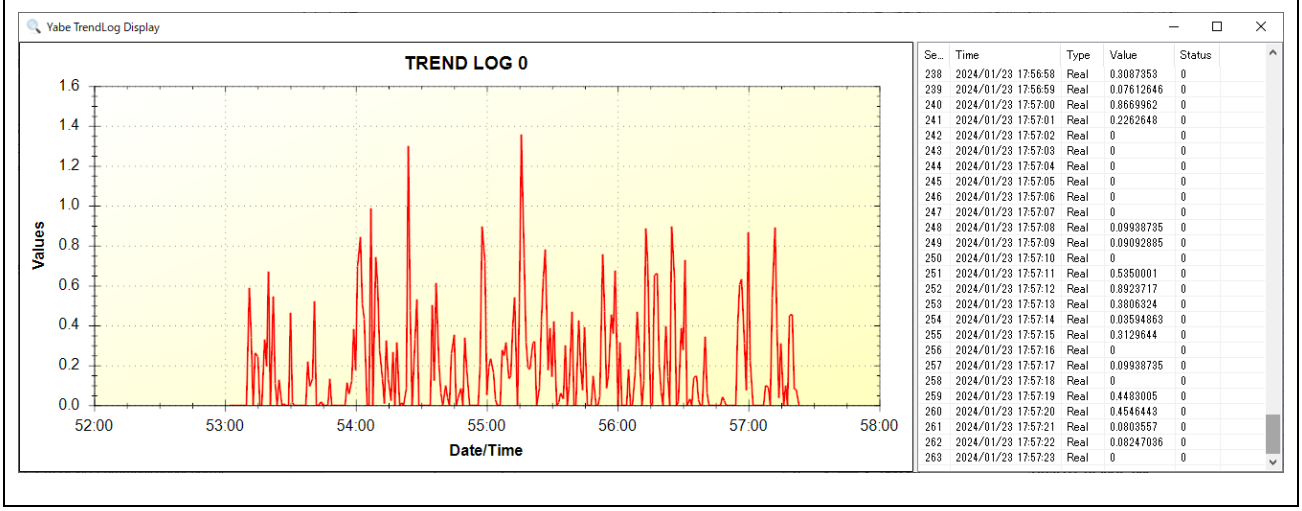

Fig.4-105 Show TrendLog

To stop logging, set the “Enable” property value to False.

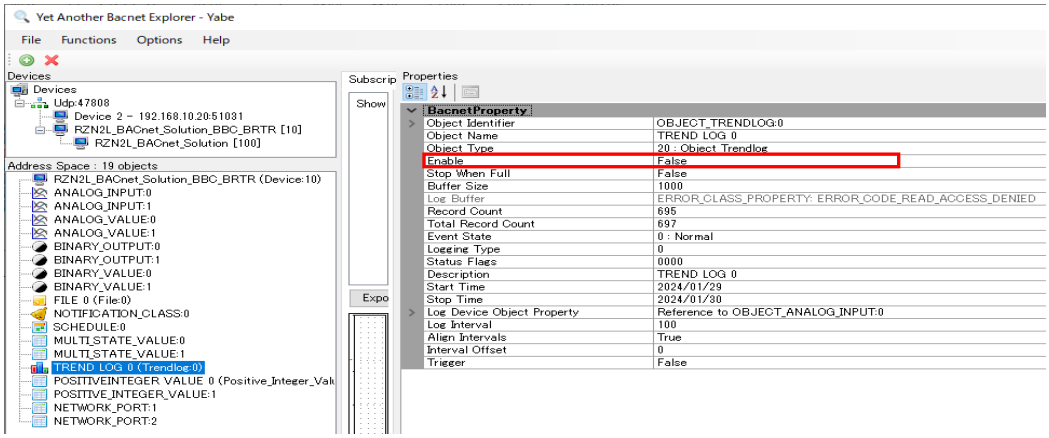

Fig.4-106 Trend Log object(4)

The following MS/TP network capture shows a ReadProperty service request (DS-RP-A) from B-BC to B-SS and its response being executed in one second cycles. The MAC address 0x05 displayed in Source and Destination indicates a B-BC master, and 0x81 indicates a B-SS slave.

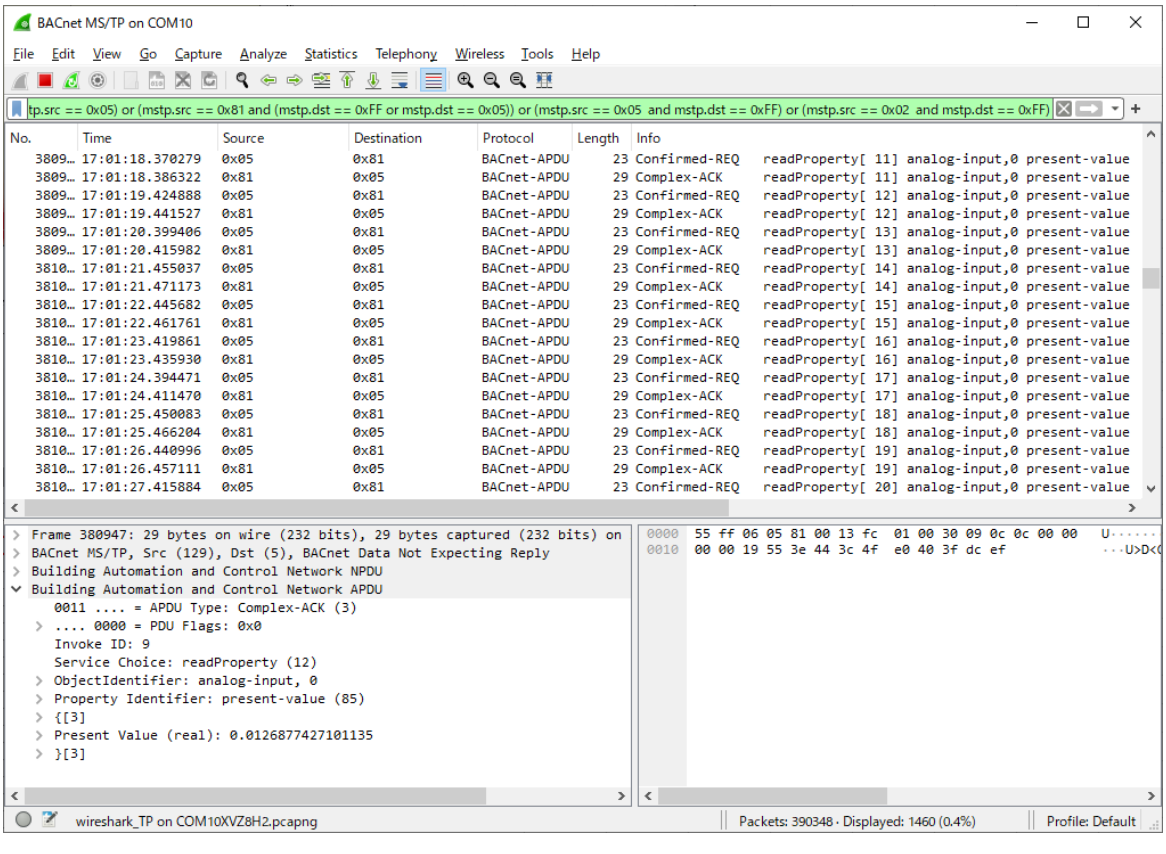

Fig.4-107 ReadProperty service request to B-SS (DS-RP-A) capture

The following capture of the BIP network shows a ReadRange service request from Yabe to B-BC and its response.

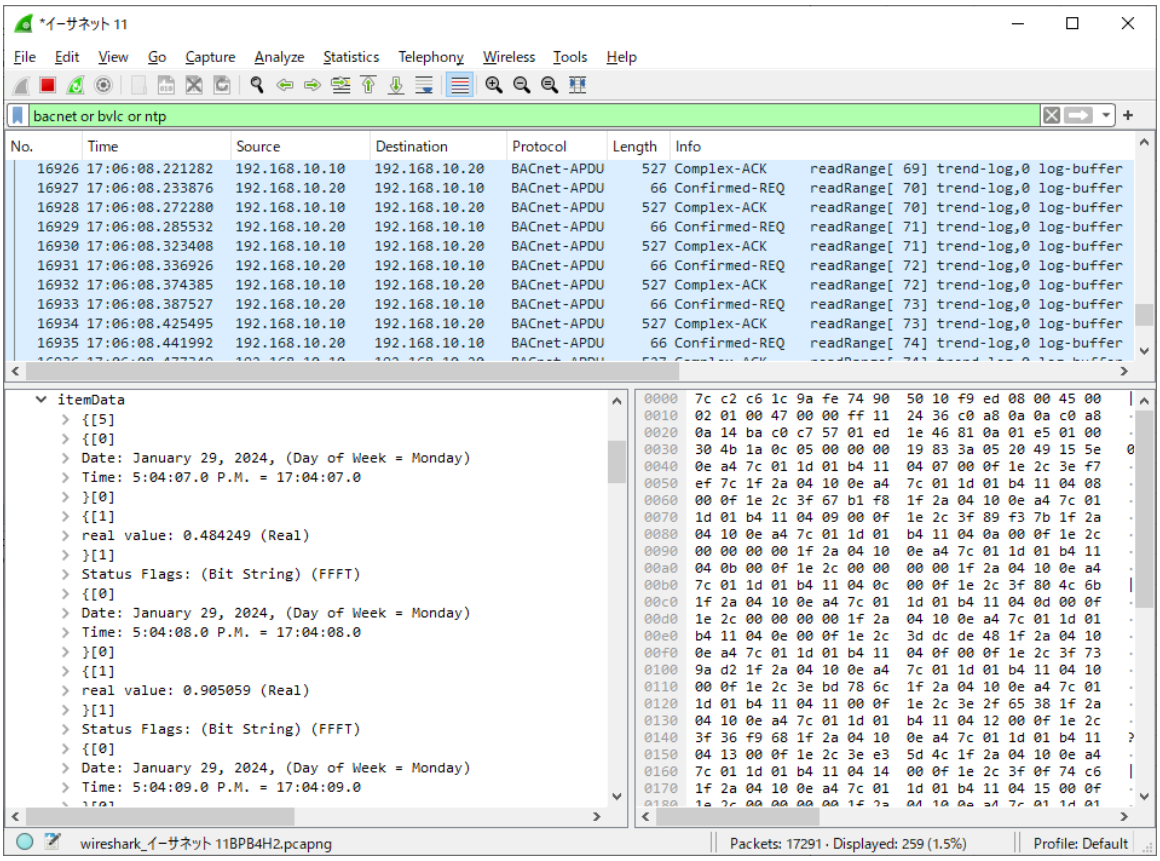

Fig.4-108 ReadRange service request capture to B-BC

4.6.2.2 Scheduling

The B-BC requests the WriteProperty service to the B-SS (DS-WP-A) for changing the PresentValue property of the BinaryOutput,0~3 objects of the B-SS assigned to the LEDs to be turned on and off according to the weekly schedule set in the Schedule,0 object of the B-BC.

Click on RZN2L_BACnet_Solution_BBC_BRTR [10] that appears in the “Devices” window of Yabe.

Right-click “SCHEDULE 0 (Schedule:0)” in the “Address Space” window and select “Show Schedule”. The “List Of Object Property References” property in the center of the “Properties” window is assigned to Device,100 BinaryOutput,0 PresentValue as the Schedule target, so there is no need to change it.

If the device instance number of the B-SS is changed to other than 100, for example, the List Of Object Property References property should be changed. Not only BinaryOutput objects, but also other output objects can be changed to Schedule target.

The following output objects of B-SS have been verified as schedule targets.

AnalogOutput, AnalogValue, BinaryOutput, BinaryValue, MultiStateValue, PositiveIntegerValue

In addition, if the device instance number is changed to 10, B-BC's own Output object can be scheduled.

The following output objects of B-BC have been verified as schedule targets.

AnalogValue, BinaryOutput, BinaryValue, MultiStateValue, PositiveIntegerValue

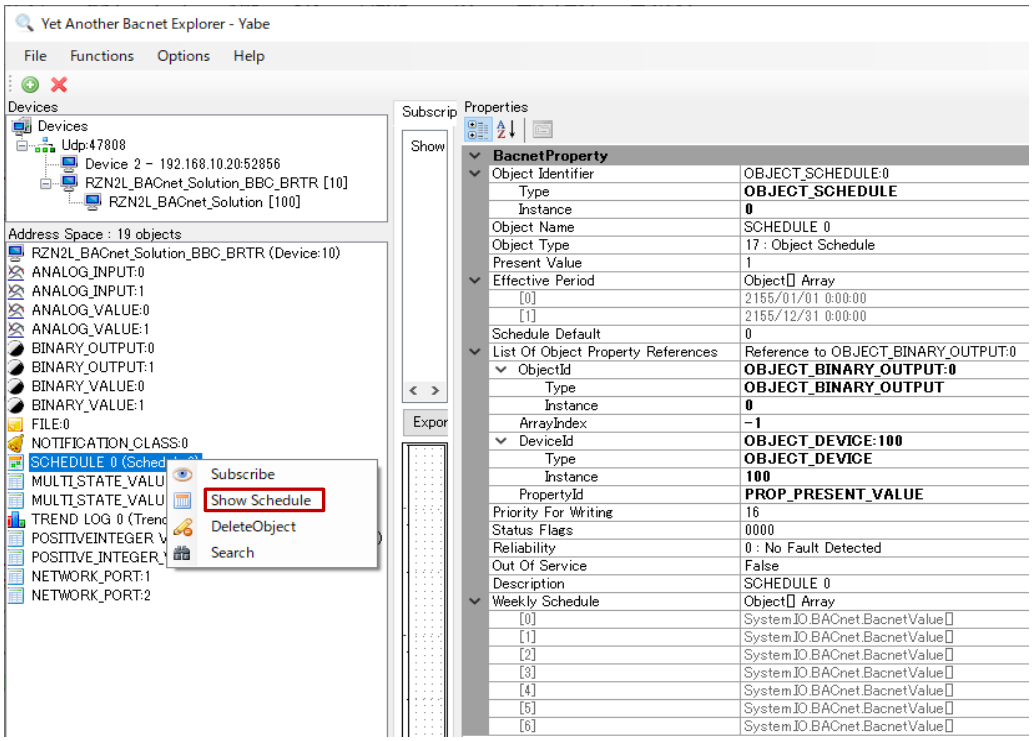

Fig.4-109 Schedule object

Select “Show Schedule” to open the “Simple Schedule Editor” and change the properties.

- Set today's date in “Validity Start Date”.
- Set tomorrow's date to “Validity End Date”.

Set the general-purpose LED0 on the B-SS board to turn on and then off.

- Right-click on today's day in “Weekly Schedules”, select “Modify” and set hh:mm:ss = 1.
- Right-click on today's day in “Weekly Schedules” and select “Add” and set hh:mm:ss = 0.

Finally, click Update & Read back to close the Simple Schedule Editor.

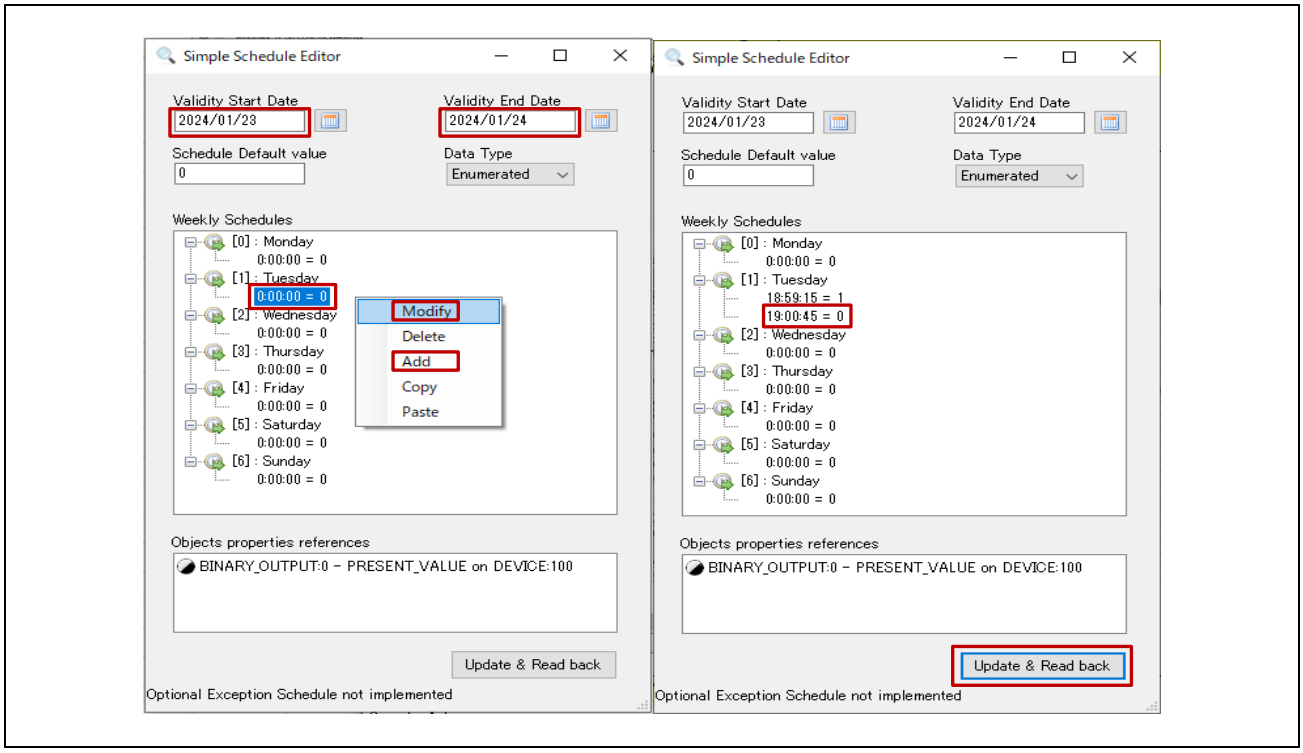

Fig.4-110 Simple Schedule Editor

The following MS/TP network capture screen shot shows a WriteProperty service request (DS-WP-A) to B-SS and its response at 18:59:15 and 19:00:45 of the Schedule time. At the same time, LED0 on the B-SS board turns on and off.

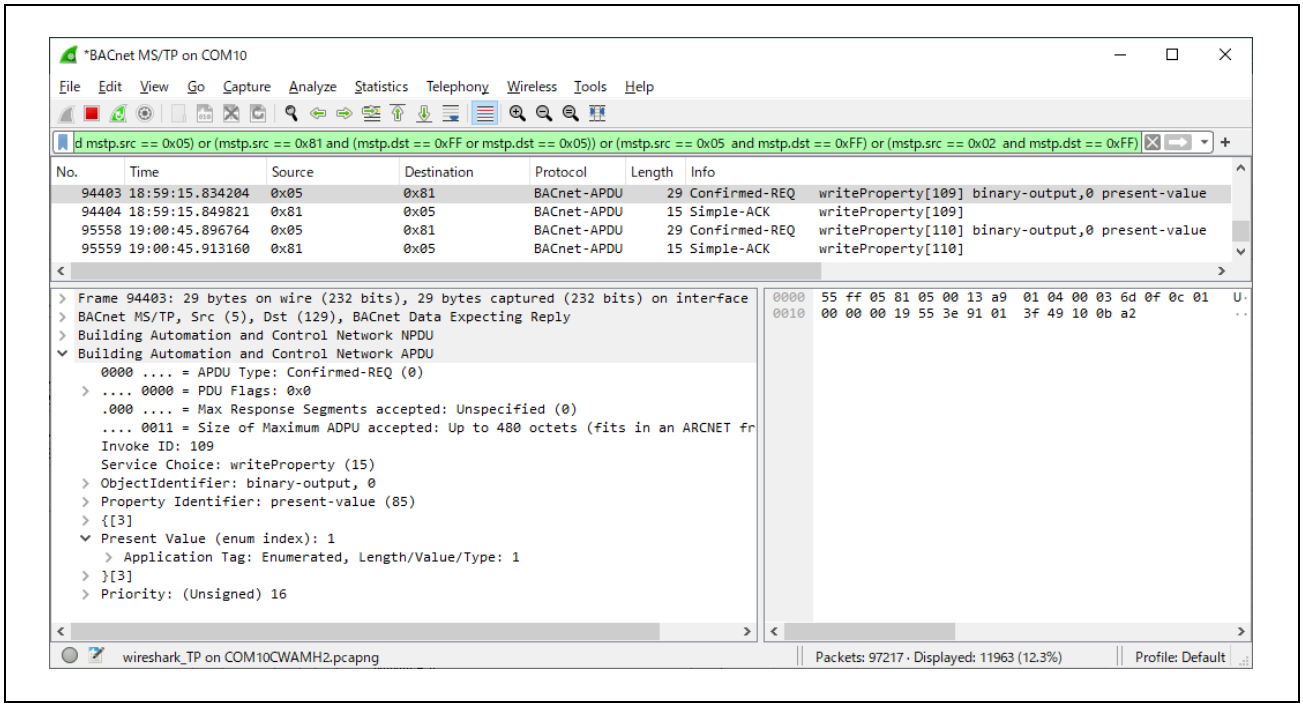

Fig.4-111 WriteProperty service request to B-SS (DS-WP-A) capture

4.6.3 EventNotification / GetEventInformation / AcknowledgeAlarm

B-BC's AnalogInput object supports the Out Of Range event algorithm. If the PresentValue property value deviates from the range specified for the Low_Limit and High_Limit properties, the ConfirmedEventNotification service or the UnconfirmedEventNotification service will be notified to the BIP client.

The BIP client requests the GetEventInformation service to retrieve all "active event states".

The BIP client also requests an AcknowledgeAlarm service request to confirm that the BIP client has acknowledged the event notification from the B-BC and returned an Ack.

Right-click “NOTIFICATION CLASS 0 (Notification_Class:0)” in the “Address Space” window and select “Show Notification”.

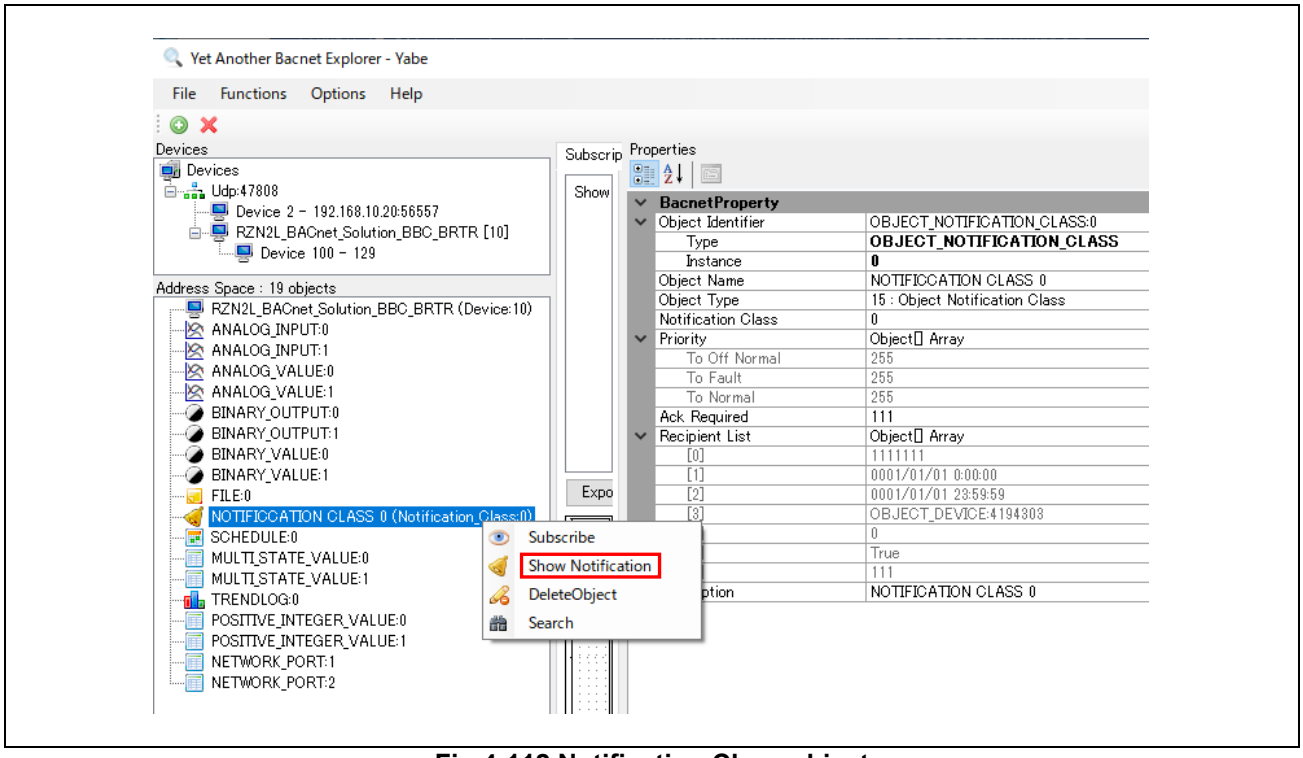

Fig.4-112 Notification Class object

Select Show Notification to open the Notification Editor and change the properties.

- Ack Required selects whether ConfirmedEventNotification is notified to the recipient (BIP client) (with ✓) or UnconfirmedEventNotification is notified (without ✓).
- Process Id is a process handle in the recipient that receives event notifications.

EventType

- To_OffNormal notifies an event when a transition occurs to a state of neither normal nor fault.
- To_Fault notifies an event when a transition occurs to the fault state.
- To_Normal notifies an event when a transition occurs to the normal state.

Validity

- The days of the week and times when Event notifications are activated.

Receiver

- Set either the instance number or IP address of the device that receives Event notifications. However, IP address cannot be selected because this B-BC sample software does not support it.

Priority

- Priority of each Event Notification. The range is 0 to 255.

The above properties are already initialized, so at least only the “Receiver” needs to be changed. The number set in the following example is the device instance number of Yabe. It is configurable from the Options menu bar in Yabe.

Finally, click “Write & Read back” to close the Notification Editor.

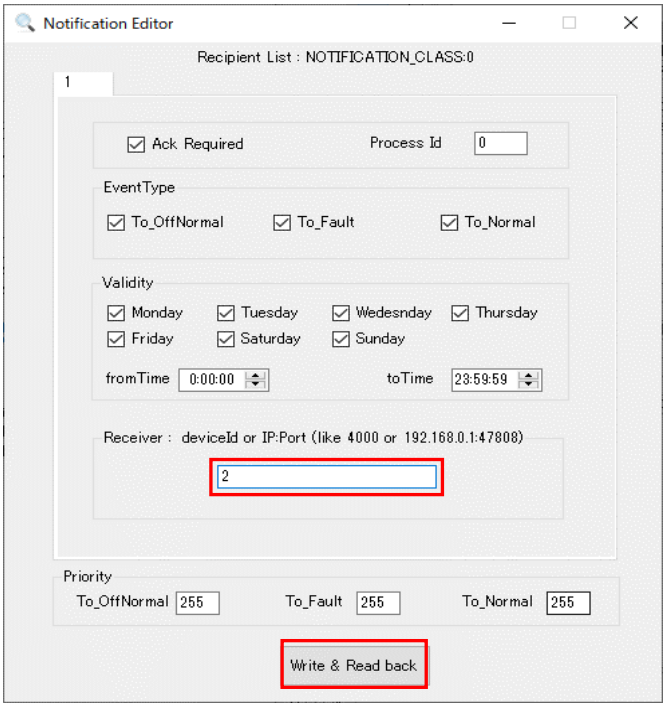

Fig.4-113 Notification Editor

Next click on ANALOG INPUT 0 (Analog_Input:0) from the Address Space tree and select True for Out Of Service in the Properties window. Setting Out Of Service to True allows the Present Value to be changed.

Then, set the Present Value to 100.1 so that it exceeds the High Limit (100).

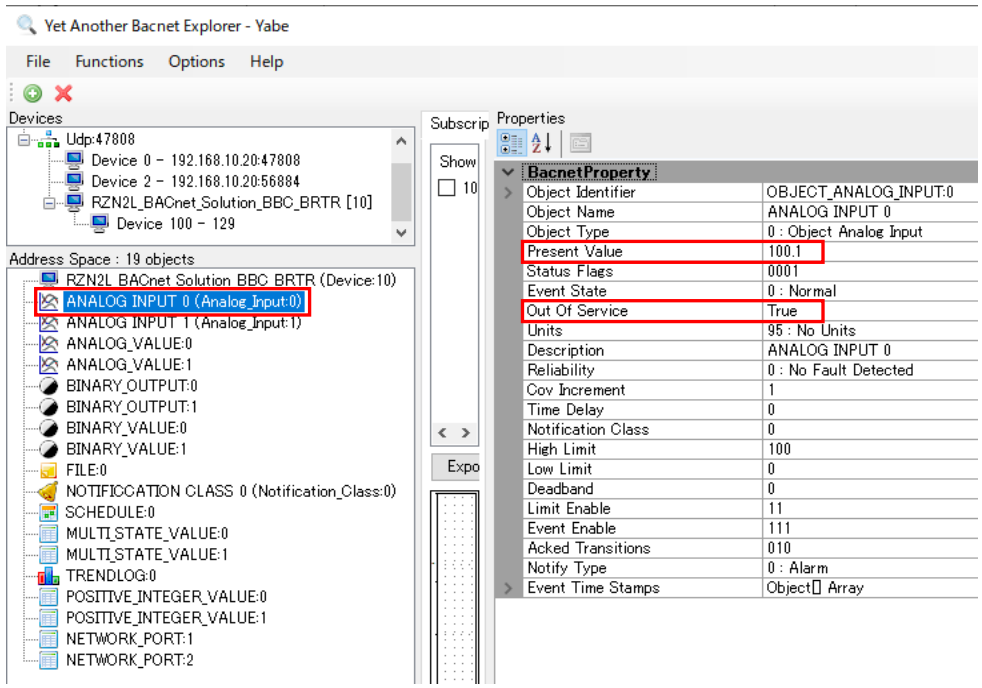

Fig.4-114 AnalogInput,0 object(1)

Next, set the Present Value to 99.9, which is less than the High Limit (100), to transit to the normal state.

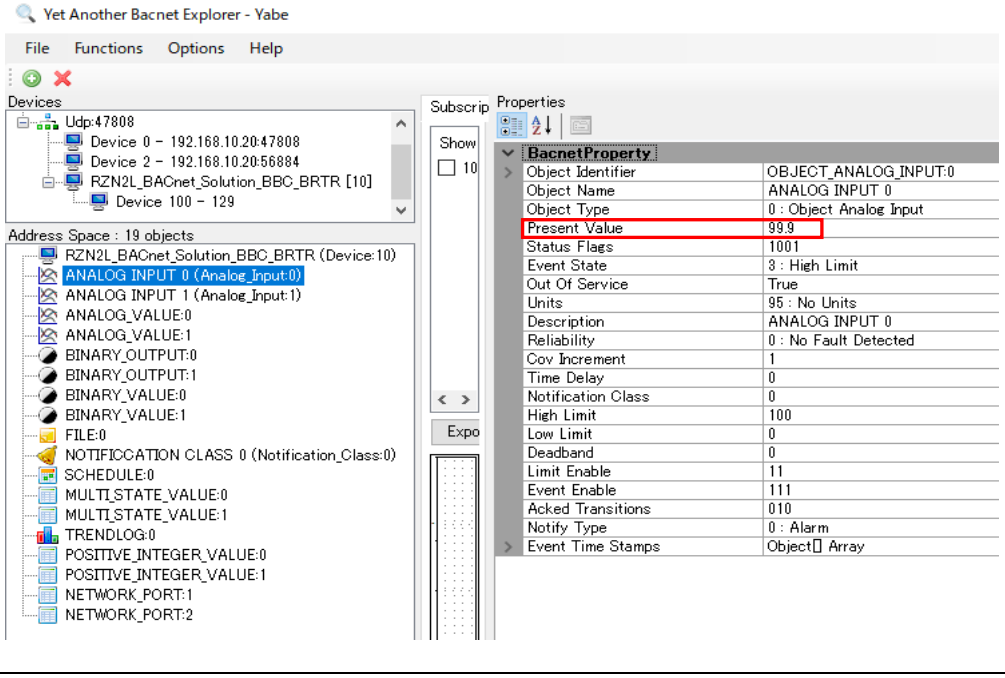

Fig.4-115 AnalogInput,0 object(2)

Then, right-click B-BC in the Devices window and select Alarm Summary.

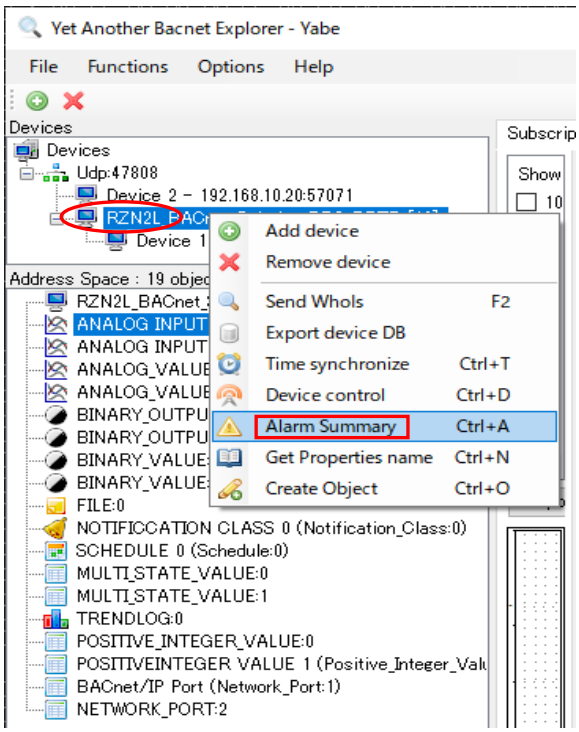

Fig.4-116 Select Alarm Summary(1)

The dialog box on the left side of Fig.4-117 is displayed so that you can confirm that the time stamps of the Event occurrence and return correspond to the Event Time Stamps of the AnalogInput,0 object on the right side.
Click “Ack selected alarm(s)” to close the dialog.

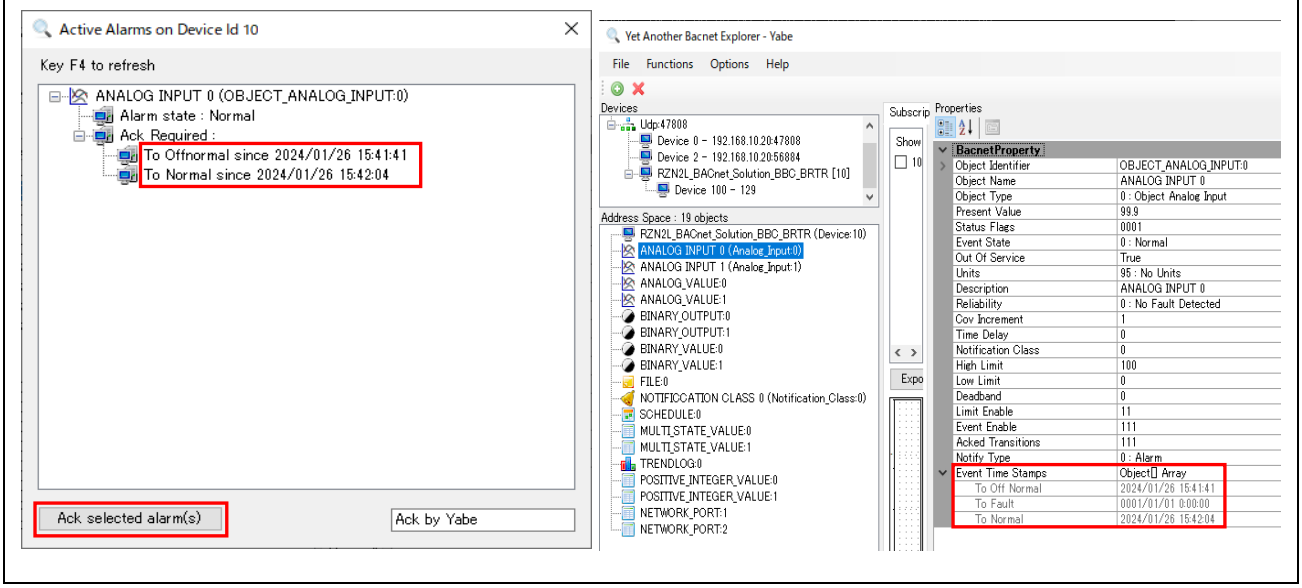

Fig.4-117 Select Alarm Summary(2)

The following Wireshark capture shows the service request and Ack when the described sequence of steps is performed.

No.14074 : B-BC requests Who-Is service to obtain the IP address of the Recipient configured in the Notification Editor. (DM-DDB-A)

No.14075 : Recipient (Yabe) responds I-Am. B-BC binds the IP address of the Recipient of device instance number (2) from the I-Am. (DM-DDB-A)
No.14076 : ConfirmedEventNotification from B-BC when transitioning to OffNormal.
No.14077 : Ack from Recipient.
No.14096 : ConfirmedEventNotification from B-BC when transitioning to Normal.
No.14114 : GetEventInformation service request from Recipient. This service request is triggered when the Alarm Summary described in the procedure is selected.
No.14145 : Result response Ack from B-BC.
No.14140 and 14142 : AcknowledgeAlarm service request from Recipient. This is a service request that is triggered when clicking on the “Ack selected alarm(s)” in the dialog described in the procedure. This is an AcknowledgeAlarm service request for two Event notifications of Offnormal and Normal transitions notified by the B-BC.
No.14144 : ConfirmedEventNotification to notify that the timestamp of the latest event passed in the AcknowledgeAlarm service request matches that of the B-BC end.

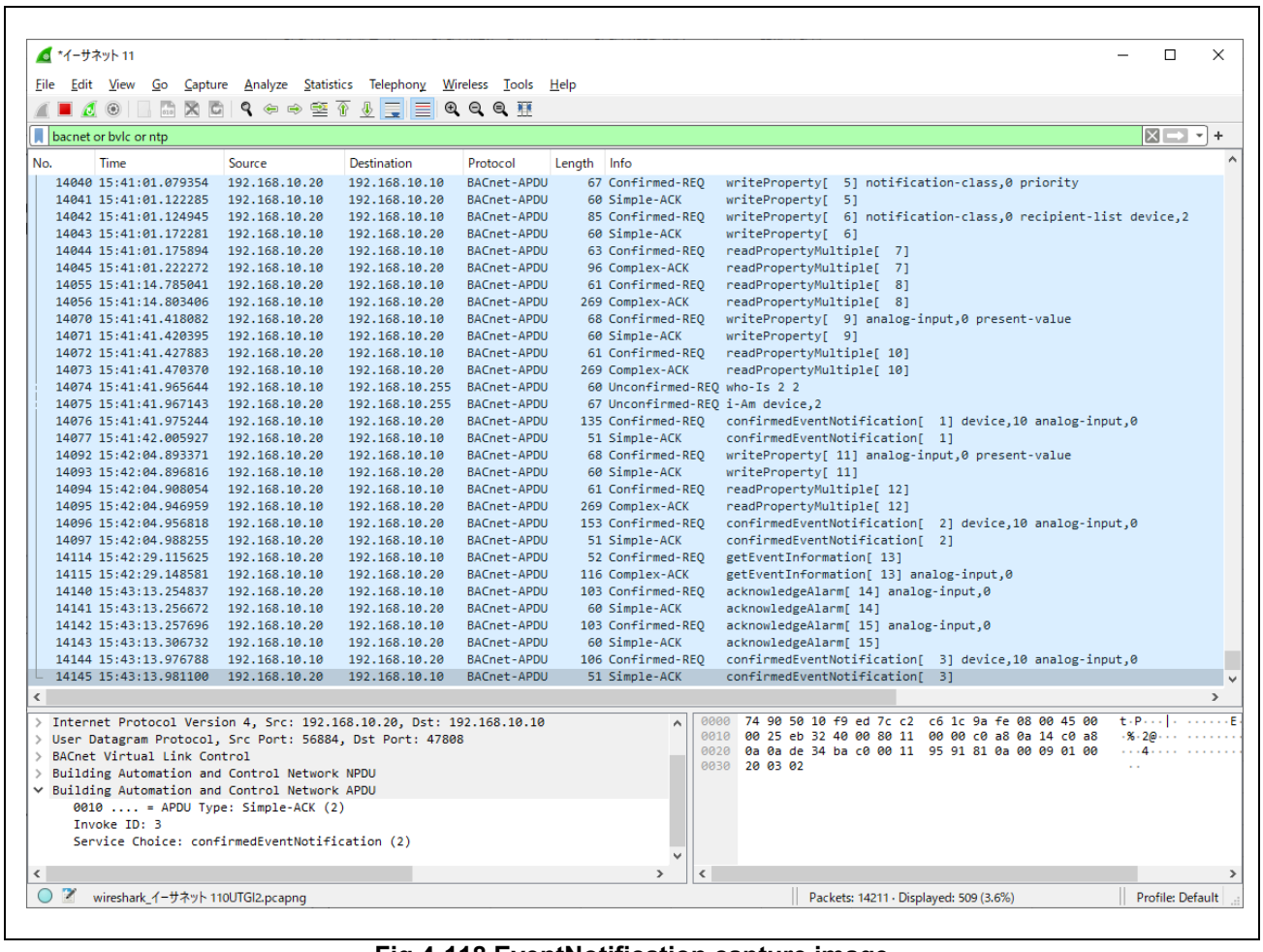

Fig.4-118 EventNotification capture image

4.6.4 AtomicReadFile

See also chapter 4.5.11, which describes the AtomicReadFile service from VTS. This chapter describes the procedure from Yabe. Yabe allows you to save files read from B-BC on your PC.

Open the “Settings” screen from Yabe's Options and set UdpMaxPayload to 480.

If changed, restart Yabe and start from Add Device.

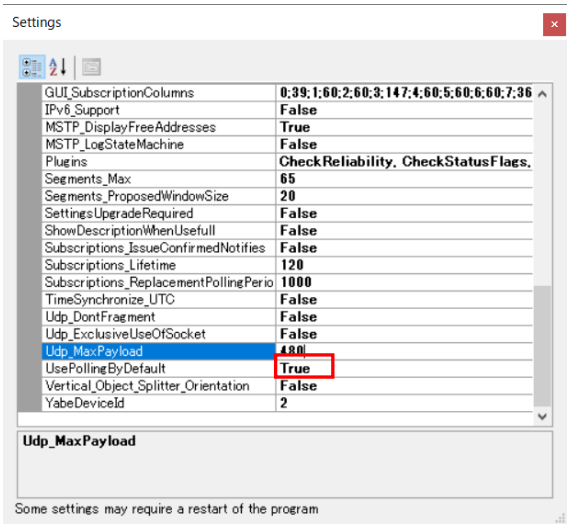

Fig.4-119 Udp_MaxPayload setting of Yabe

Select “FILE 0 (File:0)” in the “Address Space” window and right-click to select “Download File”.

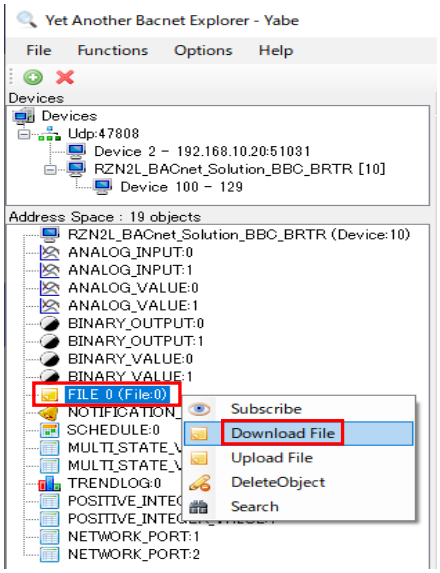

Fig.4-120 Select Download File

In the next dialog, name and save the file; click OK on the Done pop-up screen.

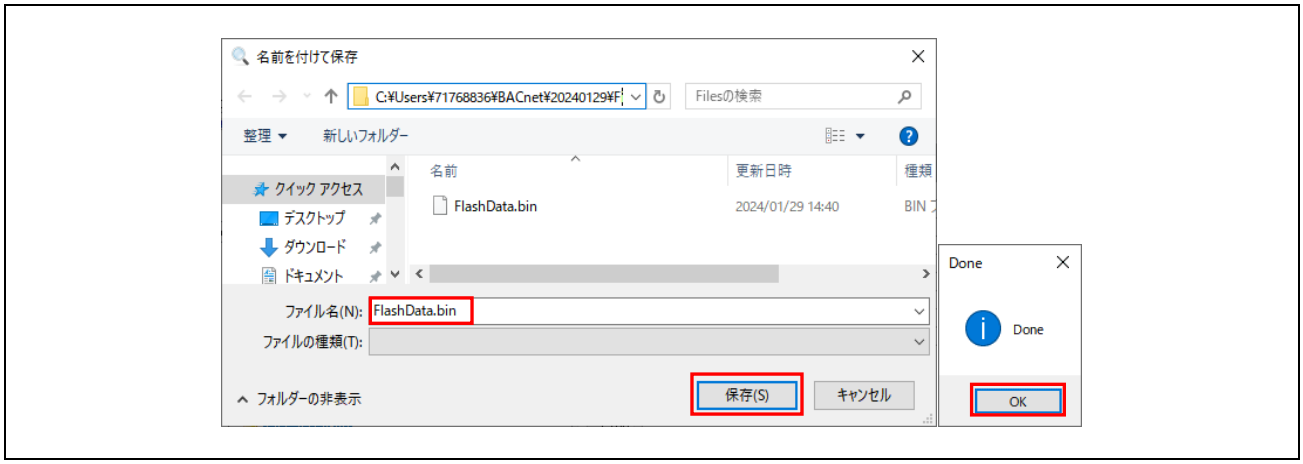

Fig.4-121 Save file with file name

4.6.5 AtomicWriteFile

See also chapter 4.5.12, which describes the AtomicWriteFile service from VTS. This chapter describes the procedure from Yabe.

By using Yabe, you can select files from your PC without any awareness of the Hex data stream (content) as with VTS. The selecting file is the file saved with AtomicReadFile service.

Select “FILE 0 (File:0)” in the “Address Space” window and right-click to select “Upload File”.

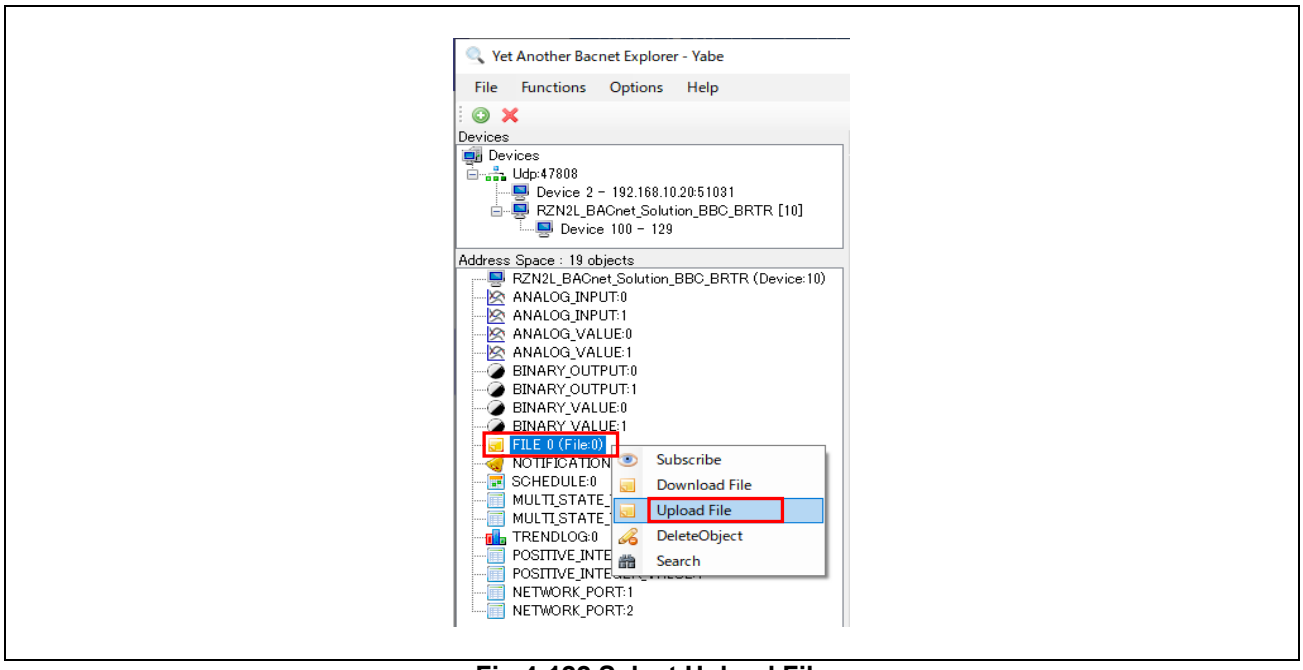

Fig.4-122 Select Upload File

Select and open the file in the following dialog; click OK on the Done pop-up screen.

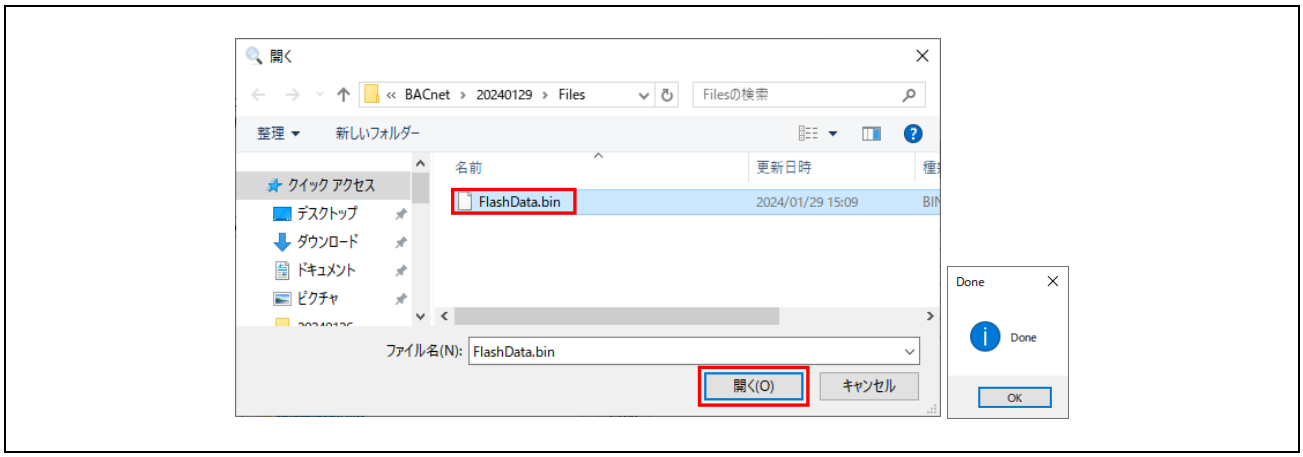

Fig.4-123 Open file

4.6.6 ReinitializeDevice

Note) The ReinitializeDevice service resets the target device. When the debugger is connected, push S3 RESET button (red) on the RSK board to disconnect the debugger connection. If this service is executed while the debugger is connecting, Ethernet communication will not be established after rebooting the B-BC.

ReinitializeDevice from VTS was explained in Chapter 4.5.9, and this section describes the procedure of it from Yabe.

Restrictions) The following State parameter of ReinitializeDevice is not yet supported by the B-BC sample software.

STARTBACKUP, ENDBACKUP, STARTRESTORE, ENDRESTORE, ABORTRESTORE

Yabe allows selection of the service parameter ACTIVATE_CHANGES, which could not be selected from VTS. Enter "filister" in Password and click OK.

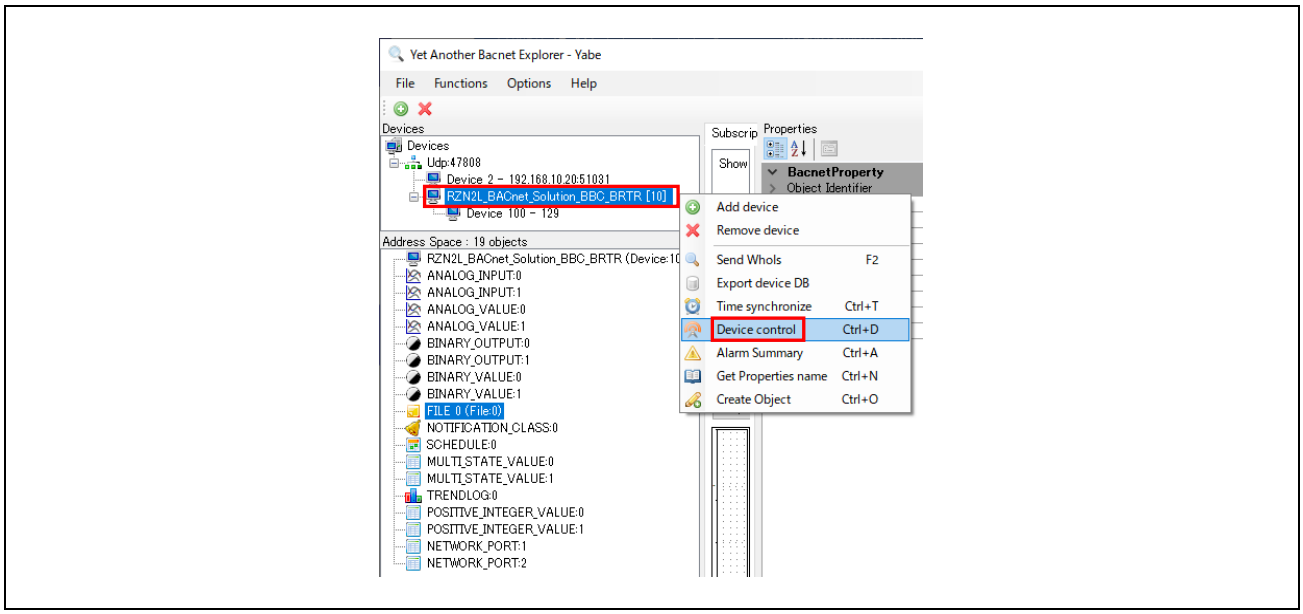

Fig.4-124 Select Device control

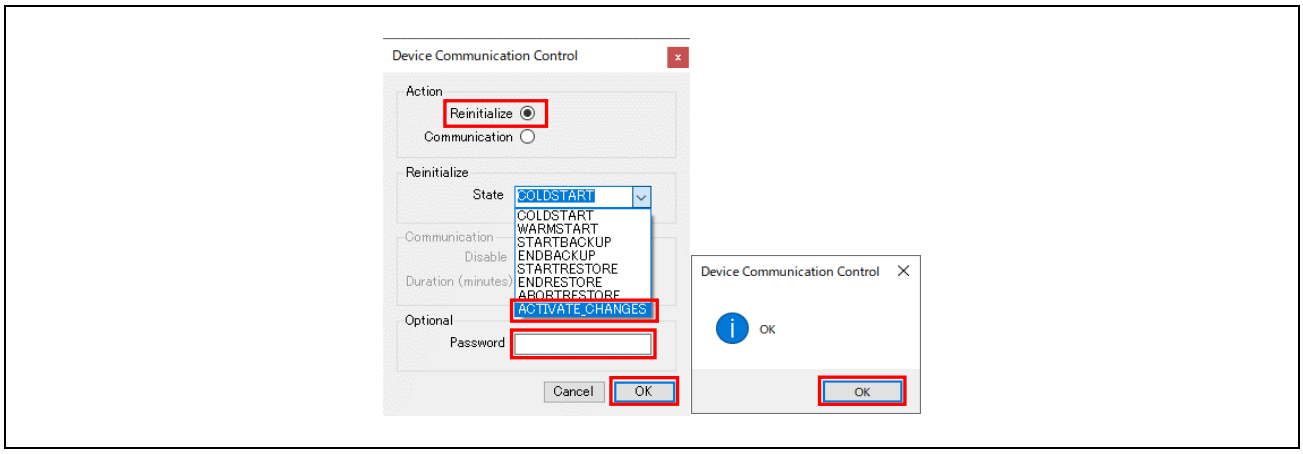

Fig.4-125 Select ACTIVATE_CHANGES

5. Initial Settings

This chapter describes each parameter setting in this sample software, including initial property values for each object.

5.1 Initial Values

Initial values of property for each object or parameters in this sample software are shown below. Please refer to the links shown in the Reference column of the table for how to change the initial values of each property.

5.1.1 Ethernet MAC address(IP)

In principle, Ethernet MAC address is uniquely assigned to all network devices, therefore, please set it individually for each RSK board with reference to chapter 5.2.1.

Especially when multiple RSK boards are connected on the same BACnet/IP network, be sure to change it.

Table 5-1 Ethernet MAC address

No.	Ethernet MAC address(IP)	Initial value	Reference
1	MAC address printed on CN14 of RSK board	uint8_t g_ether0_mac_address[6] = { 0x00, 0x11, 0x22, 0x33, 0x44, 0x55 };	5.2.1 Ethernet MAC address(IP)

5.1.2 Device

Initial values of device object properties are shown.

Table 5-2 Device,10 object properties

No.	Object	Property	Initial value	Reference
1	device,10	object-identifier	device, 10	5.2.2 Device instance
2		object-name	RZN2L_BACnet_Solution_BBC_BRTR	5.2.3 Device name
3		object-type	device (8)	
4		system-status	operational (0)	
5		vendor-name	UTF-8 'Renesas Electronics Corporation'	
6		vendor-identifier	(Unsigned) 9999	
7		model-name	UTF-8 'RZN2L_BBC_BRTR'	
8		firmware-revision	UTF-8 '1.0.0'	
9		application-software-version	UTF-8 '1.0.0'	
10		protocol-version	(Unsigned) 1	
11		protocol-revision	(Unsigned) 23	
12		protocol-service-supported	acknowledgeAlarm = TRUE getAlarmSummary = TRUE subscribeCOV = TRUE atomicReadFile = TRUE atomicWriteFile = TRUE readProperty = TRUE readPropertyMultiple = TRUE writeProperty = TRUE writePropertyMultiple = TRUE deviceCommunicationControl = TRUE reinitializeDevice = TRUE i-Am = TRUE timeSynchronization = TRUE who-Has = TRUE who-Is = TRUE readRange = TRUE utcTimeSynchronization = TRUE getEventInformation = TRUE	
13		protocol-object-type-supported	analog-input = TRUE analog-value = TRUE binary-output = TRUE binary-value = TRUE device = TRUE file = TRUE notification-class = TRUE schedule = TRUE multi-state-value = TRUE trend-log = TRUE positive-integer-value = TRUE network-port = TRUE	
14		object-list	device, 10 network-port, 1 network-port, 2 analog-input, 0 analog-input, 1 analog-value, 0 analog-value, 1 binary-output, 0 binary-output, 1 binary-value, 0 binary-value, 1 notification-class, 0 multi-state-value, 0 multi-state-value, 1 trend-log, 0 file, 0 positive-integer-value, 0 positive-integer-value, 1 schedule, 0	5.2.4 Number of objects
15		max-apdu-length-accepted	(Unsigned) 480	
16		segmentation-supported	no-segmentation (3)	
17		apdu-timeout	(Unsigned) 3000	
18		number-of-apdu-retries	(Unsigned) 3	
19		device-address-binding	DeviceIdentifier:-	empty

No.	Object	Property	Initial value	Reference
			network-number:- MAC-address:Port:-	
20		database-revision	(Unsigned) 3	
21		max-master	(Unsigned) 127	
22		max-info-frames	(Unsigned) 1	
23		description	UTF-8 'Renesas RZN2L_BACnet_Solution'	
24		local-time	0:01:34.0 A.M. = 00:01:34.0	
25		utc-offset	(Signed) -540	5.2.5 UTC_Offset
26		local-date	January 1, 2000, (Day of Week = Saturday)	
27		daylights-savings-status	FALSE	
28		location	UTF-8 'Tokyo,Japan'	
29		active-cov-subscriptions	Subscription 1 Recipient>Recipient Process>Recipient network-number:- MAC-address:- Port:- ProcessIdentifier:- Monitored Property Reference ObjectIdentifier:- Property Identifier:- Issue Confirmed Notifications:- Time Remaining:-	empty
30		property-list	system-status (112) vendor-name (121) vendor-identifier (120) model-name (70) firmware-revision (44) application-software-version (12) protocol-version (98) protocol-revision (139) protocol-services-supported (97) protocol-object-types-supported (96) object-list (76) max-apdu-length-accepted (62) segmentation-supported (107) apdu-timeout (11) number-of-APDU-retries (73) device-address-binding (30) database-revision (155) max-master (64) max-info-frames (63) description (28) local-time (57) utc-offset (119) local-date (56) daylights-savings-status (24) location (58) active-cov-subscriptions (152))	

5.1.3 Analog Input

Table 5-3 AnalogInput,0 object properties

No.	Object	Property	Initial value		Reference
1	analog-input,0	object-identifier	analog-input, 0		
2		object-name	ANALOG INPUT 0		
3		object-type	analog-input (0)		
4		present-value	0.0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE fault = FALSE overridden = FALSE out-of-service = FALSE	
6		event-state	normal (0)		
7		out-of-service	FALSE		5.2.16 OutOfService
8		units	No Units (95)		
9		description	UTF-8 'ANALOG INPUT 0'		
10		reliability	no-fault-detected (0)		
11		cov-increment	1.000000 (Real)		
12		time-delay	(Unsigned) 0		
13		notification-class	(Unsigned) 0		
14		high-limit	100.000000 (Real)		
15		low-limit	0.000000 (Real)		
16		deadband	0.000000 (Real)		
17		limit-enable	(Bit String) (TT)	low-limit = TRUE high-limit = TRUE	
18		event-enable	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
19		acked-transition	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
20		Notify Type	alarm (0)		
21		eventTimeStamps	TO-OFFNORMAL	Date: any Time: any	
			TO-FAULT	Date: any Time: any	
			TO-NORMAL	Date: any Time: any	
22		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) units (117) description (28) reliability (103) cov-increment (22) time-delay (113) notification-class (17) high-limit (45) low-limit (59) deadband (25) limit-enable (52) event-enable (35) acked-transition (0) notify-type (72) event-time-stamp (130)		

Table 5-4 AnalogInput,1 object properties

No.	Object	Property	Initial value		Reference
1	analog-input,1	object-identifier	analog-input, 1		
2		object-name	ANALOG INPUT 1		
3		object-type	analog-input (0)		
4		present-value	0.0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE fault = FALSE overridden = FALSE out-of-service = FALSE	
6		event-state	normal (0)		

No.	Object	Property	Initial value		Reference
7		out-of-service	FALSE		5.2.16 OutOfService
8		units	No Units (95)		
9		description	UTF-8 'ANALOG INPUT 1'		
10		reliability	no-fault-detected (0)		
11		cov-increment	1.000000 (Real)		
12		time-delay	(Unsigned) 0		
13		notification-class	(Unsigned) 0		
14		high-limit	100.000000 (Real)		
15		low-limit	0.000000 (Real)		
16		deadband	0.000000 (Real)		
17		limit-enable	(Bit String) (TT)	low-limit = TRUE high-limit = TRUE	
18		event-enable	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
19		acked-transition	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
20		Notify Type	alarm (0)		
21		eventTimeStamps	TO-OFFNORMAL	Date: any Time: any	
			TO-FAULT	Date: any Time: any	
			TO-NORMAL	Date: any Time: any	
22		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) units (117) description (28) reliability (103) cov-increment (22) time-delay (113) notification-class (17) high-limit (45) low-limit (59) deadband (25) limit-enable (52) event-enable (35) acked-transition (0) notify-type (72) event-time-stamp (130)		

5.1.4 Analog Value

Table 5-5 AnalogValue,0 object properties

No.	Object	Property	Initial value		Reference
1	analog-value,0	object-identifier	analog-value, 0		
2		object-name	ANALOG VALUE 0		
3		object-type	analog-value (2)		
4		present-value	(real) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE fault = FALSE overridden = FALSE out-of-service = FALSE	
6		event-state	normal (0)		
7		out-of-service	FALSE		5.2.16 OutOfService
8		units	No Units (95)		
9		description	UTF-8 'ANALOG VALUE 0'		
10		cov-increment	1.000000 (Real)		
11		time-delay	(Unsigned) 0		
12		notification-class	(Unsigned) 4194303		
13		high-limit	0.000000 (Real)		
14		low-limit	0.000000 (Real)		
15		deadband	0.000000 (Real)		
16		limit-enable	(Bit String) (TT)	low-limit = FALSE high-limit = FALSE	
17		event-enable	(Bit String) (TTT)	to-offnormal = FALSE to-fault = FALSE to-normal = FALSE	
18		acked-transition	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
19		Notify Type	alarm (0)		
20		eventTimeStamps	TO-OFFNORMAL	Date: any Time: any	
			TO-FAULT	Date: any Time: any	
			TO-NORMAL	Date: any Time: any	
21		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) units (117) description (28) cov-increment (22) time-delay (113) notification-class (17) high-limit (45) low-limit (59) deadband (25) limit-enable (52) event-enable (35) acked-transition (0) notify-type (72) event-time-stamp (130)		

Table 5-6 AnalogValue,1 object properties

No.	Object	Property	Initial value		Reference
1	analog-value,1	object-identifier	analog-value, 1		
2		object-name	ANALOG VALUE 1		
3		object-type	analog-value (2)		
4		present-value	(real) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	

No.	Object	Property	Initial value		Reference
				fault = FALSE	
				overridden = FALSE	
				out-of-service = FALSE	
6		event-state	normal (0)		
7		out-of-service	FALSE		5.2.16 OutOfService
8		units	No Units (95)		
9		description	UTF-8 'ANALOG VALUE 1'		
10		cov-increment	1.000000 (Real)		
11		time-delay	(Unsigned) 0		
12		notification-class	(Unsigned) 4194303		
13		high-limit	0.000000 (Real)		
14		low-limit	0.000000 (Real)		
15		deadband	0.000000 (Real)		
16		limit-enable	(Bit String) (TT)	low-limit = FALSE high-limit = FALSE	
17		event-enable	(Bit String) (TTT)	to-offnormal = FALSE to-fault = FALSE to-normal = FALSE	
18		acked-transition	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
19		Notify Type	alarm (0)		
20		eventTimeStamps	TO-OFFNORMAL	Date: any Time: any	
			TO-FAULT	Date: any Time: any	
			TO-NORMAL	Date: any Time: any	
21		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) units (117) description (28) cov-increment (22) time-delay (113) notification-class (17) high-limit (45) low-limit (59) deadband (25) limit-enable (52) event-enable (35) acked-transition (0) notify-type (72) event-time-stamp (130)		

5.1.5 Binary Output

Table 5-7 BinaryOutput,0 object properties

No.	Object	Property	Initial value		Reference
1	binary-output,0	object-identifier	binary-output, 0		
2		object-name	BINARY OUTPUT 0		
3		object-type	binary-output (4)		
4		present-value	(enum index) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
6				fault = FALSE	
7				overridden = FALSE	
8				out-of-service = FALSE	
9		event-state	normal (0)		
10		out-of-service	FALSE		5.2.16 OutOfService
11		polarity	0		
12		priority-array[1]	NULL		
13		priority-array[2]	NULL		
14		priority-array[3]	NULL		
15		priority-array[4]	NULL		
16		priority-array[5]	NULL		
17		priority-array[6]	NULL		
18		priority-array[7]	NULL		
19		priority-array[8]	NULL		
20		priority-array[9]	NULL		
21		priority-array[10]	NULL		
22		priority-array[11]	NULL		
23		priority-array[12]	NULL		
24		priority-array[13]	NULL		
25		priority-array[14]	NULL		
26		priority-array[15]	NULL		
27		priority-array[16]	NULL		
28		relinquish-default	0		
29		Current-command-priority	NULL		
30		Reliability	no-fault-detected (0)		
31		description	UTF-8 'BINARY OUTPUT 0'		
32		active-text	UTF-8 'Active'		
33		inactive-text	UTF-8 'Inactive'		
34		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) polarity (84) priority-array (87) relinquish-default (104) current-command-priority (431) reliability (103) description (28) active-text (4) inactive-text (46)		

Table 5-8 BinaryOutput,1 object properties

No.	Object	Property	Initial value	Reference
1	binary-output,1	object-identifier	binary-output, 1	
2		object-name	BINARY OUTPUT 1	
3		object-type	binary-output (4)	
4		present-value	(enum index) 0	

No.	Object	Property	Initial value		Reference
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
6				fault = FALSE	
7				overridden = FALSE	
8				out-of-service = FALSE	
9		event-state	normal (0)		
10		out-of-service	FALSE		5.2.16 OutOfService
11		polarity	0		
12		priority-array[1]	NULL		
13		priority-array[2]	NULL		
14		priority-array[3]	NULL		
15		priority-array[4]	NULL		
16		priority-array[5]	NULL		
17		priority-array[6]	NULL		
18		priority-array[7]	NULL		
19		priority-array[8]	NULL		
20		priority-array[9]	NULL		
21		priority-array[10]	NULL		
22		priority-array[11]	NULL		
23		priority-array[12]	NULL		
24		priority-array[13]	NULL		
25		priority-array[14]	NULL		
26		priority-array[15]	NULL		
27		priority-array[16]	NULL		
28		relinquish-default	0		
29		Current-command-priority	NULL		
30		Reliability	no-fault-detected (0)		
31		description	UTF-8 'BINARY OUTPUT 0'		
32		active-text	UTF-8 'Active'		
33		inactive-text	UTF-8 'Inactive'		
34		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) polarity (84) priority-array (87) relinquish-default (104) current-command-priority (431) reliability (103) description (28) active-text (4) inactive-text (46)		

5.1.6 Binary Value

Table 5-9 BinaryValue,0 object properties

No.	Object	Property	Initial value		Reference
1	binary-value,0	object-identifier	binary-value, 0		
2		object-name	BINARY VALUE 0		
3		object-type	binary-value (5)		
4		present-value	(enum index) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
6				fault = FALSE	
7				overridden = FALSE	
8				out-of-service = FALSE	
9		event-state	normal (0)		
10		out-of-service	FALSE		5.2.16 OutOfService
11		description	UTF-8 'BINARY VALUE 0'		

No.	Object	Property	Initial value	Reference
12		reliability	no-fault-detected (0)	
13		priority-array[1]	NULL	
14		priority-array[2]	NULL	
15		priority-array[3]	NULL	
16		priority-array[4]	NULL	
17		priority-array[5]	NULL	
18		priority-array[6]	NULL	
19		priority-array[7]	NULL	
20		priority-array[8]	NULL	
21		priority-array[9]	NULL	
22		priority-array[10]	NULL	
23		priority-array[11]	NULL	
24		priority-array[12]	NULL	
25		priority-array[13]	NULL	
26		priority-array[14]	NULL	
27		priority-array[15]	NULL	
28		priority-array[16]	NULL	
29		relinquish-default	0	
30		current-command-priority	NULL	
31		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) description (28) reliability (103) priority-array (87) relinquish-default (104) current-command-priority (431)	

Table 5-10 BinaryValue,1 object properties

No.	Object	Property	Initial value		Reference
1	binary-value,1	object-identifier	binary-value, 1		
2		object-name	BINARY VALUE 1		
3		object-type	binary-value (5)		
4		present-value	(enum index) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
6				fault = FALSE	
7				overridden = FALSE	
8				out-of-service = FALSE	
9		event-state	normal (0)		
10		out-of-service	FALSE		5.2.16 OutOfService
11		description	UTF-8 'BINARY VALUE 1'		
12		reliability	no-fault-detected (0)		
13		priority-array[1]	NULL		
14		priority-array[2]	NULL		
15		priority-array[3]	NULL		
16		priority-array[4]	NULL		
17		priority-array[5]	NULL		
18		priority-array[6]	NULL		
19		priority-array[7]	NULL		
20		priority-array[8]	NULL		
21		priority-array[9]	NULL		
22		priority-array[10]	NULL		
23		priority-array[11]	NULL		
24		priority-array[12]	NULL		
25		priority-array[13]	NULL		
26		priority-array[14]	NULL		
27		priority-array[15]	NULL		
28		priority-array[16]	NULL		
29		relinquish-default	0		
30		current-command-priority	NULL		
31		property-list	present-value (85) status-flags (111)		

No.	Object	Property	Initial value	Reference
			event-state (36) out-of-service (81) description (28) reliability (103) priority-array (87) relinquish-default (104) current-command-priority (431)	

5.1.7 File

Table 5-11 File,0 object properties

No.	Object	Property	Initial value	Reference
1	file,0	object-identifier	file, 0	
2		object-name	FILE 0	
3		object-type	file (10)	
4		file-type	UTF-8 'application/octet-stream'	
5		file-size	(Unsigned) 148	
6		modification-date	Date: April 1, 2006, (Day of Week = Saturday) Time: 7:00:03.1 A.M. = 07:00:03.1	
7		archive	FALSE	
8		read-only	FALSE	
9		file-access-method	stream-access (1)	
10		description	UTF-8 'FlashData.bin'	
11		property-list	file-type (43) file-size (42) modification-date (71) archive (13) read-only (99) file-access-method (41) description (28)	

5.1.8 Notification Class

Table 5-12 Notification Class,0 object properties

No.	Object	Property		Initial value		Reference
1	Notification-class,0	object-identifier		notification-class, 0		
2		object-name		NOTIFICCATION CLASS 0		
3		object-type		notification-class (15)		
4		notification-class		(Unsigned) 0		
5		priority	To Off ormal	(Unsigned) 255		
			To Fault	(Unsigned) 255		
			To Normal	(Unsigned) 255		
6		ack-required		(Bit String) (TTT)	To_OffNormal = TRUE To_Fault = TRUE To_Normal = TRUE	
7		recipient-list	valid Days	(Bit String) (TTTTTTT)	Monday = TRUE Tuesday = TRUE Wednesday = TRUE Thursday = TRUE Friday = TRUE Saturday = TRUE Sunday = TRUE	
			from time	00:00:00.0		
			to time	23:59:59.0		
			DeviceIdentifier	device, 4194303		
			ProcessIdentifier	0		
			issue confirmed notifications	TRUE		
			transitions	(Bit String) (TTT)	to-offnormal = TRUE to-fault = TRUE to-normal = TRUE	
8		description		UTF-8 'NOTIFICATION CLASS 0'		
9		property-list		notification-class (17) priority (86) ack-required (1) recipient-list (102) description (28)		

5.1.9 Schedule

Table 5-13 Schedule,0 object properties

No.	Object	Property			Initial value		Reference
1	schedule,0	object-identifier			schedule, 0		
2		object-name			SCHEDULE 0		
3		object-type			schedule (17)		
4		Present Value			(enum index) 1		
5		effective-period			January 1, any year, (Day of Week = any day of week)		
					December 31, any year, (Day of Week = any day of week)		
6		schedule-default			0		
7		list-of-object-property-references	ObjectIdentifier		binary-output, 0		
			Property Identifier		present-value (85)		
			DeviceIdentifier		device, 100		
8		priority-for-writing			(Unsigned) 16		
9		status-flags			(Bit String) (FFFF)	in-alarm = FALSE fault = FALSE overridden = FALSE out-of-service = FALSE	
10		reliability			no-fault-detected (0)		
11		out-of-service			FALSE		
12		description			UTF-8 'SCHEDULE 0'		
13		weekly-schedule	Monday	Time	00:00:00.0		
				Value	0		
			Tuesday	Time	00:00:00.0		
				Value	0		
			Wednesday	Time	00:00:00.0		
				Value	0		
			Thursday	Time	00:00:00.0		
				Value	0		
			Friday	Time	00:00:00.0		
				Value	0		
			Saturday	Time	00:00:00.0		
				Value	0		
			Sunday	Time	00:00:00.0		
				Value	0		
14		property-list			present-value (85) effective-period (32) schedule-default (174) list-of-object-property-references (54) priority-for-writing (88) status-flags (111) reliability (103) out-of-service (81) description (28) weekly-schedule (123)		

5.1.10 Multi State Value

Table 5-14 MultiStateValue,0 object properties

No.	Object	Property	Initial value		Reference
1	multi-state-value,0	object-identifier	multi-state-value, 0		
2		object-name	MULTISTATE VALUE 0		
3		object-type	multi-state-value (19)		
4		present-value	(uint) 1		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
				fault = FALSE	
				overridden = FALSE	
				out-of-service = FALSE	
6		event-state	normal (0)		
7		out-of-service	FALSE		5.2.16 OutOfService
8		number-of-states	(Unsigned) 3		5.2.6 Number of states
9		description	UTF-8 'MULTISTATE VALUE 0'		
10		state-text[0][3][64]	UTF-8 'State 1' UTF-8 'State 2' UTF-8 'State 3'		5.2.7 State text
11		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) number-of-states (74) description (28) state-text (110)		

Table 5-15 MultiStateValue,1 object properties

No.	Object	Property	Initial value		Reference
1	multi-state-value,1	object-identifier	multi-state-value, 1		
2		object-name	MULTISTATE VALUE 1		
3		object-type	multi-state-value (19)		
4		present-value	(uint) 1		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
				fault = FALSE	
				overridden = FALSE	
				out-of-service = FALSE	
6		event-state	normal (0)		
7		out-of-service	FALSE		5.2.16 OutOfService
8		number-of-states	(Unsigned) 3		5.2.6 Number of states
9		description	UTF-8 'MULTISTATE VALUE 1'		
10		state-text[1][3][64]	UTF-8 'State 1' UTF-8 'State 2' UTF-8 'State 3'		5.2.7 State text
11		property-list	present-value (85) status-flags (111) event-state (36) out-of-service (81) number-of-states (74) description (28) state-text (110)		

5.1.11 Trend Log

Table 5-16 Trend Log,0 object properties

No.	Object	Property		Initial value		Reference
1	trendlog,0	object-identifier		trend-log, 0		
2		object-name		TREND LOG 0		
3		object-type		trend-log (20)		
4		enable		TRUE		
5		stop-when-full		FALSE		
6		buffer-size		(Unsigned) 1000		
7		log-buffer				
8		record-count		(Unsigned) 0		
9		total-record-count		(Unsigned) 0		
10		event-state		normal (0)		
11		logging-type		polled (0)		
12		status-flags		(Bit String) (FFFF)	in-alarm = FALSE	
					fault = FALSE	
					overridden = FALSE	
					out-of-service = FALSE	
13		description		UTF-8 'TREND LOG 0'		
14		start-time	Date	January 1, 2009, (Day of Week = Thursday)		
			Time	00:00:00.0		
15		stop-time	Date	December 22, 2020, (Day of Week = Tuesday)		
			Time	23:59:59.99		
16		log-device-object-property	ObjectIdentifier:	analog-input, 0		
			Property Identifier	present-value (85)		
			DeviceIdentifier	device, 10		
17		log-interval		(Unsigned) 90000		
18		align-intervals		TRUE		
19		interval-offset		(Unsigned) 0		
20		trigger		FALSE		
21		property-list		enable (133) stop-when-full (144) buffer-size (126) log-buffer (131) record-count (141) total-record-count (145) event-state (36) logging-type (197) status-flags (111) description (28) start-time (142) stop-time (143) log-device-object-property (132) log-interval (134) align-intervals (193) interval-offset (195) trigger (205)		

5.1.12 Positive Integer Value

Table 5-17 PositiveIntegerValue,0 object properties

No.	Object	Property	Initial value		Reference
1	positive-integer-value,0	object-identifier	positive-integer-value, 0		
2		object-name	POSITIVEINTEGER VALUE 0		
3		object-type	positive-integer-value (48)		
4		present-value	(uint) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
6				fault = FALSE	
7				overridden = FALSE	
8				out-of-service = FALSE	
9		units	No Units (95)		
10		description	UTF-8 'POSITIVEINTEGER VALUE 0'		
11		event-state	normal (0)		
12		out-of-service	FALSE		5.2.16 OutOfService
13		property-list	present-value (85) status-flags (111) units (117) description (28) event-state (36) out-of-service (81)		

Table 5-18 PositiveIntegerValue,1 object properties

No.	Object	Property	Initial value		Reference
1	positive-integer-value,0	object-identifier	positive-integer-value, 1		
2		object-name	POSITIVEINTEGER VALUE 1		
3		object-type	positive-integer-value (48)		
4		present-value	(uint) 0		
5		status-flags	(Bit String) (FFFF)	in-alarm = FALSE	
6				fault = FALSE	
7				overridden = FALSE	
8				out-of-service = FALSE	
9		units	No Units (95)		
10		description	UTF-8 'POSITIVEINTEGER VALUE 1'		
11		event-state	normal (0)		
12		out-of-service	FALSE		5.2.16 OutOfService
13		property-list	present-value (85) status-flags (111) units (117) description (28) event-state (36) out-of-service (81)		

5.1.13 Network Port

Initial values of NetworkPort object properties are shown.

Table 5-19 NetworkPort,1 object properties(for BIP)

No.	Object	Property		Initial value		Reference
1	network-port,1	object-identifier		network-port, 1		
2		object-name		BACnet/IP Port		
3		object-type		network-port (56)		
4		status-flags		(Bit String) (FFFF)	in-alarm = FALSE fault = FALSE overridden = FALSE out-of-service = FALSE	
5		reliability		no-fault-detected (0)		
6		out-of-service		FALSE		
7		network-type		ipv4 (5)		
9		protocol-level		bacnet-application (2)		
10		changes-pending		FALSE		
11		description		UTF-8 'NETWORK PORT 1'		
12		mac-address		c0a80a0abac0 (hex)		5.2.11 BACnet IP address
13		bacnet-ip-mode		normal (0)		
14		ip-address		c0a80a0a (hex)		5.2.11 BACnet IP address
15		bacnet-ip-udp-port		(Unsigned) 47808		5.2.11 BACnet IP address
16		ip-subnet-mask		ffffff00 (hex)		
17		ip-default-gateway		c0a80a01 (hex)		
18		ip-dns-server		00000000 (hex)		
19		fd-bbmd-address	ip-address	00000000		5.2.13 FD_BBMD_Address
20			port	(Unsigned) 47808		5.2.13 FD_BBMD_Address
21		fd-subscription-lifetime		(Unsigned) 60000		5.2.14 FD_Subscription_Lifetime
22		property-list		status-flags (111) reliability (103) out-of-service (81) network-type (427) protocol-level (482) changes-pending (416) description (28) mac-address (423) bacnet-ip-mode (408) bacnet-ip-address (400) bacnet-ip-udp-port (412) bacnet-ip-subnet-mask (411) bacnet-ip-default-gateway (401) bacnet-ip-dns-server (406) fd-bbmd-address (418) fd-subscription-lifetime (419)		

Table 5-20 NetworkPort,2 object properties(for MS/TP)

No.	Object	Property	Initial value		Reference
1	network-port,100	object-identifier	network-port, 2		
2		object-name	MS/TP Port		
3		object-type	network-port (56)		
4		status-flags	(Bit String) (FFFF)	in-alarm = FALSE fault = FALSE overridden = FALSE out-of-service = FALSE	
5		reliability	no-fault-detected (0)		
6		out-of-service	FALSE		
7		network-type	mstp (2)		

8		protocol-level	bacnet-application (2)	
9		network-number	(Unsigned) 2	5.2.8 Network number
10		network-number-quality	configured (3)	
11		changes-pending	FALSE	
12		apdu-length	(Unsigned) 480	
13		link-speed	115200.000000 (Real)	5.2.9 Link speed
14		description	UTF-8 'NETWORK PORT 2'	
15		mac-address	05	5.2.10 MAC address
16		max-master	(Unsigned) 127	
17		max-info-frames	(Unsigned) 1	
18		property-list	status-flags (111) reliability (103) out-of-service (81) network-type (427) protocol-level (482) network-number (425) network-number-quality (426) changes-pending (416) apdu-length (399) link-speed (420) description (28) mac-address (423) max-master (64) max-info-frames (63)	

5.1.14 Password

When B-BC receives ReinitializeDevice service or DeviceCommunicationControl service, it checks the password sent by the BACnet user and executes the service only when it matches.

The password implemented in the BACnet stack is used as the default value, refer to Section 5.2.15 to change it.

Table 5-21 Password

Service	Initial value	Reference
ReinitializeDevice	filister	5.2.15 Password
DeviceCommunicationControl	filister	

5.2 Change Initial Values

This section explains how to change each property initial value and supplementary information for this sample software.

To change the initial values, the source code and macro settings in the sample project need to be changed, rebuilt and rerun, but some data can also be changed through terminal software using the initial configuration commands described in Chapter 5.3. If the value is changed by the initial configuration command, the value becomes effective after the board reset.

The following data can be changed by the initial configuration commands.

5.2.1 Ethernet MAC address(IP)
5.2.2 Device instance
5.2.3 Device name
5.2.5 UTC_Offset
5.2.10 MAC address
5.2.11 BACnet IP address
5.2.16 OutOfService

5.2.1 Ethernet MAC address(IP)

Set the MAC address pasted on CN14 shown in Fig. 5-1.

Fig. 5-1 MAC address pasted on CN14

(1) Setting by Rebuilding

Open Smart Configurator by double-clicking on configuration.xml from the tree shown in Fig. 5-2.

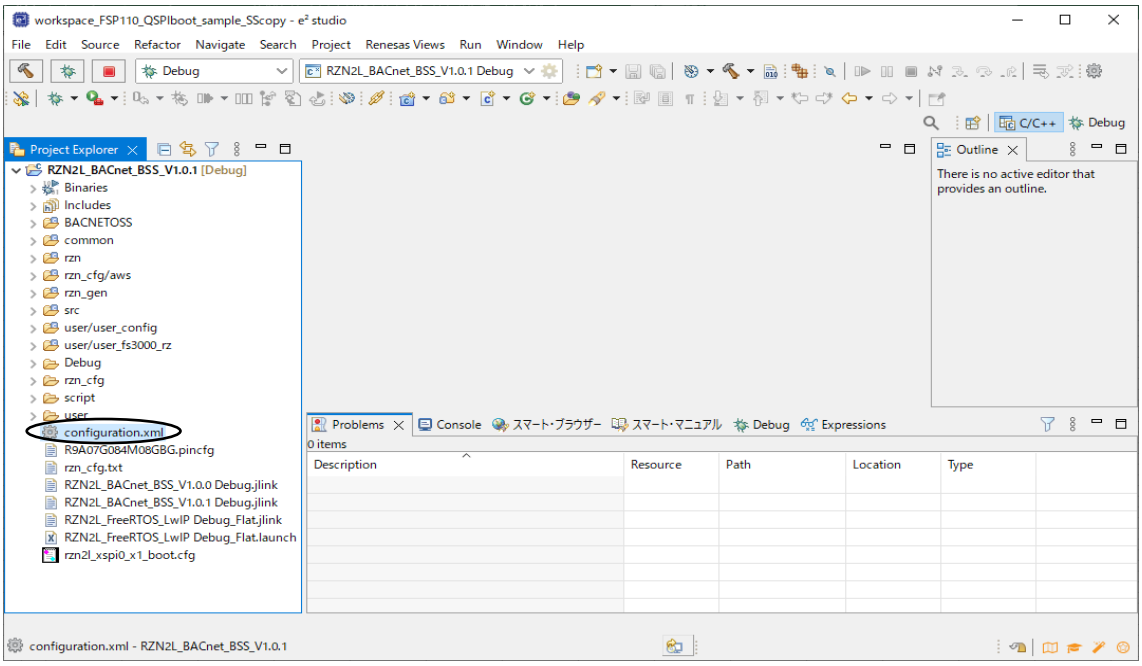

Fig. 5-2 Double click configuration.xml

Open the Stacks tab and click on g_ether0 Ethernet Driver on r_ether to select it.

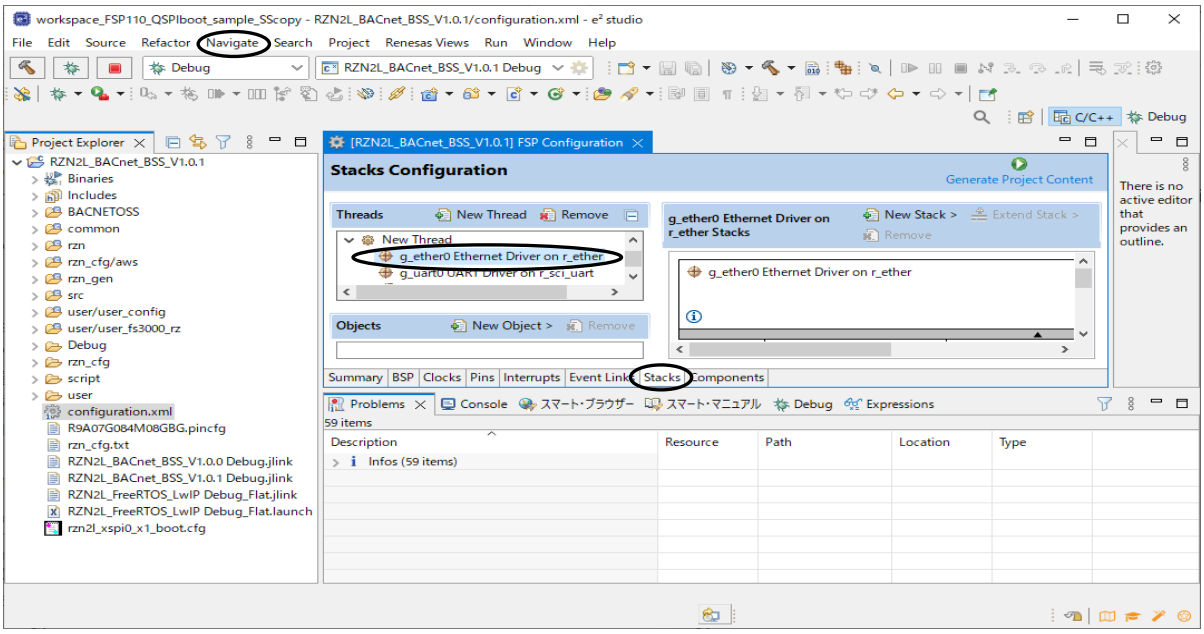

Fig. 5-3 Click Navigate

Then open the Navigate menu and click Show In>Properties.

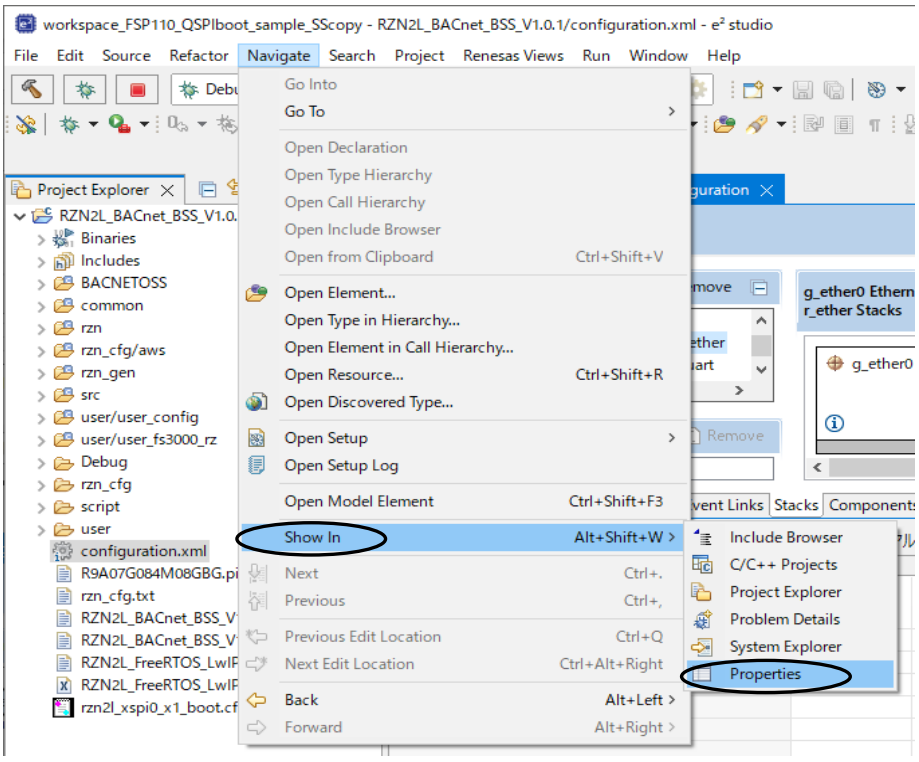

Fig. 5-4 Click Properties

Open the Properties tab and enter General>MAC address (e.g. 74:90:50:10:05:B0). Click Generate Project Content after entering the information. Finally, rebuild the project. See chapter 4.4.2 for build procedure.

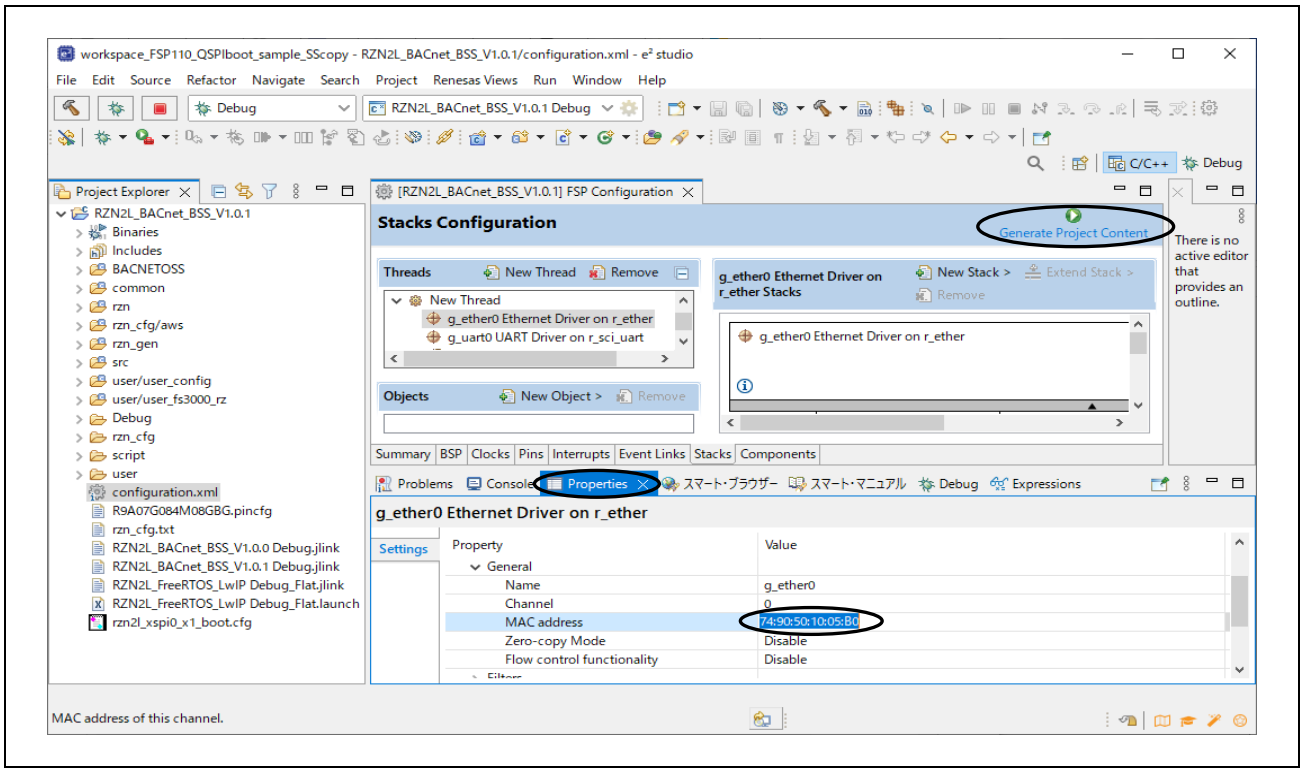

Fig. 5-5 Enter MAC address

(2) Setting by command

Refer to Chapter 5.3. The setting value by the initial configuration commands takes precedence over the setting value by building.

5.2.2 Device instance

The instance number, like other objects, is combined with the object type to form the Object Identifier property.

Only the instance number of the Device object type must be unique over the entire BACnet internetwork.

The setting range of Device instance is 0~4194303. However, 4194303 means invalid and is not used.

Refer to Chapter 5.3.

5.2.3 Device name

Refer to Chapter 5.3.

5.2.4 Number of objects

The following Symbol represents the initial value of the number of objects. This chapter describes how to change this Value.

Here, do not change the number of objects in #BACNET_NETWORK_PORTS_MAX from 2.

#MAX_ANALOG_INPUTS
#MAX_ANALOG_VALUES
#MAX_BINARY_OUTPUTS
#MAX_BINARY_VALUES
#MAX_FILES
#MAX_MULTISTATE_VALUES
#MAX_NOTIFICATION_CLASSES
#MAX_POSITIVEINTEGER_VALUES
#MAX_SCHEDULES
#MAX_TREND_LOGS

The change procedure is as follows.

Select the project name in the Project Explorer window, then open Properties in the Project menu. Click "Edit..." to make changes.

After making changes, click Apply and Close to apply the settings. Click Yes on the pop-up dialog. Finally, rebuild. See Chapter 4.4.2 for build procedures.

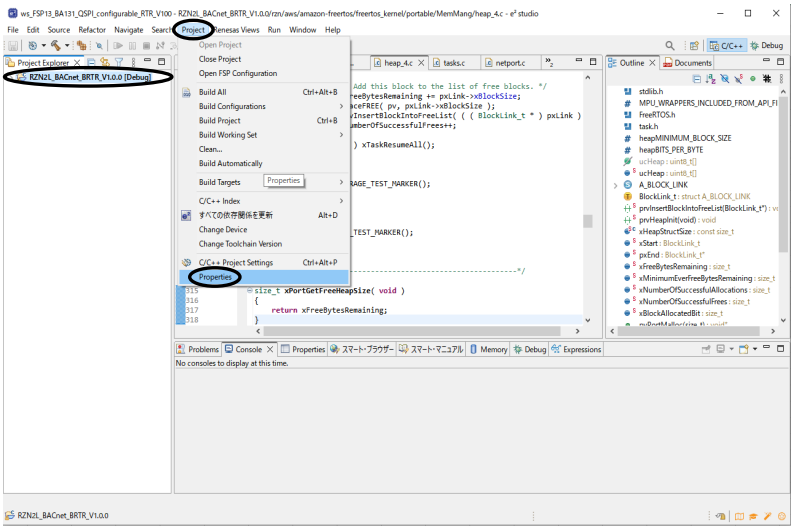

Fig. 5-6 Open Properties

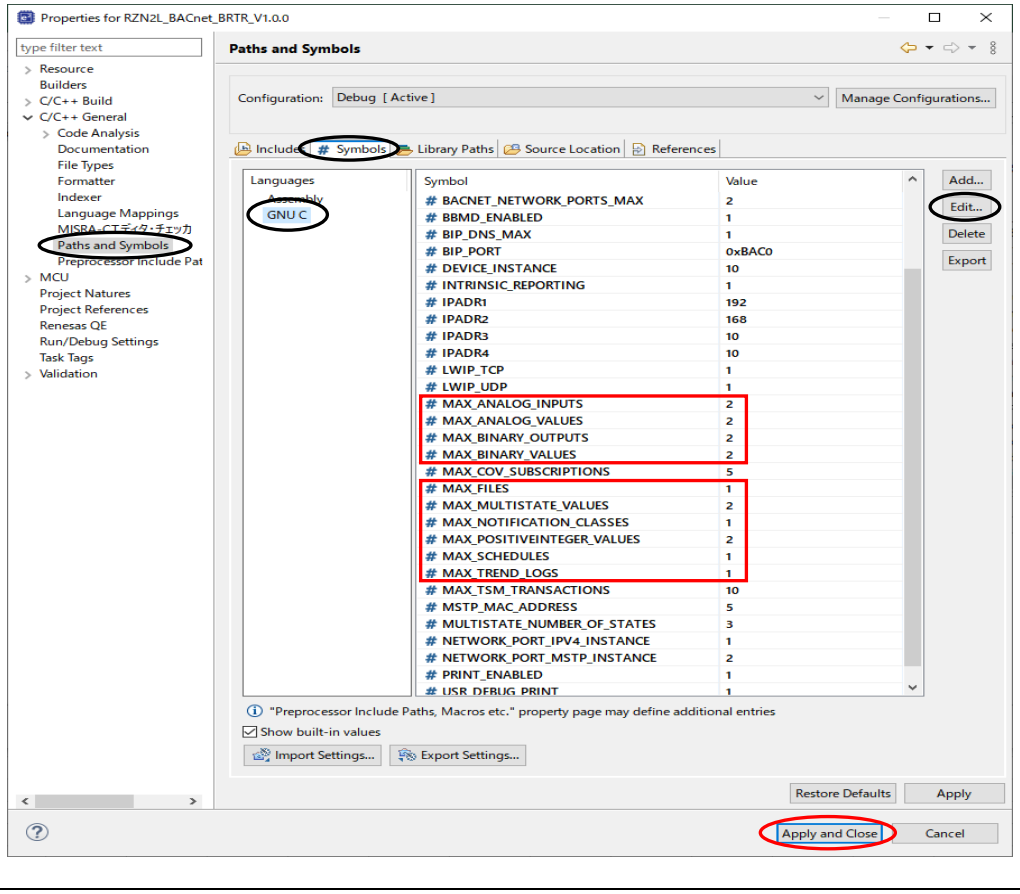

Fig. 5-7 Change Number of objects

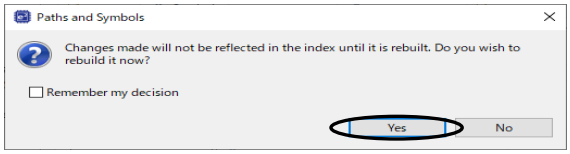

Fig. 5-8 Click Yes

5.2.5 UTC_Offset

Refer to Chapter 5.3.

5.2.6 Number of states

The Number of states property indicates the number of states represented by the present value of the Multi State Value object and can be changed in the range of 1 to 254. When changing it, the number of State text also needs to be increased or decreased at the same time. See chapter 5.2.7 to change the State text property
Table 5-22 shows the relation between Number of states, Present value, and State text.

Table 5-22 Other properties related to number of states

Number of states	Present value	State text(string)
3	1	State 1
	2	State 2
	3	State 3

Change the Value of the following Symbol to modify the Number of states. The change procedure is the same as in section 5.2.4.

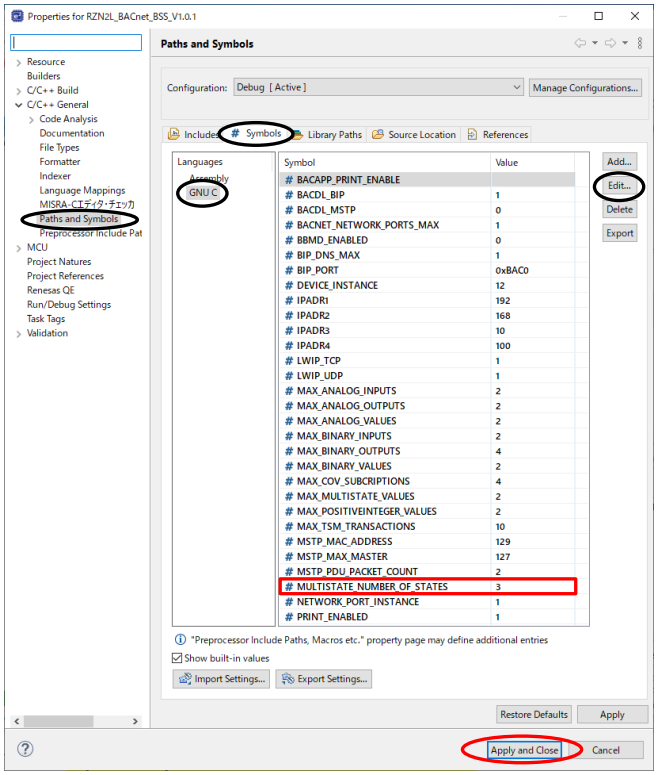

Fig. 5-9 Change Number of states

5.2.7 State text

The State text property is a string that represents the state, such as large, medium, small, etc., which the present value represents. The description string is 64 bytes or less. The default value of "Number of states" is 3, so it is assumed to be 3 here as well. If the value of "Number of states" is changed, the description of State text needs to be modified.
Search for "state_name[MULTISTATE_NUMBER_OF_STATES][64]" in the source code and change the setting.

5.2.8 Network number

Network number is an MS/TP-specific property that represents the BACnet network number associated with the network. The range of this property is 0~65534, where 0 means unknown.

When WriteProperty or WritePropertyMultiple services are executed for this property, the set value is written to Flash memory. See chapter 5.3.

5.2.9 Link speed

Link speed is expressed as bits per second. A value of 0 means that the communication speed is unknown.

It is valid for MS/TP connections and represents the baud rate of the UART. To change the baud rate, select from Table 5-23.

Table 5-23 Baud rate

Baud rate	Requirement
9600	Required
19200	Optional
38400	Required
57600	Optional
76800	Optional
115200	Optional

Link speed is changed by the following procedure.
Open the Properties tab with g_uart5 UART Driver on r_sci_uart selected in the Smart Configurator and enter Baud>Baud Rate. After input, click “Generate Project Content”.
See chapter 5.2.1 for activating the smart configurator.

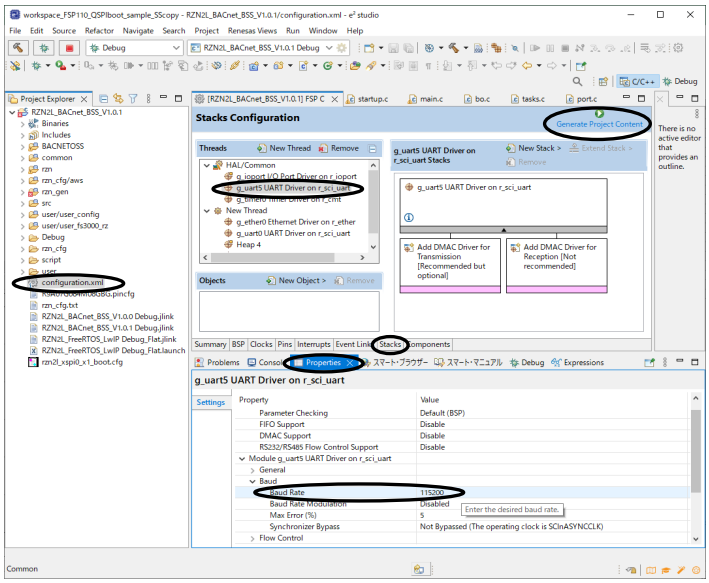

Fig. 5-10 Enter Baud Rate

In addition, modify the corresponding part in the source code.
BACNETOSS\sample.h

```
/* #define UART_BAUDRATE 38400 */
#define UART_BAUDRATE 115200
```

Fig. 5-11 Change UART_BAUDRATE

Finally, rebuild. See chapter 4.4.2 for build procedure.

5.2.10 MAC address

Refer to Chapter 5.3.

5.2.11 BACnet IP address

Refer to Chapter 5.3.

5.2.12 BACnet_IP_Mode

The BACnet_IP_Mode property of the Network Port object is for BIP only. The B-BC supports NORMAL and FOREIGN, does not support BBMD

NORMAL: The device is operating as neither a foreign device nor a BBMD over this network port.

FOREIGN: The device is operating as a foreign device over this network port.

BBMD: The device is operating as a BBMD over this network port.

This property is modified by the WriteProperty or WritePropertyMultiple services. Executing the WriteProperty and WritePropertyMultiple services write the setting value to Flash memory and set the Changes_Pending property to TRUE. The value becomes valid on reboot or when ReinitializeDevice service request is received with ACTIVATE_CHANGES or WARMSTART

There is no initial configuration command for this property. Refer to chapter 5.3 for details.

5.2.13 FD_BBMD_Address

FD_BBMD_Address property of the Network Port object is a BIP-specific property, and it consists of the IP address and UDP port number of the BBMD device. When BACnet_IP_Mode is FOREIGN, B-BC sends a Register-Foreign-Device BVLL message to the BBMD device to register itself as a foreign device.

This property is modified by the WriteProperty and WritePropertyMultiple services. Executing the WriteProperty and WritePropertyMultiple services write the setting value to Flash memory and set the Changes_Pending property to TRUE. The value becomes valid on reboot or when ReinitializeDevice service request is received with ACTIVATE_CHANGES or WARMSTART

Refer to (1) for the procedure to change the property.

There is no initial configuration command for this property. Refer to chapter 5.3 for details.

(1) FD_BBMD_Address setting with VTS

The following is the procedure for setting the FD_BBMD_Address property with VTS. Click ID... in the WriteProperty dialog. Configure the followings in the Object ID dialog.

Select Reserved for Object Type.

Set Reserved Type to 56 (OBJECT_NETWORK_PORT). (): Defined value in bacenum.h

Set Instance to 1. Click OK.

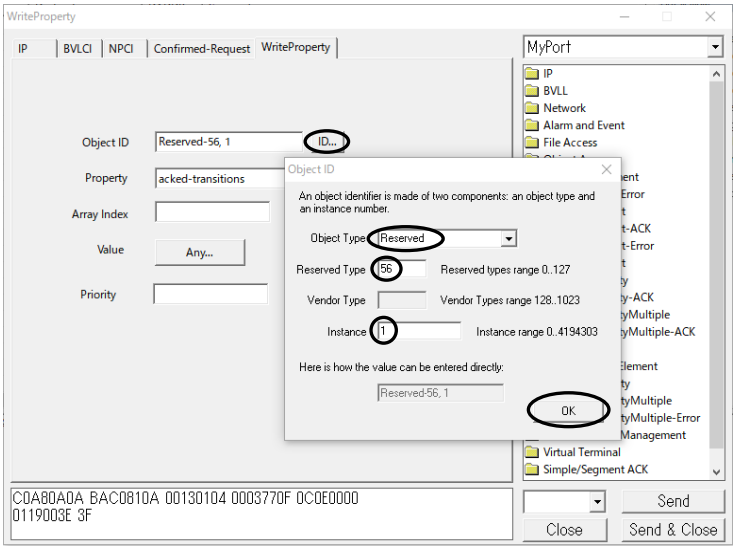

Fig. 5-12 WriteProperty dialog(1)

Select "< Enter numeric value >" from Property in the WriteProperty dialog.

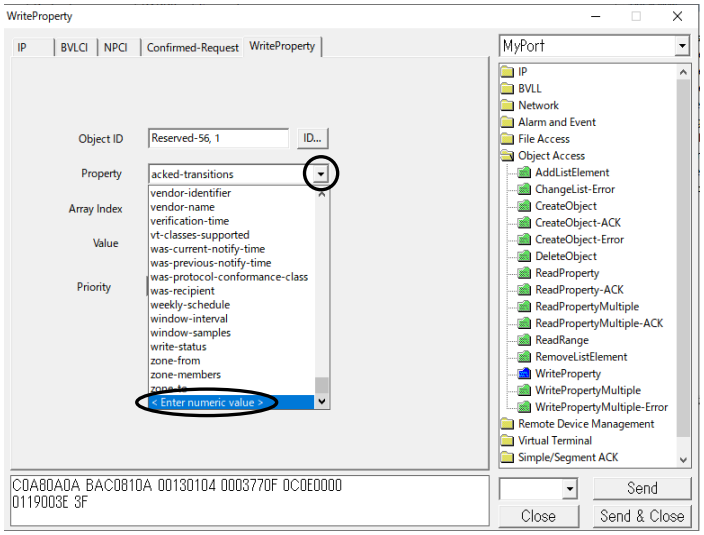

Fig. 5-13 WriteProperty dialog(2)

Set 418 (PROP_FD_BBMD_ADDRESS) in the PropertyIdentifier dialog and click OK.

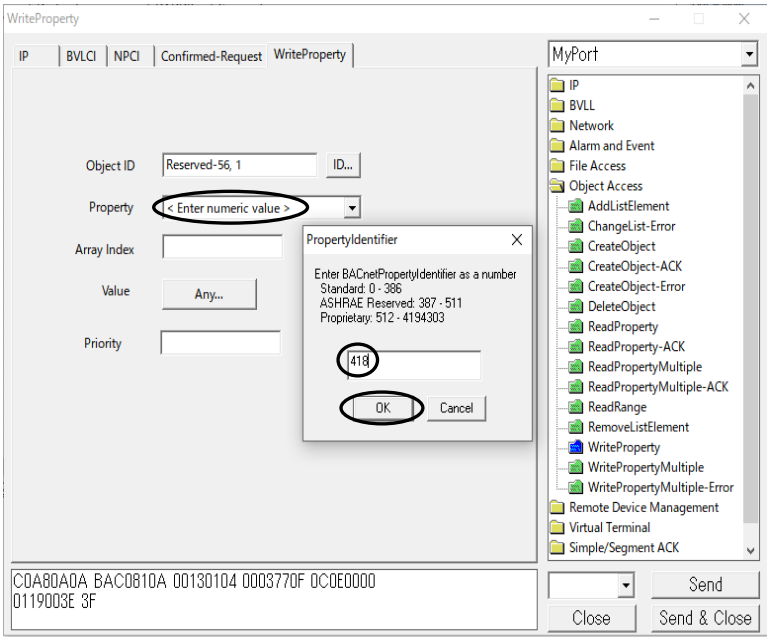

Fig. 5-14 WriteProperty dialog(2)

Set 418 (PROP_FD_BBMD_ADDRESS) in the PropertyIdentifier dialog, click OK, and then click "Any..." under Value.

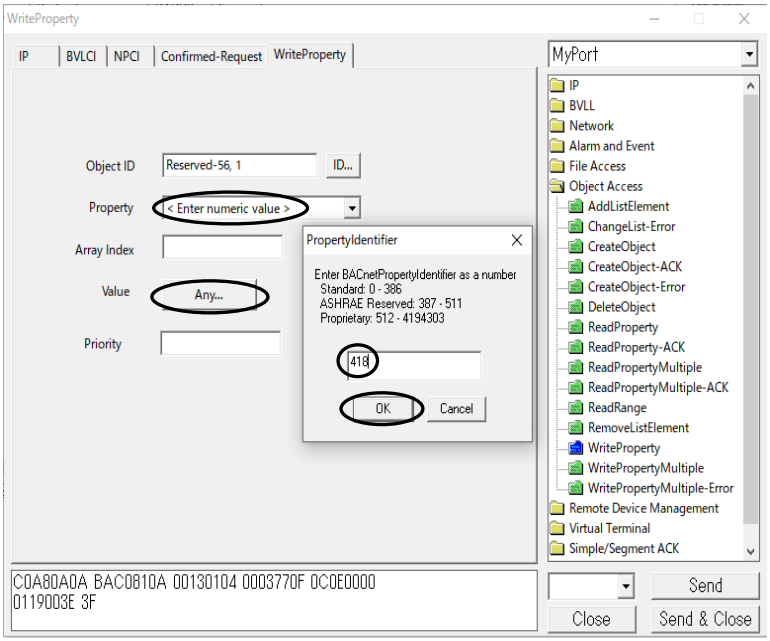

Fig. 5-15 WriteProperty dialog(2)

Click Add in the ABSTRACT-SYNTAX.&Type dialog, select Opening Tag for Type, and set Context to 0. Do not click OK yet.

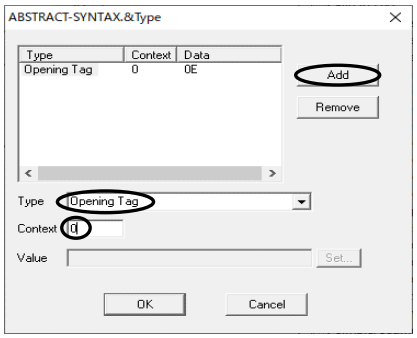

Fig. 5-16 ABSTRACT-SYNTAX.&Type dialog(1)

Click Add in the ABSTRACT-SYNTAX.&Type dialog again, select Octet String for Type, and set Context to 1. Set the IP address (in the example, enter C0A80A14 in Hex for 192.168.10.20 of the connecting PC) in Value. Do not click OK yet.

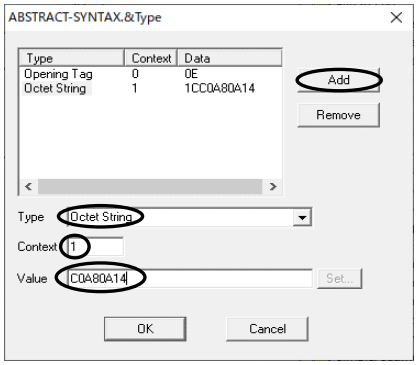

Fig. 5-17 ABSTRACT-SYNTAX.&Type dialog(2)

Click Add in the ABSTRACT-SYNTAX.&Type dialog again, select Closing Tag for Type, and set Context to 0. Do not click OK yet.

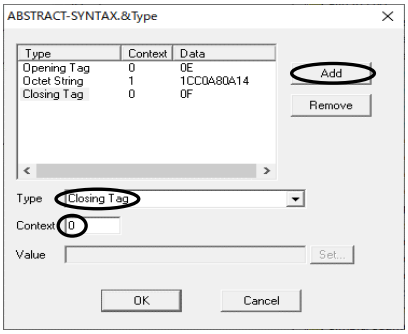

Fig. 5-18 ABSTRACT-SYNTAX.&Type dialog(3)

Click Add in the ABSTRACT-SYNTAX.&Type dialog again, select Unsigned for Type, set Context to 1, set Value to the UDP port number (decimal value of 47808 in the example), and click OK.

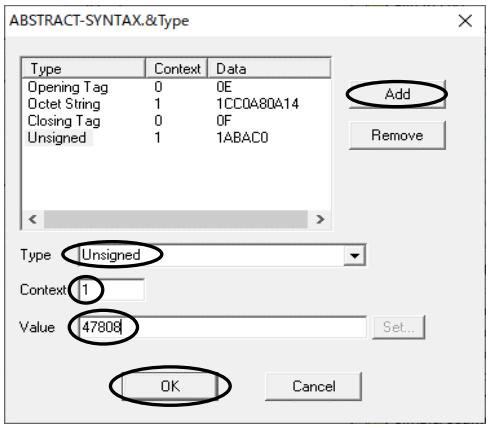

Fig. 5-19 ABSTRACT-SYNTAX.&Type dialog(4)

Finally, click Send on the WriteProperty dialog.

5.2.14 FD_Subscription_Lifetime

FD_Subscription_Lifetime property of the Network Port object is a BIP-specific property. It indicates the Time-To-Live value in seconds used in the Register-Foreign-Device BVLL message. Once this Time-To-Live value reaches 0, the B-BC resends the Register-Foreign-Device BVLL message.

This property is modified by the WriteProperty and WritePropertyMultiple services. Executing the WriteProperty and WritePropertyMultiple services write the setting value to Flash memory and set the Changes_Pending property to TRUE. The value becomes valid on reboot or when ReinitializeDevice service request is received with ACTIVATE_CHANGES or WARMSTART

There is no initial configuration command for this property. Refer to chapter 5.3 for details.

5.2.15 Password

For changing the Password, search the “Search word” shown in **Table 5-24** and change it in the source code.

Table 5-24 Password changes

Service	File to be changed	Search word
ReinitializeDevice	BACNETOSS\bacnet\basic\object\device.c	*Reinit_Password
DeviceCommunicationControl	BACNETOSS\bacnet\basic\service\h_dcc.c	My_Password[32]

5.2.16 OutOfService

When WriteProperty or WritePropertyMultiple services are executed for OutOfService property, the value is also written to Flash memory. Refer to chapter 5.3 for details.

5.3 Initial Configuration Command

5.3.1 Configurable Properties

There are some Properties in BACnet that are required to maintain their changed values even if device reboot occurs due to power failure etc. In this sample software, the Configurable Property shown in **Table 5-25** to **Table 5-27** is stored in Flash memory (QSPI0 Flash ROM) and the values are maintained after the B-BC device is rebooted.

These Configurable Properties can be set by executing the initial configuration commands via the serial interface. Connect the CN16 terminal where SCI0 peripheral modules are assigned to the PC with a USB cable, and execute the initial configuration commands from the terminal software.

(1) Configurable Properties

Configurable properties common to BIP and MSTP are listed in **Table 5-25**.

Table 5-25 Configurable Properties

No.	Configurable Property	Object type	Command	Num of arrays	Min value	Max value	Example value
1	DeviceName	Device	Name_of_device_obj =	64			RZN2L_BACnet_Solution
2	DeviceInstance		Instance_of_dev =		1	4194303	100
3	UTC_Offset		UTC_Offset =		-1440	1440	-540 (means TOKYO/JAPAN)
4	OutOfService	AnalogInput	OOS_AI_0(or 1) =		false	true	
5		AnalogValue	OOS_AV_0(or 1) =				
6		BinaryOutput	OOS_BO_0(or 1) =				
7		BinaryValue	OOS_BV_0(or 1) =				
8		Schedule	OOS_SC_0 =				
9		Multi-stateVale	OOS_MSV_0(or 1) =				
10		PositiveIntegerValue	OOS_PIV_0(or 1) =				

(2) BIP-specific Configurable Properties

In addition to the above, BIP-specific Configurable Properties are shown in **Table 5-26**.

No. 1 Ethernet_MAC is the MAC address of the Ethernet PHY; it is not a specified property in the BACnet standard, but can be changed with the initial configuration command without rebuilding.

No.2 The MAC_Address of BIP consists of the IP address and UDP port number. It can be changed with the initial configuration command without rebuilding.

No. 3 IP_Address and No. 4 BACnet_IP_UDP_Port are reflected from No. 2 MAC_Address setting value when rebooting, so there is no dedicated command.

The values of No.5 BACnet_IP_Mode, No.6 FD_BBMD_Address and No.7 FD_Subscription_Lifetime are written to the Flash ROM when the WriteProperty or WritePropertyMultiple services are executed for these properties. Therefore, there is no dedicated command, and each property is reflected at reboot and when the ReinitializeDevice service request is received as ACTIVATE_CHANGES or WARMSTART.

Table 5-26 BIP-specific configurable properties

No.	Configurable Property	Object type	Note	Command	Example value
1			Other than BACnet	Ethernet_mac_address =	74:90:50:10:05:B0
2	MAC_Address	NetworkPort		Bac_IP_mac_address =	192.168.10.10:47808

3	IP_Address		“MAC_Address” setting value is reflected.	no command	
4	BACnet_IP_UDP_Port				
5	BACnet_IP_Mode		Flash writing when executing WriteProperty and WritePropertyMultiple.		
6	FD_BBMD_Address				
7	FD_Subscription_Lifetime				

(3) MSTP-specific Configurable Properties

Table 5-27 shows the MSTP-specific Configurable Property.

No.1 MAC_Address of MS/TP can be changed by initial configuration command without rebuilding.

No.2 Network_Number is written to Flash memory when the WriteProperty or WritePropertyMultiple services are executed for the property. Therefore, there is no dedicated command, and the property is reflected on reboot or when a ReinitializeDevice service request is received with ACTIVATE_CHANGES or WARMSTART.

Table 5-27 MSTP-specific configurable properties

No.	Configurable Property	Object type	Note	Command	Min value	Max value	Example value
1	MAC_Address	NetworkPort		MSTP_mac_address =	128	254	129
2	Network_Number		Flash writing when executing WriteProperty and WritePropertyMultiple.	no command	0	65534	

5.3.2 Setup

Connect CN16 on the RSK board to the PC with a USB cable.

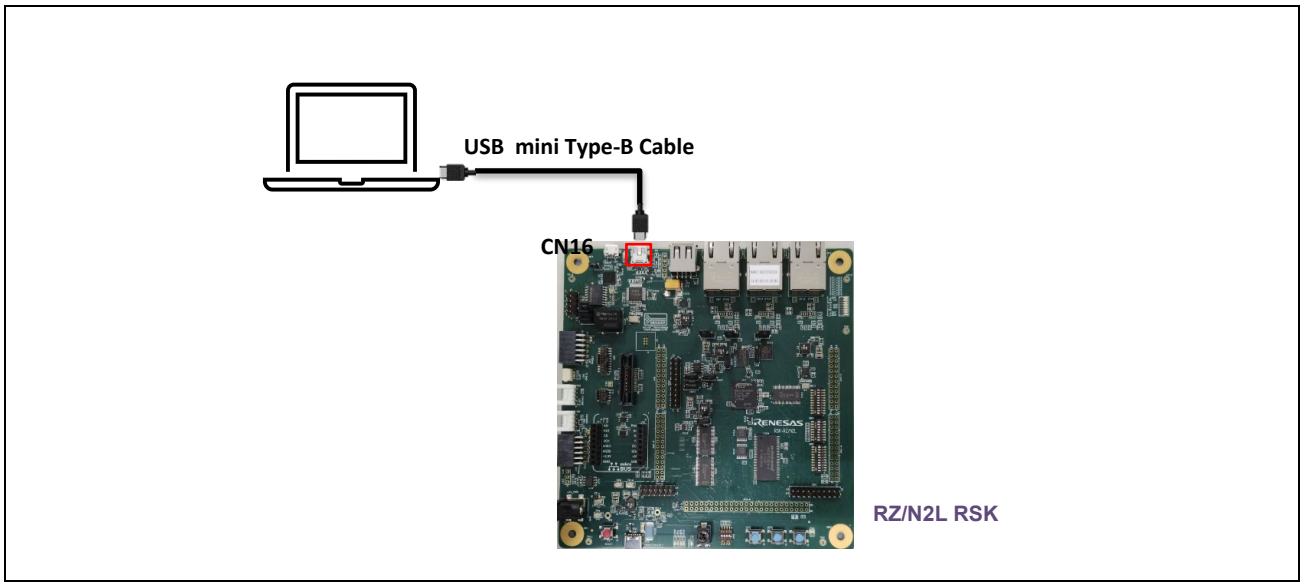

Fig. 5-20 Connect CN16 of the RSK to PC

Start the terminal software on your PC. Any terminal software will be suitable, but here TeraTerm is used as an example. It is available at the following link.

Releases · TeraTermProject/teraterm (github.com)

The serial port setup is as follows

Speed : 115200, Data : 8bit, Parity : none, Stop bits : 1bit, Flow control : none

The terminal setup is as follows

Append LF the transmitted data and Disable Local Echo.

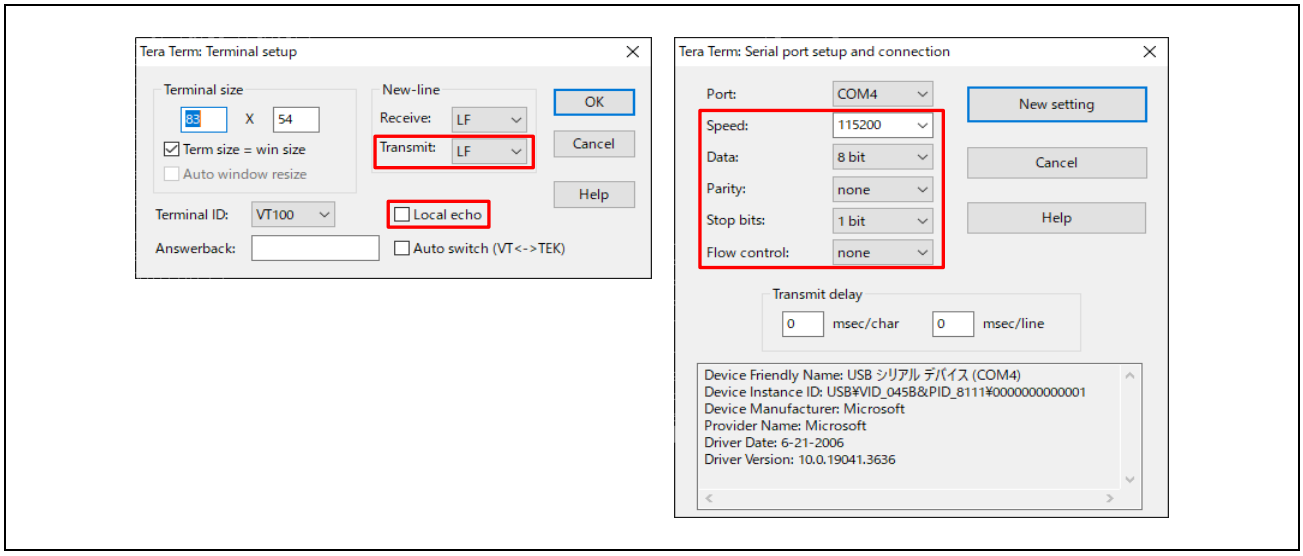

Fig. 5-21 Terminal software setup

5.3.3 Command Execution

When the RSK board is reset, it displays a memory dump of the end area of QSPI0 Flash ROM, where the configurable data (Configurable Property) is stored. Fig. 5-22 shows the case of BIP.

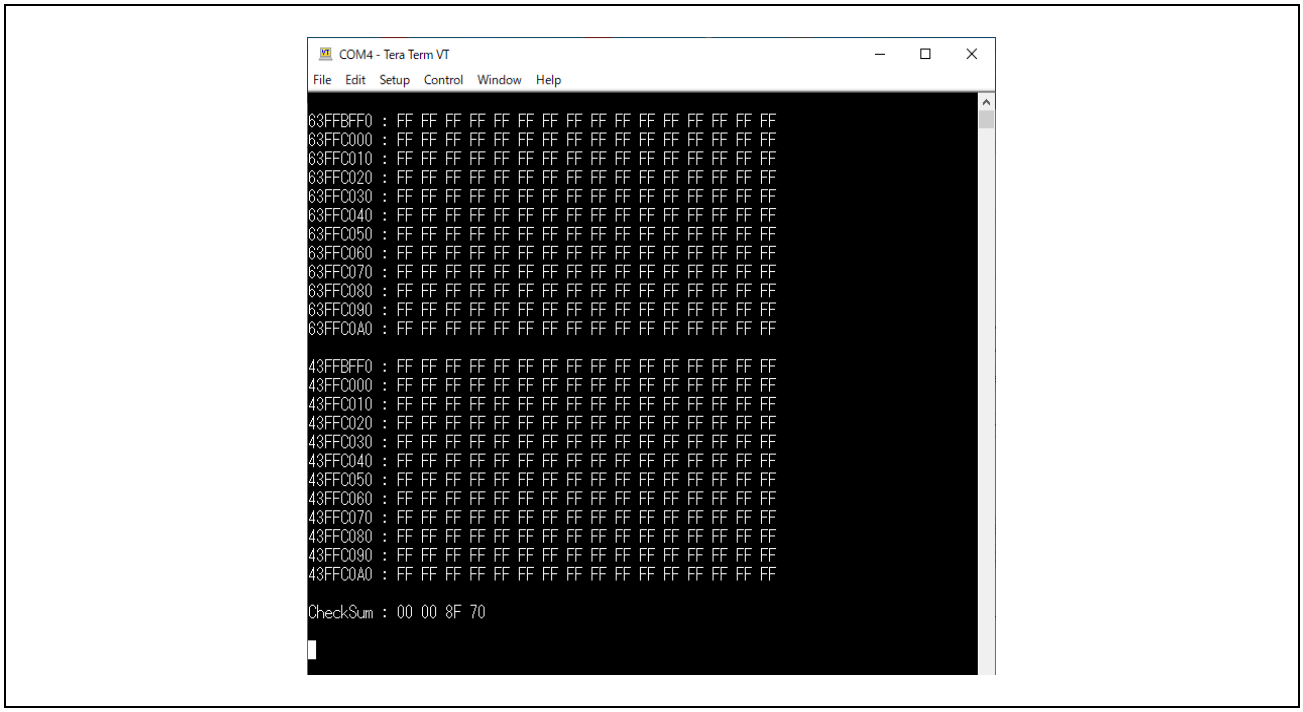

Fig. 5-22 Memory dump of data storage area

Upper part: 0x63FFC000 to 0x63FFC093: Data storage area of xSPI0_CS0_SPACE

Lower part: 0x43FFC000 to 0x43FFC093: Data storage area for xSPI0_CS0_SPACE_MIRROR

Continue typing any key, such as Enter, and the command format will appear.

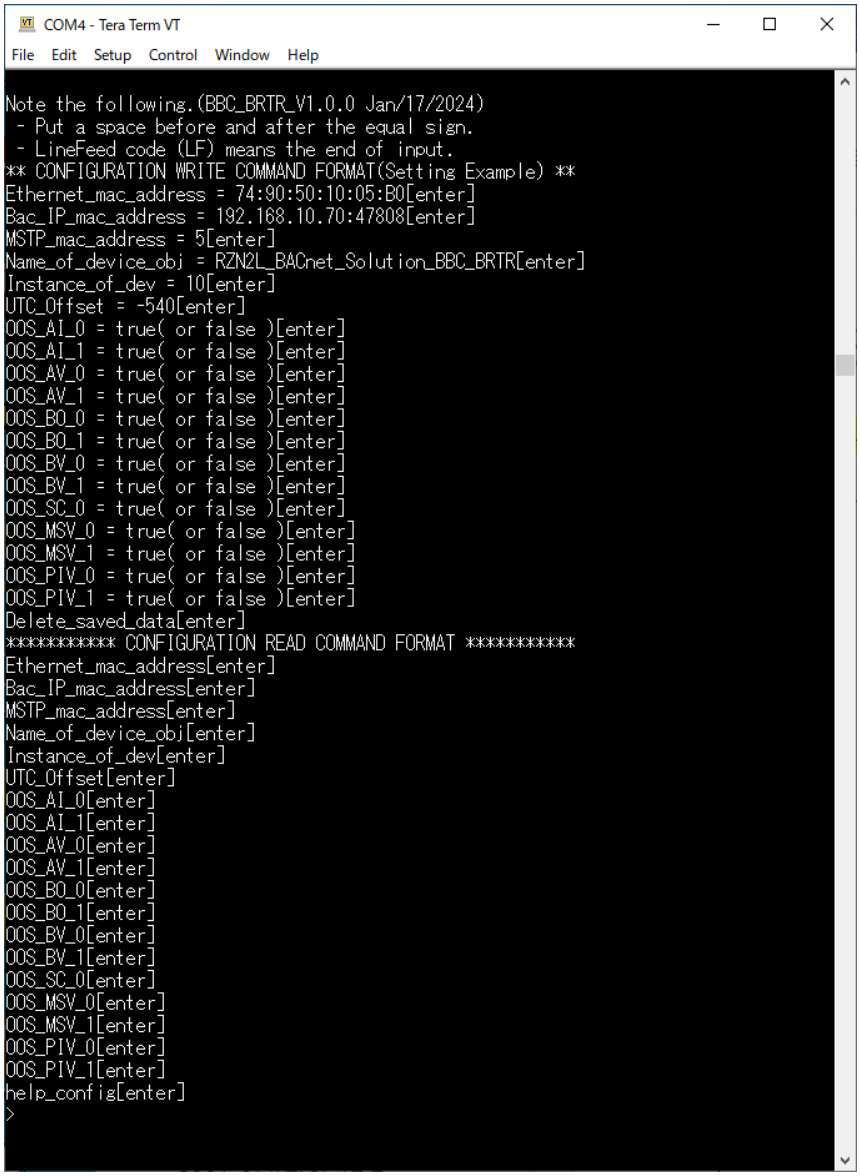

Fig. 5-23 Command format

**** CONFIGURATION WRITE COMMAND FORMAT(Setting Example) ****

Displays write command format. Setting values are shown as examples. (Chapter 5.3.3.1)

********** CONFIGURATION READ COMMAND FORMAT **********

Displays read command format. (Chapter 0)

5.3.3.1 Write Command

An example of write command is shown below. Commands must be entered in correct case, capitalization included. The written value becomes effective after the board is rebooted.

(1) BIP and MSTP Common Commands

Name_of_device_obj = RZN2L_BACnet_Solution_BBC_BRTR

This sets the object name (i.e., device name) of the device object.

Instance_of_dev = 100

This sets the instance number of the device object.

UTC_Offset = -540

This sets UTC_Offset. This UTC offset (-540) indicates TOKYO/JAPAN (-9hour x 60min), with a minus value east of the meridian and a plus value west of the meridian. It is not necessary to enter a plus sign.

For example, for VANCOUVER/CANADA, UTC_Offset = 480.

OOS_AI_0 = true(or false)

This sets the OutOfService property value of the AnalogInput,0 object.

... Omitted

OOS_SC_0 = true(or false)

This sets the OutOfService property value of the Schedule,0 object.

... Omitted hereafter, see No. 4...10 in **Table 5-25**.

Delete_saved_data

This command clears (erases) the stored value to 0xFF.

(2) BIP-Specific Commands

Ethernet_mac_address = 74:90:50:10:05:B0

This sets the MAC address of the Ethernet PHY on the RSK board.

Bac_IP_mac_address = 192.168.10.10:47808

This sets the IP address and UDP port number of the B-BC.

(3) MSTP-Specific Commands

MSTP_mac_address = 5

This sets the MAC address for MS/TP master of the B-BC.

After the write commands are executed, reset the board. the settings data are saved as shown in Fig. 5-24. the data saved in Flash ROM are used as the initial values of each property.

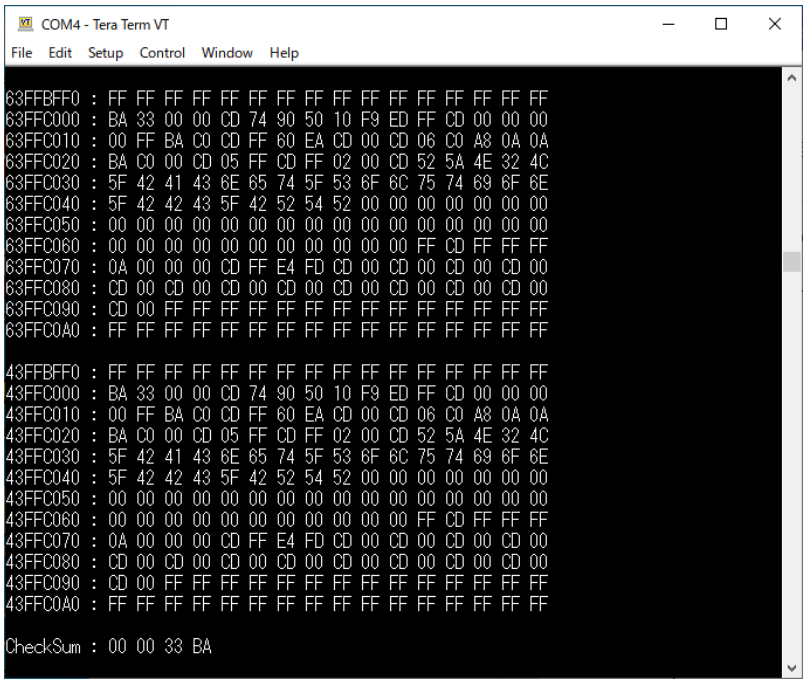

Fig. 5-24 Memory dump after writing

5.3.3.2 Read Command

Fig. 5-25 shows an example of executing each Read command. Execute a read command like ">Ethernet_mac_address" then the written value (74:90:50:10:05:B0) will be displayed.

“help_config” is help command to redisplay the command format.

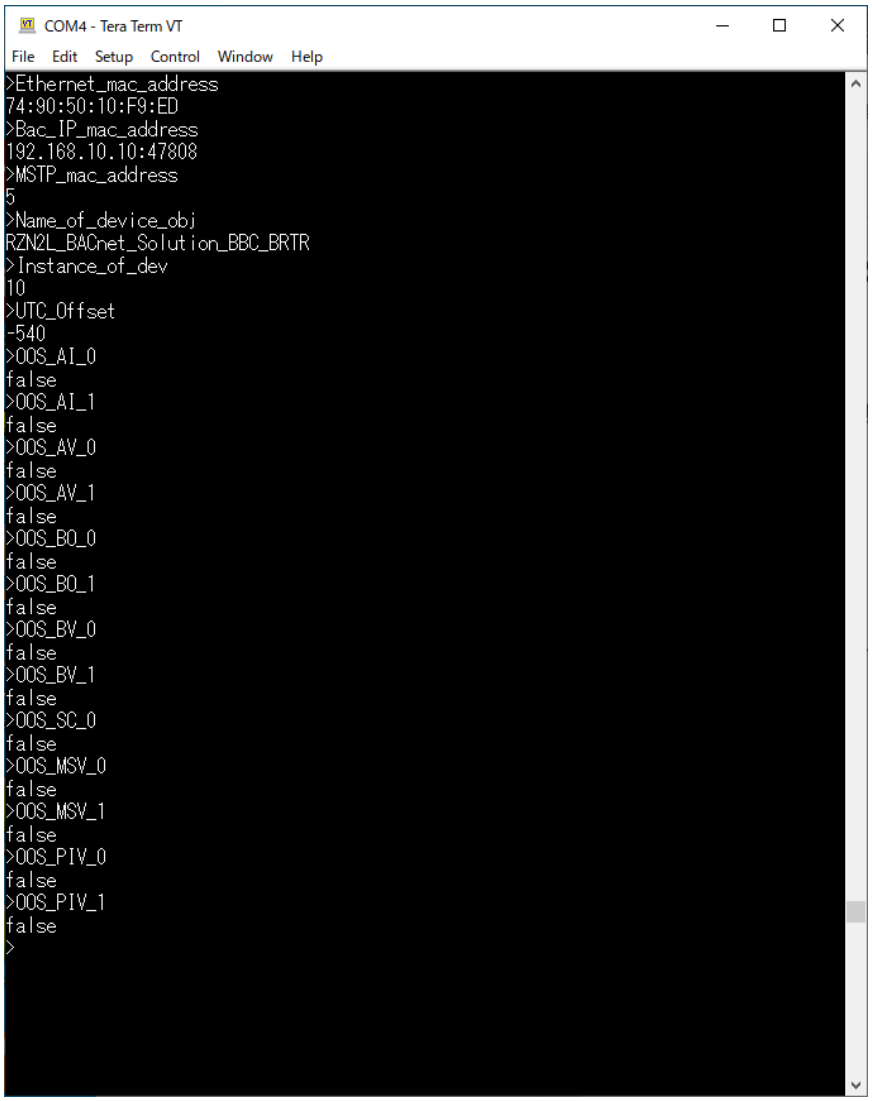

Fig. 5-25 Read command result

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar/25/2024	-	First Edition

Trademark

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

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/

Trademarks

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