
RZ/N2L グループ

BACnet Controller Sample Software

R01AN7237JJ0100
Rev.1.00
2024.03.25

要旨

本書は、ビルディングオートメーション(BA)向け通信プロトコル BACnet のコントローラプロファイル (B-BC)を RZ/N2L で実行するためのサンプルソフトウェアについて説明します。

動作確認デバイス

RZ/N2L Group

目次

1. 概要	6
1.1 概要	6
1.2 動作環境	8
1.2.1 ソフトウェア環境	8
1.2.2 ハードウェア環境	8
2. ハードウェア構成	9
2.1 RSK ボード設定	9
3. サンプルソフト構成	12
3.1 フォルダ構成	12
3.2 ブートシーケンス	13
3.3 BACnet Stack	16
3.3.1 BACnet Protocol Stack	16
3.3.2 ライセンス	16
3.3.3 仕様、サポート機能	17
3.3.3.1 BACnet Revision	17
3.3.3.2 サービス	17
3.3.3.3 制限事項	21
3.3.3.4 BIBBs	22
3.3.3.5 A デバイスのサポートサービス	25
3.3.3.6 オブジェクト	25
3.3.3.7 プロパティ	28
3.4 開発環境構築	34
3.4.1 統合開発環境 e2studio	34
3.4.1.1 インストール	34
3.4.1.2 プロジェクト立ち上げ	39
3.4.2 VTS	43
3.4.3 Yabe	43
3.4.4 Wireshark	44
3.4.5 Terminal software	44
4. 動作確認	45
4.1 接続	45
4.2 BACnet クライアント IP アドレス設定	45
4.3 Wireshark 設定	47
4.4 プロジェクト起動	49
4.4.1 ビルド設定の注意事項	49
4.4.1.1 変更禁止 Symbol	49
4.4.1.2 NTP サーバーアドレスの設定	51
4.4.2 ビルド	52
4.4.3 Debug Configurations 設定	53
4.4.4 デバッグ	56
4.5 VTS による通信確認	59
4.5.1 Who-Is と I-Am	62

4.5.2	ReadProperty	67
4.5.3	TimeSynchronization / UTCTimeSynchronization.....	72
4.5.4	Who-Has と I-Have.....	74
4.5.5	ReadPropertyMultiple.....	76
4.5.6	WriteProperty	78
4.5.7	WritePropertyMultiple	81
4.5.8	SubscribeCOV.....	86
4.5.9	ReinitializeDevice	89
4.5.10	DeviceCommunicationControl.....	91
4.5.11	AtomicReadFile	93
4.5.12	AtomicWriteFile	95
4.6	Yabe による通信確認	97
4.6.1	TimeSynchronization / UTCTimeSynchronization.....	100
4.6.2	B-BC からの B-SS 制御.....	102
4.6.2.1	Trending & ReadRange.....	102
4.6.2.2	Scheduling.....	106
4.6.3	EventNotification / GetEventInformation / AcknowledgeAlarm	108
4.6.4	AtomicReadFile	113
4.6.5	AtomicWriteFile	115
4.6.6	ReinitializeDevice	116
5.	初期設定	117
5.1	初期値.....	117
5.1.1	Ethernet MAC address(IP)	117
5.1.2	Device.....	118
5.1.3	Analog Input	120
5.1.4	Analog Value	122
5.1.5	Binary Output	124
5.1.6	Binary Value	125
5.1.7	File.....	127
5.1.8	Notification Class.....	127
5.1.9	Schedule.....	128
5.1.10	Multi State Value	129
5.1.11	Trend Log	130
5.1.12	Positive Integer Value	131
5.1.13	Network Port.....	132
5.1.14	Password.....	133
5.2	初期値変更.....	134
5.2.1	Ethernet MAC address(IP)	134
5.2.2	Device instance	137
5.2.3	Device name	137
5.2.4	Number of objects	137
5.2.5	UTC_Offset.....	138
5.2.6	Number of states	139
5.2.7	State text	139
5.2.8	Network number	139
5.2.9	Link speed	140

5.2.10	MAC address.....	141
5.2.11	BACnet IP address.....	141
5.2.12	BACnet_IP_Mode.....	141
5.2.13	FD_BBMD_Address.....	141
5.2.14	FD_Subscription_Lifetime.....	145
5.2.15	Password.....	145
5.2.16	OutOfService.....	145
5.3	初期設定コマンド.....	146
5.3.2	セットアップ.....	148
5.3.3	コマンド実行方法.....	149
5.3.3.1	Write Command.....	151
5.3.3.2	Read Command.....	153

用語解説

本書で使用する用語は、以下に示すように定義して使用します。

用語	説明
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

関連文書

資料名	資料番号
RZ/N2L データシート	R01DS0397JJ****
RZ/N2Lグループ ユーザーズマニュアル ハードウェア編	R01UH0955JJ****
Renesas Starter Kit+ for RZ/N2L ユーザーズマニュアル	R20UT4984JG****
RZ/N2L Group TCP/IP lwIP Sample Program Package	R01AN6588EJ****
RZ/N2L Group BACnet Sample Software	R01AN6789JJ****

1. 概要

1.1 概要

BACnetはビルディングオートメーション(BA)用の主要な通信プロトコルです。

本書は、産業ネットワーク用 RZ プロセッサ RZ/N2L において、異なるネットワーク環境下にある BIP(BACnet/IP)デバイスと MS/TP デバイスの相互運用を実現する BACnet ルータ機能(B-RTR)を持つ BACnet コントローラ(B-BC)のサンプルソフトについて、その構成、およびその使い方について説明します。



Fig. 1-1 RSK+ for RZ/N2L

尚、本書をご参照いただくうえで不都合の無い箇所については他の BACnet デバイスアプリケーションノートで使用していた図を一部流用しております。ご了承下さい。

本書で説明するサンプルソフトは、BACnet Router という機器同士を繋ぐインターフェースであることから、本書での動作確認は Fig. 1-2 のように BACnet Client として PC 上のアプリケーションを用い、対向の MS/TP スレーブとして、[RZ/N2L BACnet Sample Software \(R01AN6789xJ****\)](#) を使用します。

説明の都合上、本書で説明する BACnet Router を B-BC、対向の BACnet スレーブを B-SS と呼びます。

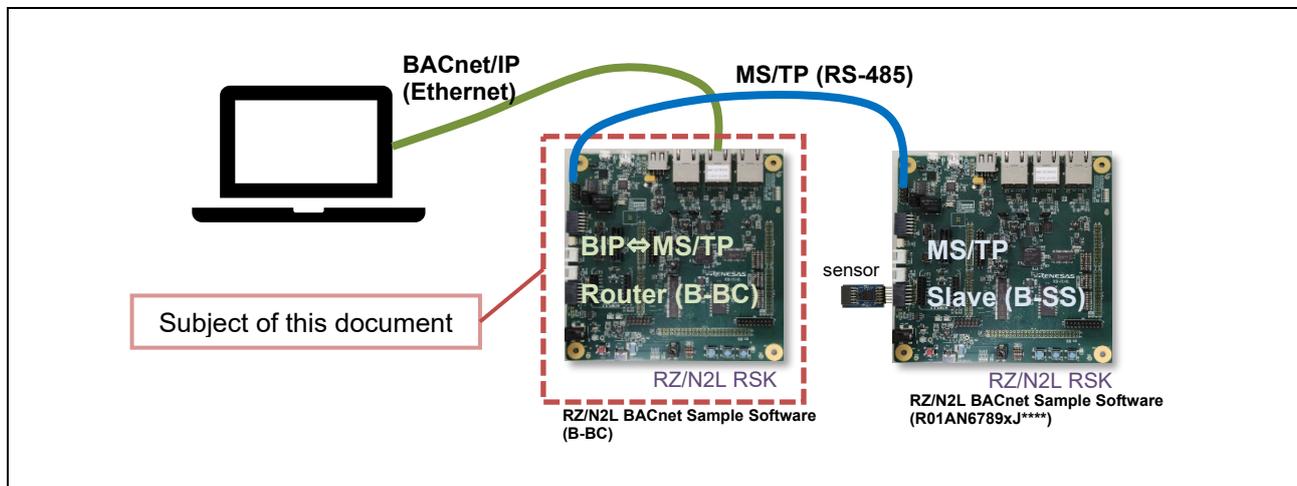


Fig. 1-2 Subject of this document and test setup

1.2 動作環境

1.2.1 ソフトウェア環境

本サンプルソフトの動作環境を Table 1-1 に示します。

Table 1-1 Operating Environment

Category	Name	Version	Link	備考
RZ/N2L BACnet サンプルソフト	サンプルパッケージ			
統合開発環境	e2studio	23.7.0	https://github.com/renesas/rzn-fsp/releases/download/v1.3.0/setup_rznfsp_v1_3_0_e2s_v2023-07.exe	e2studio インストーラーに同梱
Flexible Software Package	FSP	1.3.0		e2studio インストーラーに同梱
GNU Arm Embedded Toolchain	GCC Toolchain	V9.3.1.20200408 (注1)		e2studio インストーラーに同梱
BACnet/IP クライアントツール	VTS	3.6.7.0	Visual Test Shell for BACnet download SourceForge.net	
BACnet/MSTP マスタツール	Yabe	1.3.0.0	Yet Another Bacnet Explorer download SourceForge.net	
Packet analyzer	Wireshark	4.0.3	Wireshark · Download	
MS/TP キャプチャツール	mstpcap.exe		Capturing MS/TP packets – Optigo Networks (zendesk.com)	Integration with Wireshark.
ターミナルソフト	TeraTerm	4.108	Releases · TeraTermProject/teraterm (github.com)	

注1. FSP v1.3.0 の GCC Toolchain 推奨バージョンは v12.2.1.arm-12-24 ですが、本サンプルソフトでは V9.3.1.20200408 にて動作確認を行っております。

1.2.2 ハードウェア環境

本サンプルソフトは Table 1-2 のハードウェア環境にて、動作確認を行っております。

Table 1-2 ハードウェア環境

Name	Type Name	Maker	Link	Note
Renesas Starter Kit+ for RZ/N2L	RTK9RZN2L0S00 000BE	Renesas Electronics	www.renesas.com/rskrzn2l	RSK ボード
空気速度センサ Pmod™ ボード	US082-FS3000EVZ	Renesas Electronics	US082-FS3000EVZ - 空気速度センサ Pmod™ ボード (ルネサス クイックコネク ト IoT) Renesas	ルネサス クイックコネク ト IoT
USB/RS485 変換器	BOB-09822	SparkFun	SparkFun USB to RS-485 Converter - BOB-09822 - SparkFun Electronics	2pcs (Yabe 及び Wireshark 用)

2. ハードウェア構成

本サンプルソフトを実行するハードウェア構成について説明します。

2.1 RSK ボード設定

本サンプルソフトウェアを実行するときの、RSK ボード設定を Fig. 2-1 に示します。

- ブートモードは xSPI0 ブートモードを使用
- BACnet MS/TP 通信は RS-485 半 2 重通信モードを選択

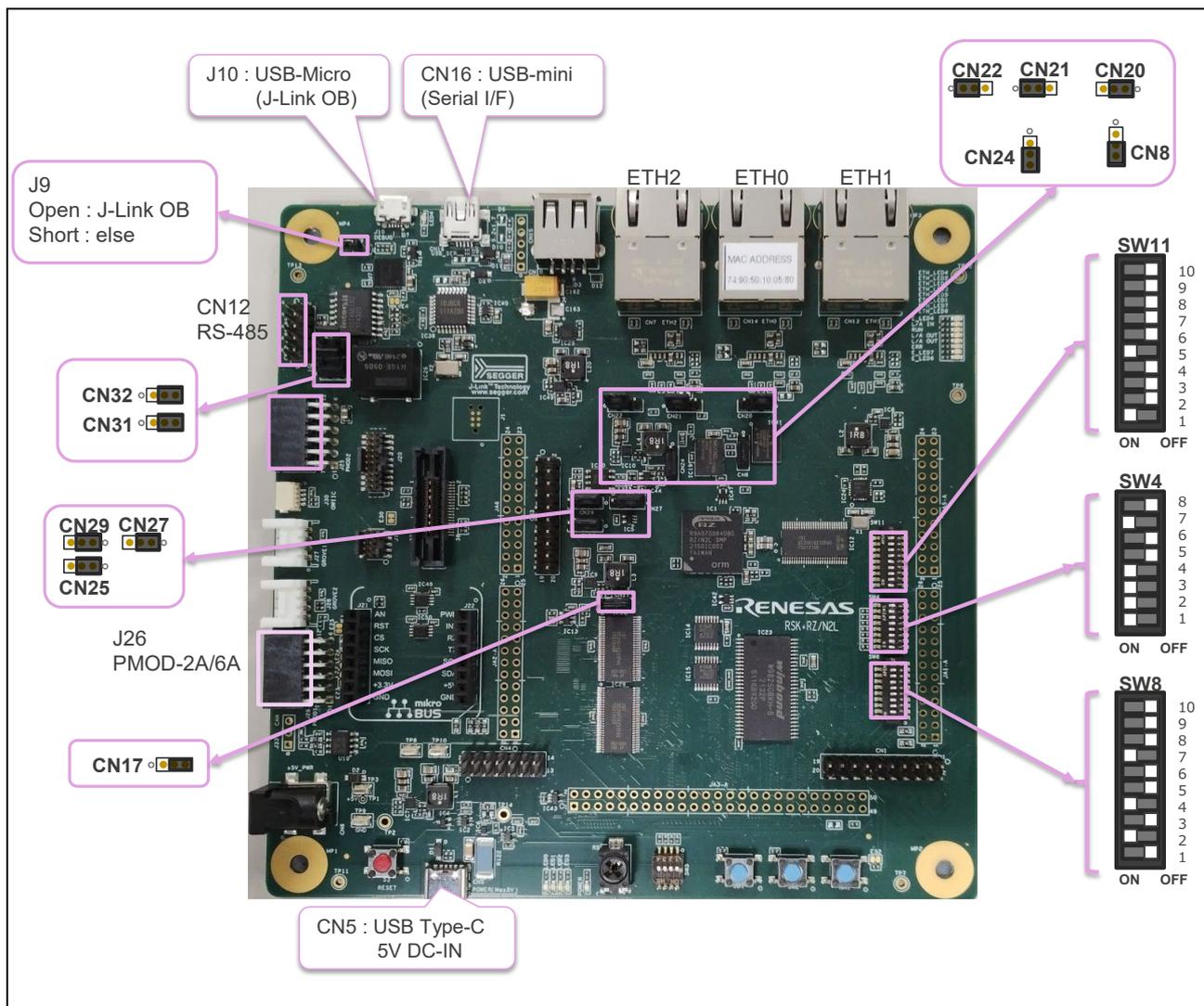


Fig. 2-1 Board Configuration

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. サンプルソフト構成

本章では、サンプルソフトについて説明します。

尚、FSP バージョンについて以降に掲載の図において FSP v1.1.0 となっている個所がありますが、v1.3.0 に読み替えてください。

3.1 フォルダ構成

本サンプルソフトのフォルダ構成を以下に示します。太字は目安として、本サンプルソフトをカスタマイズするときに必要なファイルが含まれるフォルダを示しています。

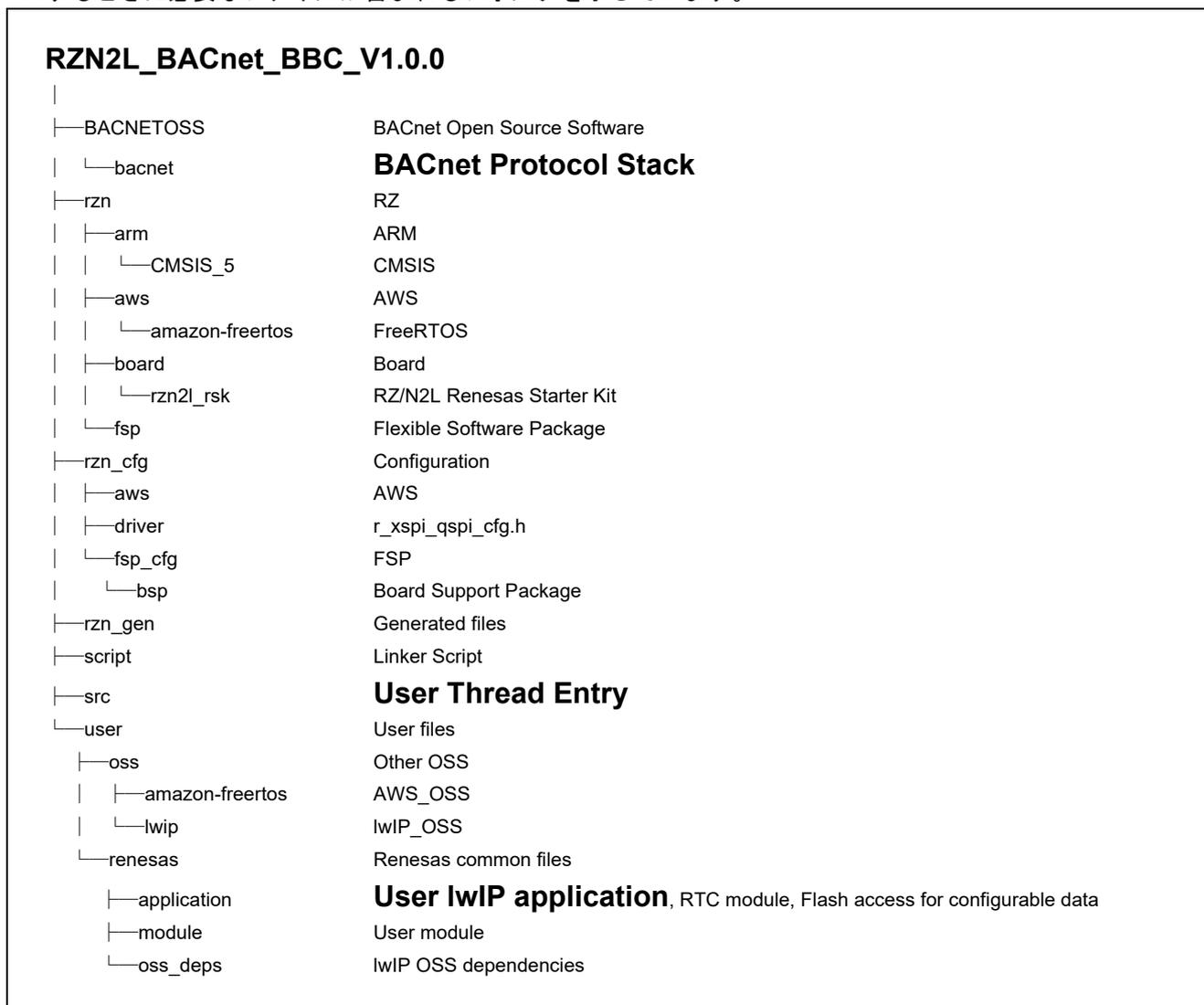


Fig.3-1 Folder Structure

3.2 ブートシーケンス

ブート手順とメモリ配置について説明します。

本サンプルソフトは(xSPI0 x1 boot mode)です。以下の図はスマート・コンフィグレータの BSP タグを表示したものです。

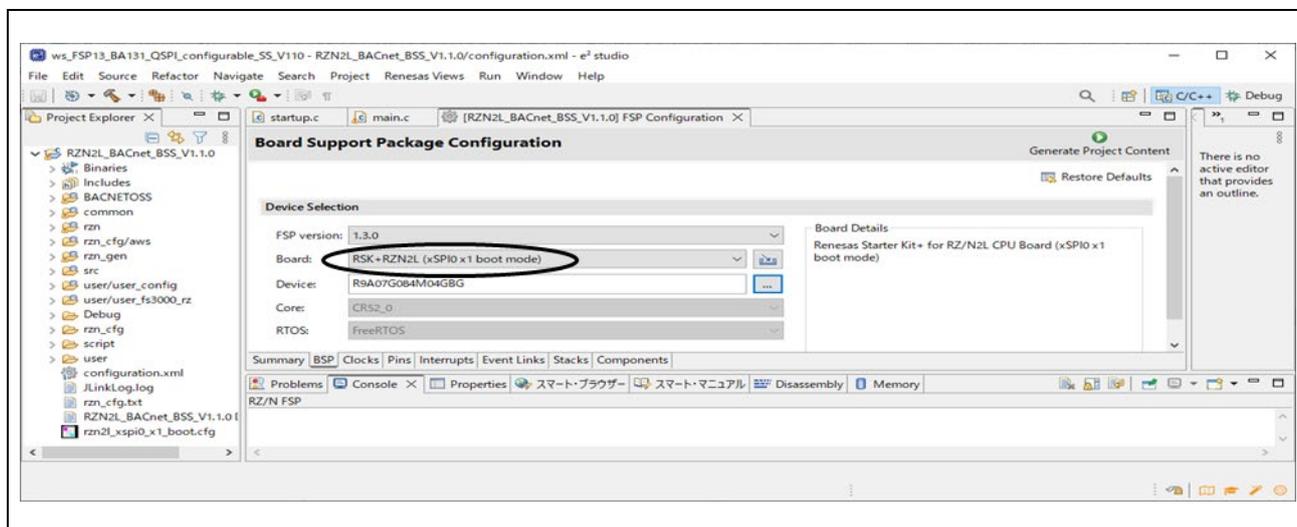


Fig.3-2 Boot mode

フラッシュメモリデバイスにダウンロード後はデバッグ接続なしの状態です。RSK ボード上の RESET ボタンを押下または電源 OFF/ON を行うとボード単独で動作します。引き続きデバッグを接続して評価を行うこともできます。ただし、RSK ボードの J9 をショートするとデバッグ(J-Link OB)は接続できません。

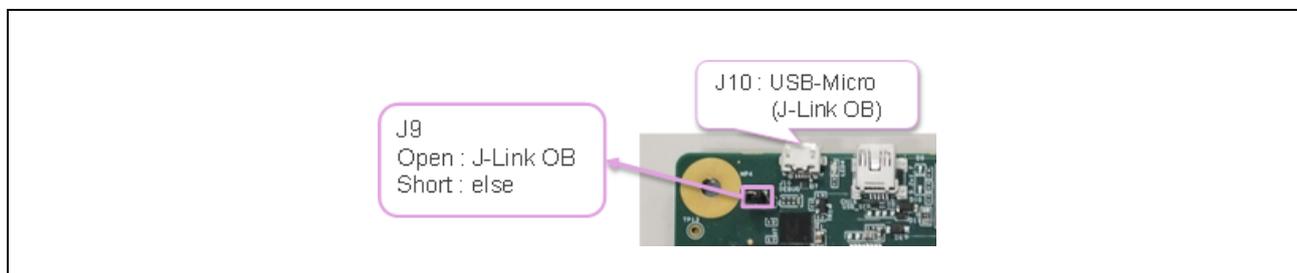


Fig.3-3 J9

シリアルフラッシュメモリデバイスの端子設定(Pins タグ)を示すスマート・コンフィグレータ画面です。設定済みのため変更は不要です。

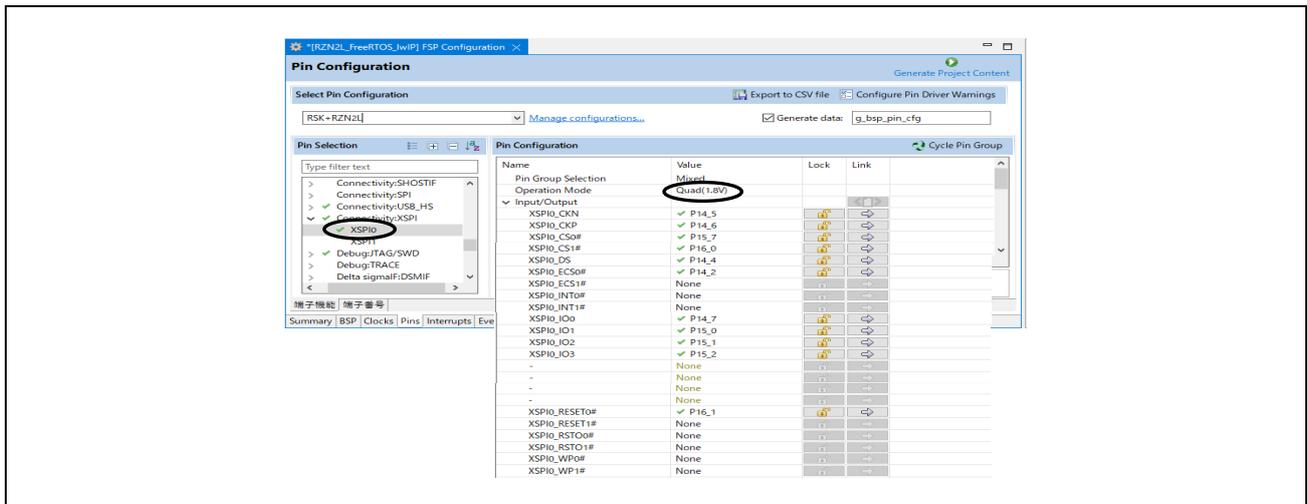


Fig.3-4 XSPI0 Pin Configuration

以下のメモリ配置表の writing order 欄にブートシーケンスにおけるメモリ書き込み順番を示しています。(1)から(4)の順番で書き込みを行います。ただし(5)はブートシーケンスとは関係なくデバイスの再起動後に維持したいデータの保存領域です。そのため、システムの稼働中に書き込みが行われます。詳細については5.3章を参照ください。

Address	Memory	Content	Length	writing order	remarks	
0x00000000	ATCM	intvec(64B)	0x00020000	128KB	(3)	
0x00000040		Unused				
0x00000100		hal_entry_ROMdata				
0x00020000	Reserved area	-	-	-	-	
0x00100000	BTCM	Unused	0x00020000	128KB	(2)	
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)	
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)
0x6000004C		Loader program(24KB)				
0x6000604C		Body of program and data				
0x60FFFE00		Reserved area				
0x63FF8000		Unused				
0x63FFC000		Configurable properties				
0x63FFC092	Unused				(5)	
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) シリアルフラッシュ書き込み

e2studio はダウンロードを行うと xSPI0_CS0_SPACE メモリに割り当てた拡張シリアルペリフェラルインタフェース(xSPI)を持つシリアルフラッシュメモリデバイスに書き込みます。

ダウンロードデータには、ローダーパラメータ、ローダープログラムとそのデータ、ユーザープログラム本体とそのデータが含まれます。これらのメモリへの割り当てはリンクスクリプト(拡張子.ld)ファイル `fsp_xspi0_boot.ld` で指定します。

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

(2) ローダープログラム展開

CPU はダウンロードデータに含まれるローダープログラムを自動的に BTCM メモリに展開します。展開後、ローダープログラム上の初期設定プログラム先頭にある `system_init()` で Break します。

(3) ATCM 展開

ローダープログラムの初期設定プログラムは ATCM メモリに割り当てられたユーザーデータをフラッシュメモリから ATCM メモリに展開します。

(4) SYSTEM_RAM_MIRROR 展開

ローダープログラムの初期設定プログラムは SYSTEM_RAM_MIRROR メモリに割り当てられたユーザープログラムをフラッシュメモリから SYSTEM_RAM_MIRROR メモリに展開します。

3.3 BACnet Stack

BACnet (Building Automation and Control Network) は、ASHRAE/ANSI Standard 135 で規格化されている主要な BA (Building Automation) 通信プロトコルです。空調、照明、防災、アクセス制御など統合してビルの制御、監視を行うことができます。

BACnet デバイスはオペレータやコントローラなど、機能・用途に応じて様々なプロファイルに分類されます。主要なプロファイルとしては、中央監視プロファイル B-OWS (BACnet Operator Workstation)、コントローラプロファイル B-BC (BACnet Building Controller)、各種センサー用プロファイル B-SS (BACnet Smart Sensor) などがあります。

BACnet 規格では、プロファイル毎に対応すべき標準的な機能ブロック(BIBBs)が定義されていますが、そこに含まれない BIBBs に対応することも許容されています。

本書では、本サンプルソフト(RZN2L_BACnet_BBC_V1.0.0)を B-BC と呼んでおりますが、プロファイルとしては B-RTR と B-BC を併設しています。

B-RTR プロファイルの機能としては、BACnet/IP ネットワークに接続された BACnet クライアントが B-BC(B-RTR プロファイル)を経由することで MS/TP ネットワークに接続された B-SS をアクセスすることを可能にしています。このとき B-BC(B-RTR プロファイル)は B-SS に対して MS/TP マスターとして機能しません。

B-BC プロファイルの機能としては、BACnet クライアントが B-BC(B-BC プロファイル)に対して、B-SS のセンサ入力値をロギングするように指示したり、B-SS の LED 点消灯を任意の曜日時刻にスケジュールさせることができます。また、B-BC(B-BC プロファイル)は B-OWS など上位の BACnet クライアントに対しては BACnet サーバーとして機能します。

3.3.1 BACnet Protocol Stack

BACnet Protocol Stack (bacnet-stack) は BACnet 通信プロトコルのオープンソーススタックです。本サンプルソフトは、BACnet Protocol Stack を RZ/N2L に移植しています。

Base Version : bacnet-stack-1.3.1

[Tags · bacnet-stack/bacnet-stack · GitHub](#)

3.3.2 ライセンス

BACnet Protocol Stack ライセンス条件に GPL with exception license となっています。

参考として原文を以下に転記いたします。詳細は、[BACnet Protocol Stack download | SourceForge.net](#) をご確認の上、ライセンス条件を遵守してご利用ください。

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 仕様、サポート機能

3.3.3.1 BACnet Revision

本サンプルソフトで使用する BACnet スタックの BACnet 規格プロトコルバージョン、およびリビジョンは以下です。

- ・ BACnet standard Protocol Version : 1
- ・ BACnet standard Protocol Revision : 23

3.3.3.2 サービス

本サンプルソフトが実装する BACnet スタックはサービス主導型です。BACnet デバイスの相互運用 (Interoperability) は、サービス (Who-Is、I-Am、ReadProperty など) を介してユーザーとプロバイダー間が接続されることで行われます。

サービスは確認なし (Unconfirmed) 型と確認あり (Confirmed) 型があります。確認なし型はユーザーが要求したサービスに対してプロバイダーは Ack を返しません。一方の確認あり型は Ack を返します。

本サンプルソフトのユーザーとは以下を指します。

BACnet/IP プロトコルを使って相互接続する BACnet デバイスの場合はクライアントに該当します。

BACnet MS/TP プロトコルの場合はマスターに該当します。

プロバイダーとは以下を指します。

BACnet/IP プロトコルを使って相互接続する BACnet デバイスの場合はサーバーに該当します。

BACnet MS/TP プロトコルの場合はスレーブに該当します。

本サンプルソフトで動作する B-BC は BACnet クライアントに対してはサーバー (プロバイダー) であり、B-SS スレーブに対してはマスター (ユーザー) です。

本サンプルソフトが実装するサービスを Table 3-1 に示します。(✓は該当、空欄は非該当)

Table 3-1 実装サービス

BACnet サービス	Initiate ¹	Execute ²
Who-Is	✓(要求)	✓
I-Am	✓(通知)	✓
Who-Has		✓
I-Have	✓(通知)	
ReadProperty	✓(要求) ³	✓
WriteProperty	✓(要求) ³	✓
DeviceCommunicationControl		✓
ReinitializeDevice		✓
AtomicReadFile		✓
AtomicWriteFile		✓
TimeSynchronization		✓
UTCTimeSynchronization		✓

BACnet サービス	Initiate ¹	Execute ²
SubscribeCOV		✓
ConfirmedCOVNotification	✓(通知)	
UnconfirmedCOVNotification	✓(通知)	
ReadPropertyMultiple	✓(要求) ³	✓
ReadPropertyConditional		
ReadRange		✓
WritePropertyMultiple	✓(要求) ³	✓
GetAlarmSummary		✓
GetEventInformation		✓
GetEnrollmentSummary		
AcknowledgeAlarm		✓
ConfirmedEventNotification	✓(通知)	
UnconfirmedEventNotification	✓(通知)	
UnconfirmedTextMessage		
ConfirmedTextMessage		
AddListElement		
RemoveListElement		
CreateObject		
DeleteObject		
UnconfirmedPrivateTransfer		
ConfirmedPrivateTransfer		
VTOpen		
VTData		
VTClose		

✓は該当、空欄は非該当

1. BACnet サービス要求または通知を送信します。
2. BACnet サービスを実行し、応答(ただし確認あり型サービスを要求された場合)を送信します。
3. B-SS へのサービス要求ですが ReadPropertyMultiple と WritePropertyMultiple は未使用です。

実装サービスの概要は次のとおりです。

Table 3-2 実装サービス概要

BACnet サービス	概要
Who-Is	Who-Is サービスは、ネットワークを共有する他の BACnet デバイスを知るために、BACnet ユーザーによって使用されます。Who-Is サービスはブロードキャスト送信され、確認なし型(Ack を要求しない)サービスです。
I-Am	I-Am サービスは、Who-Is サービス要求に応答するために使用されます。ただし、I-Am サービス要求はいつでも発行可能なブロードキャスト送信です。Who-Is サービス要求の受信が先行する必要はありません。

BACnet サービス	概要
Who-Has	Who-Has サービスは、BACnet ユーザーが、特定のオブジェクトを持つ BACnet デバイスを識別するために使用されます。Who-Has サービスはブロードキャスト送信され、確認なし型のサービスです。
I-Have	I-Have サービスは、Who-Has サービス要求に応答するために使用されます。ただし、I-Have サービス要求はいつでも発行できます。Who-Has サービス要求の受信が先行する必要はありません。I-Have サービスはブロードキャスト送信され、確認なし型のサービスです。
ReadProperty	ReadProperty サービスは、BACnet ユーザーが 1 つの BACnet オブジェクトの 1 つのプロパティの値を要求するために使用されます。BACnet プロバイダーは Ack を応答して結果を返します。
WriteProperty	WriteProperty サービスは、BACnet オブジェクトの 1 つの指定されたプロパティの値を変更するために、BACnet ユーザーによって使用されます。BACnet プロバイダーは Ack を応答します。指定されたプロパティへの書き込みアクセスを制限したい場合、「エラークラス」PROPERTY および「エラーコード」WRITE_ACCESS_DENIED のエラーが返されます。
WritePropertyMultiple	WritePropertyMultiple サービスは、1 つまたは複数の BACnet オブジェクトの 1 つまたは複数の指定されたプロパティの値を設定するために、BACnet ユーザーによって使用されます。BACnet プロバイダーは Ack を応答します。BACnet ユーザーは、任意の数のオブジェクトの任意の数のプロパティを書き込むことができます。
DeviceCommunicationControl	DeviceCommunicationControl サービスは、BACnet ユーザーが他の BACnet デバイスに、BACnet サービスの開始を指定された期間停止するように指示するために使用されます。期間は「無期限」に設定できます。BACnet プロバイダーは Ack を応答します。期間を「無期限」にした場合は DeviceCommunicationControl または ReinitializeDevice サービスによって通信を再度有効にする必要があります。
ReinitializeDevice	ReinitializeDevice サービスは、BACnet ユーザーによって使用され、他の BACnet デバイスに再起動を指示します。BACnet プロバイダーは Ack を応答します。
TimeSynchronization	TimeSynchronization サービスは BACnet ユーザーによって使用され、他の BACnet デバイスに現在時刻をブロードキャストまたはユニキャストで通知して、デバイスの時計を相互に同期できるようにします。このサービスは確認なし型のため BACnet プロバイダーは Ack を応答しません。
UTCTimeSynchronization	UTCTimeSynchronization サービスは BACnet ユーザーによって使用され、他の BACnet デバイスに子午線を基準とした UTC 現在時刻をブロードキャストまたはユニキャストで通知して、デバイスの時計を相互に同期できるようにします。このサービスは確認なし型のため BACnet プロバイダーは Ack を応答しません。このサービスを受信した BACnet プロバイダーは受け取った UTC 時刻に UTC_Offset プロパティ値を減算してローカル時刻を取得します。
SubscribeCOV	SubscribeCOV サービスは、特定のオブジェクトのプロパティ値の変化通知を受け取るために、BACnet ユーザーによって使用されます。BACnet プロバイダーは Ack を応答します。ConfirmedCOVNotification(確認あり型)および UnconfirmedCOVNotification(確認なし型)サービスは、変更通知を伝達するために BACnet プロバイダーによって使用されます。確認あり型または確認なし型の選択は、BACnet ユーザーからの SubscribeCOV サービスで指定されます。SubscribeCOV サービスを受け取った BACnet プロバイダーは COV 通知を発行する場合、常に BACnet ユーザー宛てにユニキャスト送信します。
ConfirmedCOVNotification	ConfirmedCOVNotification サービスは、特定のオブジェクトのプロパティ値の変化をサブスクライバー(BACnet ユーザー)に通知するために BACnet プロバイダーが使用します。BACnet ユーザーは Ack を応答します。
UnconfirmedCOVNotification	UnconfirmedCOVNotification サービスは、特定のオブジェクトのプロパティ値の変化をサブスクライバーにユニキャスト通知したり、多くのデバイス宛てに、あるオブジェクトプロパティ(外気温度など)をブロードキャスト通知したりするために BACnet プロバイダーによって使用されます。
ReadPropertyMultiple	ReadPropertyMultiple サービスは、1 つまたは複数の BACnet オブジェクトの 1 つまたは複数の指定されたプロパティの値を要求するために、BACnet ユーザーによって使用されます。BACnet プロバイダーは Ack と読み出し結果を応答します。BACnet ユーザーは、任意の数のオブジェクトの任意の数のプロパティを読み取ることができます。特にプロパティ識別子 ALL を使用すると、オブジェクトのプロパティとその値を一括取得できます。

BACnet サービス	概要
ReadRange	ReadRange サービスは、BACnet ユーザーによって使用されます。BACnet プロバイダーは TrendLog オブジェクトの LogBuffer プロパティ内の特定のデータ項目範囲を読み取り Ack を応答します。
GetAlarmSummary	GetAlarmsummary サービスは、BACnet ユーザーによって「アクティブアラーム」の概要を取得するために使用されます。BACnet プロバイダーはオブジェクトの値が NORMAL に等しくない EventState プロパティと、値が ALARM である NotifyType プロパティを Ack 応答します。
GetEventInformation	GetEventInformation サービスは、BACnet ユーザーによってすべての「アクティブイベント状態」の概要を取得するために使用されます。BACnet プロバイダーはイベントを通知したオブジェクトの情報を Ack 応答します。
AcknowledgeAlarm	AcknowledgeAlarm サービスは、BACnet ユーザーによって使用され、BACnet プロバイダーが通知した confirmedEventNotification サービスまたは UnconfirmedEventNotification サービスを BACnet ユーザーが確認したことを BACnet プロバイダーに伝えるために使用されます。BACnet プロバイダーは Ack を応答します。
AtomicReadFile	AtomicReadFile サービスは、BACnet ユーザーによって使用されます。BACnet プロバイダーのコンフィグレーションデータファイルのバックアップを BACnet ユーザーが保持する目的で、このファイルを読み取るために使用されます。BACnet プロバイダーは Ack を応答します。
AtomicWriteFile	AtomicWriteFile サービスは、BACnet ユーザーによって使用されます。コンフィグレーションデータのバックアップファイルを BACnet プロバイダーに転送し、BACnet プロバイダーはコンフィグレーションデータを復元します。BACnet プロバイダーは Ack を応答します。

3.3.3.3 制限事項

本サンプルソフトのリリースバージョン V1.0.0 は主に以下の制限事項がございます。

- ✓ 本サンプルソフトは B-BC デバイスとして必要な全ての機能を実装しているわけではありません。また、BTL テストは未実施です。
- ✓ 本サンプルソフトは B-BC の上位デバイスとして BACnet/IP、下位デバイスとして BACnet MS/TP のプロトコルで接続することを想定しています(Fig.3-6)。Table 3-3 に示すとおり、本サンプルソフトの B-BC は BACnet/IP クライアントデバイスとして下位の BACnet/IP サーバーデバイスと接続することはできません。また、MS/TP スレーブデバイスにはなりません。

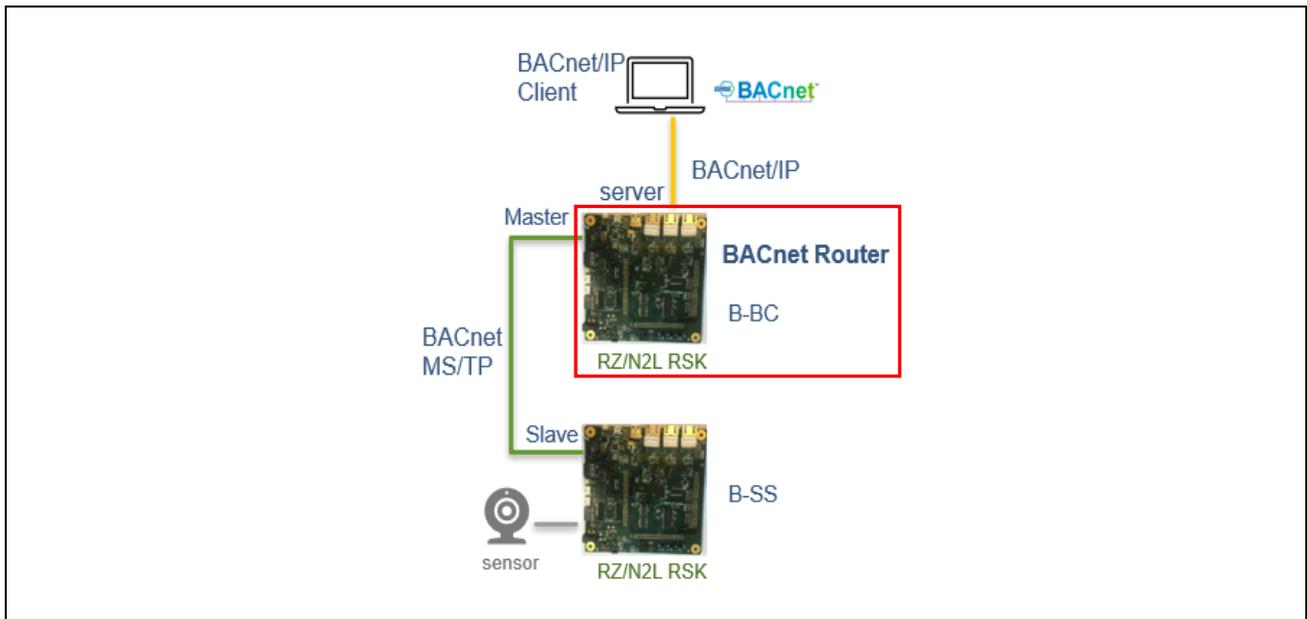


Fig.3-6 BACnet devices that can be connected to B-BC

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				✓				
Lower layer device	B-SS	BACnet/IP	Server						
		BACnet MS/TP	Master			✓(*1)			
		BACnet MS/TP	Slave				✓		

✓はサポート、空欄は非サポート

(*1) B-BC は他の MS/TP マスターデバイス間で、TOKEN の授受を行うことができますが、ReadProperty や WriteProperty などサービスの実行はサポートしていません。

3.3.3.4 BIBBs

BIBBs(BACnet Interoperability Building Blocks)は相互運用する BACnet デバイスに適用するサービスの集合を定義しています。「A」と「B」のデバイスに分かれており、「A」デバイスは BACnet ユーザー、「B」デバイスは BACnet プロバイダーを表します。

BACnet 規格(Annex L)では B-RTR(BACnet Router)、B-BC(BACnet Building Controller)以外に B-SS(BACnet Smart Sensor)、B-OWS(BACnet Operator WorkStation)など、各デバイスの性格を表す様々なデバイスプロファイルが定義されています。

本サンプルソフトの B-BC は「A」と「B」両方の性格を持つデバイスです。

本サンプルソフトの実装 BIBBs を Table 3-4 に示します。(✓は該当、空欄は非該当)

Table 3-4 実装 BIBBs(B-BC デバイスプロファイル)

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	SCHEM-I-B	ReadProperty		✓	✓	
		WriteProperty		✓	✓	
		TimeSynchronization / UTCTimeSynchronization		✓	✓	
	SCHEM-E-B	ReadProperty		✓	✓	
		WriteProperty	✓	✓	✓	

BIBB Class	BIBB	BACnet Service	Initiate ¹	Execute ²	B-BC Standardized ³
		TimeSynchronization / UTCTimeSynchronization		✓	✓
Trending	T-VMT-I-B	ReadRange		✓	✓
	T-ATR-B ⁵	ConfirmedEventNotification	✓		✓
		UnconfirmedEventNotification	✓		✓
		ReadRange		✓	✓

✓は該当、空欄は非該当

1. BACnet サービス要求または通知を送信します。
2. BACnet サービスを実行し、応答(ただし確認あり型サービスを要求された場合)を送信します。
3. BACnet 規格で B-BC プロファイルの標準と定めている BIBB です(ANNEX L.4)。
4. B-SS へのサービス要求を行うときに使用しますが ReadPropertyMultiple と WritePropertyMultiple は使用していません。
5. T-ATR-B で使用する BACnet サービスは実装していますが BUFFER_READY イベントアルゴリズムは未サポートです。

Table 3-5 実装 BIBBs(B-RTR デバイスプロファイル)

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		✓	✓	

✓は該当、空欄は非該当

1. BACnet サービス要求、メッセージまたは通知を送信します。
2. BACnet サービスを実行し、応答(ただし確認あり型サービスを要求された場合)またはメッセージの受け入れを行います。
3. BACnet 規格で B-RTR プロファイルの標準と定めている BIBBs です(ANNEX L.7)。

実装 BIBBs の概要は次のとおりです。

Table 3-6 実装 BIBBs 概要

BIBBs	概要
DS-RP-A	A デバイスは、B デバイスからの 1 つのプロパティユーザーです。
DS-RP-B	B デバイスは、A デバイスへ、1 つのプロパティ値を返します。
DS-WP-A	A デバイスは、B デバイスのプロパティを 1 つ設定します。

BIBBs	概要
DS-WP-B	B デバイスは、A デバイスからの値を 1 つのプロパティに書き込みます。
DS-RPM-A	A デバイスは、B デバイスからのデータユーザーであり、一度に複数のプロパティを要求します。
DS-RPM-B	B デバイスは、A デバイスへ、一度に複数のプロパティ値を返します。
DS-WPM-A	A デバイスは、B デバイスに複数のプロパティを一度に設定します。
DS-WPM-B	B デバイスは、A デバイスからの複数の値を一度に複数のプロパティに書き込みます。
DS-COV-B	B デバイスは、A デバイスからの COV 通知加入を受け付け、COV 通知を A デバイスに行います。
DM-DDB-A	A デバイスは、他のデバイスに識別要求を行い、デバイスのアナウンスを解釈します。
DM-DDB-B	B デバイスは、A デバイスからの識別要求に応答します。
DM-DOB-B	B デバイスは、A デバイスから指定されたオブジェクトを持つ識別要求に応答します。
DM-DCC-B	B デバイスは、A デバイスからの通信停止要求に応答します。
DM-TS-B	B デバイスは、A デバイスからの時刻同期を受け入れます。
DM-RD-B	B デバイスは、A デバイスからの再初期化要求に応答します。
DM-BR-B	B デバイスは、その設定ファイルを A デバイスに提供し、A デバイスから、このファイルを B デバイスに書き込んで B デバイスに障害が発生した場合にその設定を回復できるようにします。
AE-N-I-B	B デバイスは、アラームやその他のイベントに関する通知を生成します。
AE-ACK-B	B デバイスは、送信済みのアラーム/イベント通知に対する A デバイスからの確認に応答します。
AE-INFO-B	B デバイスは、イベント情報を A デバイスに提供します。
SCHED-I-B	B デバイスは、デバイス内の特定のオブジェクトの特定のプロパティの値の日付と時刻のスケジュールを提供します。
SCHED-E-B	B デバイスは、他のデバイスの特定のオブジェクトの特定のプロパティの値の日付と時刻のスケジュールを提供します。
T-VMT-I-B	B デバイスは、内部バッファにトレンドログデータレコードを収集します。
T-ATR-B	B デバイスは、トレンド ログ オブジェクト内の BUFFER_READY イベントアルゴリズムを使用して、トレンドログバッファが所定の数のデータ サンプルを取得したことを A デバイスに通知します。
NM-RC-B	B デバイスはルーター管理コマンドに応答し、規格書の BACnet ルーター要件を満たす必要があります。

3.3.3.5 A デバイスのサポートサービス

本サンプルソフトの B-BC は BIBBs に定義される A デバイス機能を持ちます。

B-BC の A デバイスとしてのサポート BIBBs とサービス要求先および Ack 送信元は次のとおりです。

詳細は「Reference」欄のリンク先をご確認ください。

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) B-SS へのサービス要求を行うときに ReadPropertyMultiple と WritePropertyMultiple は使用していません。

3.3.3.6 オブジェクト

BACnet デバイスはオブジェクトの集合で構成されます。

オブジェクトをオブジェクトタイプと 0~4194303 のインスタンス番号で表したものをオブジェクト ID と呼びます。ただし、4194303 番は無効を意味し、使用されていないことを表します。デバイス自身もオブジェクトであり、Device オブジェクトに定義します。Device のオブジェクト ID をデバイス ID と呼びます。BACnet デバイスは、必ず Device オブジェクトを持つように定められています。さらにオブジェクトは様々なデータ型のプロパティの集合で構成され、このプロパティを読み書きすることで B-SS はハードウェアにアクセスします。

本サンプルソフトの実装オブジェクトを Table 3-8 に示します。(✓は該当、空欄は非該当)

Table 3-8 実装オブジェクト

BACnet オブジェクトタイプ	オブジェクト ID	実装
Accumulator		
Analog Input	Analog Input, 0	✓
	Analog Input, 1	✓
Analog Value	Analog Value, 0	✓
	Analog Value, 1	✓
Averaging		
Binary Output	Binary Output, 0	✓
	Binary Output, 1	✓
Binary Value	Binary Value, 0	✓

BACnet オブジェクトタイプ	オブジェクト ID	実装
	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		
OctetString Value		
Time Value		
Integer Value		

BACnet オブジェクトタイプ	オブジェクト ID	実装
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		

✓は該当、空欄は非該当

実装オブジェクトタイプの概要は次のとおりです。

Table 3-9 実装オブジェクトタイプ概要

BACnet オブジェクトタイプ	概要
Analog Input	Analog Input オブジェクトはハードウェアからのアナログ入力を表すプロパティを持ちます。
Analog Value	Analog Value オブジェクトは BACnet デバイスのメモリに常駐するアナログ値を表すプロパティを持ちます。
Binary Output	Binary Output オブジェクトはハードウェアへの出力であり、ACTIVE か INACTIVE かの 2 つの状態を表すプロパティを持ちます。
Binary Value	Binary Value オブジェクトは BACnet デバイスのメモリに常駐する ACTIVE か INACTIVE かの 2 つの状態を表すプロパティを持ちます。
Device	BACnet デバイスには、確実に 1 つの Device オブジェクトが必要です。BACnet デバイス固有の Object_Identifier プロパティを持ちます。これはネットワーク全体でも固有です。
Multi state Value	Multi state Value オブジェクトは BACnet デバイスのメモリに常駐する 1 つまたは複数の状態を表すプロパティを持ちます。
Positive Integer Value	Positive Integer Value オブジェクトは BACnet デバイスがあらゆる種類の符号なしデータ値にアクセスできるようにするプロパティを持ちます。
Network Port	Network Port オブジェクトは BACnet デバイスのネットワーク構成を表すプロパティを持ち、少なくとも 1 つのネットワークポートオブジェクトが含まれている必要があります。
File	File オブジェクトはファイルサービスを使用してアクセスできるデータファイルのプロパティを持ちます。
Notification Class	Notification Class オブジェクトは、BACnet システム内でのイベント通知に必要なプロパティを持ちます。
Schedule	Schedule オブジェクトは、任意の日付の任意の時間に、指定した日付の範囲内で繰り返される定期的なスケジュールと特定のオブジェクトの特定プロパティへの指定された値の書き込みを紐づけるためのプロパティを持ちます。

BACnet オブジェクトタイプ	概要
Trend Log	Trend Log オブジェクトは、参照されるオブジェクトのプロパティを監視し、定義された条件が満たされると、プロパティの値とタイムスタンプを LogBuffer プロパティで表される内部バッファに保存します。LogBuffer プロパティの読み出しには ReadRange サービスが必要です。

3.3.3.7 プロパティ

BACnet のオブジェクトは、プロパティと呼ばれる様々なデータ要素を持ち、サービスを用いて各プロパティにアクセスします。Conformance Code に必須(R: Required) と定義されているプロパティはそのオブジェクトをサポートするときには必ずサポートしなければならないプロパティです。オプションプロパティ(O: Option)のサポートは任意ですが、条件によりサポート (もしくは非サポート)が定められているプロパティも多くあります。

Table 3-10 から Table 3-22 に本サンプルソフトに含まれる各サポートオブジェクトとそのプロパティを示します。表中の凡例を以下に示します。

1. CC : Conformance Code

R:必須、O:オプション、W:書き込み可

2. Configurable

(*1) 初期設定コマンドおよび WriteProperty サービスで設定可能

5.3 初期設定コマンドを使ってプロパティ値を Flash メモリに書き込み後、ボードをリセットするとプロパティに初期値が反映されます。WriteProperty または WritePropertyMultiple サービスを実行した場合にも値が Flash メモリに書き込まれます。

(*2) 初期設定コマンドで設定可能

6.3 初期設定コマンドを使ってプロパティ値を Flash メモリに書き込み後、ボードをリセットすると初期値がプロパティに反映されます。

(*3) WriteProperty サービスで設定可能

WriteProperty または WritePropertyMultiple サービスを実行した場合にプロパティ値が Flash メモリに書き込まれます。ボードをリセットするとプロパティに反映され、初期化されます。

3. Access

R:ReadProperty または ReadPropertyMultiple サービスを使って読み取り可能

W:WriteProperty または WritePropertyMultiple サービスを使って書き込み可能

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 開発環境構築

3.4.1 統合開発環境 e2studio

3.4.1.1 インストール

Table 1-1 に記載のバージョンをダウンロードして、お使いの PC にインストールしてください。

FSP 最新バージョンでは、FSP、e2studio、GCC ツールチェーンが 1 つのパッケージとして同梱されたインストーラーがダウンロードできます。

- ・ダウンロードした setup_rznfsp_v1_3_0_e2s_v2023-07.exe をダブルクリックします。

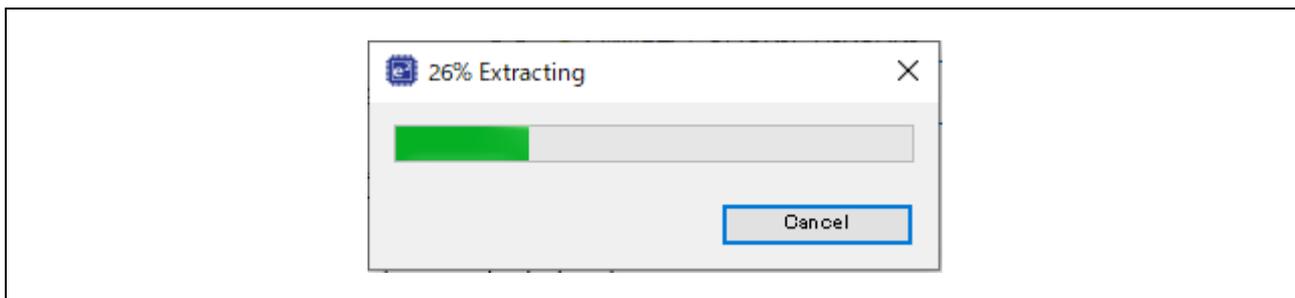


Fig.3-7 e2studio Install (1)

- ・以下を選択します。

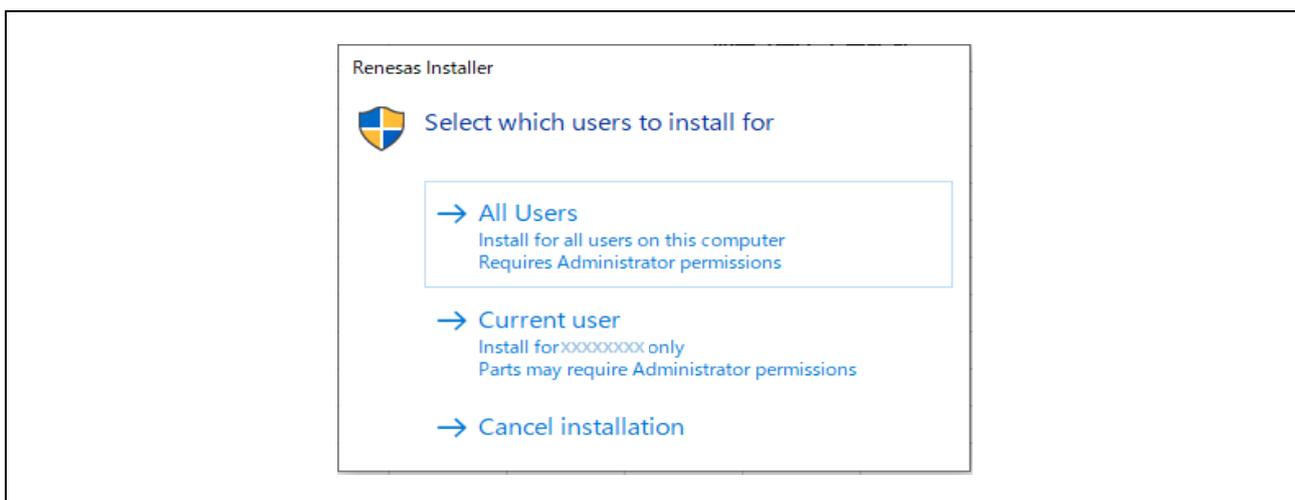


Fig.3-8 e2studio Install (2)

- ・ 以下を選択します。

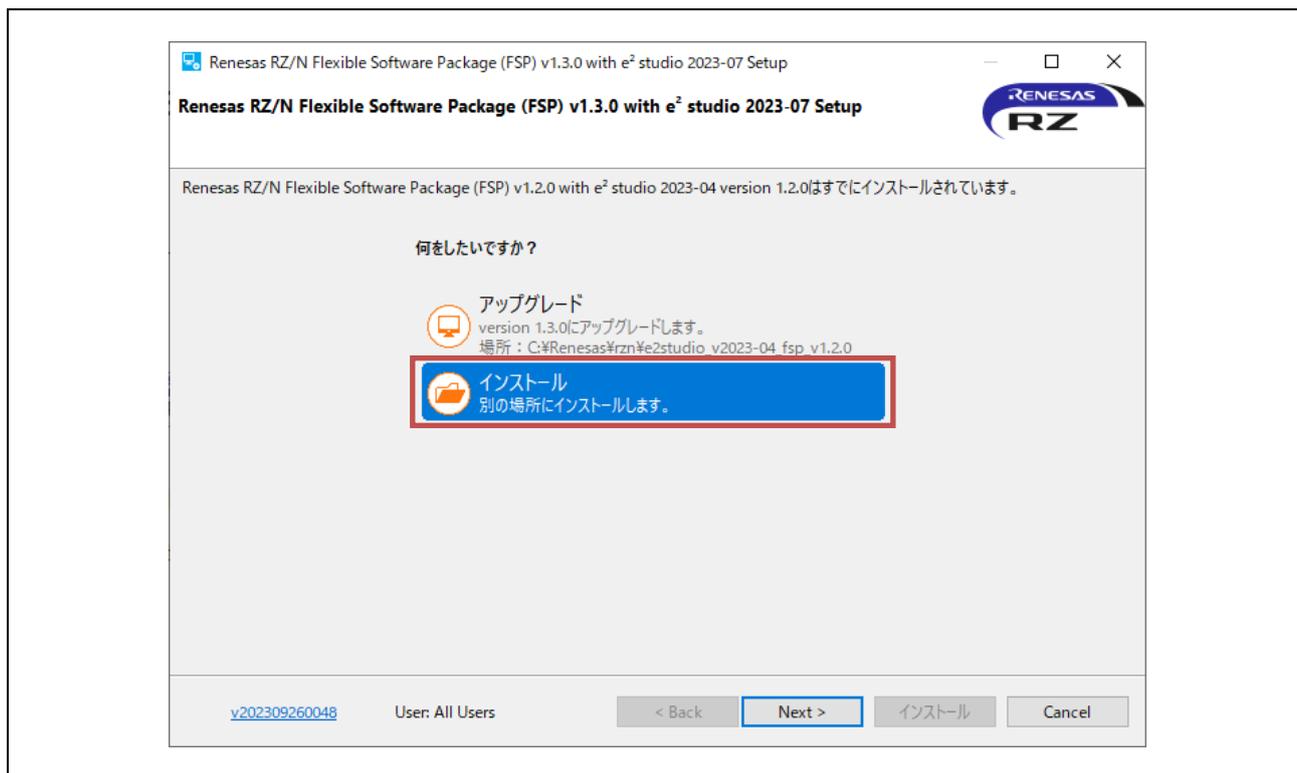


Fig.3-9 e2studio Install (3)

- ・ 以下を選択します。

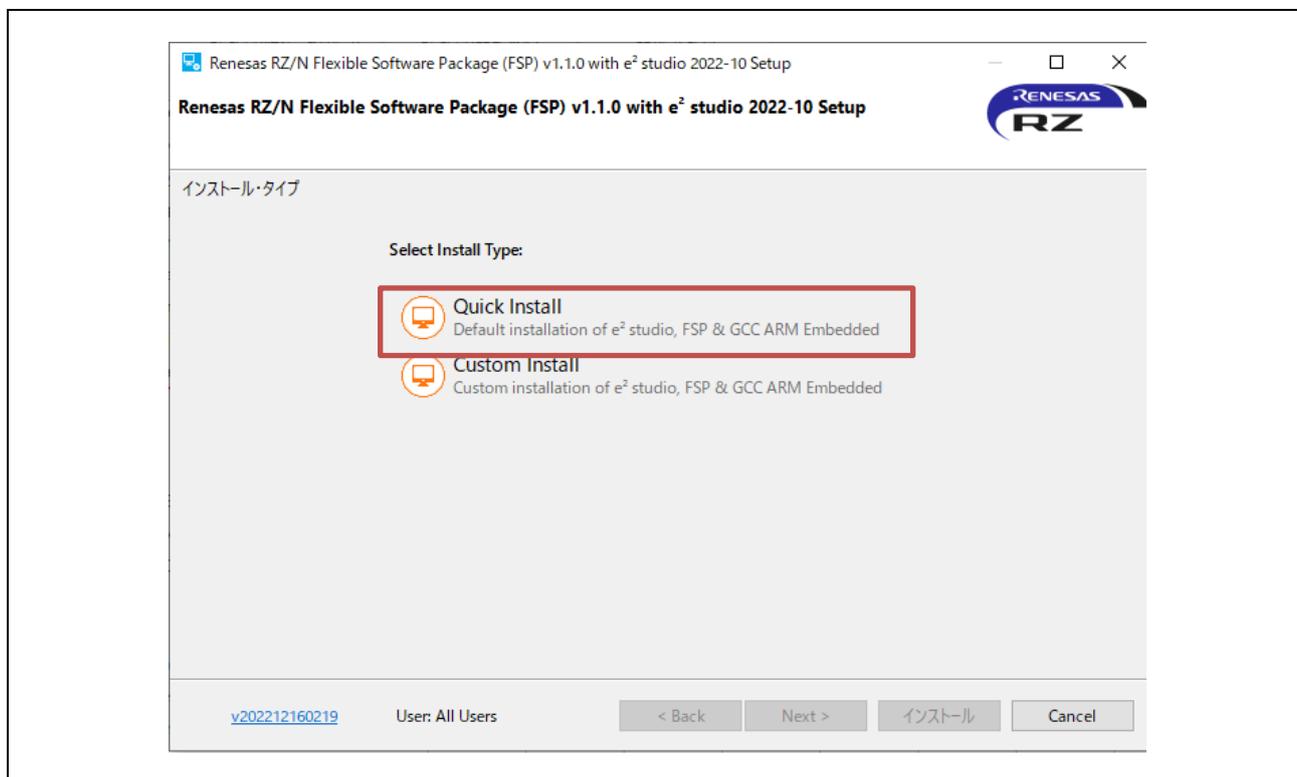


Fig.3-10 e2studio Install (4)

- ・ インストールフォルダを設定します。

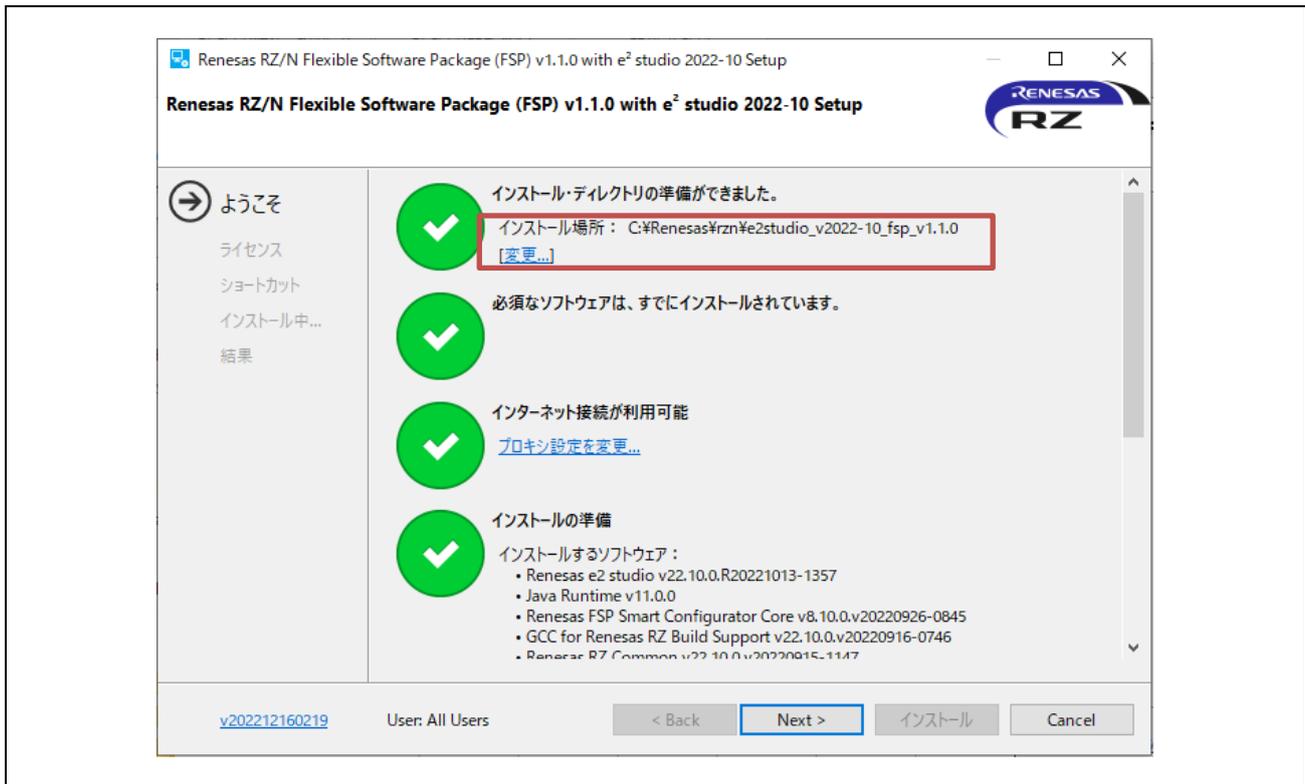


Fig.3-11 e2studio Install (5)

- ・ 同意をチェックした後、Next をクリックします。



Fig.3-12 e2studio Install (6)

- ・インストールをクリックします。

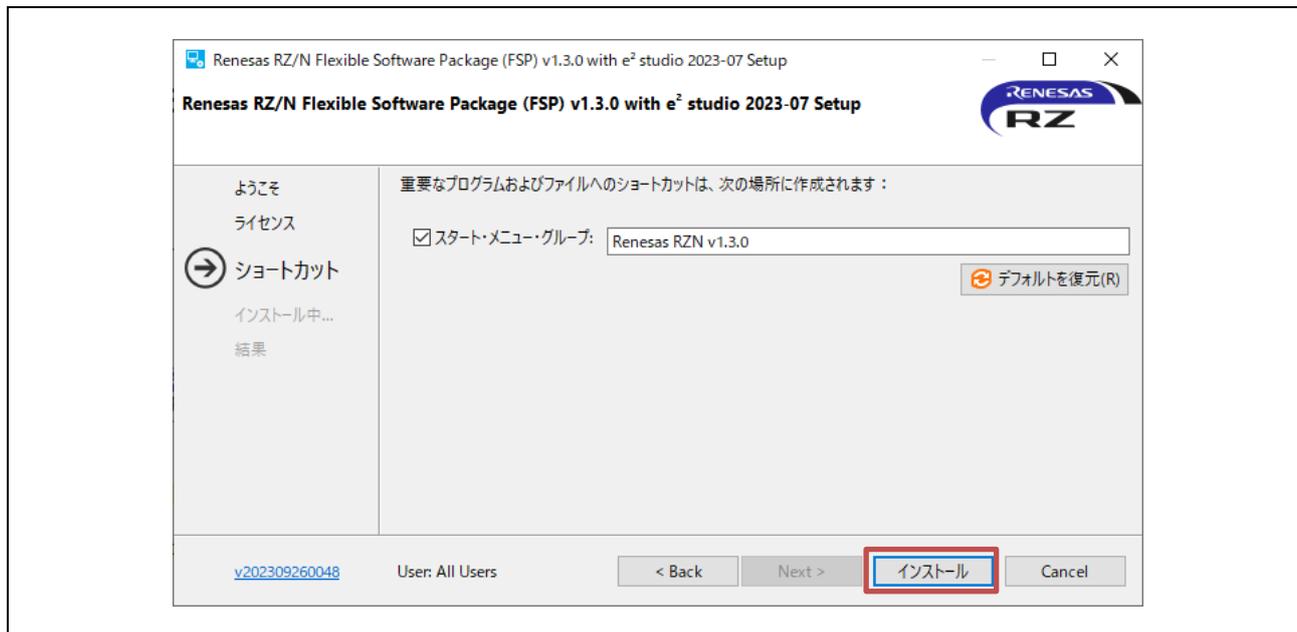


Fig.3-13 e2studio Install (7)

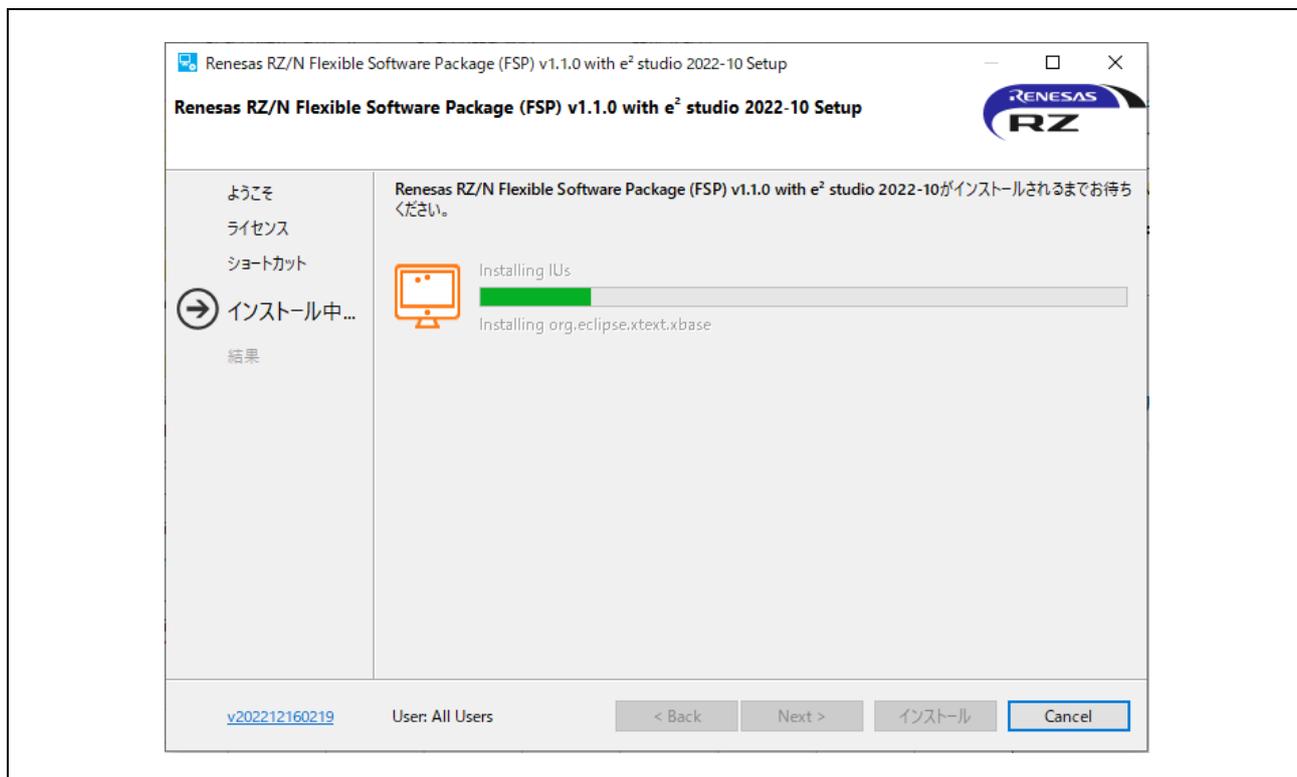


Fig.3-14 e2studio Install (8)

- ・ OK をクリックします。



Fig.3-15 e2studio Install (9)

3.4.1.2 プロジェクト立ち上げ

(1) zip ファイル解凍

まず、アーカイブされた本サンプルソフトのパッケージ (RZN2L_BACnet_BBC_V***.zip) を解凍し、任意のフォルダに格納します。e2studio はフォルダ階層が深くてフルパスが長すぎると認識できませんので、フルパスが短くなるよう配置してください。また、日本語のパスも使用しないでください。

(2) e2studio 起動

次に、e2studio を起動します。インストールされたフォルダの” e2studio.exe” を実行してください。デフォルトのインストール先は下記になります。

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

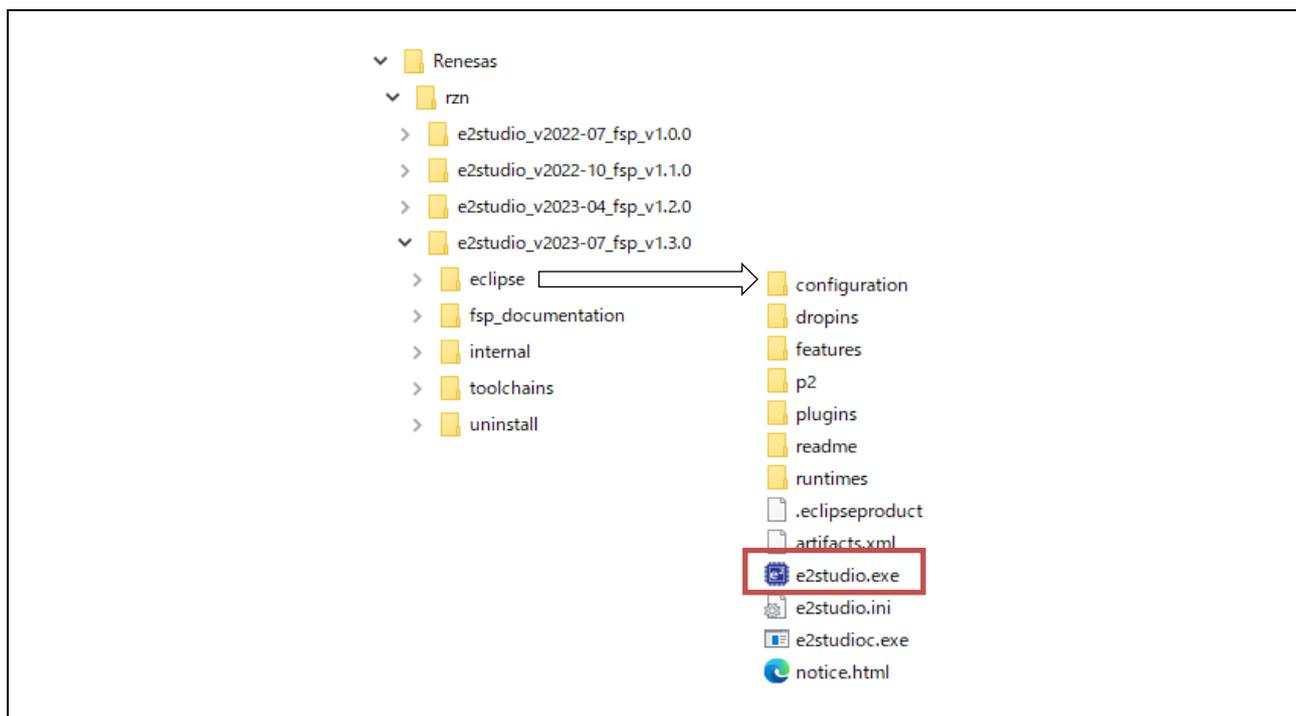


Fig.3-16 Launch project (1)

(3) プロジェクトのインポート

- ・ 任意のワークスペースディレクトリを入力して、Launch をクリックします。

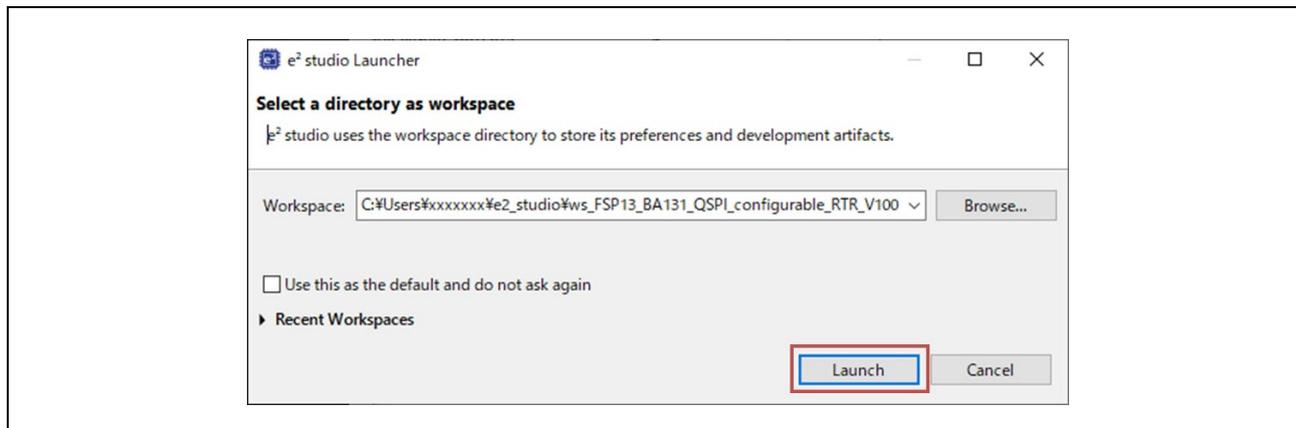


Fig.3-17 Launch project (2)

- ・ Import existing projects をクリックします。

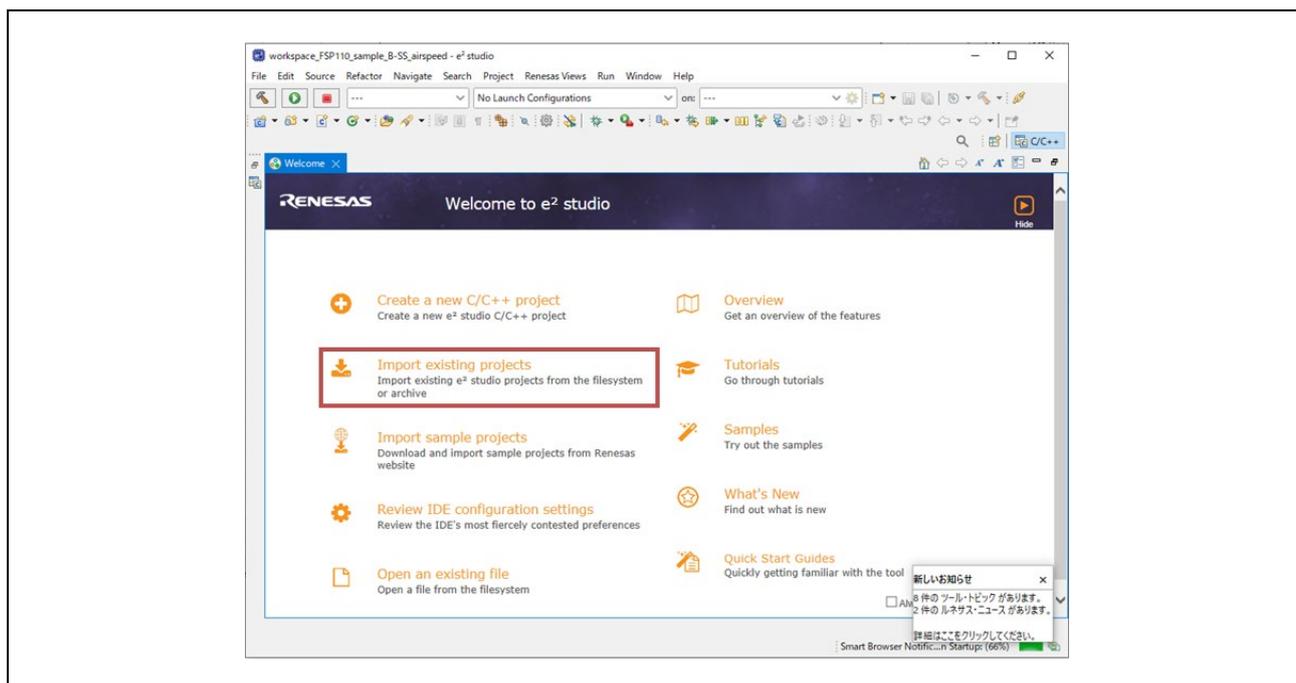


Fig.3-18 Launch project (3)

- ・ Select root directory: の Browse をクリックして、インポートするプロジェクトフォルダを入力します。
Copy projects into workspace にチェックを入れるとインポートプロジェクトがコピーされます。

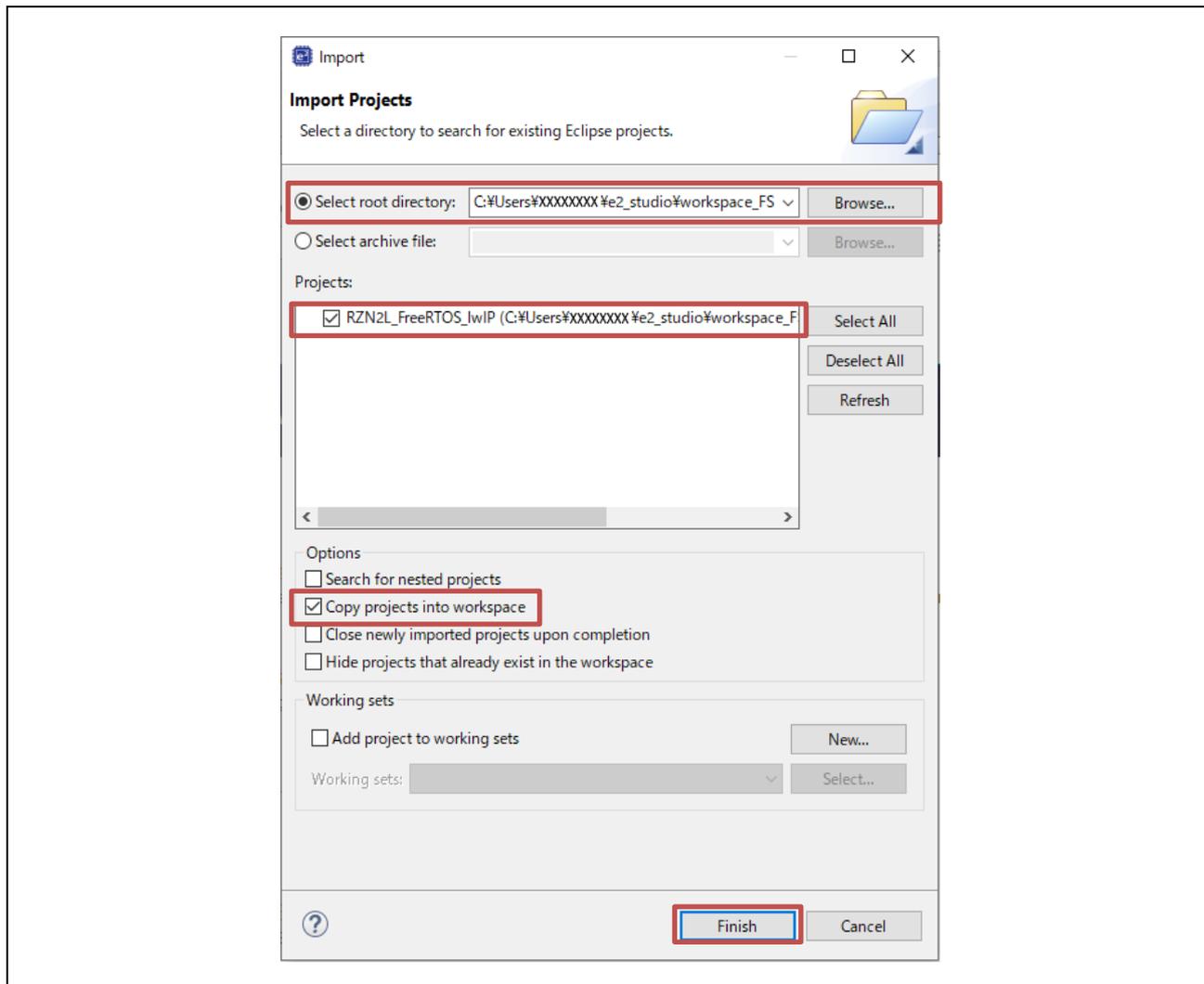


Fig.3-19 Launch project (4)

- ・ Finish をクリックすると以下が表示されますので Yes To All をクリックします。

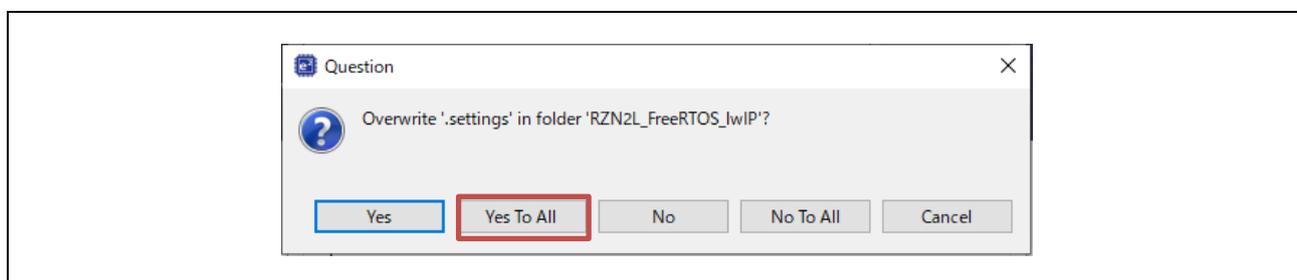


Fig.3-20 Launch project (5)

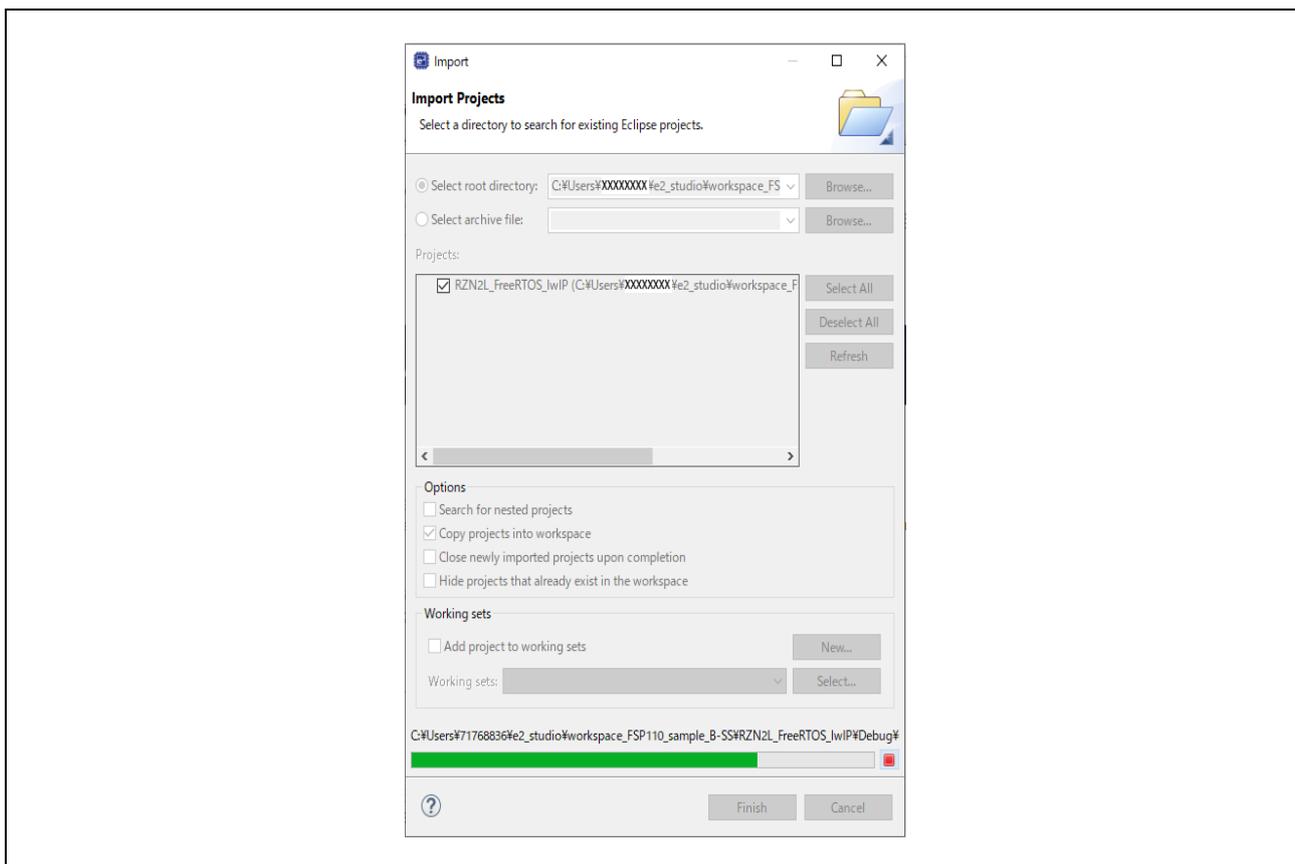


Fig.3-21 Launch project (6)

- ・プロジェクトのインポートが終了すると以下が表示されます。これ以降は4章にて説明します。

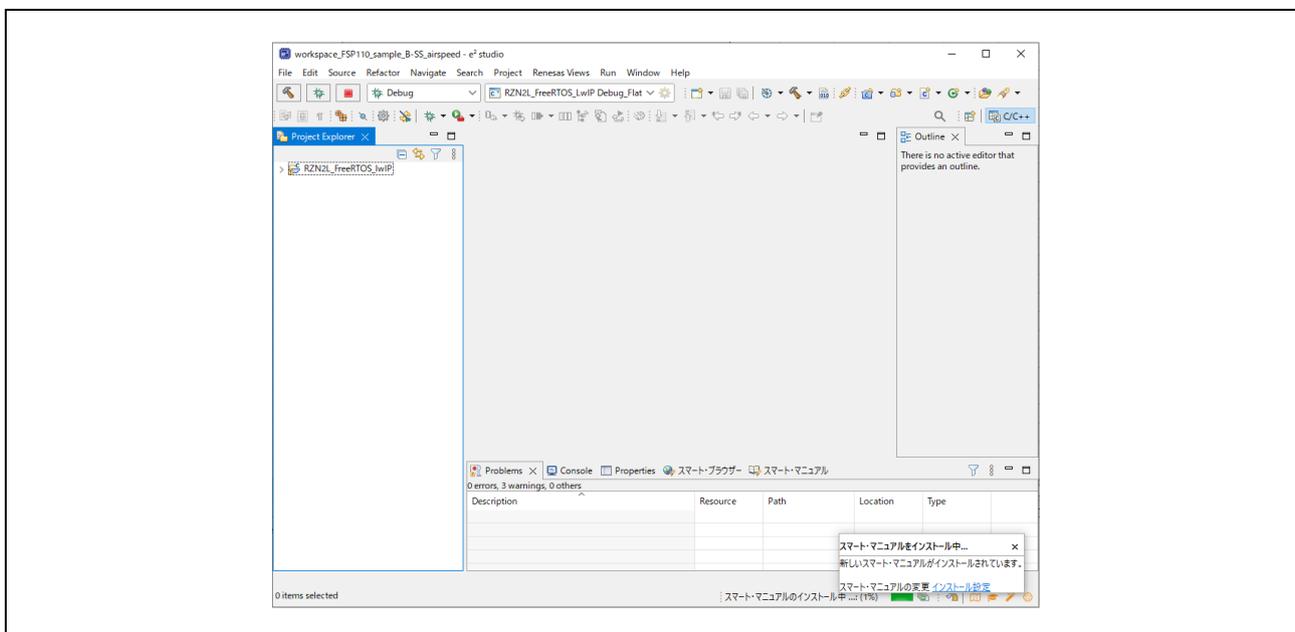


Fig.3-22 Launch project (7)

3.4.2 VTS

Visual Test Shell (VTS) は、BACnet/IP プロトコルを使用するシステムで使用される BACnet 機能をテストするためのアプリケーションです。

Table 1-1 に記載のバージョンを以下の web サイトからダウンロードして、お使いの PC にインストールしてください。ダウンロードファイルを解凍後、インストール手順は"Docs"フォルダの QuickStart.html を開き、Quick Start Guide を参照ください。

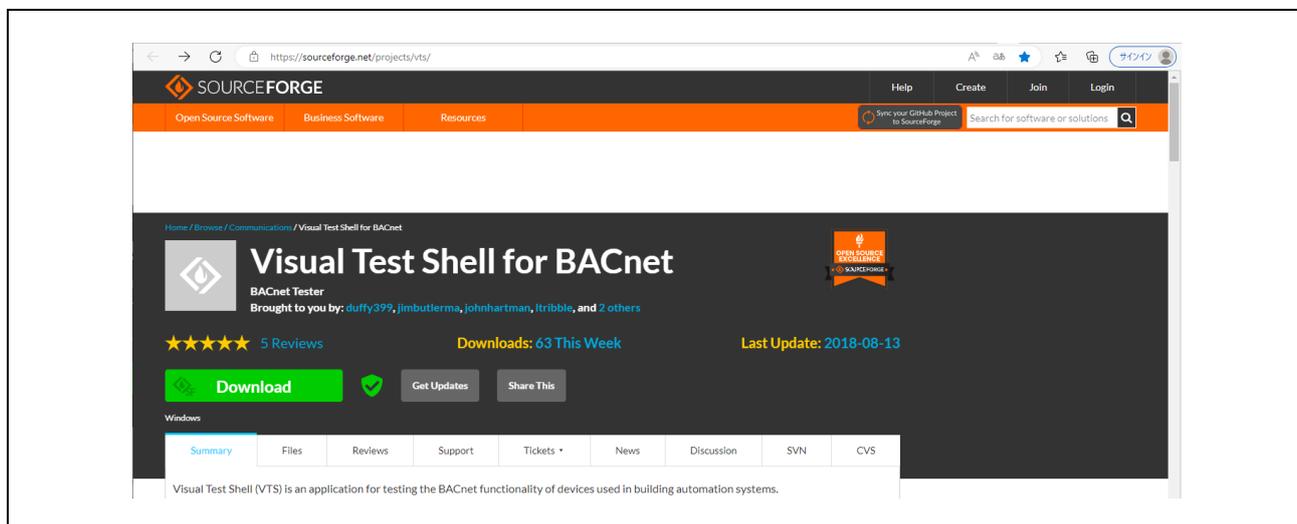


Fig.3-23 VTS

3.4.3 Yabe

Yet Another Bacnet Explorer (YABE) は BACnet デバイスを探索およびナビゲートするためのグラフィカルウィンドウプログラムです。VTS のような 1 つ 1 つのサービスを出力するインターフェースを持ちませんが簡単な操作で BACnet MS/TP と BACnet/IP プロトコルを使用するシステムを試験できます。

Table 1-1 に記載のバージョンを以下の web サイトからダウンロードして、お使いの PC にインストールしてください。

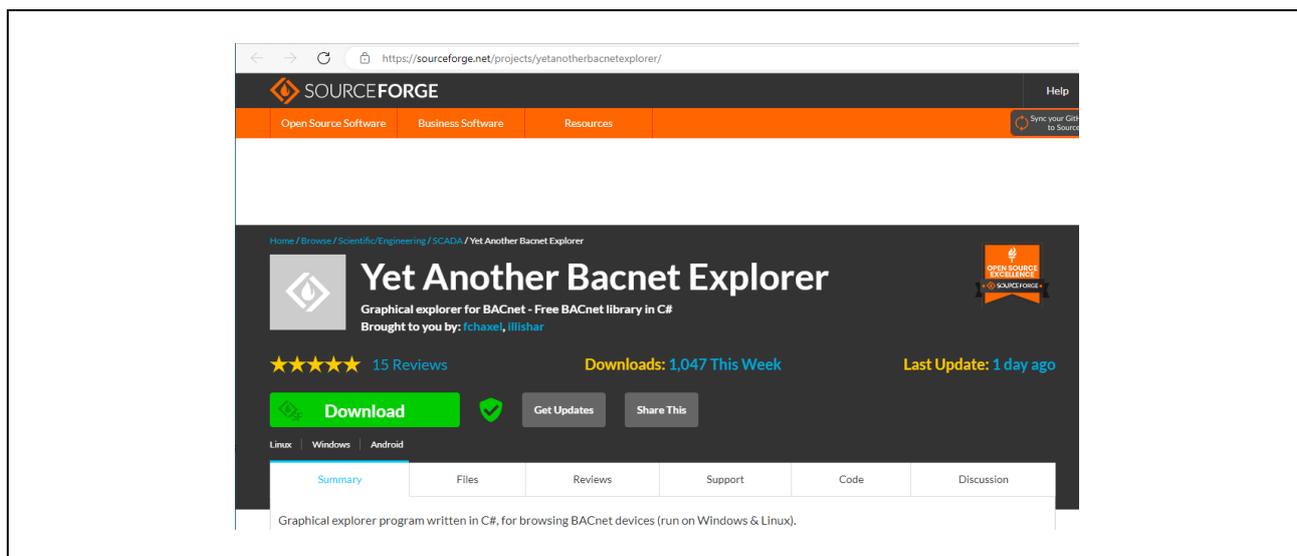


Fig.3-24 Yabe

3.4.4 Wireshark

Wiresharkは無償で使えるネットワークプロトコルアナライザーです。Table 1-1 のリンクから Wireshark をダウンロードしてインストールします。

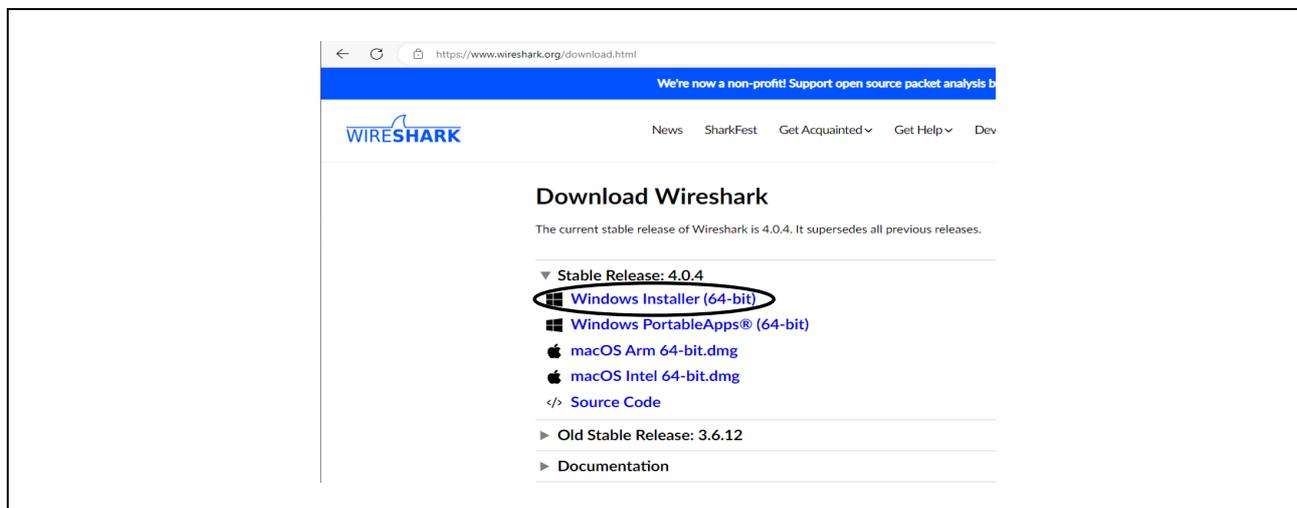


Fig.3-25 download Wireshark

3.4.5 Terminal software

TeraTerm などの無償で使えるターミナルソフトウェアをダウンロードしてインストールします。

このターミナルソフトウェアは構成設定可能なプロパティ値(configurable properties)を Flash メモリに保存するコマンドを実行するために使います。詳細は、5.3 章をご参照ください。

4. 動作確認

4.1 接続

Fig.4-1 にサンプルソフト実行時の接続図を示します。RZ/N2L RSK ボードに Ethernet ケーブル、J-Link OB デバッガ、5V DC の各ケーブルを接続してください。B-SS 側ボードを接続する場合は同ボードの J26 コネクタに空気速度センサを接続してください。

2.1 章のボード設定では、ETH0~ETH1 のどの Ethernet コネクタに Ethernet ケーブルを接続しても構いません。RSK ボード上のデバッガ J-Link OB を使用する場合は J9 をオープンとし、USB Micro のケーブルを接続します。B-BC 側ボードと B-SS 側ボードのボード設定は同一です。

RS-485 は 2 線式半 2 重通信で、B-BC と B-SS のボード間をプラス側 CN12-6 ピン (RS485_A) 同士を、マイナス側 CN12-3 ピン (RS485_B) 同士を接続します。PC ツールからのサービス要求をマスターデバイスとしての B-BC を介して B-SS に送信し、B-SS 応答を PC ツールに送信します。

BACnet MS/TP 通信の packets を Wireshark でモニタする場合は、PC の USB ポートを MS/TP 通信用と分ける必要があるため、RS485/USB 変換器を用意し、RS485_A/B 信号を分岐して接続します。(Fig.4-1 点線部)

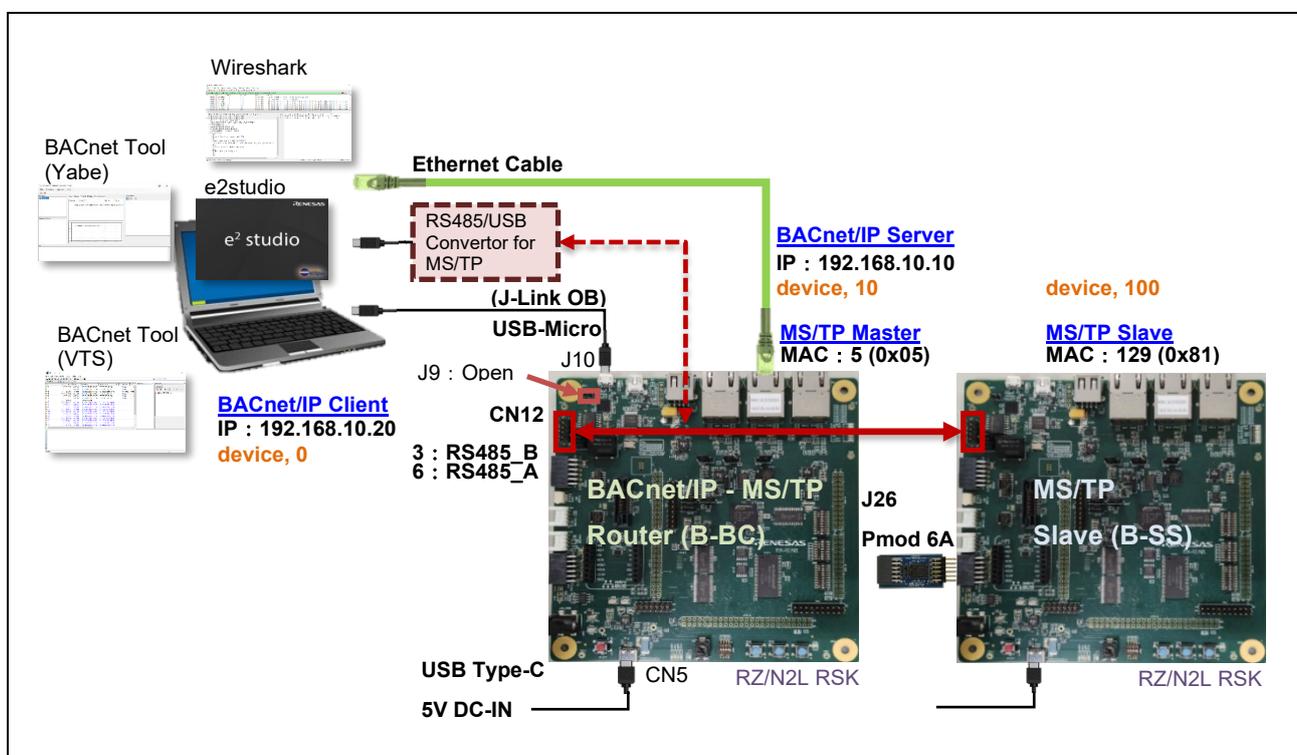


Fig.4-1 BACnet/IP-MS/TP Hardware Diagram

4.2 BACnet クライアント IP アドレス設定

PC 側イーサネットのアドレス設定を行います。

Windows のスタート  の設定  をクリックします。

以下のとおりに進み、IP アドレスを設定してください。

設定 > ネットワークとインターネット > アダプターのオプションを変更する > イーサネット
> プロパティ > インターネットプロトコルバージョン 4(TCP/IPv4) > プロパティ



Fig.4-2 network connection

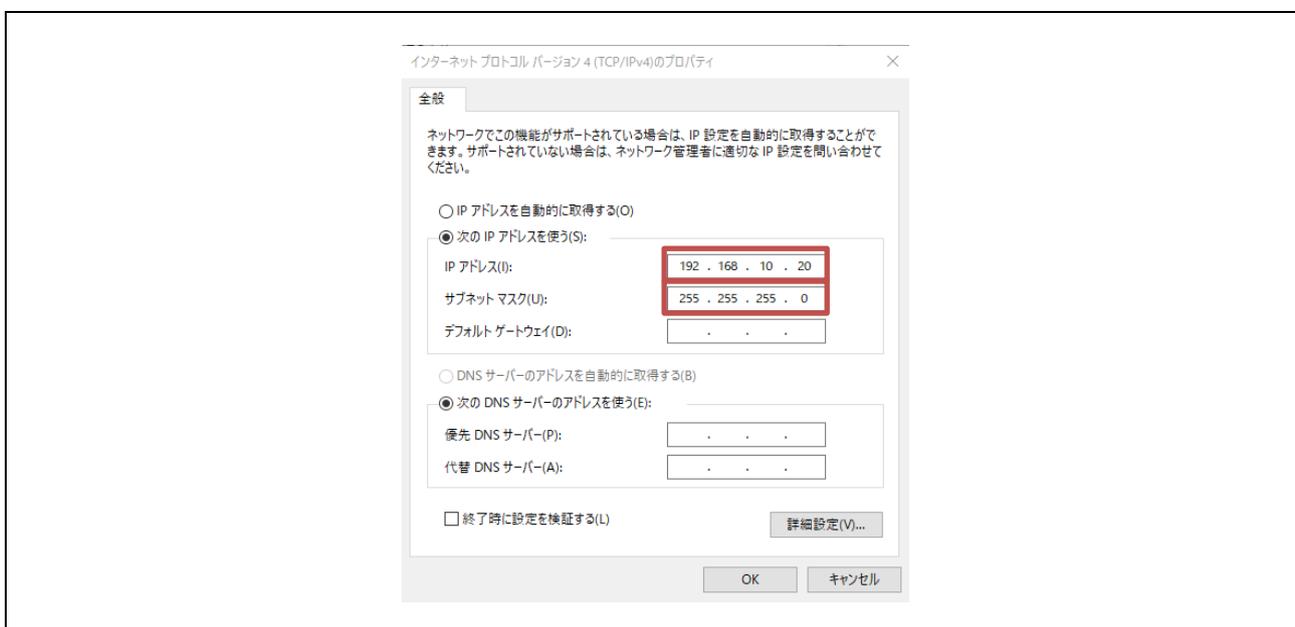


Fig.4-3 TCP/IPv4 properties

B-BC サンプルソフトで使用する RSK ボードの IP アドレスは 192.168.10.10 です。PC 側の IP アドレスを 192.168.10.XXX に設定する必要があります。本ドキュメントでは 192.168.10.20 を設定しています。

4.3 Wireshark 設定

Wireshark で PC と B-BC 間の BACnet/IP 通信パケットをキャプチャすることができます。

また、Wireshark で MS/TP プロトコルパケットのキャプチャを可能にする場合は、Table 1-1 のリンクから mstpcap.exe をダウンロードします。

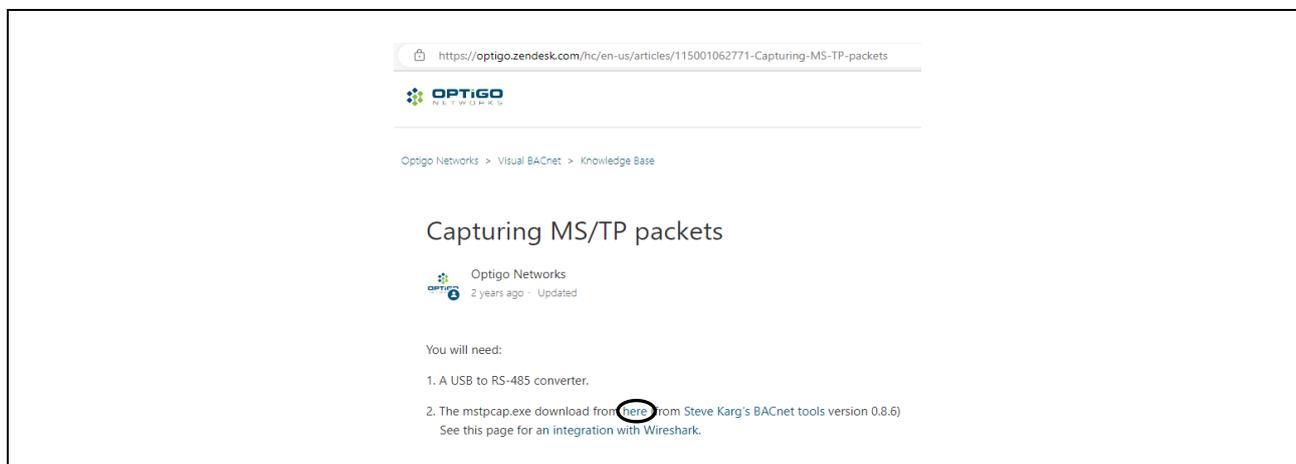


Fig.4-4 Download mstpcap.exe

mstpcap.exe を /Program Files/Wireshark/extcap フォルダ下に貼り付けします。

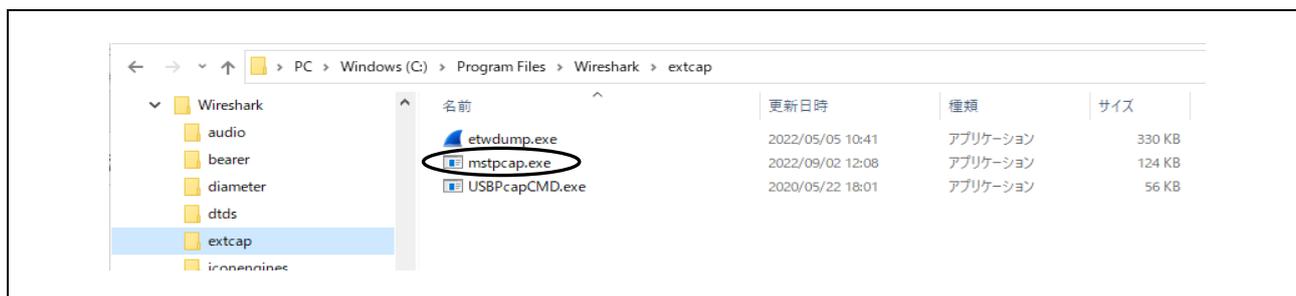


Fig.4-5 Paste mstpcap.exe into extcap folder

ワイヤーシャークを立ち上げてワイヤーシャーク用の COM ポート設定  をクリックします。

ポップアップダイアログの Baud Rate 115200 を選択して Save します。

パケットキャプチャ開始を  クリックします。

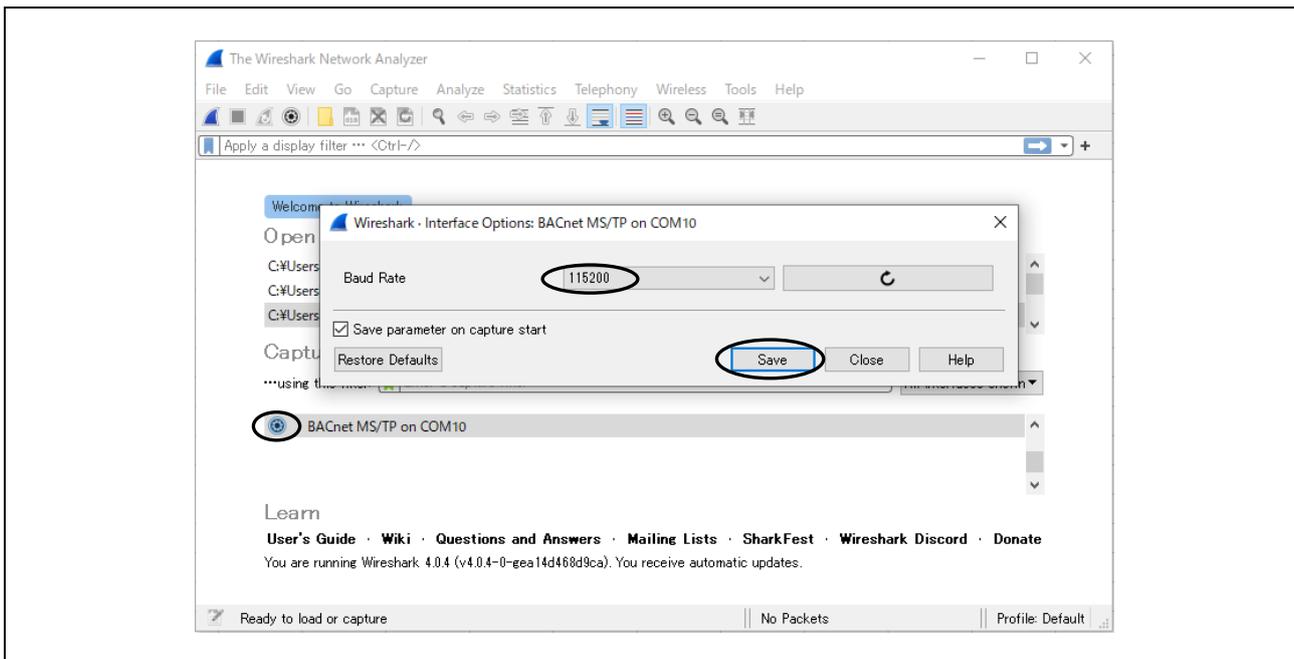


Fig.4-6 Baud rate selection

ワイヤーシャークの MS/TP キャプチャ画面が表示されます。

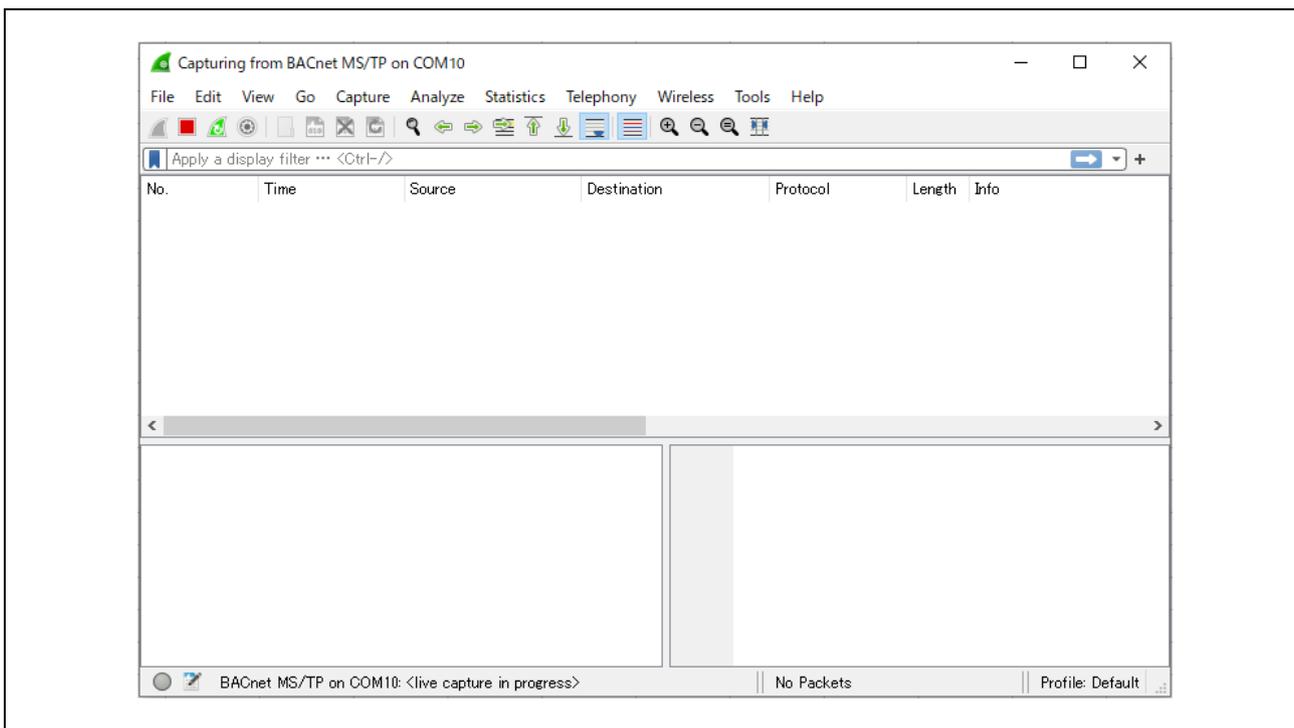


Fig.4-7 MS/TP protocol packet capture screen

4.4 プロジェクト起動

まず、3.4.1.2 章の手順で、プロジェクトをインポートします。

4.4.1 ビルド設定の注意事項

ビルドは各種 Symbol 定義を参照します。詳細は 5.2.4 章および 5.2.6 章を参照ください。

4.4.1.1 変更禁止 Symbol

値を変更すると B-BC をビルドできなくなる Symbol がありますので、以下に示します。

Project Explorer ウィンドウのプロジェクト名を選択したうえで、Project メニューの Properties を開きます。

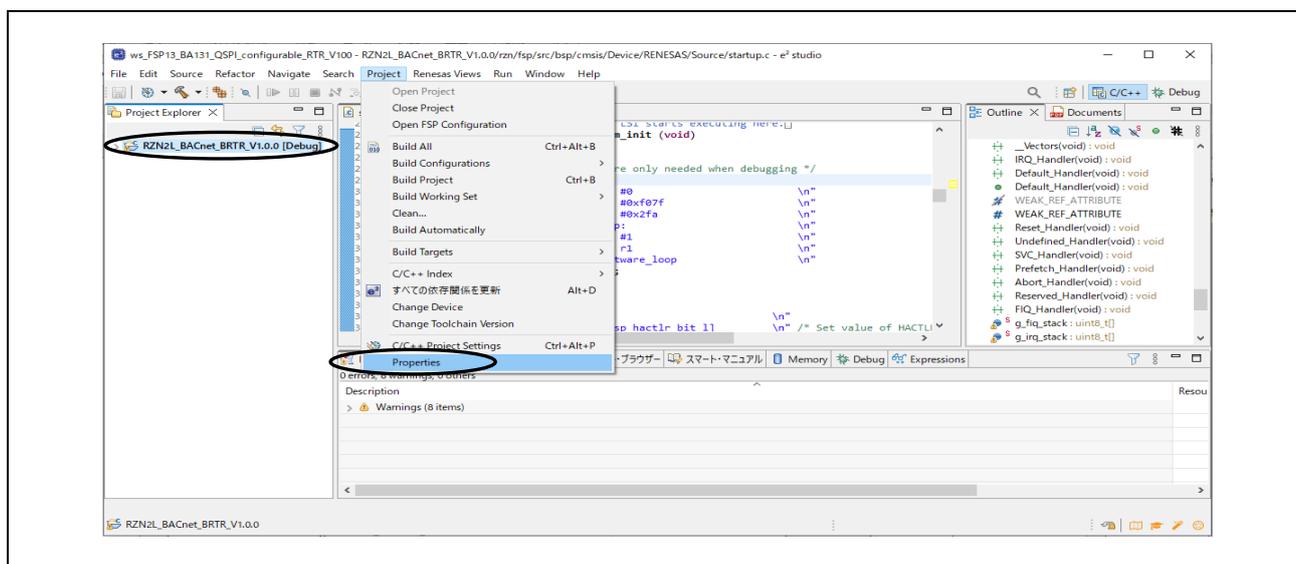


Fig.4-8 Open project properties

C/C++ General > Paths and Symbols の #Symbols タグから Languages の GNU C を選択するとビルド用パラメータ Symbol が表示されます。

Symbol の #BACDL_ALL、#BACDL_BIP、#BACDL_MSTP については変更しないでください。これらを変更すると B-BC がビルドできなくなります。

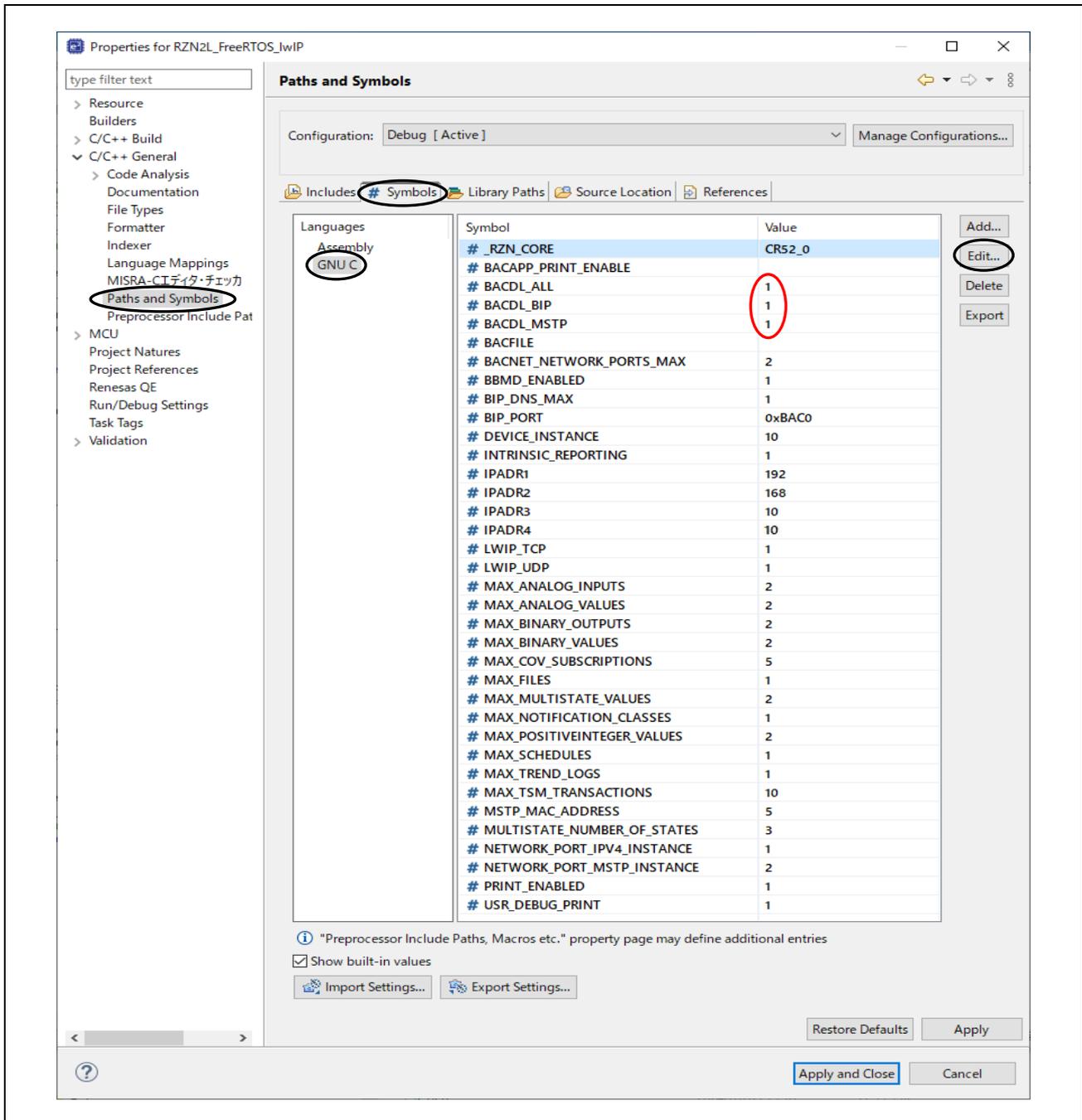


Fig.4-9 Prohibit changes to BACDL_ALL, BACDL_BIP, BACDL_MSTP

4.4.1.2 NTP サーバーアドレスの設定

B-BC はオープンソース lwIP の SNTP(Simple Network Time Protocol)を使用して NTP(Network Time Protocol)サーバーから自動的に現在時刻を取得するために NTP クライアント機能を実装しています。

ただし、接続 PC が NTP サーバーになることを前提としており、B-BC は NTP サーバーを IP アドレスで識別しています。NTP サーバー名による識別はサポートしておりません。

接続する PC に NTP サーバー機能がない場合でも問題ありません。B-BC は VTS(または Yabe)から現在時刻を取得します。4.5.3 章および 4.6.1 章の時刻同期サービスを参照ください。

B-BC と BIP 接続する PC が NTP サーバーとして使用できる場合のみ、以下のコードを変更してください。この IP アドレスは 4.2 章で設定したものと同一である必要があります。192.168.10.20 はデフォルト値です。

user\renesas\application\lwip_port_main.c

```

1400 /*****
1401  * @brief SNTP initial settings
1402  *****/
1403 /*****
1404  * Function Name: user_sntp_init
1405  * Description : 8.3a and 8.3c valid function (8.3a REQUIRED)
1406  * Arguments : void
1407  * Return Value : None
1408  *****/
1409 void user_sntp_init(void);
1410
1411 void user_sntp_init(void)
1412 {
1413     ip_addr_t sntpIPADDR;
1414
1415     sntp_setoperatingmode(SNTP_OPMODE_POLL);
1416     #if SNTP_SERVER_DNS
1417     sntp_setservername(0, "pool.ntp.org"); /* DNS not supported */
1418     #else
1419     IP_ADDR4(&sntpIPADDR, 192, 168, 10, 20);
1420     sntp_setserver(0, &sntpIPADDR);
1421     #endif
1422     sntp_init();
1423 }

```

Fig.4-10 Setting NTP server address

次の Wireshark(フィルタは ntp を使用)キャプチャ画像は NTP プロトコルパケットを示しています。B-BC から NTP サーバーへの時刻要求周期は 1 時間です。

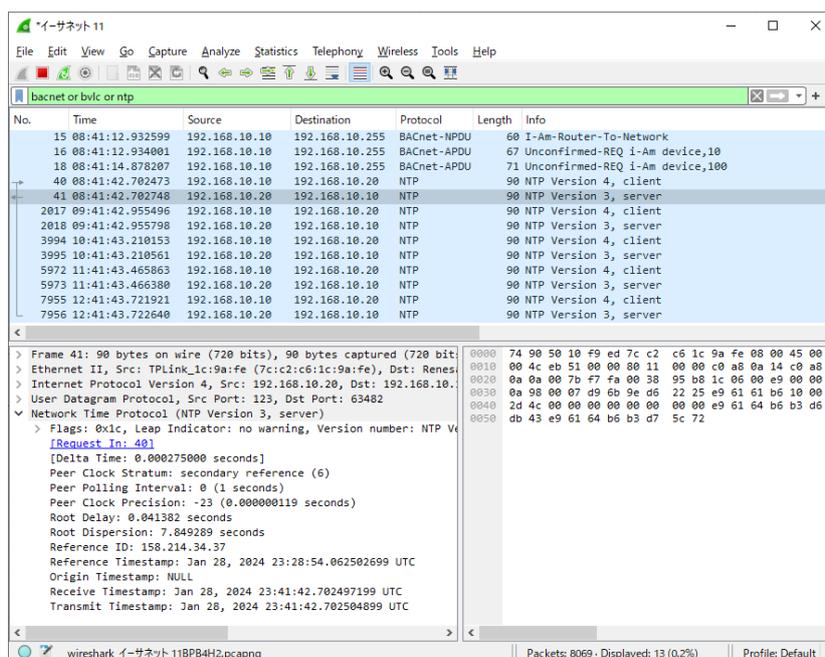


Fig.4-11 Capture image of NTP protocol packet

4.4.2 ビルド

Project Explorer ウィンドウのプロジェクト名を選択したうえで、Project メニューの Clean... をクリックします。

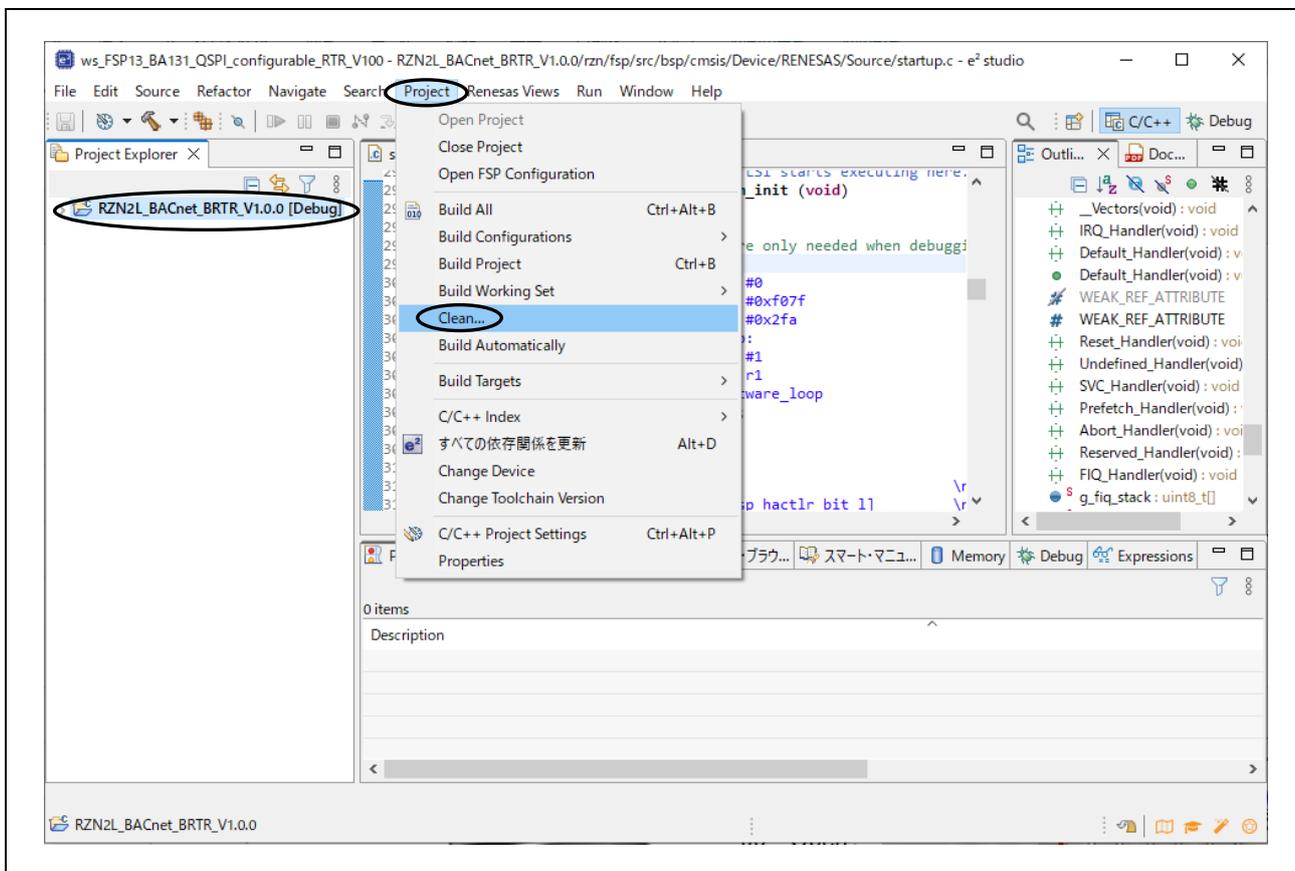


Fig.4-12 Open project Clean...

ポップアップダイアログの以下を有効にして Clean をクリックすると全ビルドを開始します。

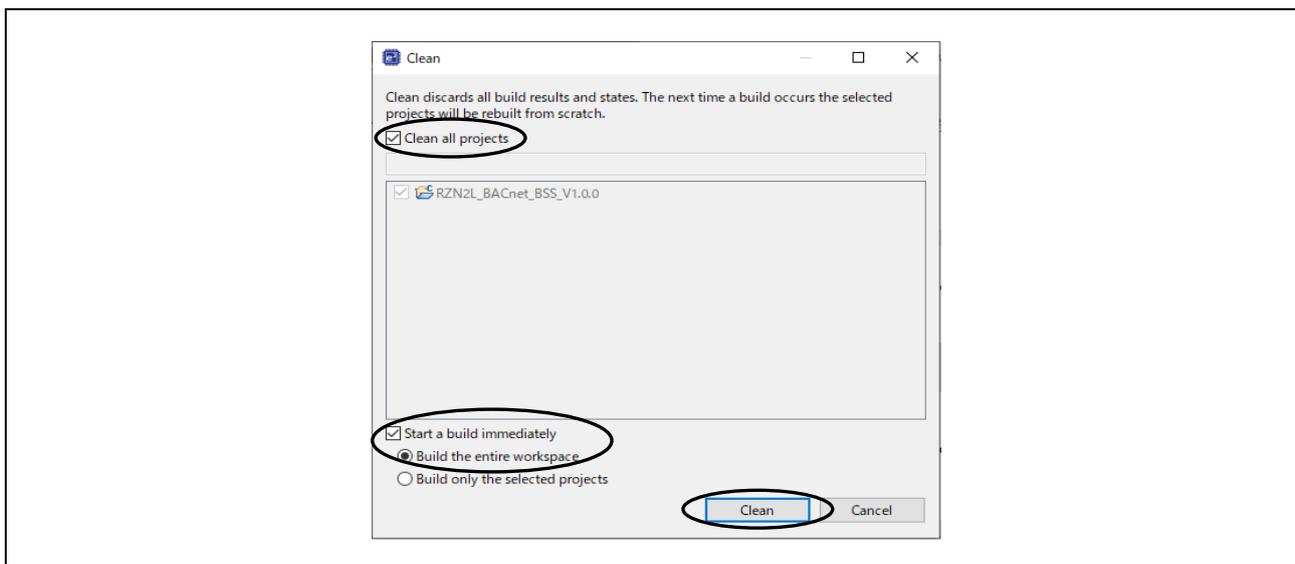


Fig.4-13 clean and rebuild

4.4.3 Debug Configurations 設定

全ビルド結果が 0 errors であることを確認後、Project Explorer ウィンドウのプロジェクト名を選択したうえで、Run メニューの Debug Configurations... をクリックします。Warning が発生しますが、無視してください。

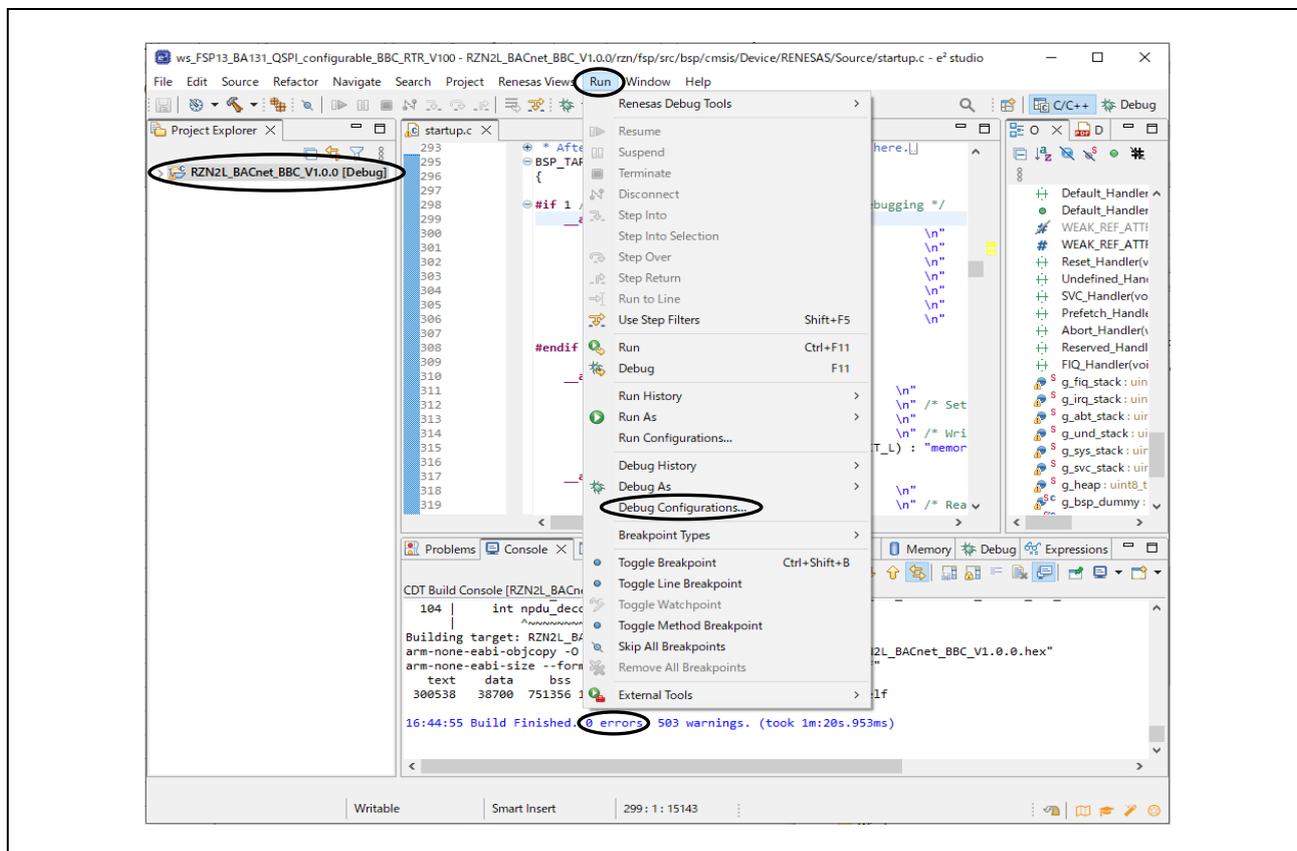


Fig.4-14 Open Debug Configurations...

- プロジェクトインポート後、初回デバッグ起動するときの操作
プロジェクトをインポートして、初回にデバッグ起動を行うときだけ、次の操作を行ってください。
 - a. RZN2L_BACnet_BBC_V*** Debug[local]を生成
 - b. Target Device を選択
 - c. デバッグ・ツール設定
 - d. マクロ登録

上記は次に続く説明を参照ください。

a. RZN2L_BACnet_BBC_V*** Debug[local]を生成

Renesas GDB Hardware Debugging をダブルクリックして、RZN2L_BACnet_BBC_V*** Debug[local]を生成します。

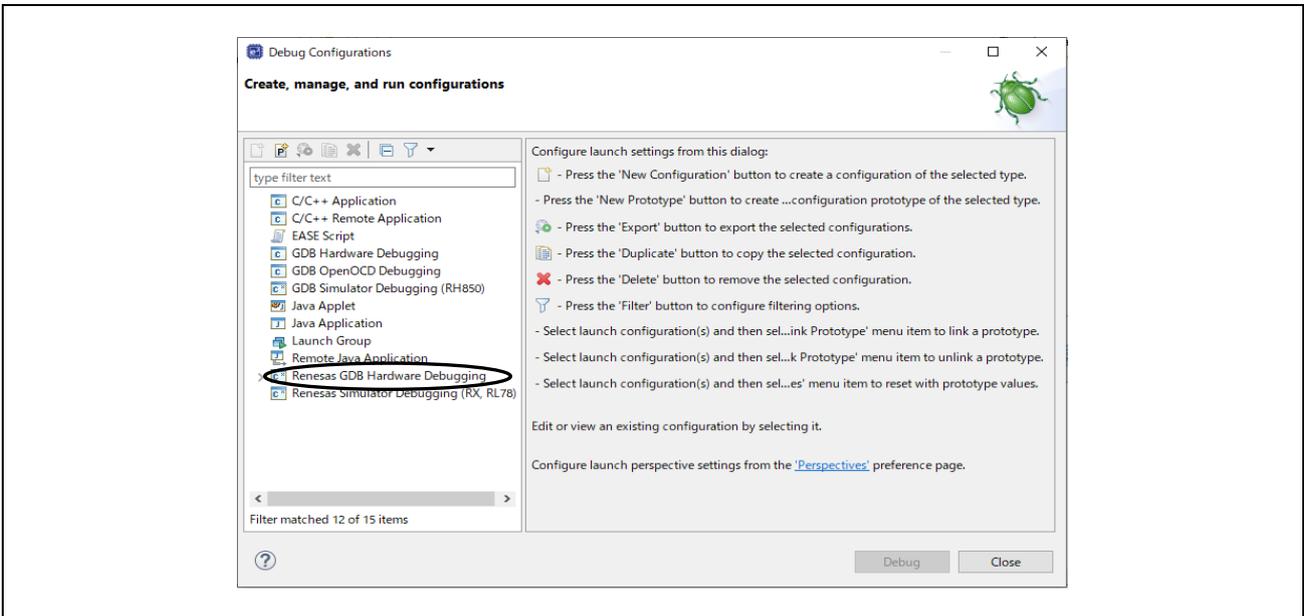


Fig.4-15 Debug Configurations(1)

b. Target Device を選択

表示されたダイアログの Debugger タグをクリックして、Target Device を選択します。

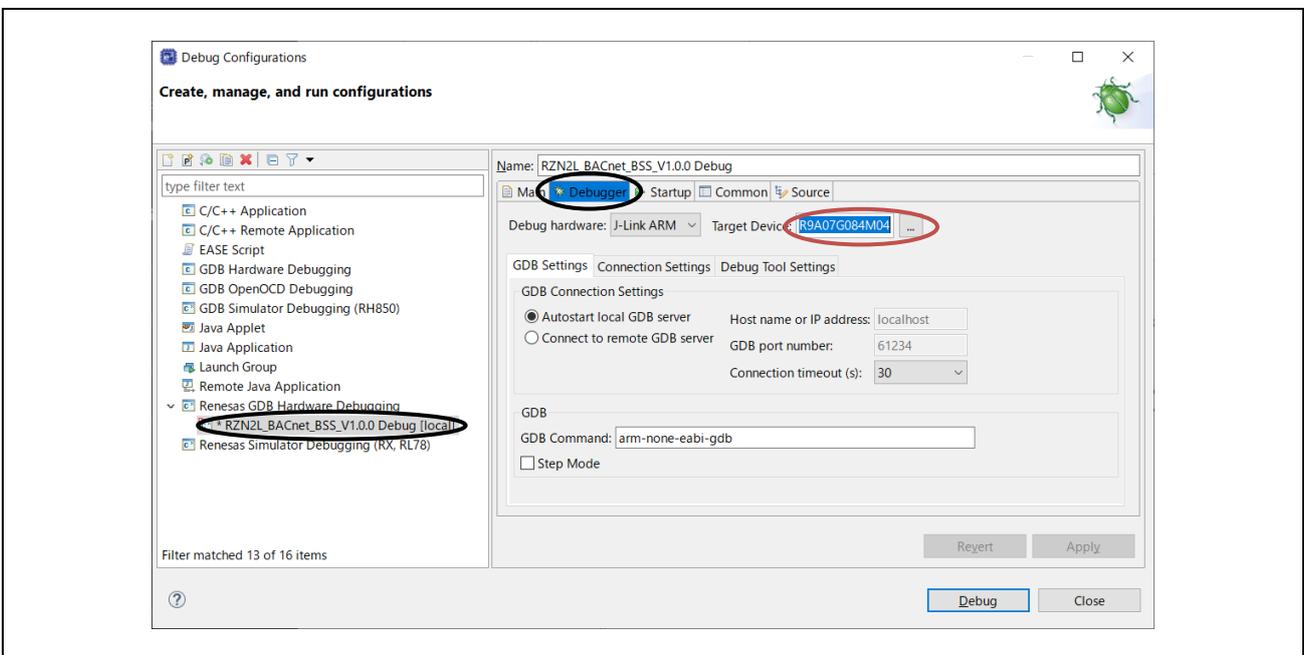


Fig.4-16 Debug Configurations(2)

R9A07G084M04 を選択して OK をクリックします。

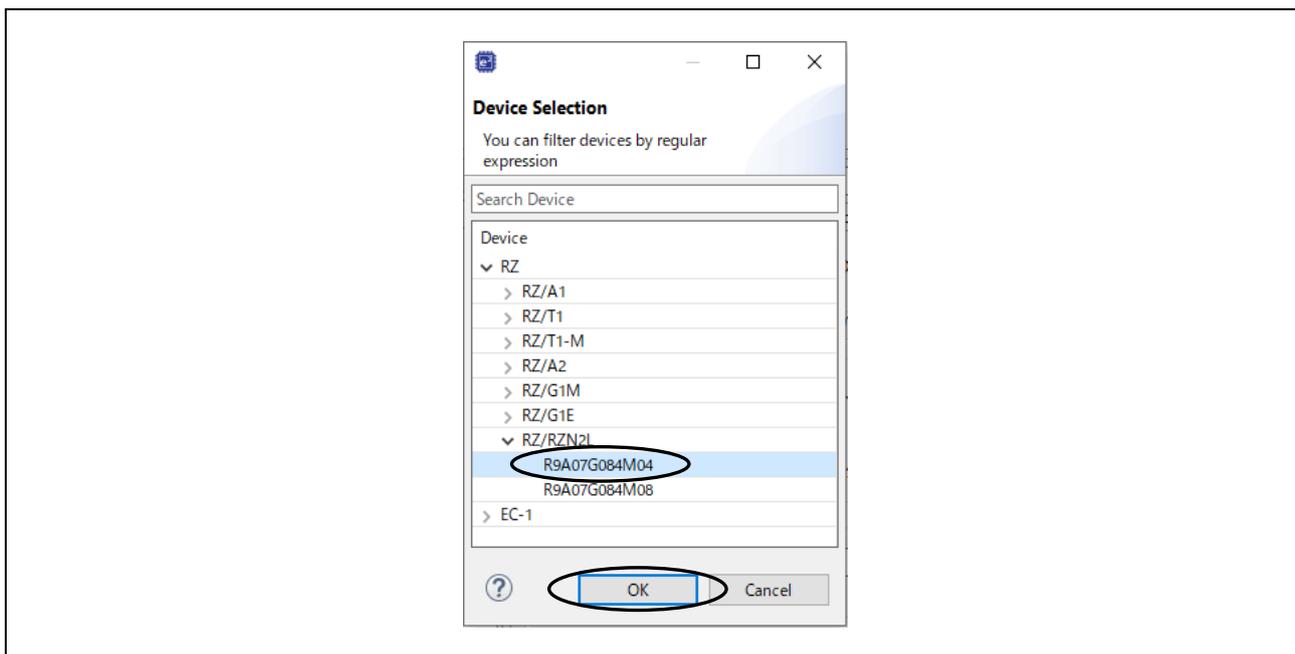


Fig.4-17 Debug Configurations(3)

c. デバッグ・ツール設定

デバッグ・ツール設定タグをクリックして、Operating Frequency [MHz]に 400 を入力します。

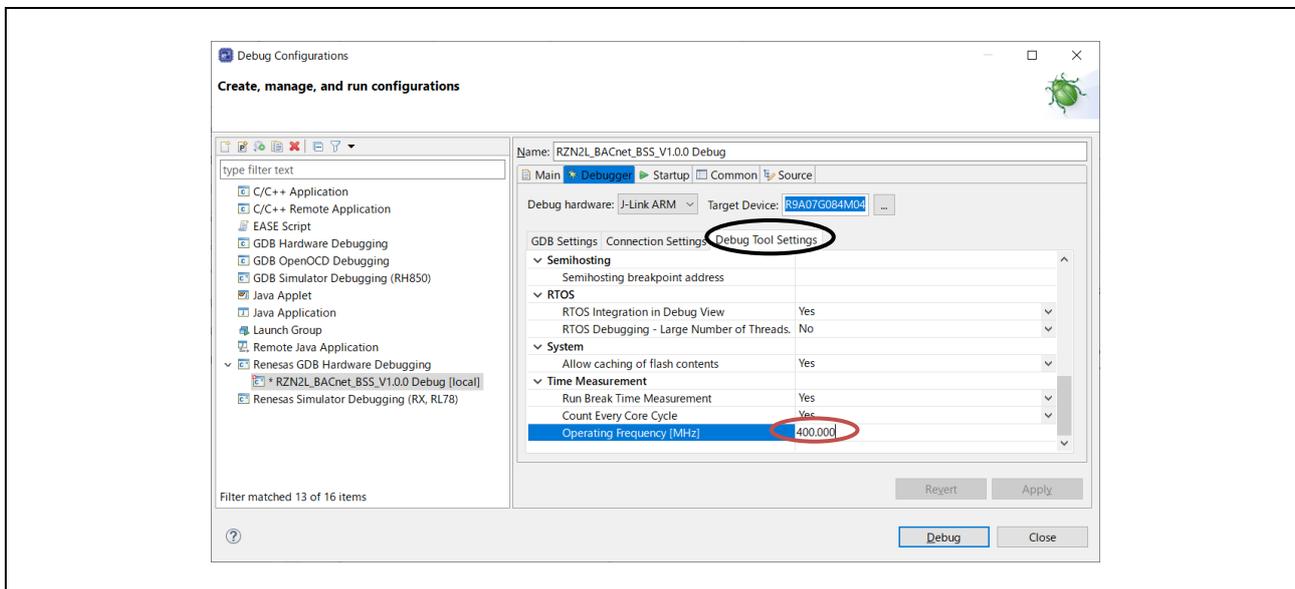


Fig.4-18 Debug Configurations(4)

d. マクロ登録

Startup タグをクリックして Run Commands に、source rzn2l_xspi0_x1_boot.cfg を入力し、Apply をクリックします。

Debug をクリックするとダウンロードを開始します。続けて Fig.4-21 の手順を参照してください。

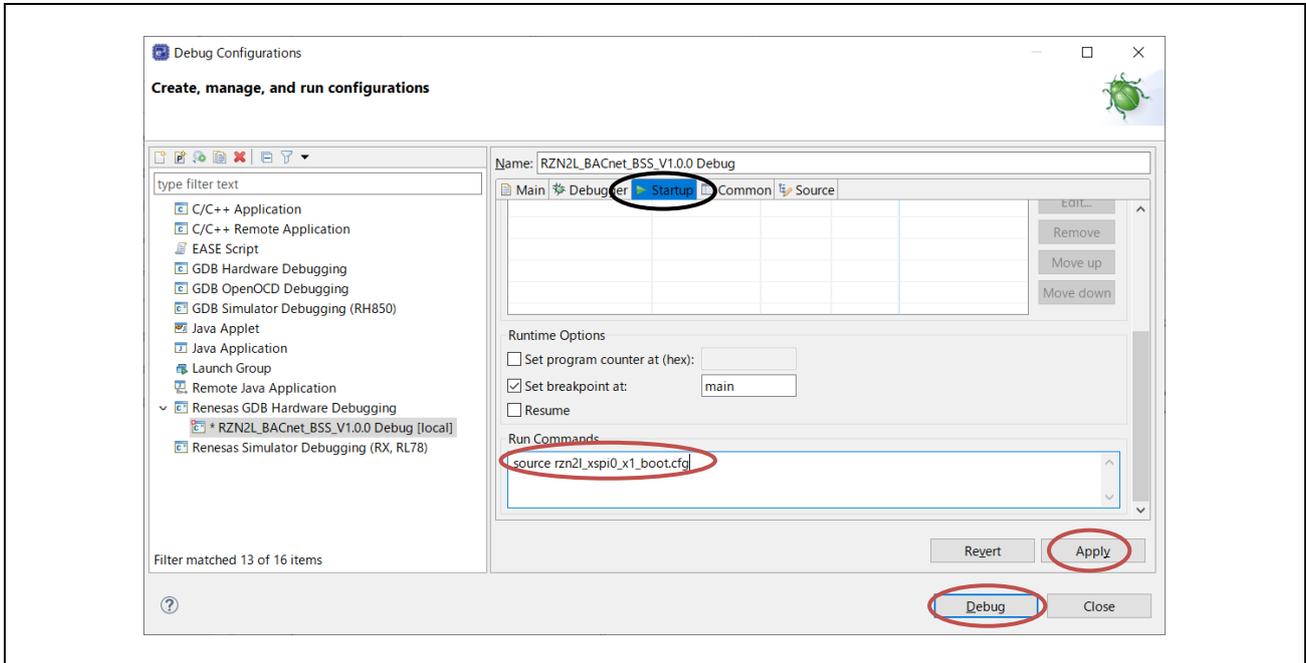


Fig.4-19 Debug Configurations(5)

4.4.4 デバッグ

ビルド終了後のダウンロード手順を以下に示します。

2 回目以降のデバッガ起動時は、C/C++ビューのプロジェクト名を選択した状態で、Run メニューをクリックします。Debug As にカーソルを置いて、Renesas GDB Hardware Debugging をクリックします。

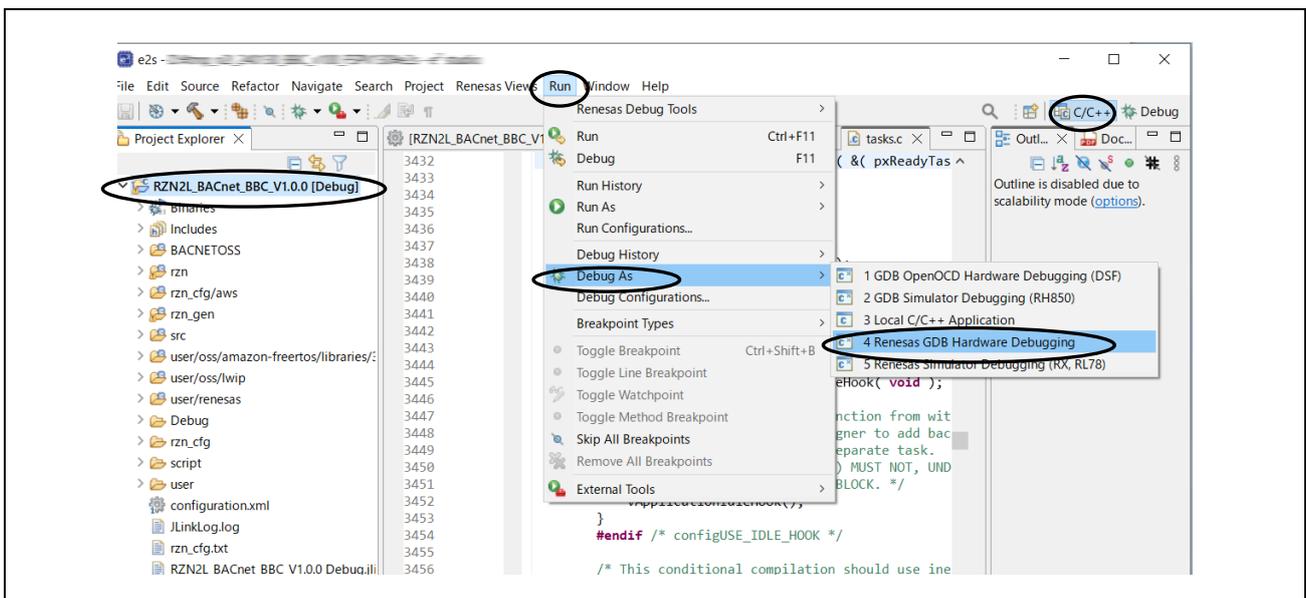


Fig.4-20 Run menu Debug As

シリアルフラッシュ ROM にプログラムをダウンロードします。

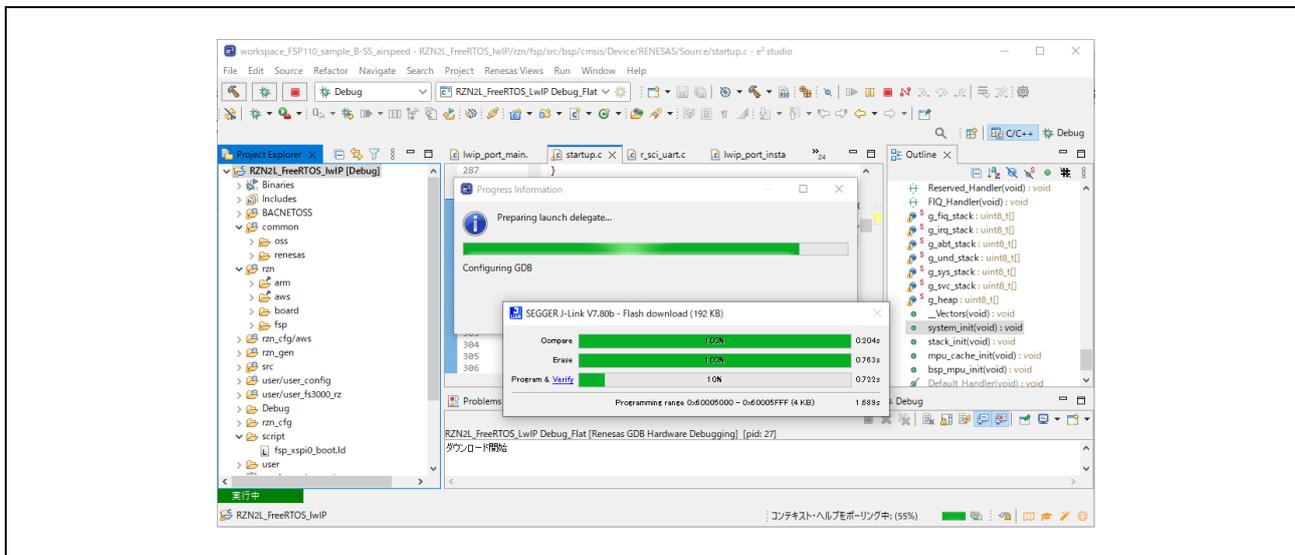


Fig.4-21 Download

- ・ デバッグビューへ切り替えるために Switch をクリックします。

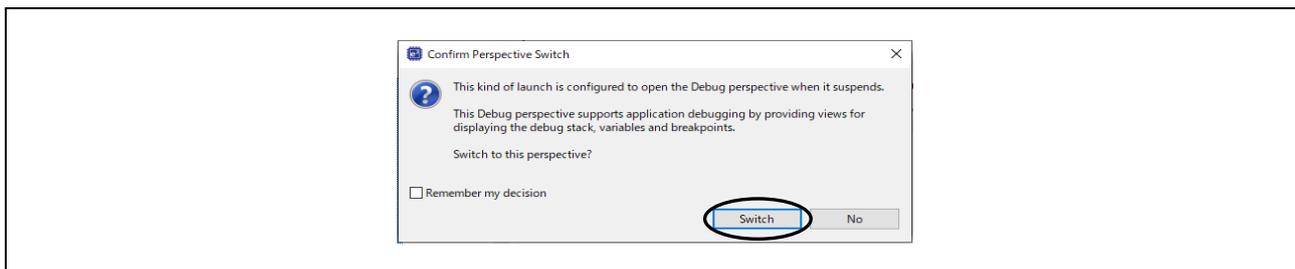


Fig.4-22 Perspective Switch

CPU はダウンロードデータに含まれるローダープログラムを自動的に BTCM メモリに展開します。展開後、ローダープログラム上の初期設定プログラム先頭にある `system_init()` で Break します。

※ デバッガを使用せず RSK ボード単独で動作させる場合は、ここでボードの電源を OFF し、デバッガケーブルを外してから電源を再度 ON してください。

デバッガを使用する場合は、Debug 画面に切り替わったら、まず **reset アイコン** をクリックしてから **resume** をクリックしてください。

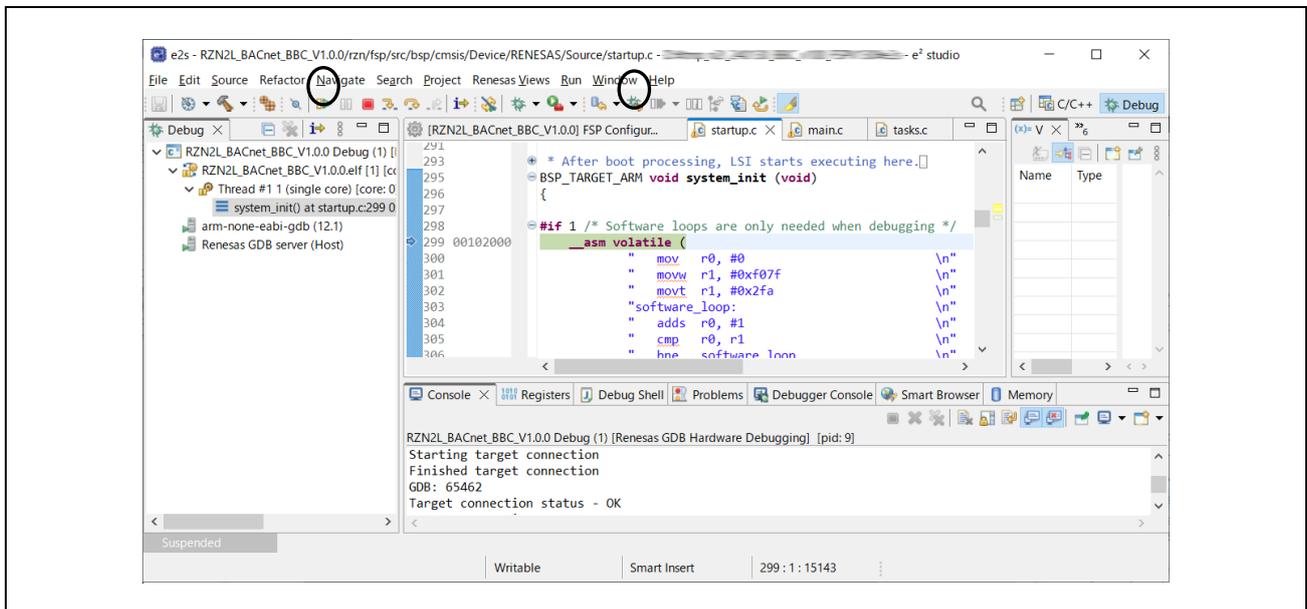


Fig.4-23 Break at system_init()

ローダープログラムは初期設定を終了すると main() の先頭で Break します。続けて resume をクリックしてプログラムを実行します。

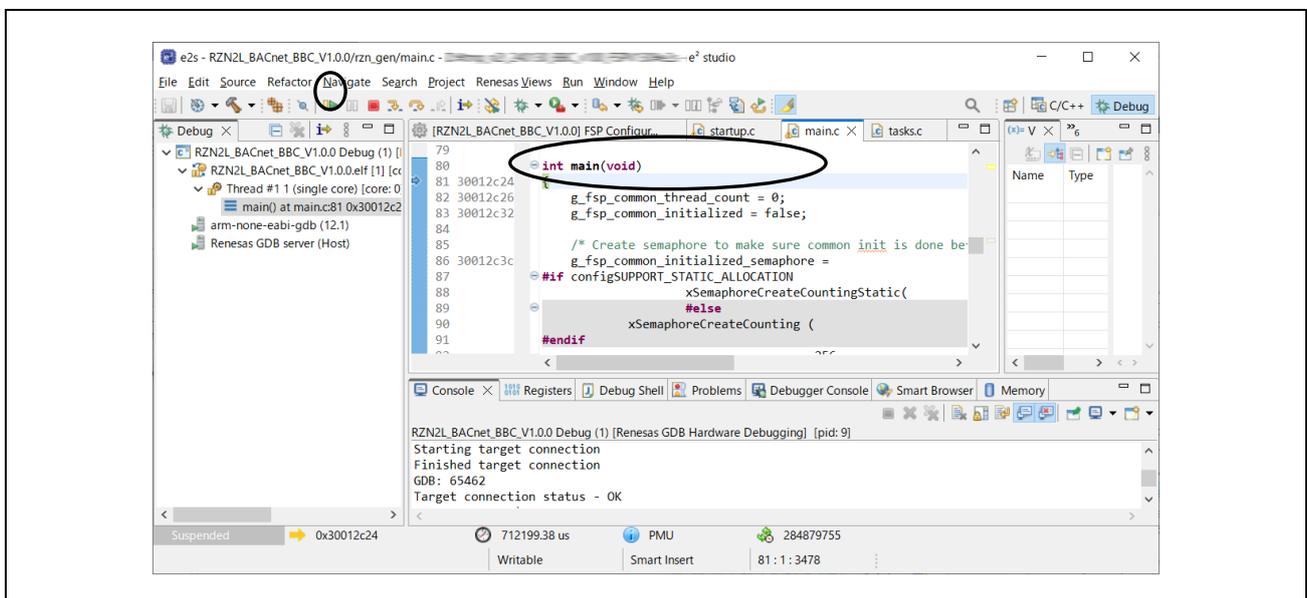


Fig.4-24 Break at main()

4.5 VTS による通信確認

以下では、VTS を起動する前に、Wireshark(イーサネット)を起動した状態を前提にします。

B-BC は初期設定終了後に自身の属する BACnet/IP ネットワークに対して I-Am-Router-To-Network および I-Am サービスフレームをブロードキャスト配信します。以下は Wireshark がこれらのパケットをキャプチャした様子を示しています。

Source の **192.168.10.10** は B-BC サーバーアドレスです。

Destination の **192.168.10.255** はブロードキャストアドレスを意味します。

BACnet の BIP は UDP 通信です。port はデフォルトの 47808(0xBAC0)を使用します。

Wireshark のフィルタは "bacnet"(下図では "bacnet or bvlc or ntp")を選択できます。

No.	Time	Source	Destination	Protocol	Length	Info
6	11:30:59.472922	192.168.10.10	192.168.10.255	BACnet-NPDU	60	I-Am-Router-To-Network
7	11:30:59.475052	192.168.10.10	192.168.10.255	BACnet-APDU	67	Unconfirmed-REQ i-Am device,10
25	11:31:29.239394	192.168.10.10	192.168.10.20	NTP	90	NTP Version 4, client
26	11:31:29.239596	192.168.10.20	192.168.10.10	NTP	90	NTP Version 3, server

Packet 7 details:

- Frame 7: 67 bytes on wire (536 bits), 67 bytes captured (536 bit)
- Ethernet II, Src: RenesasElect_10:f9:ed (74:90:50:10:f9:ed), Dst: ff:ff:ff:ff:ff:ff
- Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.10.255
- User Datagram Protocol, Src Port: 47808, Dst Port: 47808
- BACnet Virtual Link Control
- Building Automation and Control Network NPDU
- Building Automation and Control Network APDU
 - 0001 = APDU Type: Unconfirmed-REQ (1)
 - Unconfirmed Service Choice: i-Am (0)
 - ObjectIdentifier: device, 10
 - Maximum ADPU Length Accepted: (Unsigned) 480
 - Segmentation Supported: no-segmentation (3)
 - Vendor ID: Unknown Vendor (9999)

Hex dump:

```

0000 ff ff ff ff ff ff 74 90 50 10 f9 ed 08 00 45 00
0010 00 35 00 01 00 00 ff 11 25 5d c0 a8 0a 0a c0 a8
0020 0a ff ba c0 ba c0 00 21 ce 5d 81 0b 00 19 01 20
0030 ff ff 00 ff 10 00 c4 02 00 00 0a 22 01 e0 91 03
0040 22 27 0f
  
```

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

- ・ VTS の立ち上げ

VTS を展開したフォルダの VTS.exe をダブルクリックします。

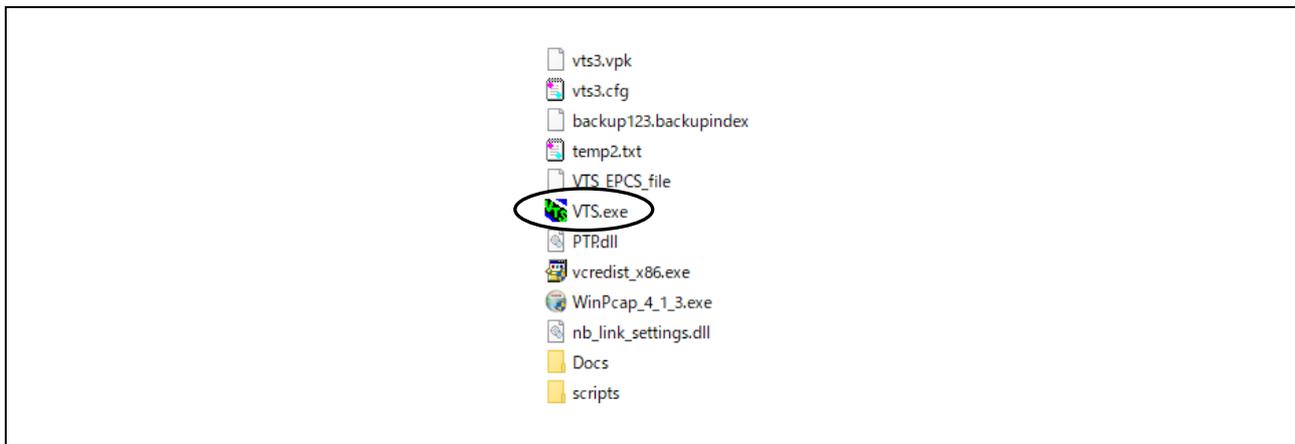


Fig.4-26 Launch VTS

次からの一連の表示内容は B-SS サンプルソフトの評価で使用了した設定内容です。
使用環境に応じて変更してください。

Edit メニューの Devices... をクリックします。

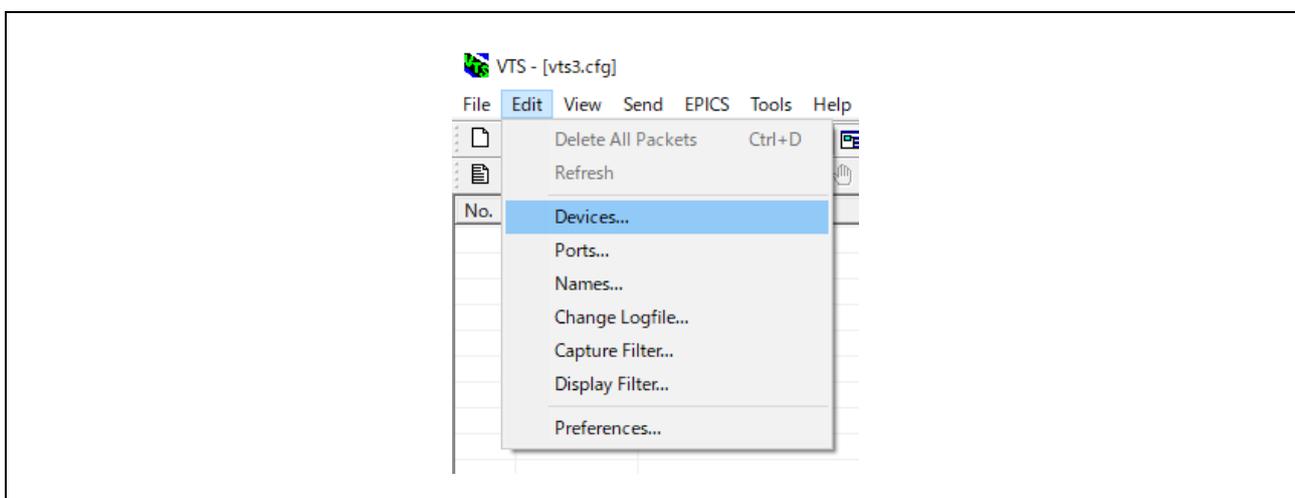


Fig.4-27 Edit menu Devices(1)

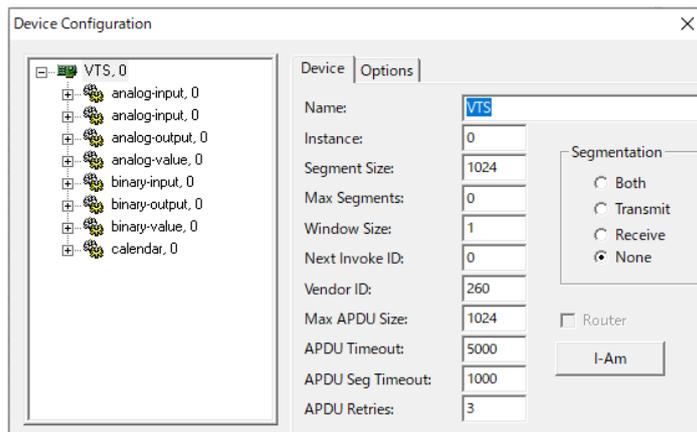


Fig.4-28 Edit menu Devices(2)

EditメニューのPorts...をクリックします。

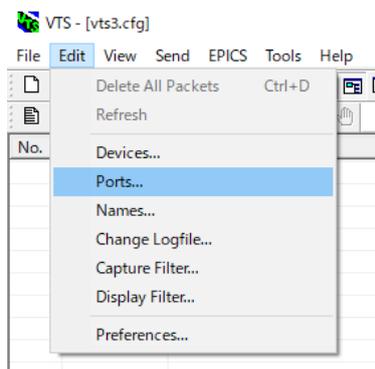


Fig.4-29 Edit menu Ports(1)

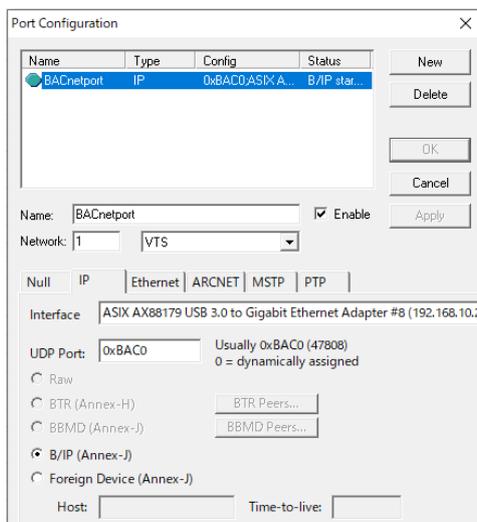


Fig.4-30 Edit menu Ports(2)

Edit メニューの Names... をクリックします。

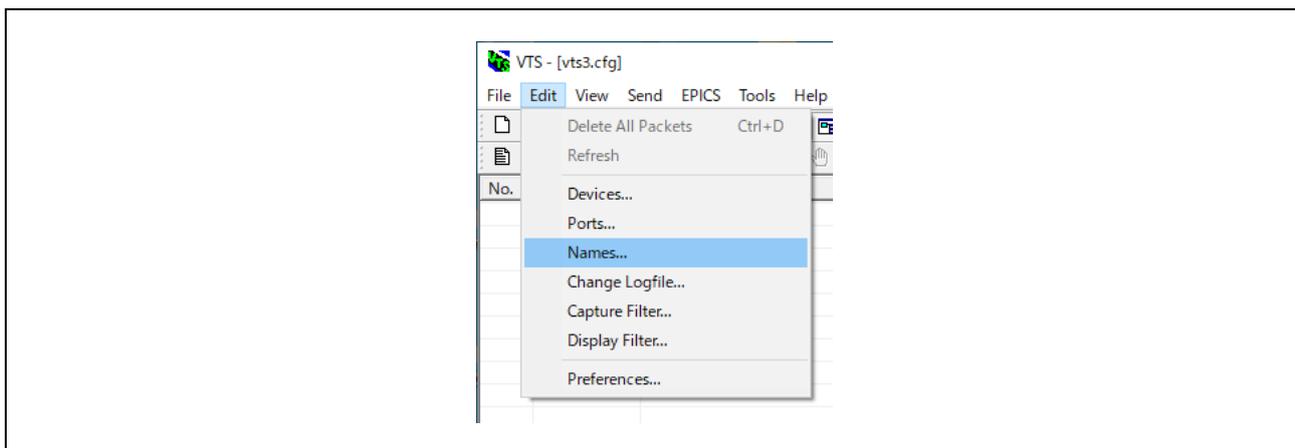


Fig.4-31 Edit menu Names(1)

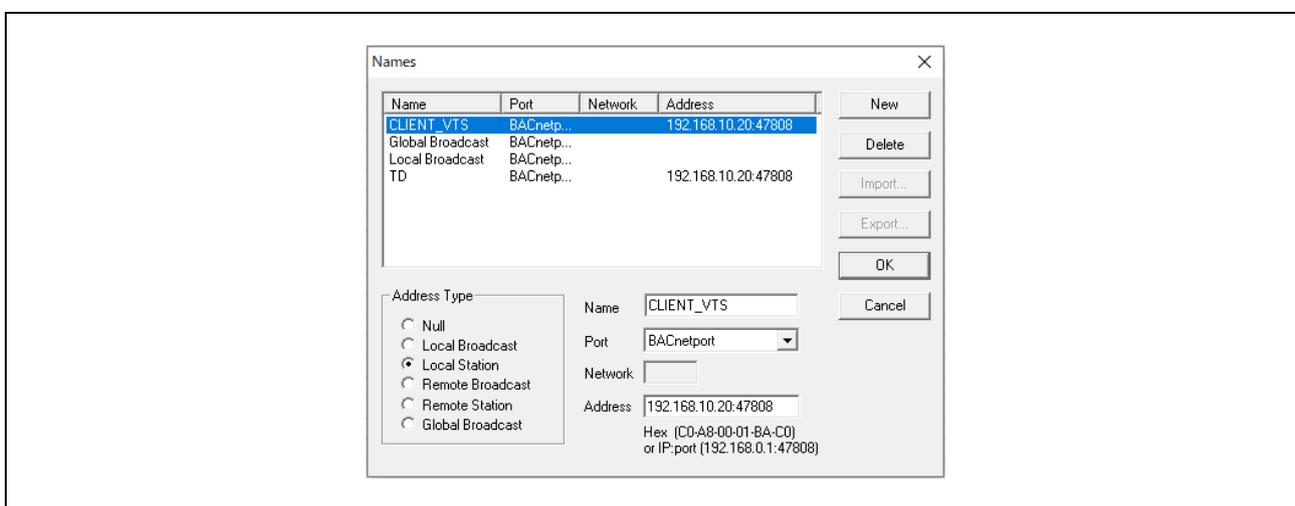


Fig.4-32 Edit menu Names(2)

4.5.1 Who-Is と I-Am

(1) Discover Devices

VTS の Tools メニューにある Discover Devices... をクリックします。

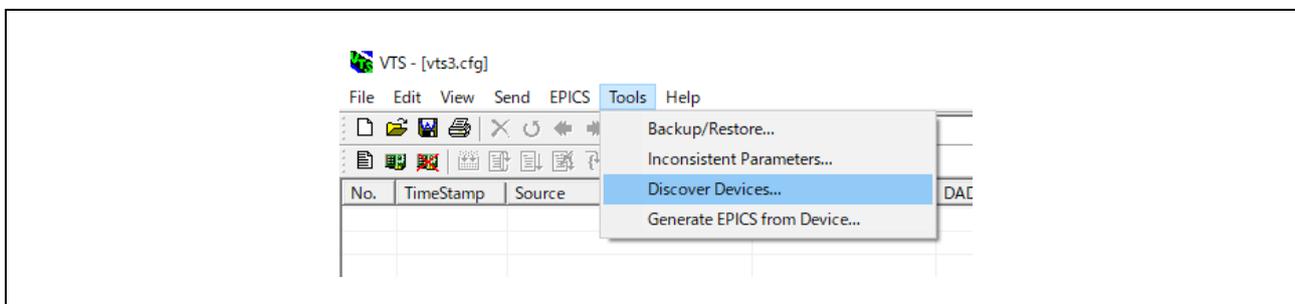


Fig.4-33 Tools menu Discover Devices(1)

表示されるダイアログの Device: を Global Broadcast に選択して、OK をクリックします。

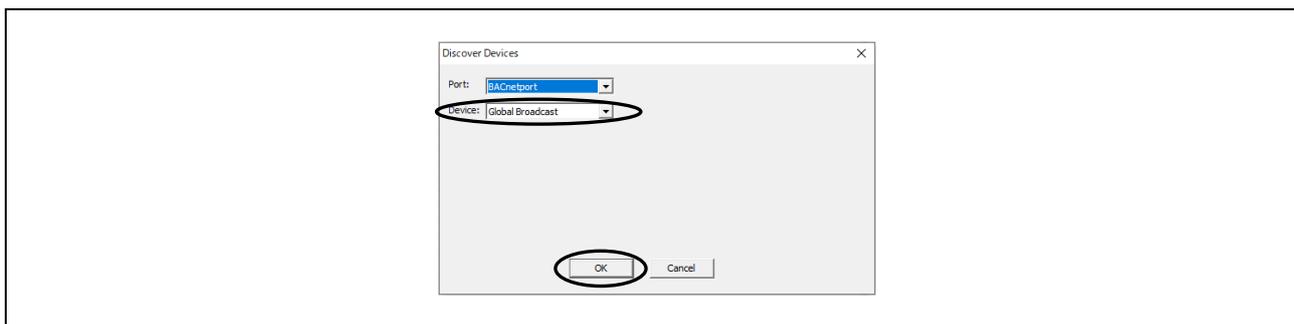


Fig.4-34 Tools menu Discover Devices(2)

次の Wireshark キャプチャ画面は VTS クライアントからの Who-Is サービス要求および Who-Is-Router-To-Network ネットワークレイヤメッセージに対して B-BC サーバーが I-Am device,10 および I-Am-Router-To-Network の応答を返した様子です。VTS から最初の Who-Is はすべてのデバイス ID(0~4194303)に対して行われ、3つのデバイスが I-Am 応答しています。I-Am device,0 は VTS 自身を示し、I-Am device,100 は MS/TP ネットワークに接続された B-SS です。I-Am device,100 の送信元 IP アドレスは B-BC の IP アドレスを経由したことが分かります。

VTS クライアントは B-BC と B-SS に対して ReadProperty サービスによって 4 種類のプロパティ値を要求しています。B-BC と B-SS は結果を含む Complex-Ack を応答しています。

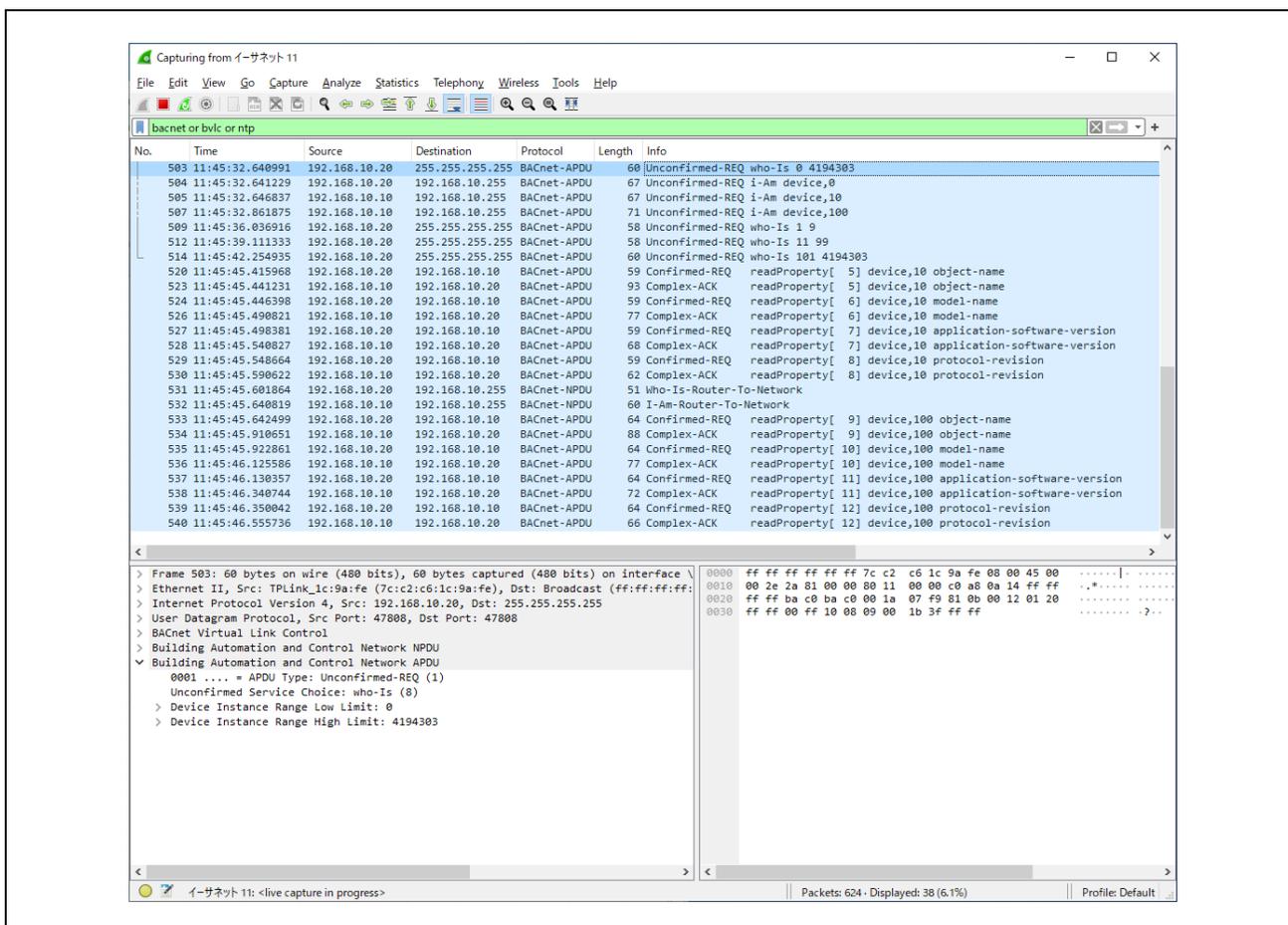


Fig.4-35 Tools menu Discover Devices(2)

VTS は送受信フレームを次のとおり表示します。表示内容は Wireshark と同様です。

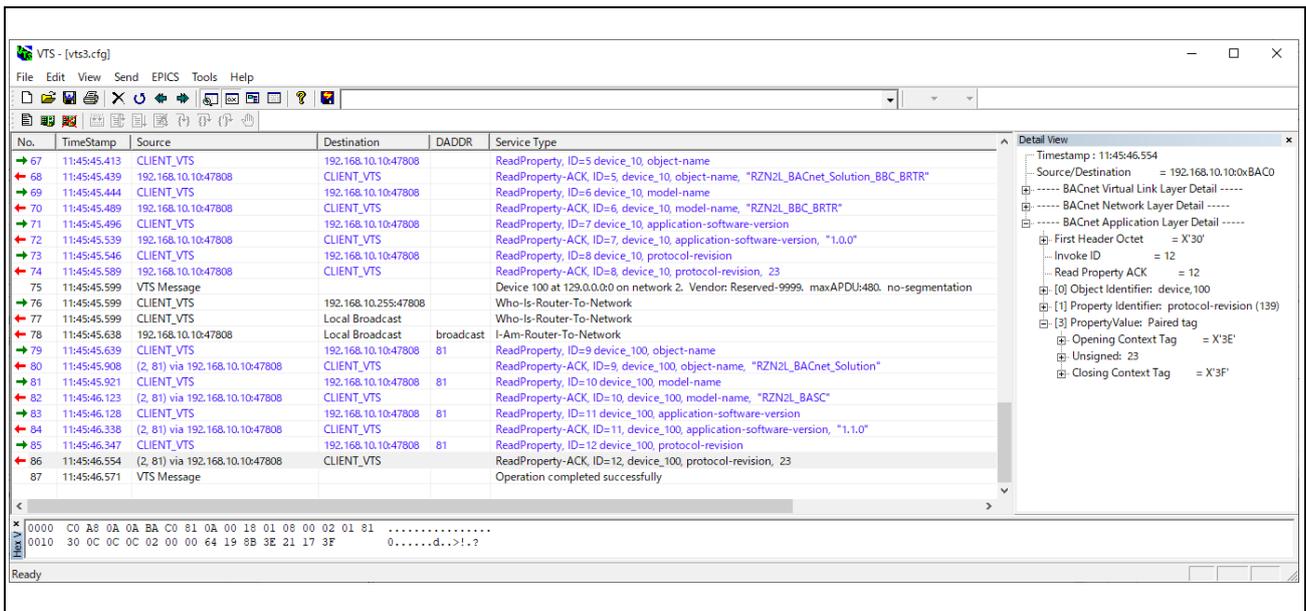


Fig.4-36 VTS log screen

(2) Remote Device Management

Tools メニューから Who-Is を送信する以外に Send メニューからも送信できます。通常のサービス要求は、Send メニューから行います。Remote Device Management > Who-Is をクリックします。

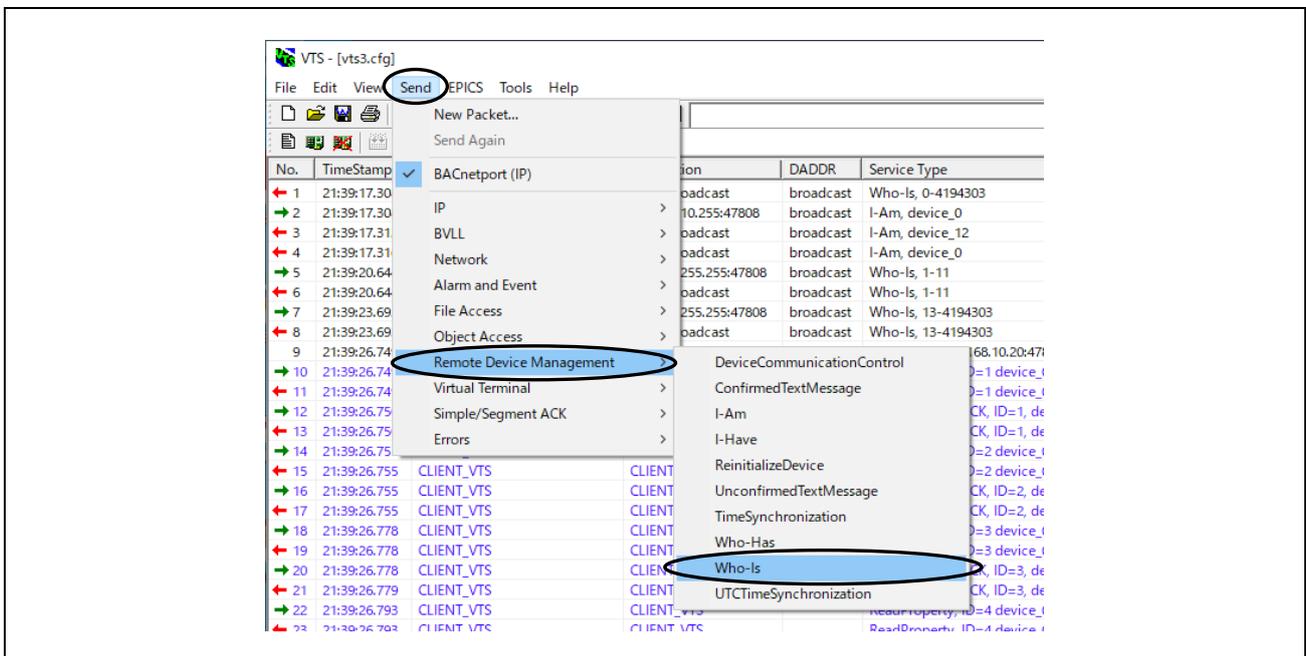


Fig.4-37 Send menu Who-Is

Who-Is ダイアログの IP タグをクリックし、Global Broadcast または Local Broadcast を選択するとアドレスが自動入力されます。

Global Broadcast:255.255.255.255:47808

Local Broadcast:192.168.10.255:47808

アドレスを直接入力することもできます。

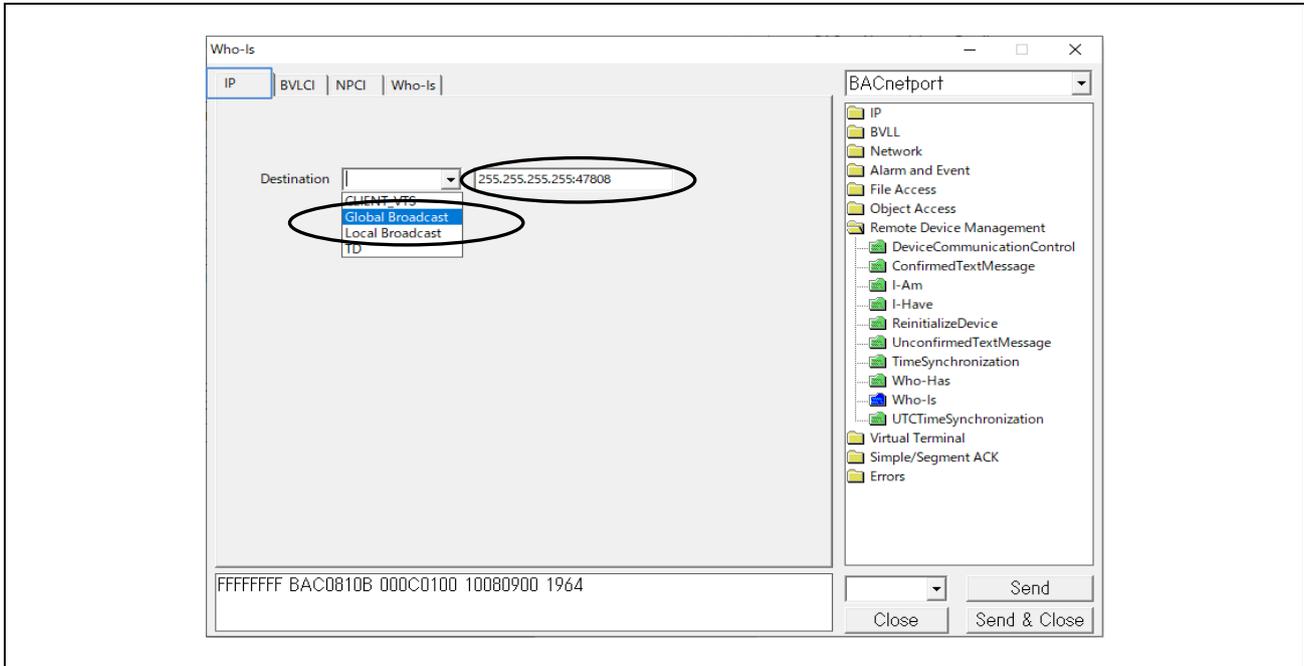


Fig.4-38 Who-Is parameters(1)

Who-Is ダイアログの BVLCI タグをクリックし、Original Uncast または Original Broadcast のどちらかが選択されていることを確認します。

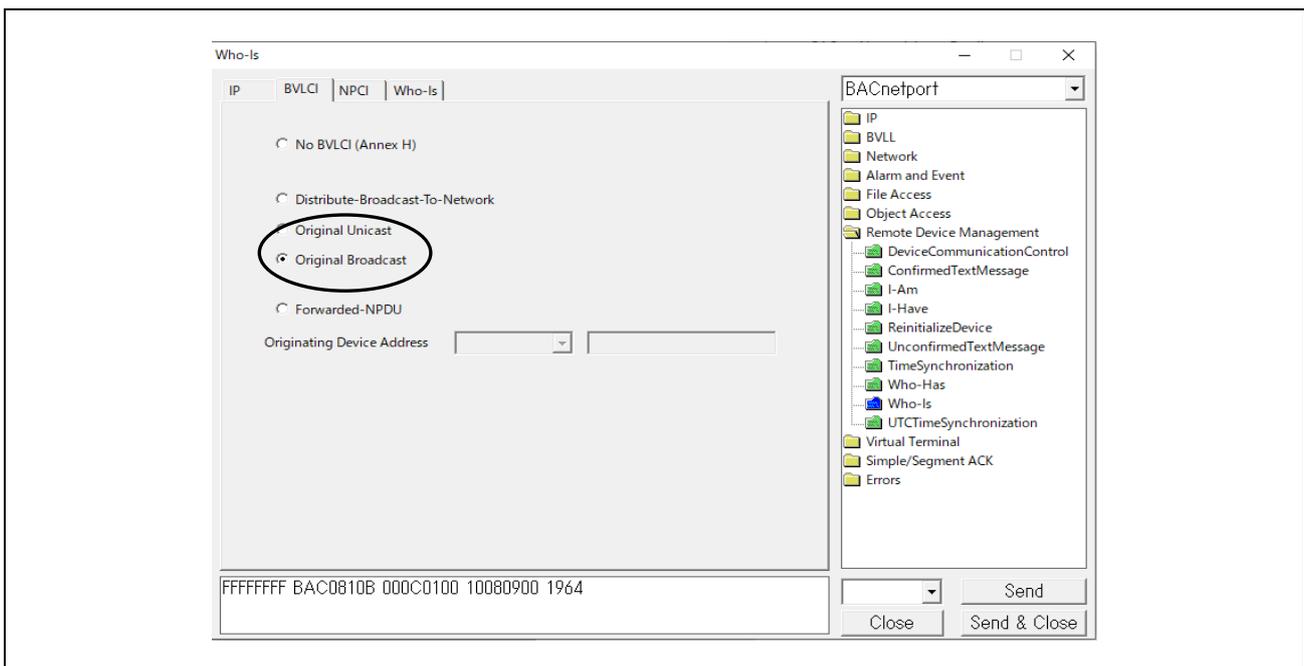


Fig.4-39 Who-Is parameters(2)

Who-Is ダイアログの NPCI タグをクリックし、DNET/DLEN/DADR Present が選択されていることを確認します。

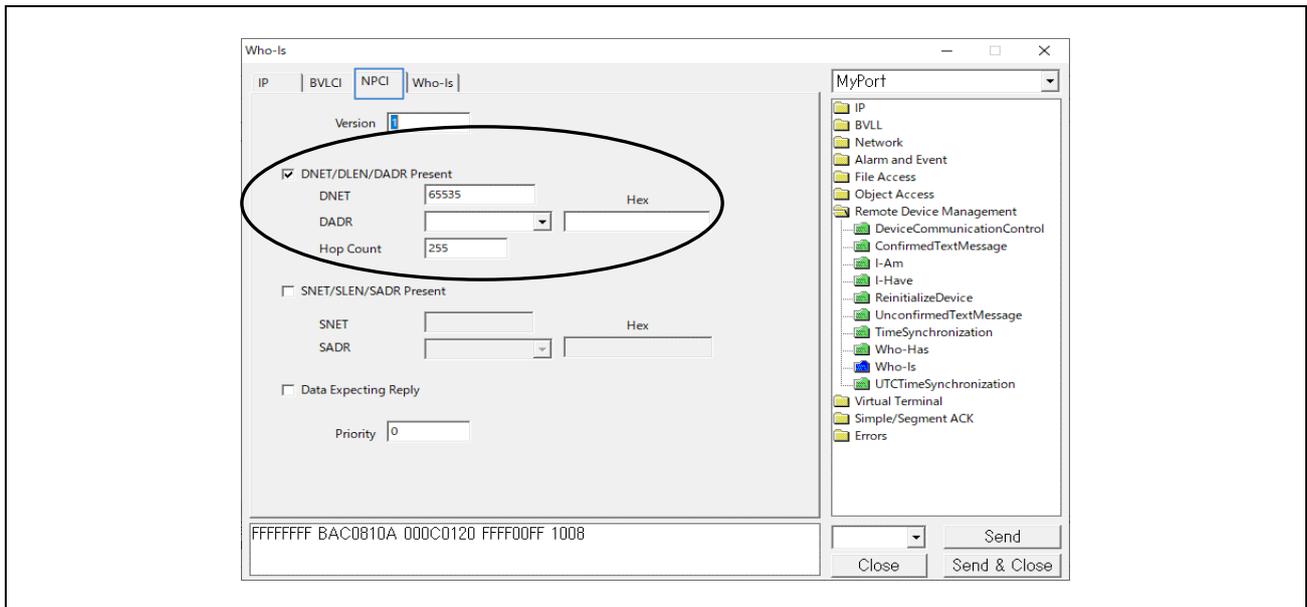


Fig.4-40 Who-Is parameters(3)

Who-Is ダイアログの Who-Is タグをクリックし、探索するデバイスの ID 範囲を入力します。空欄にすると全範囲の 0~4194303 になります。最後に Send をクリックします。

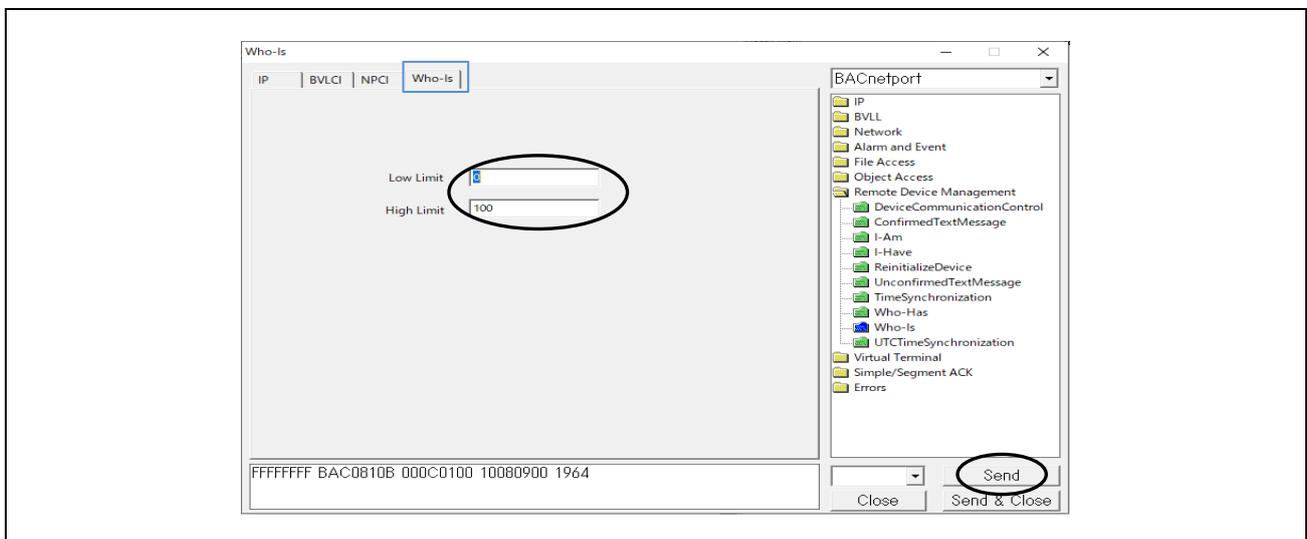


Fig.4-41 Who-Is parameters(4)

Send メニューからの Who-Is サービス要求に対する B-BC および B-SS の I-Am 応答キャプチャ結果を示します。

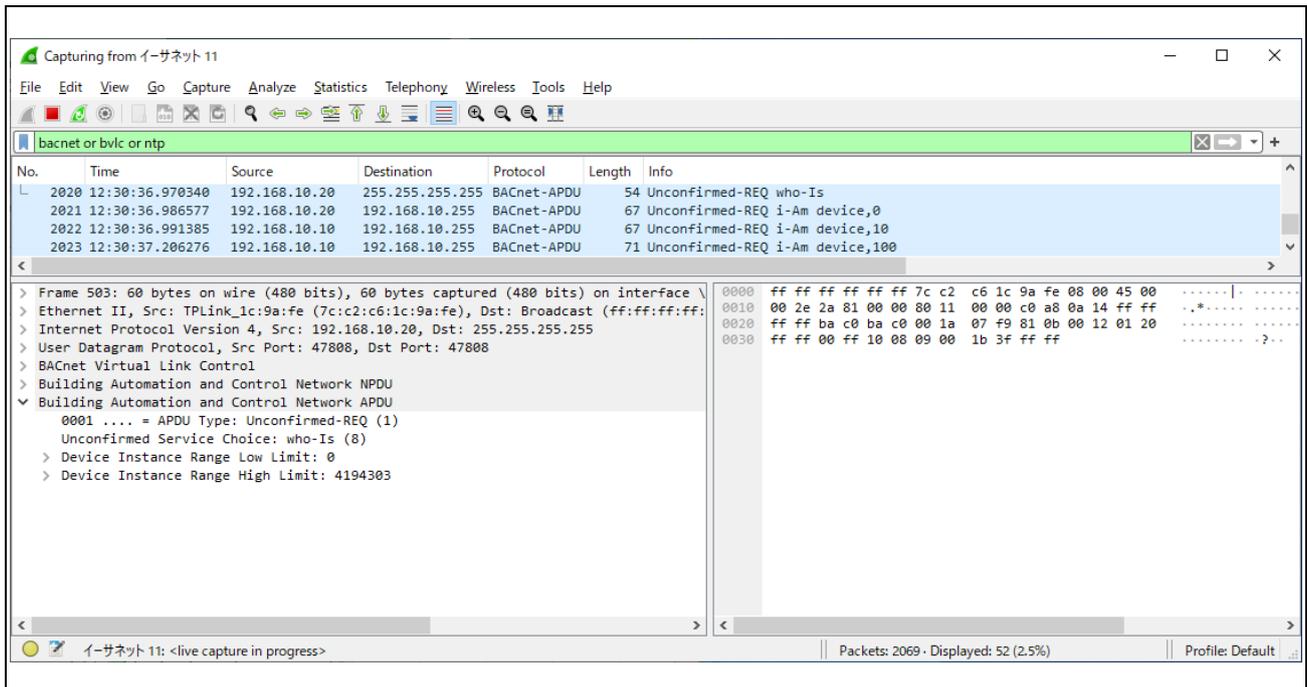


Fig.4-42 Capture Who-Is and I-Am

4.5.2 ReadProperty

Send メニューから Object Access > ReadProperty をクリックします。

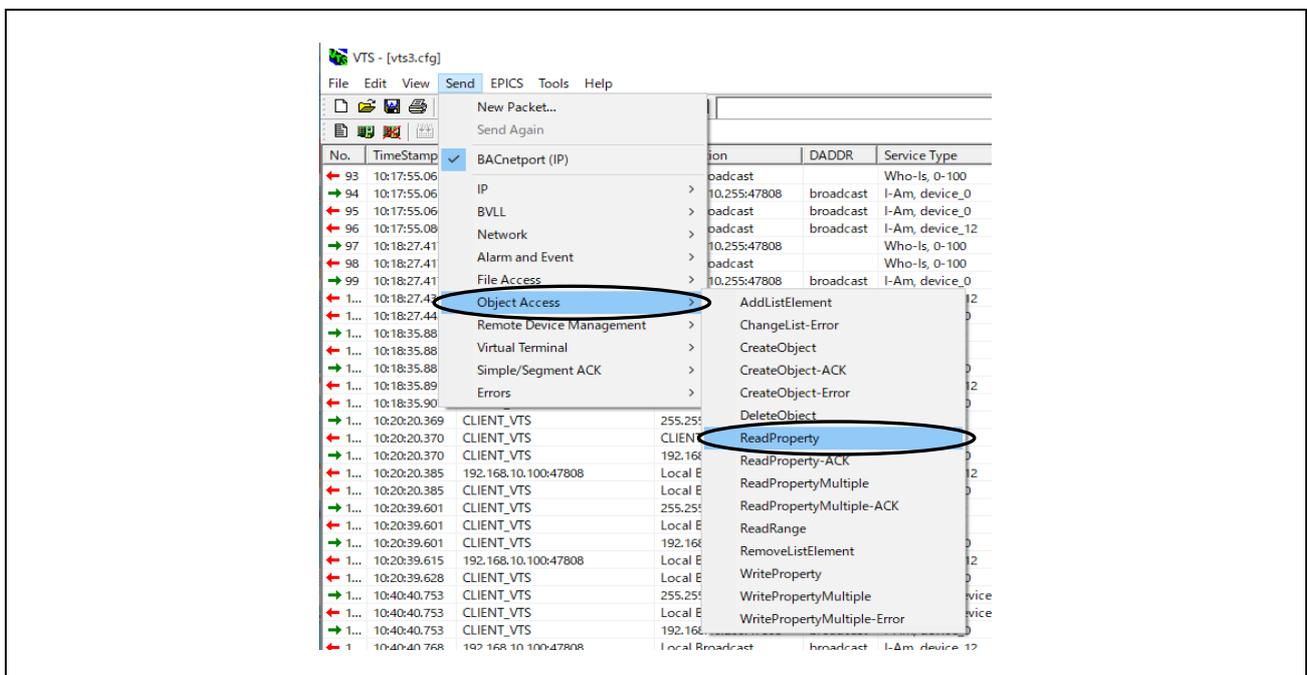


Fig.4-43 Send menu ReadProperty

ReadProperty ダイアログの IP タグをクリックし、B-BC サーバーアドレスを直接入力します。

192.168.10.10:47808

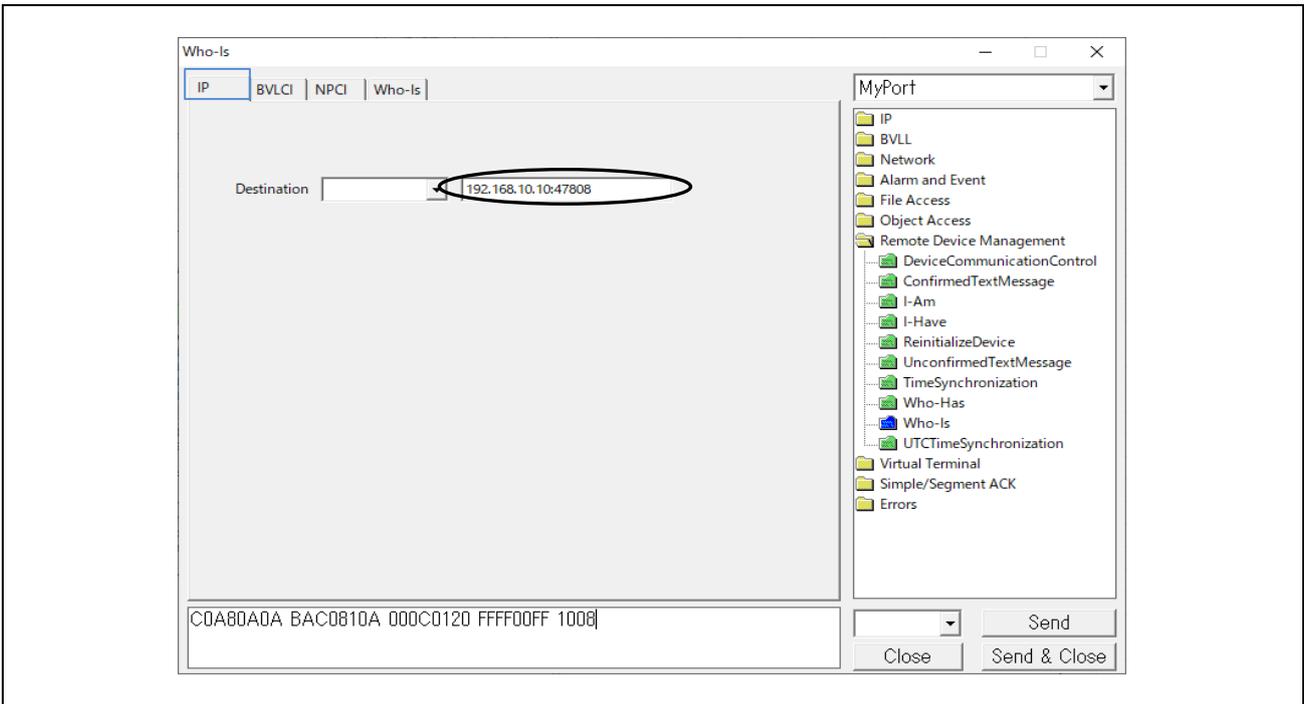


Fig.4-44 IP tag parameters

ReadProperty ダイアログの NPCI タグをクリックし、このダイアログで B-BC か B-SS を選択します。

B-BC を選択する場合は Fig.4-45 に示します。

Data Expecting Reply が✓されていることを確認します。

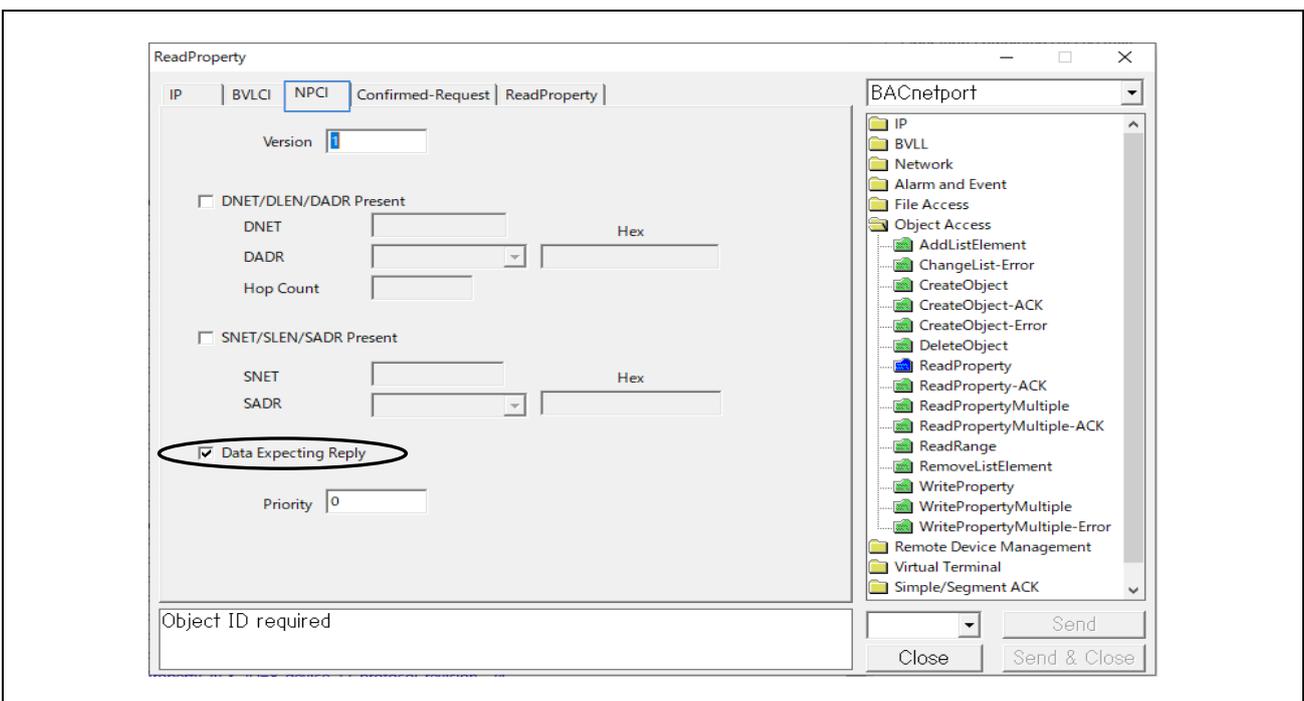


Fig.4-45 NPCI tag parameters to B-BC

B-SS を選択する場合は Fig.4-46 に示します。DNET に B-SS の Network_Number プロパティ値 2 を入力します。DADR に B-SS の MAC アドレス 129 を Hex で入力(81)します。Hop Count に 255 を入力します。B-SS に要求するときの NPCI ダイアログ設定は ReadProperty 以外の各サービスについても同様です。

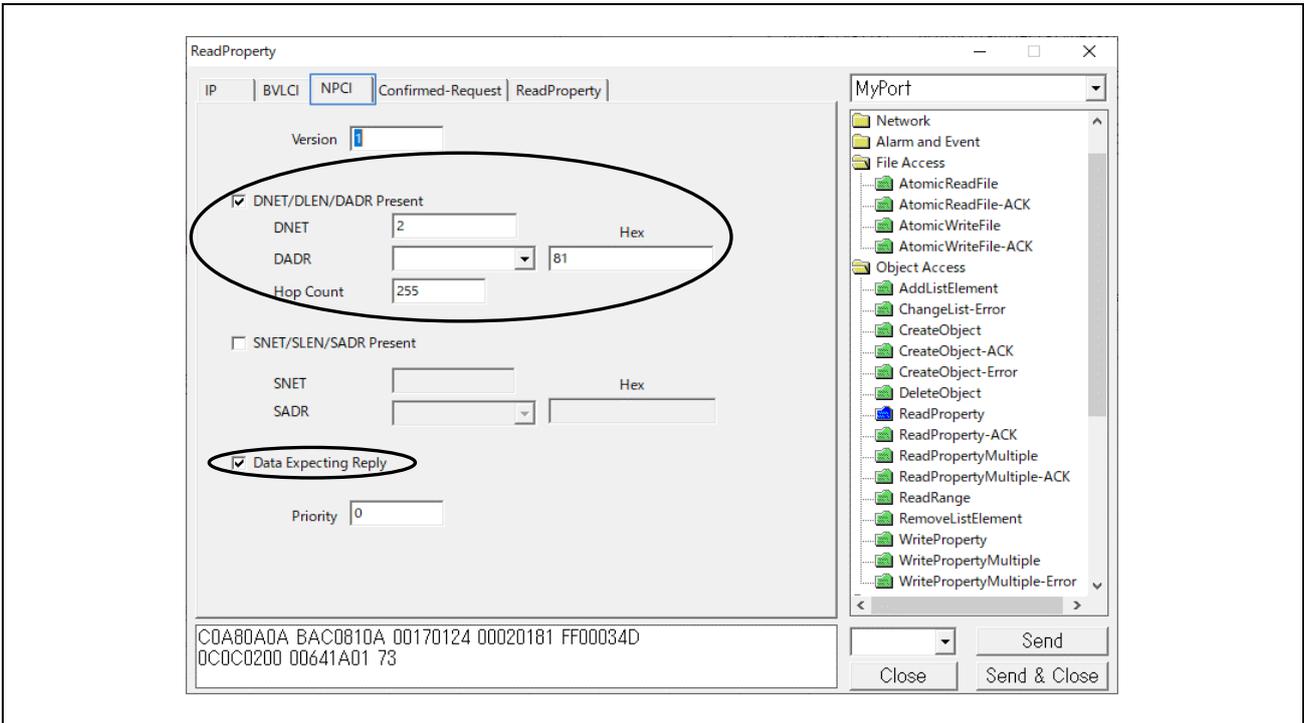


Fig.4-46 NPCI tag parameters to B-SS

ReadProperty ダイアログの Confirmed-Request タグをクリックし、Max APDU length accepted は 480 を選択します。

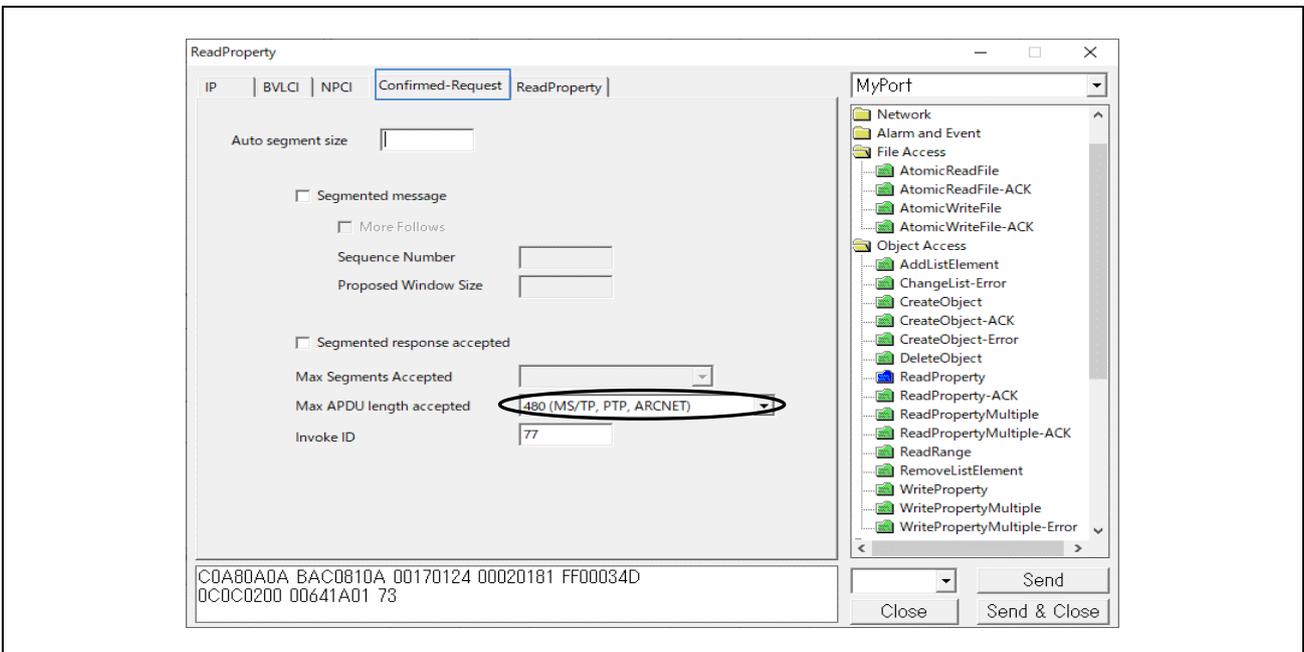


Fig.4-47 Confirmed Request tag parameters

ReadProperty ダイアログの ReadProperty タグをクリックし、ID... > Object Type を選択し、その Instance を入力します。例では B-SS の場合で device,100 を入力していますが B-BC を選択した場合は device,10 を入力します。

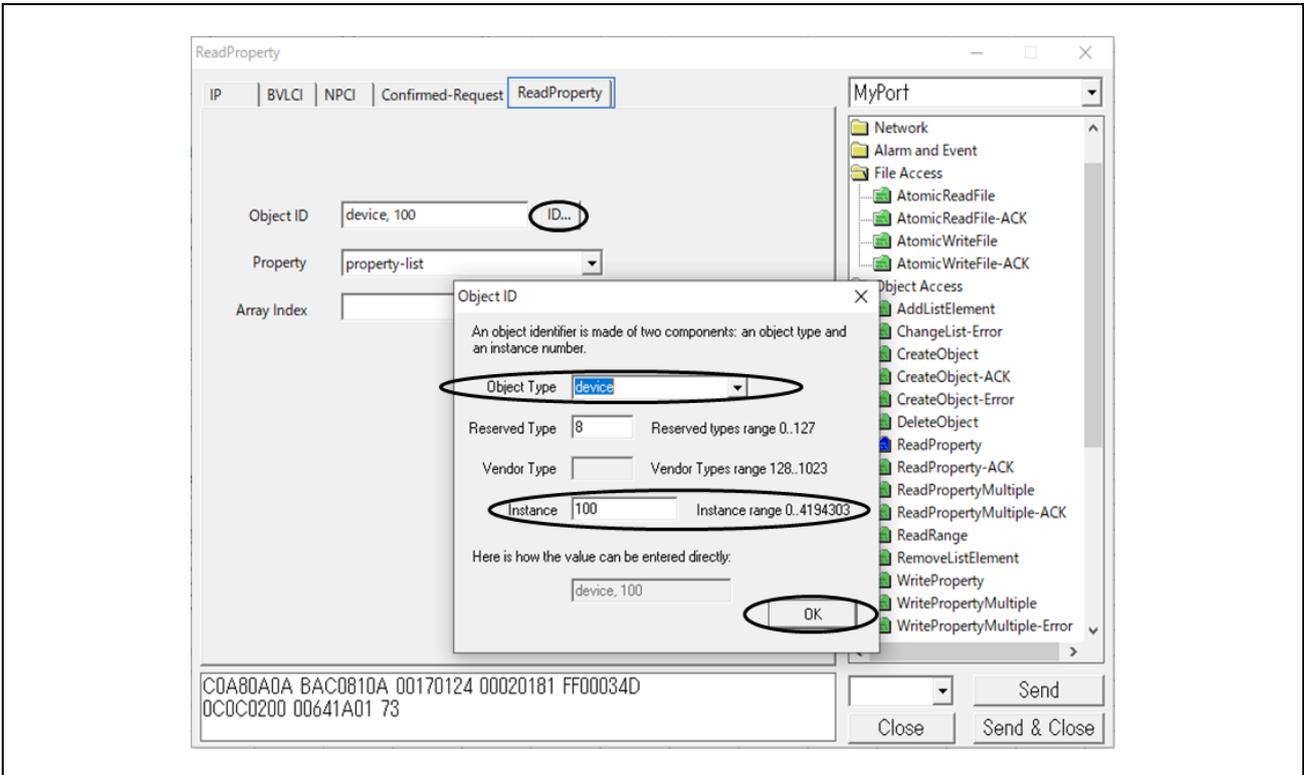


Fig.4-48 ReadProperty parameters

続けて Property を選択します。例は property-list を設定しています。最後に Send をクリックします。

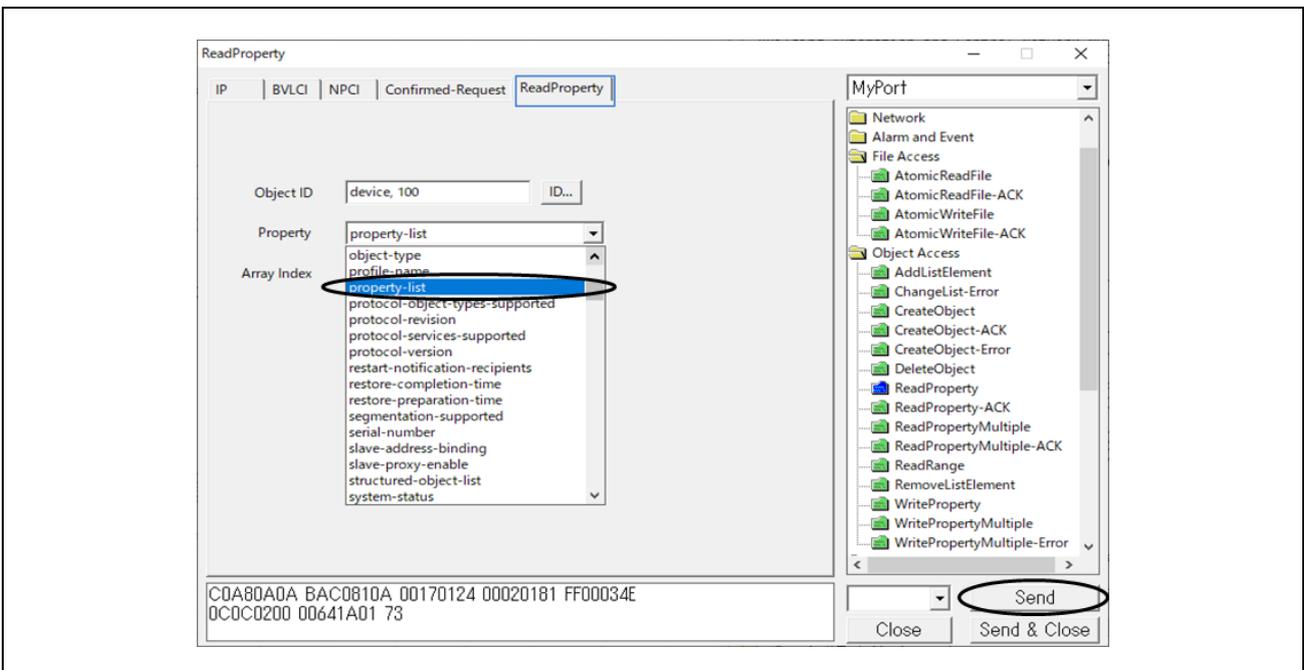


Fig.4-49 ReadProperty parameters(2)

B-SS は device,100 オブジェクトの property-list プロパティを Complex-Ack で応答しています。

The image shows a Wireshark capture window titled "Capturing from イーサネット 11". The filter is set to "bacnet or bvlc or ntp". The packet list shows two packets:

No.	Time	Source	Destination	Protocol	Length	Info
3628	13:19:06.277040	192.168.10.20	192.168.10.10	BACnet-APDU	65	Confirmed-REQ readProperty[77] device,100 property-list
3631	13:19:06.572567	192.168.10.10	192.168.10.20	BACnet-APDU	113	Complex-ACK readProperty[77] device,100 property-list

The packet details for packet 3631 are expanded to show the following structure:

- Frame 3631: 113 bytes on wire (904 bits), 113 bytes captured (904 bits) on interface
- Ethernet II, Src: RenesasElect_10:f9:ed (74:90:50:10:f9:ed), Dst: TPLink_1c:9a:...
- Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.10.20
- User Datagram Protocol, Src Port: 47808, Dst Port: 47808
- BACnet Virtual Link Control
- Building Automation and Control Network NPDU
- Building Automation and Control Network APDU
 - 0011 = APDU Type: Complex-ACK (3)
 - ... 0000 = PDU Flags: 0x0
 - Invoke ID: 77
 - Service Choice: readProperty (12)
 - ObjectIdentifier: device, 100
 - Property Identifier: property-list (371)
 - {[3]
 - property-list: system-status (112)
 - property-list: vendor-name (121)
 - property-list: vendor-identifier (120)
 - property-list: model-name (70)
 - property-list: firmware-revision (44)
 - property-list: application-software-version (12)
 - property-list: protocol-version (98)
 - property-list: protocol-revision (139)
 - property-list: protocol-services-supported (97)
 - property-list: protocol-object-types-supported (96)
 - property-list: object-list (76)
 - property-list: max-apdu-length-accepted (62)
 - property-list: segmentation-supported (107)
 - property-list: apdu-timeout (11)
 - property-list: number-of-APDU-retries (73)
 - property-list: device-address-binding (30)
 - property-list: database-revision (155)
 - property-list: description (28)
 - property-list: local-time (57)
 - property-list: utc-offset (119)
 - property-list: local-date (56)
 - property-list: daylight-savings-status (24)
 - property-list: location (58)
 - property-list: active-cov-subscriptions (152)
 - }] [3]

The packet bytes pane shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates "Packets: 4024 · Displayed: 56 (1.4%) | Profile: Default".

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

4.5.3 TimeSynchronization / UTCTimeSynchronization

前回使用したダイアログに表示されているサービスツリーから Remote Device Management をダブルクリックします。

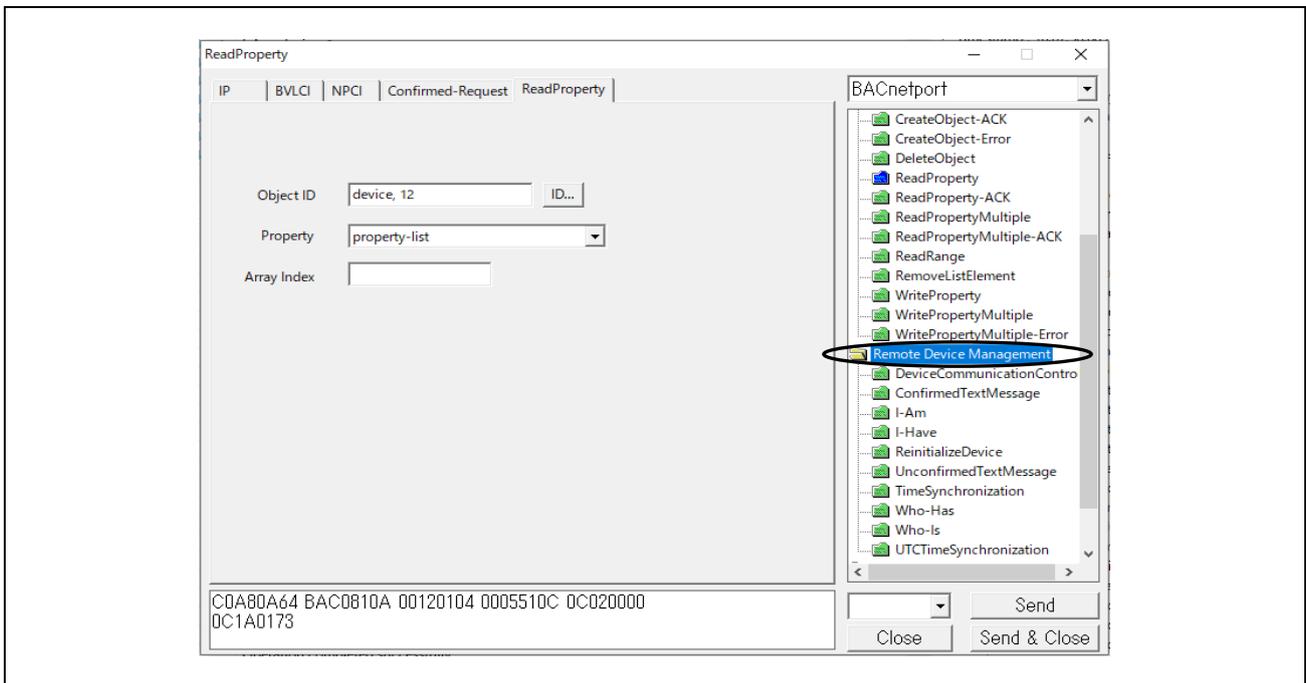


Fig.4-51 TimeSynchronization parameters(1)

展開された Remote Device Management の TimeSynchronization(または UTCTimeSynchronization)をクリックします。TimeSynchronization タグの Synchronize with VTS に✓を入れて Send をクリックします。

現在時刻以外を入力する場合は Synchronize with VTS の✓を外して Date と Time を直接入力します。

送信先 IP アドレスは前回値が適用されます。

送信先 IP アドレスを変更する場合は IP タグから IP アドレスを入力してください。以下同様です。

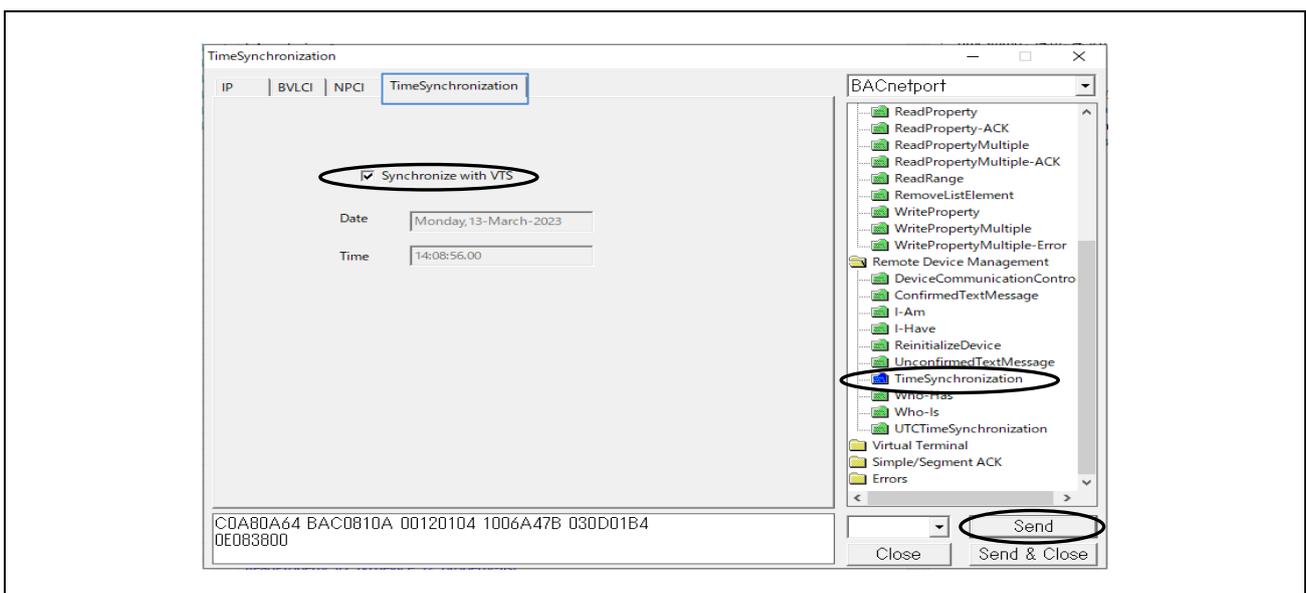


Fig.4-52 TimeSynchronization parameters(2)

TimeSynchronization は確認なし型サービスです。ただし送信先 IP アドレスはユニキャストでも問題ありません。以下のキャプチャ画面は VTS クライアントから B-BC サーバーにユニキャストした場合を示しています。

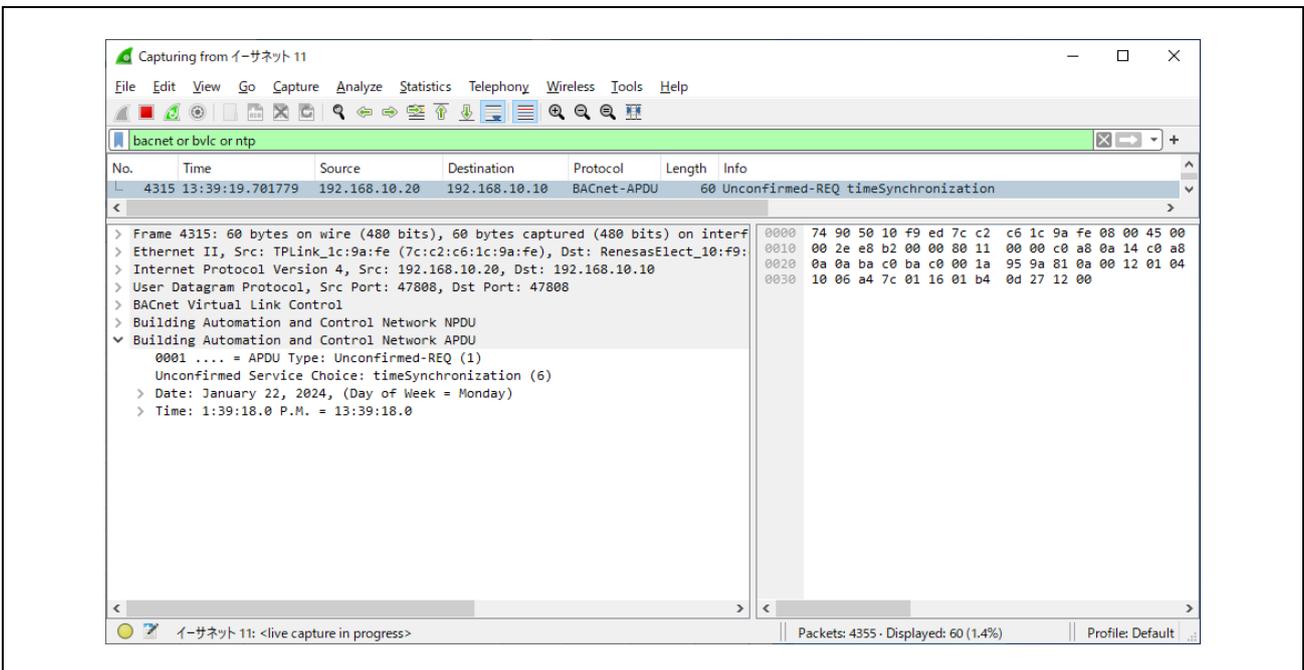


Fig.4-53 Capture TimeSynchronization

TimeSynchronization サービスの代わりに UTCTimeSynchronization サービスを送信する場合も同様です。

4.5.4 Who-Has と I-Have

前回使用したダイアログに表示されているサービスツリーから Remote Device Management > Who-Has をクリックします。探索 ID 範囲の Low Limit と High Limit を入力します。Object ID プルダウンメニューからオブジェクトタイプを選択し、インスタンス番号を入力後、OK をクリックします。

例では analog-input,0 オブジェクトを選択しています。Object ID と Object Name はどちらかしか選択できません。

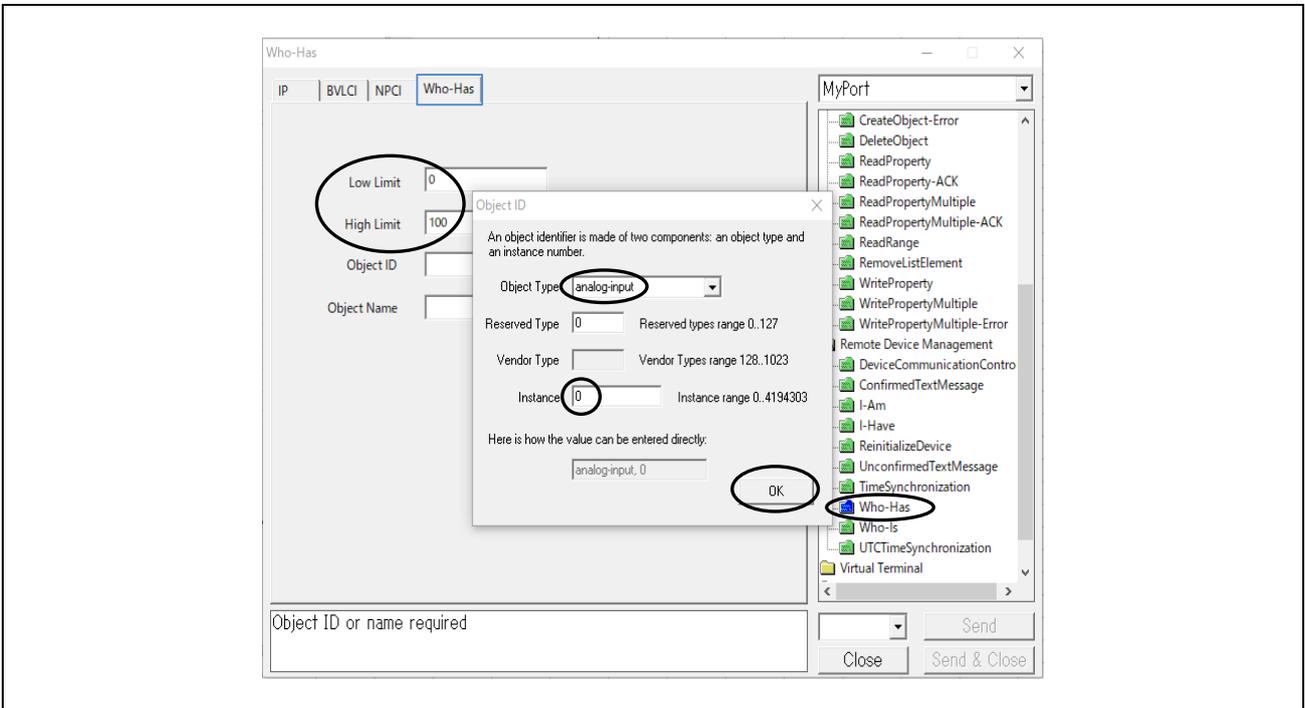


Fig.4-54 Who-Has parameters(1)

Object Name を入力した場合の例を示します。

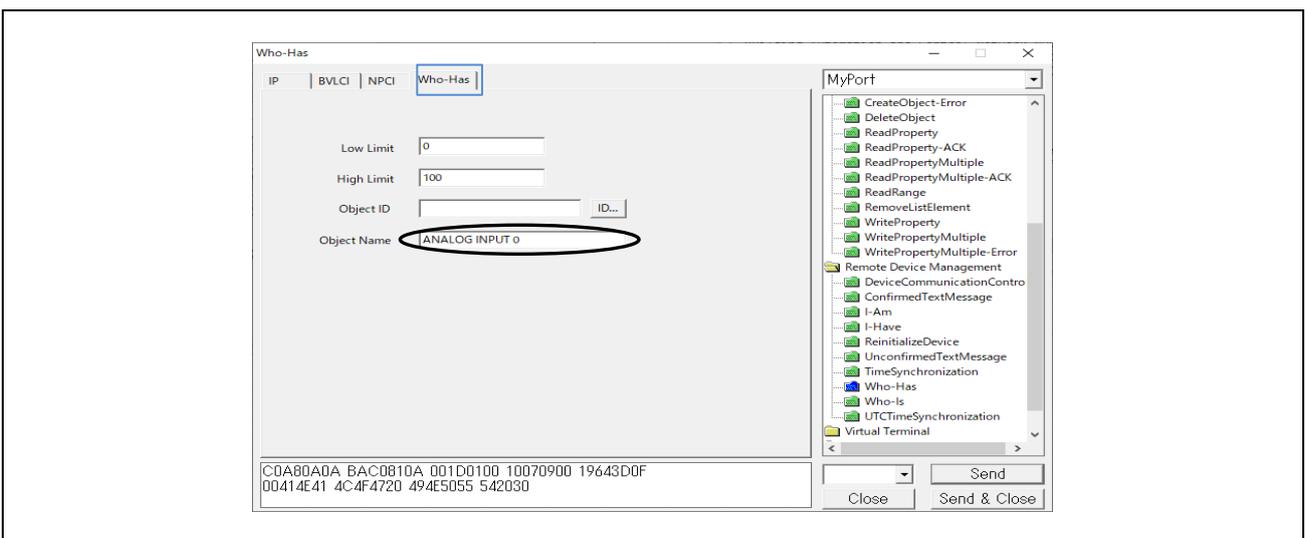


Fig.4-55 Who-Has parameters(2)

次に IP タグをクリックして Global Broadcast または Local Broadcast を選択したら Send をクリックします。

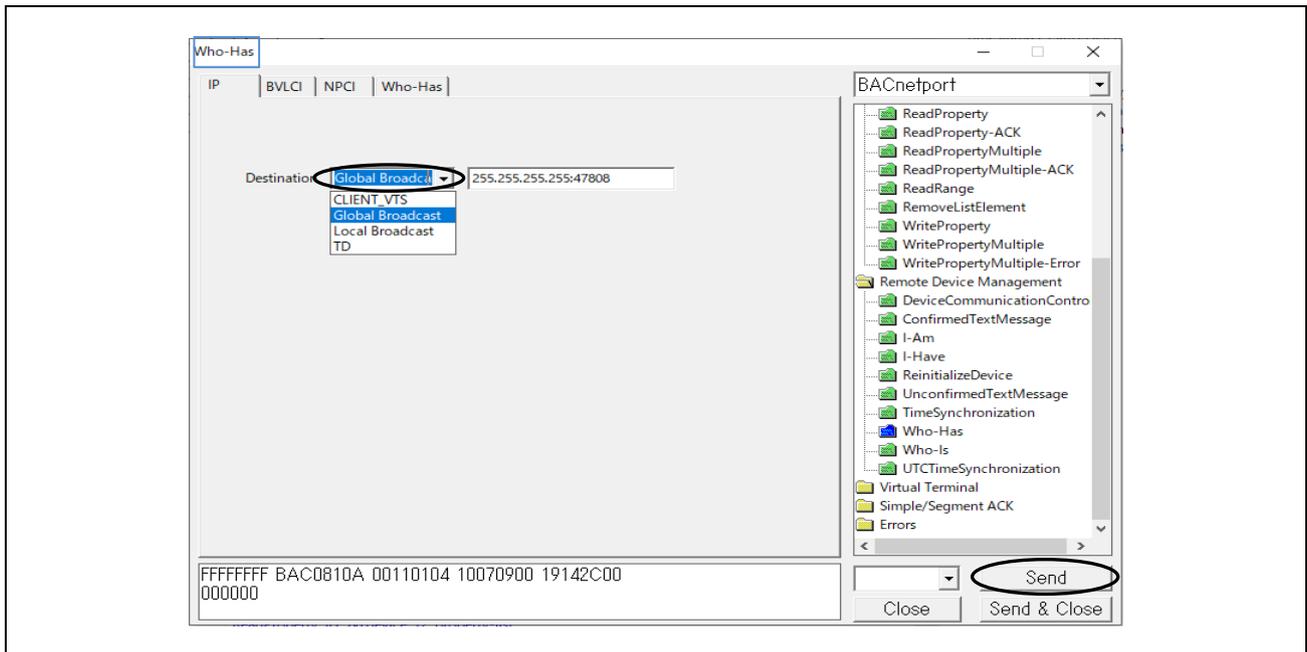


Fig.4-56 Who-Has parameters(3)

次のキャプチャ画面では Who-Has サービス要求による analog-input,0 オブジェクトを持っているデバイス探索ブロードキャストに対して、B-BC サーバーは I-Have 応答をローカルブロードキャストしています。

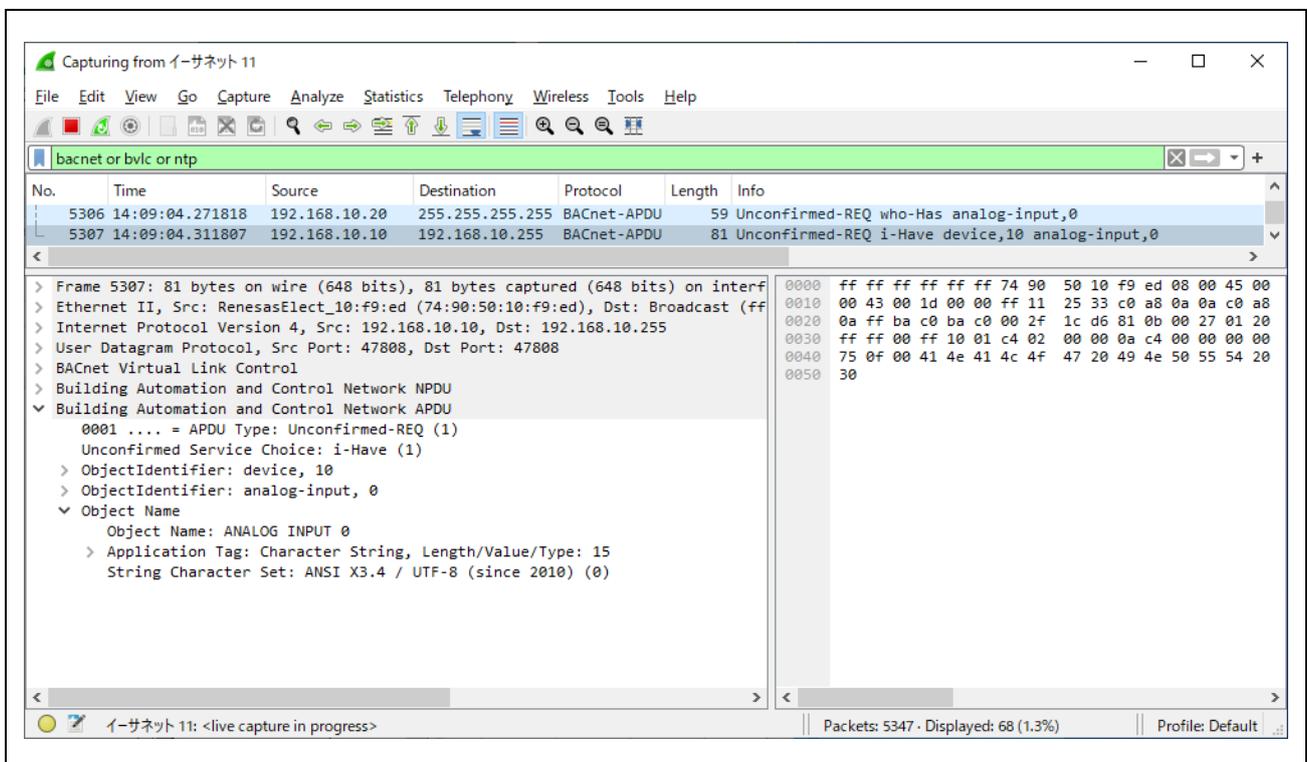


Fig.4-57 Capture Who-Has and I-Have

4.5.5 ReadPropertyMultiple

前回使用したダイアログに表示されているサービスツリーから Object Access > ReadPropertyMultiple をクリックします。

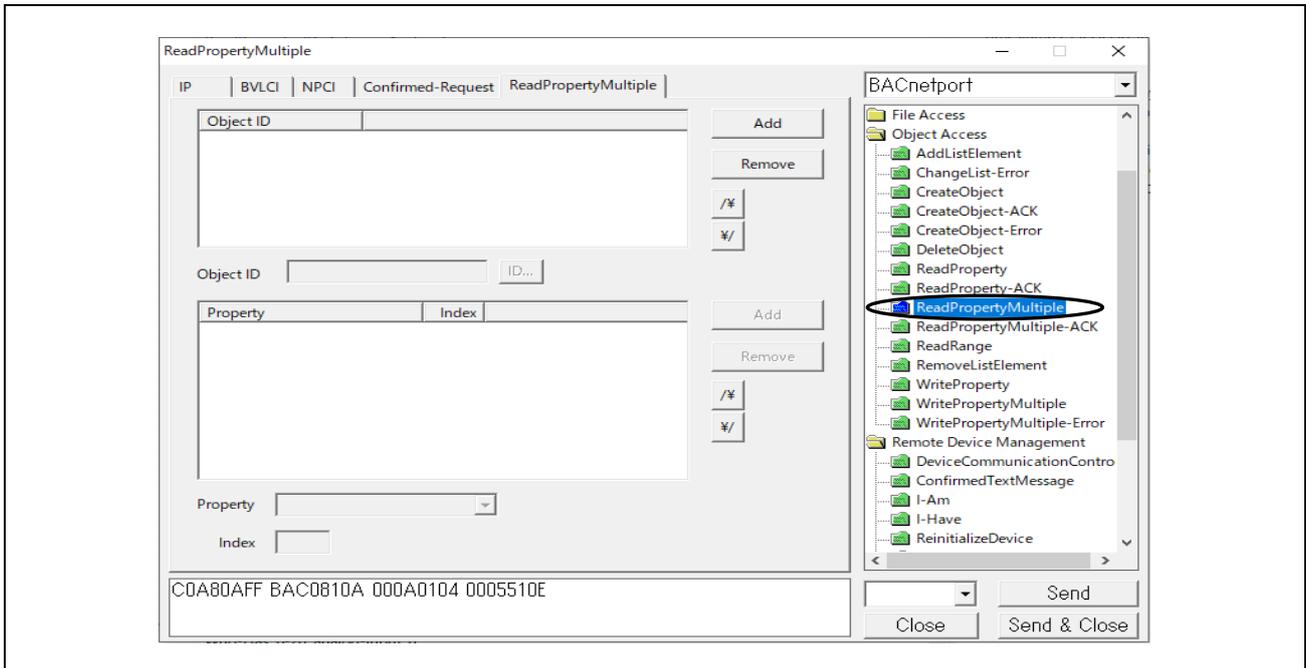


Fig.4-58 ReadPropertyMultiple parameters(1)

表示された ReadPropertyMultiple ダイアログの上部にある Add をクリックします。ID... をクリックして Object ID を選択したら OK をクリックします。例では device,10 オブジェクトを選択しています。

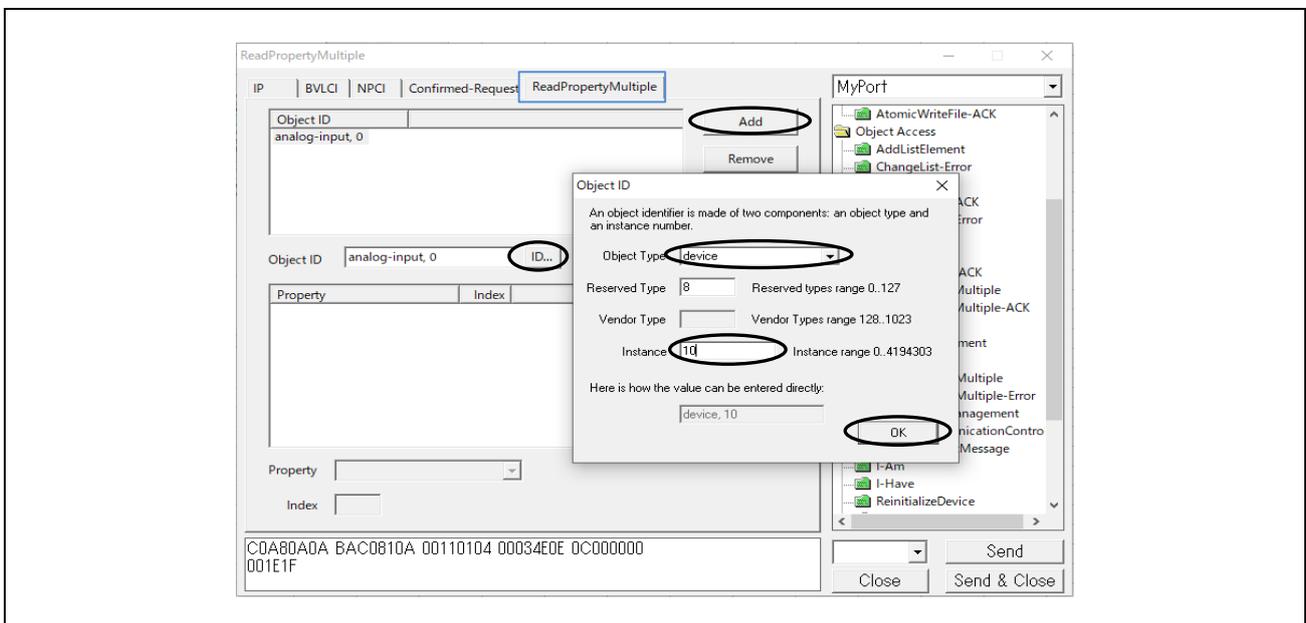


Fig.4-59 ReadPropertyMultiple parameters(2)

ダイアログに表示されている中央の Add をクリックします。Property のプルダウンメニューからプロパティを選択します。例ではすべてのプロパティ "all" を選択しています。Send をクリックします。

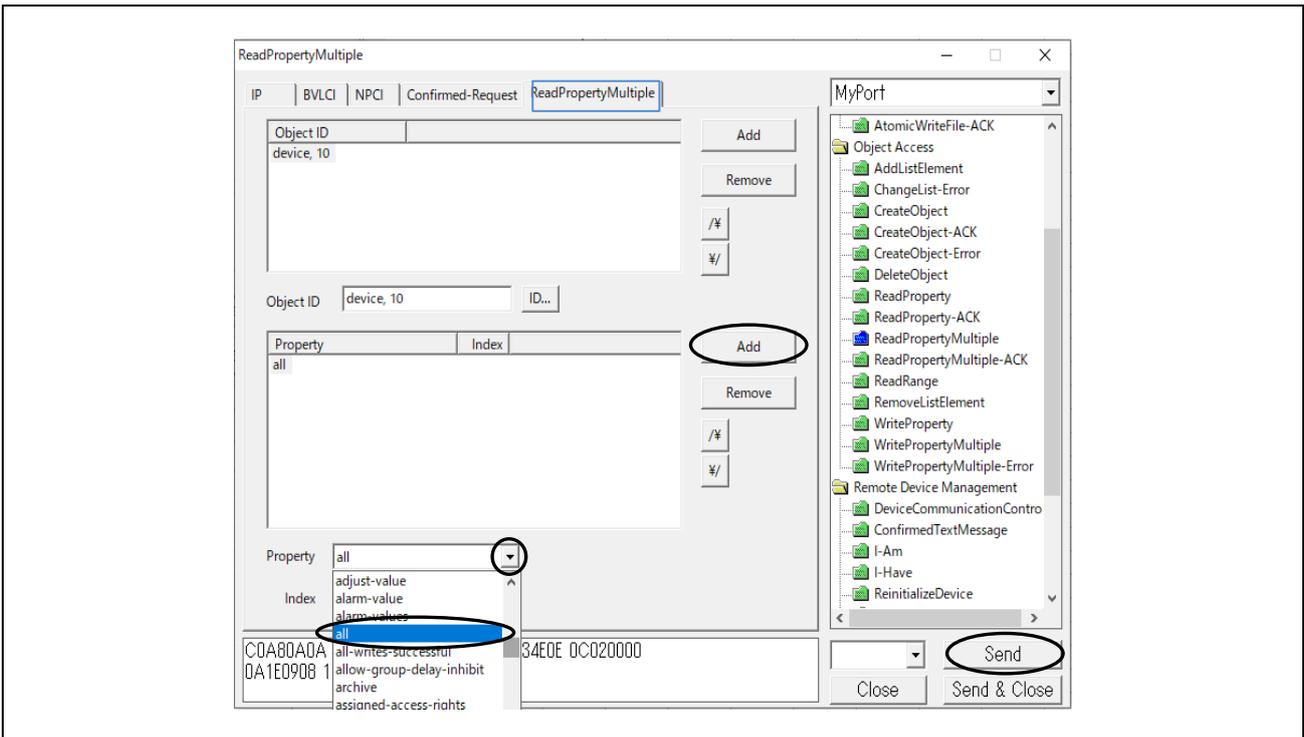


Fig.4-60 ReadPropertyMultiple parameters(3)

次のキャプチャ画面では VTS クライアントから、ReadPropertyMultiple サービスによる device,10 オブジェクトの "all" プロパティ要求と B-BC サーバーからの結果を含む Complex-Ack 応答を示しています。

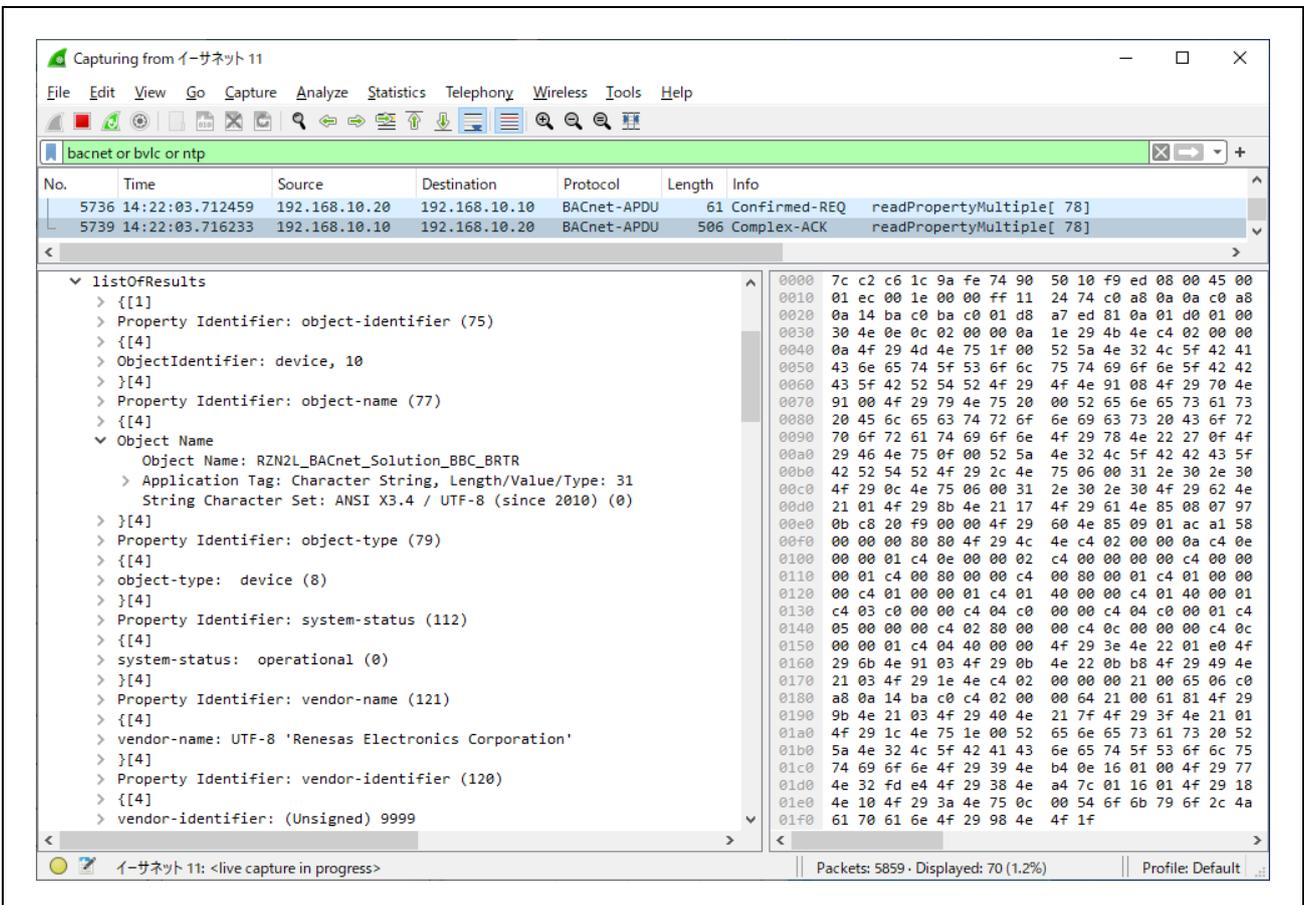


Fig.4-61 Capture ReadPropertyMultiple device,10 all

4.5.6 WriteProperty

前回使用したダイアログに表示されているサービスツリーから Object Access > WriteProperty をクリックします。Object ID を選択して OK をクリックします。例では multi-state-value,0 オブジェクトを選択しています。

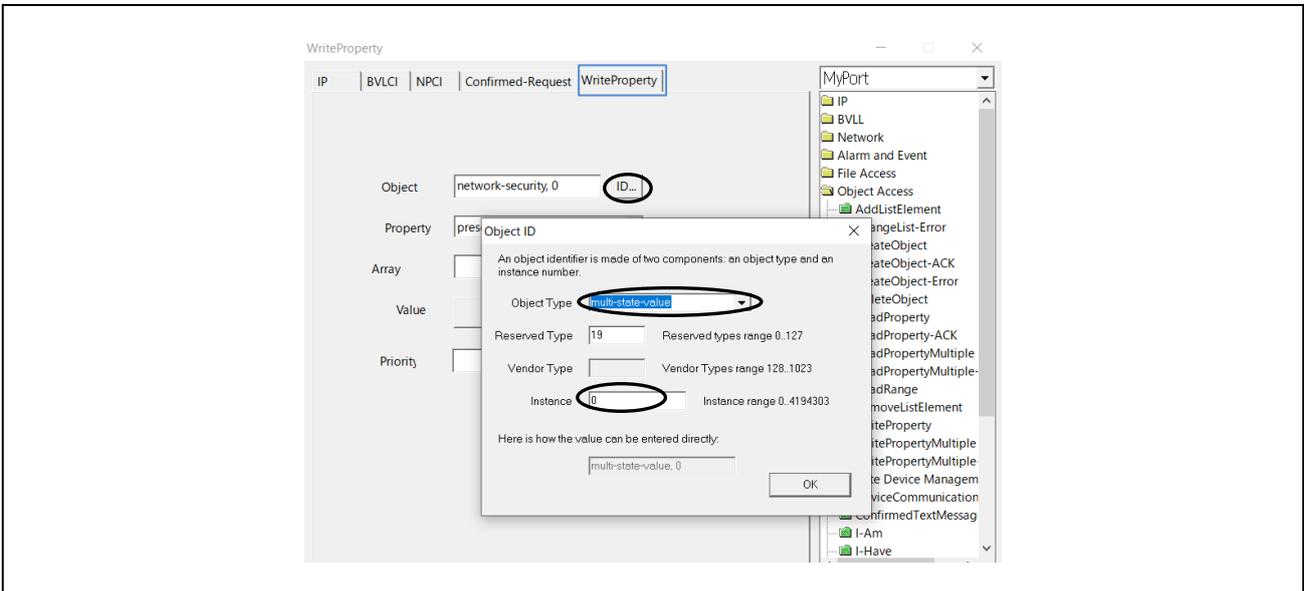


Fig.4-62 WriteProperty parameters(1)

次に Property から present-value を選択します。

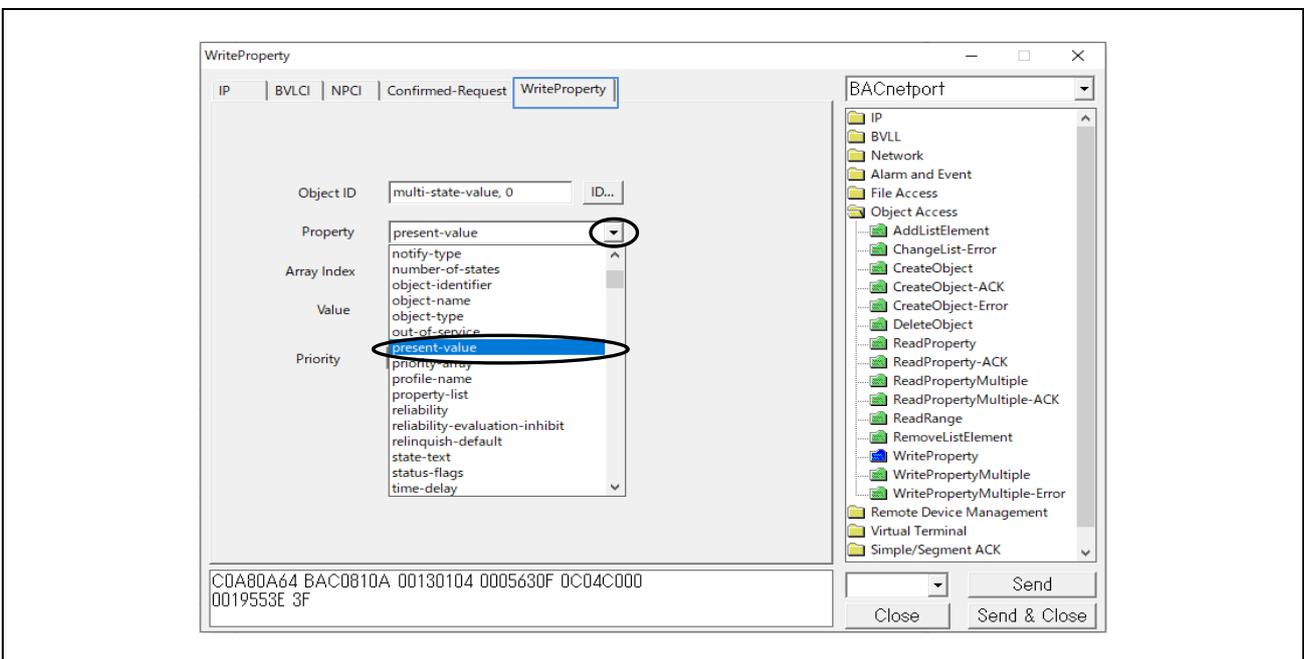


Fig.4-63 WriteProperty parameters(2)

次に Any... をクリックします。ポップアップダイアログの Add をクリックすると Null が表示されるので、この Null を選択した状態で、Type のプルダウンメニューからデータの型を選択します。例の場合は multi-state-value のデータ型 Unsigned を選択しています。

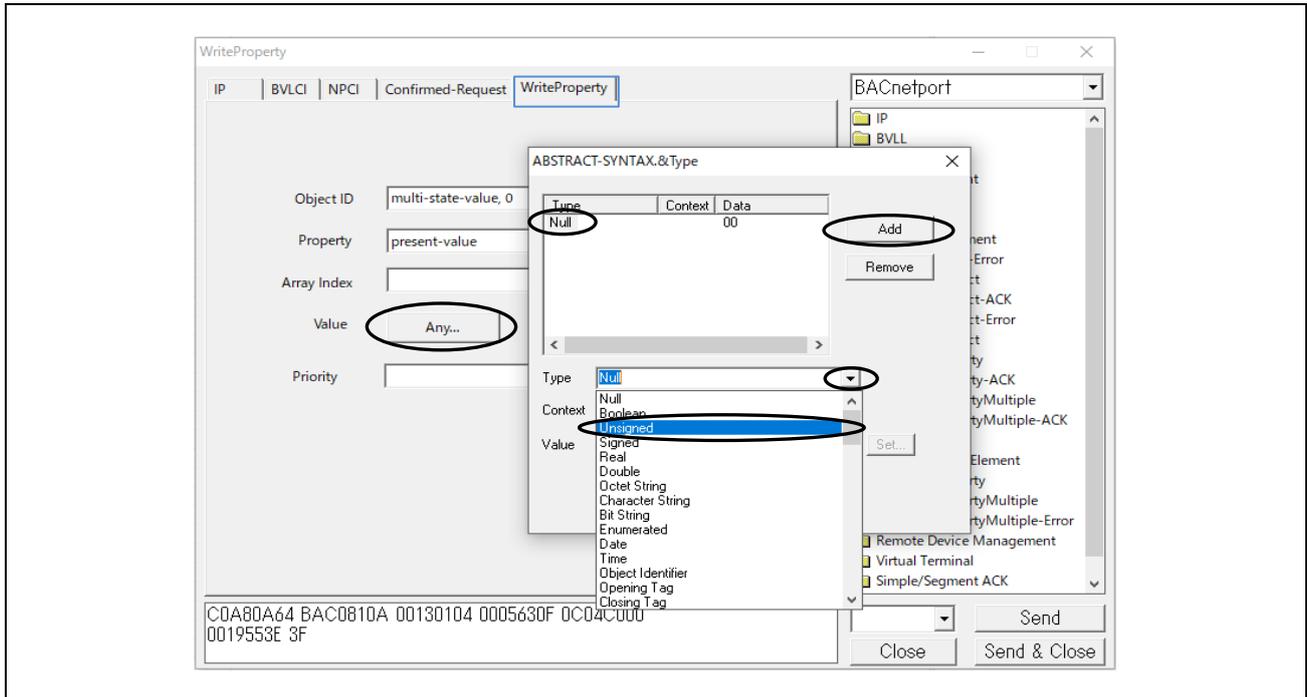


Fig.4-64 WriteProperty parameters(3)

ポップアップダイアログの Value を入力します。本サンプルソフトの場合、multi-state-value の設定範囲は 1~3 です。また、multi-state-value は常に 0 より大きい値でなければいけません。さらにプロパティのデータ型は 1 つずつ規格書で厳密に定義されていますので Type プルダウンメニューに表示されるデータ型を規格書に照らし合わせて適切に選択する必要があります。[12 MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS]

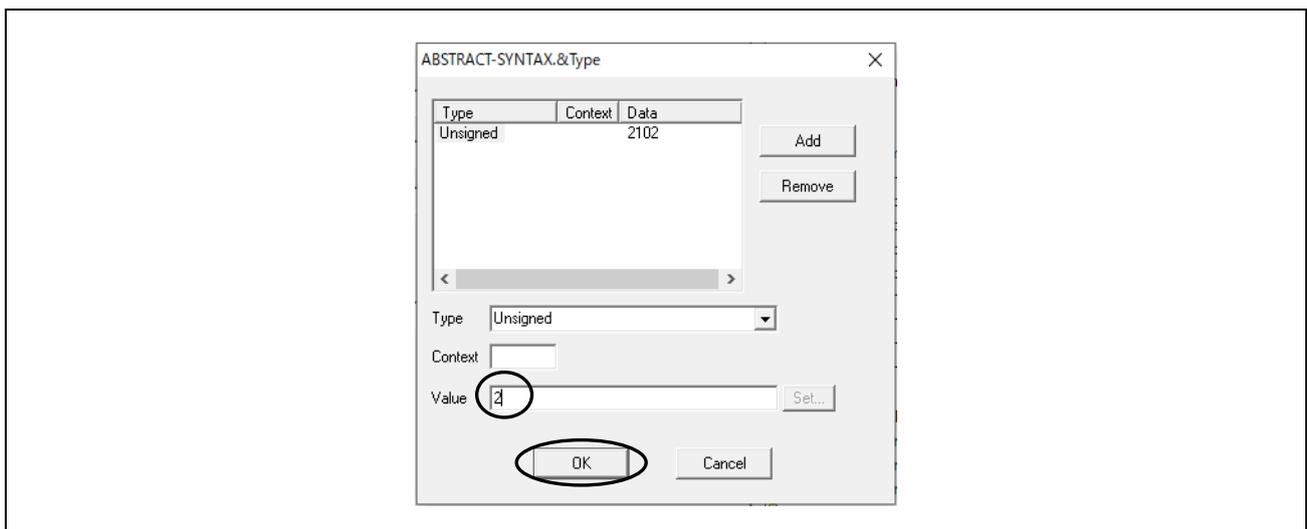


Fig.4-65 WriteProperty parameters(4)

Send をクリックします。

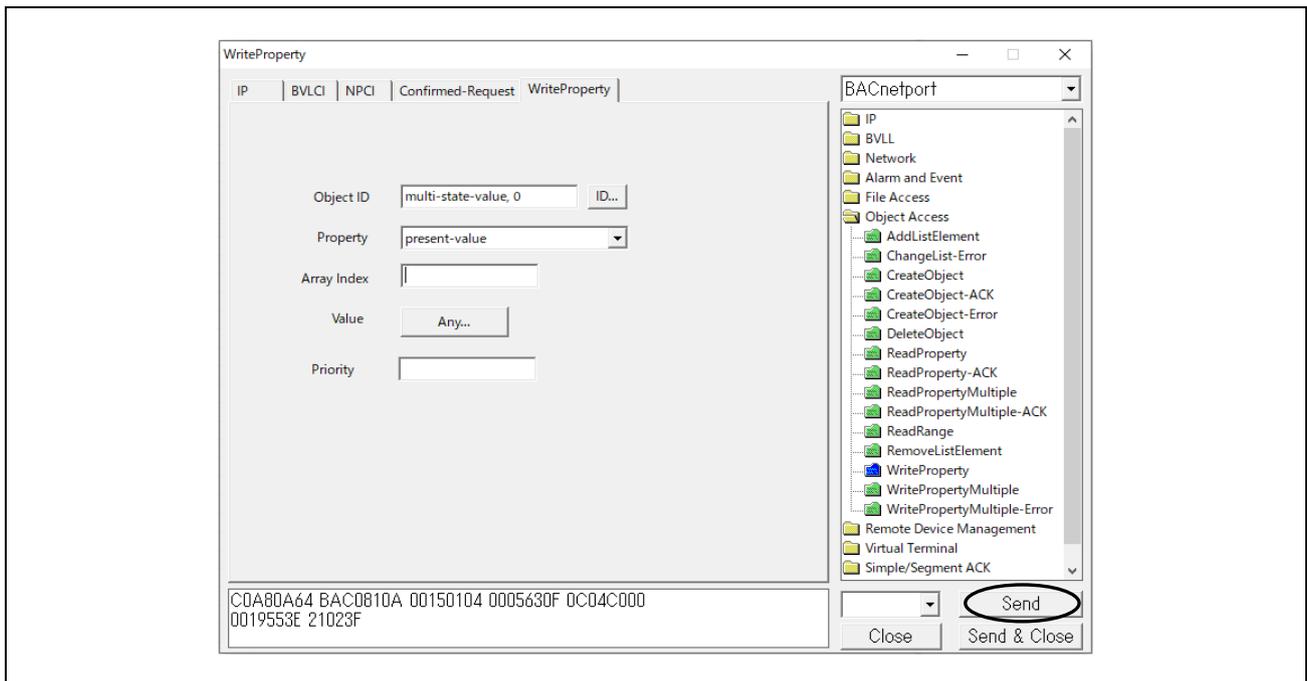


Fig.4-66 WriteProperty parameters(5)

次のキャプチャ画面では VTS クライアントから、WriteProperty サービスによる multi-state-value,0 オブジェクトの present-value プロパティ変更と B-SS サーバーからの Simple-Ack 応答を示しています。

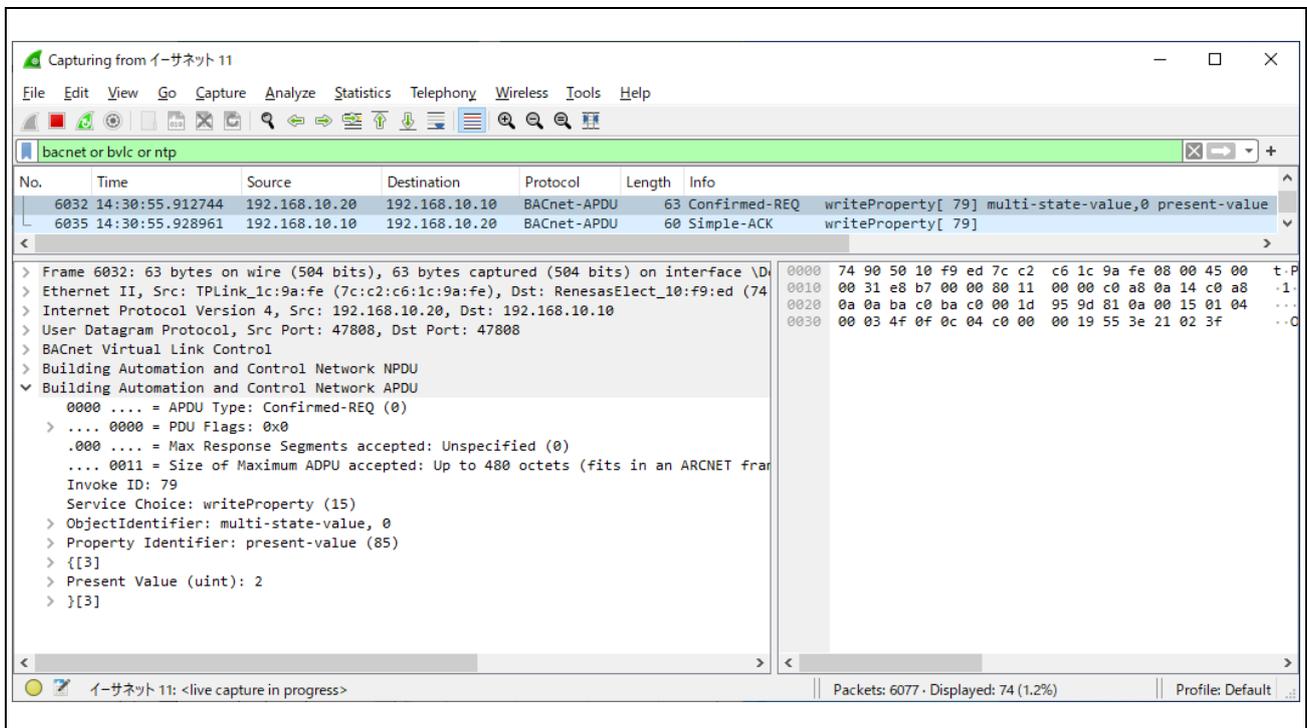


Fig.4-67 Capture WriteProperty multi-state-value,0 present-value

4.5.7 WritePropertyMultiple

引き続き VTS のサービスツリーから Object Access > WritePropertyMultiple をクリックします。Object ID の Add、ID... を順にクリックしてオブジェクトを選択し、OK をクリックします。

例では WritePropertyMultiple サービスを使って B-BC から VTS クライアントへの ConfirmedEventNotification の動作確認手順を示しています。Fig.4-68 は notification-class,0 オブジェクトを選択しています。

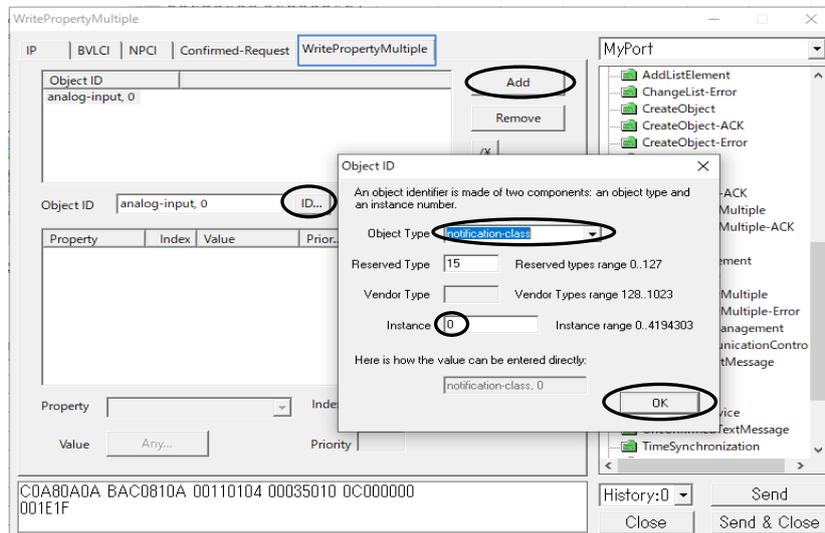


Fig.4-68 WritePropertyMultiple parameters(1)

Property の Add をクリックして、ack-required を選択します。続けて、Any... をクリックするとデータ型を選択するダイアログが表示されます。Add をクリックして Type からデータ型を選択します。Set... をクリックすると対応するダイアログが表示されますので値を選択します。OK をクリックして、それぞれのダイアログを閉じます。

例ではデータ型 EventTransitionBits を選択し、イベント通知条件を 3 つともすべて選択しています。

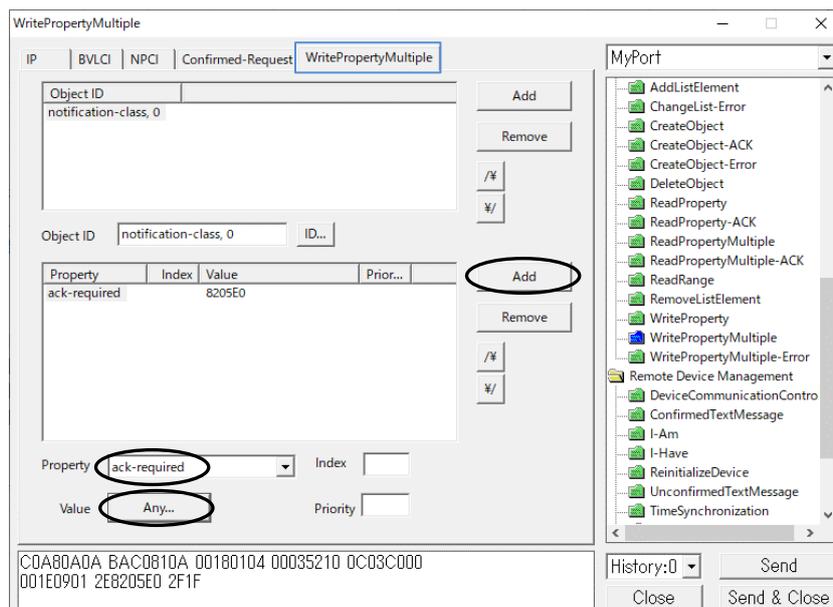


Fig.4-69 WritePropertyMultiple parameters(2)

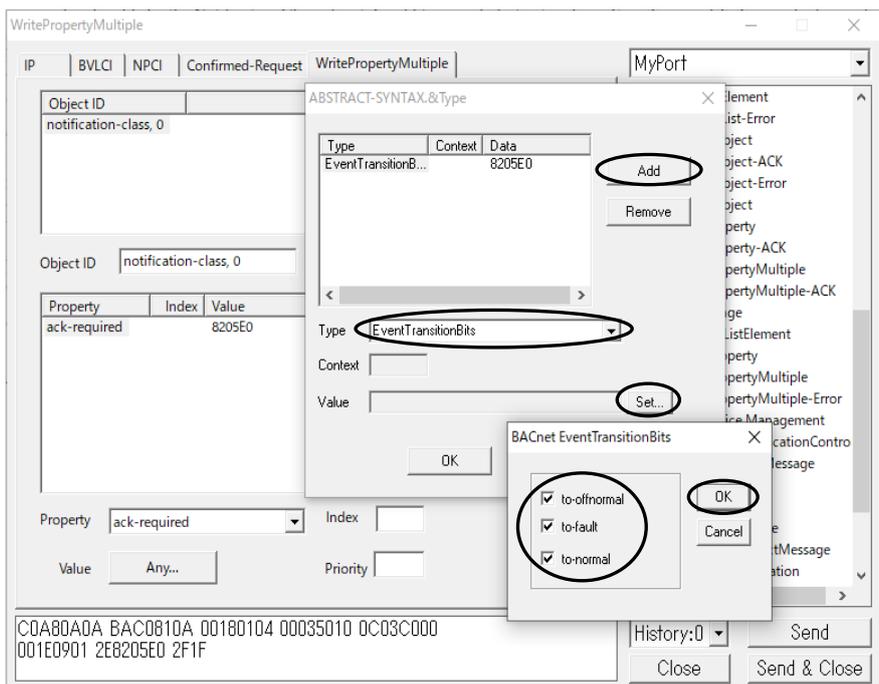


Fig.4-70 WritePropertyMultiple parameters(3)

同様に Property の Add をクリックして、recipient-list を選択します。
 例ではデータ型 Destination を選択し、イベント受信者を VTS(device,0)、VTS からの Ack あり、イベント通知条件を 3 つともすべて選択、有効期間の開始と終了を設定しています。

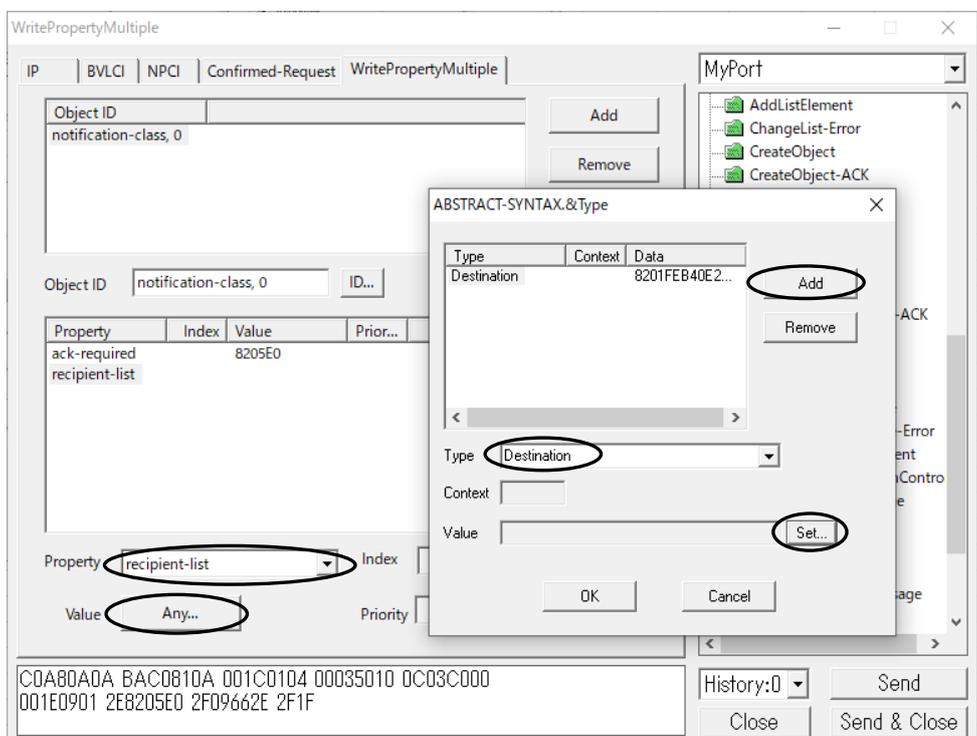


Fig.4-71 WritePropertyMultiple parameters(4)

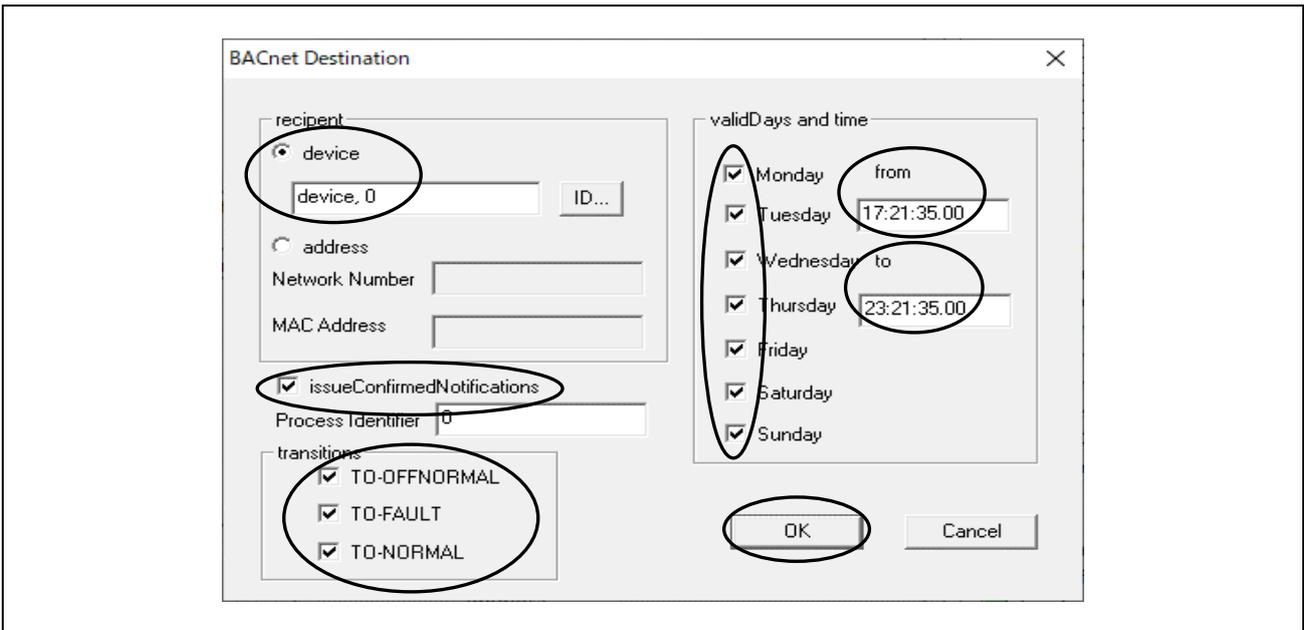


Fig.4-72 WritePropertyMultiple parameters(5)

さらに Property の Add をクリックして、priority を選択します。最後に Send をクリックします。例ではデータ型 PriorityArray を選択し、3 つの通知条件に対応する優先度(255:最低)を設定しています。

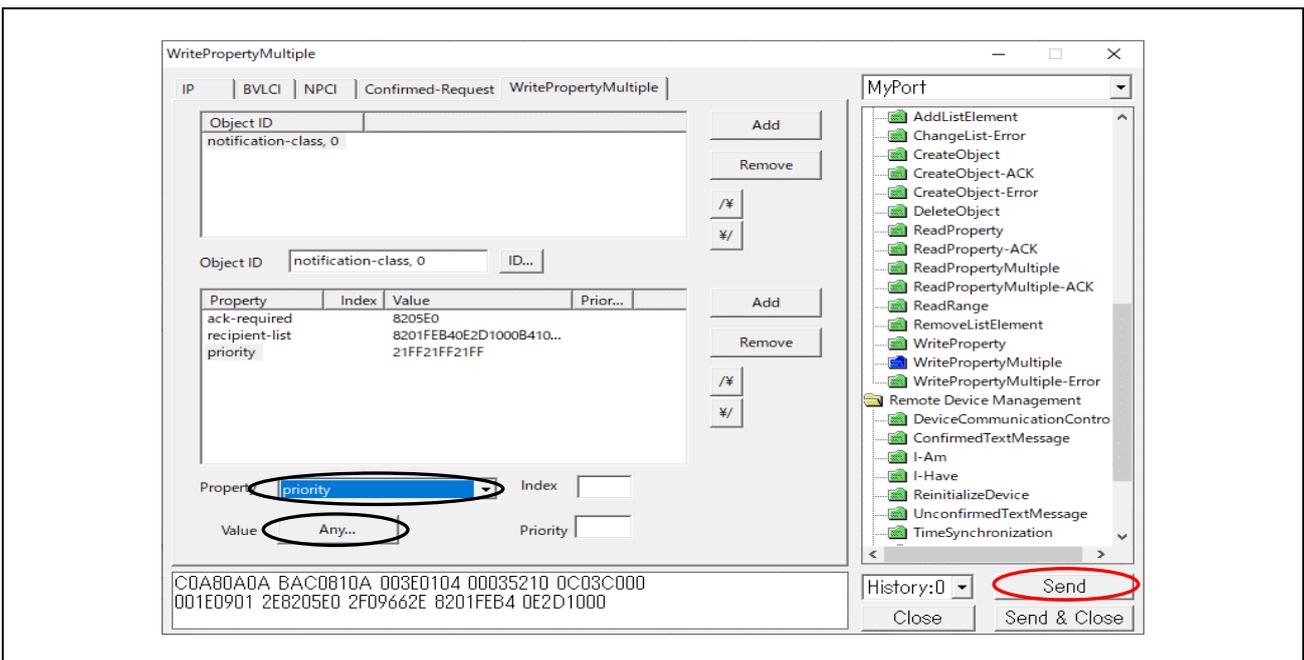


Fig.4-73 WritePropertyMultiple parameters(6)

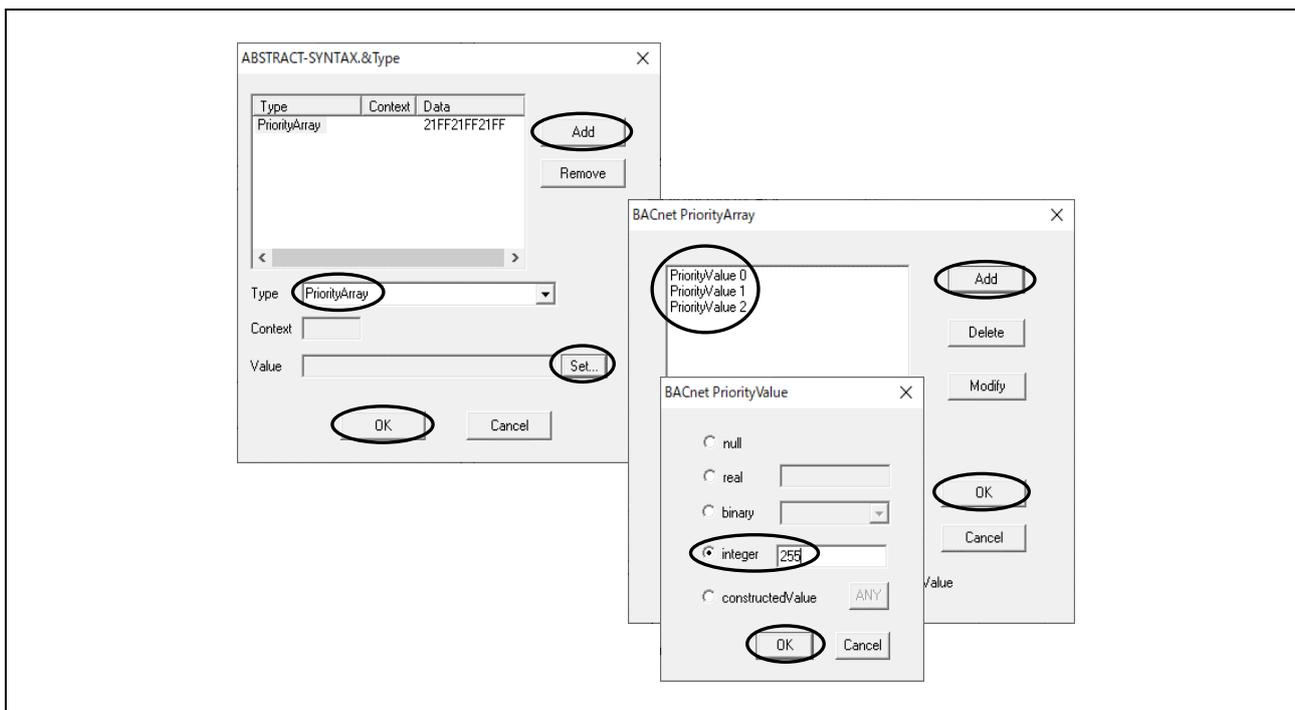


Fig.4-74 WritePropertyMultiple parameters(7)

次に一旦 Object ID に設定した古い値(notification-class,0)を選択して、Remove をクリックして旧設定値を削除します。新たに Object ID の Add、ID…を順にクリックしてオブジェクトを選択し、OK をクリックします。

例では analog-input,0 オブジェクトを選択しています(Fig.4-75)。Property の Add をクリックして、out-of-service=true と present-value=-0.1 を設定します(Fig.4-76)。この設定により OutOfRange イベントアルゴリズムの LowLimit プロパティ値(0.0)を下回り、イベント通知が発生することを確認できます。ちなみに HighLimit プロパティ値(100.0)を超えてもイベント通知が発生します。最後に Send をクリックします。

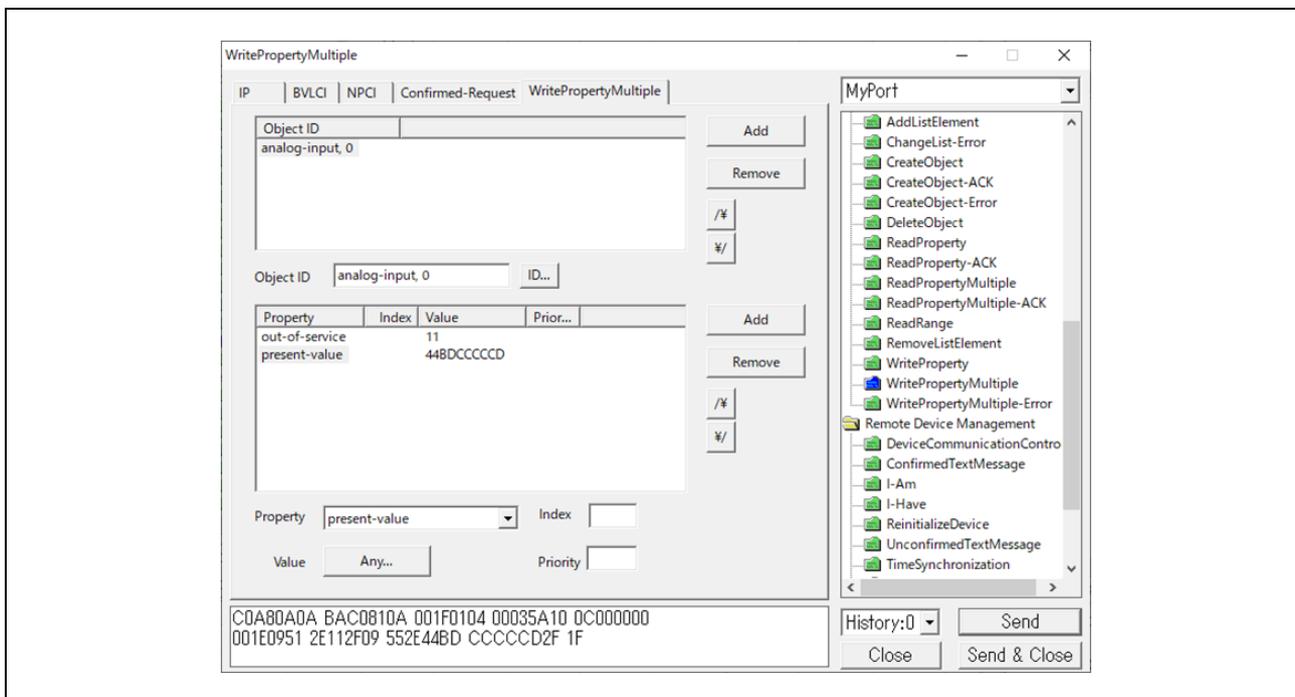


Fig.4-75 WritePropertyMultiple parameters(8)

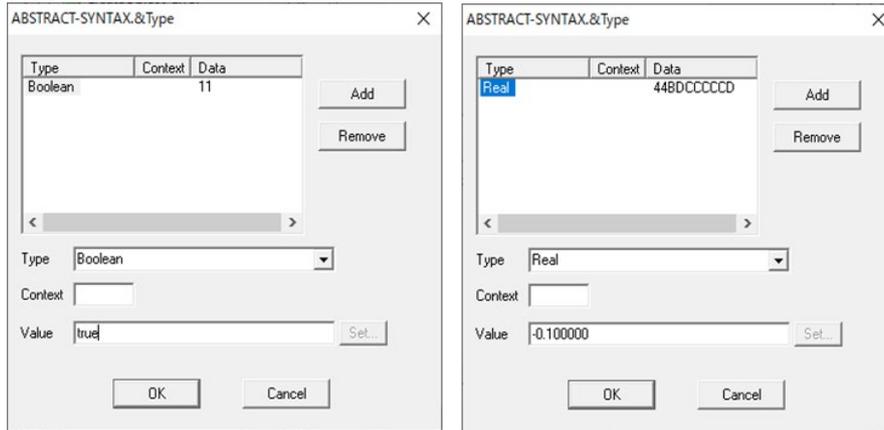


Fig.4-76 WritePropertyMultiple parameters(9)

次のキャプチャ画面では VTS クライアントから、WritePropertyMultiple サービスによる notification-class,0 オブジェクトと analog-input,0 オブジェクトの設定後、B-BC サーバーからの ConfirmedEventNotification サービス要求と、それを受信した VTS クライアントからの Ack を示しています。

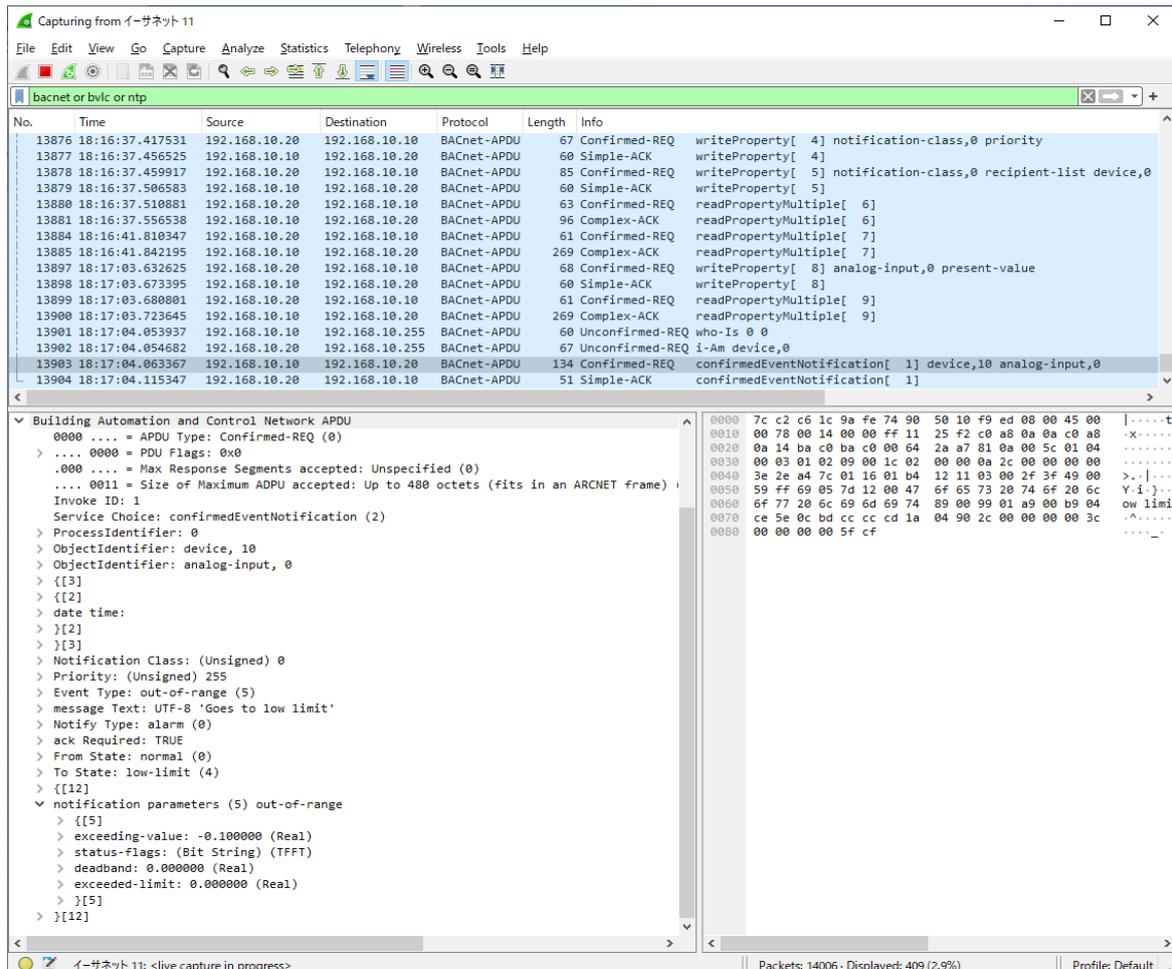


Fig.4-77 Capture WritePropertyMultiple and ConfirmedEventNotification

4.5.8 SubscribeCOV

引き続き VTS のサービスツリーから Alarm and Event > SubscribeCOV をクリックします。

- Subscribe Process ID は COV クライアント、すなわち VTS 内のプロセスを識別するために使用されます。この ID は B-BC サーバーが COV 通知を行うときやサブスクライブをキャンセルされたとき、どの COV クライアントかを識別するためのものです。値 0 は予約されており、サブスクライバーなしの時に COV 通知に使用されます。これは外気温度などをブロードキャストするときに COV サーバーが使用します。
- Monitored Object ID は値変化を検出するプロパティを保持するオブジェクトを指定するためのものです。
- Issue Confirmed Notifications は True/False を指定します。True のときは COV サーバーに確認あり型 COV 通知 ConfirmedCOVNotification を要求し、COV クライアントは COV 通知に対して Ack 応答を COV サーバーに返します。COV サーバーは Ack を受け取るまでは次の COV 通知を行いません。False のときは確認なし型 COV 通知 UnconfirmedCOVNotification を要求します。
- Lifetime はサブスクライブ期間のことで、単位は分です。COV サーバーは Lifetime が経過すると Subscribe Process ID に対応する COV クライアントへ COV 通知を行いません。このパラメータが空欄の場合は無期限を表します。COV クライアントがサブスクライブをキャンセルする場合は Issue Confirmed Notifications と Lifetime の両方とも空欄にします。

Send をクリックして SubscribeCOV サービス要求を送信します。

例では空気速度センサの入力値を割り当てた analog-input,0 オブジェクトを選択し、確認なし型 COV 通知を無期限で指定しています。Fig.4-78 は Fig.4-46 と同じく SubscribeCOV サービス要求先に B-SS を選択しています。

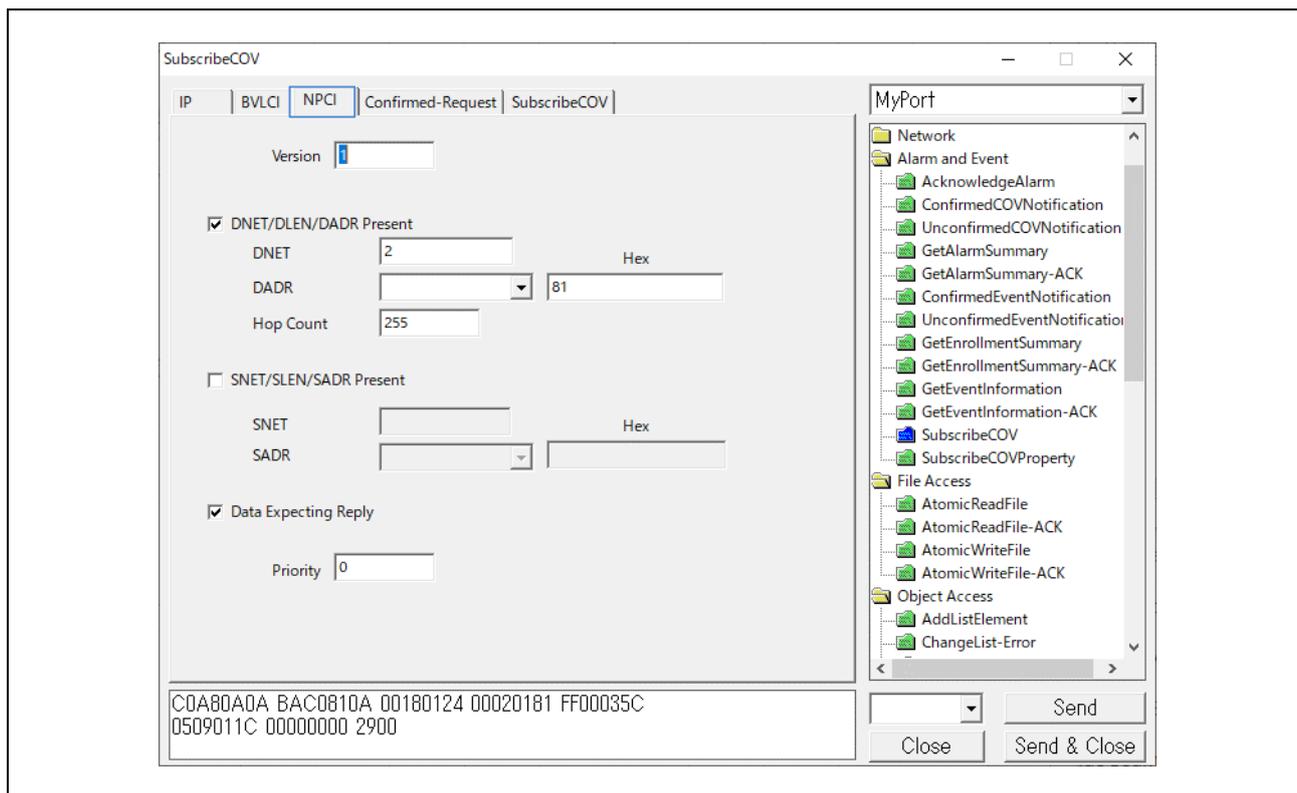


Fig.4-78 Select SubscribeCOV destination (B-SS)

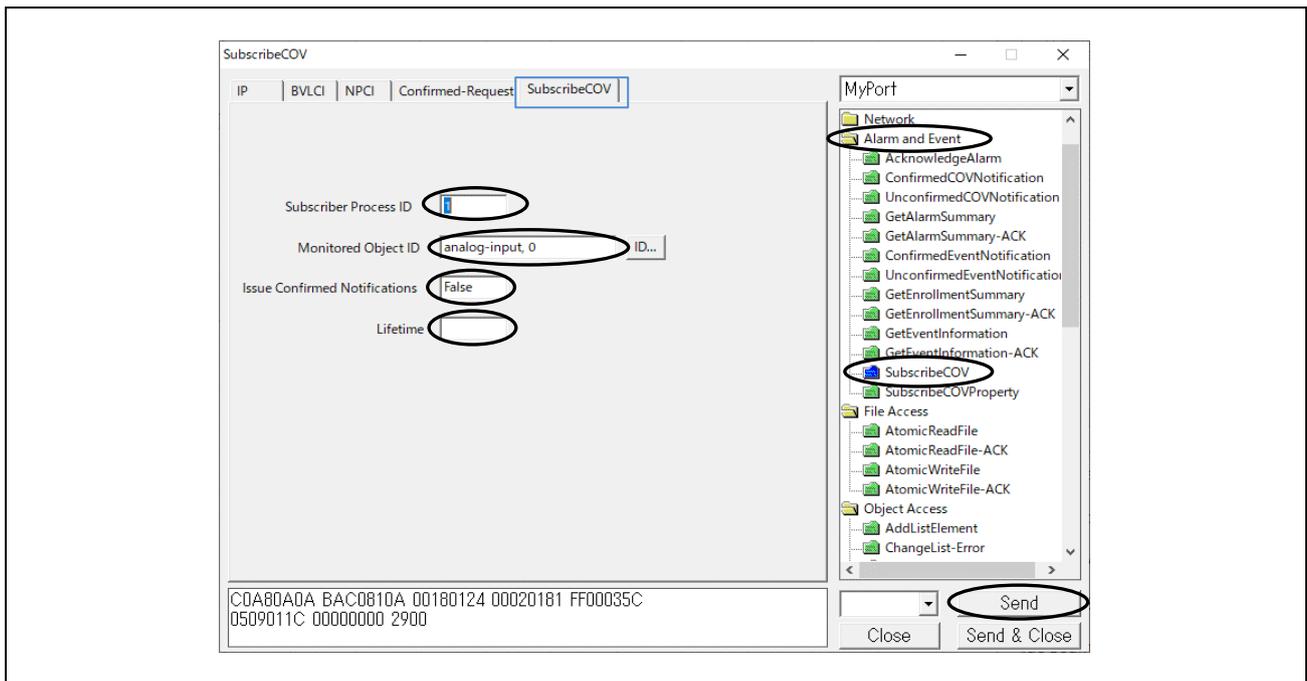


Fig.4-79 SubscribeCOV parameters

次のキャプチャ画面では VTS クライアントから、SubscribeCOV サービスによる analog-input,0 オブジェクトのプロパティ値変化検出を要求し、B-SS スレーブからの Simple-Ack 応答を示しています。空気速度変化を検出した B-SS スレーブから UnconfirmedCOVNotification サービスにより present-value および status-flags プロパティ値を通知しています。Time remaining はサブスクリプション期間の残り時間を表示しますが無期限を要求されたため 0.00.00 を返しています。

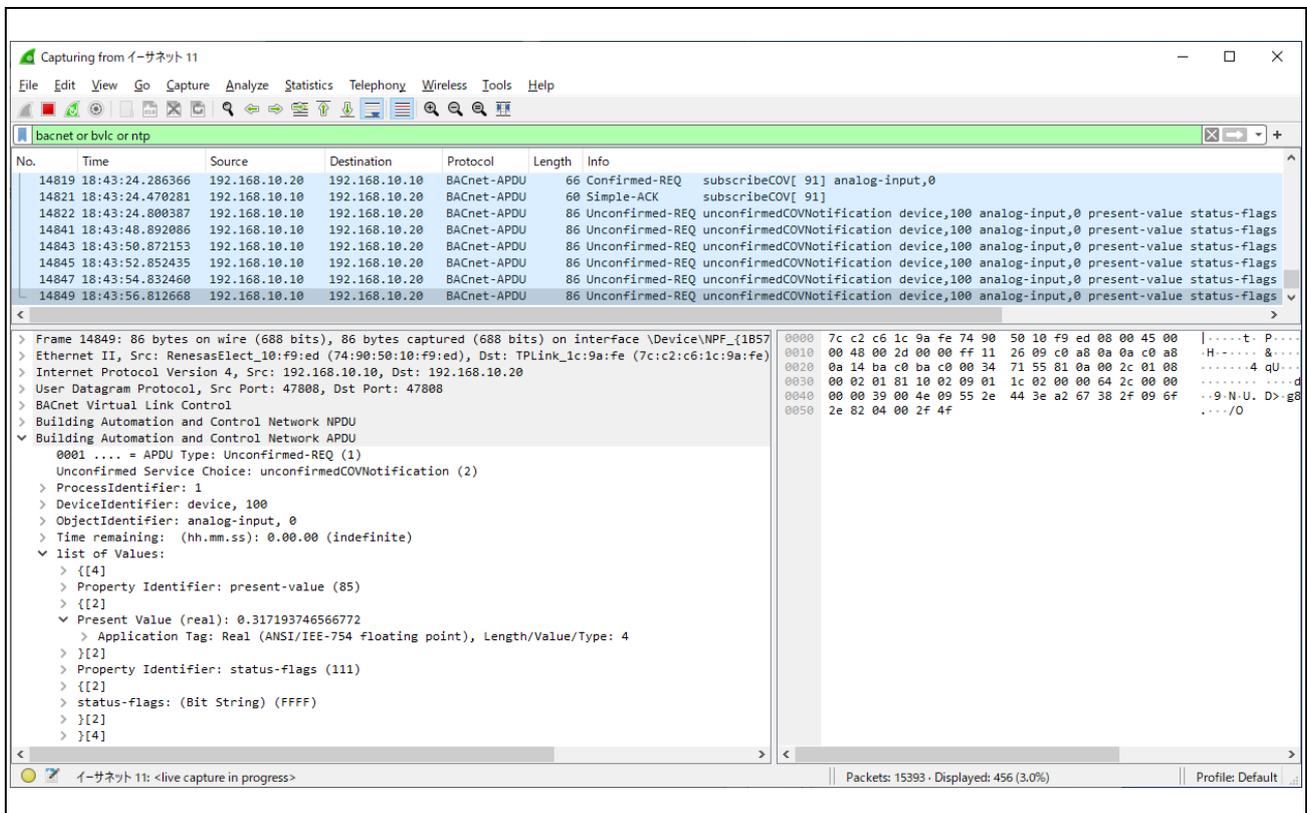


Fig.4-80 Capture SubscribeCOV and UnconfirmedCOVNotification

次のキャプチャ画面では VTS クライアントから、Issue Confirmed Notifications を True にして、SubscribeCOV サービス要求し、B-SS スレーブからの Simple-Ack 応答を示しています。空気速度変化を検出した B-SS スレーブから ConfirmedCOVNotification サービス通知を行い、COV クライアントは Simple-Ack を応答しています。

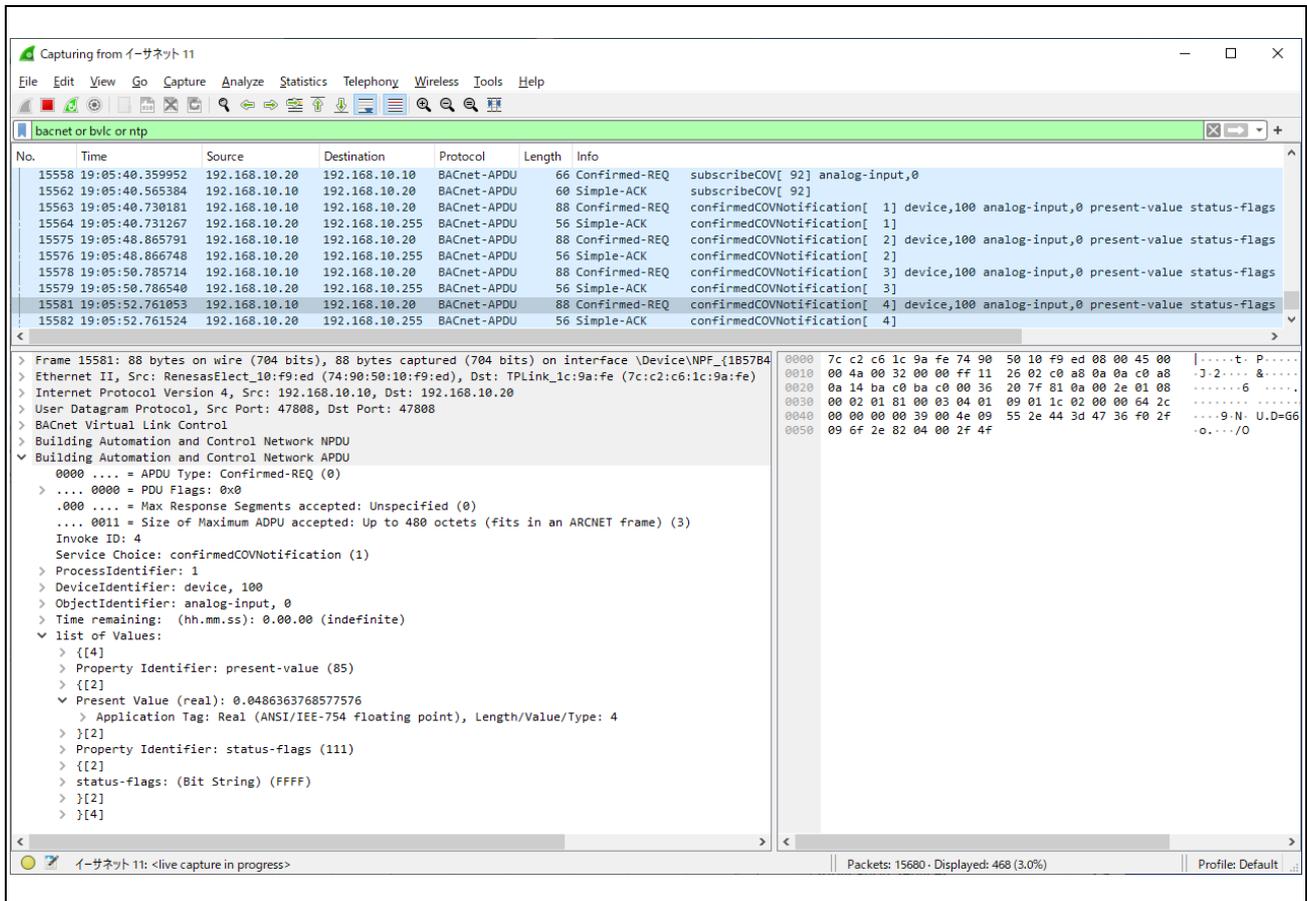


Fig.4-81 Capture SubscribeCOV and ConfirmedCOVNotification

4.5.9 ReinitializeDevice

注意) ReinitializeDevice サービスはターゲットデバイスをリセットします。デバッグ接続状態の場合は RSK ボードの S3 RESET プッシュスイッチ(赤)を押してデバッグ接続を解除してください。デバッグ接続状態のままで、このサービスを実行すると B-BC がリポートした後 Ethernet 通信が確立しません。

制限事項) ReinitializeDevice の以下の State パラメータについて、本 B-BC サンプルソフトでは未サポートです。

STARTBACKUP

ENDBACKUP

STARTRESTORE

ENDRESTORE

ABORTRESTORE

引き続き VTS サービスツリーから Remote Device Management > ReinitializeDevice をクリックします。State は Cold Start または Warm Start を選択します。それ以外にも選択できますが Activate Changes(ただし、VTS からは選択できません。)を除き未対応です。Password に「filister」を入力してから Send をクリックします。

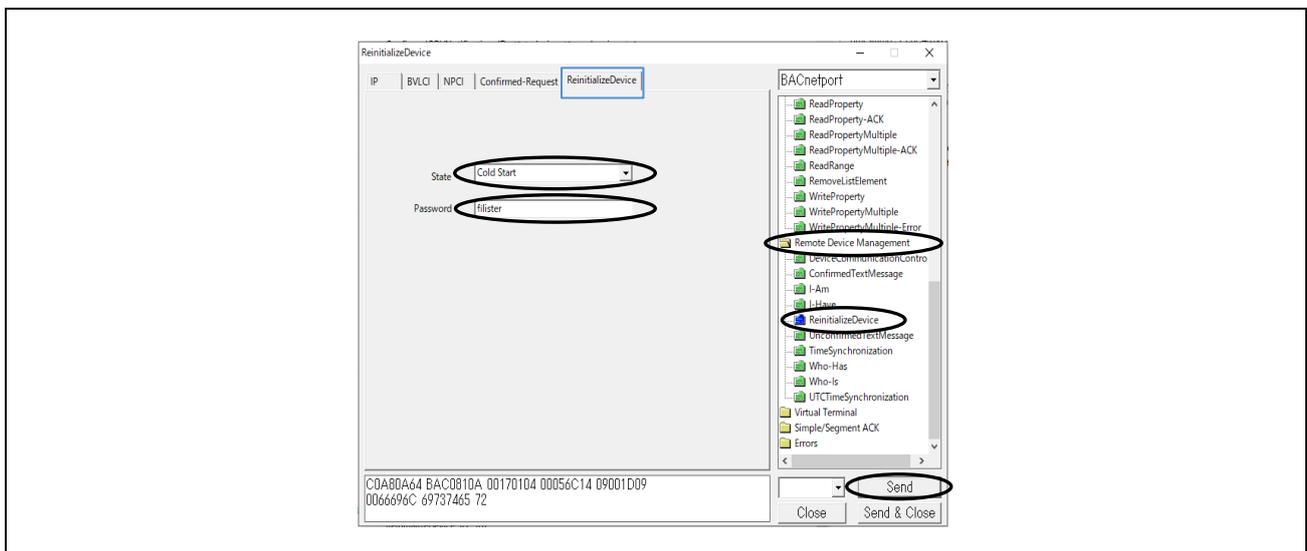


Fig.4-82 ReinitializeDevice parameters

次のキャプチャ画面では VTS クライアントから、ReinitializeDevice サービスを要求し、B-BC または B-SS からの Simple-Ack 応答を示しています。B-BC または B-SS はリポートすると I-Am サービスをローカルブロードキャストしています。

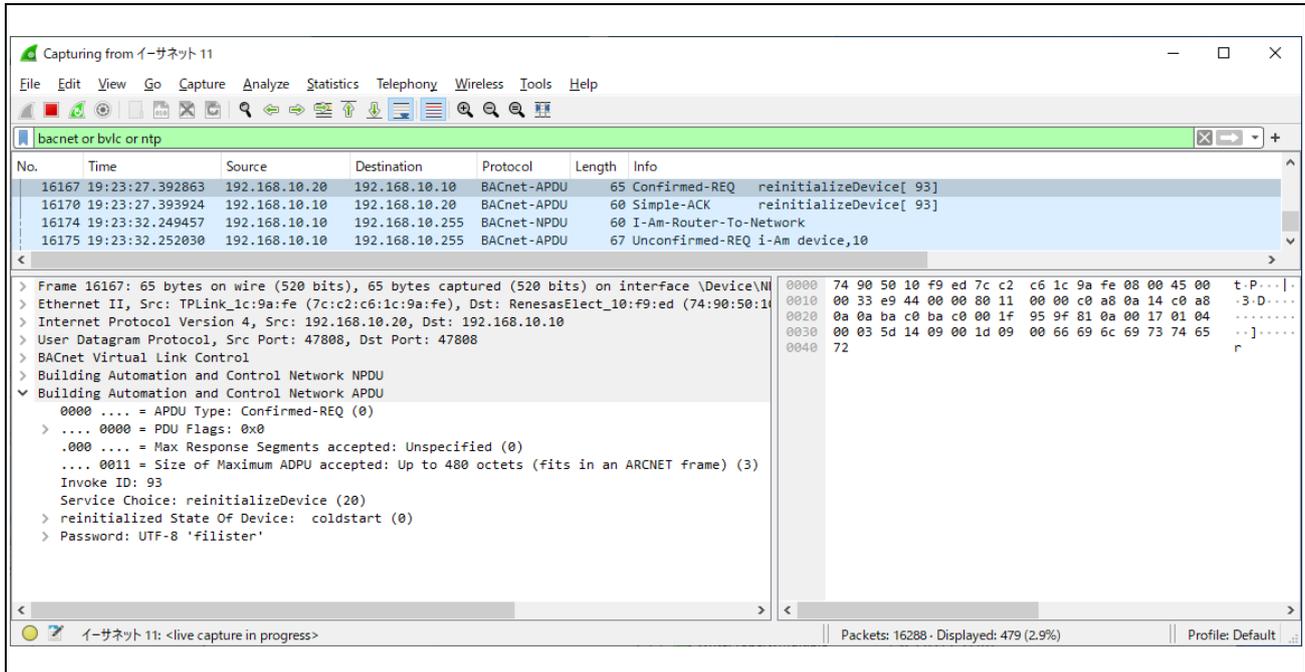


Fig.4-83 Capturing ReinitializeDevice (to B-BC)

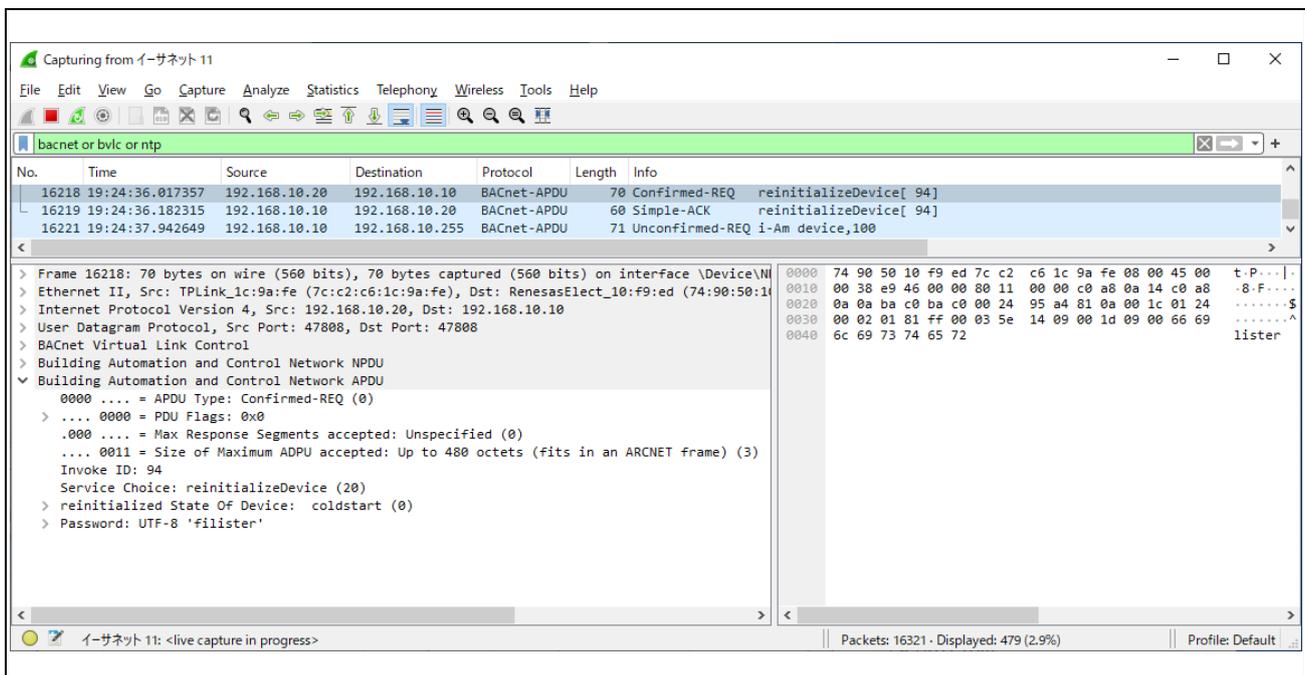


Fig.4-84 Capturing ReinitializeDevice (to B-SS)

4.5.10 DeviceCommunicationControl

引き続き VTS に表示されているサービスツリーから Remote Device Management > DeviceCommunicationControl をクリックします。

- Time duration は通信停止期間を分単位で入力します。通信停止期間が経過するとサーバーは通信を再開します。空欄にすると無期限になります。
- Disable を選択すると通信停止を要求しますが **BACnet Protocol Revision 20 以降は受け入れなくなりました**。サーバーは Disable 要求を無視して ErrorClass = SERVICES、ErrorCode = SERVICE_REQUEST_DENIED のエラーPDU を送信します。
- Disable Initiation を選択すると I-Am サービスを除く、サーバーからの通知を停止します。クライアントからのサービス要求に対する Ack 応答は停止しません。
- Enable を選択すると通信停止解除を要求します。Time duration は無視されます。

Password に「filister」を入力してから Send をクリックします。例では無期限の Disable Initiation を選択しています。

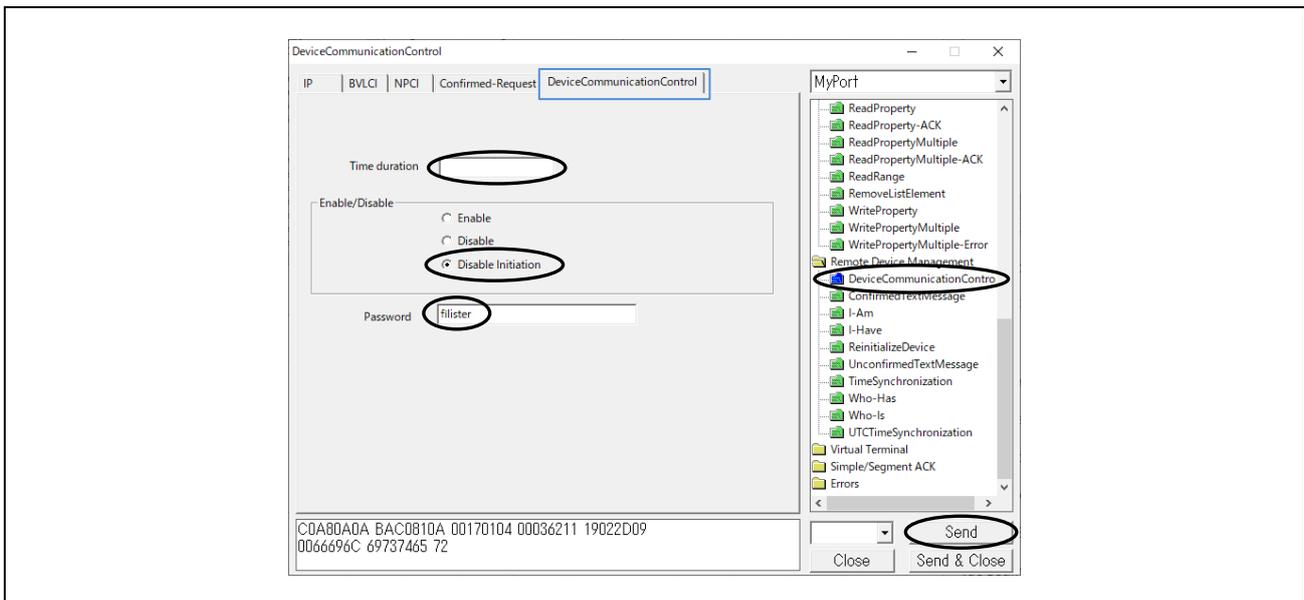


Fig.4-85 DeviceCommunicationControl parameters(Disable Initiation)

次のキャプチャ画面では VTS クライアントから、B-SS に対して DeviceCommunicationControl サービスを要求し、B-SS スレーブからの通知サービス停止を示しています。
 No.565 は UnconfirmedCOVNotification サービスを通知しています。
 No.566 は Disable initiation を要求しています。
 No.574 の Simple-Ack を応答後 UnconfirmedCOVNotification サービスの通知が停止しています。
 No.623 は Who-Has サービスを要求していますが I-Have サービス応答を返していません。
 No.650 は Who-Is サービスを要求しています。
 No.653 は I-Am サービス応答を返しています。
 No.683 は Enable を要求しています。
 No.685 以降に UnconfirmedCOVNotification サービス通知を再開しています。

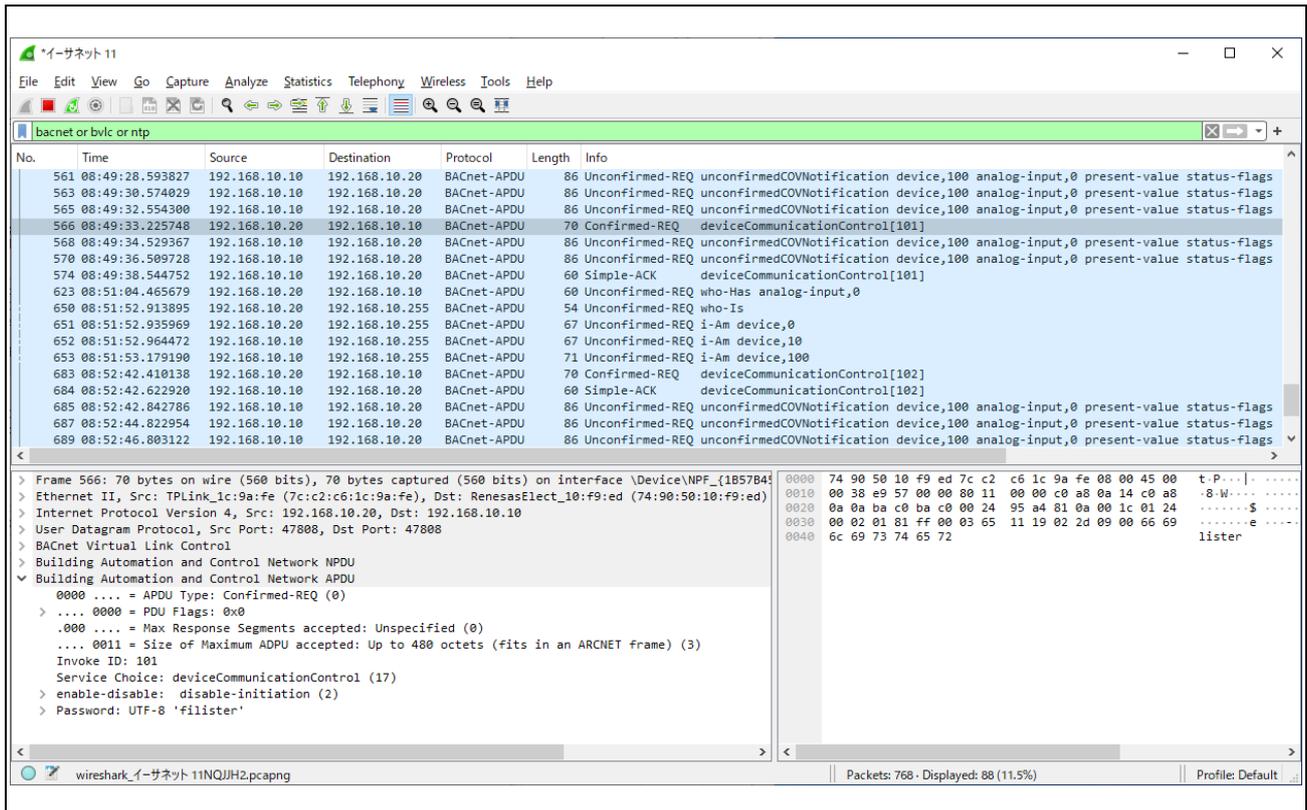


Fig.4-86 Capture DeviceCommunicationControl

4.5.11 AtomicReadFile

多くの BACnet デバイスにはベンダー独自の構成(configuration)ツールによってセットアップされた構成データが含まれます。

AtomicReadFile サービスは B-BC 内部の構成データを読み出します。クライアントは読み出したデータをバックアップファイルとして保存します。また、AtomicWriteFile サービスによって保存したバックアップファイルを B-BC に転送し、B-BC は受信ファイルデータを内部の構成データに上書きして復元します。ファイルの内容と形式はローカルマターです。

B-BC の構成データ(変数名:FlashData) は 4 バイトでアライメントされた構造体(FLASH_DATA_STRUCT)で定義されます。

```

203 #pragma pack(4)
204 typedef struct
205 {
206     uint32_t CheckSum;
207     FLASH_DATA_ETHER_MAC emac;
208     FLASH_DATA_BBMD_MAC bmac;
209     FLASH_DATA_FD_LIFETIME fdsubscription;
210     FLASH_DATA_BIP_MODE mode;
211     FLASH_DATA_BIP_MAC mac_bip;
212     FLASH_DATA_MSTP_MAC mac_mstp;
213     FLASH_DATA_NW_NUMBER mstp;
214     FLASH_DATA_NAME devn;
215     FLASH_DATA_INSTANCE devi;
216     FLASH_DATA_UTC utc;
217     FLASH_DATA_OOS obj[OBJ_MAX_OOS];
218 }FLASH_DATA_STRUCT;
219 #pragma pack()

```

Fig.4-87 FLASH_DATA_STRUCT FlashData

FlashData はシステム RAM 上に配置していますが運用中に更新され、xSPI0 ドライバにより QSPI フラッシュメモリに保存されます。リセットすると構成データは QSPI フラッシュメモリから読み出されてシステム RAM 上の FlashData に展開されます。FlashData の詳細は以下を参照ください。

Fig.3-5 Memory layout

5.3 初期設定コマンド

user\renesas\application\configurable_property.c

user\renesas\application\configurable_property.h

VTS に表示されているサービスツリーから File Access > AtomicReadFile をクリックします。

- File ID は File オブジェクトタイプとインスタンス番号です。

Stream Access

- Start Position はファイル先頭からの読み取りを開始するオクテット数です。0 は先頭を表します。
- Octet Count は Start Position から始まる、ファイルから読み取られるオクテット数です。下図の例のように大きめの値を設定すると実際のファイルサイズを読み取ります。

Record Access

レコード指向のファイルアクセスは未サポートです。

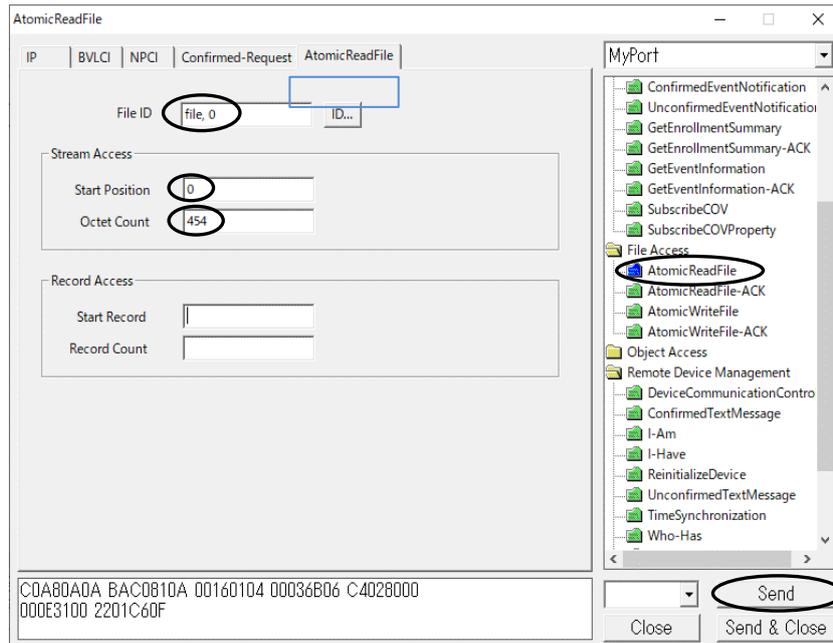


Fig.4-88 AtomicReadFile parameters

次のキャプチャ画面では VTS クライアントから、B-BC に対して AtomicReadFile サービスを要求し、B-BC からファイルデータを含む Ack を応答しています。
Length: 148 オクテット、End Of File: TRUE が表示されています。

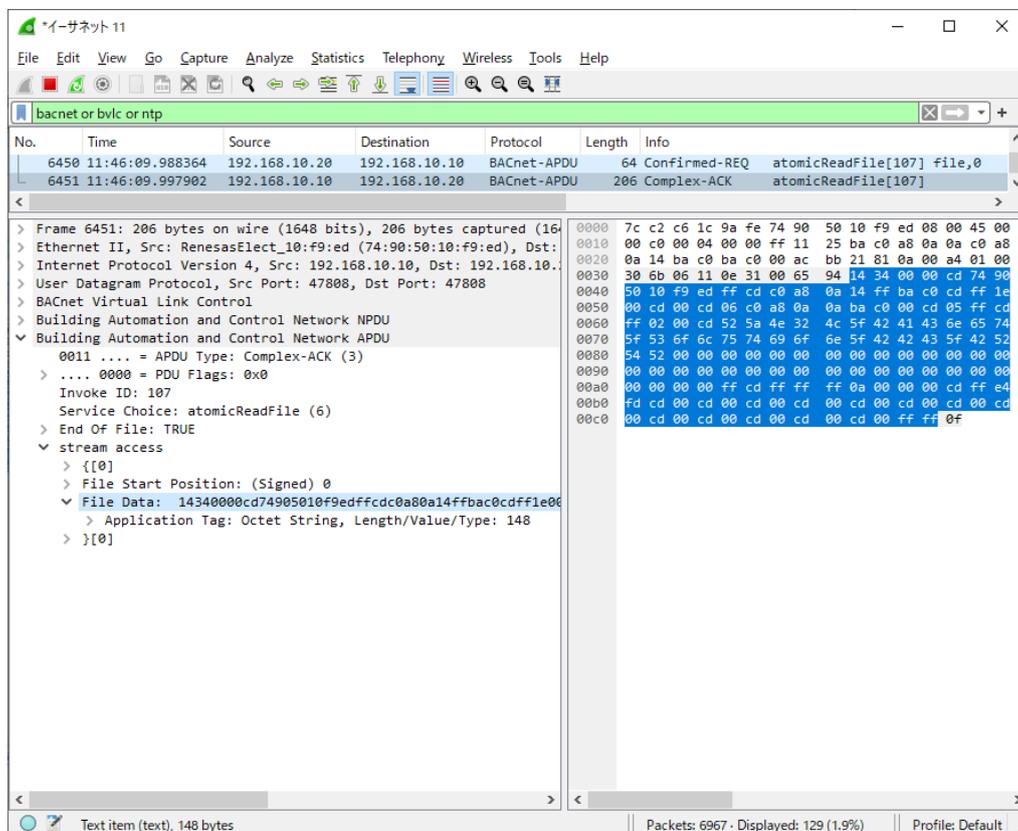


Fig.4-89 Capturing AtomicReadFile

4.5.12 AtomicWriteFile

AtomicWriteFile サービスは B-BC 内部の構成データを上書きします。クライアントは以前に読み出したバックアップファイルを AtomicWriteFile サービスによって、B-BC に転送し、B-BC は受信ファイルデータを内部の構成データに上書きして復元します。ファイルの内容と形式はローカルマターです。4.5.11 章を参照ください。

VTS に表示されているサービスツリーから File Access > AtomicWriteFile をクリックします。

- File ID は File オブジェクトタイプとインスタンス番号です。

Stream Access

- Start Position はファイルの先頭からの書き込みを開始するオクテット数です。0 は先頭を表します。Start Position に-1 を設定すると、現在のファイルの終わりから追加する操作を示します。
- Data はファイルに書き込まれる OCTET STRING で構成されます。1434000cd74....などの Hex データストリームを設定します。

Record Access

レコード指向のファイルアクセスは未サポートです。

上記パラメータの Data について、作り方を以下に示します。

Wireshark ログの AtomicReadFile サービスに対する B-BC からの Complex-ACK 行(以下の例では No.7902)をクリックして選択します。左下 window の > File Data: にカーソルを置き右クリックすると表示されるメニューの Copy > ...as a Hex Stream をクリックします。VTS の AtomicWriteFile パラメータダイアログの Data に貼り付けて Send します。

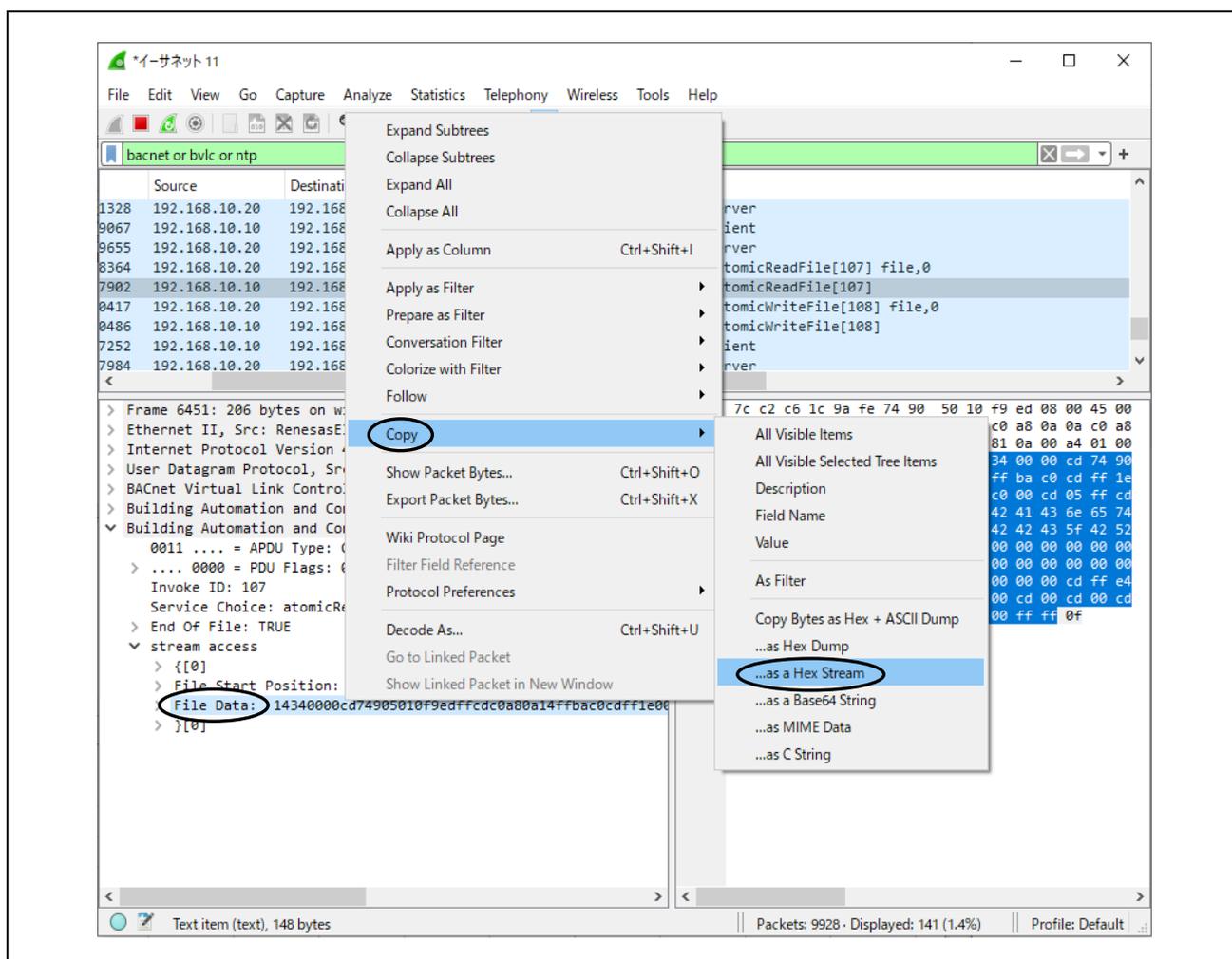


Fig.4-90 AtomicWriteFile parameters

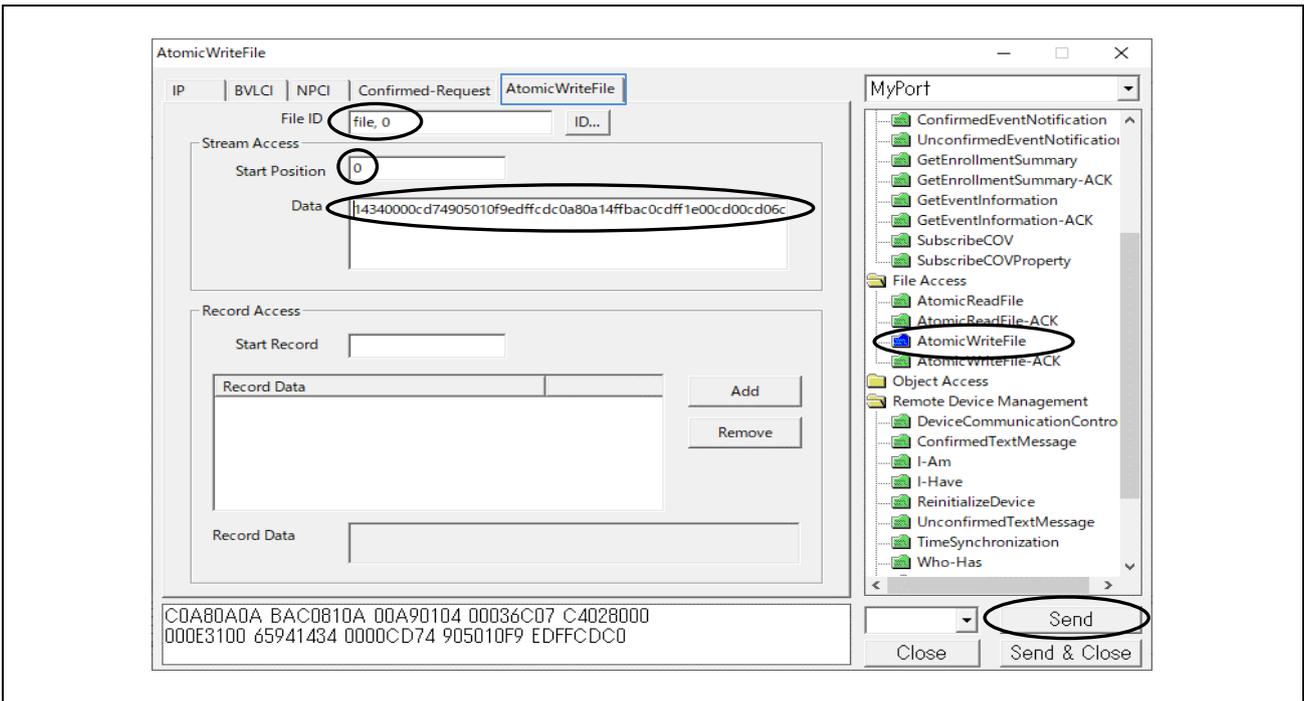


Fig.4-91 AtomicWriteFile parameters

次のキャプチャ画面では VTS クライアントから、B-BC に対して AtomicWriteFile サービスを要求し、B-BC から Ack を応答しています。

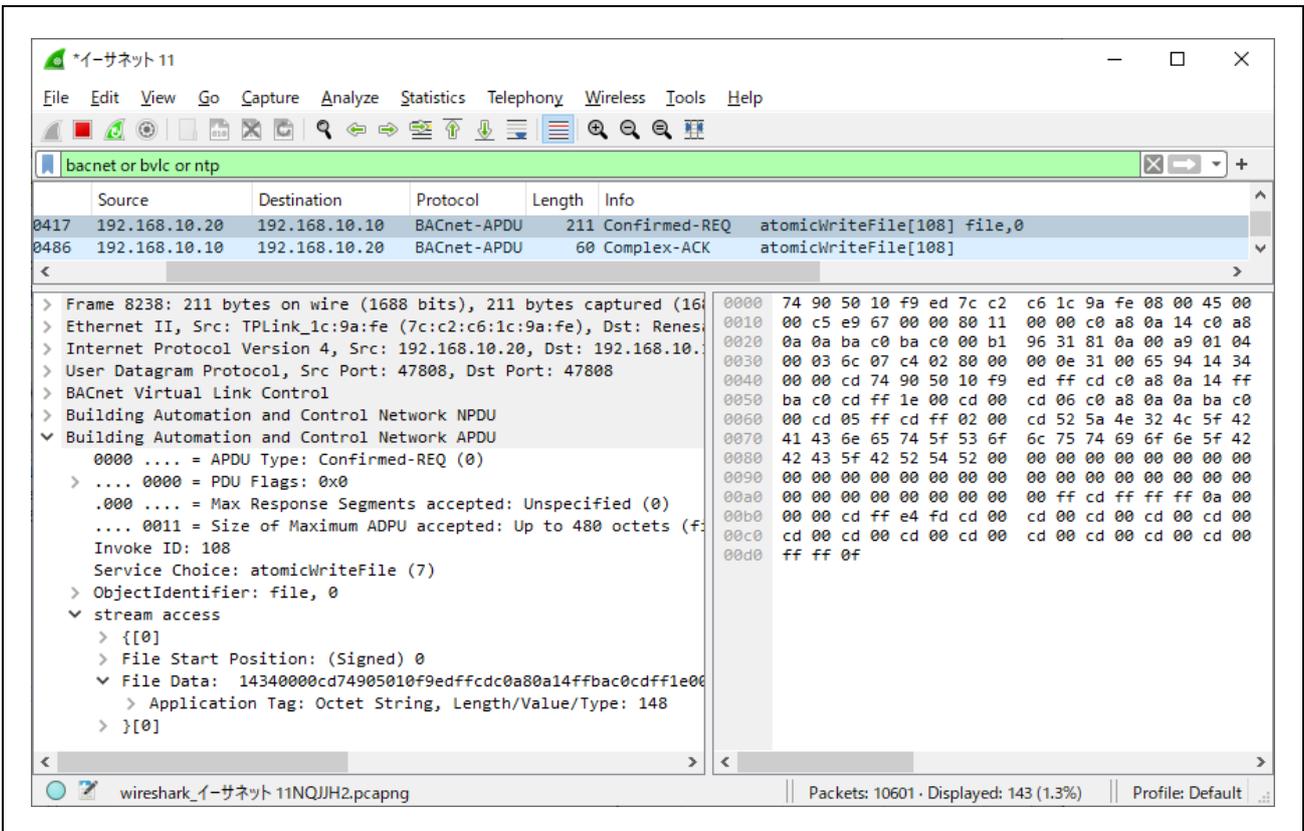


Fig.4-92 Capturing AtomicWriteFile

4.6 Yabe による通信確認

以降の動作確認は VTS から可能ですが操作が容易な Yabe を使って説明します。

Windows のスタートを開いて Yabe をクリックすると Yabe が立ち上がります。

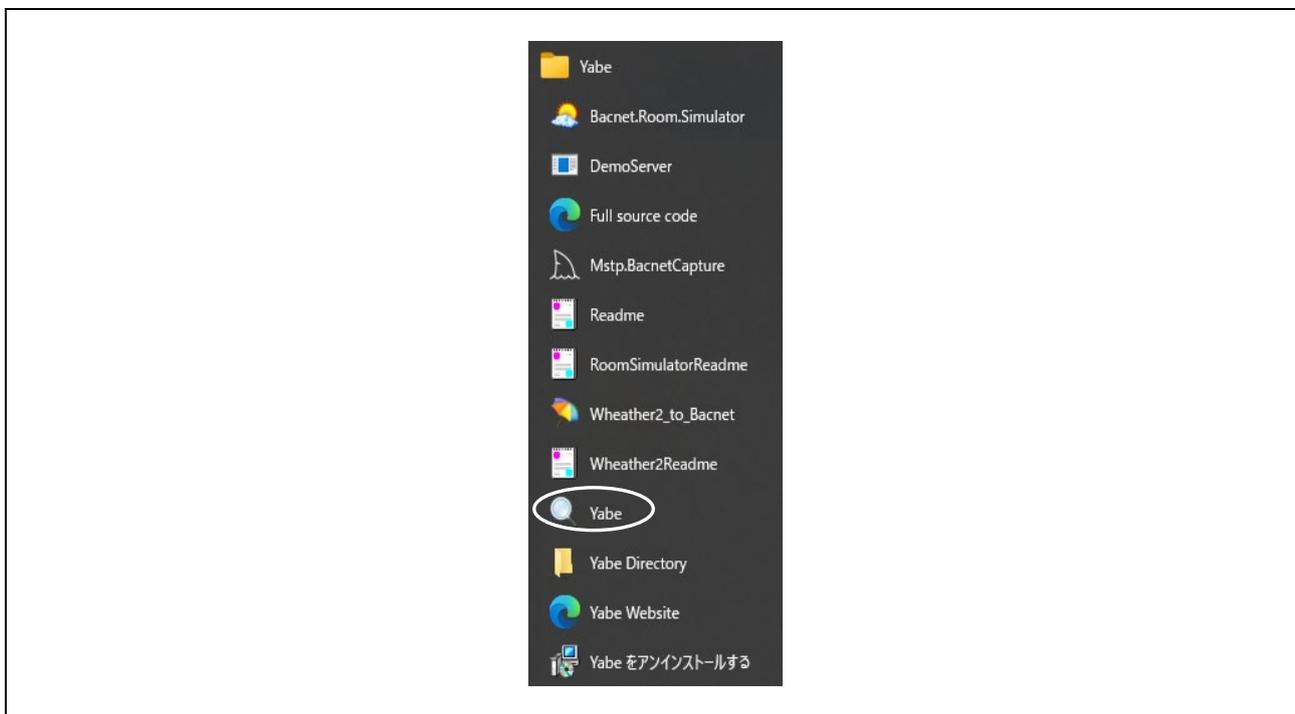


Fig.4-93 Launch Yabe

Yabe からの BACnet/IP 通信を行う場合について説明します。Yabe を立ち上げ後デバイスを追加します。

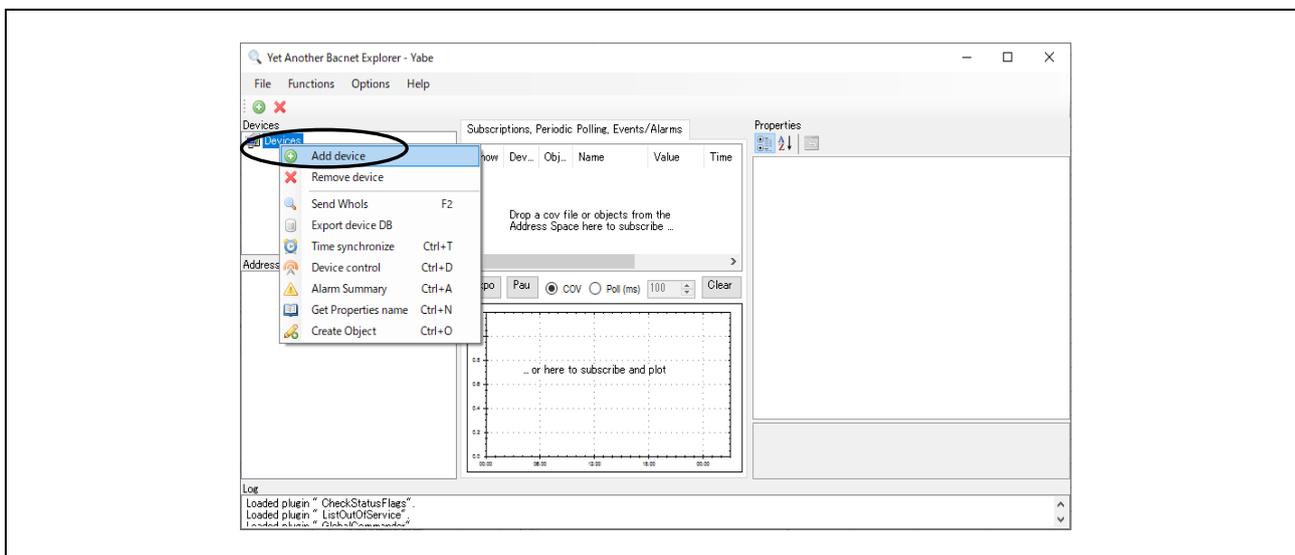


Fig.4-94 Yabe Add device(1)

表示されたダイアログの Port に BAC0(47808)を入力し、PC クライアントの IP アドレスを選択します。Start をクリックして通信を開始します。

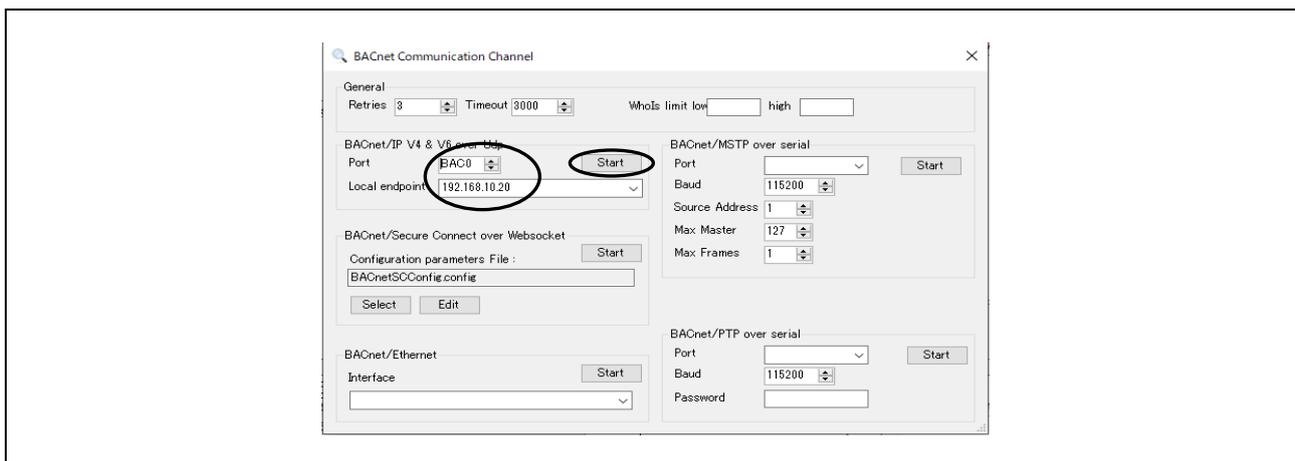


Fig.4-95 Yabe Add device(2)

Devices ウィンドウに表示された Device 10 が B-BC、Device 100 が B-SS、Device 2 は Yabe です。ここで、Yabe の Options > Settings 画面から YabeDeviceID を 2 に設定しています。(4.6.1 章参照)

Device 10 をクリックします。

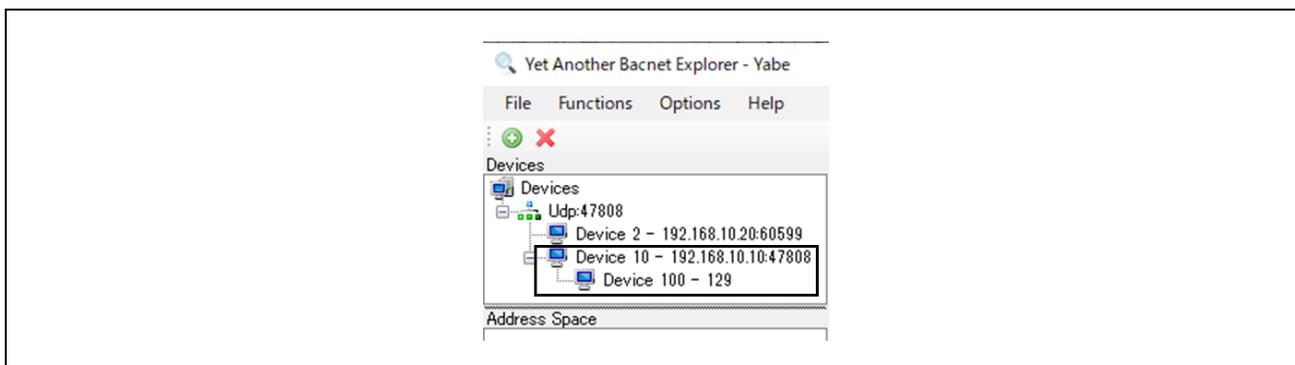


Fig.4-96 Yabe Add device(3)

Address Space ウィンドウに Device 10 のオブジェクトリストが表示されます。

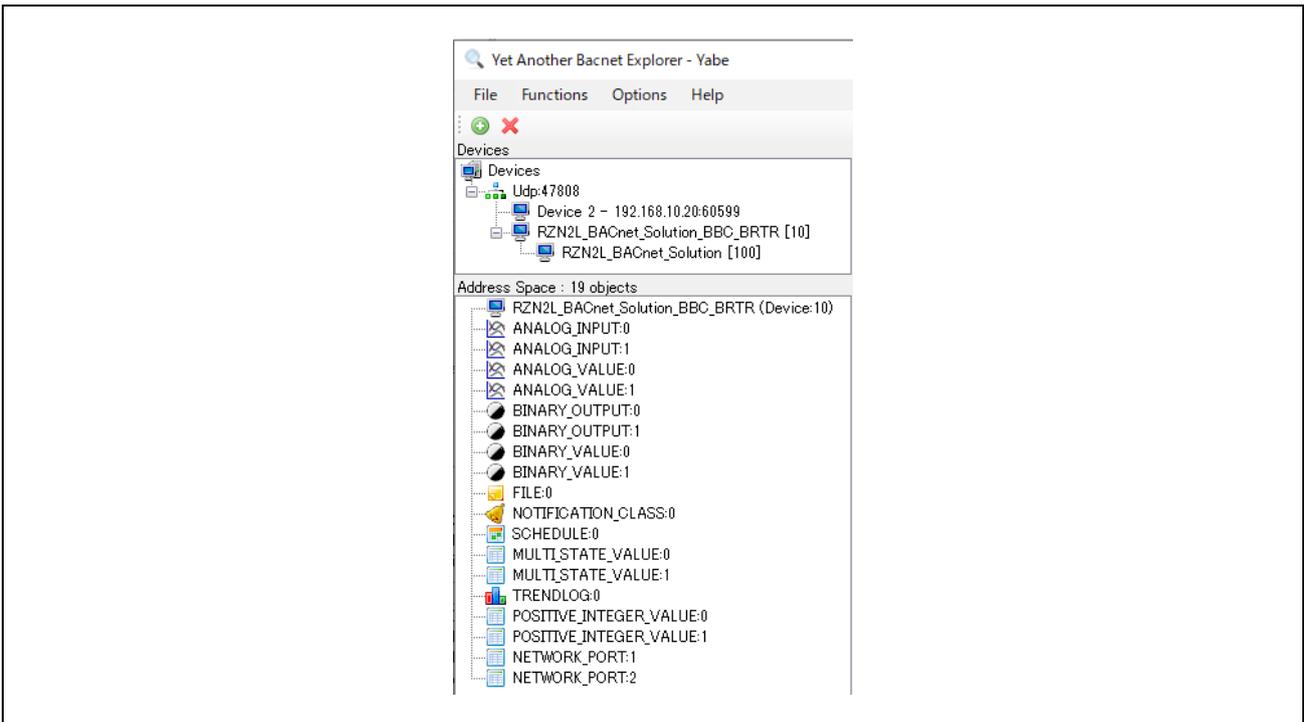


Fig.4-97 Yabe Add device(4)

Devices ウィンドウに表示された Device 100 をクリックします。

Address Space ウィンドウは Device 100 のオブジェクトリスト表示に切り替わります。

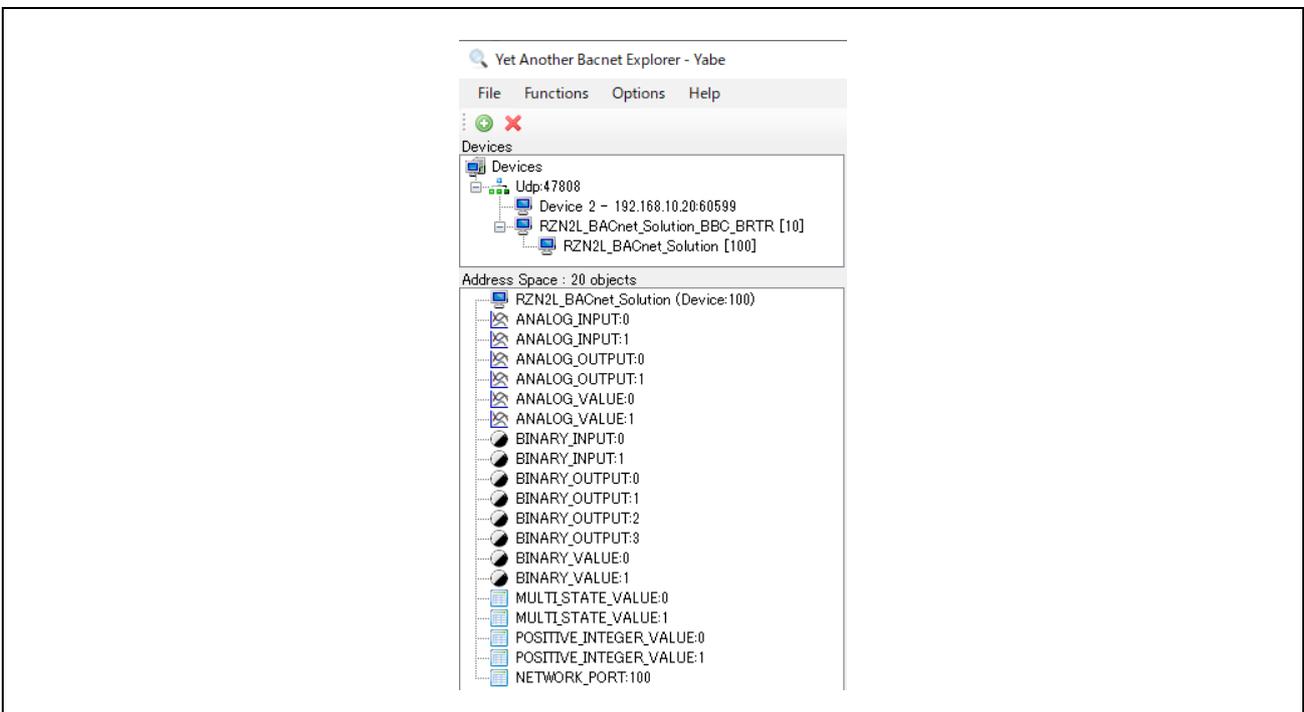


Fig.4-98 Yabe Add device(5)

4.6.1 TimeSynchronization / UTCTimeSynchronization

VTS からの TimeSynchronization / UTCTimeSynchronization を 4.5.3 章で説明しましたが Yabe からの設定手順について説明します。

時刻同期サービスは TimeSynchronization か UTCTimeSynchronization かを Options > Settings から選択します。

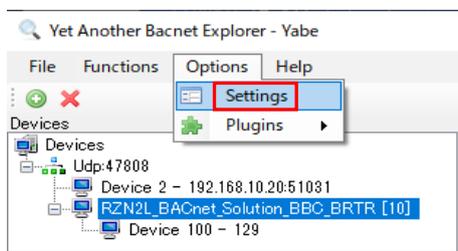


Fig.4-99 Selecting a time synchronization service(1)

次の Settings 画面の TimeSynchronize_UTC を False にすると TimeSynchronization が選択され、True にすると UTCTimeSynchronization が選択されます。

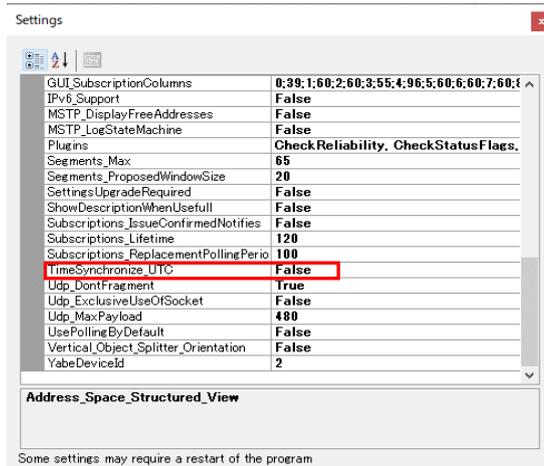


Fig.4-100 Selecting a time synchronization service(2)

Devices ウィンドウの B-BC を選択し、右クリックすると表示されるコマンドから Time_synchronize を選択します。OK ポップアップ画面の OK をクリックします。

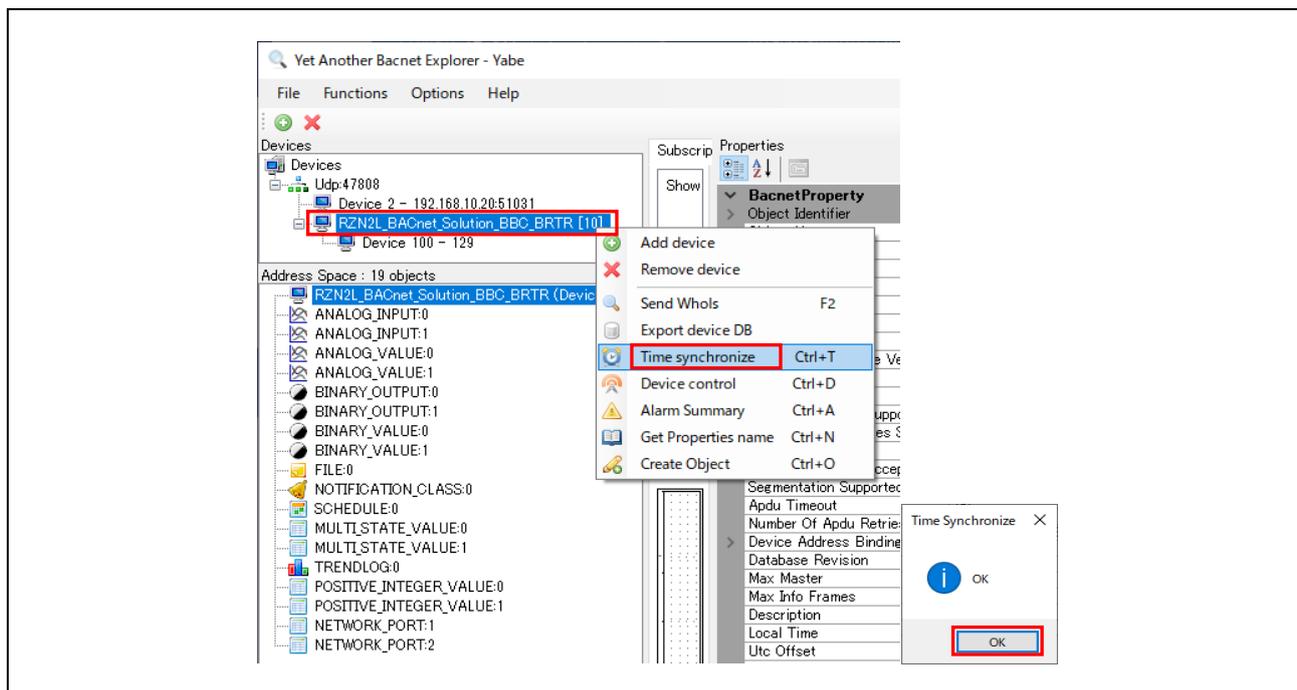


Fig.4-101 Selecting a time synchronization service(2)

同様に、B-SS に対しても Time_synchronize を行います。

4.6.2 B-BC からの B-SS 制御

B-BC デバイスは DS-RP-A、DS-WP-A プロファイルをサポートしており、自らが ReadProperty および WriteProperty サービスを Initiate することができます。本章では、DS-RP-A、DS-WP-A をもつ B-BC から外部デバイスである B-SS を制御します。

本章で取り扱う TrendLog や Schedule オブジェクトはタイムスタンプを使用しますので、事前に TimeSynchronization もしくは UTCTimeSynchronization サービス要求を B-BC、B-SS に対して行ってください。(4.6.1 章)

4.6.2.1 Trending & ReadRange

センサから入力した B-SS の AnalogInput,0 オブジェクトの PresentValue プロパティを B-BC の TrendLog,0 オブジェクトの LogBuffer プロパティにログレコードとして収集するために B-BC は B-SS に ReadProperty サービスを要求(DS-RP-A)します。

Devices ウィンドウに表示された RZN2L_BACnet_Solution_BBC_BRTR [10]をクリックします。

Address Space ウィンドウの TREND LOG 0(Trendlog:0)をクリックします。

Properties ウィンドウの以下のプロパティを変更します。

- Start Time に今日の日付を設定します。
- Stop Time に明日の日付を設定します。
- DeviceID の Instance を B-SS デバイスのインスタンス番号(100)に変更します。インスタンス番号が 10 の場合は B-BC 自身の AnalogInput,0 オブジェクト PresentValue がロギング対象です。この場合の注意点として B-BC 自身の AnalogInput,0 オブジェクト PresentValue を変更するには AnalogInput,0 オブジェクトの OutOfService=True にする必要があります。
- Log Interval の表示は 10[ms]単位ですが設定単位は秒単位です。
1~99 を入力すると 100(100×10[ms]=1000[ms])に丸められます。
1 秒を設定するときは 100(100×10[ms]=1000[ms]=1[s])を設定します。
デフォルトは 90000(90000×10[ms]=900000[ms]=900[s]=15[min])です。

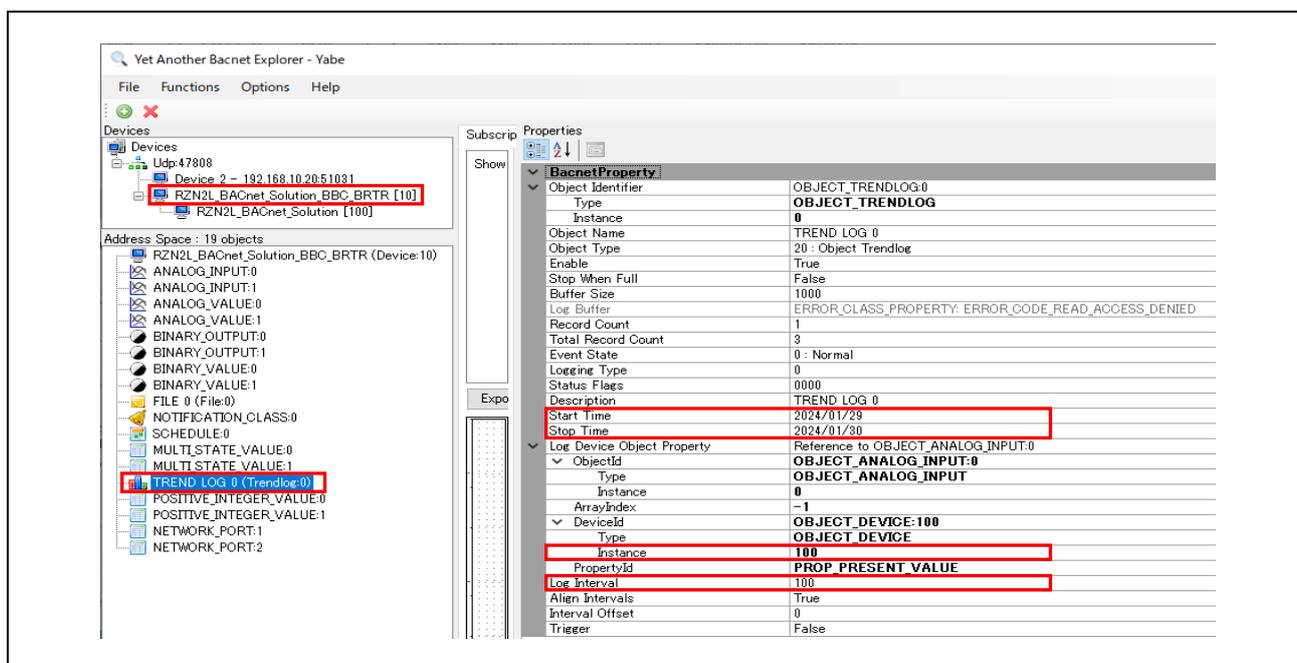


Fig.4-102 Trend Log object(1)

Address Space ウィンドウの TREND LOG 0 (Trendlog:0)をリフレッシュするために、一旦ほかのオブジェクトをクリック後、再度 TREND LOG 0 (Trendlog:0)をクリックします。Record Count プロパティ値が十分増えたことを確認します。

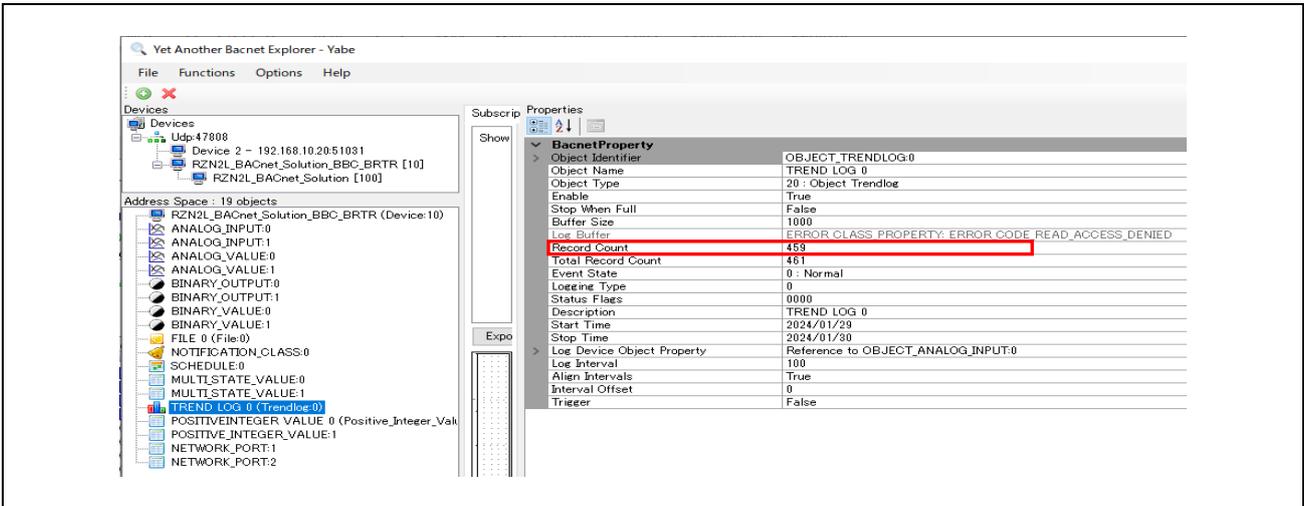


Fig.4-103 Trend Log object(2)

TREND LOG 0 (Trendlog:0)を右クリックして Show TrendLog を選択すると Yabe が ReadRange サービス要求を B-BC に行います。

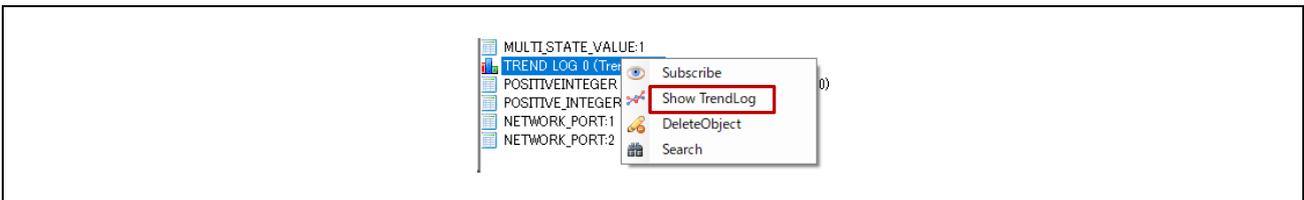


Fig.4-104 Trend Log object(3)

B-SS のセンサ入力値のタイムスタンプ付きログとグラフが表示されます。ロギング中はセンサが風を検知するようにしてください。

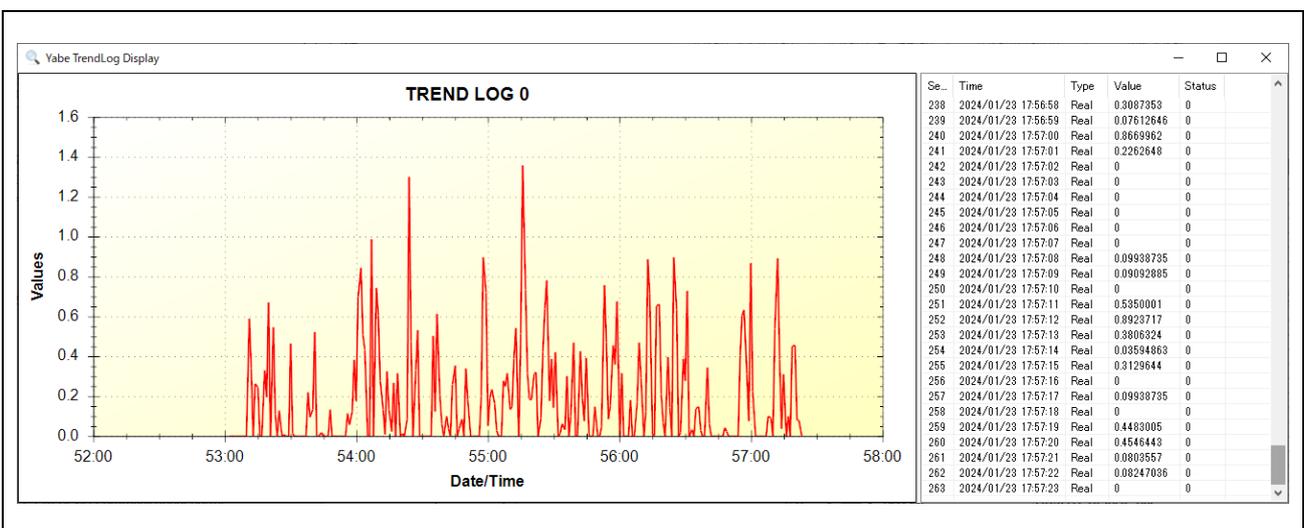


Fig.4-105 Show TrendLog

ロギングを停止する場合は Enable プロパティ値を False にします。

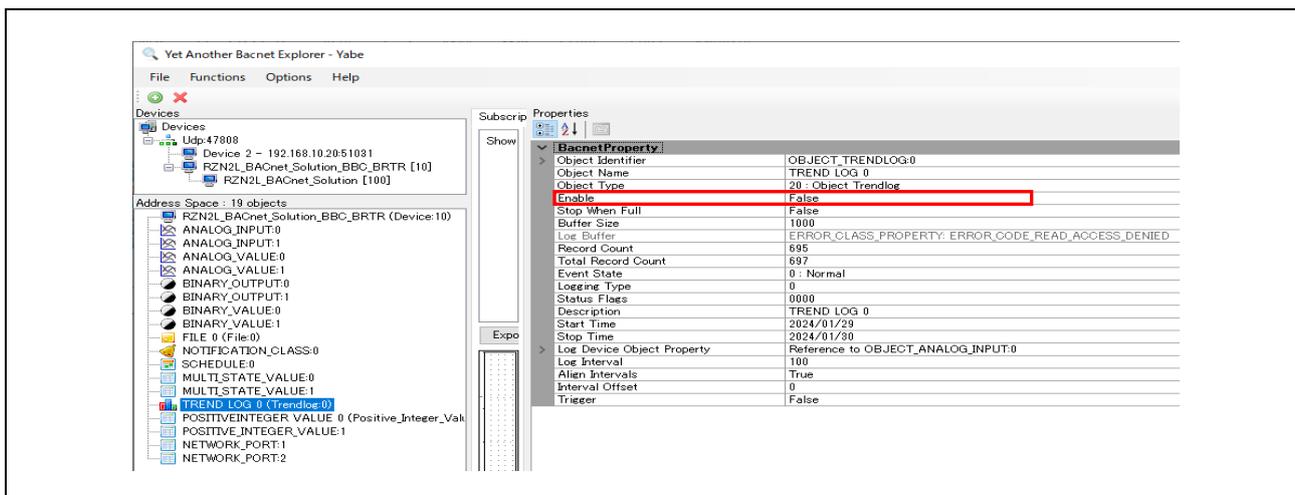


Fig.4-106 Trend Log object(4)

次の MS/TP ネットワークのキャプチャ画像は B-BC から B-SS への ReadProperty サービス要求(DS-RP-A)と、その応答を 1 秒周期で実行されていることを示しています。Source および Destination に表示された MAC アドレス 0x05 は B-BC マスターを示し、0x81 は B-SS スレーブを示しています。

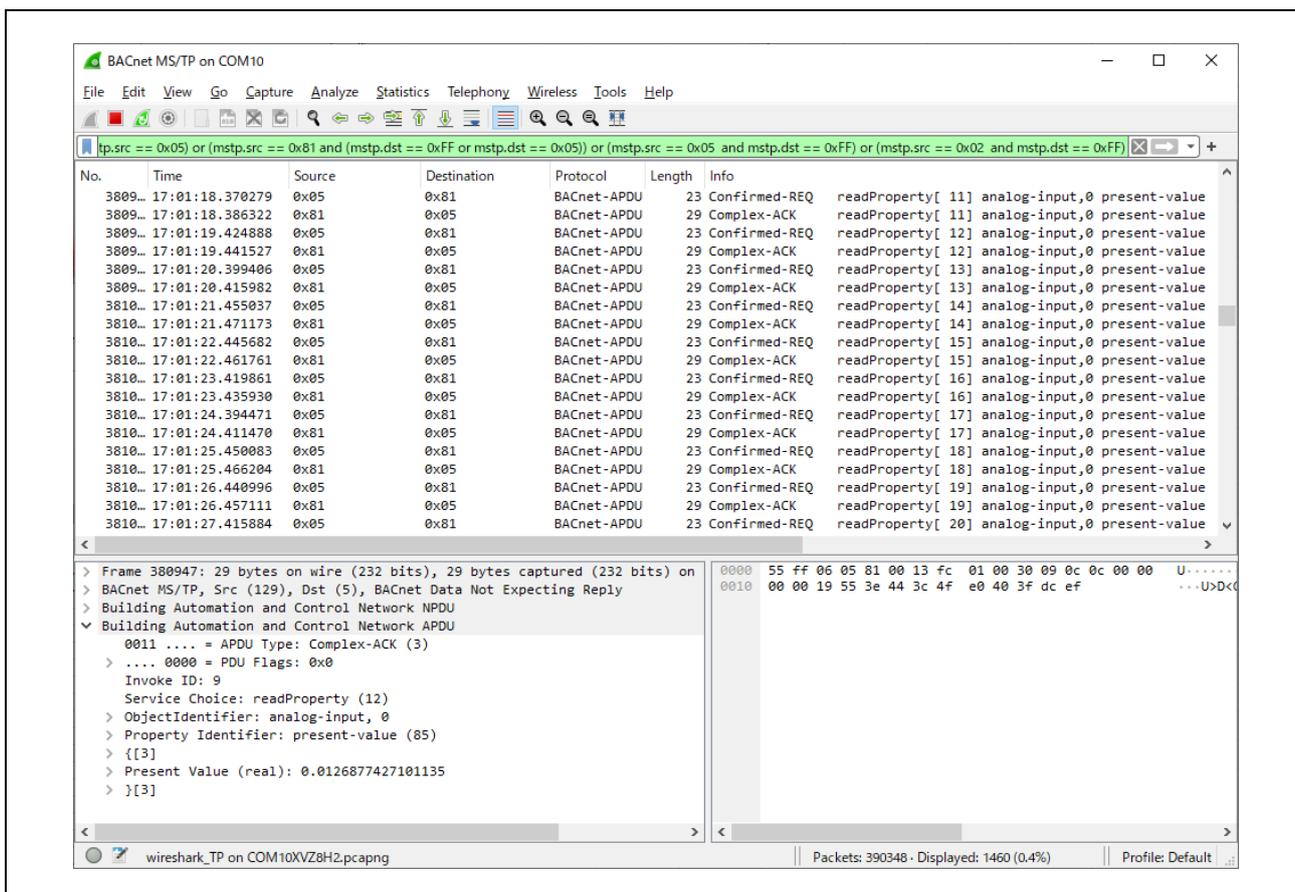


Fig.4-107 ReadProperty service request to B-SS (DS-RP-A) capture image

次のBIPネットワークのキャプチャ画像はYabeからB-BCへのReadRangeサービス要求と、その応答を示しています。

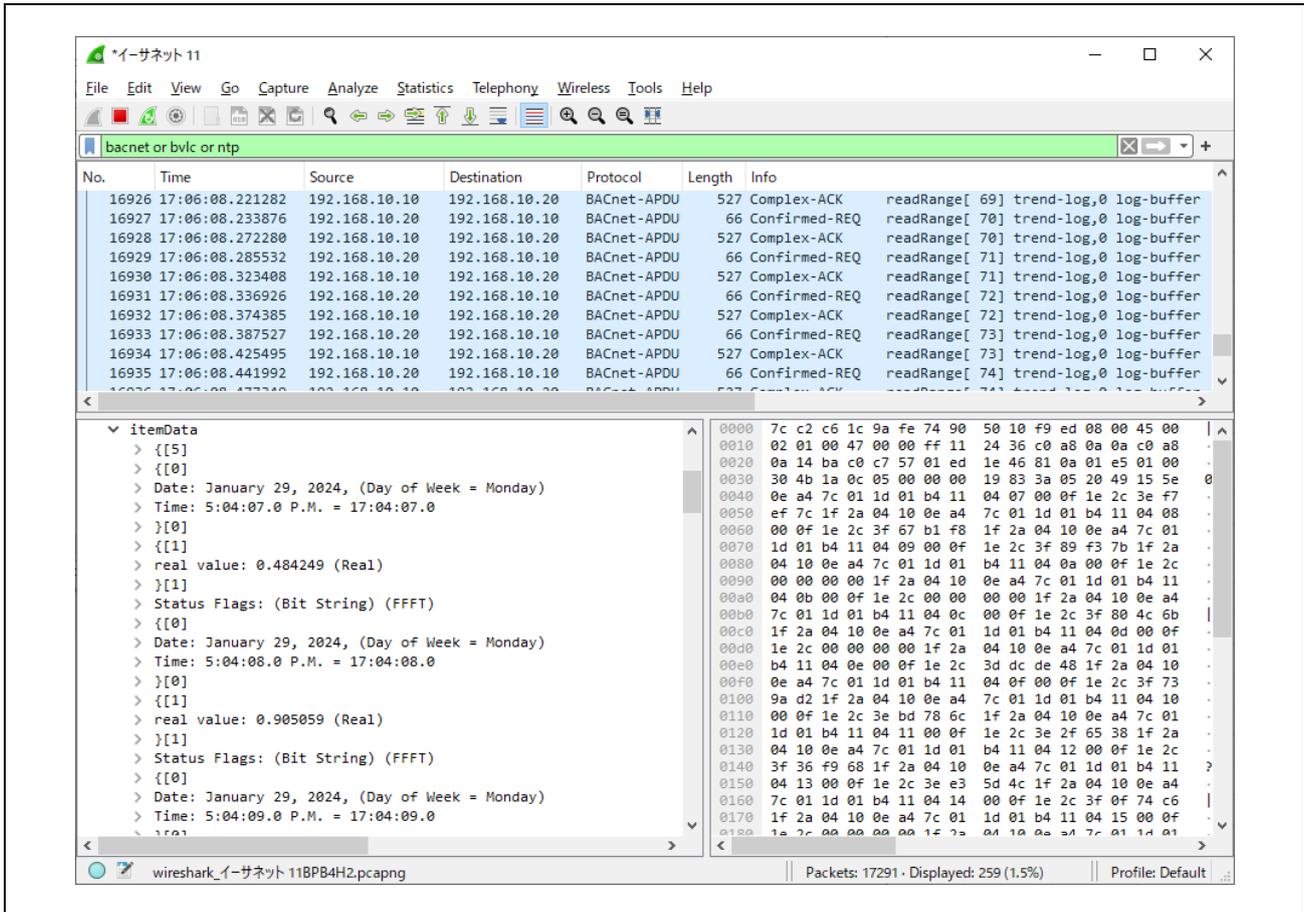


Fig.4-108 ReadRange service request capture image to B-BC

4.6.2.2 Scheduling

また、LED の点消灯に紐づけされた B-SS の BinaryOutput,0~3 オブジェクトの PresentValue プロパティを B-BC の Schedule,0 オブジェクトに設定した週単位のスケジュールに従って、変更するために B-BC は B-SS に WriteProperty サービスを要求(DS-WP-A)します。

Yabe の Devices ウィンドウに表示された RZN2L_BACnet_Solution_BBC_BRTR [10]をクリックします。

Address Space ウィンドウの SCHEDULE 0(Schedule:0)を右クリックして、Show Schedule を選択します。Properties ウィンドウの中央にある List Of Object Property References プロパティは Device,100、BinaryOutput,0 の PresentValue を Schedule 対象として割り当てられているため、変更する必要はありません。

B-SS のデバイスインスタンス番号が 100 以外に変更されたときなどは List Of Object Property References プロパティを変更してください。BinaryOutput オブジェクトだけでなく、その他の Output 系オブジェクトを Schedule 対象に変更できます。

確認済み B-SS の Output 系オブジェクトは以下です。

AnalogOutput、AnalogValue、BinaryOutput、BinaryValue、MultiStateValue、PositiveIntegerValue

また、デバイスインスタンス番号を 10 に変更すると B-BC 自身の Output 系オブジェクトを Schedule 対象とすることができます。

確認済み B-BC の Output 系オブジェクトは以下です。

AnalogValue、BinaryOutput、BinaryValue、MultiStateValue、PositiveIntegerValue

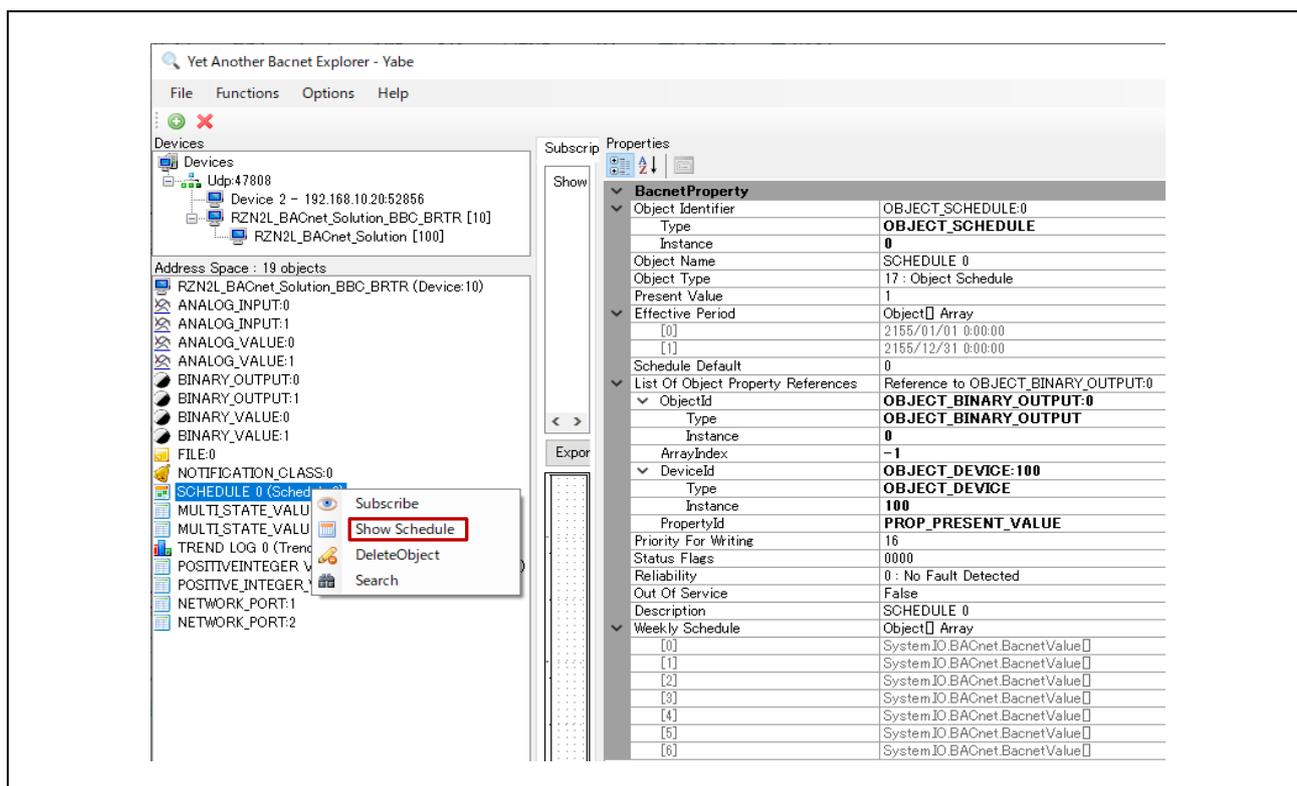


Fig.4-109 Schedule object

Show Schedule を選択すると Simple Schedule Editor が開きますのでプロパティを変更します。

- ・ Validity Start Date に今日の日付を設定します。
- ・ Validity End Date に明日の日付を設定します。

B-SS ボードの汎用 LED0 を点灯した後、消灯するための設定を行います。

- Weekly Schedules の今日の曜日を右クリックして Modify を選択して hh:mm:ss = 1 に変更します。
- Weekly Schedules の今日の曜日を右クリックして Add を選択します。hh:mm:ss = 0 を設定します。

最後に Update & Read back をクリックして Simple Schedule Editor を閉じます。

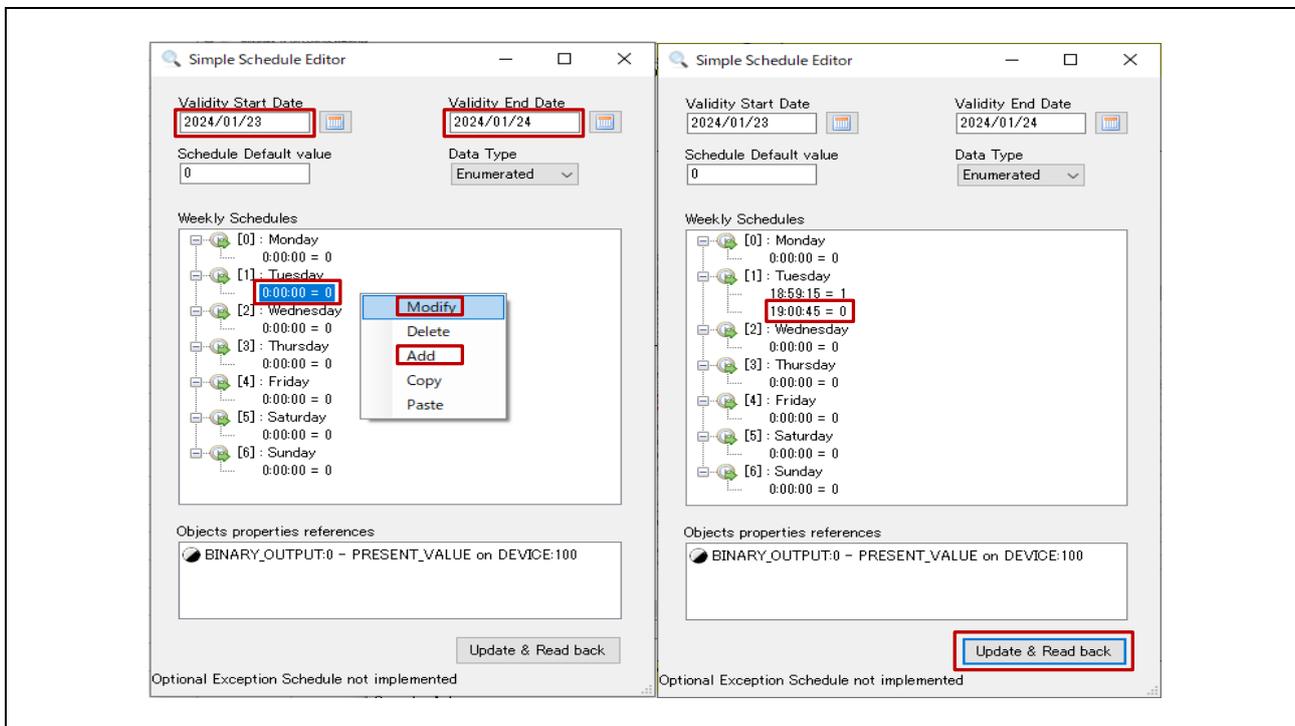


Fig.4-110 Simple Schedule Editor

次の MS/TP ネットワークのキャプチャ画面は Schedule 時刻の 18:59:15 と 19:00:45 に B-SS への WriteProperty サービス要求(DS-WP-A)と、その応答がされていることを示しています。また同時刻に B-SS ボードの LED0 が点消灯します。

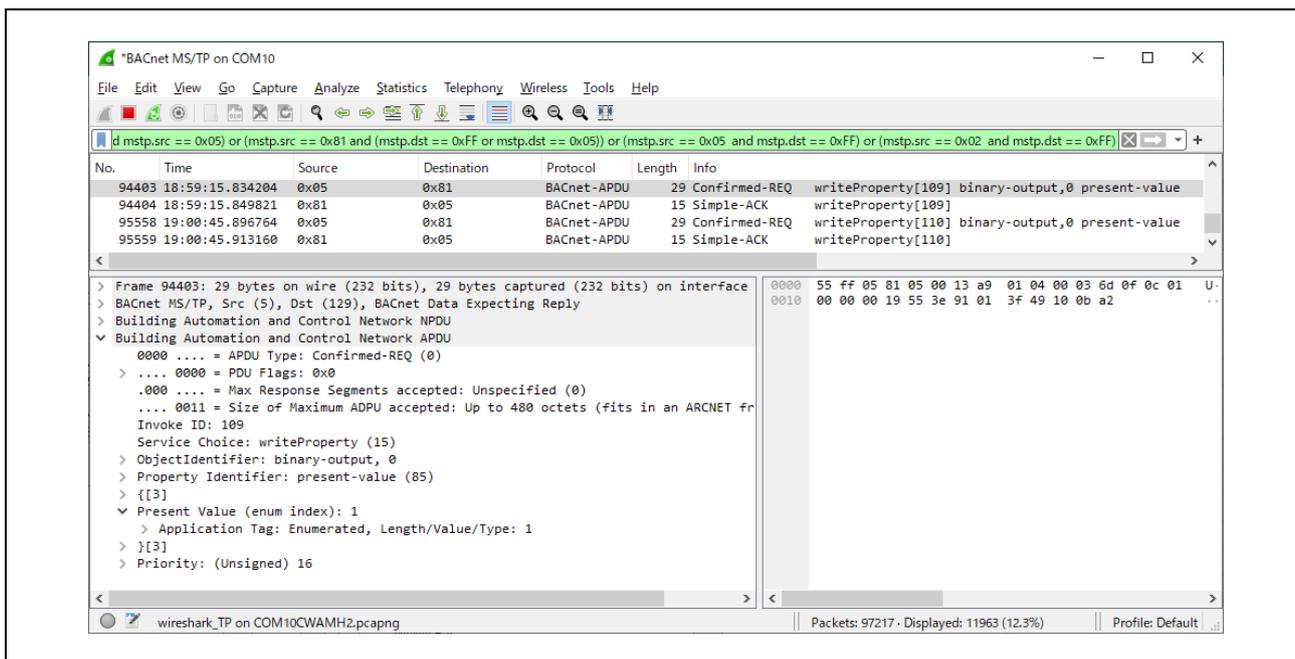


Fig.4-111 WriteProperty service request to B-SS (DS-WP-A) capture image

4.6.3 EventNotification / GetEventInformation / AcknowledgeAlarm

B-BC の AnalogInput オブジェクトは Out Of Range イベントアルゴリズムをサポートしています。PresentValue プロパティ値が Low_Limit プロパティと High_Limit プロパティで示される範囲から逸脱すると ConfirmedEventNotification サービスまたは UnconfirmedEventNotification サービスを BIP クライアントに通知します。

BIP クライアントはすべての "active event states" を取得するために GetEventInformation サービス要求を行います。

また、BIP クライアントは B-BC からのイベント通知を確認し、Ack を返したことを確認するために AcknowledgeAlarm サービス要求を行います。

Address Space ウィンドウの NOTIFICATION CLASS 0 (Notification_Class:0) を右クリックして、Show Notification を選択します。

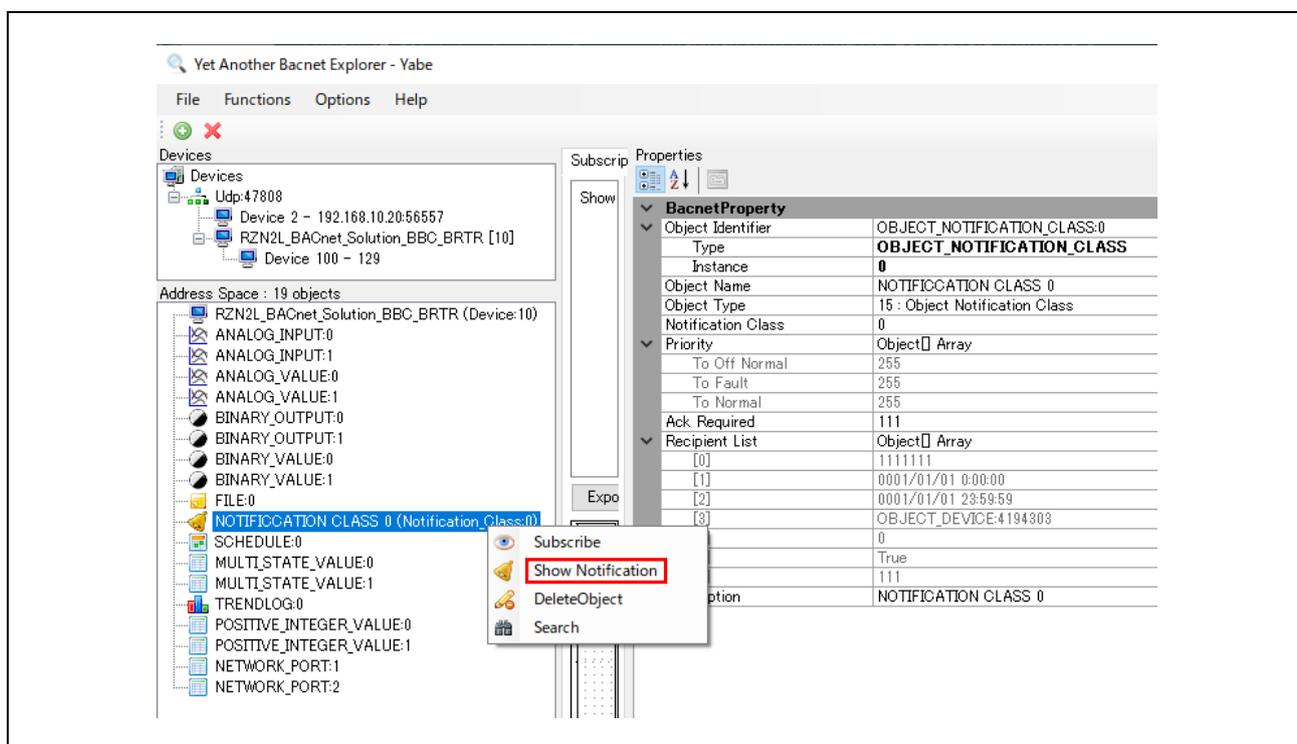


Fig.4-112 Notification Class object

Show Notification を選択すると Notification Editor が開きますのでプロパティを変更します。

- ・ Ack Required は ConfirmedEventNotification をレシピエント(BIP クライアント)に通知(✓あり)するか UnconfirmedEventNotification を通知(✓なし)するかを選択します。
- ・ Process Id はイベント通知を受信するレシピエント内のプロセスハンドルです。

EventType

- ・ To_OffNormal は正常状態でも障害状態でもない状態に遷移したときにイベント通知を行います。
- ・ To_Fault は障害状態に遷移したときにイベント通知を行います。
- ・ To_Normal は正常状態に遷移したときにイベント通知を行います。

Validity

- ・ Event 通知を有効にする曜日と時間帯です。

Receiver

- ・ Event 通知を受け取るデバイスのインスタンス番号または IP アドレスのどちらかを設定します。ただし、IP アドレスは本 B-BC サンプルソフトは未サポートのため選択できません。

Priority

- ・ 各 Event 通知の優先順位です。範囲は 0 ~ 255 です。

上記のプロパティ値は初期化済みのため少なくとも Receiver のみを変更します。例で設定した番号は Yabe のデバイスインスタンス番号です。Yabe のデバイスインスタンス番号はメニューバーの Options で設定可能です。最後に Write & Read back をクリックして Notification Editor を閉じます。

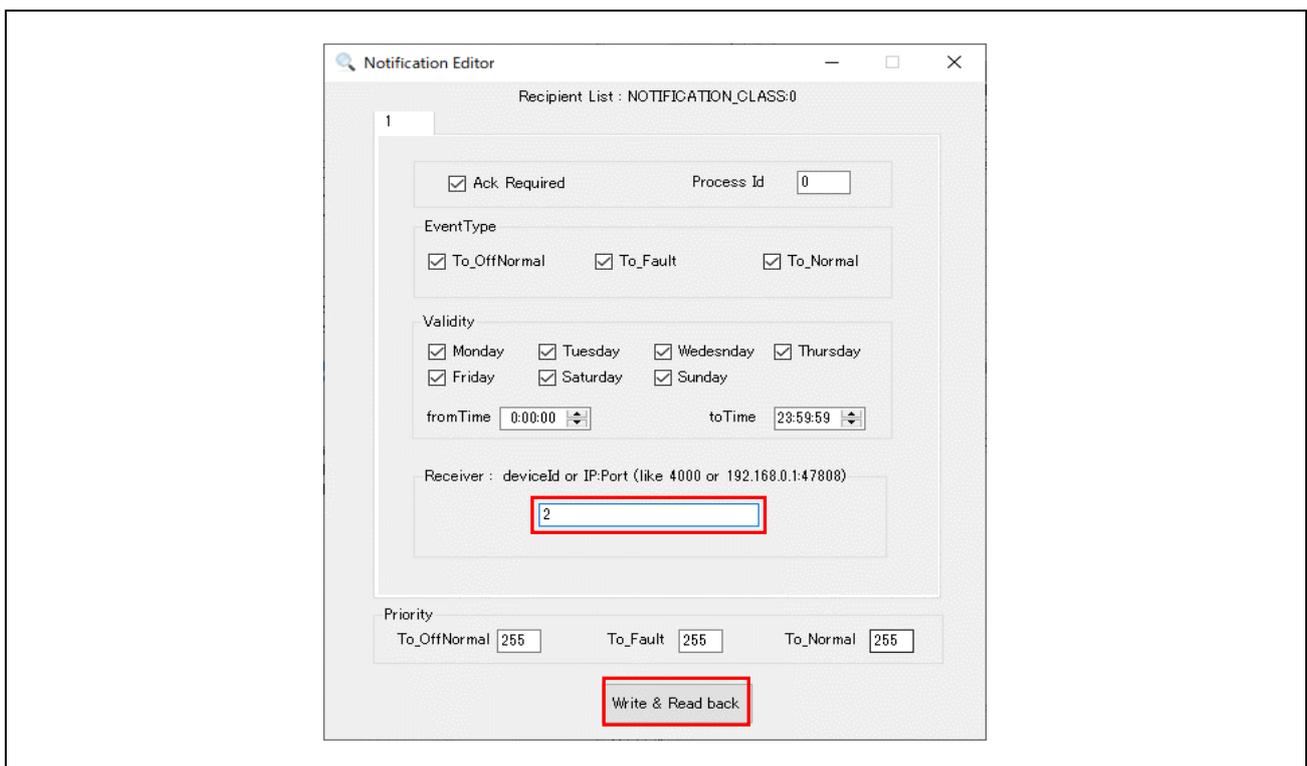


Fig.4-113 Notification Editor

次に Address Space のツリーから ANALOG INPUT 0 (Analog_Input:0)をクリックして Properties ウィンドウに表示される Out Of Service を True に選択します。Out Of Service を True にすることで Present Value を変更可能になります。続けて Present Value に High Limit(100)を超過するように 100.1 を設定します。

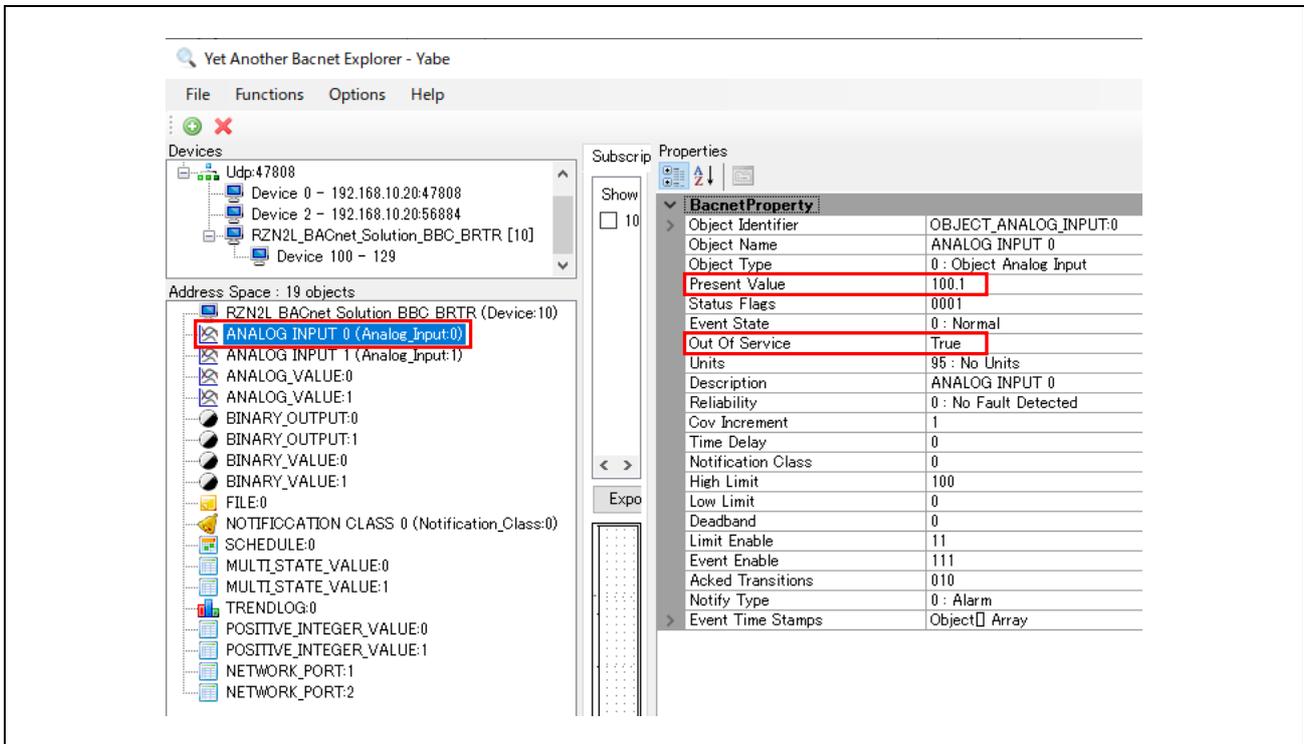


Fig.4-114 AnalogInput,0 object(1)

次に正常遷移させるために Present Value に High Limit(100)未満の 99.9 を設定します。

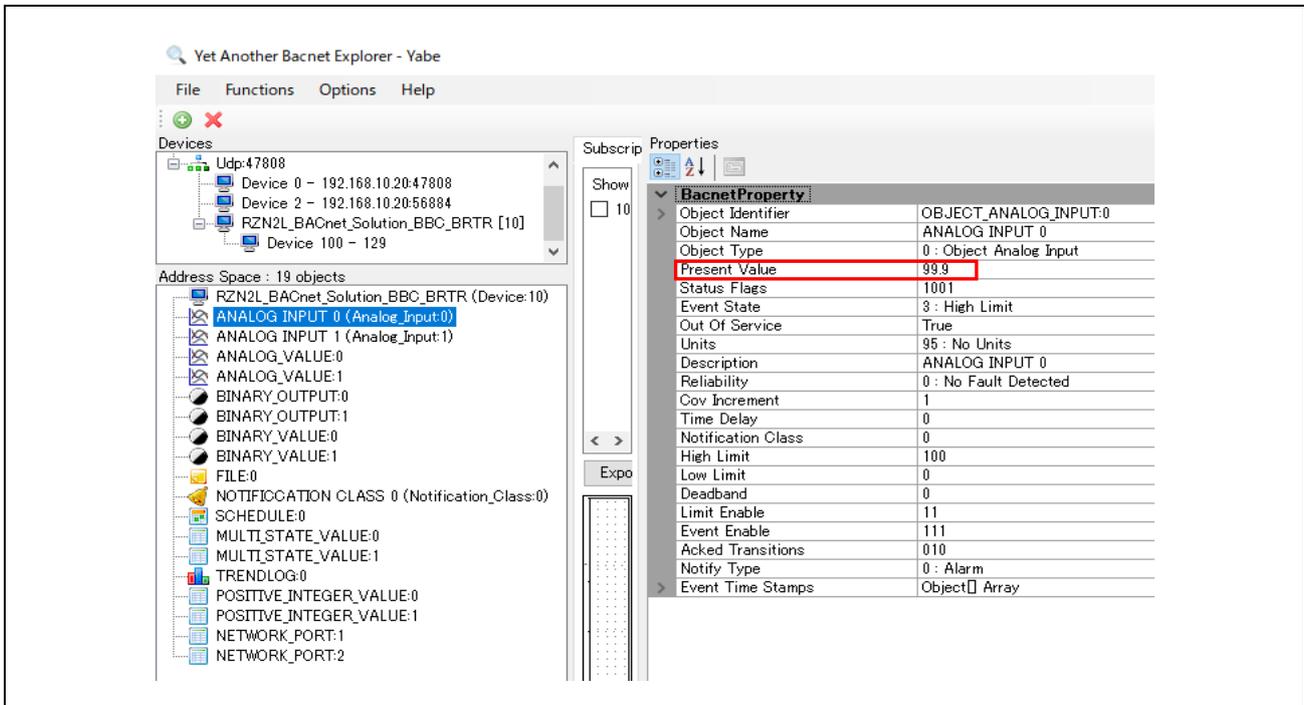


Fig.4-115 AnalogInput,0 object(2)

続けて、Devices ウィンドウの B-BC を右クリックして、Alarm Summary を選択します。

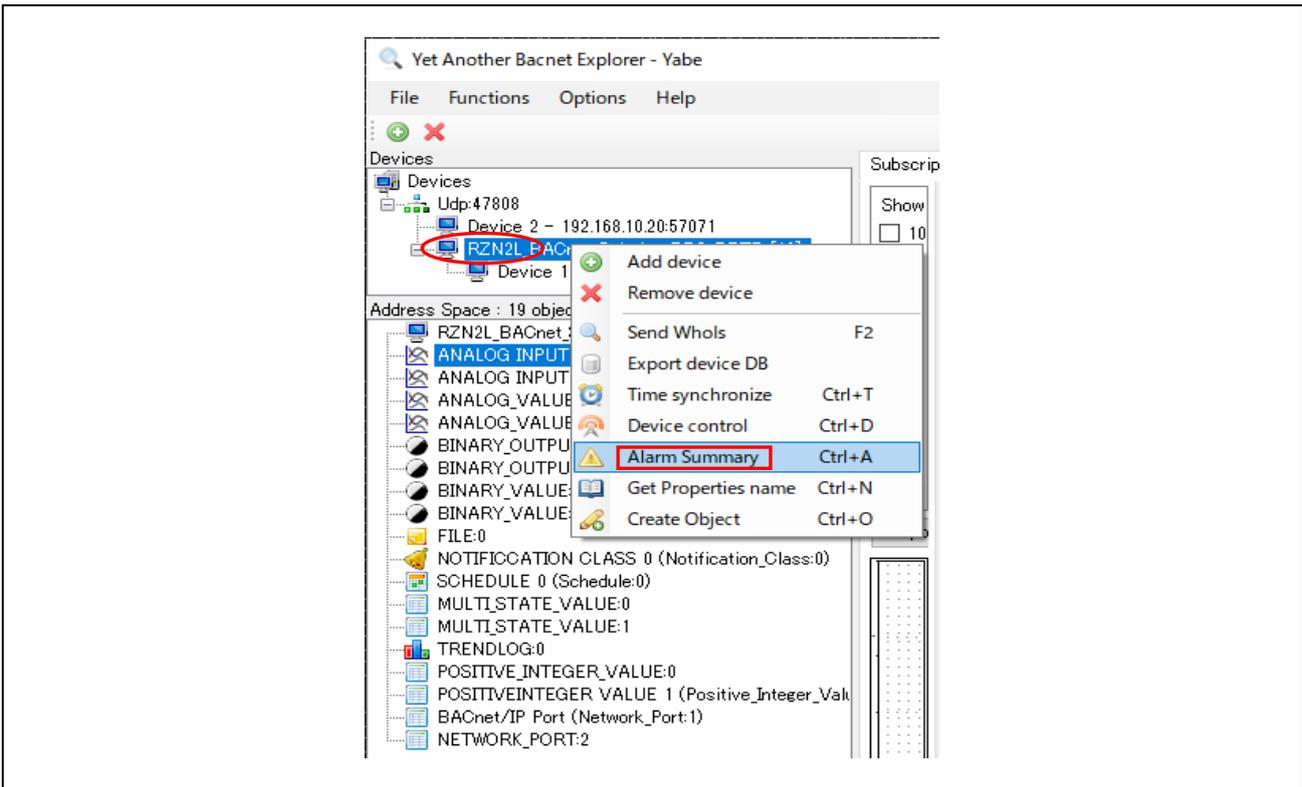


Fig.4-116 Select Alarm Summary(1)

Fig.4-117 の左側のダイアログが表示されますので Event 発生と復帰のタイムスタンプと右側の AnalogInput,0 オブジェクトの Event Time Stamps との一致を確認できます。
Ack selected alarm(s)をクリックしてダイアログを閉じます。

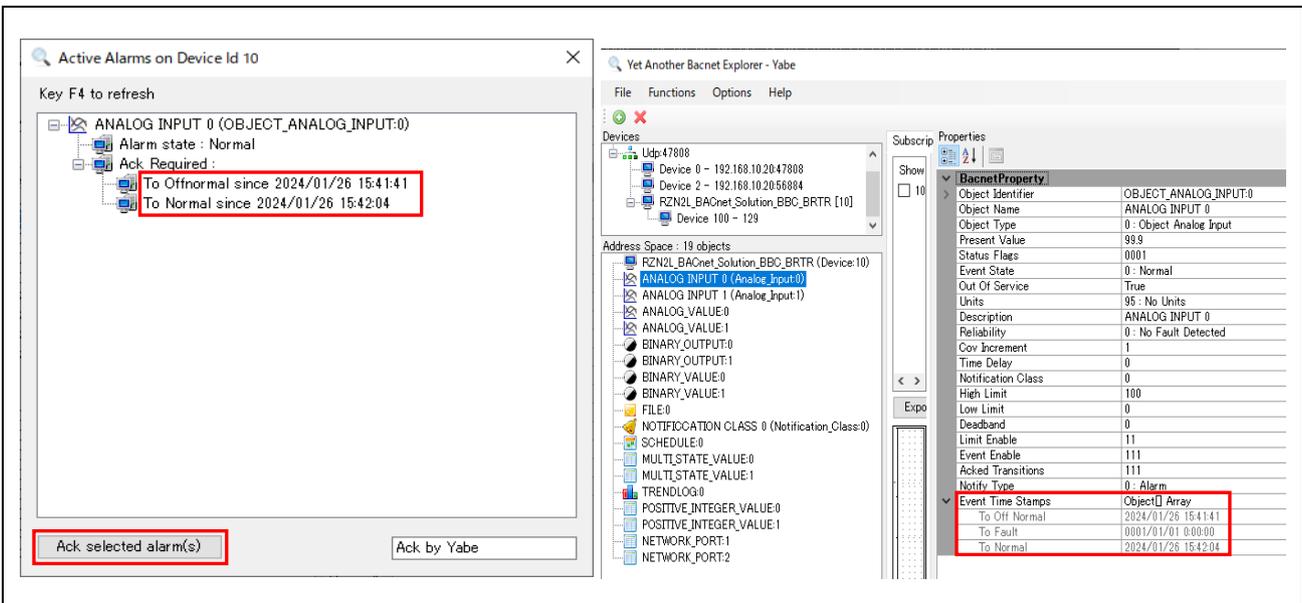


Fig.4-117 Select Alarm Summary(2)

次の Wireshark キャプチャ画像は説明した一連の手順を行ったときのサービス要求と Ack が確認できます。No.14074 の Who-Is サービス要求は B-BC が Notification Editor で設定した Recipient の IP アドレスを取得するために行っています。(DM-DDB-A)

No.14075 の I-Am は Recipient(Yabe)が応答しています。B-BC は、この I-Am からデバイスインスタンス番号(2)の Recipient の IP アドレスを紐づけます。(DM-DDB-A)

No.14076 は OffNormal に遷移したときの B-BC からの ConfirmedEventNotification です。

No.14077 は Recipient からの Ack です。

No.14096 は Normal に遷移したときの B-BC からの ConfirmedEventNotification です。

No.14114 は Recipient からの GetEventInformation サービス要求です。これは手順で説明した Alarm Summary を選択したときがトリガとなるサービス要求です。

No.14145 は B-BC からの結果応答 Ack です。

No.14140 と 14142 は Recipient からの AcknowledgeAlarm サービス要求です。これは手順で説明したダイアログの Ack selected alarm(s)をクリックしたときがトリガとなるサービス要求です。B-BC から通知した Offnormal 遷移と Normal 遷移の 2 回分の Event 通知に対する AcknowledgeAlarm サービス要求です。

No.14144 は AcknowledgeAlarm サービス要求で渡された最新イベントのタイムスタンプが B-BC 側と一致したことを通知する ConfirmedEventNotification です。

No.	Time	Source	Destination	Protocol	Length	Info
14040	15:41:01.079354	192.168.10.20	192.168.10.10	BACnet-APDU	67	Confirmed-REQ writeProperty[5] notification-class,0 priority
14041	15:41:01.122285	192.168.10.10	192.168.10.20	BACnet-APDU	60	Simple-ACK writeProperty[5]
14042	15:41:01.124945	192.168.10.20	192.168.10.10	BACnet-APDU	85	Confirmed-REQ writeProperty[6] notification-class,0 recipient-list device,2
14043	15:41:01.172281	192.168.10.10	192.168.10.20	BACnet-APDU	60	Simple-ACK writeProperty[6]
14044	15:41:01.175894	192.168.10.20	192.168.10.10	BACnet-APDU	63	Confirmed-REQ readPropertyMultiple[7]
14045	15:41:01.222272	192.168.10.10	192.168.10.20	BACnet-APDU	96	Complex-ACK readPropertyMultiple[7]
14055	15:41:14.785041	192.168.10.20	192.168.10.10	BACnet-APDU	61	Confirmed-REQ readPropertyMultiple[8]
14056	15:41:14.803406	192.168.10.10	192.168.10.20	BACnet-APDU	269	Complex-ACK readPropertyMultiple[8]
14070	15:41:41.418082	192.168.10.20	192.168.10.10	BACnet-APDU	68	Confirmed-REQ writeProperty[9] analog-input,0 present-value
14071	15:41:41.420395	192.168.10.10	192.168.10.20	BACnet-APDU	60	Simple-ACK writeProperty[9]
14072	15:41:41.427883	192.168.10.20	192.168.10.10	BACnet-APDU	61	Confirmed-REQ readPropertyMultiple[10]
14073	15:41:41.470370	192.168.10.10	192.168.10.20	BACnet-APDU	269	Complex-ACK readPropertyMultiple[10]
14074	15:41:41.965644	192.168.10.10	192.168.10.255	BACnet-APDU	60	Unconfirmed-REQ who-Is 2 2
14075	15:41:41.967143	192.168.10.20	192.168.10.255	BACnet-APDU	67	Unconfirmed-REQ i-Am device,2
14076	15:41:41.975244	192.168.10.10	192.168.10.20	BACnet-APDU	135	Confirmed-REQ confirmedEventNotification[1] device,10 analog-input,0
14077	15:41:42.005927	192.168.10.20	192.168.10.10	BACnet-APDU	51	Simple-ACK confirmedEventNotification[1]
14092	15:42:04.893371	192.168.10.20	192.168.10.10	BACnet-APDU	68	Confirmed-REQ writeProperty[11] analog-input,0 present-value
14093	15:42:04.896816	192.168.10.10	192.168.10.20	BACnet-APDU	60	Simple-ACK writeProperty[11]
14094	15:42:04.908054	192.168.10.20	192.168.10.10	BACnet-APDU	61	Confirmed-REQ readPropertyMultiple[12]
14095	15:42:04.946959	192.168.10.10	192.168.10.20	BACnet-APDU	269	Complex-ACK readPropertyMultiple[12]
14096	15:42:04.956818	192.168.10.10	192.168.10.20	BACnet-APDU	153	Confirmed-REQ confirmedEventNotification[2] device,10 analog-input,0
14097	15:42:04.988255	192.168.10.20	192.168.10.10	BACnet-APDU	51	Simple-ACK confirmedEventNotification[2]
14114	15:42:29.115625	192.168.10.20	192.168.10.10	BACnet-APDU	52	Confirmed-REQ getEventInformation[13]
14115	15:42:29.148581	192.168.10.10	192.168.10.20	BACnet-APDU	116	Complex-ACK getEventInformation[13] analog-input,0
14140	15:43:13.254837	192.168.10.20	192.168.10.10	BACnet-APDU	103	Confirmed-REQ acknowledgeAlarm[14] analog-input,0
14141	15:43:13.256672	192.168.10.10	192.168.10.20	BACnet-APDU	60	Simple-ACK acknowledgeAlarm[14]
14142	15:43:13.257696	192.168.10.20	192.168.10.10	BACnet-APDU	103	Confirmed-REQ acknowledgeAlarm[15] analog-input,0
14143	15:43:13.306732	192.168.10.10	192.168.10.20	BACnet-APDU	60	Simple-ACK acknowledgeAlarm[15]
14144	15:43:13.976788	192.168.10.10	192.168.10.20	BACnet-APDU	106	Confirmed-REQ confirmedEventNotification[3] device,10 analog-input,0
14145	15:43:13.981100	192.168.10.20	192.168.10.10	BACnet-APDU	51	Simple-ACK confirmedEventNotification[3]

Packet 14145 details:

```

> Internet Protocol Version 4, Src: 192.168.10.20, Dst: 192.168.10.10
> User Datagram Protocol, Src Port: 56884, Dst Port: 47808
> BACnet Virtual Link Control
> Building Automation and Control Network NPDU
  Building Automation and Control Network APDU
    0010 .... = APDU Type: Simple-ACK (2)
      Invoke ID: 3
      Service Choice: confirmedEventNotification (2)
  
```

Hex and ASCII data for packet 14145:

```

0000 74 90 50 10 f9 ed 7c c2 c6 1c 9a fe 08 00 45 00  t.P...E
0010 00 25 eb 32 40 00 80 11 00 00 c0 a8 0a 14 c0 a8  %-2@...
0020 0a 0a de 34 ba c0 00 11 95 91 81 0a 00 09 01 00  ...4...
0030 20 03 02  ..
  
```

Fig.4-118 EventNotification capture image

4.6.4 AtomicReadFile

VTS からの AtomicReadFile を説明した 4.5.11 章も参照ください。Yabe からの設定手順について説明します。

Yabe を使用すると PC 上に B-BC から読み出したファイルを保存できます。

Yabe の Options から Settings 画面を開きます。UdpMaxPayload を 480 に変更します。変更した場合は Yabe を再起動し、Add Device を行います。

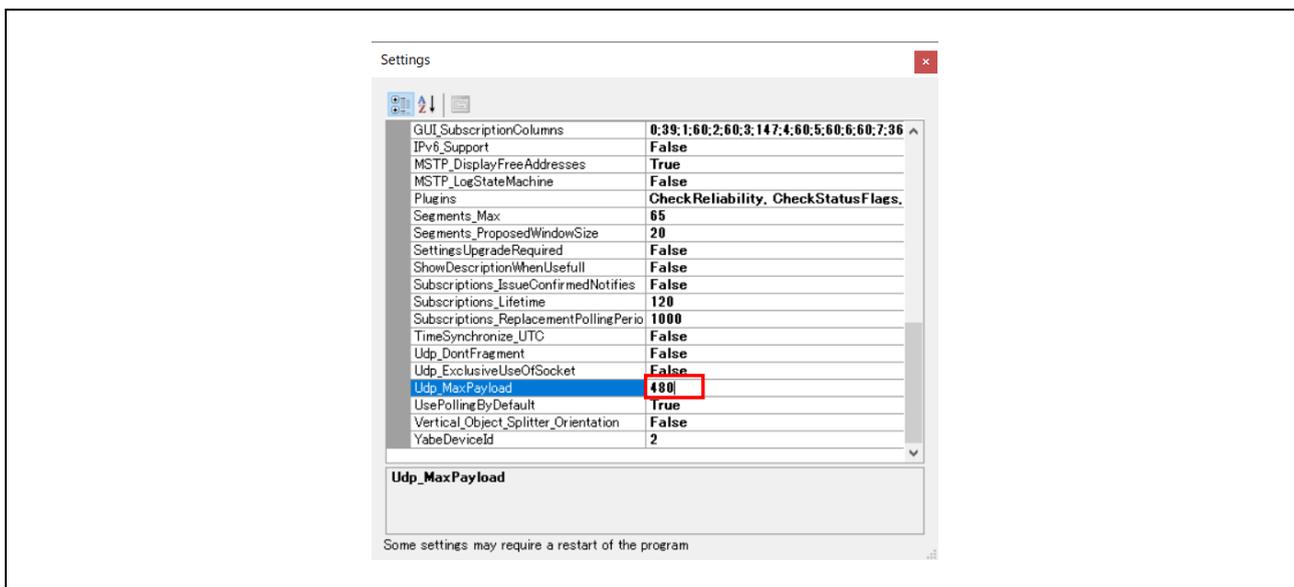


Fig.4-119 Udp_MaxPayload setting of Yabe

Address Space ウィンドウの FILE 0(File:0)を選択し、右クリックで Download File を選択します。

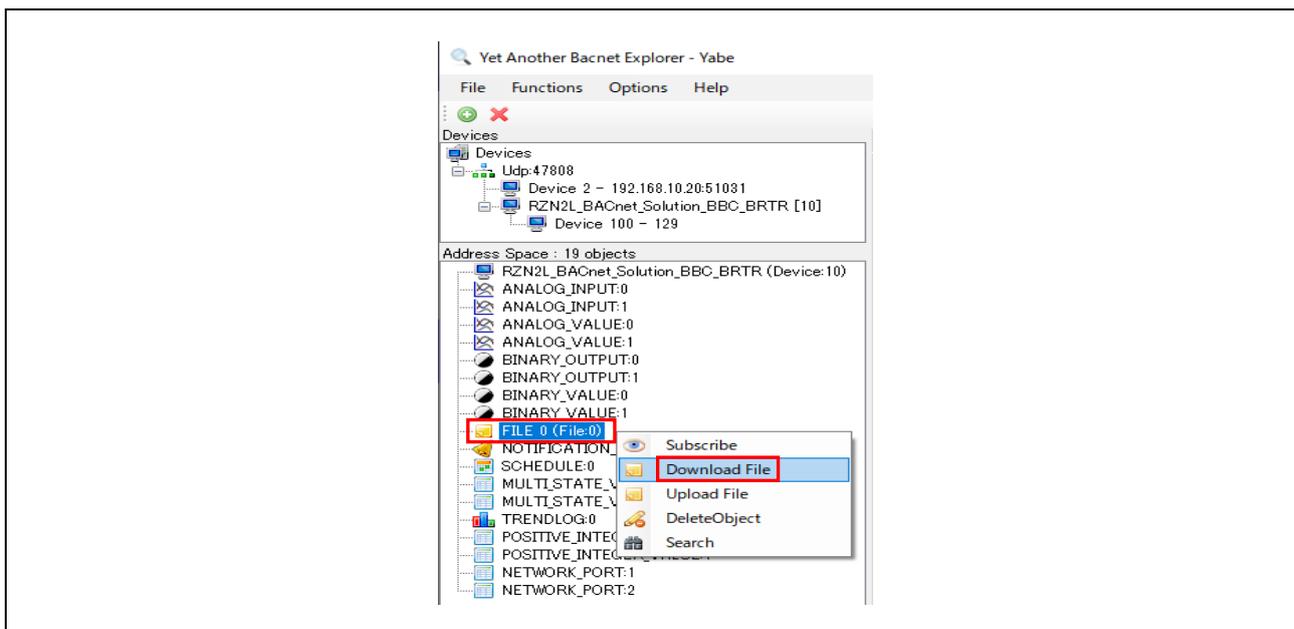


Fig.4-120 Select Download File

次のダイアログでファイル名を付けて保存します。Done ポップアップ画面の OK をクリックします。

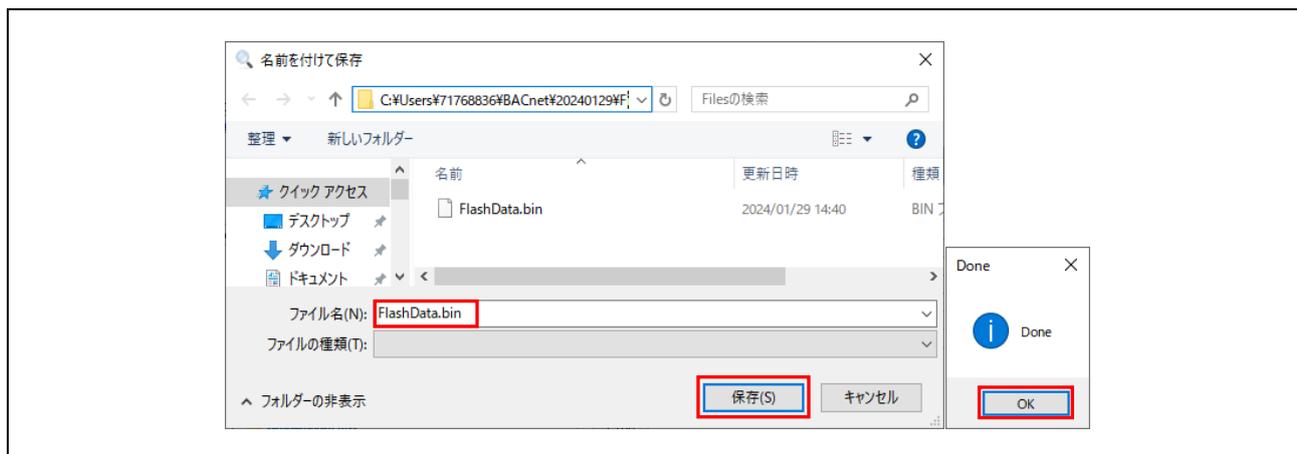


Fig.4-121 Save file with file name

4.6.5 AtomicWriteFile

VTS からの AtomicWriteFile を説明した 4.5.12 章も参照ください。Yabe からの設定手順について説明します。

Yabe を使用すると VTS のように Hex データストリーム(コンテンツ)を意識することなく PC 上からファイルを選択できます。選択されるファイルは AtomicReadFile で保存したファイルのことを指します。

Address Space ウィンドウの FILE 0(File:0)を選択し、右クリックで Upload File を選択します。

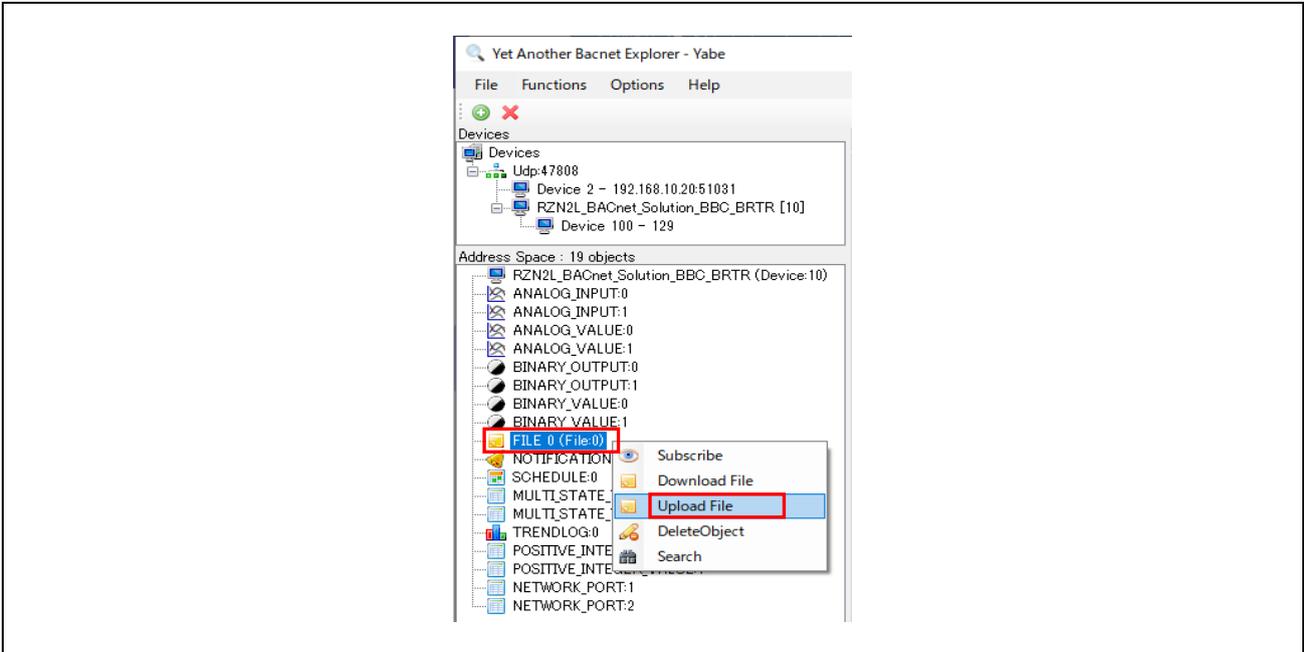


Fig.4-122 Select Upload File

次のダイアログでファイルを選択して開きます。Done ポップアップ画面の OK をクリックします。

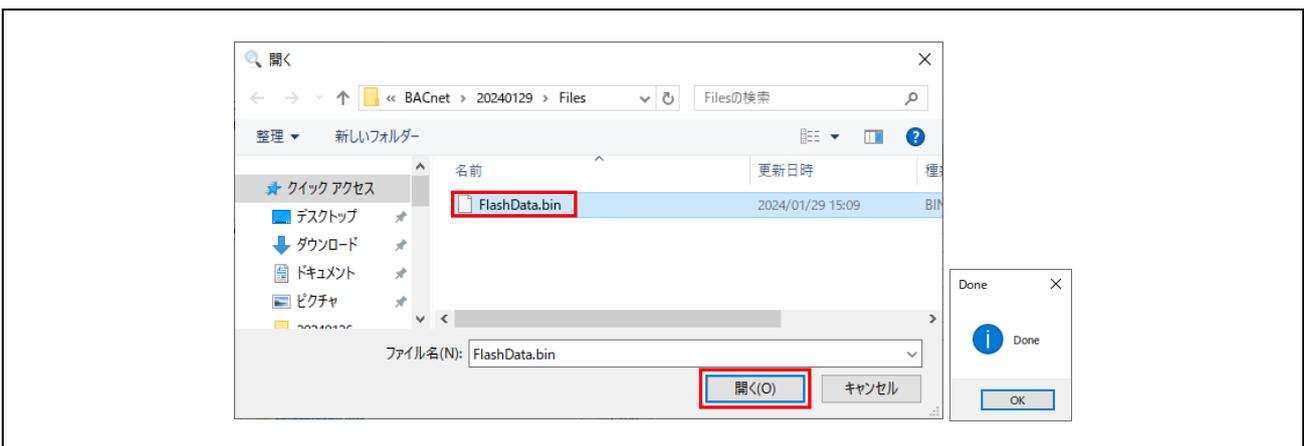


Fig.4-123 Open file

4.6.6 ReinitializeDevice

注意) ReinitializeDevice サービスはターゲットデバイスをリセットします。デバッグ接続状態の場合は RSK ボードの S3 RESET プッシュスイッチ(赤)を押してデバッグ接続を解除してください。デバッグ接続状態のまま、このサービスを実行すると B-BC がリポートした後 Ethernet 通信が確立しません。

VTS からの ReinitializeDevice を 4.5.9 章で説明しましたが Yabe からの設定手順について説明します。

制限事項) ReinitializeDevice の以下の State パラメータを B-BC は未サポートです。

STARTBACKUP, ENDBACKUP, STARTRESTORE, ENDRESTORE, ABORTRESTORE

Yabe を使用すると VTS から選択できなかったサービスパラメータ ACTIVATE_CHANGES を選択できます。Password に"filister"を入力して OK をクリックします。OK ポップアップの OK をクリックします。

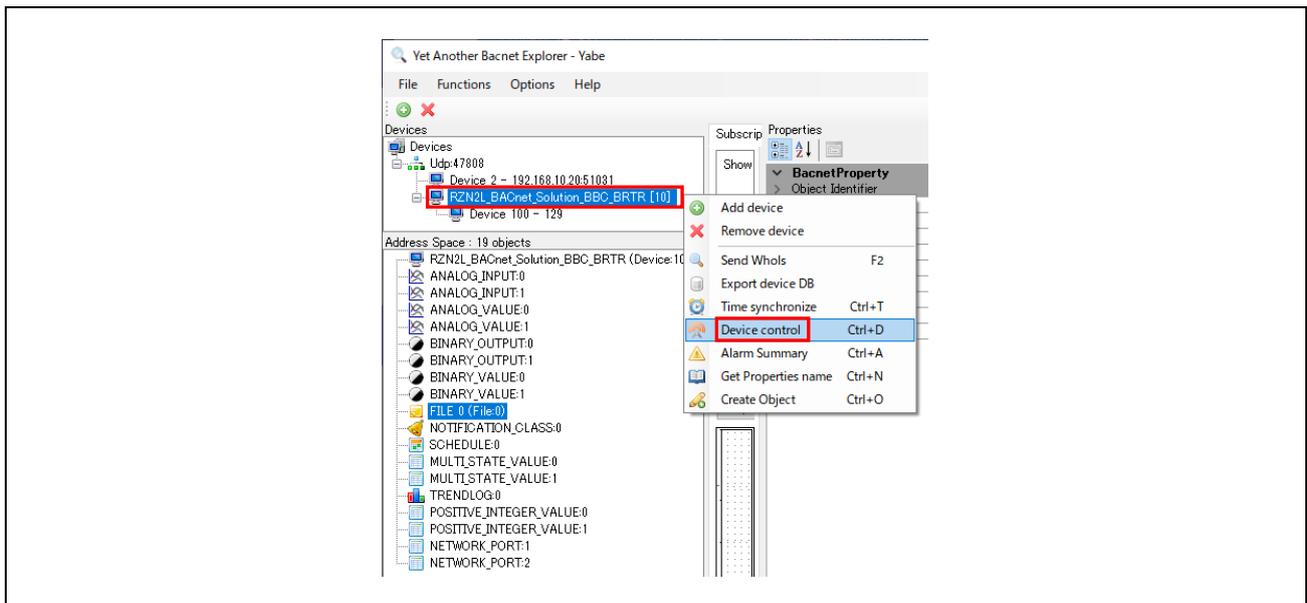


Fig.4-124 Select Device control

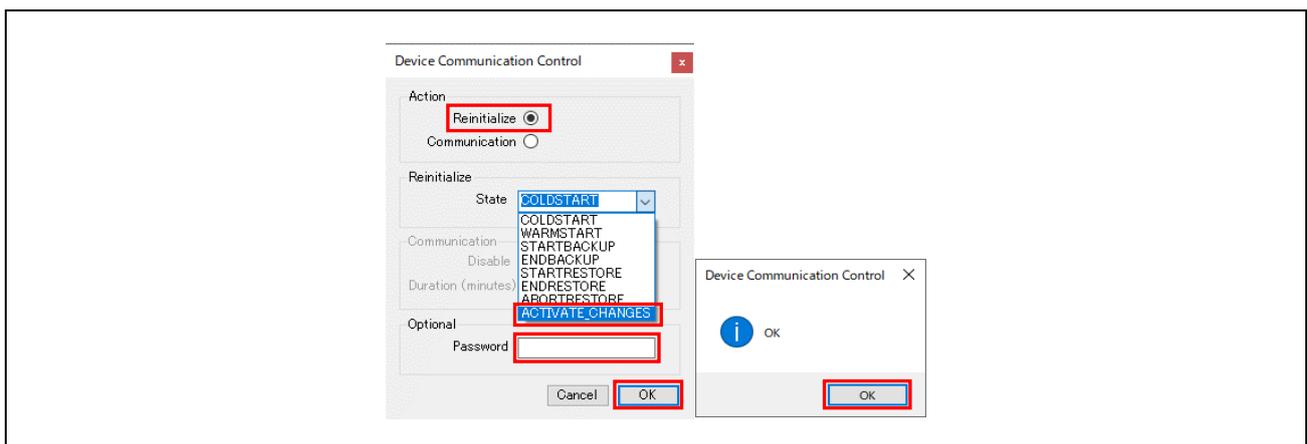


Fig.4-125 Select ACTIVATE_CHANGES

5. 初期設定

本章では、各オブジェクトのプロパティ初期値など、本サンプルソフトにおける各パラメータ設定について説明します。

5.1 初期値

本サンプルソフトの各オブジェクトにおけるプロパティやパラメータの初期値を示します。

各初期値の変更方法については表の Reference 欄に示されたリンクに記載してあります。

5.1.1 Ethernet MAC address(IP)

イーサネットの MAC アドレスは原則として全てのネットワーク機器に一意に割り振られるもので、5.2.1 章を参考に RSK ボード毎に個別に設定してください。

特に同一 BACnet/IP ネットワーク上に複数の RSK ボードを接続する場合は必ず変更してください。

Table 5-1 Ethernet MAC address

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

5.1.2 Device

デバイスオブジェクトのプロパティ初期値を示します。

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 TO-FAULT TO-NORMAL	Date: any Time: any Date: any Time: any 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)		
7		out-of-service	FALSE		5.2.16 OutOfService
8		units	No Units (95)		

No.	Object	Property	Initial value	Reference
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
	fault = FALSE			

No.	Object	Property	Initial value		Reference
				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	
5		status-flags	(Bit String) (FFFF) in-alarm = FALSE	

No.	Object	Property	Initial value	Reference
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'		
12		reliability	no-fault-detected (0)		
13		priority-array[1]	NULL		

No.	Object	Property	Initial value	Reference
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		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 fault = FALSE overridden = FALSE out-of-service = FALSE		
6					
7					
8					
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	binary-value,1	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)		

No.	Object	Property	Initial value	Reference
			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	NOTIFICATION CLASS 0			
3		object-type	notification-class (15)			
4		notification-class	(Unsigned) 0			
5		priority	To Off normal	(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'			

No.	Object	Property	Initial value	Reference
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 Time	January 1, 2009, (Day of Week = Thursday) 00:00:00.0	
15		stop-time	Date Time	December 22, 2020, (Day of Week = Tuesday) 23:59:59.99	
16		log-device-object-property	ObjectIdentifier: Property Identifier DeviceIdentifier	analog-input, 0 present-value (85) 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

ネットワークポートオブジェクトのプロパティ初期値を示します。

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

B-BC は ReinitializeDevice サービスまたは DeviceCommunicationControl サービスを受信すると BACnet ユーザーから送られたパスワードを照合して、一致した場合のみサービスを実行します。BACnet スタックに実装されたパスワードを初期値としていますので変更するには 5.2.15 章を参照してください。

Table 5-21 Password

Service	Initial value	Reference
ReinitializeDevice	filister	5.2.15 Password
DeviceCommunicationControl	filister	

5.2 初期値変更

本サンプルソフトの各プロパティ初期値の変更方法および補足事項について説明します。

初期値を変更する手段として、基本的にコードを変更してリビルドします。データによっては、ターミナルソフトウェアから 5.3 初期設定コマンドを使って変更できるものがあります。この場合の変更はボードリセット後から有効になります。コマンドにより変更可能なデータは次のとおりです。

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)

Fig. 5-1 に示す CN14 に貼り付けた MAC アドレスを設定します。



Fig. 5-1 MAC address pasted on CN14

(1) リビルドする場合

Fig. 5-2 に示すツリーから configuration.xml をダブルクリックしてスマート・コンフィグレータを開きます。

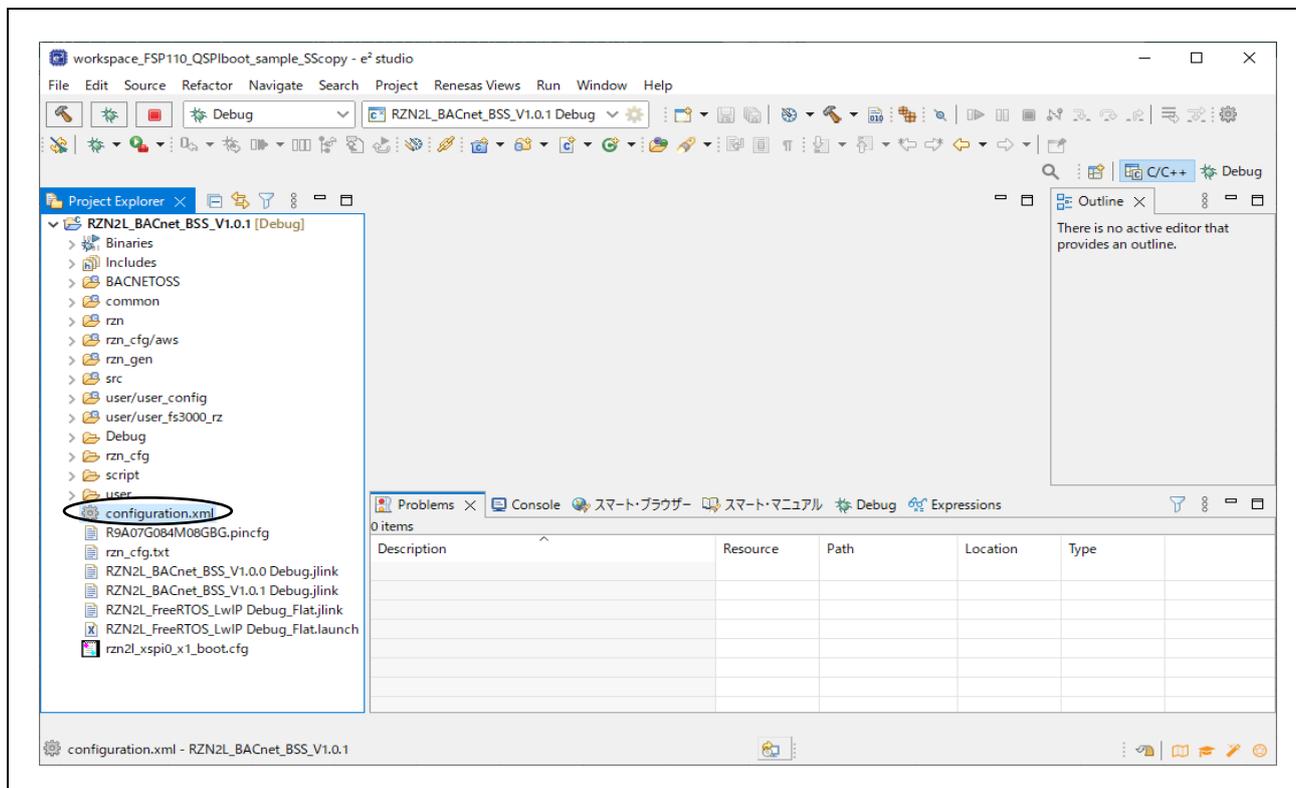


Fig. 5-2 Double click configuration.xml

Stacks タブを開き、g_ether0 Ethernet Driver on r_ether をクリックして選択します。

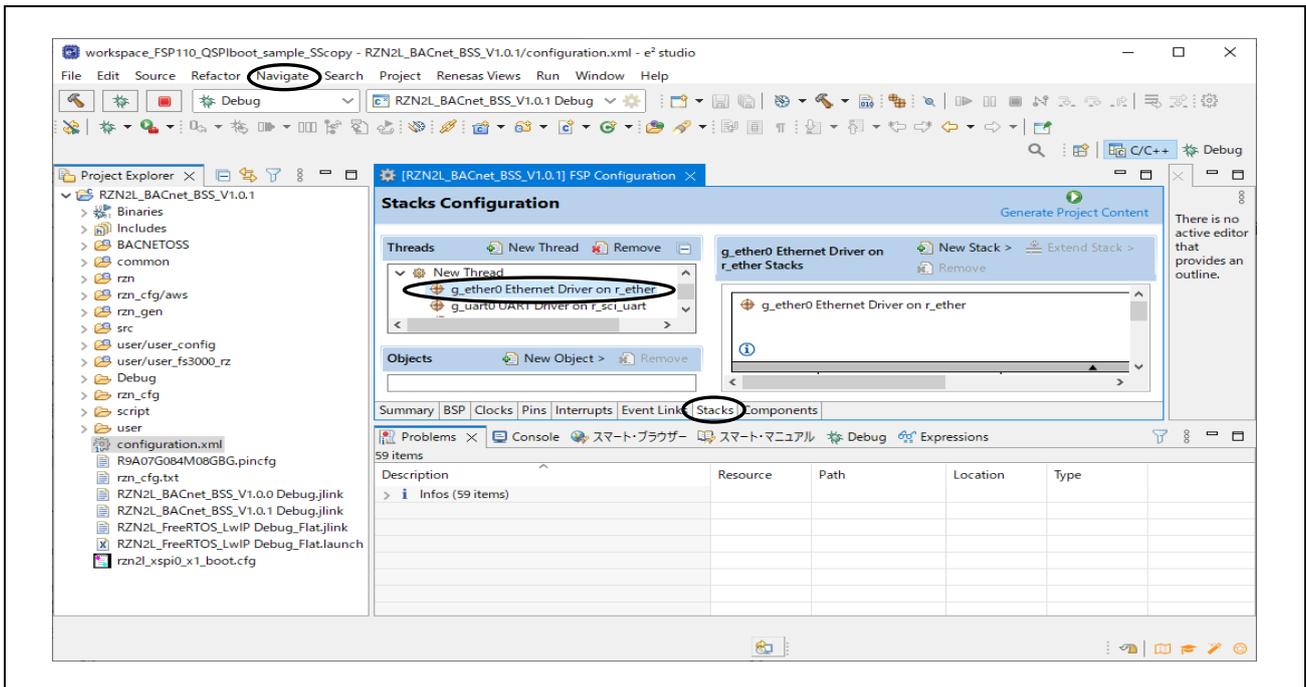


Fig. 5-3 Click Navigate

次に Navigate メニューを開き、Show In>Properties をクリックします。

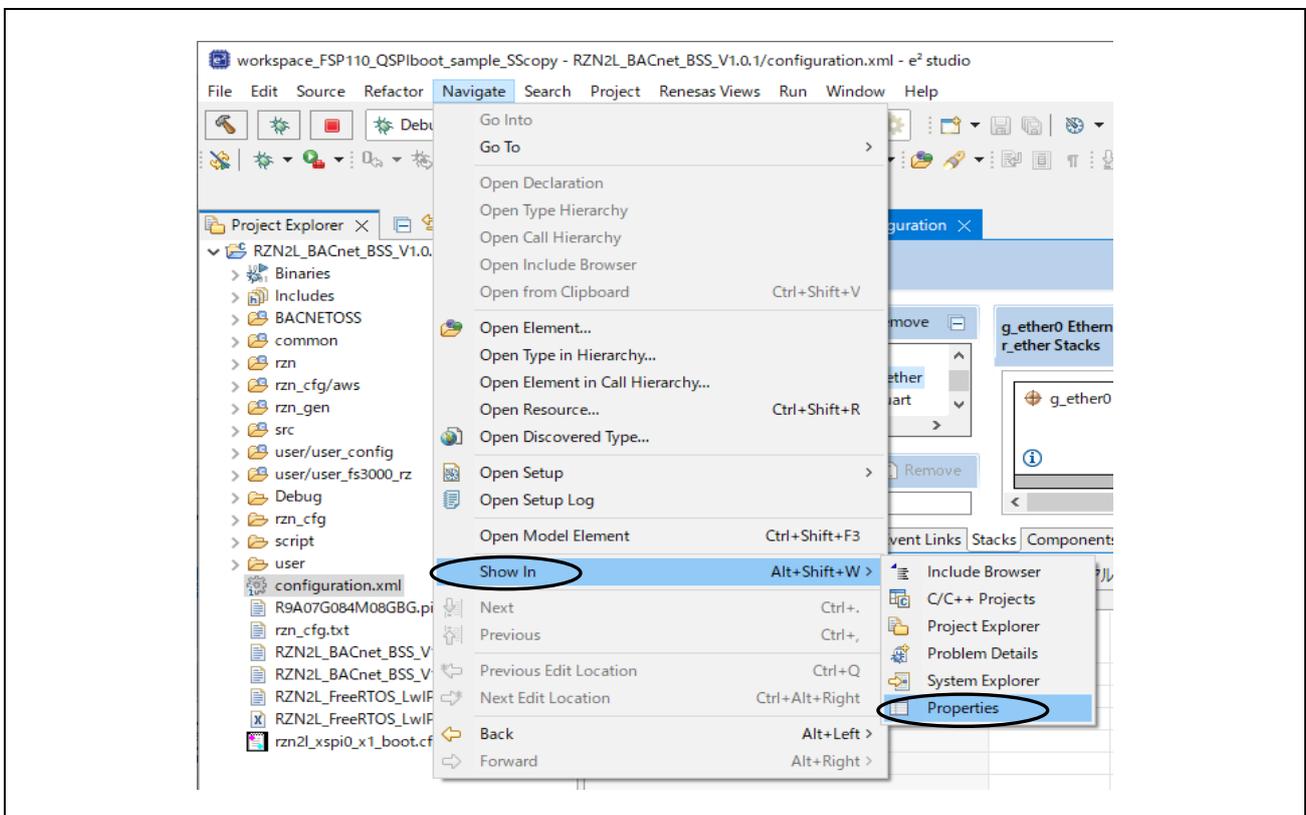


Fig. 5-4 Click Properties

Properties タブを開き、General>MAC address(例 : 74:90:50:10:05:B0)を入力します。
入力後 Generate Project Content をクリックします。最後にリビルドします。
ビルド手順については、4.4.2 章を参照ください。

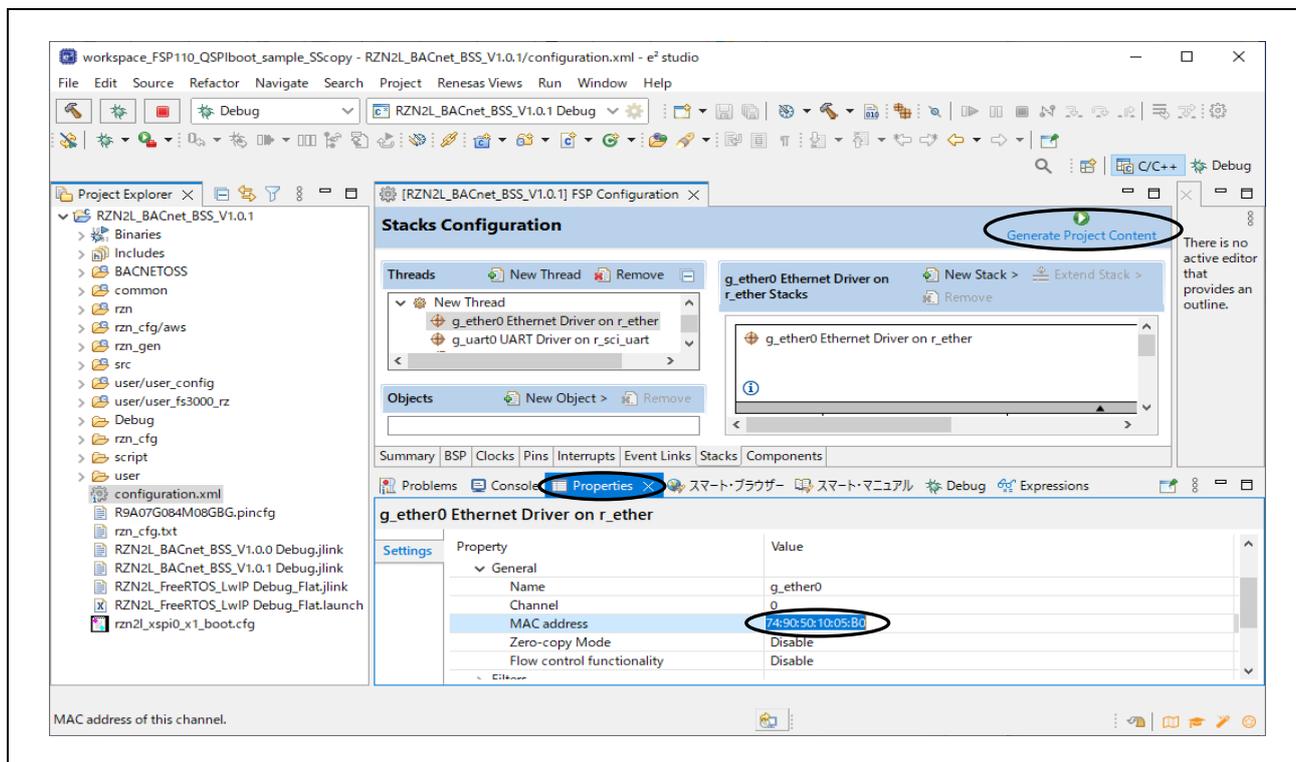


Fig. 5-5 Enter MAC address

(2) コマンドにより設定する場合

5.3 初期設定コマンド を参照ください。ビルド設定値よりも、コマンド設定値が優先されます。

5.2.2 Device instance

インスタンス番号は他のオブジェクト同様にオブジェクトタイプと組み合わせて Object Identifier プロパティを形成します。

Device オブジェクトタイプのインスタンス番号だけは BACnet のインターネットワーク全体でユニークでなければなりません。

Device インスタンスの設定範囲は 0~4194303 です。ただし、4194303 は無効を意味し、使われていないことを表します。

5.3 初期設定コマンド を参照ください。

5.2.3 Device name

5.3 初期設定コマンド を参照ください。

5.2.4 Number of objects

次の Symbol の Value 設定値がオブジェクト数の初期値を表しています。この Value を変更します。

ただし、#BACNET_NETWORK_PORTS_MAX のオブジェクト数 2 は変更しないでください。

```
#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
```

変更手順は次のとおりです。

Project Explorer ウィンドウのプロジェクト名を選択したうえで、Project メニューの Properties を開きます。変更は “Edit...” をクリックすると可能になります。

変更後、Apply and Close をクリックし、設定を適用します。ポップアップダイアログの Yes をクリックします。最後にリビルドします。

ビルド手順については、4.4.2 章を参照ください。

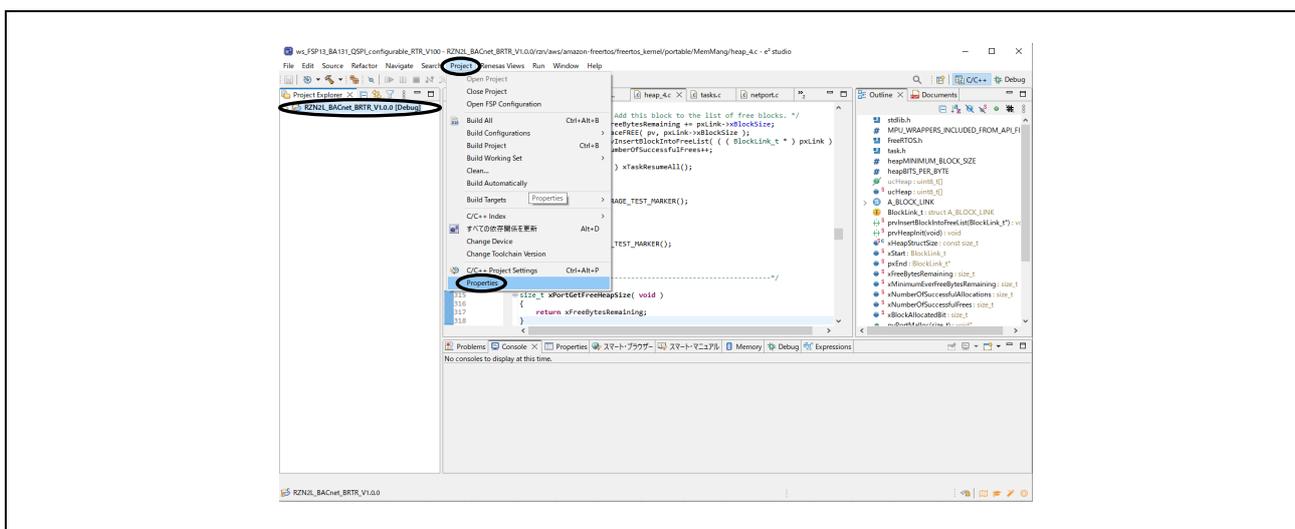


Fig. 5-6 Open Properties

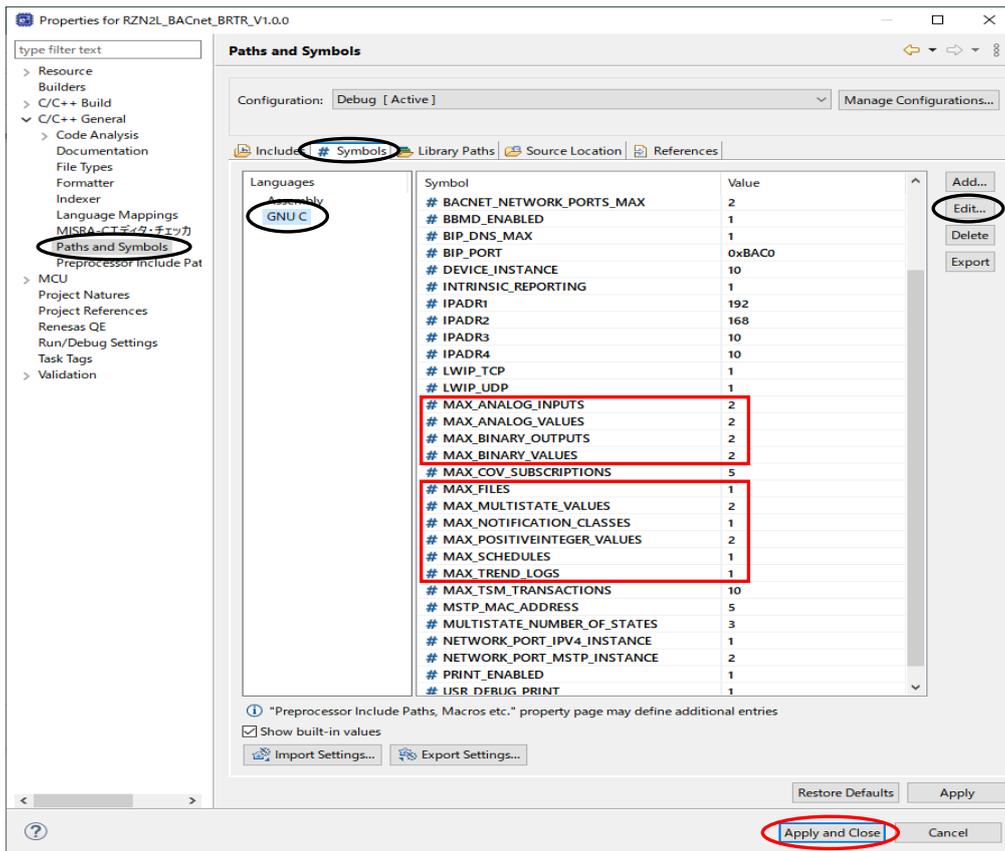


Fig. 5-7 Change Number of objects

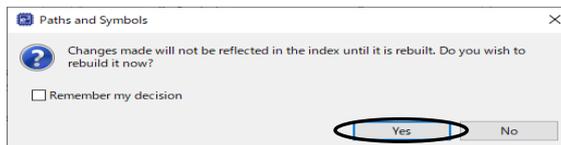


Fig. 5-8 Click Yes

5.2.5 UTC_Offset

5.3 初期設定コマンド を参照ください。

5.2.6 Number of states

これは Multi State Value オブジェクトの Present value が表す状態数を示し、1~254 の範囲で変更可能です。この Number of states を変更すると同時に State text 数も増減させる必要があります。State text の変更は 5.2.7 に記載しています。

Number of states、Present value、State text の関係を Table 5-22 に示します。

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

Number of states は次の Symbol の Value を変更します。変更手順は 5.2.4 章と同様です。

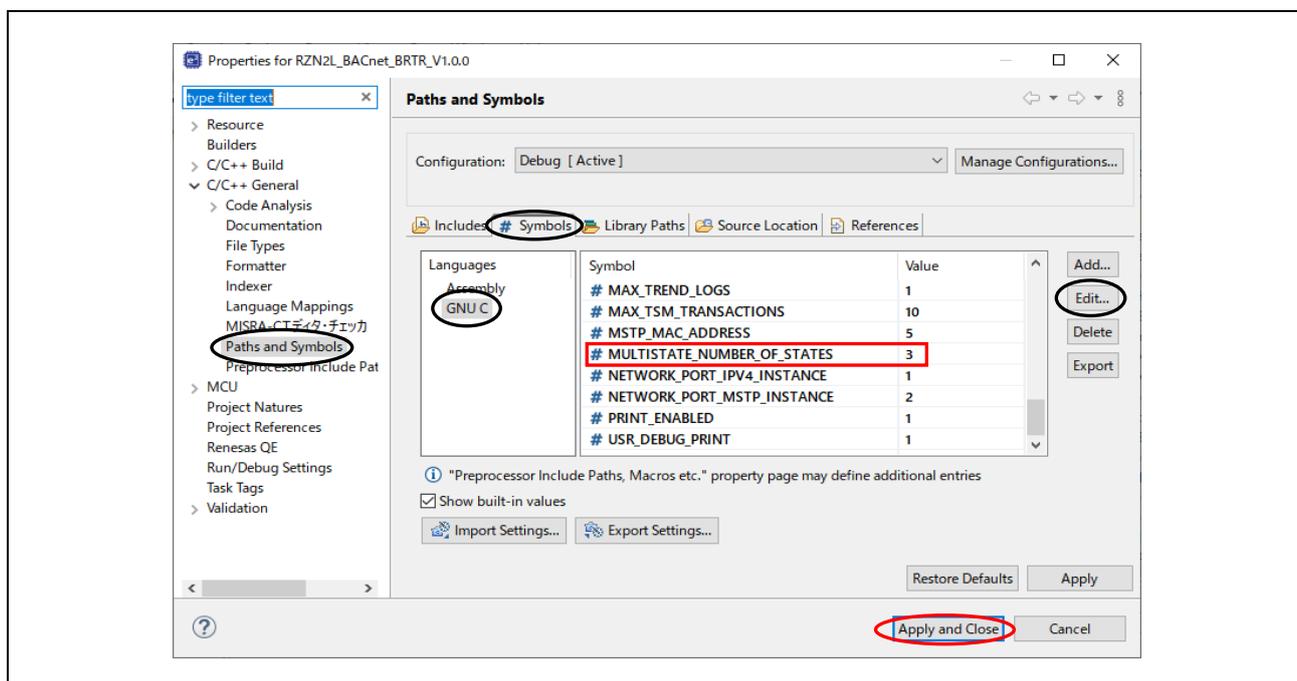


Fig. 5-9 Change Number of states

5.2.7 State text

State text は present value が表す大・中・小など、状態の説明文字列です。説明文字列は 64 バイト以下です。Number of states の初期値は 3 のため説明も 3 つとしています。Number of states の値を変更した場合は State text の説明も増減させてください。

コード上の設定箇所を” state_name[MULTISTATE_NUMBER_OF_STATES][64]”で検索して、変更してください。

5.2.8 Network number

Network number は、MS/TP 固有のプロパティで、ネットワークに関連付けられた BACnet ネットワーク番号を表します。このプロパティの範囲は 0~65534 ですが 0 は不明を意味します。

WriteProperty または WritePropertyMultiple サービスを実行すると、このプロパティ値を Flash メモリに書き込みます。5.3 初期設定コマンド を参照ください。

5.2.9 Link speed

Link speed は、1 秒あたりのビット数として表します。値 0 は、通信速度が不明であることを意味します。

MS/TP 接続時に有効であり、UART のボーレートを表します。

ボーレートの変更は Table 5-23 から選択してください。

Table 5-23 Baud rate

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

Link speed は次の手順で変更します。

スマート・コンフィグレータの g_uart5 UART Driver on r_sci_uart を選択した状態で Properties タブを開き、Baud>Baud Rate を入力します。入力後 Generate Project Content をクリックします。

スマート・コンフィグレータの起動は 5.2.1 章を参照ください。

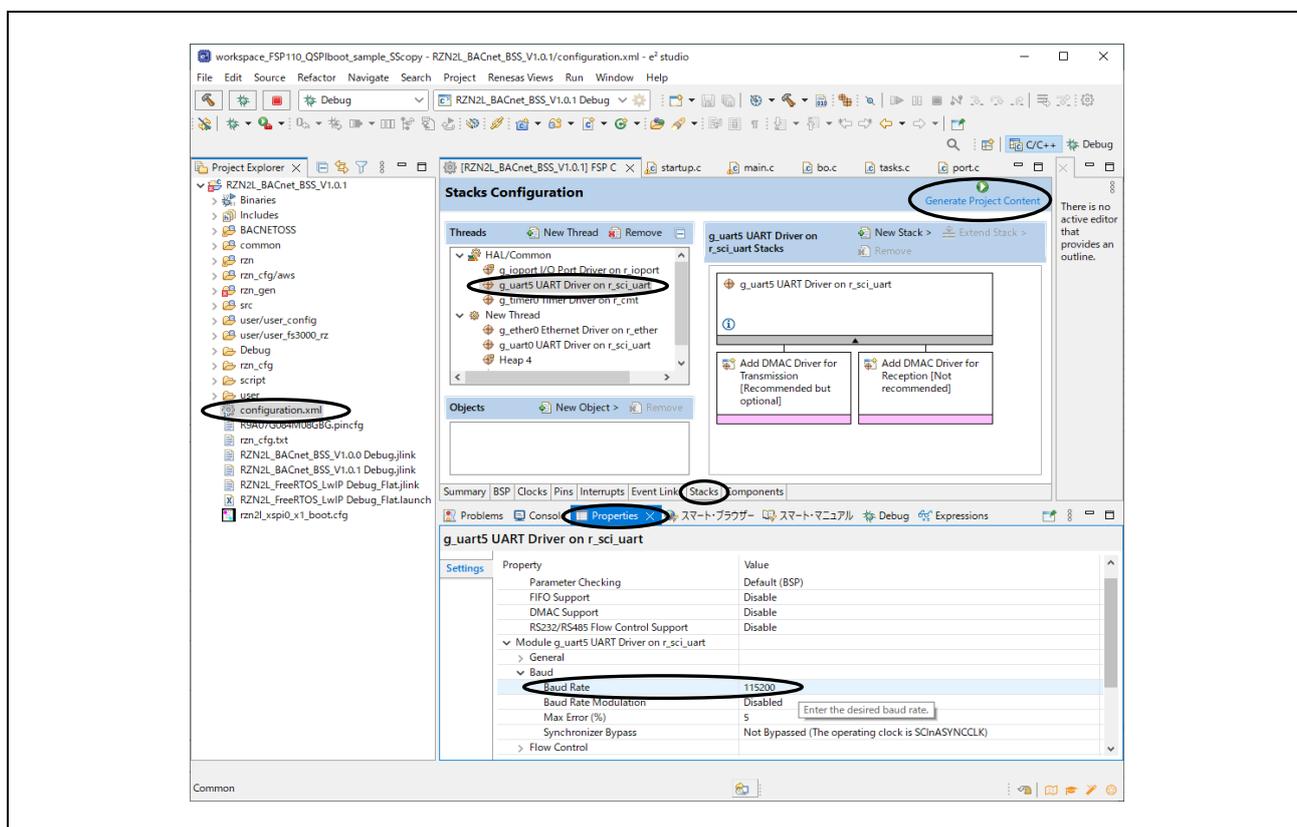


Fig. 5-10 Enter Baud Rate

さらにコード上の設定箇所を変更してください。

BACNETOSS\sample.h

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

Fig. 5-11 Change UART_BAUDRATE

最後にリビルドします。
ビルド手順については、4.4.2 章を参照ください。

5.2.10 MAC address

5.3 初期設定コマンド を参照ください。

5.2.11 BACnet IP address

5.3 初期設定コマンド を参照ください。

5.2.12 BACnet_IP_Mode

NetworkPort オブジェクトの BACnet_IP_Mode プロパティは BIP 固有です。

B-BC は NORMAL か FOREIGN をサポートし、BBMD をサポートしません。

NORMAL : デバイスは、このネットワーク ポート上で外部デバイスとしても BBMD としても動作しない。

FOREIGN: デバイスは、このネットワーク ポート上で外部デバイスとして動作する。

BBMD: デバイスは、このネットワーク ポート上で BBMD として動作する。

このプロパティの変更は、WriteProperty および WritePropertyMultiple サービスで行います。WriteProperty および WritePropertyMultiple サービスを実行すると設定値が Flash メモリに書き込まれ、Changes_Pending プロパティが TRUE に設定されます。リセット時および ReinitializeDevice サービス要求を ACTIVATE_CHANGES または WARMSTART で受信した場合にこの値が有効になります。

このプロパティに対しての初期設定コマンドはございません。詳細は、5.3 章を参照ください。

5.2.13 FD_BBMD_Address

Network Port オブジェクトの FD_BBMD_Address プロパティは、BIP 固有のプロパティで、BBMD デバイスの IP アドレスと UDP ポート番号です。BACnet_IP_Mode が FOREIGN の場合に、B-BC は自身を外部デバイスとして登録するために Register-Foreign-Device BVLL メッセージを BBMD デバイスに送信します。

このプロパティの変更は、WriteProperty および WritePropertyMultiple サービスで行います。WriteProperty および WritePropertyMultiple サービスを実行すると設定値が Flash メモリに書き込まれ、Changes_Pending プロパティが TRUE に設定されます。リセット時および ReinitializeDevice サービス要求を ACTIVATE_CHANGES または WARMSTART で受信した場合にこの値が有効になります。

プロパティ 変更手順は(1)を参照ください。

このプロパティに対しての初期設定コマンドはございません。詳細は、5.3 章を参照ください。

(1) VTS からの FD_BBMD_Address 設定

VTS から FD_BBMD_Address プロパティを設定する手順を示します。

WriteProperty ダイアログの ID... をクリックします。

Object ID ダイアログの以下を設定します。

Object Type に Reserved を選択します。

Reserved Type に 56(OBJECT_NETWORK_PORT)を設定します。()内 : bacenum.h 定義値

Instance に 1 を設定します。OK をクリックします。

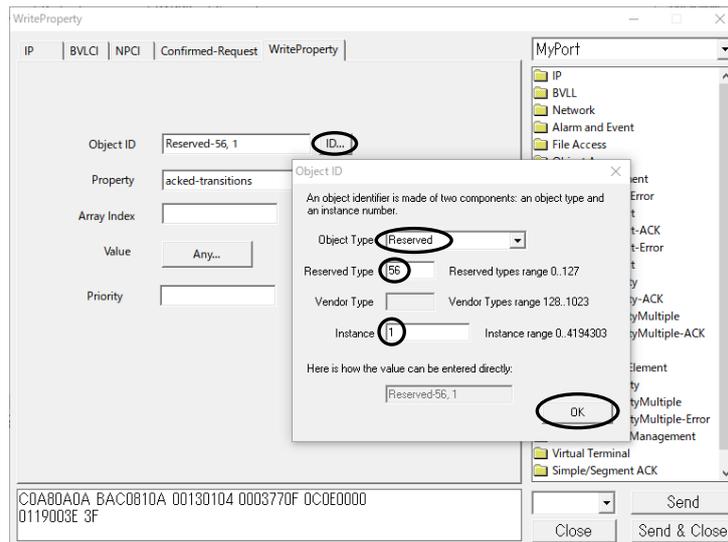


Fig. 5-12 WriteProperty dialog(1)

WriteProperty ダイアログの Property から "< Enter numeric value >" を選択します。

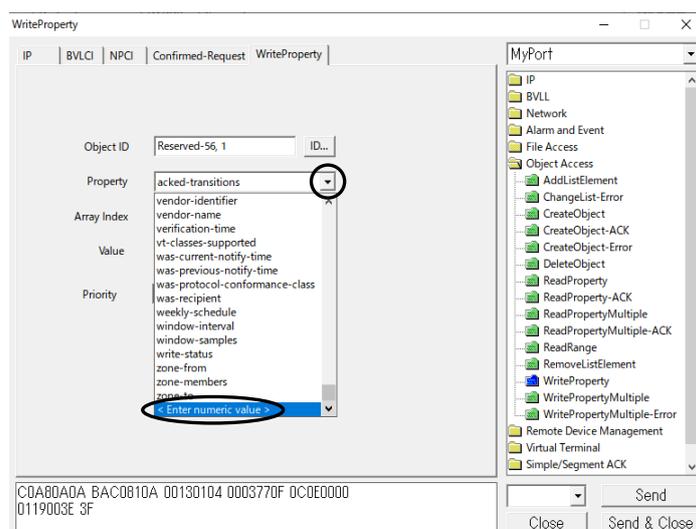


Fig. 5-13 WriteProperty dialog(2)

PropertyIdentifier ダイアログに 418(PROP_FD_BBMD_ADDRESS)を設定して OK をクリックします。

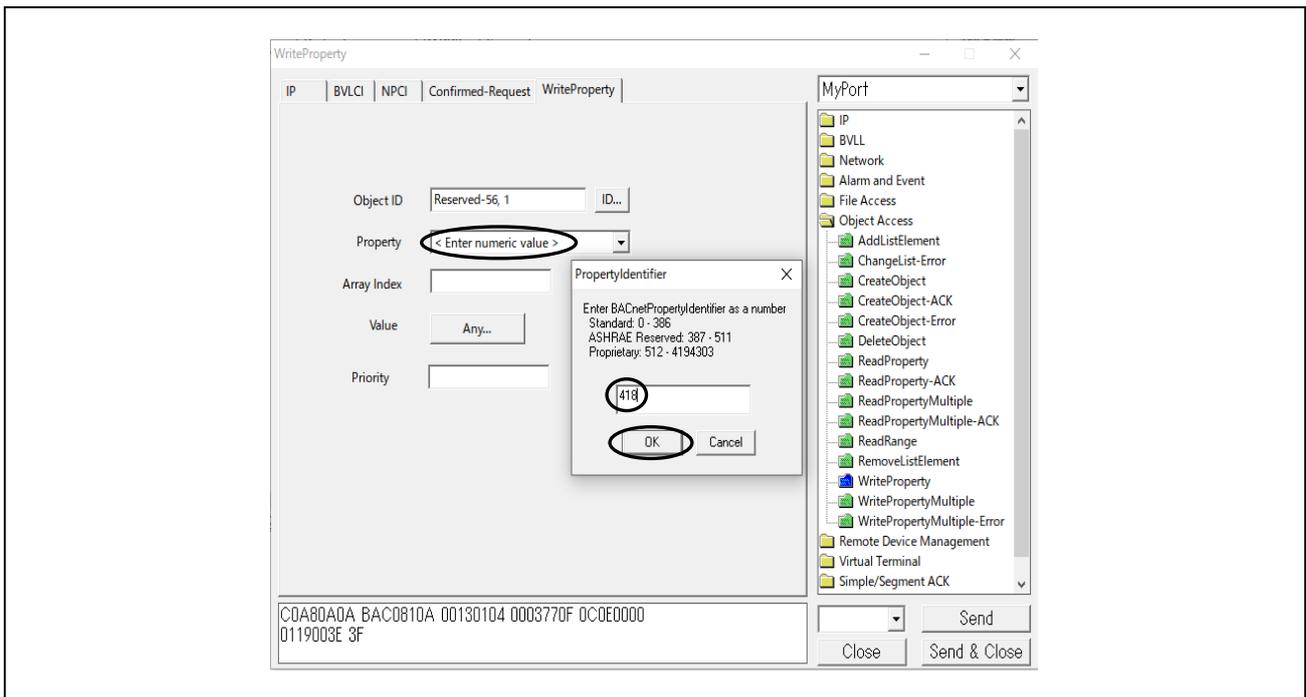


Fig. 5-14 WriteProperty dialog(2)

PropertyIdentifier ダイアログに 418(PROP_FD_BBMD_ADDRESS)を設定して OK をクリック後、Value の "Any..." をクリックします。

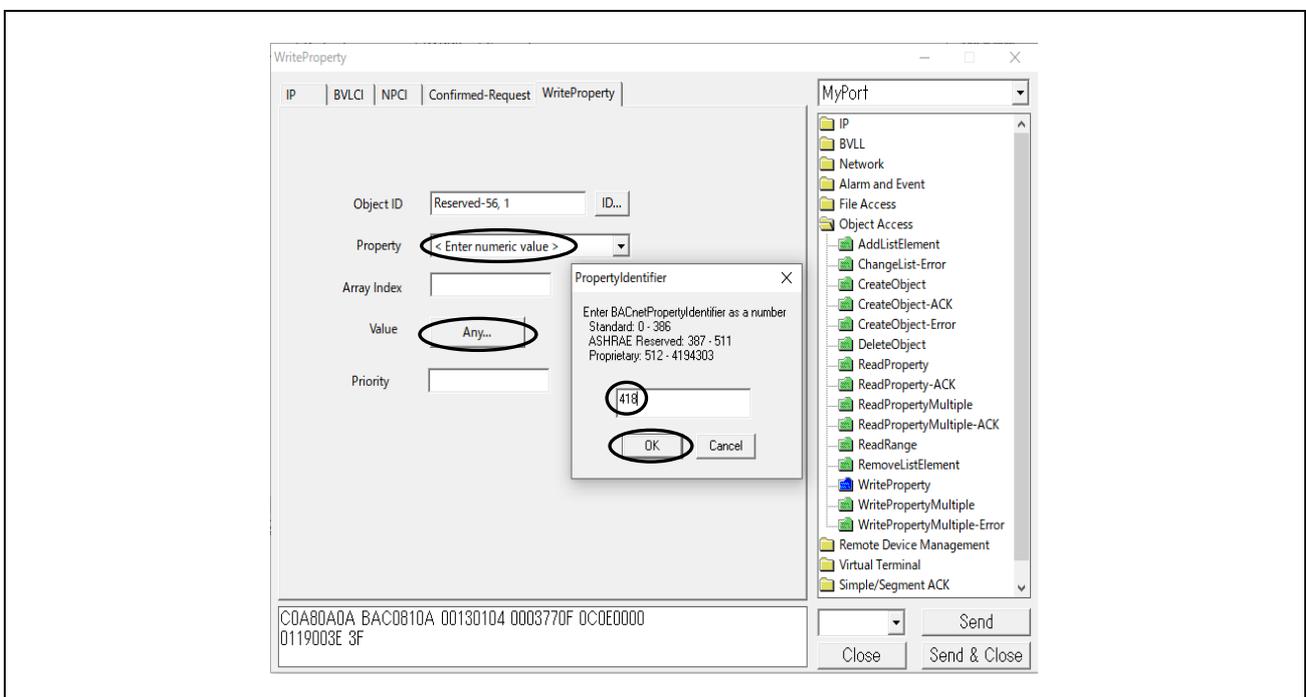


Fig. 5-15 WriteProperty dialog(2)

ABSTRACT-SYNTAX.&Type ダイアログの Add をクリックし、Type に Opening Tag を選択、Context に 0 を設定します。まだ OK をクリックしません。

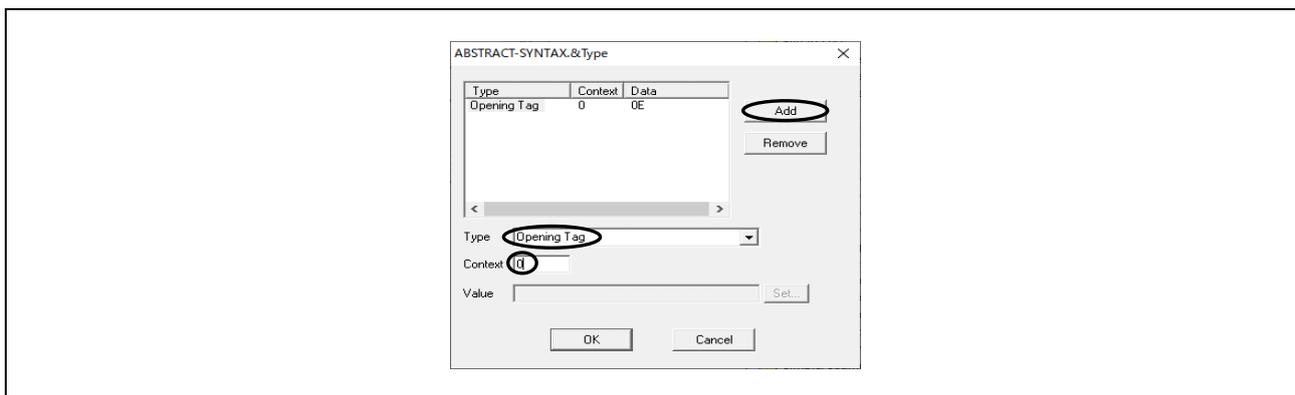


Fig. 5-16 ABSTRACT-SYNTAX.&Type dialog(1)

再び ABSTRACT-SYNTAX.&Type ダイアログの Add をクリックし、Type に Octet String を選択、Context に 1 を設定します。Value に IP アドレス(例では接続 PC の 192.168.10.20 を Hex で C0A80A14 入力)を設定します。まだ OK をクリックしません。

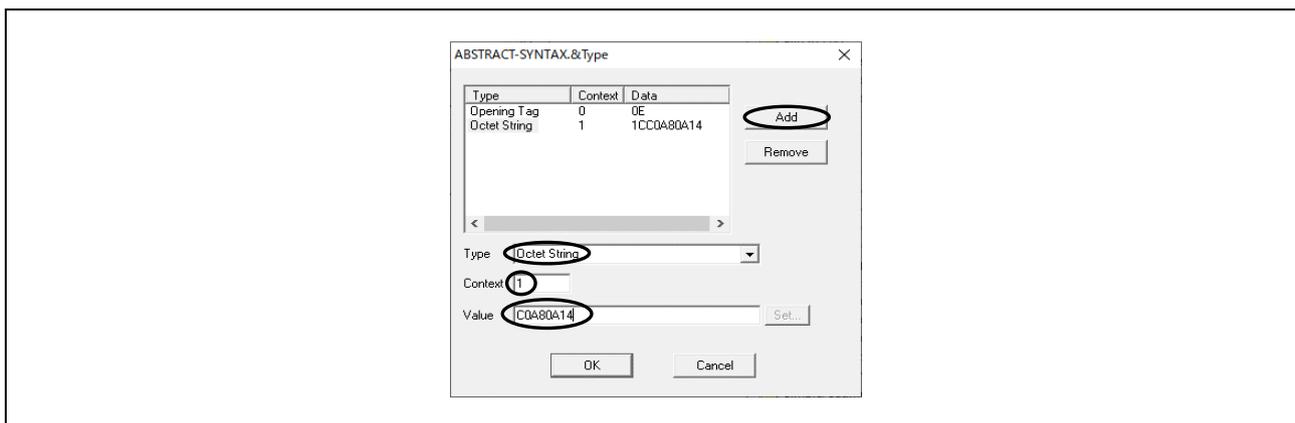


Fig. 5-17 ABSTRACT-SYNTAX.&Type dialog(2)

再び ABSTRACT-SYNTAX.&Type ダイアログの Add をクリックし、Type に Closing Tag を選択、Context に 0 を設定します。まだ OK をクリックしません。

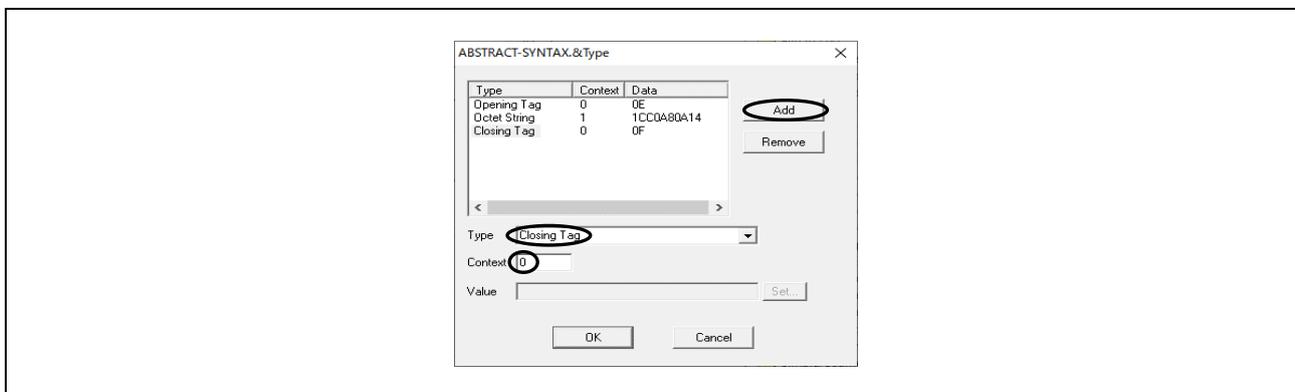


Fig. 5-18 ABSTRACT-SYNTAX.&Type dialog(3)

再び ABSTRACT-SYNTAX.&Type ダイアログの Add をクリックし、Type に Unsigned を選択、Context に 1 を設定します。Value に UDP ポート番号(例では 47808 の十進値)を設定後、OK をクリックします。

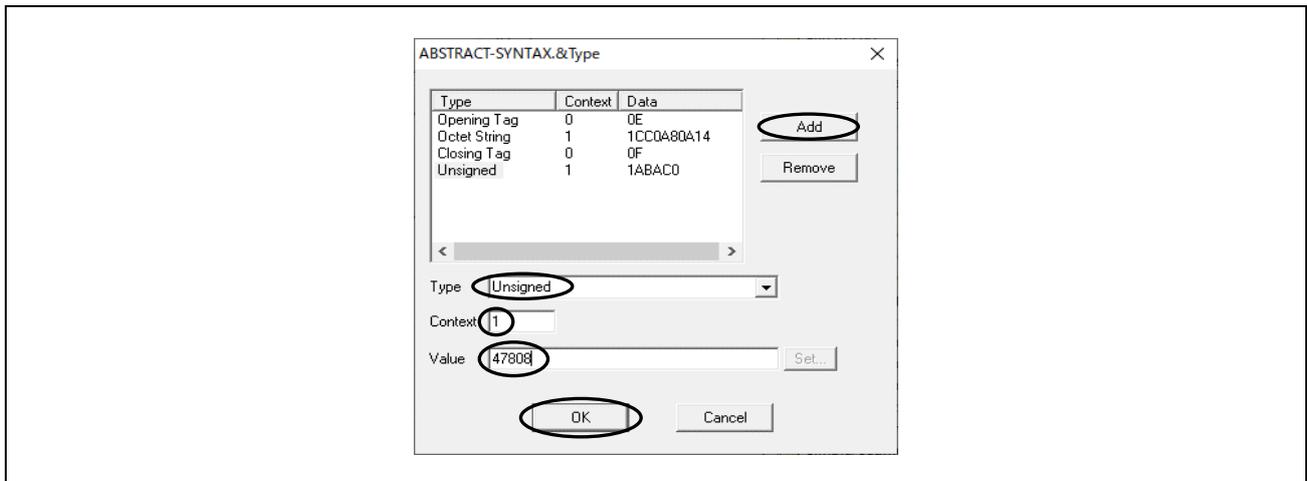


Fig. 5-19 ABSTRACT-SYNTAX.&Type dialog(4)

最後に WriteProperty ダイアログの Send をクリックします。

5.2.14 FD_Subscription_Lifetime

Network Port オブジェクトの FD_Subscription_Lifetime プロパティは、BIP 固有のプロパティで、Register-Foreign-Device BVLL メッセージで使用される Time-To-Live 値を秒単位で示します。この Time-To-Live 値が 0 になると、B-BC は Register-Foreign-Device BVLL メッセージを再送します。

このプロパティの変更は、WriteProperty および WritePropertyMultiple サービスで行います。WriteProperty および WritePropertyMultiple サービスを実行すると設定値が Flash メモリに書き込まれ、Changes_Pending プロパティが TRUE に設定されます。リセット時および ReinitializeDevice サービス要求を ACTIVATE_CHANGES または WARMSTART で受信した場合にこの値が有効になります。

このプロパティに対しての初期設定コマンドはございません。詳細は、5.3 章を参照ください。

5.2.15 Password

Password は Table 5-24 に示すコード上の設定箇所を Search word で検索して変更してください。

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

WriteProperty または WritePropertyMultiple サービスを実行すると、このプロパティ値を Flash メモリに書き込みます。5.3 初期設定コマンド を参照ください。

5.3 初期設定コマンド

BACnet では電源障害などによりデバイスの再起動が起こっても変更された値を維持する必要がある Property があります。本サンプルソフトでは、Table 5-25 から Table 5-27 に示す Configurable Property を Flash メモリ(QSPI0 Flash ROM)に保存して B-BC デバイスの再起動後に値を維持します。

これらの Configurable Property は、シリアルインタフェースにより初期設定コマンド実行して設定することができます。SCI0 周辺モジュールが割り当てられた CN16 端子と PC を USB ケーブルで接続し、ターミナルソフトから初期設定コマンドを実行します。

(1) BIP、MSTP 共通の Configurable Property

BIP、MSTP 共通の Configurable Property を Table 5-25 に示します。

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
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 固有の Configurable Property

BIP 固有の Configurable Property を Table 5-26 に示します。

No.1 の Ethernet_MAC は BACnet 規格のプロパティではありませんがリビルドする代わりに初期設定コマンドで変更可能です。

No.2 の MAC_Address はリビルドする代わりに初期設定コマンドで変更可能です。

No.3 と 4 は No.2 のコマンド実行結果がリセット後に反映されるため専用のコマンドはありません。

No.5,6,7 を WriteProperty または WritePropertyMultiple サービスを実行するときに Flash に書き込みます。そのため、専用のコマンドはなく、リセット後に各プロパティに反映されます。

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	"MAC_Address" setting value is reflected.	Bac_IP_mac_address =	192.168.10.10:47808	
3	IP_Address			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 固有の Configurable Property

MSTP 固有の Configurable Property を **Table 5-27** に示します。

No.1 の MAC_Address をリビルドする代わりに初期設定コマンドで変更可能です。

No.2 の Network_Number を WriteProperty または WritePropertyMultiple サービスを実行するときに Flash に書き込みます。そのため、専用のコマンドはなく、リセット後にプロパティに反映されます。

Table 5-27 MSTP-specific configurable properties

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

5.3.2 セットアップ

RSK ボードの CN16 と PC を USB ケーブルで接続します。

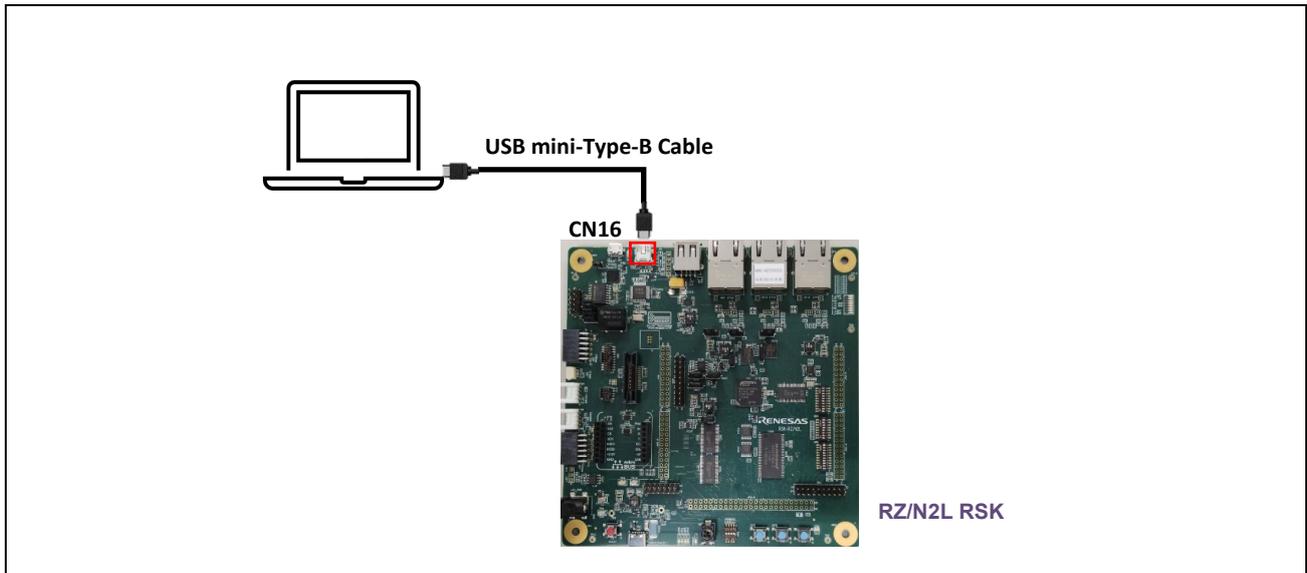


Fig. 5-20 Connect CN16 of the RSK to PC

PC 上のターミナルソフトウェアを起動します。

どのターミナルソフトでも問題ありませんが、ここでは例として TeraTerm を使用しています。次のリンク先から入手可能です。

[Releases · TeraTermProject/teraterm \(github.com\)](https://github.com/TeraTermProject/teraterm/releases)

シリアルポートのセットアップは次のとおりです。

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

ターミナルのセットアップは次のとおりです。

送信データに LF を付加、およびローカルエコーを無効にします。

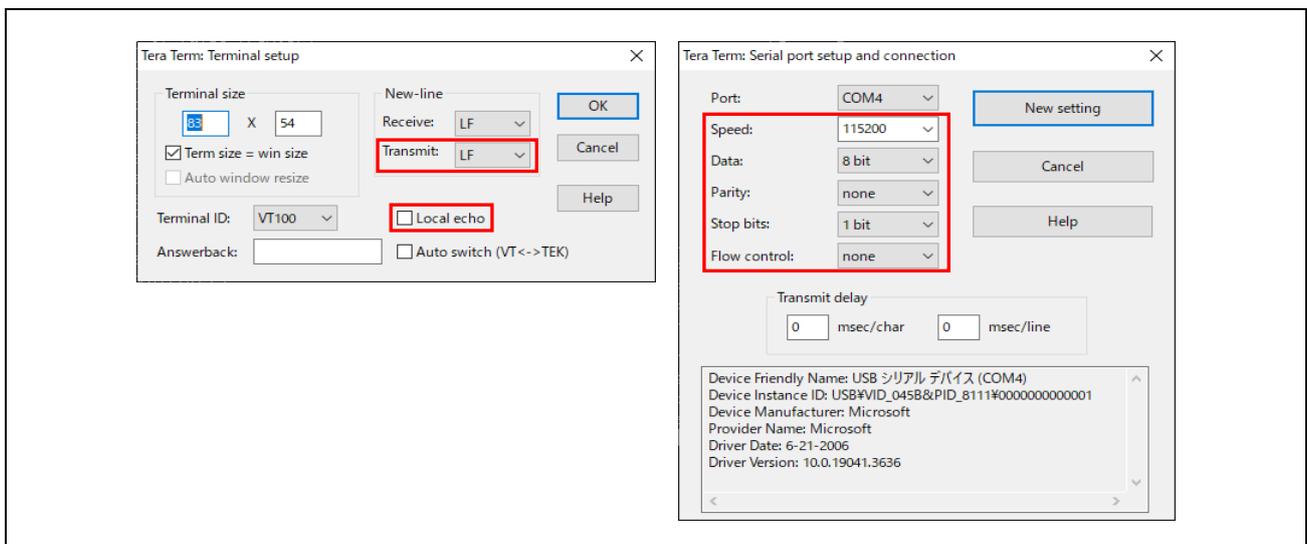
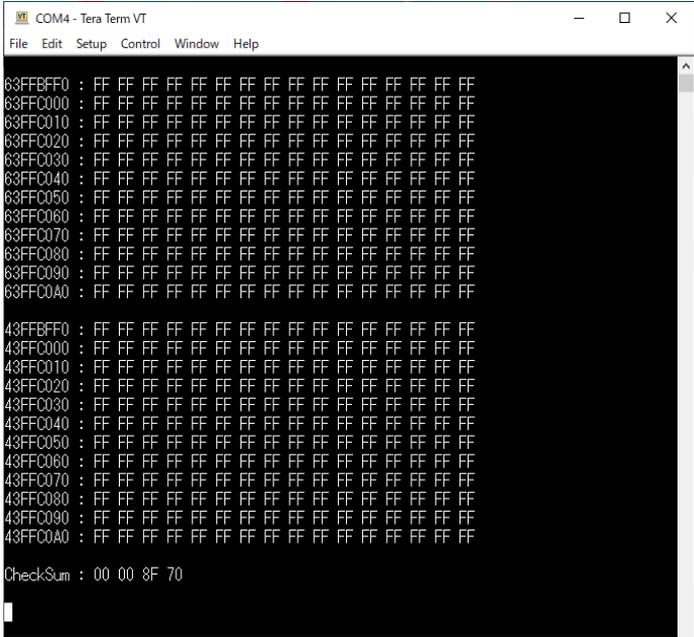


Fig. 5-21 Terminal software setup

5.3.3 コマンド実行方法

RSK ボードをリセットすると構成設定可能データ (Configurable Property) の格納先である QSPI0 フラッシュ後端領域のメモリダンプが行われます。



```
COM4 - Tera Term VT
File Edit Setup Control Window Help

63FFBFF0 : FF FF
63FFC000 : FF FF
63FFC010 : FF FF
63FFC020 : FF FF
63FFC030 : FF FF
63FFC040 : FF FF
63FFC050 : FF FF
63FFC060 : FF FF
63FFC070 : FF FF
63FFC080 : FF FF
63FFC090 : FF FF
63FFC0A0 : FF FF

43FFBFF0 : FF FF
43FFC000 : FF FF
43FFC010 : FF FF
43FFC020 : FF FF
43FFC030 : FF FF
43FFC040 : FF FF
43FFC050 : FF FF
43FFC060 : FF FF
43FFC070 : FF FF
43FFC080 : FF FF
43FFC090 : FF FF
43FFC0A0 : FF FF

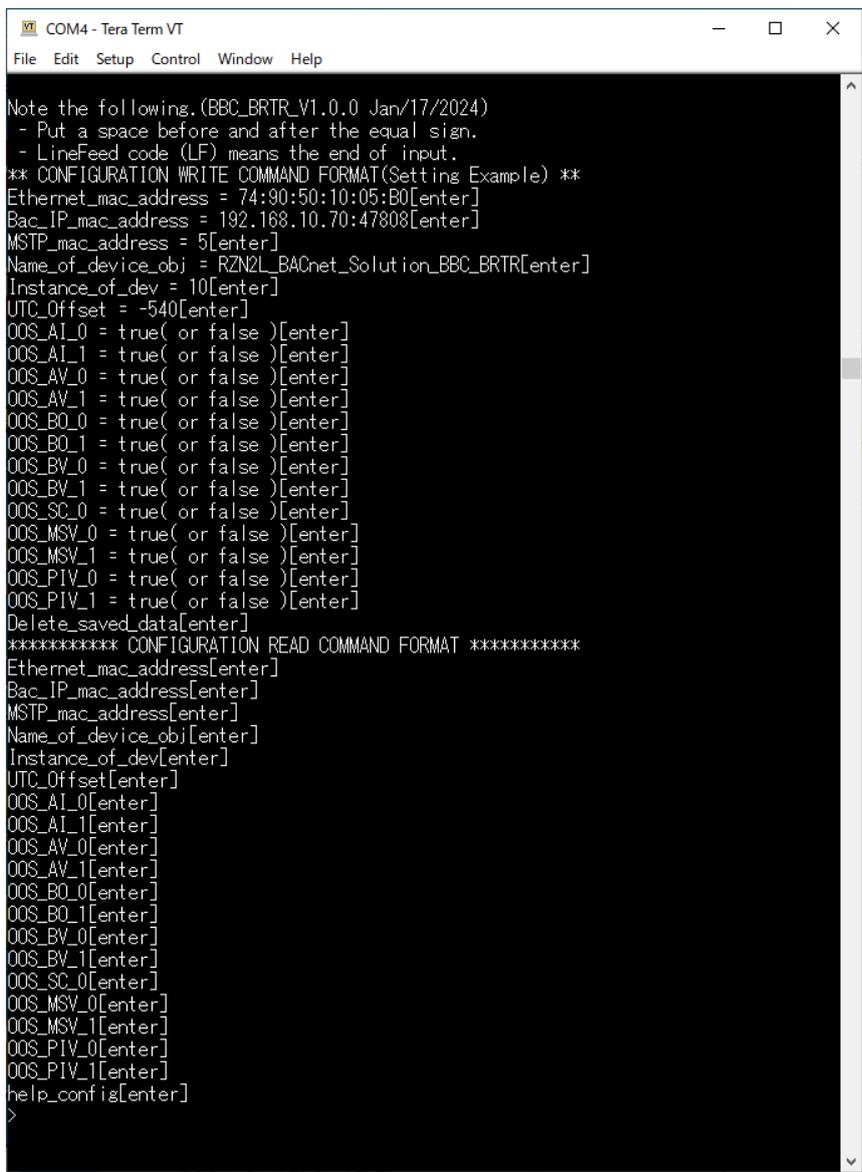
CheckSum : 00 00 8F 70
```

Fig. 5-22 Memory dump of data storage area

上段 : 0x63FFC000 から 0x63FFC093 : xSPI0_CS0_SPACE のデータ保存領域

下段 : 0x43FFC000 から 0x43FFC093 : xSPI0_CS0_SPACE_MIRROR のデータ保存領域

引き続き Enter など、どれかのキーを入力するとコマンド書式が表示されます。



The screenshot shows a terminal window titled "COM4 - Tera Term VT" with a menu bar (File, Edit, Setup, Control, Window, Help). The terminal content is as follows:

```
Note the following.(BBC_BRTR_V1.0.0 Jan/17/2024)
- Put a space before and after the equal sign.
- LineFeed code (LF) means the end of input.
** CONFIGURATION WRITE COMMAND FORMAT(Setting Example) **
Ethernet_mac_address = 74:90:50:10:05:B0[enter]
Bac_IP_mac_address = 192.168.10.70:47808[enter]
MSTP_mac_address = 5[enter]
Name_of_device_obj = RZN2L_BACnet_Solution_BBC_BRTR[enter]
Instance_of_dev = 10[enter]
UTC_Offset = -540[enter]
OOS_AI_0 = true( or false )[enter]
OOS_AI_1 = true( or false )[enter]
OOS_AV_0 = true( or false )[enter]
OOS_AV_1 = true( or false )[enter]
OOS_BO_0 = true( or false )[enter]
OOS_BO_1 = true( or false )[enter]
OOS_BV_0 = true( or false )[enter]
OOS_BV_1 = true( or false )[enter]
OOS_SC_0 = true( or false )[enter]
OOS_MSV_0 = true( or false )[enter]
OOS_MSV_1 = true( or false )[enter]
OOS_PIV_0 = true( or false )[enter]
OOS_PIV_1 = true( or false )[enter]
Delete_saved_data[enter]
***** CONFIGURATION READ COMMAND FORMAT *****
Ethernet_mac_address[enter]
Bac_IP_mac_address[enter]
MSTP_mac_address[enter]
Name_of_device_obj[enter]
Instance_of_dev[enter]
UTC_Offset[enter]
OOS_AI_0[enter]
OOS_AI_1[enter]
OOS_AV_0[enter]
OOS_AV_1[enter]
OOS_BO_0[enter]
OOS_BO_1[enter]
OOS_BV_0[enter]
OOS_BV_1[enter]
OOS_SC_0[enter]
OOS_MSV_0[enter]
OOS_MSV_1[enter]
OOS_PIV_0[enter]
OOS_PIV_1[enter]
help_config[enter]
>
```

Fig. 5-23 Command format

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

書き込みコマンド書式を表示します。設定値は例を示しています。(5.3.3.1 章)

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

読み出しコマンド書式を表示します。(5.3.3.2 章)

5.3.3.1 Write Command

次のように書き込みコマンドを実行します。コマンドは大文字/小文字も含めて正しく入力する必要があります。書き込み値は設定例を表し、ボードをリセット後に有効になります。

(1) BIP、MSTP 共通のコマンド

Name_of_device_obj = RZN2L_BACnet_Solution_BBC_BRTR

デバイスオブジェクトのオブジェクト名(すなわちデバイス名)です。

Instance_of_dev = 10

デバイスオブジェクトのインスタンス番号です。

UTC_Offset = -540

UTC_Offset を設定します。この UTC オフセット(-540)は TOKYO/JAPAN(-9hour×60min)を示しており、子午線より東側はマイナス値で西側はプラス値になります。プラス符号を入力する必要はありません。

例えば、VANCOUVER/CANADA の場合、UTC_Offset = 480 です。

OOS_AI_0 = true(または false)

AnalogInput,0 オブジェクトの OutOfService プロパティ値です。

・・・省略

OOS_SC_0 = true(または false)

Schedule,0 オブジェクトの OutOfService プロパティ値です。

・・・省略、Table 5-25 の No. 4...10 を参照ください。

Delete_saved_data

保存値を FF クリア(イレーズ)するコマンドです。

(2) BIP 固有のコマンド

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

RSK ボードの Ethernet PHY の MAC アドレスを設定します。

Bac_IP_mac_address = 192.168.10.10:47808

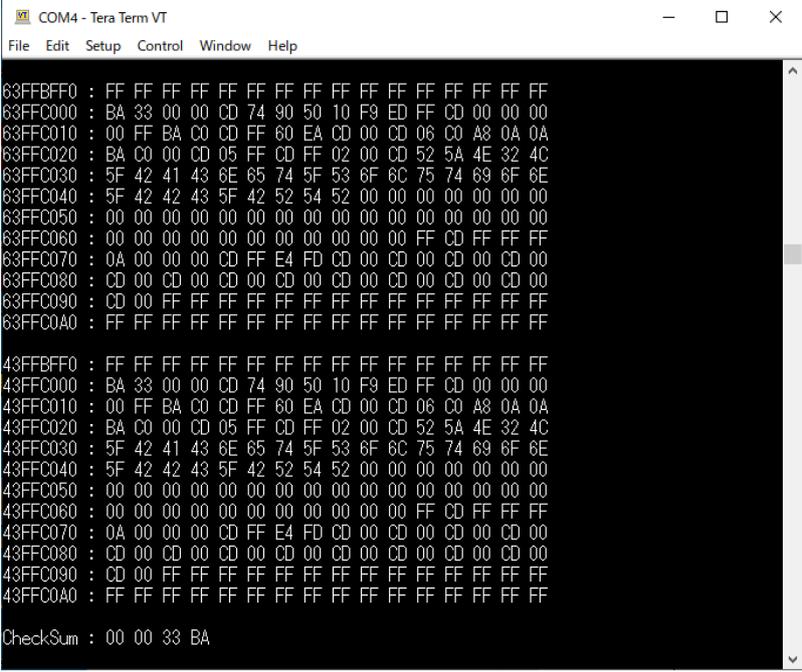
B-BC の IP アドレスと UDP ポート番号です。

(3) MSTP 固有のコマンド

MSTP_mac_address = 5

B-BC の MS/TP マスターMAC アドレスを設定します。

書き込みコマンド実行後、ボードをリセットします。Fig. 5-24 のとおり、設定データが保存されています。Flash 保存データは各プロパティの初期値として展開されます。



```
COM4 - Tera Term VT
File Edit Setup Control Window Help
63FFBFF0 : FF FF
63FFC000 : BA 33 00 00 CD 74 90 50 10 F9 ED FF CD 00 00 00
63FFC010 : 00 FF BA C0 CD FF 60 EA CD 00 CD 06 C0 A8 0A 0A
63FFC020 : BA C0 00 CD 05 FF CD FF 02 00 CD 52 5A 4E 32 4C
63FFC030 : 5F 42 41 43 6E 65 74 5F 53 6F 6C 75 74 69 6F 6E
63FFC040 : 5F 42 42 43 5F 42 52 54 52 00 00 00 00 00 00
63FFC050 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
63FFC060 : 00 00 00 00 00 00 00 00 00 00 FF CD FF FF FF
63FFC070 : 0A 00 00 00 CD FF E4 FD CD 00 CD 00 CD 00 CD 00
63FFC080 : CD 00 CD 00
63FFC090 : CD 00 FF FF
63FFC0A0 : FF FF

43FFBFF0 : FF FF
43FFC000 : BA 33 00 00 CD 74 90 50 10 F9 ED FF CD 00 00 00
43FFC010 : 00 FF BA C0 CD FF 60 EA CD 00 CD 06 C0 A8 0A 0A
43FFC020 : BA C0 00 CD 05 FF CD FF 02 00 CD 52 5A 4E 32 4C
43FFC030 : 5F 42 41 43 6E 65 74 5F 53 6F 6C 75 74 69 6F 6E
43FFC040 : 5F 42 42 43 5F 42 52 54 52 00 00 00 00 00 00
43FFC050 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
43FFC060 : 00 00 00 00 00 00 00 00 00 00 FF CD FF FF FF
43FFC070 : 0A 00 00 00 CD FF E4 FD CD 00 CD 00 CD 00 CD 00
43FFC080 : CD 00 CD 00
43FFC090 : CD 00 FF FF
43FFC0A0 : FF FF

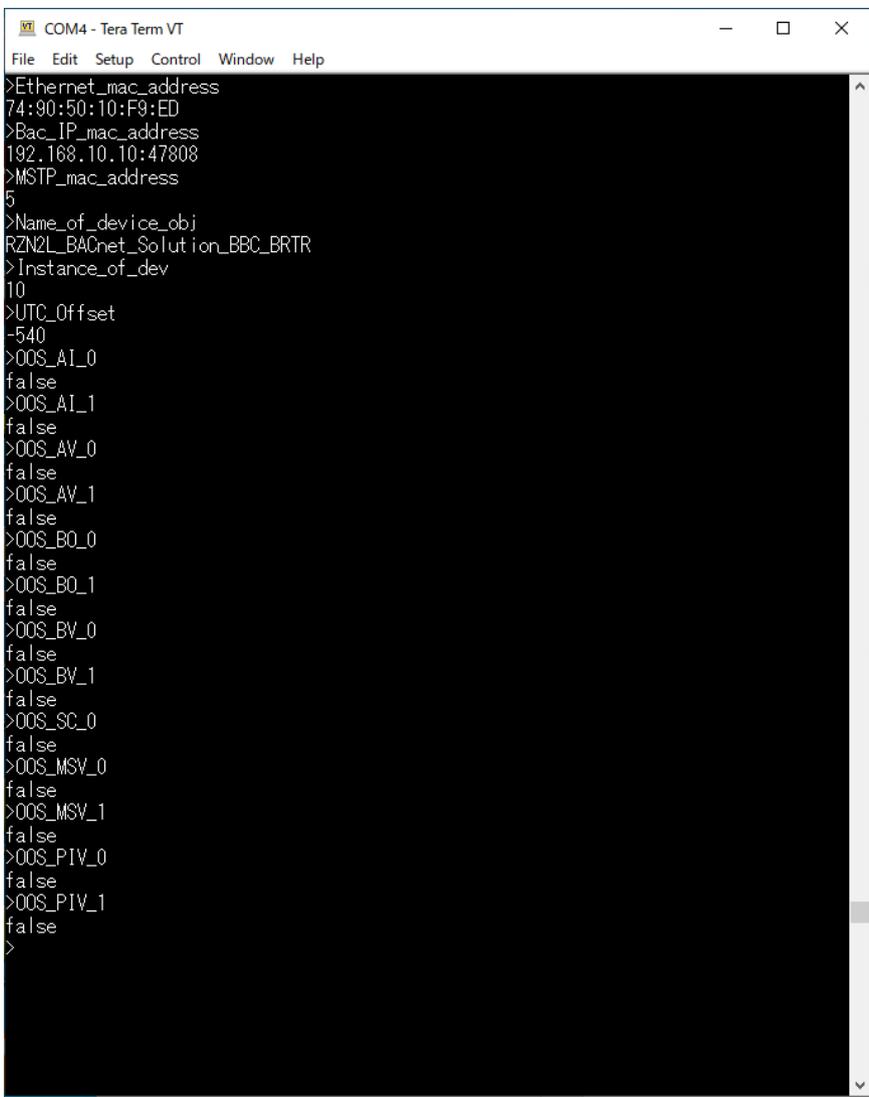
Checksum : 00 00 33 BA
```

Fig. 5-24 Memory dump after writing

5.3.3.2 Read Command

Fig. 5-25 に各 Read コマンドを実行した例を示します。">Ethernet_mac_address"のように Read コマンドを実行すると書き込んだ値(74:90:50:10:F9:ED)が表示されます。

"help_config"はコマンド書式を再表示するコマンドです。



```
COM4 - Tera Term VT
File Edit Setup Control Window Help
>Ethernet_mac_address
74:90:50:10:F9:ED
>Bac_IP_mac_address
192.168.10.10:47808
>MSTP_mac_address
5
>Name_of_device_obj
RZ/N2L_BACnet_Solution_BBC_BRTR
>Instance_of_dev
10
>UTC_Offset
-540
>OOS_AI_0
false
>OOS_AI_1
false
>OOS_AV_0
false
>OOS_AV_1
false
>OOS_BO_0
false
>OOS_BO_1
false
>OOS_BV_0
false
>OOS_BV_1
false
>OOS_SC_0
false
>OOS_MSGV_0
false
>OOS_MSGV_1
false
>OOS_PIV_0
false
>OOS_PIV_1
false
>
```

Fig. 5-25 Read command result

改定記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2024/03/25	—	新規作成

商標

- * Arm および Cortex は、Arm Limited（またはその子会社）の EU またはその他の国における登録商標です。
- * Ethernet およびイーサネットは、富士ゼロックス株式会社の登録商標です。
- * その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
 5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または盗竊その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。