

RZ/T2, RZ/N2 グループ

OPC UA Server and PubSub Sample Software

Introduction

本書は、リアルタイム OS 環境での RZ/T2, N2 で、OPC UA Server, PubSub を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。

Target Device

RZ/T シリーズ: RZ/T2M, RZ/T2H グループ

RZ/N シリーズ: RZ/N2L, RZ/N2H グループ

Contents

1. Overview	4
1.1 Abbreviations / Definitions	4
1.2 Reference	5
2. Features	6
2.1 Function List	6
2.1.1 OPC UA Server	6
2.1.2 OPC UA PubSub	7
2.1.3 Other Protocols	7
2.1.4 Security	8
2.2 Package folder structure	9
3. Requirements (Software&Hardware)	11
3.1 RZ/T2M	11
3.2 RZ/N2L	11
3.3 RZ/T2H	12
3.4 RZ/N2H	12
3.5 Common	13
4. Hardware Setup	14
4.1 RZ/T2M RSK Board	14
4.1.1 Board Setup	17
4.2 RZ/N2L RSK Board	18
4.2.1 Board Setup	21
4.3 RZ/T2H Evaluation Board	22
4.3.1 Board Setup	27
4.4 RZ/N2H Evaluation Board	28
4.4.1 Board Setup	32
5. Set up the Host	33

5.1	Host IP Address.....	33
5.2	Installation of EWARM environment.....	34
5.3	Installation of e ² studio environment	34
5.4	UaExpert.....	35
5.5	Wireshark	35
5.6	OpCmd Utility.....	36
5.7	OPC UA Demo Publisher	36
5.8	Node-RED	37
6.	Running the Sample Application	39
6.1	When using EWARM.....	39
6.1.1	For Dual core project.....	40
6.1.2	For Single core project	48
6.2	When using e ² studio.....	51
6.2.1	For Dual core project.....	54
6.2.2	For Single core project	64
7.	Demonstration of the Sample Application	68
7.1	Application Behavior.....	68
7.2	OPC UA Operation	68
7.2.1	Server	68
7.2.2	Publisher.....	77
7.2.3	Subscriber	78
8.	Software Specifications	80
8.1	Software Structure.....	80
8.1.1	Overview of Software Components.....	80
8.1.2	Overview of Each Component.....	81
8.2	Task Structure	85
8.2.1	Task List	85
8.2.2	Task Operation	86
8.3	Footprint (ROM and RAM Usage).....	89
8.3.1	RZ/T2M project.....	89
8.3.2	RZ/N2L project	97
8.3.3	RZ/T2H, N2H project (CR52_dual)	101
8.3.4	RZ/T2H, N2H project (CA55_dual).....	109
9.	Appendix	117
9.1	File Generation of open62541	117
9.1.1	Linux environment Setup.....	117
9.1.2	Install CMake.....	119
9.1.3	open62541 File Generation.....	120
9.1.4	Changes in Generated Files.....	123

9.2	TFTP.....	124
9.2.1	File Writing.....	124
9.2.2	File Reading	124
9.3	LLDP, SNMP MIB.....	125
9.4	When configuring OPC UA PubSub settings using the Method	126
9.4.1	Publisher.....	126
9.4.2	Subscriber	129
9.4.3	OPC UA Client for PubSub Connection	133
9.5	Configuration for launching FSP Smart Configurator from EWARM.....	135
9.6	How to Change Various Settings in the Sample Software.....	136
9.6.1	When using e ² studio.....	136
9.6.2	When using EWARM.....	142
	Revision History.....	147

1. Overview

産業用アプリケーションの相互運用を実現する OPC UA は、ファクトリーオートメーションだけでなく広く様々な産業分野にも使われるようになってきています。

本書は、産業ネットワーク用 RZ プロセッサ RZ/T2, RZ/N2 において、リアルタイム OS を使用して、OPC UA Server, PubSub を実現するためのサンプルソフトの構成、およびその使い方について説明します。

1.1 Abbreviations / Definitions

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。

Table 1.1. Abbreviations/Definitions

Index	Abbreviations/Definitions	Description
1	BPS	Board Support Package
2	CA55	Cortex®-A55
3	CR52	Cortex®-R52
4	CRL	Certificate Revocation List
5	EVB	Evaluation Board
6	EWARM	Embedded Workbench® for ARM
7	FAT	File Allocation Table
8	FSP	Flexible Software Package
9	FSP SC	FSP Smart Configurator
10	LLDP	Link Layer Discovery Protocol
11	MIB	Management Information Base
12	MQTT	Message Queuing Telemetry Transport
13	NTP	Network Time Protocol
14	OPC UA	Open Platform Communications Unified Architecture
15	PubSub	Publisher Subscriber
16	RSK	Renesas Starter Kit
17	SNMP	Simple Network Management Protocol
18	SNTP	Simple Network Time Protocol
19	TFTP	Trivial File Transfer Protocol
20	TLV	Type Length Value
21	UADP	UA Datagram Protocol
22	XML	Extensible Markup Language

1.2 Reference

Table 1.2. Technical Inputs for RZ/T2M

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZ/T2M hardware.	Renesas Starter Kit+ for RZ/T2M ユーザーズマニュアル	R20UT4939*****
User's Manual	Provides technical details of the RZ/T2M microprocessor.	RZ/T2M グループ ユーザーズマニュアル ハードウェア編	R01UH0916*****

Table 1.3. Technical Inputs for RZ/N2L

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZ/N2L hardware.	Renesas Starter Kit+ for RZ/N2L ユーザーズマニュアル	R20UT4984*****
User's Manual	Provides technical details of the RZ/N2L microprocessor.	RZ/N2L グループ ユーザーズマニュアル ハードウェア編	R01UH0955*****

Table 1.4. Technical Inputs for RZ/T2H

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RZ/T2H Evaluation Board hardware.	RZ/T2H Evaluation Board Kit ユーザーズマニュアル	R20UT5405*****
User's Manual	Provides technical details of the RZ/T2H, RZ/N2H microprocessors.	RZ/T2H および RZ/N2H グループ ユーザーズマニュアル ハードウェア編	R01UH1039*****

Table 1.5. Technical Inputs for RZ/N2H

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RZ/N2H Evaluation Board hardware.	RZ/N2H Evaluation Board Kit ユーザーズマニュアル	R20UT5522*****
User's Manual	Provides technical details of the RZ/T2H, RZ/N2H microprocessors.	RZ/T2H および RZ/N2H グループ ユーザーズマニュアル ハードウェア編	R01UH1039*****

Table 1.6. Technical Inputs for RZ/T2, N2 (Common)

Document Type	Description	Document Title	Document No.
Application Note	RZ/T2, N2 FSP スタートガイド	RZ/T2, RZ/N2 Getting Started with Flexible Software Package	R01AN6434*****

2. Features

このパッケージは RZ/T2, N2 シリーズ上で動作する OPC UA のソフトウェアを提供します。

2.1 Function List

本サンプルソフトがサポートする機能は下記の通りです。

2.1.1 OPC UA Server

本サンプルソフトの OPC UA Server 仕様は Table 2.1 の通りです。

Table 2.1. OPC UA Server Specifications

Items	
Server Interface	
Profile	Micro Embedded Device 2017 Server
SecurityPolicy	None Basic256 Basic128Rsa15 Basic256Sha256 Aes128-Sha256-RsaOaep
SecurityMode	None Sign SignAndEncrypt
UserAuthenticationPolicy	Anonymous Username / Password 1. "user1"/"password" 2. "user2"/"password1" X509 (Certificate)
Server Capabilities	
Max sessions	2
Max subscriptions per session	2
Max monitored items per subscription	100
Min publishing interval	500ms
Min sampling interval	250ms

2.1.2 OPC UA PubSub

本サンプルソフトの OPC UA PubSub 仕様は Table 2.2 の通りです。SKS との通信は現時点で未対応です。

Table 2.2. OPC UA PubSub Specifications

Items	
PubSub Interface	
Layout	UADP Periodic Fixed Layout UADP Dynamic Layout
SecurityPolicy	None PubSub-Aes128-CTR PubSub-Aes256-CTR
SecurityMode	None Sign SignAndEncrypt

2.1.3 Other Protocols

本サンプルソフトは、Table 2.3 に示すプロトコルに対応しています。それぞれの詳細は Details に記載された章をご確認ください。

Table 2.3. Sample Software Support Protocols

Protocol	Details
SNTP Client	デバイス起動時に NTP サーバから時刻情報を取得
TFTP Server	Chapter 9.2
LLDP Agent	Chapter 9.3
SNMP Agent	Chapter 9.3

2.1.4 Security

セキュリティ機能に使用する各種ファイルは、ファイルシステム（FreeRTOS+FAT）機能により管理され、評価ボードに接続された USB フラッシュドライブから読み出しされます。また、TFTP 通信によりイーサネット経由で、USB フラッシュドライブにファイル書き込みすることができます。使用するファイル名は Table 2.4 に示す通りです。Attached to Package 列に✓が記載されているファイルは、サンプルとしてパッケージ内の”keyfiles.zip”に格納されています。

本パッケージに添付される crl_1.pem には、cert_client_1.der の失効情報が記載されています。証明書失効リストの動作確認時以外には、CRL ファイルの書き込みはしないよう注意してください。

ファイルを書き込みする際は、USB フラッシュドライブの<USB_ROOT>/passw0rd/フォルダに格納してください。

Table 2.4. Key Files

File	File Name	Attached to Package
Server Certificate	cert_server.der	✓
Server Private Key	pkey_server.der	✓
Client Certificate (TrustList)	cert_client_1.der cert_client_2.der :	✓ -
IssuerList	issuer_1.der issuer_2.der :	-
CRL	crl_1.pem crl_2.pem :	✓ -
Client Certificate (TrustList) for UserAuthentication	cert_c_usr_1.der cert_c_usr_2.der :	✓ -
IssuerList for UserAuthentication	issuer_usr_1.der issuer_usr_2.der :	-
CRL for UserAuthentication	crl_usr_1.pem crl_usr_2.pem :	-
PubSub Encryption Key for AES128	pubsub_aes128_key.bin	✓
PubSub Encryption Key for AES256	pubsub_aes256_key.bin	✓

✓はパッケージに添付、-は非添付

TrustList, CRL, IssuerList の最大数の変更方法については 9.6 章を、TFTP を使用した USB フラッシュドライブへのファイル書き込み方法は 9.2 章を参照ください。

また、USB フラッシュドライブの接続無しでセキュリティ動作をすることも可能です。その場合は、9.6 章を参考にしてください。

2.2 Package folder structure

本サンプルソフトのフォルダ構成を以下に示します。

Table 2.5. RZ/T2M Folder structure

Item	Description
RZT2M_OPC_EVB_rev0400	
├── common	Common files for all projects
└── project	Projects
├── CR52_dual	Dual core project
├── e2studio	e ² studio project
├── RZT2M_RSK_OPC_Primary_0	For CR52_0 (primary core)
└── RZT2M_RSK_OPC_Secondary_1	For CR52_1 (secondary core)
└── ewarm	EWARM project
├── RZT2M_RSK_OPC_Primary_0	For CR52_0 (primary core)
└── RZT2M_RSK_OPC_Secondary_1	For CR52_1 (secondary core)

Table 2.6. RZ/N2L Folder structure

Item	Description
RZN2L_OPC_EVB_rev0400	
├── common	Common files for all projects
└── project	Projects
├── CR52_single	Single core project
├── e2studio	e ² studio project
└── RZN2L_RSK_OPC_Primary_0	For CR52_0 (primary core)
└── ewarm	EWARM project
└── RZN2L_RSK_OPC_Primary_0	For CR52_0 (primary core)

Table 2.7. RZ/T2H Folder structure

Item	Description
RZT2H_OPC_EVB_rev0400	
├── common	Common files for all projects
└── project	Projects
├── CA55_dual	Dual core project
├── e2studio	e ² studio project
├── RZT2H_EVB_OPC_CA55_dual_CR52_0_Primary_0	For CR52_0 (primary core)
└── RZT2H_EVB_OPC_CA55_dual_CA55_2_Secondary_1	For CA55_2 (secondary core)
└── ewarm	EWARM project
├── RZT2H_EVB_OPC_CA55_dual_CR52_0_Primary_0	For CR52_0 (primary core)
└── RZT2H_EVB_OPC_CA55_dual_CA55_2_Secondary_1	For CA55_2 (secondary core)
├── CR52_dual	Dual core project
├── e2studio	e ² studio project
├── RZT2H_EVB_OPC_CR52_dual_CR52_0_Primary_0	For CR52_0 (primary core)
└── RZT2H_EVB_OPC_CR52_dual_CR52_1_Secondary_1	For CR52_1 (secondary core)
└── ewarm	EWARM project
├── RZT2H_EVB_OPC_CR52_dual_CR52_0_Primary_0	For CR52_0 (primary core)
└── RZT2H_EVB_OPC_CR52_dual_CR52_1_Secondary_1	For CR52_1 (secondary core)

Table 2.8. RZ/N2H Folder structure

Item	Description
RZN2H_OPC_EVB_rev0400	
├── common	Common files for all projects
├── project	Projects
│ ├── CA55_dual	Dual core project
│ │ ├── e2studio	e ² studio project
│ │ │ ├── RZN2H_EVB_OPC_CA55_dual_CR52_0_Primary_0	For CR52_0 (primary core)
│ │ │ └── RZN2H_EVB_OPC_CA55_dual_CA55_2_Secondary_1	For CA55_2 (secondary core)
│ │ └── ewarm	EWARM project
│ │ ├── RZN2H_EVB_OPC_CA55_dual_CR52_0_Primary_0	For CR52_0 (primary core)
│ │ └── RZN2H_EVB_OPC_CA55_dual_CA55_2_Secondary_1	For CA55_2 (secondary core)
│ └── CR52_dual	Dual core project
│ ├── e2studio	e ² studio project
│ │ ├── RZN2H_EVB_OPC_CR52_dual_CR52_0_Primary_0	For CR52_0 (primary core)
│ │ └── RZN2H_EVB_OPC_CR52_dual_CR52_1_Secondary_1	For CR52_1 (secondary core)
│ └── ewarm	EWARM project
│ ├── RZN2H_EVB_OPC_CR52_dual_CR52_0_Primary_0	For CR52_0 (primary core)
│ └── RZN2H_EVB_OPC_CR52_dual_CR52_1_Secondary_1	For CR52_1 (secondary core)

Table 2.9. Common folder structure

Item	Description
common	Common files for all projects
├── oss	Open Source Software
│ ├── MBEDTLS	Mbed TLS
│ ├── OPC_UA_SERVER	open62541
│ ├── amazon-freertos	FreeRTOS
│ └── lwip	lwIP
└── renesas	Renesas Software
│ ├── application	User OPC UA Application, KeyFiles
│ └── module	Modules
│ ├── ether_netif	NETIF module
│ ├── freertos_plus_fat	FreeRTOS+FAT module
│ ├── lwip_lldp	LLDP module
│ ├── lwip_port	Port module, SNMP/MIB module
│ └── serial_io	Serial interface module
└── oss_deps	OSS dependencies
│ ├── amazon-freertos	FreeRTOS dependency
│ ├── lwip	lwIP dependency
│ └── mbedtls	Mbed TLS dependency

3. Requirements (Software&Hardware)

本プロジェクトは、以下のボードとツールを使用して開発およびテストされています。

3.1 RZ/T2M

Table 3.1. RZ/T2M environment

Category	Name	Version	Description
Board	RZ/T2M Evaluation Board	-	Renesas RZ/T2M-RSK - Renesas Starter Kit Plus for RZ/T2M Renesas
IDE	EWARM	9.60.3	IAR Systems IAR Embedded Workbench for Arm IAR
	e ² studio	2025-12	RZ FSP
Configurator	FSP Smart Configurator	2025-12	GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.1.arm-13-24	
Emulator	J-Link™	8.60	SEGGER SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace
	I-jet	-	IAR Systems IAR debug probes IAR

3.2 RZ/N2L

Table 3.2. RZ/N2L environment

Category	Name	Version	Description
Board	RZ/N2L Evaluation Board	-	Renesas RZ/N2L-RSK - Renesas Starter Kit+ for RZ/N2L Renesas
IDE	EWARM	9.60.3	IAR Systems IAR Embedded Workbench for Arm IAR
	e ² studio	2025-12	RZ FSP
Configurator	FSP Smart Configurator	2025-12	GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.1.arm-13-24	
Emulator	J-Link™	8.60	SEGGER SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace
	I-jet	-	IAR Systems IAR debug probes IAR

3.3 RZ/T2H

Table 3.3. RZ/T2H environment

Category	Name	Version	Description
Board	RZ/T2H Evaluation Board	-	Renesas RZ/T2H-EVKIT - Evaluation Board Kit for RZ/T2H Renesas
IDE	EWARM	9.60.3	IAR Systems IAR Embedded Workbench for Arm IAR RZ/T2H, N2H 向けパッチ EWARM_Patch_for_RZT2H_N2H_rev1.0.zip の適用が必要
	e ² studio	2025-12	RZ FSP
Configurator	FSP Smart Configurator	2025-12	GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.1.arm-13-24	GitHub
	GNU ARM A-Profile (AArch64 bare-metal)	13.2.1.20231009	
Emulator	J-Link™	8.60	SEGGER SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace
	I-jet	-	IAR Systems IAR debug probes IAR

3.4 RZ/N2H

Table 3.4. RZ/N2H environment

Category	Name	Version	Description
Board	RZ/N2H Evaluation Board	-	Renesas RZ/N2H-EVKIT - Evaluation Board Kit for RZ/N2H Renesas
IDE	EWARM	9.60.3	IAR Systems IAR Embedded Workbench for Arm IAR RZ/T2H, N2H 向けパッチ EWARM_Patch_for_RZT2H_N2H_rev1.0.zip の適用が必要
	e ² studio	2025-12	RZ FSP
Configurator	FSP Smart Configurator	2025-12	GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.1.arm-13-24	GitHub
	GNU ARM A-Profile (AArch64 bare-metal)	13.2.1.20231009	
Emulator	J-Link™	8.60	SEGGER SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace
	I-jet		IAR Systems IAR debug probes IAR

3.5 Common

Table 3.5. Common environment

Category	Name	Version	Link	Notes
USB フラッシュドライブ (USB メモリ)	-	-	-	ファイルシステムのファイル保存先として使用 FAT32 形式へのフォーマットが必要 ※USB フラッシュドライブ無しでも動作可能です。
OPC UA Client ツール	UaExpert	1.7.2	OPC UA Clients - Unified Automation (unified-automation.com)	
Packet analyzer	Wireshark	4.4.7	Wireshark · Download	
OPC 通信汎用ツール	OpcCmd Utility	-	OpcCmd Utility Download - OPC Labs Knowledge Base	OPC UA PubSub Subscriber として使用
OPC UA Publisher ツール	UADemoPublisher	-	OPC UA Demo Publisher Download - OPC Labs Knowledge Base	
OPC UA Client for PubSub Connection	client_pubsub_connection	1.2.0	-	パッケージに同梱 OPC UA PubSub の設定に使用
ローコード開発ツール	Node-RED	4.0.3	Node-RED (nodered.org) Node.js — Run JavaScript Everywhere (nodejs.org)	OPC UA Client として使用 Node の読み出し値をグラフで表示

4. Hardware Setup

このドキュメントでは、ハードウェアの主要な機能を説明します。詳細については各評価ボードのユーザーマニュアルおよび、回路図を参照してください。

4.1 RZ/T2M RSK Board

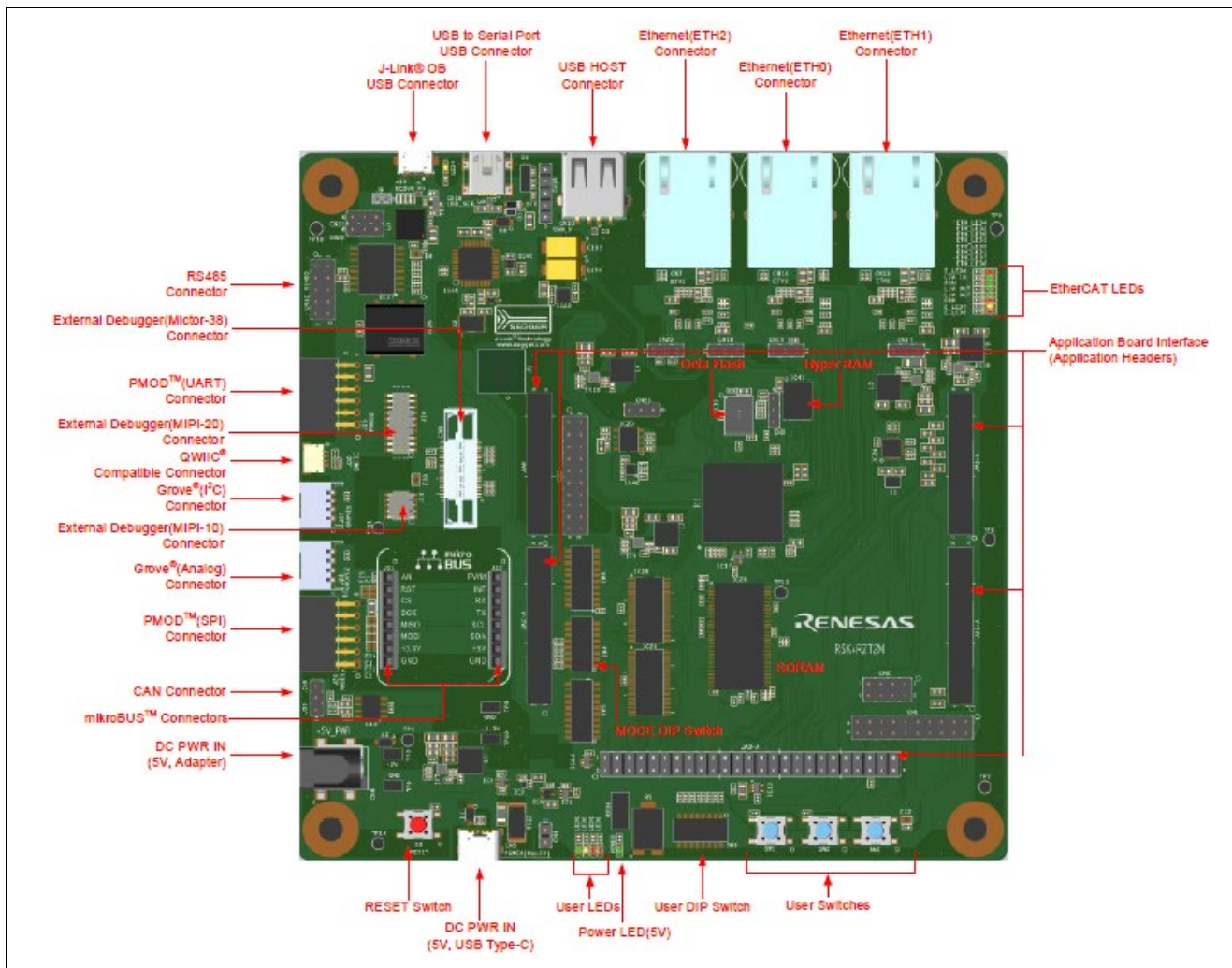


Figure 4.1. RZ/T2M RSK board layout

表中の赤文字は評価ボードの工場出荷時の設定から変更が必要な箇所を表しています。

Table 4.1. Jumper and Switch Configuration

Reference	Jumper Position	Description
CN8	Short 1-2	Use Octa Flash
	Short 2-3	Use QSPI Serial Flash
CN17	Short 1-2	VCC1833_2 Use power supply at 3.3V (Use SDRAM)
	Short 2-3	VCC1833_2 Use power supply at 1.8V
CN18	Short 1-2	Use 3 ports in the same PHY mode
	Short 2-3	Use ports 0, 1 in the same PHY mode, use port 2 in different PHY modes
CN19	Short 1-2	Use 3 ports in the same PHY mode
	Short 2-3	Use ports 0, 1 in the same PHY mode, use port 2 in different PHY modes
CN20	Short 1-2	Use 3 ports in the same PHY mode
	Short 2-3	Use ports 0, 1 in the same PHY mode, use port 2 in different PHY modes
CN21	Short 1-2	Use RS485 transmission method in full duplex
	Short 2-3	Use RS485 transmission method in half duplex
CN22	Short 1-2	Use RS485 transmission method in full duplex
	Short 2-3	Use RS485 transmission method in half duplex
CN23	Short 1-2	Not use CS5#
	Short 2-3	Use CS5#
J9	Open	Use J-Link OB
	Short	Use External Debugger

Table 4.2. SW4 Settings

SW4	Setting	Description
SW4-1	ON	16bit Bus boot mode (NOR Flash ROM Boot).
SW4-2	OFF	
SW4-3	ON	
SW4-4	ON	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	OFF	MDW=1, ACTM 1 wait, should be set when TCM is used with CPU operating frequencies above 400MHz
SW4-6	OFF	
SW4-7	OFF	
SW4-8	OFF	

Table 4.3. SW5 Setting.

SW5	Setting	Description
SW5-1	OFF	-
SW5-2	OFF	-
SW5-3	ON	Enable the "SCI_RTS" signal.
SW5-4	OFF	
SW5-5	ON	Enable the "SCI_RXD" signal.
SW5-6	OFF	
SW5-7	OFF	
SW5-8	OFF	Enable the "SCK3" signal.
SW5-9	ON	
SW5-10	OFF	

Table 4.4. SW6 Setting

SW6	Setting	Description
SW6-1	ON	Enable external bus
SW6-2	OFF	-
SW6-3	ON	TRACE_CTL signal is enabled
SW6-4	OFF	
SW6-5	OFF	SCI_TXD signal is enabled
SW6-6	ON	
SW6-7	OFF	MB_RST# signal is enabled
SW6-8	ON	
SW6-9	OFF	CAN_RX_OB signal is enabled
SW6-10	ON	

4.1.1 Board Setup

Figure 4.2に OPC UA のサンプルソフト実行時の接続図を示します。RZ/T2M RSK ボードに Ethernet ケーブル、デバッガ、5V DC の各ケーブルを接続してください。

ETH0~ETH1 のいずれの 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。I-jet を使用する場合は、J9 をショートとし、J20 に I-jet を接続します。

本サンプルソフトでは、SNTP による時刻同期を行っているため、NTP Server に接続する必要があります。動作確認時には、Windows PC を NTP Server として使用しています。

OPC UA のセキュリティ機能で使用する鍵ファイルを FreeRTOS+FAT で管理する場合、RSK ボードへの USB フラッシュドライブの接続が必要になります。

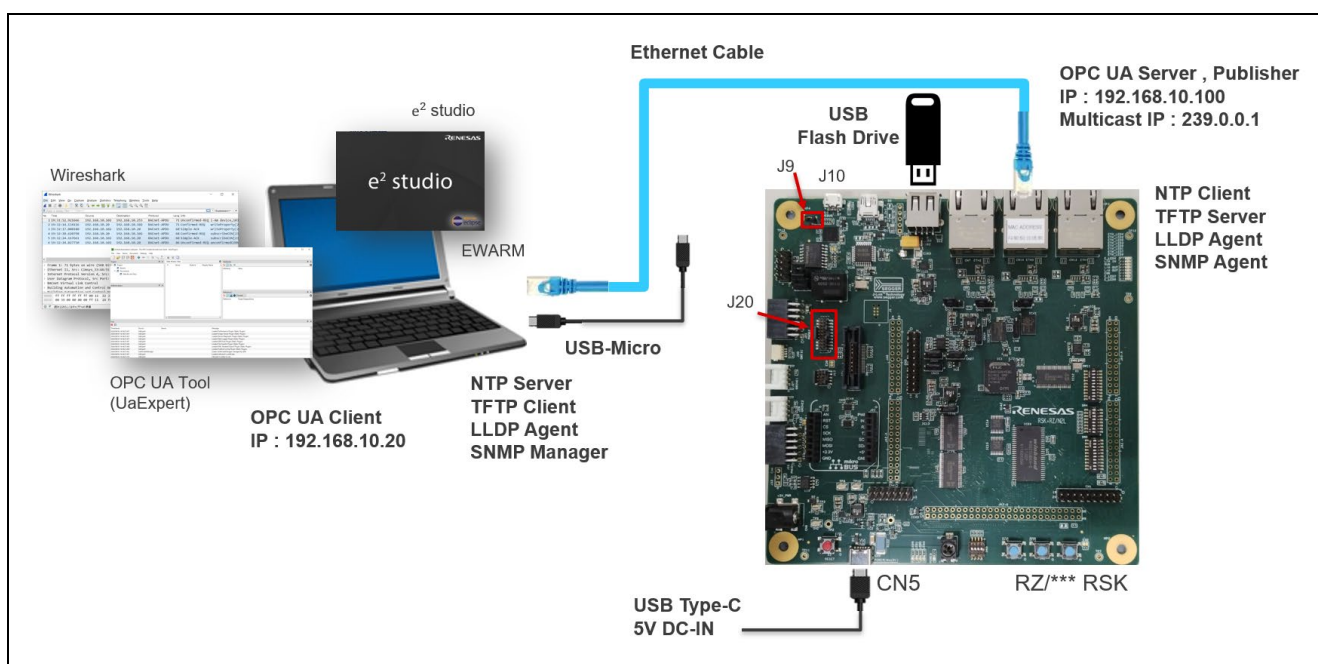


Figure 4.2. RZ/T2M RSK board OPC UA Server hardware connection diagram

4.2 RZ/N2L RSK Board

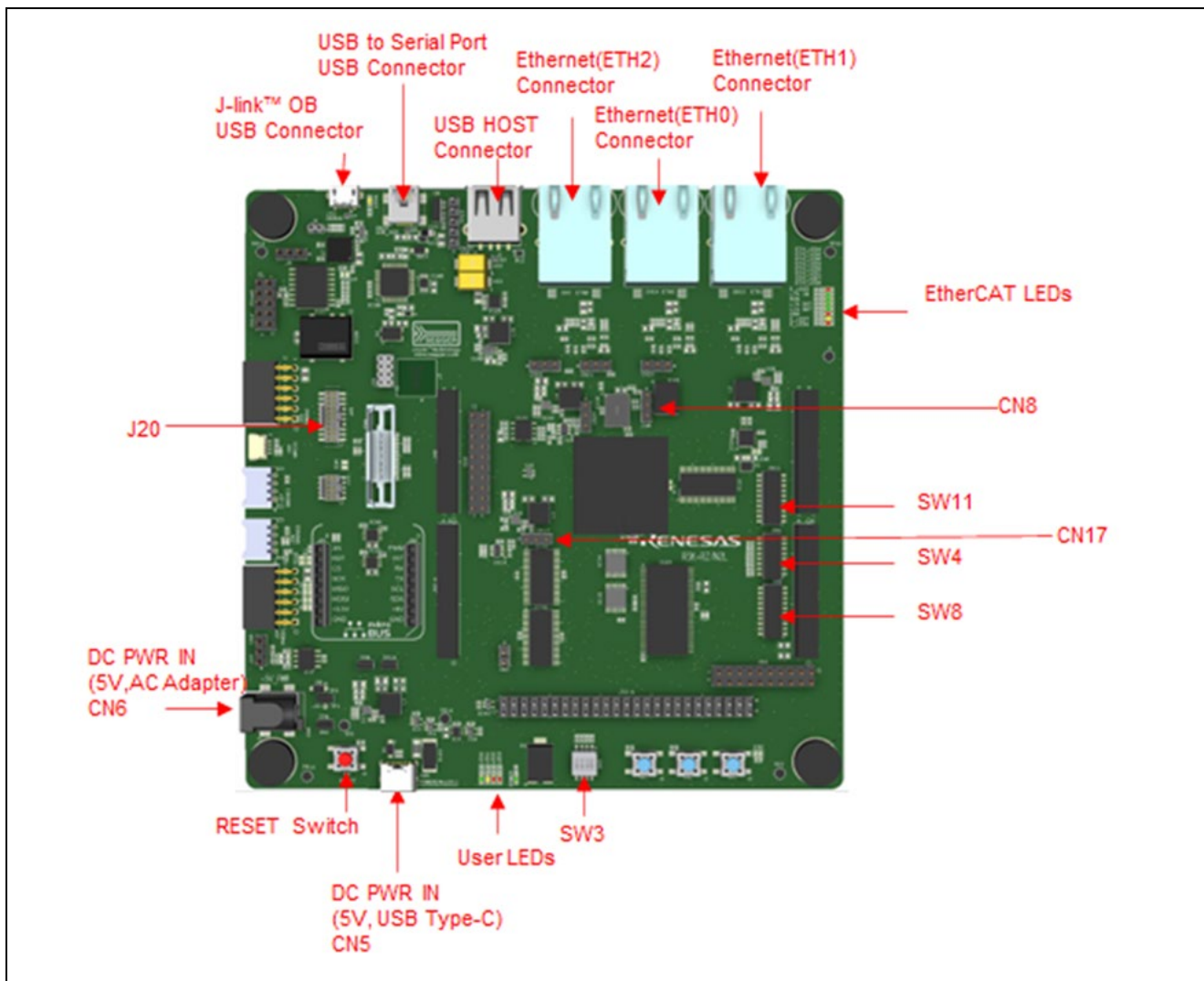


Figure 4.3. RZN2L RSK board layout

表中の赤文字は評価ボードの工場出荷時の設定から変更が必要な箇所を表しています。

Table 4.5. Jumper and Switch Configuration

Reference	Jumper Position	Description
CN8	Short 1-2	Use Octa Flash
	Short 2-3	Use QSPI Serial Flash
CN17	Short 1-2	VCC1833_2 Use power supply at 3.3V (Use SDRAM)
	Short 2-3	VCC1833_2 Use power supply at 1.8V
CN20	Short 1-2	Use 3 ports in the same PHY mode
	Short 2-3	Use ports 0, 1 in the same PHY mode, use port 2 in different PHY modes
CN21	Short 1-2	Use 3 ports in the same PHY mode
	Short 2-3	Use ports 0, 1 in the same PHY mode, use port 2 in different PHY modes
CN22	Short 1-2	Use 3 ports in the same PHY mode
	Short 2-3	Use ports 0, 1 in the same PHY mode, use port 2 in different PHY modes
CN24	Short 1-2	VCC1833_2 Use power supply at 3.3V (Use SDRAM)
	Short 2-3	VCC1833_2 Use power supply at 1.8V
CN25	Short 1-2	Use other than the SHOST interface
	Short 2-3	Use the SHOST interface
CN27	Short 1-2	Use HyperRAM
	Short 2-3	Use the SHOST interface
CN29	Short 1-2	Use the USB Serial
	Short 2-3	Use the SHOST interface
CN31	Short 1-2	Use RS485 transmission method in full duplex
	Short 2-3	Use RS485 transmission method in half duplex
CN32	Short 1-2	Use RS485 transmission method in full duplex
	Short 2-3	Use RS485 transmission method in half duplex
J9	Open	Use J-Link OB
	Short	Use External Debugger

Table 4.6. SW4 Settings

SW4	Setting	Description
SW4-1	ON	16bit Bus boot mode (NOR Flash ROM Boot)
SW4-2	OFF	
SW4-3	ON	
SW4-4	ON	MDD=0, JTAG Authentication by Hash is disabled.
SW4-5	OFF	MDW=1, ACTM 1 wait, should be set when TCM is used with CPU operating frequencies above 400MHz
SW4-6	OFF	Enables signals other than the trace signal. (Motor, RS485, etc)
SW4-7	OFF	Enables signals other than the external bus. (CAN, Emulator, I2C, etc.)
SW4-8	OFF	Enable SW3.

Table 4.7. SW8 Setting.

SW8	Setting	Description
SW8-1	OFF	Enable the "LED_GREEN" signal.
SW8-2	ON	
SW8-3	OFF	
SW8-4	ON	Enable the "LED5" signal.
SW8-5	OFF	
SW8-6	OFF	RS485_DE & M2_VN Configuration Switch Setting
SW8-7	ON	
SW8-8	OFF	CAN_TX & IRQ4 & P02_2 Configuration Switch Setting
SW8-9	OFF	
SW8-10	ON	

Table 4.8. SW11 Setting

SW11	Setting	Description
SW11-1	ON	Enable the "LED_RED2" signal.
SW11-2	OFF	
SW11-3	OFF	
SW11-4	OFF	RS485_RX & M2_UP Configuration Switch Setting
SW11-5	ON	
SW11-6	OFF	P21_5 & M2_VP Configuration Switch Setting
SW11-7	ON	
SW11-8	OFF	CAN_RX & ADTRG & P01_7 Configuration Switch Setting
SW11-9	OFF	
SW11-10	ON	

4.2.1 Board Setup

Figure 4.4 に OPC UA の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、デバッガ、5V DC の各ケーブルを接続してください。

ETH0~ETH1 のいずれの 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。I-jet を使用する場合は、J9 をショートとし、J20 に I-jet を接続します。

本サンプルソフトでは、SNTP による時刻同期を行っているため、NTP Server に接続する必要があります。動作確認時には、Windows PC を NTP Server として使用しています。

OPC UA のセキュリティ機能で使用する鍵ファイルを FreeRTOS+FAT で管理する場合、RSK ボードへの USB フラッシュドライブの接続が必要になります。

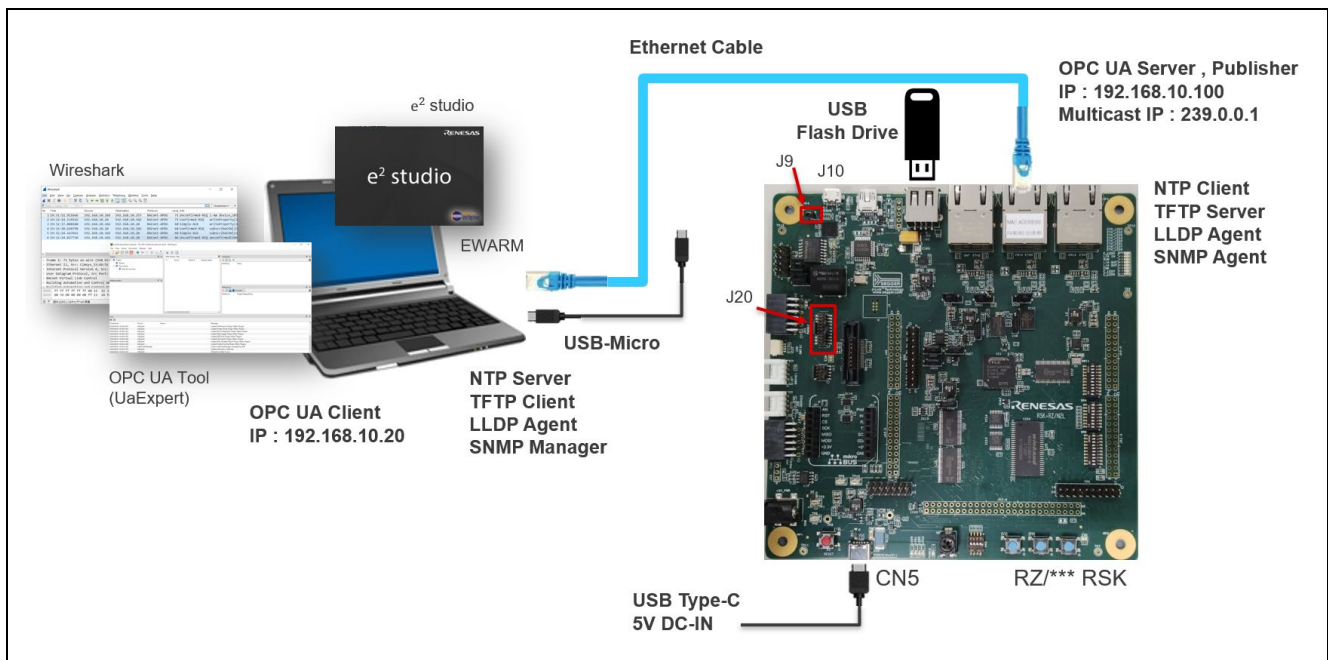


Figure 4.4. RZ/N2L RSK board OPC UA Server hardware connection diagram

4.3 RZ/T2H Evaluation Board

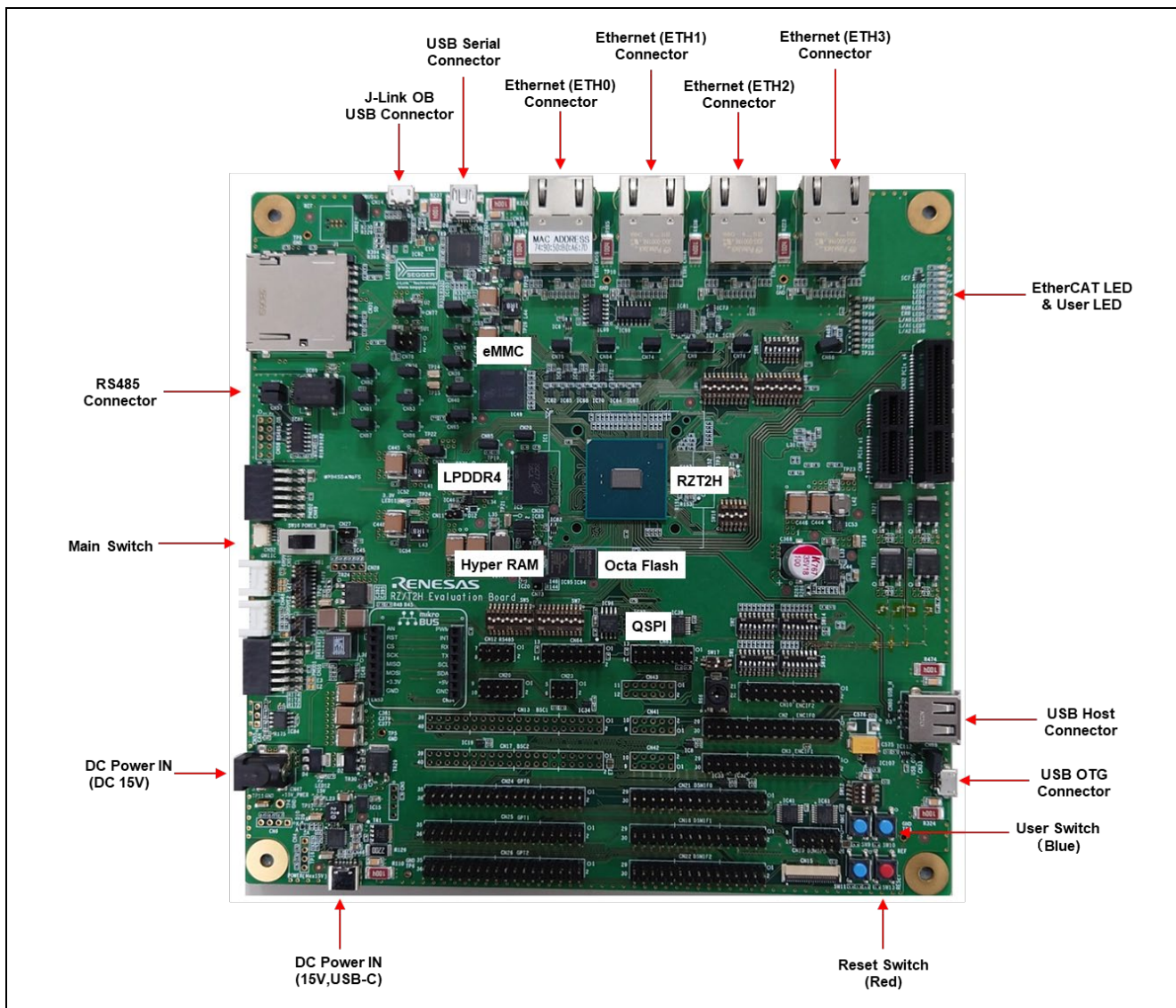


Figure 4.5. RZT2H Evaluation board layout

表中の赤文字は評価ボードの工場出荷時の設定から変更が必要な箇所を表しています。

Table 4.9. Jumper and Switch Configuration

Reference	Jumper Position	Description
CN62	Short	The on-board debugging function J-Link™ OB is disabled.
CN73	Open	PCIe reset is not included in the system reset factors.
CN9	Short 1-2	VCC1833_0 is supplied to VCC_ETH2_MDIO (SW2-6 is ON: When P21_4 and P21_5 are selected for MDIO).
	Short 2-3	VCC1833_2 is supplied to VCC_ETH2_MDIO (SW2-6 is OFF: When P30_5 and P30_6 are selected for MDIO).
CN37	Short 1-2	1.8-V power is supplied to VCC1833_0 (for Ethernet port 0).
	Short 2-3	3.3-V power is supplied to VCC1833_0 (for Ethernet port 0).
CN38	Short 1-2	1.8-V power is supplied to VCC1833_1 (for Ethernet port 1).
	Short 2-3	3.3-V power is supplied to VCC1833_1 (for Ethernet port 1).
CN39	Short 1-2	1.8-V power is supplied to VCC1833_2 (for Ethernet port 2).
	Short 2-3	3.3-V power is supplied to VCC1833_2 (for Ethernet port 2).
CN40	Short 1-2	1.8-V power is supplied to VCC1833_3 (for Ethernet port 3).
	Short 2-3	3.3-V power is supplied to VCC1833_3 (for Ethernet port 3).
CN77	Short 1-2	3.3-V power is supplied to VCC1833_7 (for SD1).
	Short 2-3	The output from the power-supply control IC for SD1 is supplied to VCC1833_7 (for SD1).
CN78	Short 1-2	3.3-V power is supplied to VCC1833_6 (for SD0).
	Short 2-3	The output from the power-supply control IC for SD0 is supplied to VCC1833_6 (for SD0).
	Short 5-6	1.8-V power is supplied to VCC1833_6 (for SD0).
CN56	Short 1-2	Full-duplex communication
CN57	Short 2-3	Half-duplex communication

Table 4.10. SW14 Settings

SW14	Setting	Description
SW14-1	ON	xSPI1 boot mode (x1 boot serial flash)
SW14-2	OFF	
SW14-3	ON	
SW14-4	OFF	CPU0 ATCM wait cycle = 1 wait cycle
SW14-5	OFF	CPU1 ATCM wait cycle = 1 wait cycle
SW14-6	OFF	Supply voltage of boot peripheral is 3.3 V
SW14-7	ON	JTAG mode = Normal mode
SW14-8	OFF	Not used

Table 4.11. SW1 Settings

SW1	Setting	Description
SW1-1	ON	XTALSEL = 'L' Select oscillator for RZ/T2H clock input.
SW1-2	OFF	(At the time of shipment = OFF)
SW1-3	ON	P35_3,4,5,6 are connected to SW12 and used as user DIPSW inputs.
SW1-4	OFF	P13_4, P13_5, and P14_0 are used as RXD3, TXD3, and DE3 of the RS485.
SW1-5	ON	P00_0,1,2 are used as USB power supply IC control signals.
SW1-6	ON	P01_0,1,2,4,5,6,7 and P02_0,1,2,3 are used as XSPI1 signals.
SW1-7	OFF	(At the time of shipment = OFF)
SW1-8	OFF	(At the time of shipment = OFF)

Table 4.12. SW2 Settings

SW2	Setting	Description
SW2-1	ON	P12_0 to 7, P13_0 to 2 are connected to eMMC.
SW2-2	ON	
SW2-3	ON	P17_4, P08_5, and P08_6 are used as SD1 control signals.
SW2-4	OFF	(At the time of shipment = OFF)
SW2-5	OFF	(At the time of shipment = OFF)
SW2-6	ON	GMAC0 (P21_4, P21_5) are connected to MDC/MDIO of Ethernet Port2.
SW2-7	ON	P29_1 to 7, P30_0 to 4, P31_2 to 5 are used as Ethernet Port2 control signals.
SW2-8	ON	P27_2, P33_2 to P33_7, P34_0 to P34_5, P34_7, and P35_0 to P35_2 are used as control signals for Ethernet port 3.

Table 4.13. SW4 Settings

SW4	Setting	Description
SW4-1	ON	P27_0 is used as ETH1_CRS.
SW4-2	OFF	
SW4-3	ON	P27_1 is used as ETH1_COL.
SW4-4	OFF	
SW4-5	ON	P27_4 is used as RXD0 of USB-to-serial conversion.
SW4-6	OFF	
SW4-7	ON	P27_5 is used as TXD0 of USB-to-serial conversion.
SW4-8	OFF	

Table 4.14. SW5 Settings

SW5	Setting	Description
SW5-1	OFF	P32_2 is used as USER_LED1.
SW5-2	ON	
SW5-3	OFF	When SW2-3 = ON, P08-6 is used as SD1_IOVS.
SW5-4	ON	
SW5-5	OFF	P07_5 is used as XSPI0_ECS# for OctaFlash.
SW5-6	ON	
SW5-7	OFF	P23_0 is used as ESC_LINKACT1.
SW5-8	ON	
SW5-9	OFF	P22_7 is used as ESC_LINKACT0.
SW5-10	ON	

Table 4.15. SW6 Settings

SW6	Setting	Description
SW6-1	OFF	P11_0 is used as ESC_RESETOUT2#.
SW6-2	OFF	
SW6-3	ON	
SW6-4	OFF	P11_0 is used as ESC_RESETOUT01#.
SW6-5	ON	
SW6-6	OFF	(At the time of shipment = OFF)
SW6-7	ON	P23_3 is used as ESC_I2CCLK.
SW6-8	OFF	
SW6-9	ON	P23_4 is used as ESC_I2CDATA.
SW6-10	OFF	

Table 4.16. SW7 Settings

SW7	Setting	Description
SW7-1	OFF	P24_4 is used as CAN_TX.
SW7-2	ON	
SW7-3	OFF	P24_3 is used as CAN_RX.
SW7-4	ON	
SW7-5	OFF	P23_5 is used as ESC_LINKACT2.
SW7-6	ON	
SW7-7	OFF	Use VUBUSIN for USB_Function.
SW7-8	ON	
SW7-9	OFF	P00_0 is used as USB_HF_VBUSEN.
SW7-10	ON	

Table 4.17. SW8 Settings

SW8	Setting	Description
SW8-1	ON	P18_1 is used as ESC_LED_ERR.
SW8-2	OFF	
SW8-3	ON	P18_0 is used as ESC_LED_RUN.
SW8-4	OFF	
SW8-5	ON	P16_3 is used as RXD5 of USB-to-serial conversion.
SW8-6	OFF	
SW8-7	ON	P16_4 is used as TXD5 of USB-to-serial conversion.
SW8-8	OFF	
SW8-9	ON	P23_1 is used as USER_LED0.
SW8-10	OFF	

Table 4.18. SW15 Settings

SW15	Setting	Description
SW15-1	ON	PCIe functions is used as Root Complex.
SW15-2	ON	PCIe L1 is used as a root complex.
SW15-3	OFF	The PCIe function is used in a configuration of 2 lanes × 1 port.
SW15-4	OFF	(At the time of shipment = OFF)
SW15-5	OFF	The 12-V power supply of the PCIe x4 connector CN32 is OFF.
SW15-6	OFF	The 3.3-V power supply of the PCIe x4 connector CN32 is OFF.
SW15-7	OFF	The 3.3-V power supply of the PCIe x1 connector CN8 is OFF.
SW15-8	OFF	The 12-V power supply of the PCIe x1 connector CN8 is OFF.

Table 4.19. SW17 Settings

SW17	Setting	Description
SW17-1	ON	AN000 is connected to potentiometer.
SW17-2	OFF	

Table 4.20. SW18 Settings

SW18	Setting	Description
SW18-1	OFF	AN100 is connected to MikroBUS.
SW18-2	ON	
SW18-3	OFF	AN101 is connected to Grove2.
SW18-4	ON	
SW18-5	OFF	AN102 is connected to Grove2.
SW18-6	ON	

4.3.1 Board Setup

Figure 4.6 に OPC UA のサンプルソフト実行時の接続図を示します。RZ/T2H 評価ボードに Ethernet ケーブル、デバッグ、15V DC の各ケーブルを接続してください。

ETH0~ETH1 のいずれの Ethernet コネクタに Ethernet ケーブルを接続しても構いません。評価ボード上のデバッガ J-Link OB を使用する場合は CN62 をオープンとし、USB Micro のケーブルを接続します。I-jet を使用する場合は、CN62 をショートとし、CN61 に I-jet を接続します。

本サンプルソフトでは、SNTP による時刻同期を行っているため、NTP Server に接続する必要があります。動作確認時には、Windows PC を NTP Server として使用しています。

OPC UA のセキュリティ機能で使用する鍵ファイルを FreeRTOS+FAT で管理する場合、評価ボードへの USB フラッシュドライブの接続が必要になります。

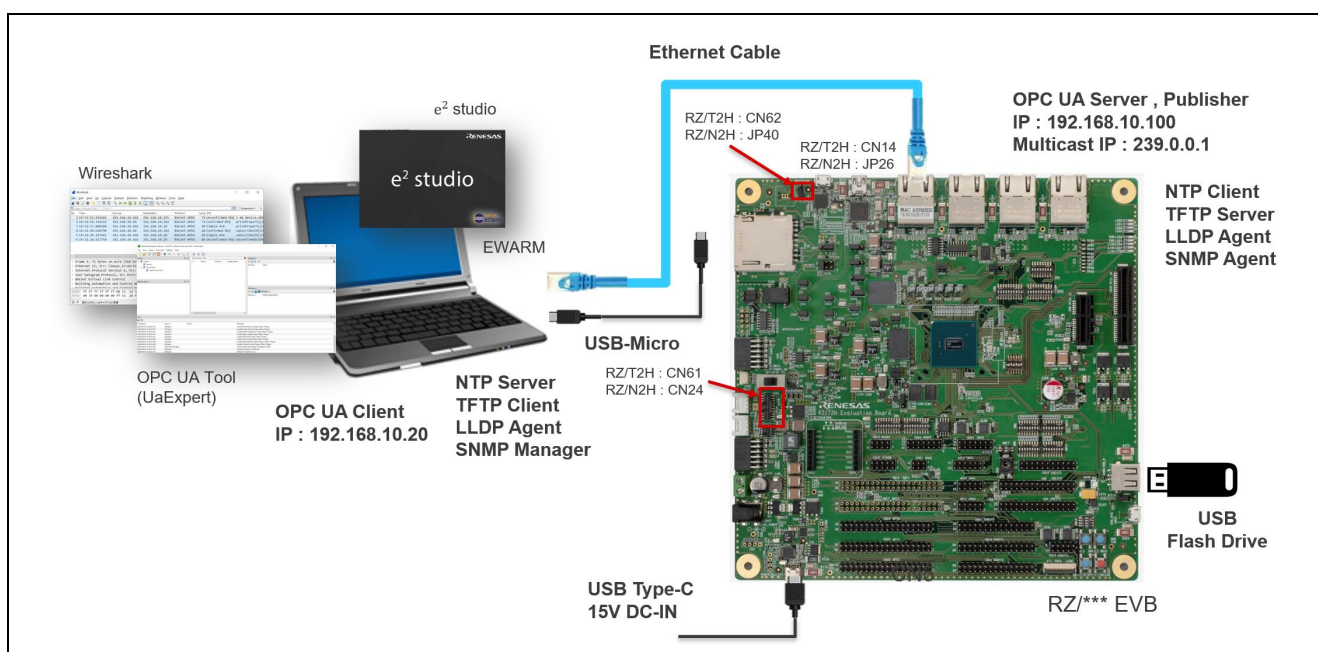


Figure 4.6. RZ/T2H EVB OPC UA Server hardware connection diagram

4.4 RZ/N2H Evaluation Board

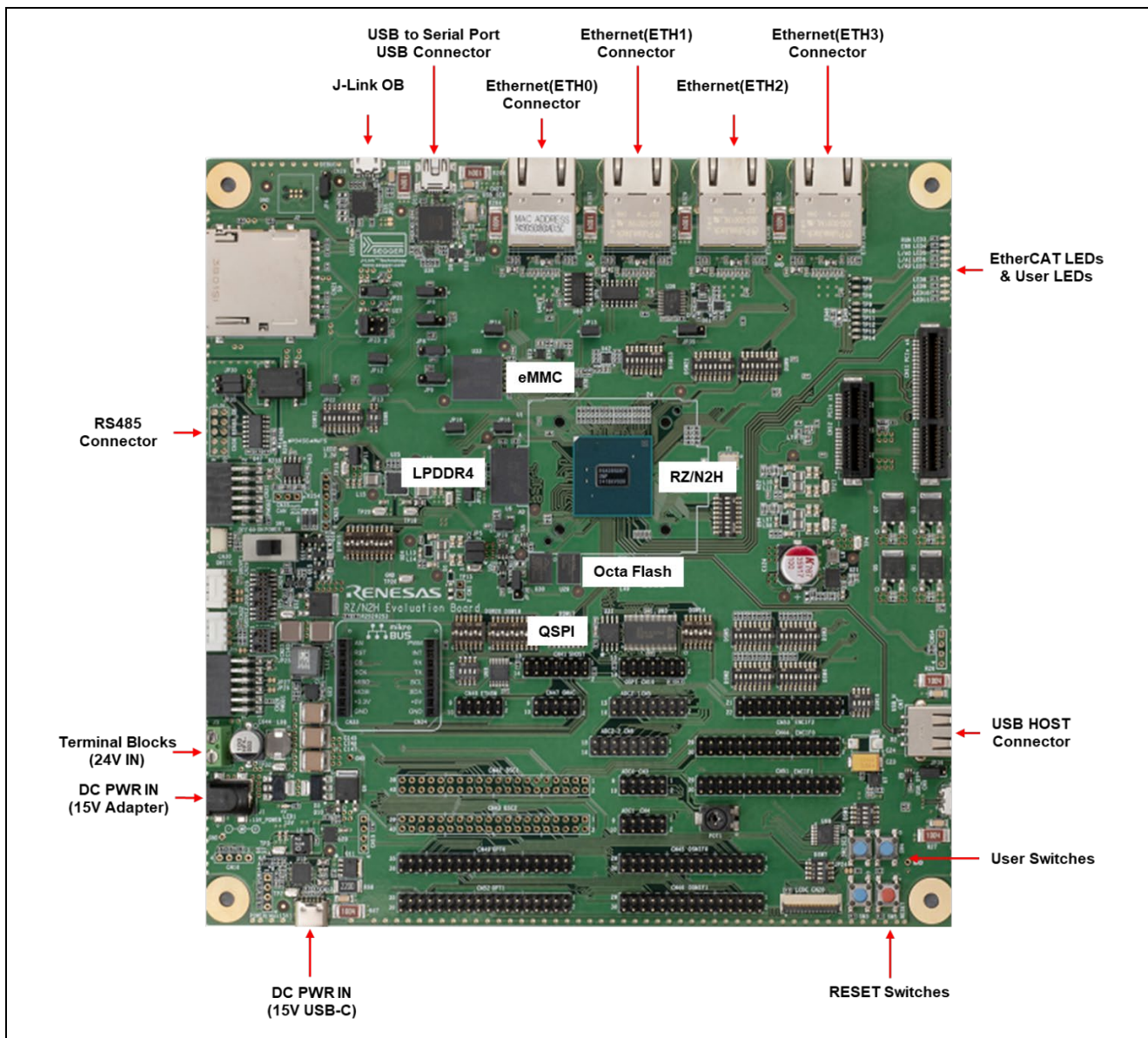


Figure 4.7. RZN2H Evaluation board layout

表中の赤文字は評価ボードの工場出荷時の設定から変更が必要な箇所を表しています。

Table 4.21. Jumper and Switch Configuration

Reference	Jumper Position	Description
JP6	Short 1-2	1.8-V power is supplied to VCC1833_0. (for Ethernet Port 0)
	Short 2-3	3.3-V power is supplied to VCC1833_0. (for Ethernet Port 0)
JP7	Short 1-2	1.8-V power is supplied to VCC1833_1. (for Ethernet Port 1)
	Short 2-3	3.3-V power is supplied to VCC1833_1. (for Ethernet Port 1)
JP8	Short 1-2	1.8-V power is supplied to VCC1833_1. (for Ethernet Port 2)
	Short 2-3	3.3-V power is supplied to VCC1833_1. (for Ethernet Port 2)
JP9	Short 1-2	1.8-V power is supplied to VCC1833_1. (for Ethernet Port 3)
	Short 2-3	3.3-V power is supplied to VCC1833_1. (for Ethernet Port 3)
JP21	Short 1-2	3.3-V power is supplied to VCC1833_7. (for SD1)
	Short 2-3	Power control IC output for SD1 is supplied to VCC1833_7. (for SD1)
JP23	Short 1-2	3.3-V power is supplied to VCC1833_6. (for SD0)
	Short 2-3	The output from the power-supply control IC for SD0 is supplied to VCC1833_6 (for SD0).
	Short 5-6	1.8-V power is supplied to VCC1833_6. (for SD0)
JP35	Short 1-2	VCC1833_0 is supplied to VCC_ETH2_MDIO (DSW5-6 is ON: When P21_4 and P21_5 are selected for MDIO).
	Short 2-3	VCC1833_2 is supplied to VCC_ETH2_MDIO. (DSW5-6 is OFF: When P30_5 and P30_6 are selected for MDIO)
JP40	Open	The on-board debugging function J-Link® OB is enabled.
	Short	The on-board debugging function J-Link™ OB is disabled. Connect an external emulator to CN23 or CN24 at debugging.
JP30	Short 1-2	Full-duplex communication
JP31	Short 2-3	Half-duplex communication

Table 4.22. SW3 Settings

SW3	Setting	Description
SW3-1	ON	xSPI1 boot mode (x1 boot serial flash)
SW3-2	OFF	
SW3-3	ON	
SW3-4	OFF	The number of ATCM wait cycles of CPU0 in the Cortex-R52 is 1 wait cycle.
SW3-5	OFF	The number of ATCM wait cycles of CPU1 in the Cortex-R52 is 1 wait cycle.
SW3-6	OFF	The power-supply voltage of the boot peripheral is 3.3 V.
SW3-7	ON	JTAG mode = Normal mode
SW3-8	OFF	(At the time of shipment = OFF)

Table 4.23. SW2 Settings

SW2	Setting	Description
SW2-1	ON	XTALSEL = 'L' Select oscillator for RZ/N2H clock input.
SW2-2	OFF	(At the time of shipment = OFF)
SW2-3	ON	P27_2, P27_3, P27_6, and P31_3 are used as inputs to user DIP switches.
SW2-4	OFF	P13_4, P13_5, and P14_0 are used as RXD3, TXD3, and DE3 of the RS485.
SW2-5	ON	P00_0 to P00_2 are used as control signals for the USB power-supply IC. In this case, set DSW14-1, DSW14-3, and DSW14-5 to ON and DSW14-2, DSW14-4, and DSW14-6 to OFF.
SW2-6	ON	P01_0, P01_2, P01_4 to P01_7, and P02_0 to P02_3 are used as XSPI1 signals.
SW2-7	OFF	(At the time of shipment = OFF)
SW2-8	OFF	(At the time of shipment = OFF)

Table 4.24. SW5 Settings

SW5	Setting	Description
SW5-1	ON	P12_0 to 7, P13_0 to 2 are connected to eMMC.
SW5-2	ON	
SW5-3	ON	P08_6 and P17_4 are used as SD1 control signal.
SW5-4	OFF	(At the time of shipment = OFF)
SW5-5	OFF	(At the time of shipment = OFF)
SW5-6	ON	MDC and MDIO of Ethernet port 2 are connected to GMAC0 (P21_4 and P21_5).
SW5-7	ON	P29_1 to P29_7, P30_0 to P30_4, P30_7, P31_2, P31_4, and P31_5 are used as control signals for Ethernet Port 2.
SW5-8	OFF	P00_0 to P00_2 are used as control signals for USB interface.

Table 4.25. SW8 Settings

SW8	Setting	Description
SW8-1	ON	P03_1_GMAC_RESETOUT2# is used as RESET for Ethernet Port 2.
SW8-2	OFF	

Table 4.26. SW12 Settings

SW12	Setting	Description
SW12-1	OFF	P00_3 is used as P00_3_ETH3_COL of Ethernet Port 3.
SW12-2	ON	
SW12-3	OFF	P11_0 is used as P11_0_ESC_RESETOUT# of Ethernet Port 0 and 1.
SW12-4	ON	
SW12-5	OFF	P03_2 is used as P03_2_GMAC_RESETOUT3# of Ethernet Port 3.
SW12-6	ON	
SW12-7	OFF	P03_1 is used as P03_1_GMAC_RESETOUT2# of Ethernet Port 2. In this case, set DSW8-1 to ON and DSW8-2 to OFF.
SW12-8	ON	

Table 4.27. SW13 Settings

SW13	Setting	Description
SW13-1	ON	P26_7 is used as P26_7_ETH1_RXER of Ethernet Port 1.
SW13-2	OFF	
SW13-3	ON	P27_0 is used as P27_0_ETH1_CRIS of Ethernet Port 1.
SW13-4	OFF	
SW13-5	ON	P27_1 is used as P27_1_ETH1_COL of Ethernet Port 1.
SW13-6	OFF	
SW13-7	OFF	P13_7 is used as MDINT of Ethernet Port 2.
SW13-8	ON	

Table 4.28. SW15 Settings

SW15	Setting	Description
SW15-1	OFF	P22_6 is used as P22_6_SD0_WP.
SW15-2	ON	
SW15-3	OFF	P22_5 is used as P22_5_SD0_CD.
SW15-4	ON	
SW15-5	OFF	P14_7 is used as P14_7_USER_LED1(LED9).
SW15-6	ON	
SW15-7	OFF	- (At the time of shipment = OFF)
SW15-8	OFF	P14_6 is used as P14_6_USER_LED0(LED8).
SW15-9	OFF	
SW15-10	ON	

Table 4.29. SW17 Settings

SW17	Setting	Description
SW17-1	OFF	P03_0 is used as USER_LED3(LED11).
SW17-2	ON	
SW17-3	OFF	P02_7 is used as USER_LED2(LED10).
SW17-4	ON	
SW17-5	OFF	P02_6 is used as P02_6_SD0_IOVS.
SW17-6	ON	
SW17-7	OFF	- (At the time of shipment = OFF)
SW17-8	ON	P02_5 is used as P02_5_SD0_PWEN.

Table 4.30. SW18 Settings

SW18	Setting	Description
SW18-1	ON	P22_7 is used as ESC_LINKACT0(LED5).
SW18-2	OFF	
SW18-3	ON	P23_0 is used as ESC_LINKACT1(LED6).
SW18-4	OFF	
SW18-5	ON	P14_3 is used as ESC_LINKACT2(LED7).
SW18-6	OFF	
SW18-7	ON	P31_6 is used as ESC_LED RUN(LED3).
SW18-8	OFF	
SW18-9	ON	P18_1 is used as ESC_LEDERR(LED4).
SW18-10	OFF	

4.4.1 Board Setup

Figure 4.8に OPC UA のサンプルソフト実行時の接続図を示します。RZ/N2H 評価ボードに Ethernet ケーブル、デバッガ、15V DC の各ケーブルを接続してください。

ETH0~ETH1 のいずれの Ethernet コネクタに Ethernet ケーブルを接続しても構いません。評価ボード上のデバッガ J-Link OB を使用する場合は JP40 をオープンとし、USB Micro のケーブルを接続します。I-jet を使用する場合は、JP40 をショートとし、CN24 に I-jet を接続します。

本サンプルソフトでは、SNTP による時刻同期を行っているため、NTP Server に接続する必要があります。動作確認時には、Windows PC を NTP Server として使用しています。

OPC UA のセキュリティ機能で使用する鍵ファイルを FreeRTOS+FAT で管理する場合、評価ボードへの USB フラッシュドライブの接続が必要になります。

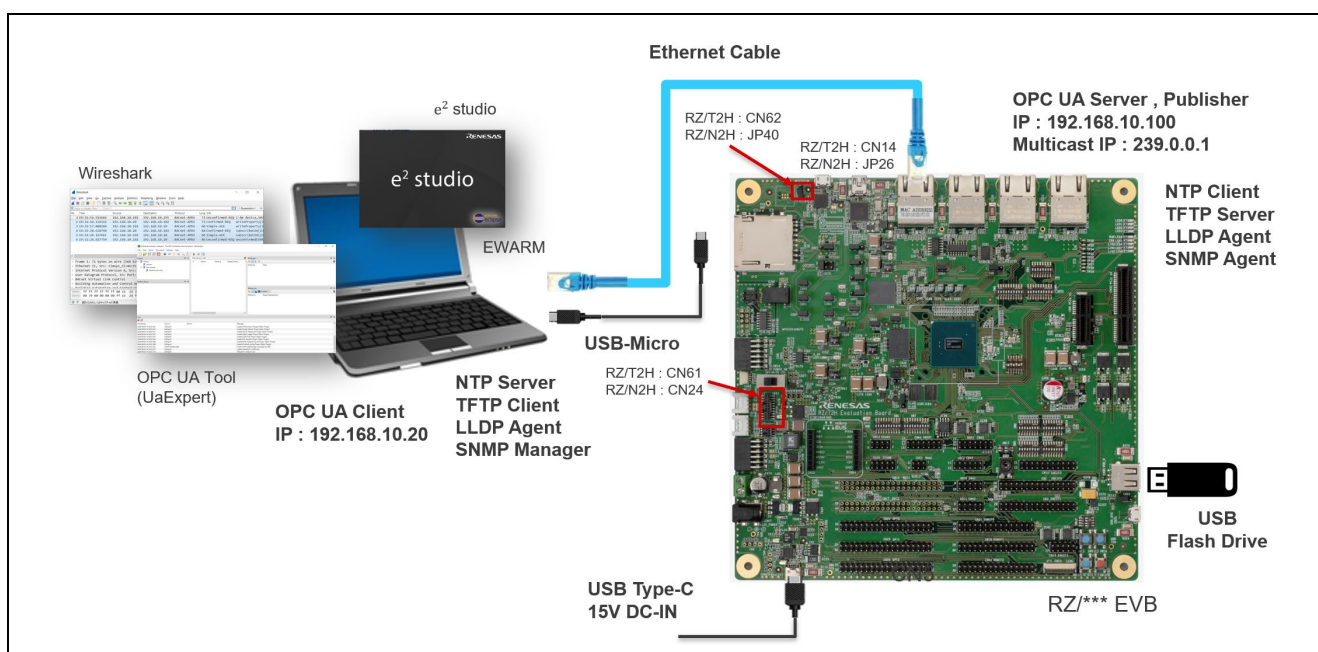


Figure 4.8. RZ/N2H EVB OPC UA Server hardware connection diagram

5. Set up the Host

5.1 Host IP Address

OPC UA Client となる PC 側イーサネットのアドレス設定を行います。コントロールパネルを開きます。以下のとおりに進み、IP アドレスを設定してください。

コントロールパネル>ネットワークとインターネット>ネットワークと共有センター>アダプターの設定の変更>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

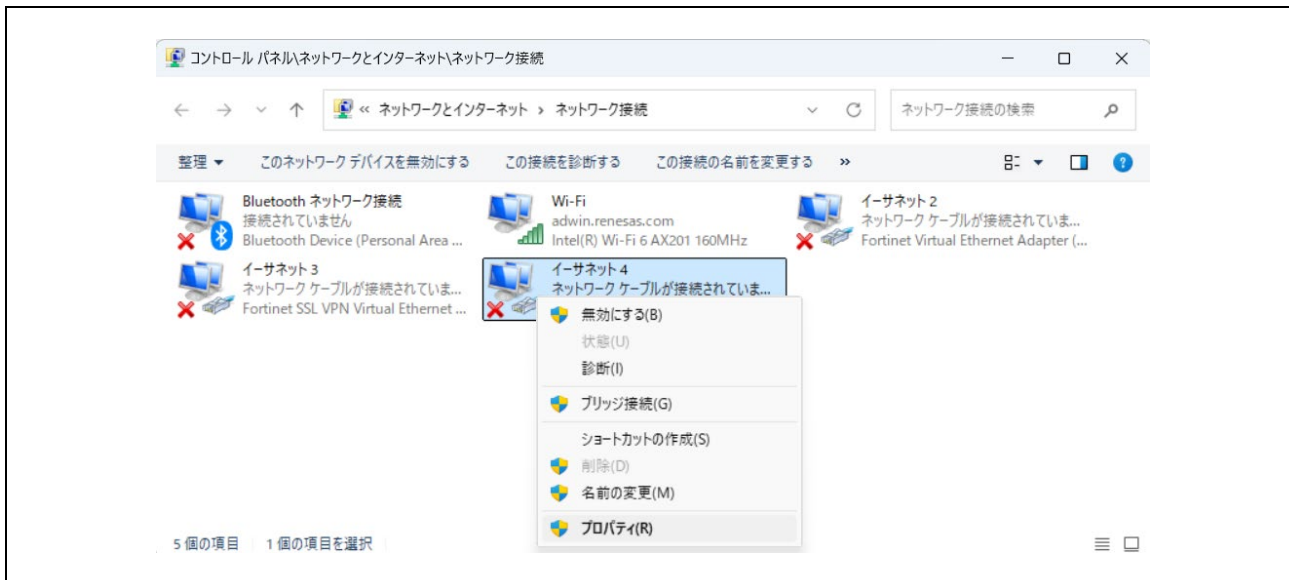


Figure 5.1. Network connection

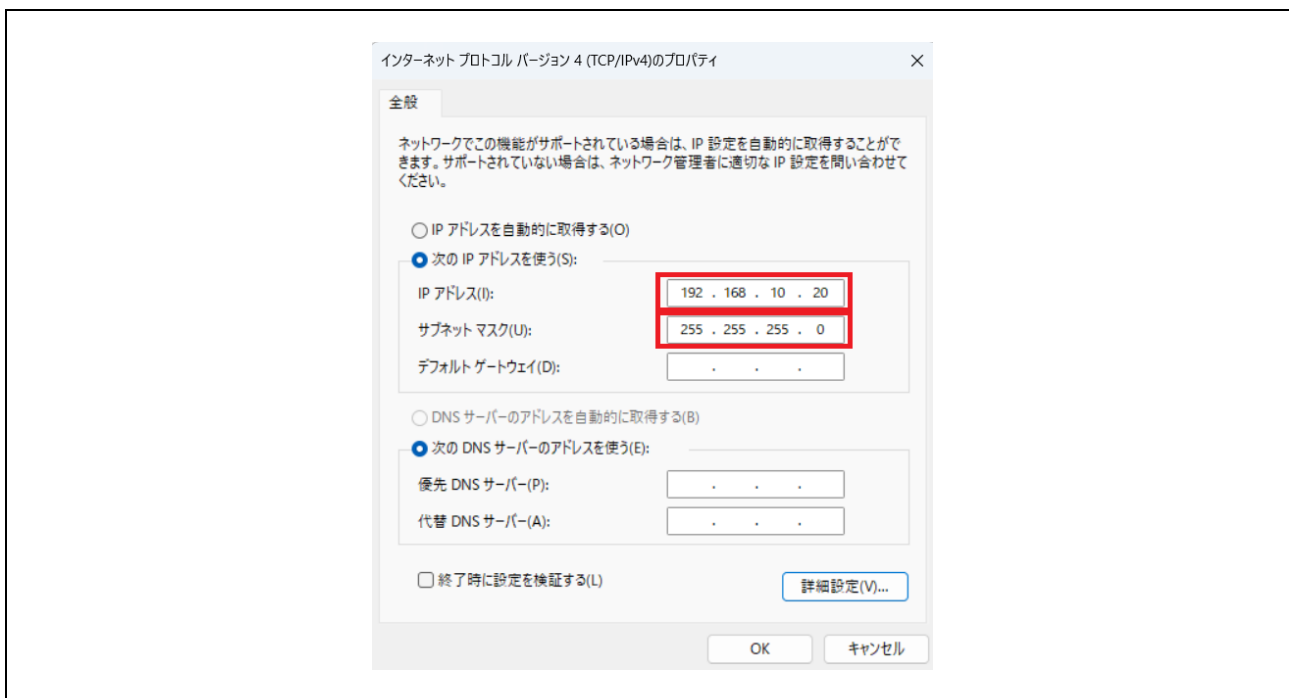


Figure 5.2. TCP/IPv4 properties

OPC UA Server サンプルソフトで使用する評価ボードの IP アドレスは 192.168.10.100 です。PC 側の IP アドレスを 192.168.10.** に設定する必要があります。本サンプルソフトでは、デフォルトとして 192.168.10.20 を NTP サーバの IP アドレスに設定しています。そのため、接続する Windows PC を NTP サーバとする場合は、PC 側の IP アドレスを 192.168.10.20 に設定してください。

5.2 Installation of EWARM environment

※ e² studio 環境での開発を行う場合、本章の作業を行う必要はありません。

- 1) 3章に記載のバージョンの EWARM インストーラファイルを実行してください。
- 2) “Install IAR Embedded Workbench® for Arm”をクリックし、指示に従ってインストールしてください。
- 3) 3章に記載のバージョンの FSP Smart Configurator インストーラファイルをダウンロードしてください。
- 4) インストーラを実行して、指示に従ってインストールしてください。

5.3 Installation of e² studio environment

※ EWARM 環境での開発を行う場合、本章の作業を行う必要はありません。

- 1) 3章に記載のバージョンの e² studio インストーラファイルをダウンロードしてください。
- 2) インストーラを実行して、指示に従ってインストールしてください。

5.4 UaExpert

UaExpertはOPC UA Client ツールです。ここでは、OPC UA Server と接続してオブジェクトノードにアクセスするために使います。

Table 3.5 に記載のバージョンを 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。ダウンロード前に Unified Automation Web サイトに登録してアカウントをアクティブ化する必要があります。すべてのコンテンツは無料で提供されますが、このページからソフトウェアをダウンロードまたはインストールすると、Unified Automation ソフトウェア使用許諾契約 (SLA) に自動的に同意したことになります。ソフトウェアおよび情報のライセンス条件は、次のリンクを参照ください。

<https://www.unified-automation.com/products/sdk-overview/licenses.html#c341>

上記使用条件を確認の上、お使いください。

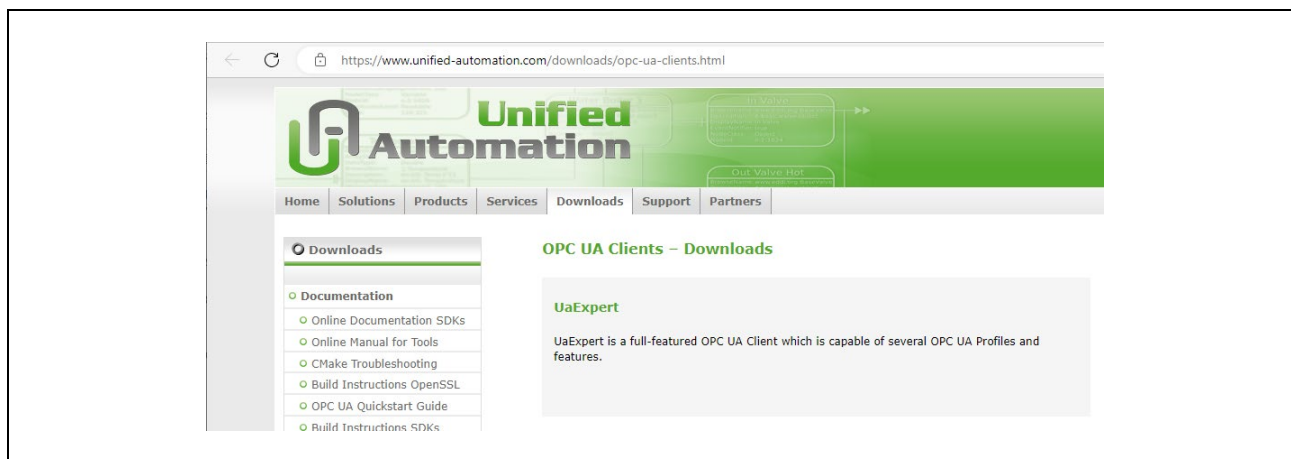


Figure 5.3. UaExpert

5.5 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 3.5 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。

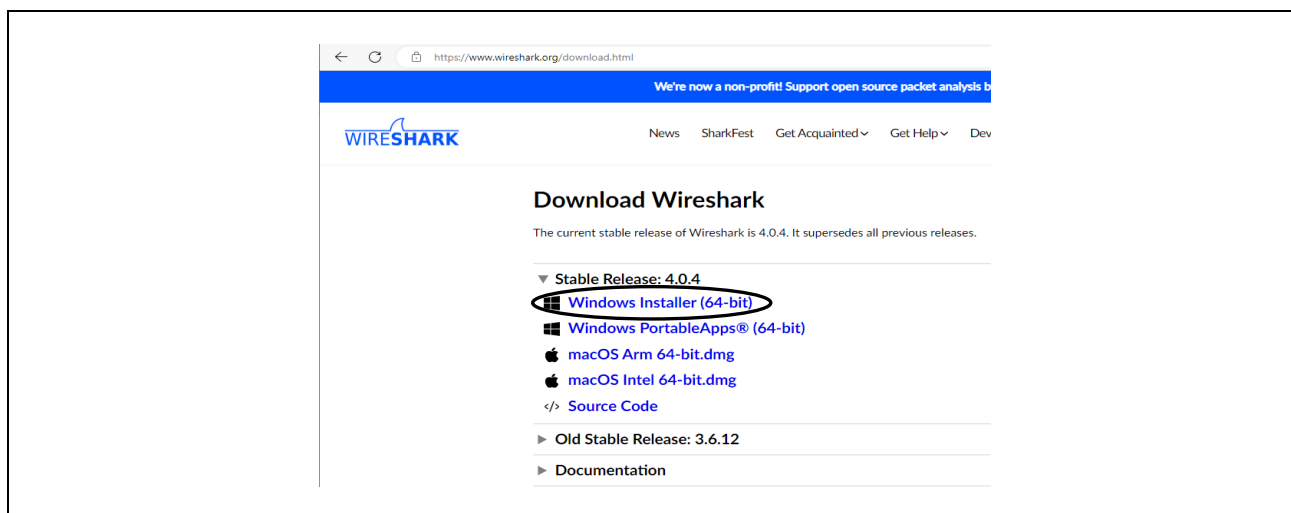


Figure 5.4. Download Wireshark

5.6 OpcCmd Utility

OpcCmd Utility はコマンドラインから使用できる、汎用の OPC 通信ツールです。本サンプルソフトでは、OPC UA PubSub の Subscriber として使用しています。Table 3.5 のリンクからダウンロードして、任意のフォルダに展開してください。インストールは不要です。ライセンス条件は Zero-Clause BSD (0BSD) です。本ツールの使い方の詳細は、下記リンクをご参照ください。

[Using OpcCmd Utility as OPC UA PubSub Subscriber - OPC Labs Knowledge Base](#)

OpcCmd Utility Download

[Page](#)
[Discussion](#)

[View source](#)
[History](#)

OpcCmd Utility

The OpcCmd utility (console application) is a program that allows performing various OPC operations from the command line. It can act as a generic OPC DA Client, OPC A&E Client, OPC UA Client, OPC UA PubSub subscriber, OPC UA Server, and be used for evaluation, experiments, and testing. [More information...](#)

Target platform and requirements	Version	Binaries	ClickOnce
Windows, Linux or macOS .NET 9.0	Current or Preview ^[1]	ZIP file: Download TGZ file: Download	
Windows .NET Framework 4.7.2 (or later)	Current or Preview ^[1]	ZIP file: Download EXE file: Download ^[2]	Download & launch Install page ^[3]

Figure 5.5. Download OpcCmd Utility

5.7 OPC UA Demo Publisher

OPC UA Demo Publisher はコマンドラインから使用できる、OPC UA Publisher ツールです。Table 3.5 のリンクからダウンロードして、任意のフォルダに展開してください。インストールは不要です。ライセンス条件は Zero-Clause BSD (0BSD) です。本ツールの使い方の詳細は、下記リンクをご参照ください。

[UADemoPublisher Basics - OPC Labs Knowledge Base](#)

OPC UA Demo Publisher Download

[Page](#)
[Discussion](#)

[View source](#)
[History](#)

The UADemoPublisher program (console application) is a demonstration publisher for OPC UA PubSub. [More information...](#)

Target platform and requirements	Version	Binaries	ClickOnce
Windows, Linux or macOS .NET 9.0	Current or Preview ^[1]	ZIP file: Download TGZ file: Download	
Windows .NET Framework 4.7.2 (or later)	Current or Preview ^[1]	ZIP file: Download EXE file: Download ^[2]	Download & launch Install page ^[3]

Figure 5.6. Download OPC UA Demo Publisher

5.8 Node-RED

Node-RED とは、多種多様なノードを結び付けてフローを作成し、シングルクリックで実行環境にデプロイすることができる、ブラウザベースのプログラミングツールです。本プロジェクトでは、OPC UA Node の時間変化をモニタリングするために使用します。

Table 3.5 に記載のリンクから、Node.js をインストールします。インストール後に、コマンドプロンプトを管理者権限で立ち上げ、下記コマンドを入力します。

```
npm install -g --unsafe-perm node-red
```

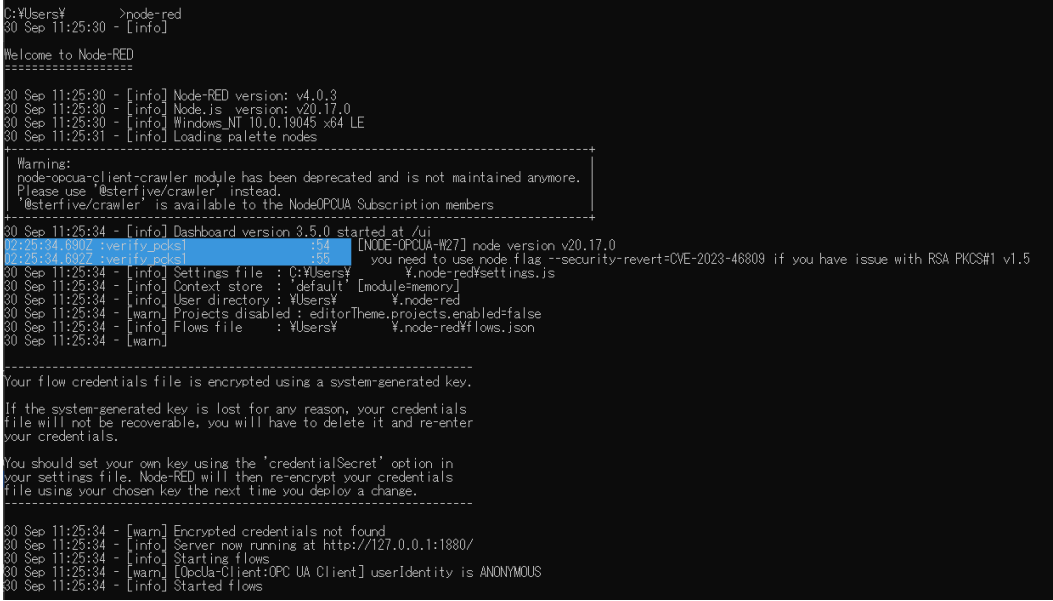
```
npm install node-red-contrib-opcua
```

```
npm i node-red-dashboard
```

再度、コマンドプロンプトを管理者権限なしで立ち上げ、下記コマンドを入力します。

```
node-red
```

Node-RED が正常に立ち上がった場合、Figure 5.7 のような状態になります。Node-RED を使用している間は、このコマンドプロンプトは閉じないでください。



```
C:\Users\Y >node-red
30 Sep 11:25:30 - [info]
Welcome to Node-RED
=====
30 Sep 11:25:30 - [info] Node-RED version: v4.0.3
30 Sep 11:25:30 - [info] Node.js version: v20.17.0
30 Sep 11:25:30 - [info] Windows_NT 10.0.19045 x64 LE
30 Sep 11:25:31 - [info] Loading palette nodes

-----
Warning:
node-opcua-client-crawler module has been deprecated and is not maintained anymore.
Please use "@sterfive/crawler" instead.
@sterfive/crawler is available to the NodeOPCUA Subscription members
-----
30 Sep 11:25:34 - [info] Dashboard version 3.5.0 started at /ui
02:25:34.890Z :verify.pcks! :84 [NODE-OPCUA-W27] node version v20.17.0
02:25:34.890Z :verify.pcks! :85 you need to use node flag --security-revert=CVE-2023-46809 if you have issue with RSA PKCS#1 v1.5
30 Sep 11:25:34 - [info] Settings file : C:\Users\Y %node-red%settings.js
30 Sep 11:25:34 - [info] Context store : default [module=memory]
30 Sep 11:25:34 - [info] User directory : %Users\ %node-red
30 Sep 11:25:34 - [warn] Projects disabled : editorTheme.projects.enabled=false
30 Sep 11:25:34 - [info] Flows file : %Users\ %node-red%flows.json
30 Sep 11:25:34 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
30 Sep 11:25:34 - [warn] Encrypted credentials not found
30 Sep 11:25:34 - [info] Server now running at http://127.0.0.1:1880/
30 Sep 11:25:34 - [info] Starting flows
30 Sep 11:25:34 - [warn] [OpCUA-Client:OPC UA Client] userIdentity is ANONYMOUS
30 Sep 11:25:34 - [info] Started flows
```

Figure 5.7. Run Node-RED

ブラウザに以下の URL を入力し、Figure 5.8 のような画面が表示されれば正常に立ち上げができています。

<http://localhost:1880>

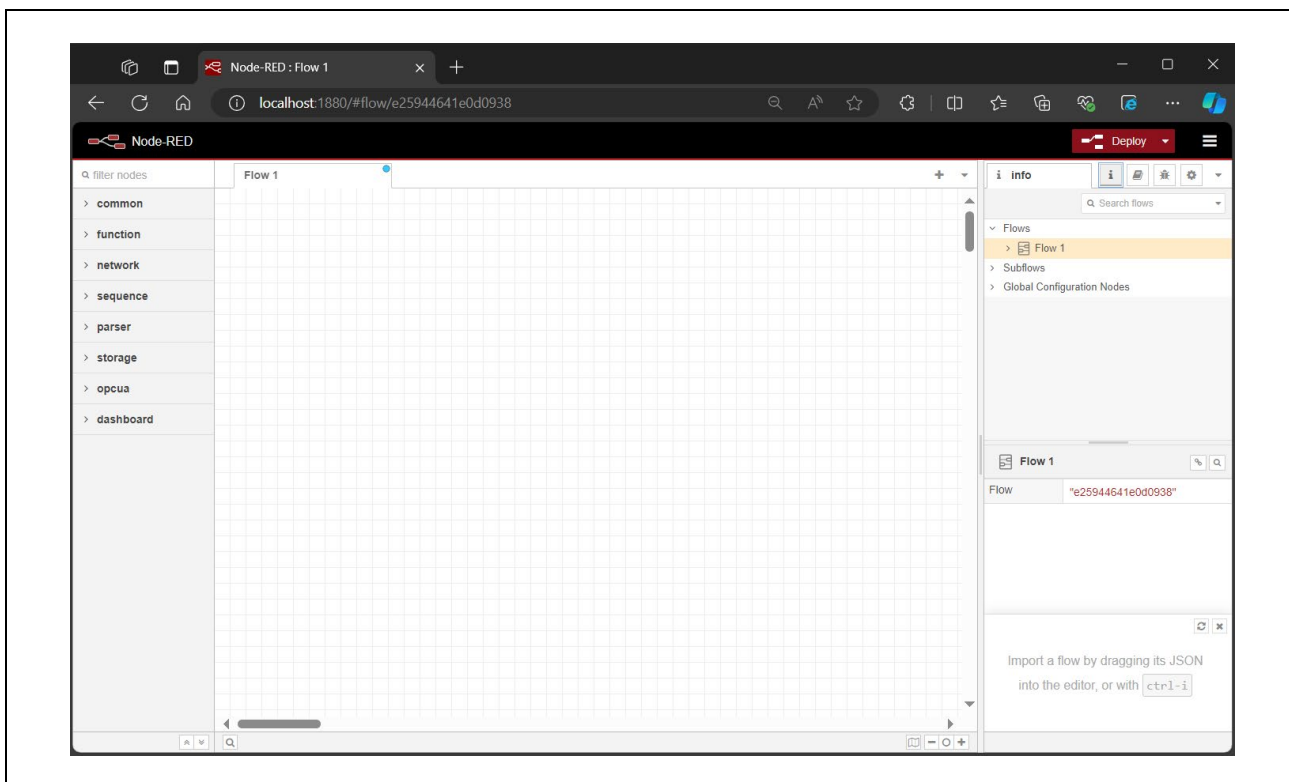


Figure 5.8. Start Node-RED

6. Running the Sample Application

この章では、プログラムのダウンロード方法と実行方法について説明します。

アーカイブされた本サンプルソフトのパッケージを解凍し、任意のフォルダに格納します。フォルダ階層が深くてフルパスが長すぎるとエラーの原因になりますので、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。

Dual Core 構成のプロジェクトでは、評価ボードの黄 LED を点滅させるプロジェクト (RZ***_EVB(RSK)_OPC_*****_Primary_0) と、OPC UA が動作するプロジェクト (RZ***_EVB(RSK)_OPC_*****_Secondary_1) が存在しています。以降、それぞれプライマリプロジェクト、セカンダリプロジェクトと表記します。

6.1 When using EWARM

EWARM を起動し、File タブの "Open Workspace..." をクリックします。

CR52_single プロジェクト : RZ***_OPC_EVB(RSK)_rev****\project\CR52_single\ewarm\

CR52_dual プロジェクト : RZ***_OPC_EVB(RSK)_rev****\project\CR52_dual\ewarm\

CA55_dual プロジェクト : RZ***_OPC_EVB(RSK)_rev****\project\CA55_dual\ewarm\

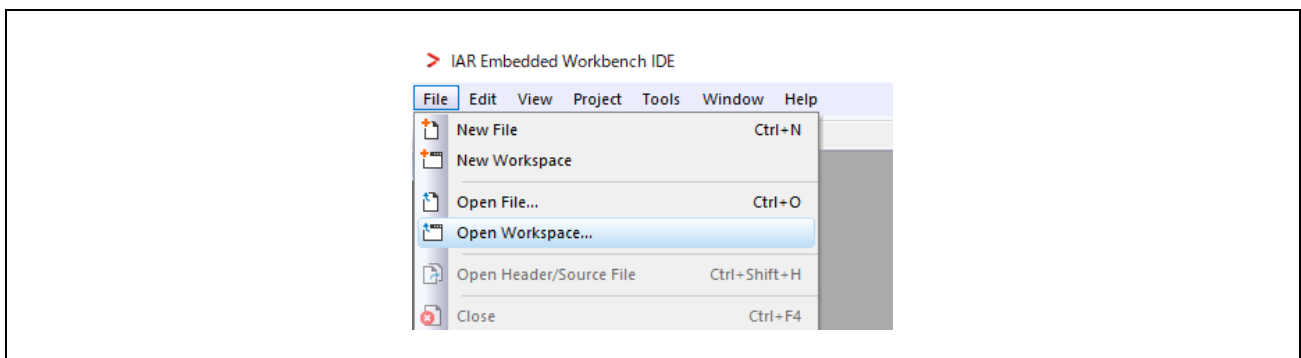


Figure 6.1. Open Workspace

ワークスペースファイル(.eww)を選択し、Open ボタンをクリックします。

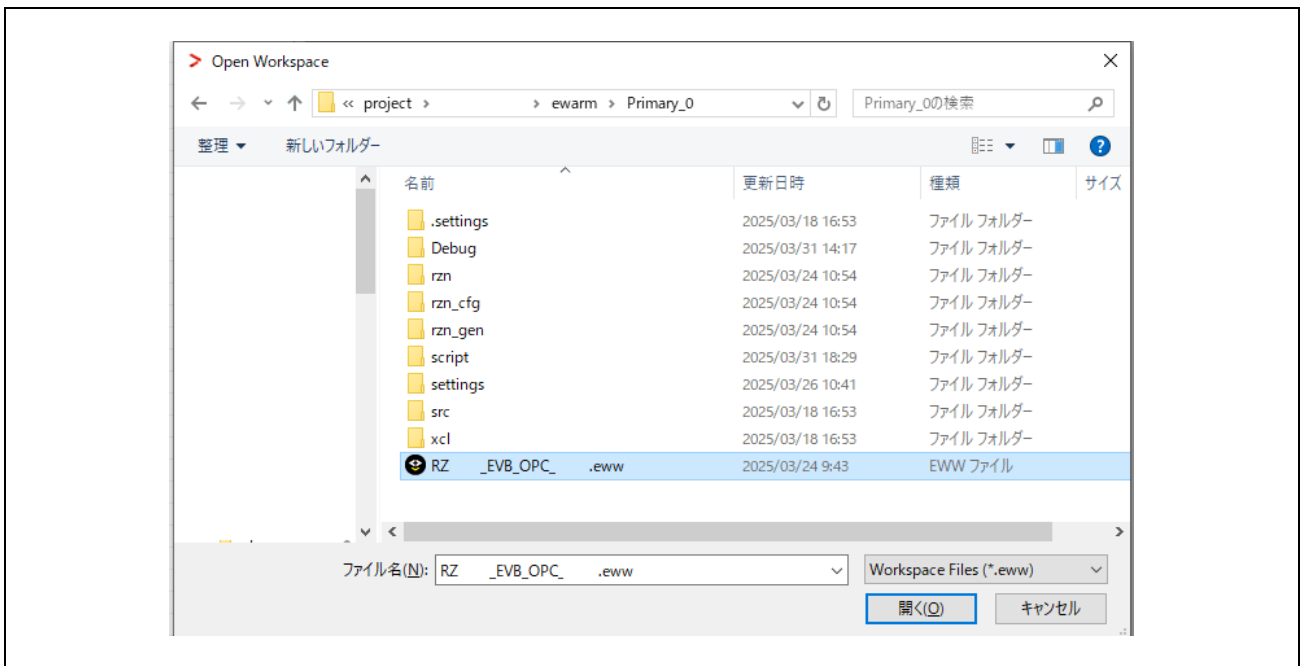


Figure 6.2. Open project file

以降の手順は使用するプロジェクトによって異なります。

6.1.1 For Dual core project

6.1.1.1 Build

EWARMにてプライマリプロジェクトを開き、Tools タブの“FSP Smart Configurator”をクリックします。このボタンが表示されていない場合、もしくはクリック後にエラーが表示される場合は 9.5 章を参照してください。

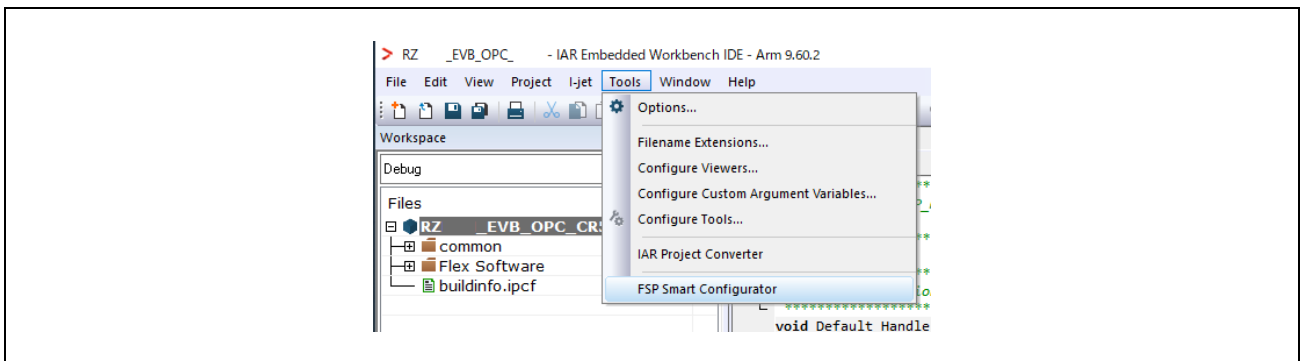


Figure 6.3. Tools tag

FSP Smart Configurator 上の“Generate Project Content”ボタンをクリックするとコードが生成されます。

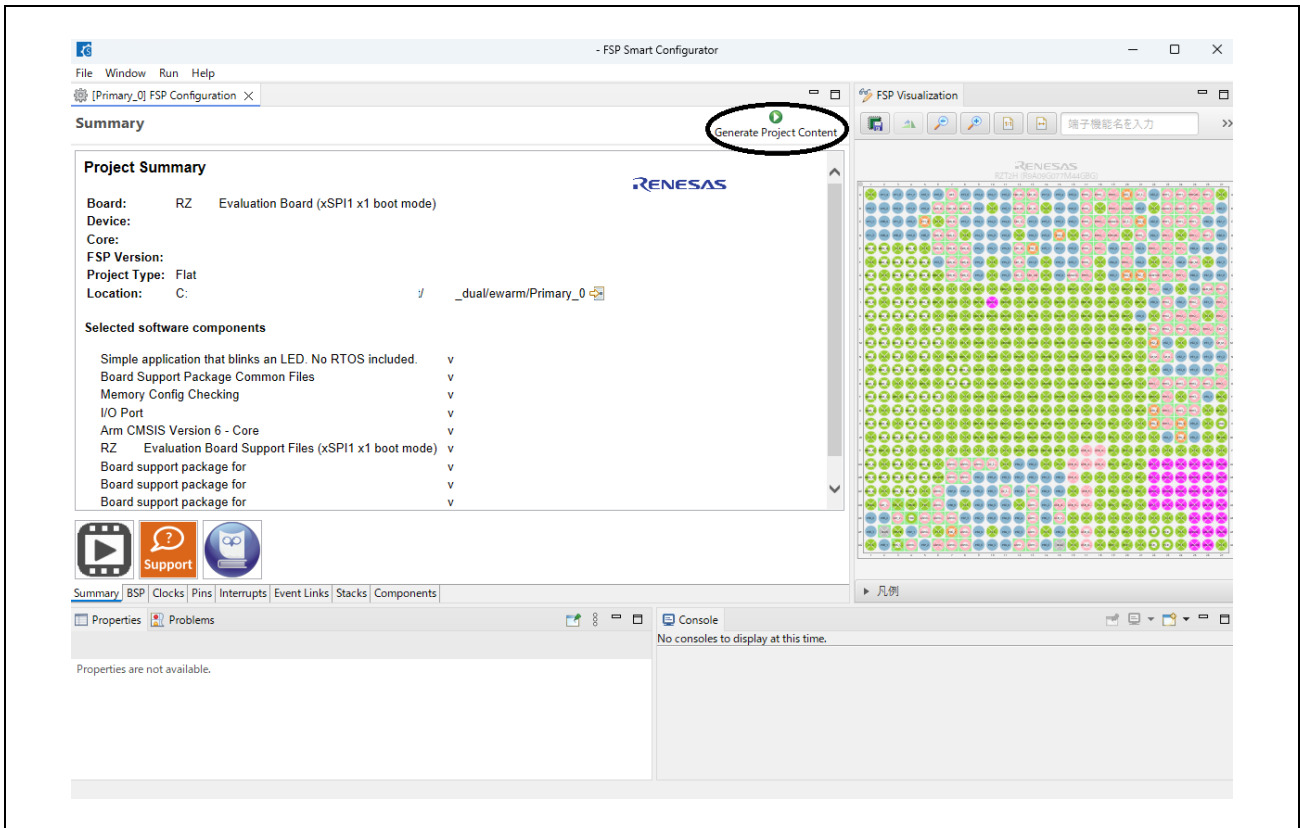


Figure 6.4. Generate Project Content

デバッガを使用してデバッグを行う場合は、下記の箇所にコアに応じたループ処理を追加してください。

“(Primary Project Location)\src\hal_entry.c “

```

BSP_ATTRIBUTE_STACKLESS void R_BSP_WarmStart_StackLess(void)
{
  /* The very beginning of the startup process. */
  #if 1 // Software loops are only needed when debugging.
    __asm volatile (
      "  mov  r0, #0      \n"
      "  movw r1, #0x68bf \n"
      "  movt r1, #0x478  \n"
      "software_loop:   \n"
      "  adds r0, #1      \n"
      "  cmp  r0, r1      \n"
      "  bne  software_loop \n"
      ::: "memory");
  #endif
  /* Do not delete. Required to return to system_init. */
  #if (0 == BSP_LP64_SUPPORT)
    __asm volatile ("BX lr");
  #else
    __asm volatile ("BR lr");
  #endif
}

```

RZ/T2H, N2H の場合のみ、buildinfo.ipcf に定義されている device tag name を下記の通りに変更してください。

Table 6.1. FSP SC Newly Created Project Debug Settings

Device name	Primary Core: CR52_0
RZ/T2H	R9A09G077M44_R52_0
RZ/N2H	R9A09G087M44_R52_0

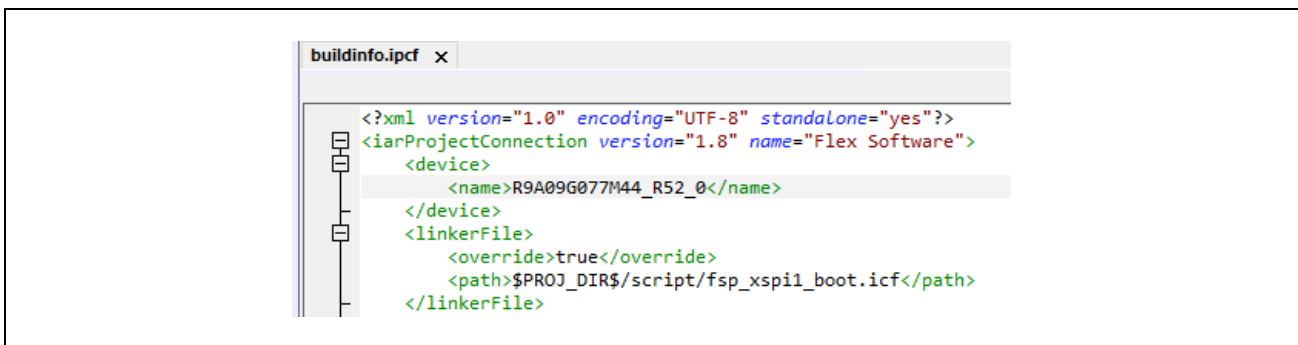


Figure 6.5. IAR EWARM Project File

Make ボタンをクリックしてビルドします。ビルドが完了すると、ビルドメッセージがビルドコンソールウィンドウに表示され、コンパイルターゲットファイルとエラー/警告の数が表示されます。

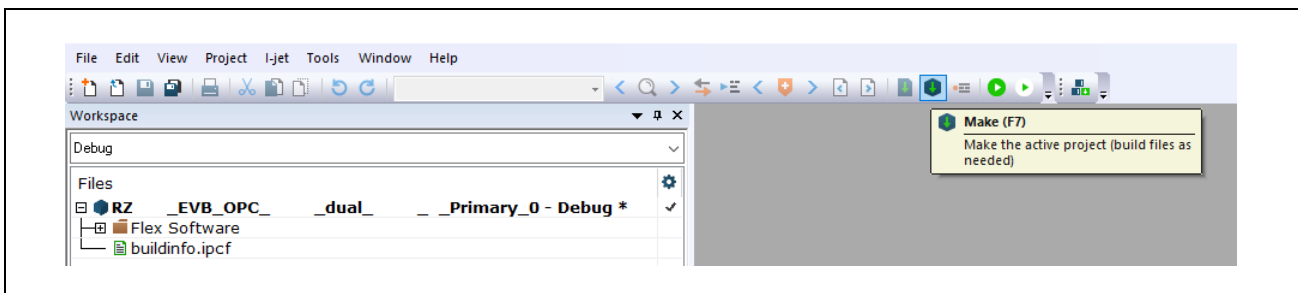


Figure 6.6. Make Primary project

次にセカンダリプロジェクトを開き、Tools タブの”FSP Smart Configurator”をクリックします。

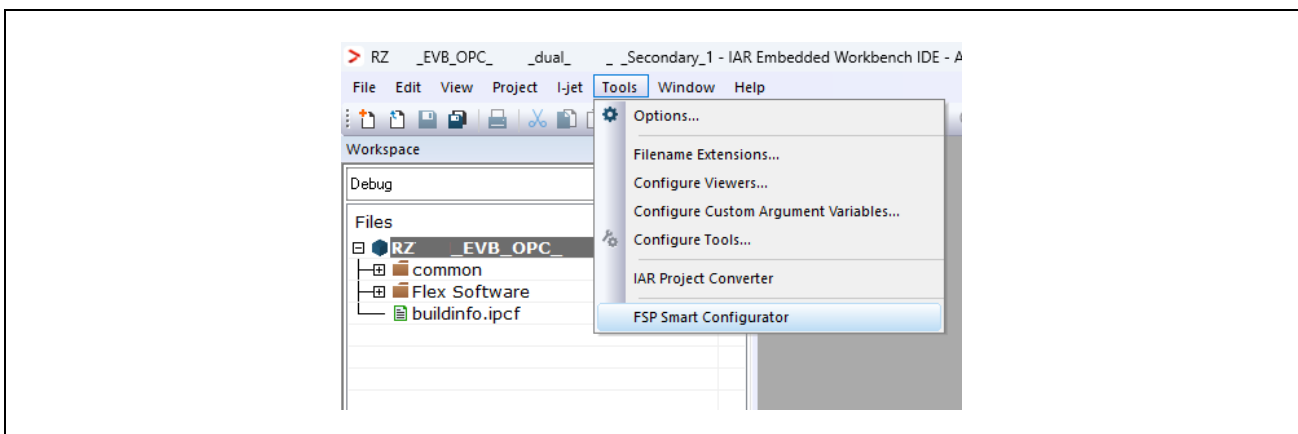


Figure 6.7. Open FSP SC

※サンプルソフトウェアの各種設定をデフォルトから変更したい場合は、このタイミングで、9.6章を参考に設定を行ってください。

FSP Smart Configurator 上の“Generate Project Content”ボタンをクリックするとコードが生成されます。

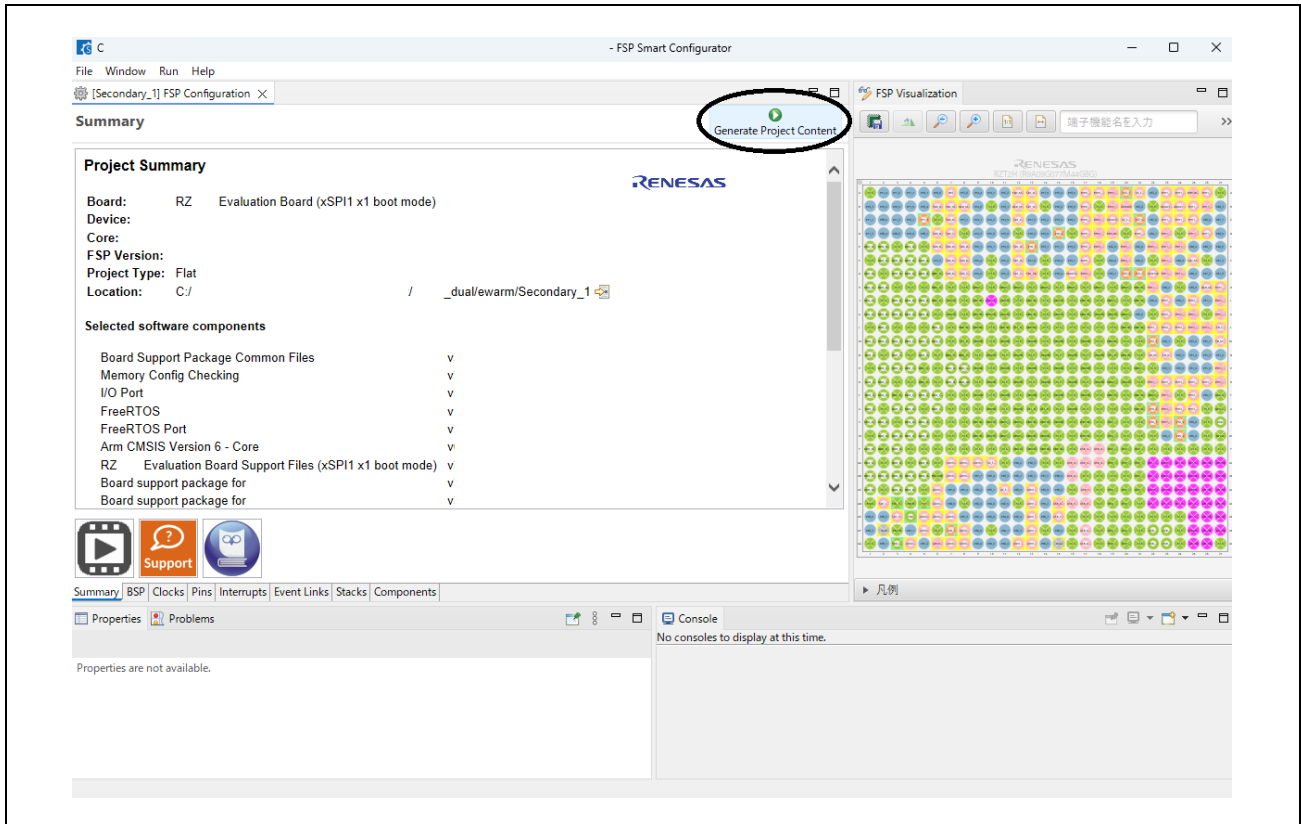


Figure 6.8. Generate Project Content

RZ/T2H, N2H の場合のみ、buildinfo.ipcf に定義されている device tag name を下記の通りに変更してください。

Table 6.2 FSP SC Newly Created Project Debug Settings

Device name	Secondary Core: CR52_1	Secondary Core: CA55_2
RZ/T2H	R9A09G077M44_R52_1	R9A09G077M44_A55
RZ/N2H	R9A09G087M44_R52_1	R9A09G087M44_A55

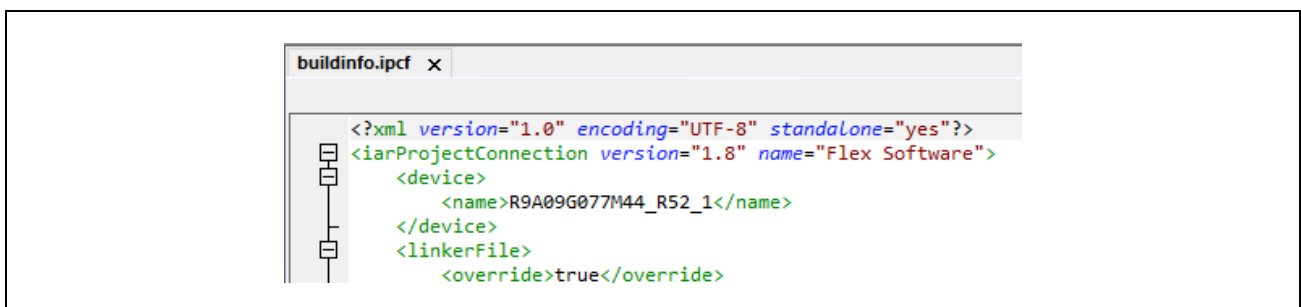


Figure 6.9. IAR EWARM Project File

Make ボタンをクリックしてビルドします。ビルドが完了すると、ビルドメッセージがビルドコンソールウィンドウに表示され、コンパイルターゲットファイルとエラー/警告の数が表示されます。

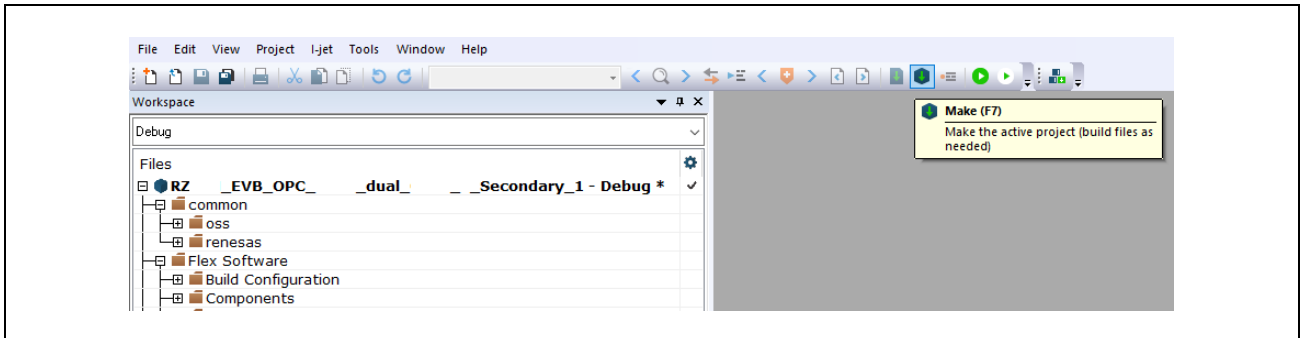


Figure 6.10. Make Secondary project

次に、再度プライマリプロジェクトを開き、Clean を実行後、Make ボタンをクリックしてビルドします。

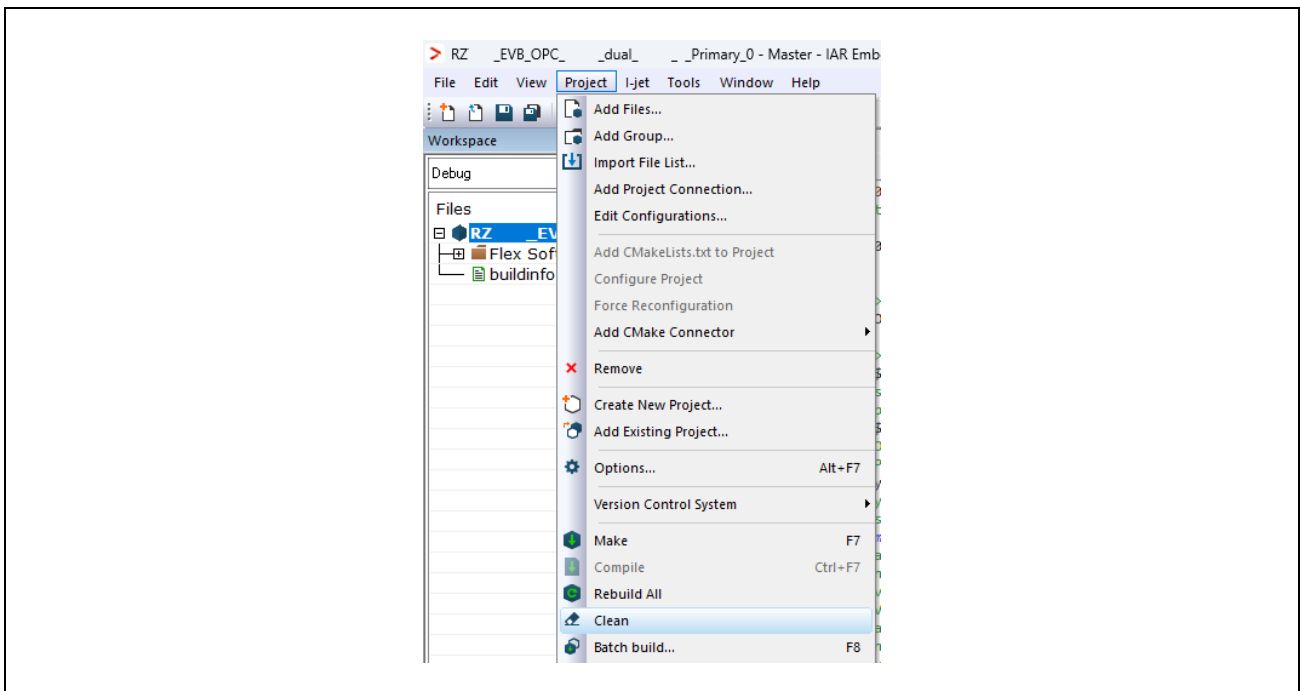


Figure 6.11. Clean Primary project

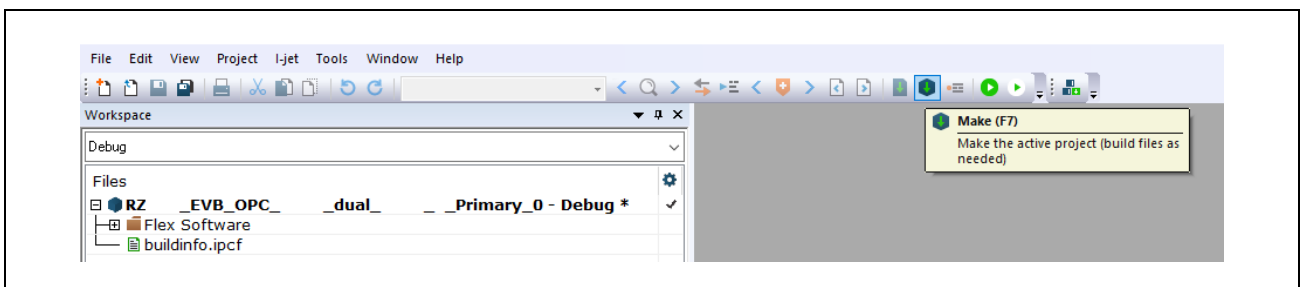


Figure 6.12. Make Primary project

6.1.1.2 Flash writing, debug

プライマリプロジェクトのオプションを開き、Debug カテゴリの Multicore タブから、Asymmetric multicore を Disabled に設定します。

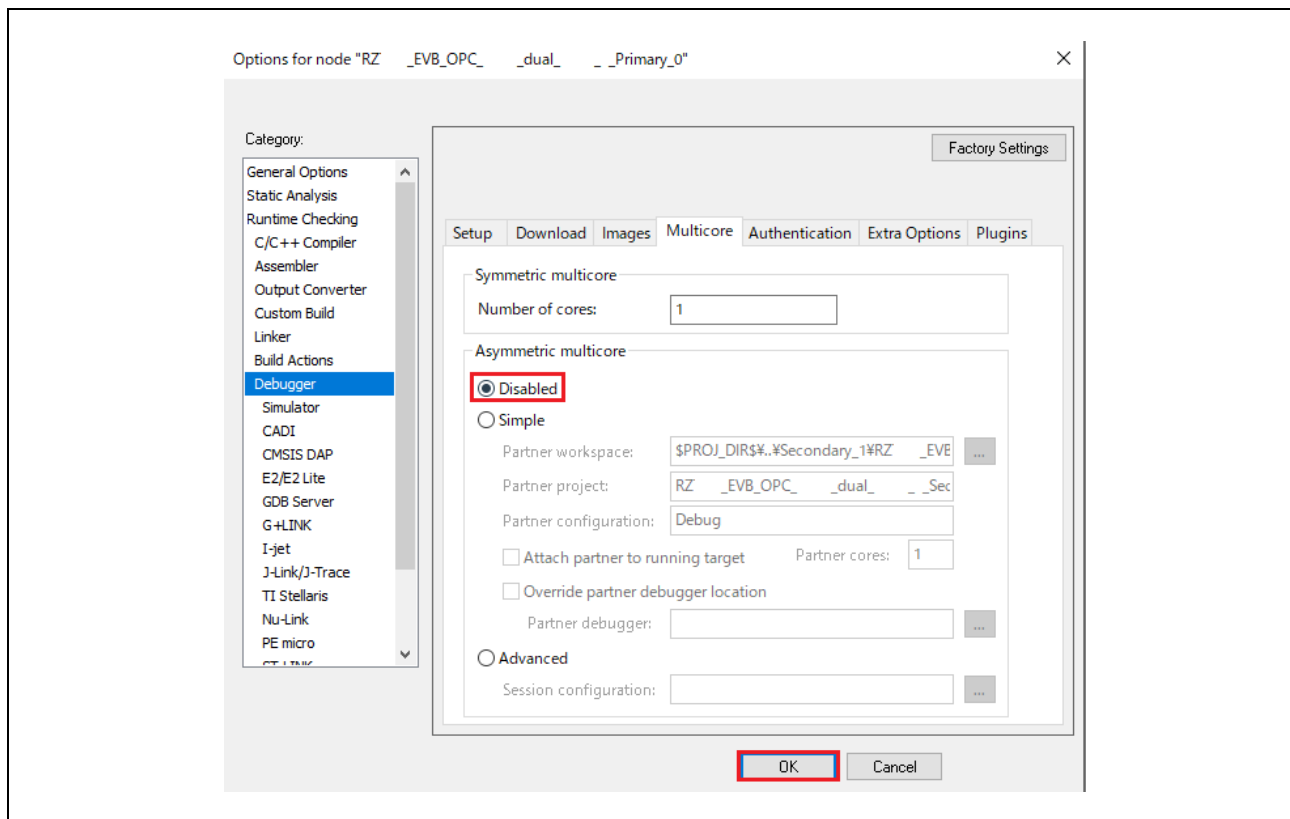


Figure 6.13. Asymmetric multicore setting

Project タグから Download>Download file...をクリックします。

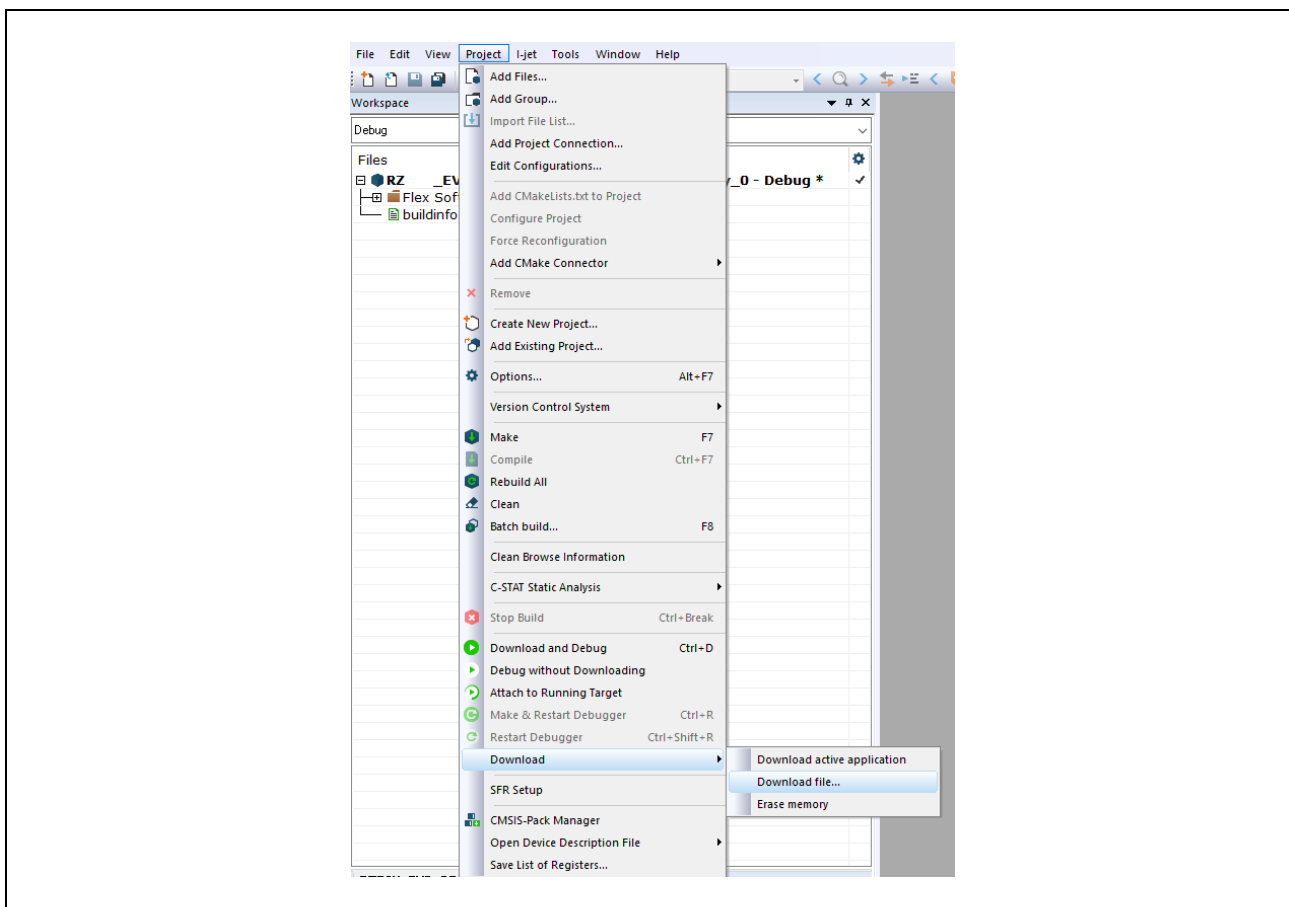


Figure 6.14. Download file...

\\(Primary Project Location)\Debug\Exe\RZ***_EVB(RSK)_OPC_***_*_Primary_0.out を選択すると、プログラムが Flash にダウンロードされます。

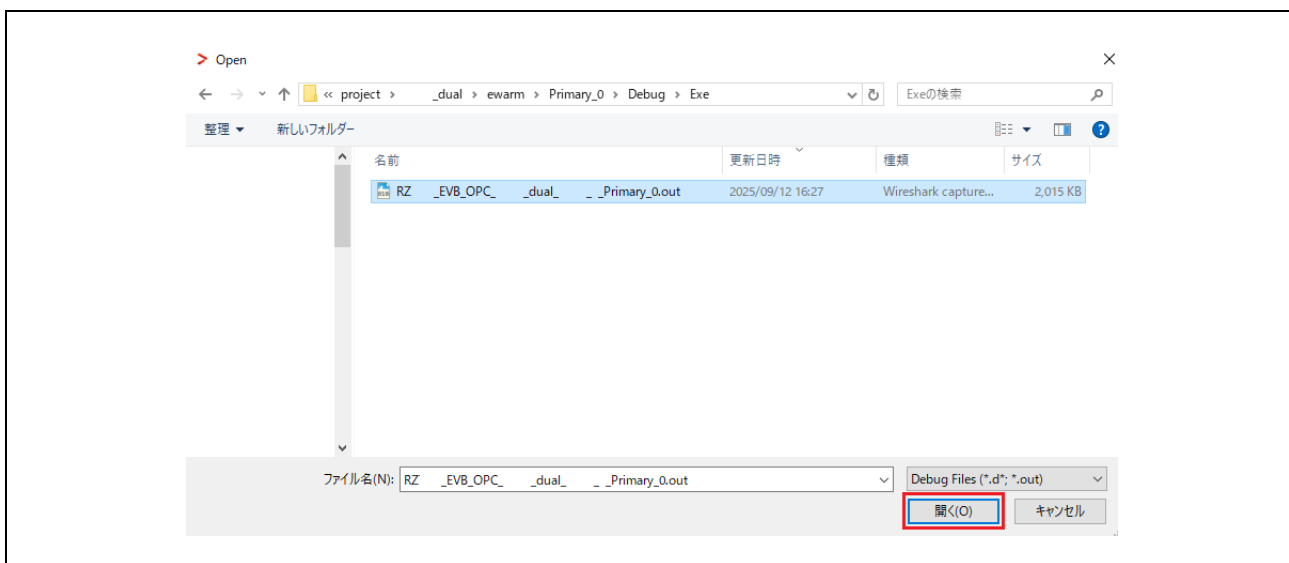


Figure 6.15. Open .out file

デバッガを使用せず評価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、オプションを開き Debug カテゴリの Multicore タグから、Asymmetric multicore から Simple を選択し、下記のように設定します。

Partner workspace : \$PROJ_DIR\$\..\(Secondary project name)\(Secondary project name).eww

Partner project : (Secondary project name)

Partner configuration : Debug

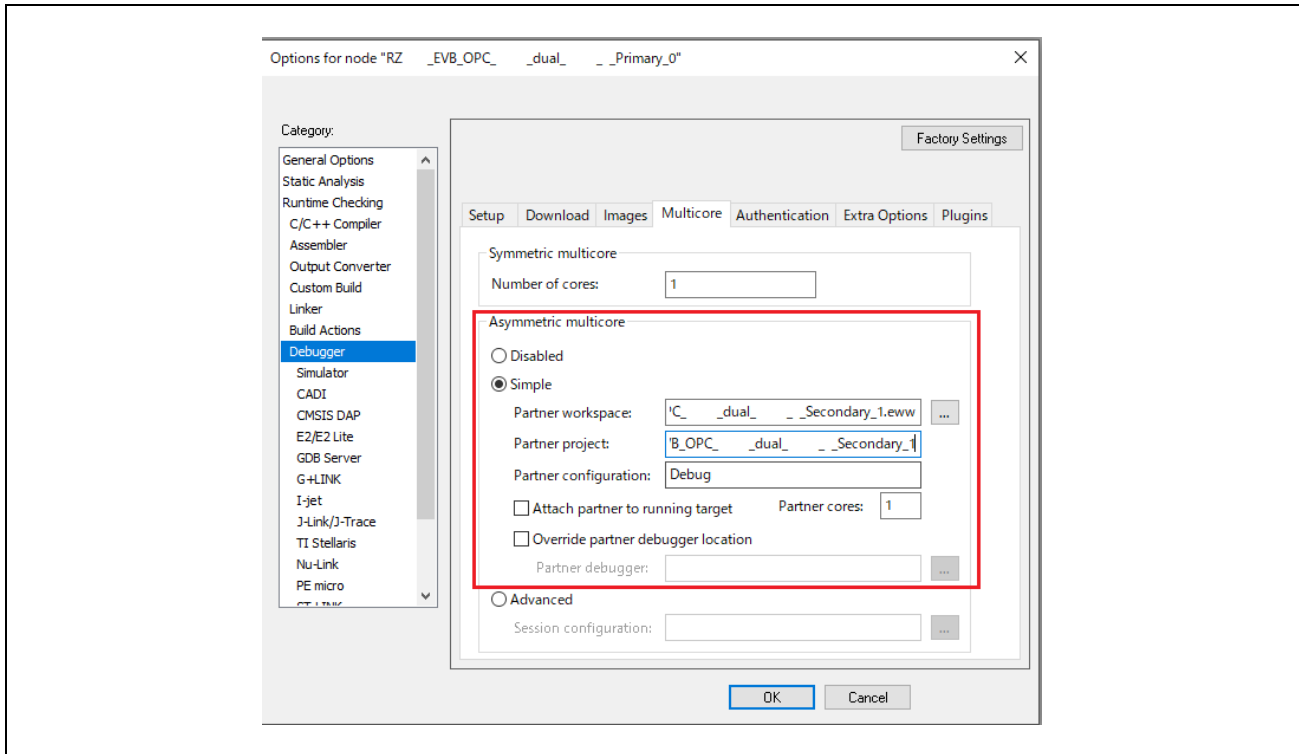


Figure 6.16. Asymmetric multicore setting

設定後、"Debug without Downloading"をクリックすると、プライマリとセカンダリそれぞれ別ウィンドウでデバッガが立ち上がります。

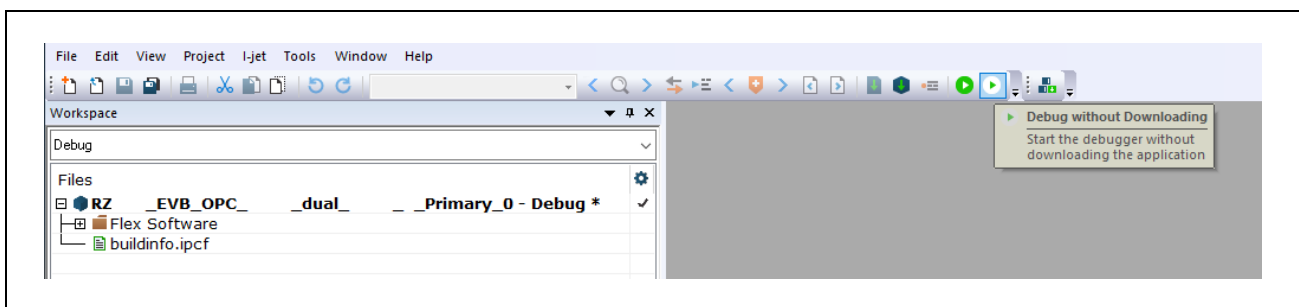


Figure 6.17. Debug without Downloading

6.1.2 For Single core project

6.1.2.1 Build

EWARMにてプロジェクトを開き、Tools タブの”FSP Smart Configurator”をクリックします。このボタンが表示されていない場合、もしくはクリック後にエラーが表示される場合は 9.5 章を参照してください。

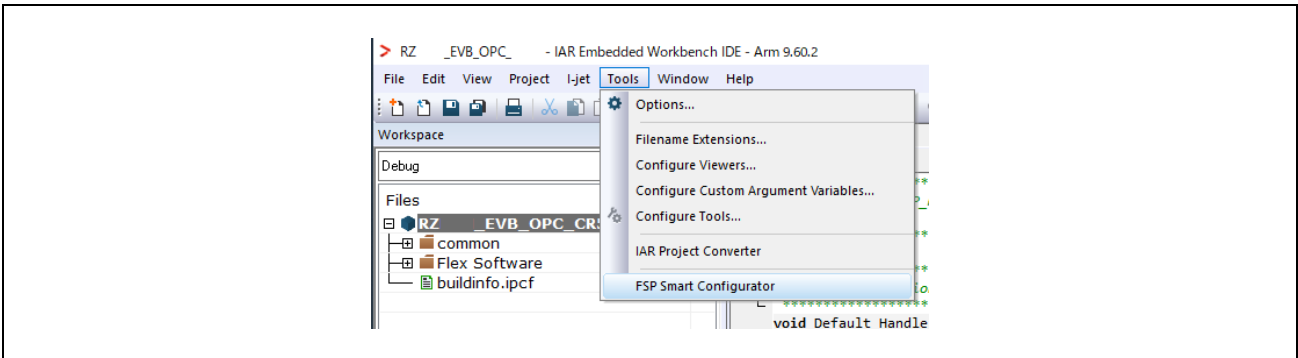


Figure 6.18. Tools tag

※サンプルソフトウェアの各種設定をデフォルトから変更したい場合は、このタイミングで、9.6 章を参考に設定を行ってください。

FSP Smart Configurator 上の“Generate Project Content”ボタンをクリックするとコードが生成されます。

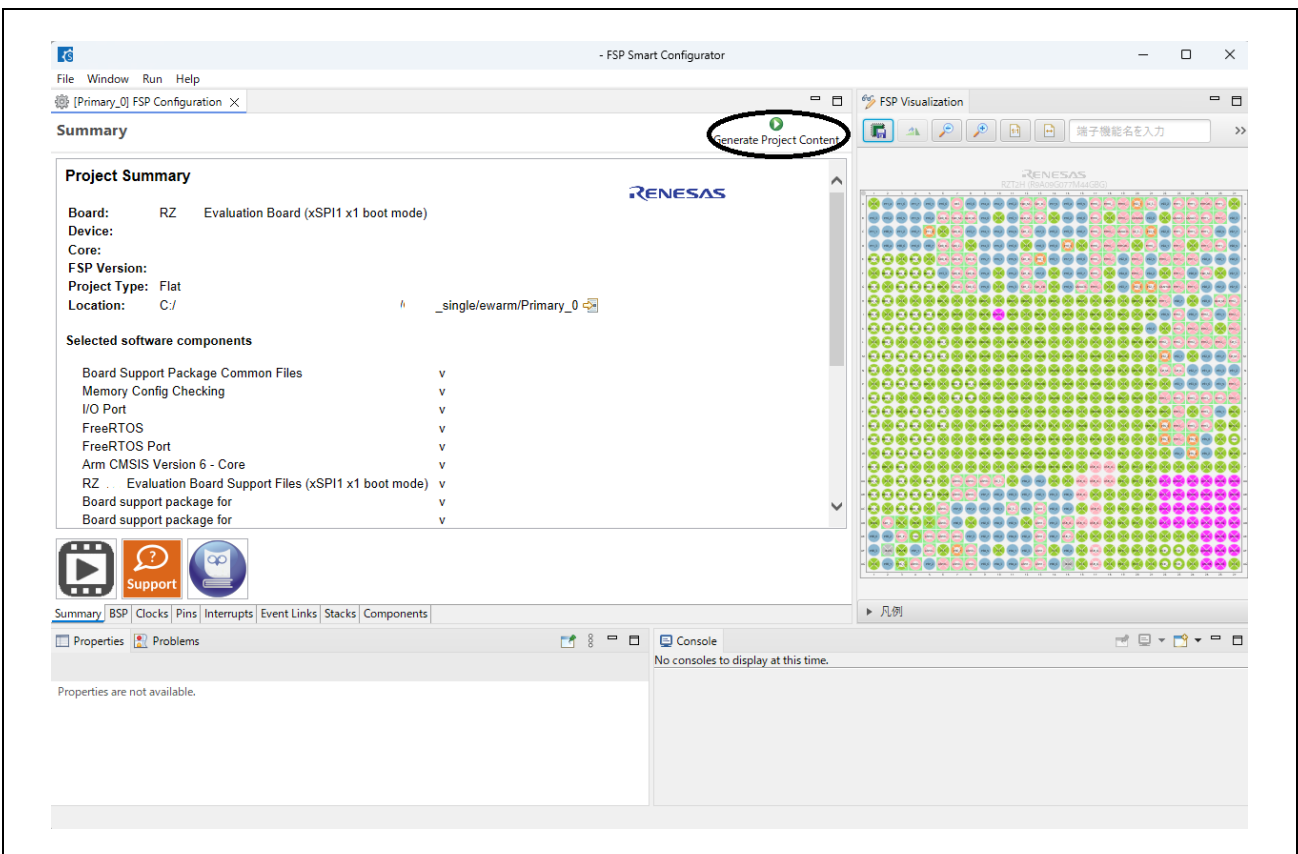


Figure 6.19. Generate Project Content

デバッガを使用してデバッグを行う場合は、下記の箇所にループ処理を追加してください。

“(Project Location)\src\hal_entry.c “

```
BSP_ATTRIBUTE_STACKLESS void R_BSP_WarmStart_StackLess(void)
{
  /* The very beginning of the startup process. */
  #if 1 // Software loops are only needed when debugging.
    __asm volatile (
      " mov r0, #0      \n"
      " movw r1, #0x68bf \n"
      " movt r1, #0x478  \n"
      "software_loop:   \n"
      " adds r0, #1      \n"
      " cmp r0, r1       \n"
      " bne software_loop \n"
      ::: "memory");
  #endif
  /* Do not delete. Required to return to system_init. */
  #if (0 == BSP_LP64_SUPPORT)
    __asm volatile ("BX lr");
  #else
    __asm volatile ("BR lr");
  #endif
}
```

EWARMにて Make ボタンをクリックしてビルドします。ビルドが完了すると、ビルドメッセージがビルドコンソールウィンドウに表示され、コンパイルターゲットファイルとエラー/警告の数が表示されます。

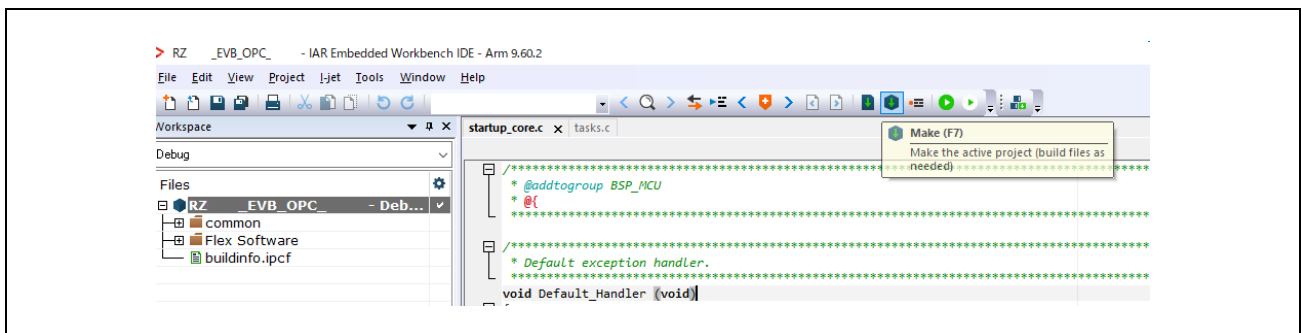


Figure 6.20. Make button

6.1.2.2 Flash writing, debug

“Download and Debug”ボタンをクリックすると、ビルドされたプログラムが評価ボードのフラッシュにダウンロードされ、デバッグが開始されます。

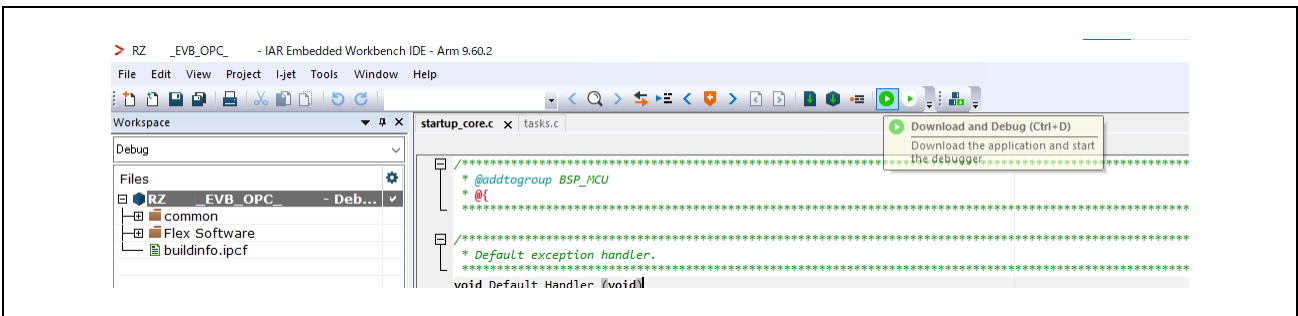


Figure 6.21. Download and Debug

デバッガを使用せず評価ボード単独で動作させる場合は、“Stop Debugging”をクリックし、デバッガが切断された後に、ボードの電源を OFF し、再度 ON してください。

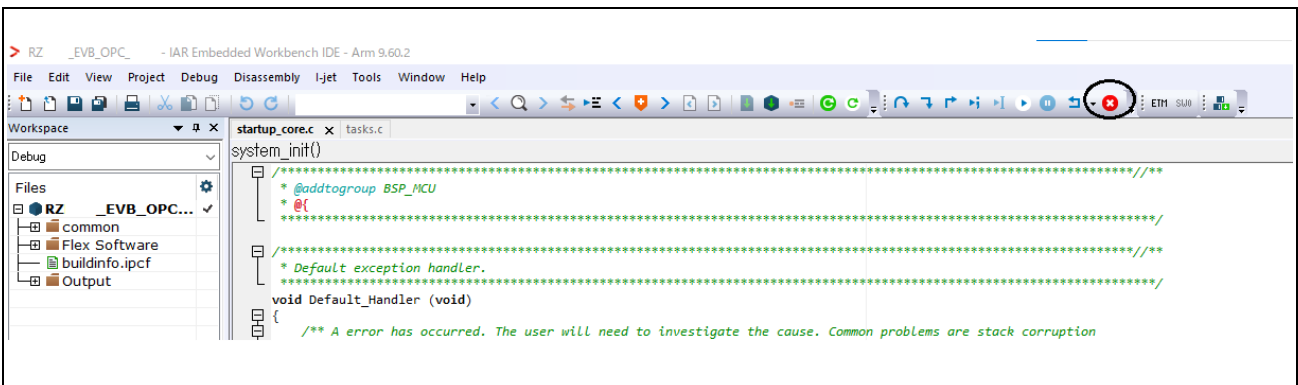


Figure 6.22. Stop Debugging

6.2 When using e² studio

e² studio 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\rz\le2studio_v20**-**_fsp_v*.*\eclipse\le2studio.exe
```

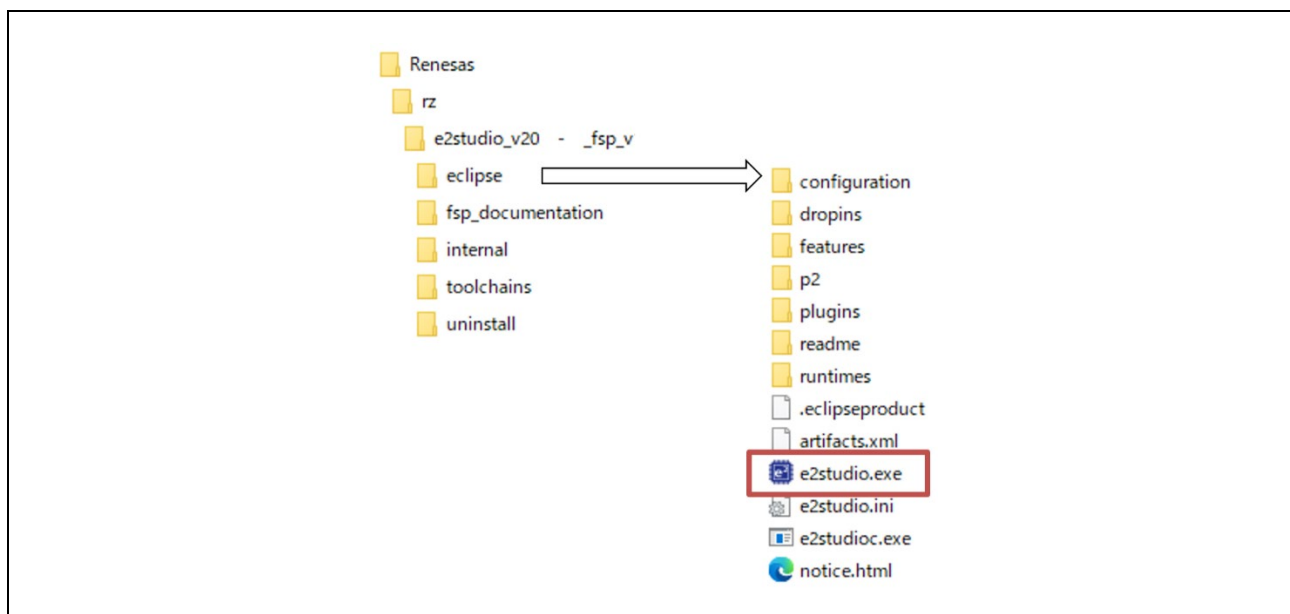


Figure 6.23. Launch project (1)

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。

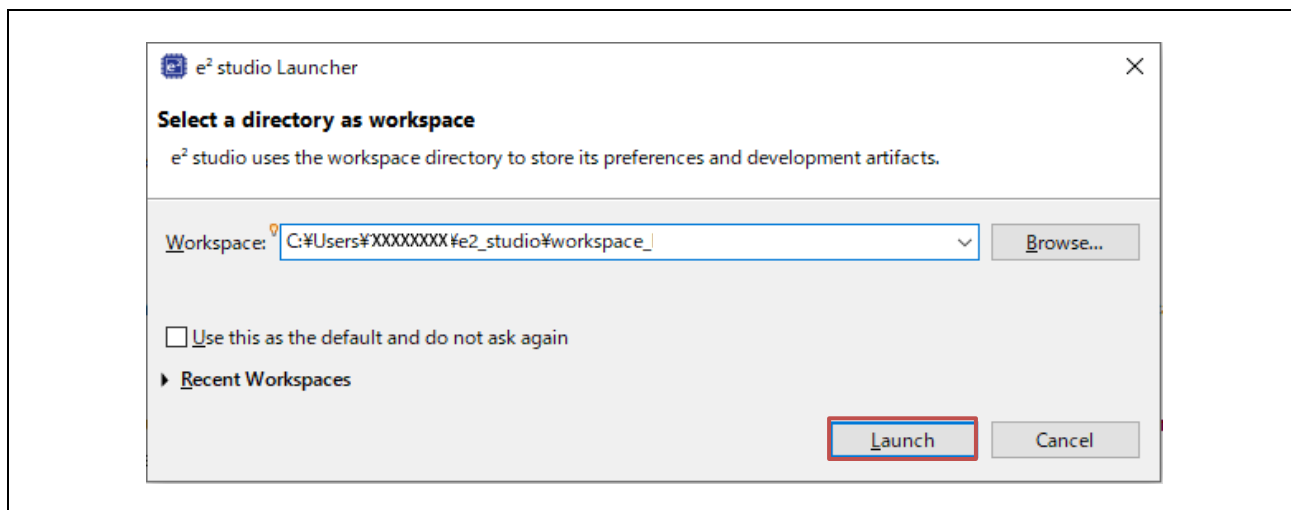


Figure 6.24. Launch project (2)

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

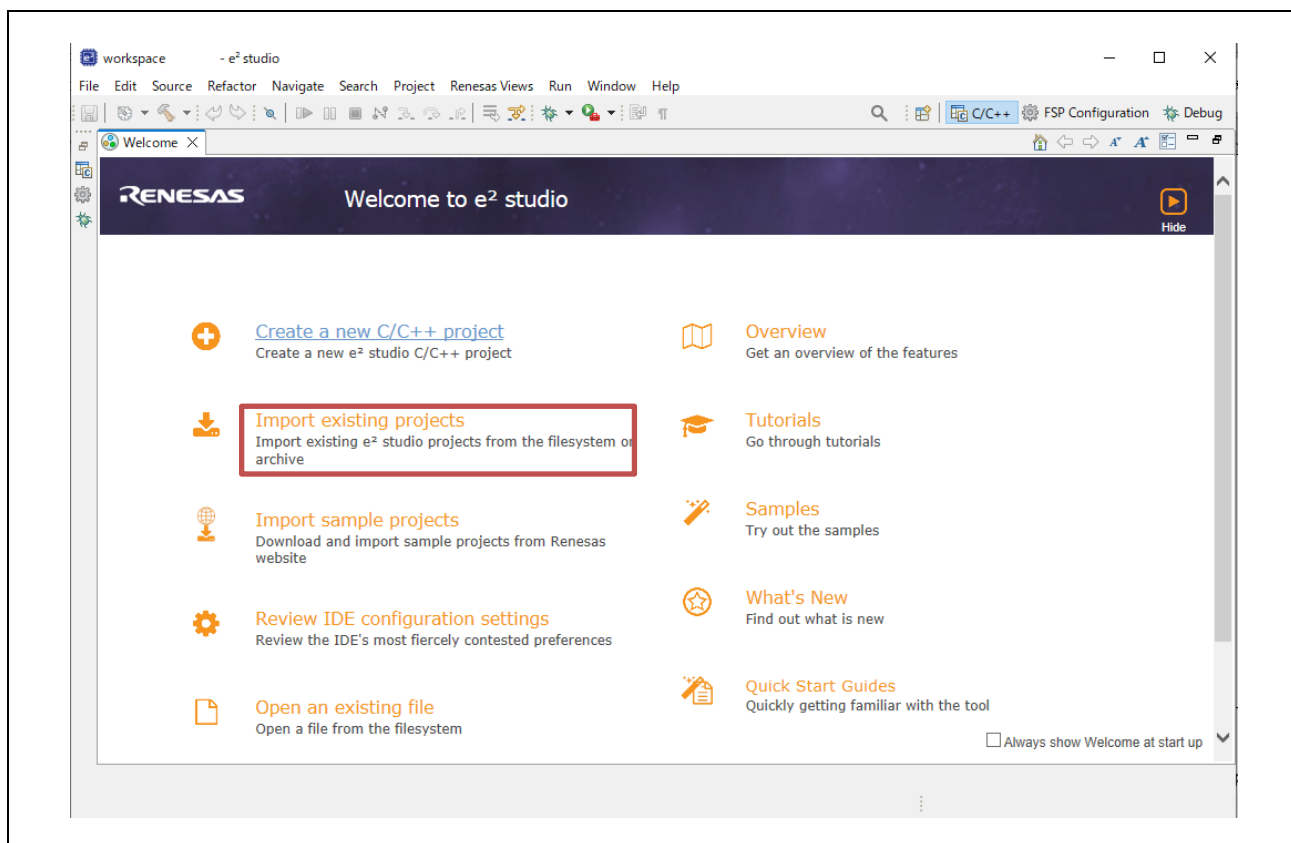


Figure 6.25. Launch project (3)

・ 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。
Dual core 構成のプロジェクトの場合 2つのプロジェクトが表示されます。

Single core (CR52)プロジェクト : RZ***_OPC_EVB(RSK)_rev****\project\CR52_single\e2studio\

Dual core (CR52-CR52)プロジェクト : RZ***_OPC_EVB(RSK)_rev****\project\CR52_dual\e2studio\

Dual core (CR52-CA55)プロジェクト : RZ***_OPC_EVB(RSK)_rev****\project\CA55_dual\e2studio\

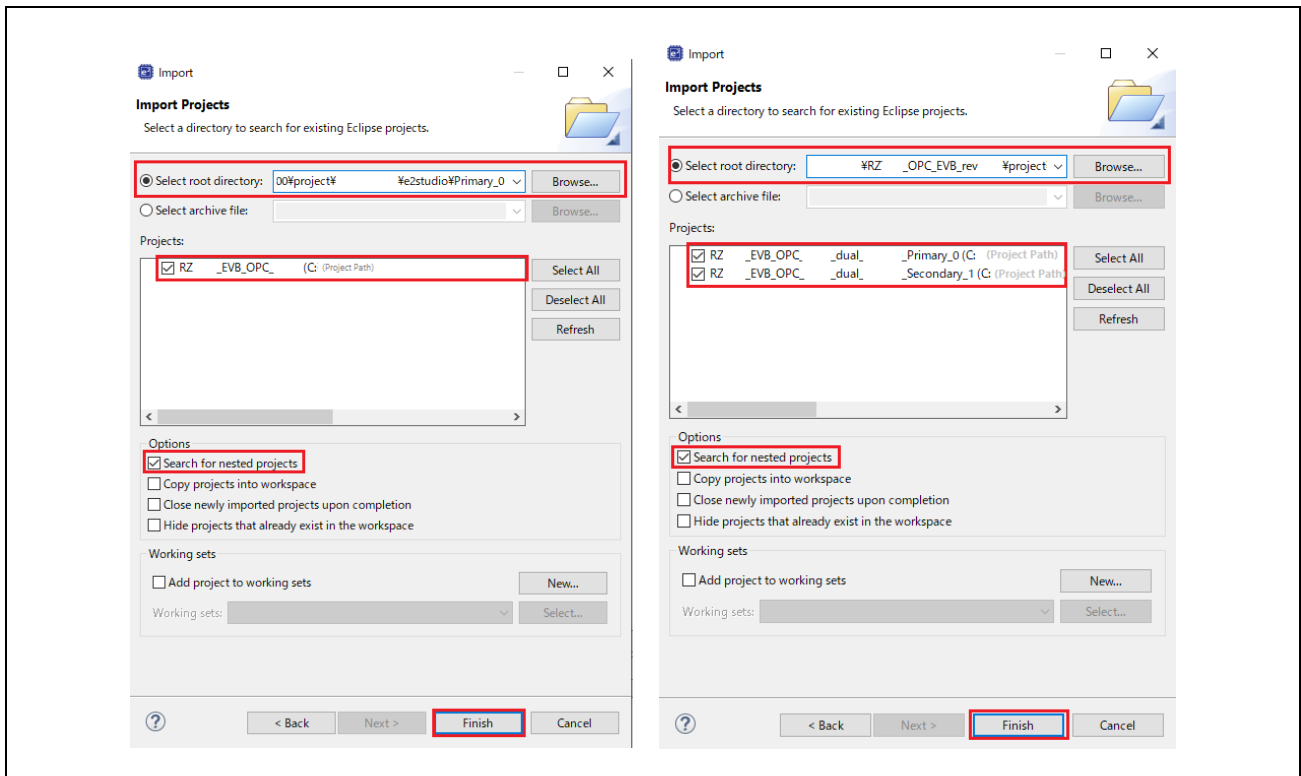


Figure 6.26. Launch project (4)

・ Figure 6.26 の Finish をクリックするとプロジェクトのインポートが開始され、終了後に自動で画面が切り替わります。

以降の手順は使用するプロジェクトによって異なります。

6.2.1 For Dual core project

6.2.1.1 Build

Project Explorer ウィンドウから、プライマリプロジェクトの 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。

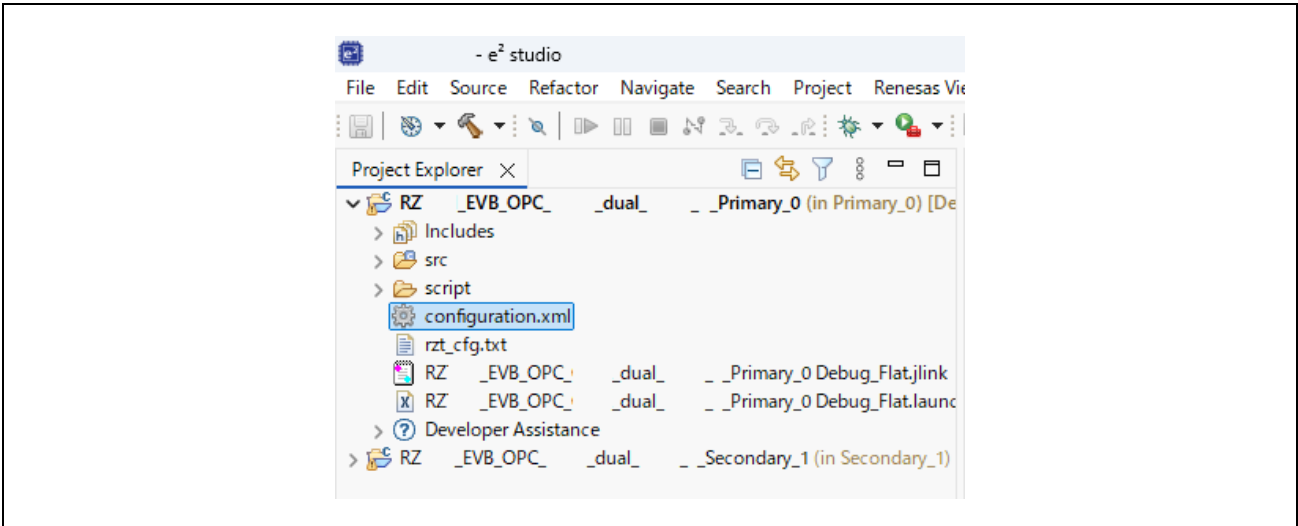


Figure 6.27. Open Smart configurator

Generate Project Content をクリックし、ソースコードを生成します。

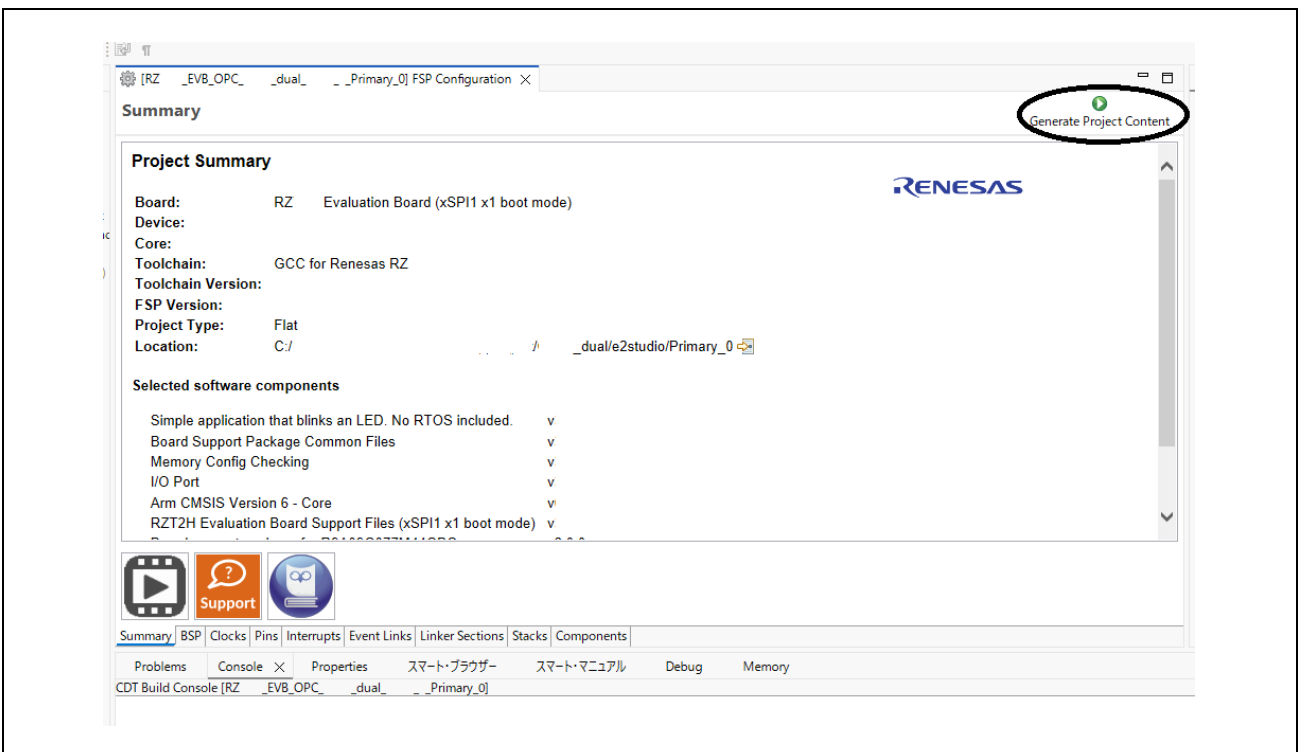


Figure 6.28. Generate Project Content

デバッガを使用してデバッグを行う場合は、下記の箇所にループ処理を追加してください。

“(Primary Project Location)\src\hal_entry.c “

```

BSP_ATTRIBUTE_STACKLESS void R_BSP_WarmStart_StackLess(void)
{
  /* The very beginning of the startup process. */
  #if 1 // Software loops are only needed when debugging.
    __asm volatile (
      "  mov  r0, #0      \n"
      "  movw r1, #0x68bf \n"
      "  movt r1, #0x478  \n"
      "software_loop:   \n"
      "  adds r0, #1      \n"
      "  cmp  r0, r1      \n"
      "  bne  software_loop \n"
      ::: "memory");
  #endif
  /* Do not delete. Required to return to system_init. */
  #if (0 == BSP_LP64_SUPPORT)
    __asm volatile ("BX lr");
  #else
    __asm volatile ("BR lr");
  #endif
}

```

プライマリプロジェクトを選択した上で、Projectメニューの Clean...をクリックします。

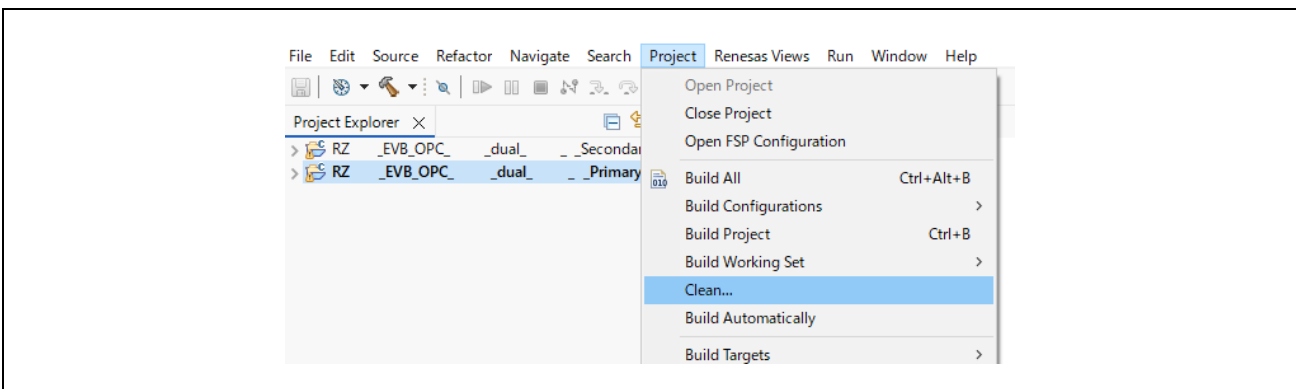


Figure 6.29. Open project Clean...

以下のように選択して Clean をクリックすると全ビルドを開始します。

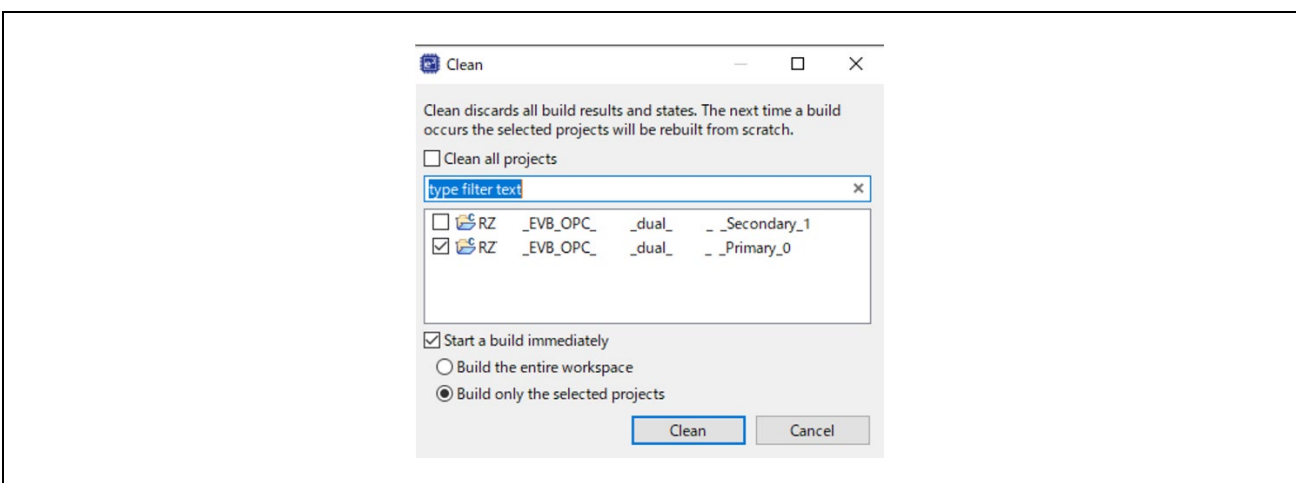


Figure 6.30. Clean and rebuild Primary project

Project Explorer ウィンドウから、セカンダリプロジェクトの configuration.xml を開きます。

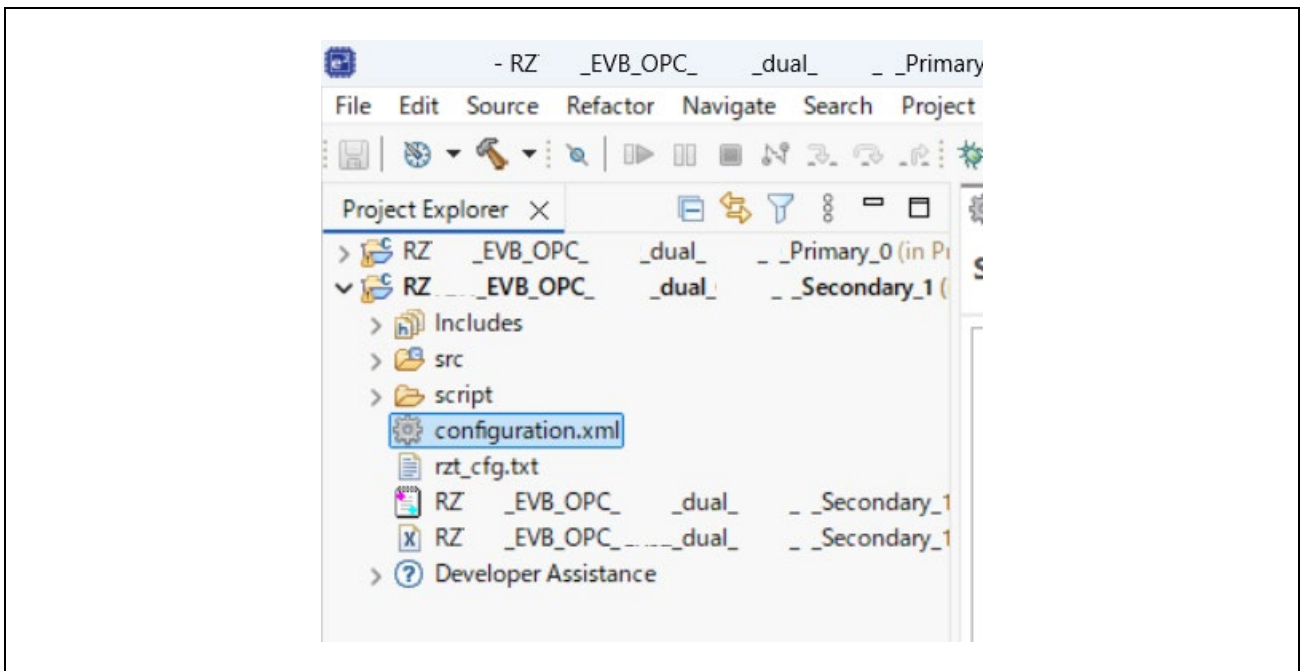


Figure 6.31. Open Smart configurator

※サンプルソフトウェアの各種設定をデフォルトから変更したい場合は、このタイミングで、9.6 章を参考に設定を行ってください。

Generate Project Content をクリックします。

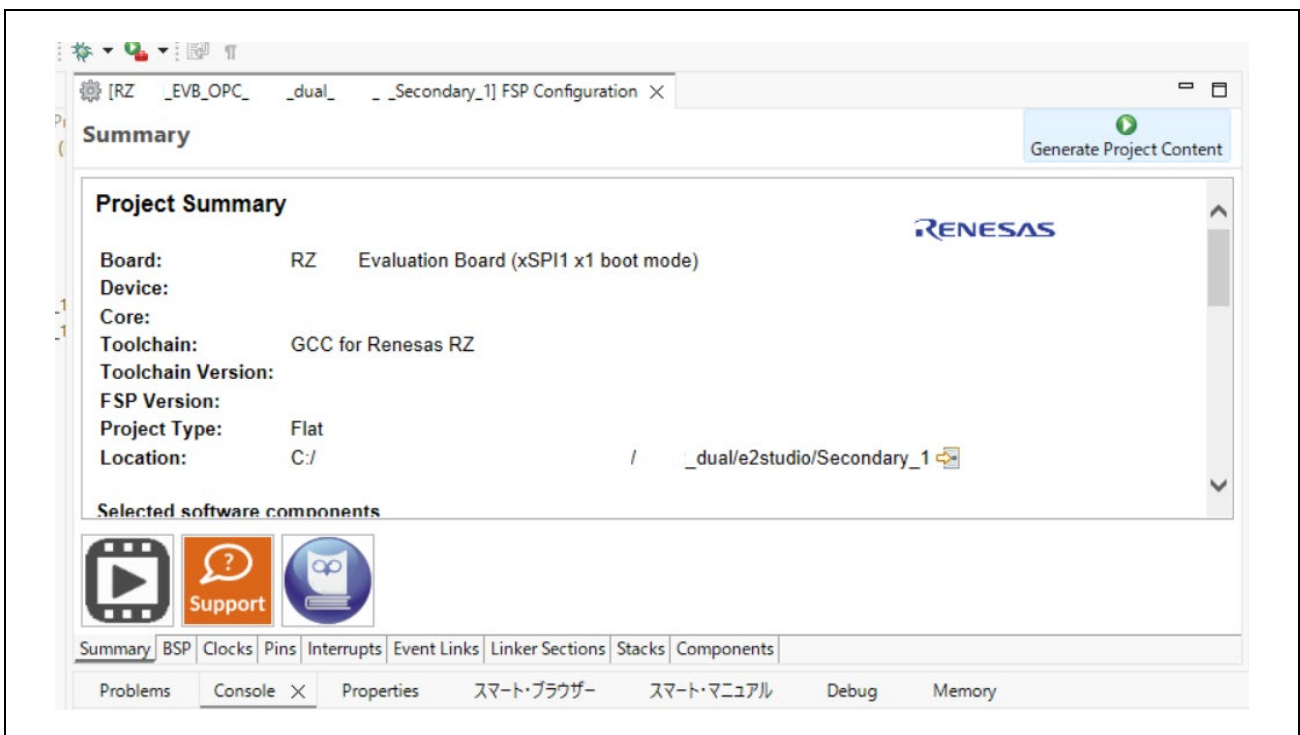


Figure 6.32. Generate Project Content

デバッガを使用してデバッグを行う場合は、下記の箇所にコアに応じたループ処理を追加してください。

“(Secondary Project Location)\src\hal_entry.c “

CR52	<pre> BSP_ATTRIBUTE_STACKLESS void R_BSP_WarmStart_StackLess(void) { /* The very beginning of the startup process. */ #if 1 // Software loops are only needed when debugging. __asm volatile (" mov r0, #0 \n" " movw r1, #0x68bf \n" " movt r1, #0x478 \n" "software_loop: \n" " adds r0, #1 \n" " cmp r0, r1 \n" " bne software_loop \n" ::: "memory"); #endif /* Do not delete. Required to return to system_init. */ #if (0 == BSP_LP64_SUPPORT) __asm volatile ("BX lr"); #else __asm volatile ("BR lr"); #endif } </pre>
CA55	<pre> BSP_ATTRIBUTE_STACKLESS void R_BSP_WarmStart_StackLess(void) { /* The very beginning of the startup process. */ #if 1 // Software loops are only needed when debugging. __asm volatile ("WFI \n" "WFI \n" "WFI \n" "WFI \n"); #endif /* Do not delete. Required to return to system_init. */ #if (0 == BSP_LP64_SUPPORT) __asm volatile ("BX lr"); #else __asm volatile ("BR lr"); #endif } </pre>

セカンダリプロジェクトを選択した上で、ProjectメニューのClean...をクリックします。

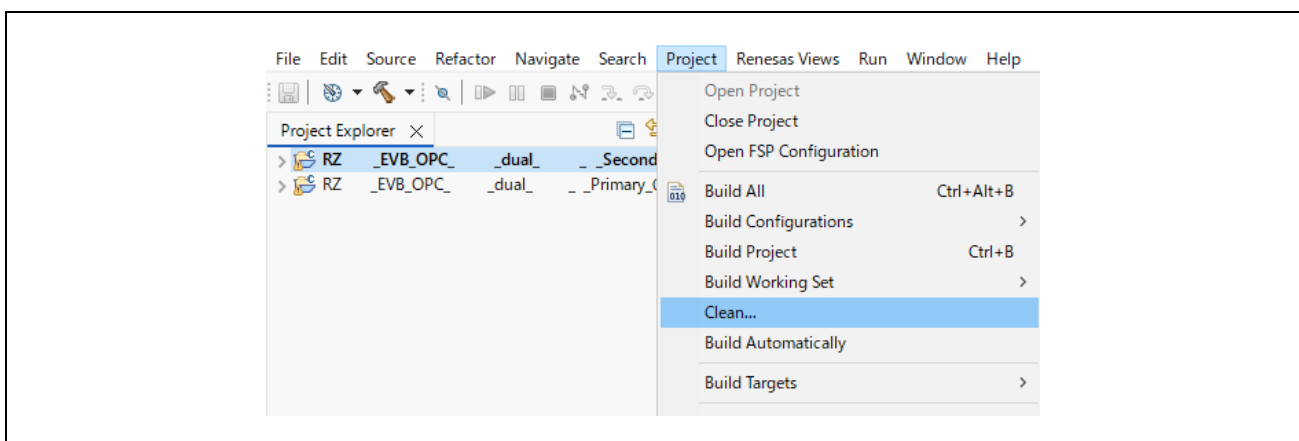


Figure 6.33. Open project Clean...

以下のように設定し Clean をクリックすると全ビルドを開始します。

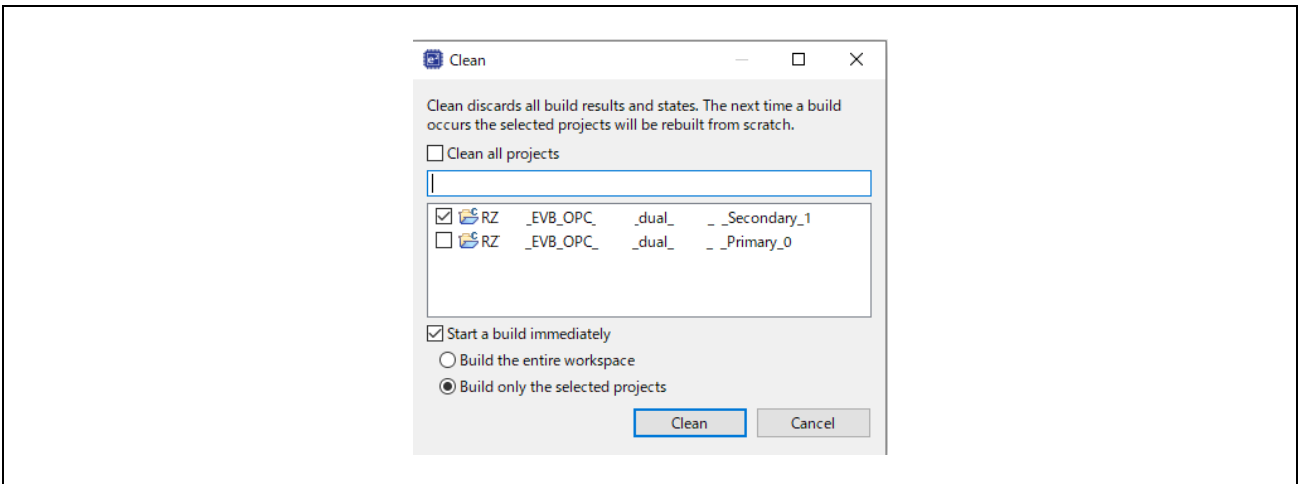


Figure 6.34. Clean and rebuild Secondary project

セカンダリプロジェクトのビルド結果が 0 errors であることを確認後、Project Explorer ウィンドウから、プライマリプロジェクトを選択した上で、Project メニューの Clean... をクリックします。

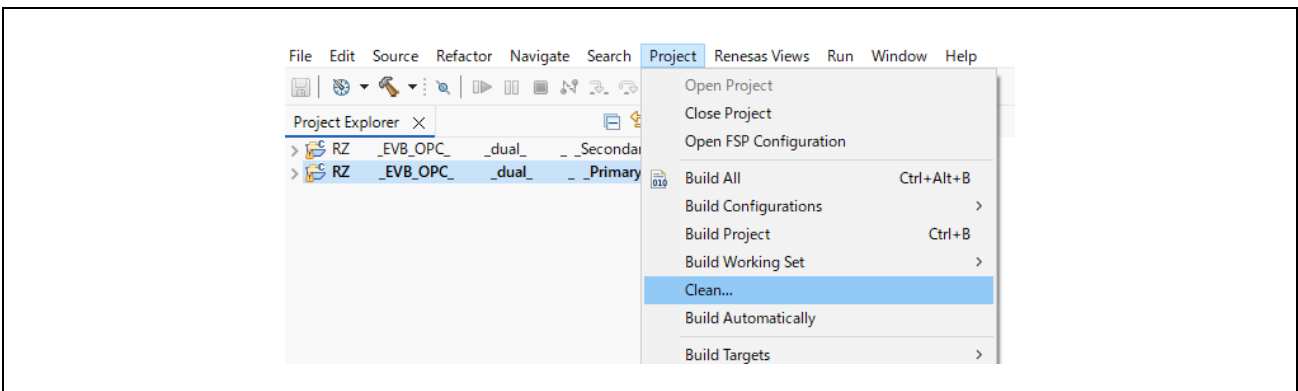


Figure 6.35. Open project Clean...

以下のように選択して Clean をクリックすると全ビルドを開始します。

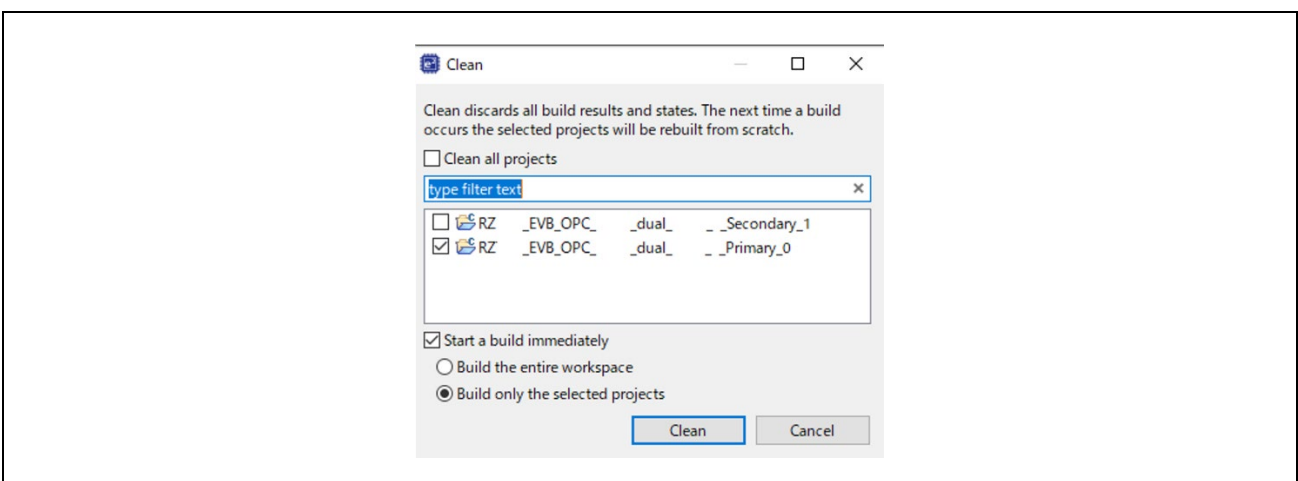


Figure 6.36. Clean and rebuild Primary project

6.2.1.2 Flash writing, debug

ビルド終了後のダウンロード手順を以下に示します。

プライマリプロジェクトを選択し、デバッグアイコンをクリックするとプログラムがFlashにダウンロードされます。

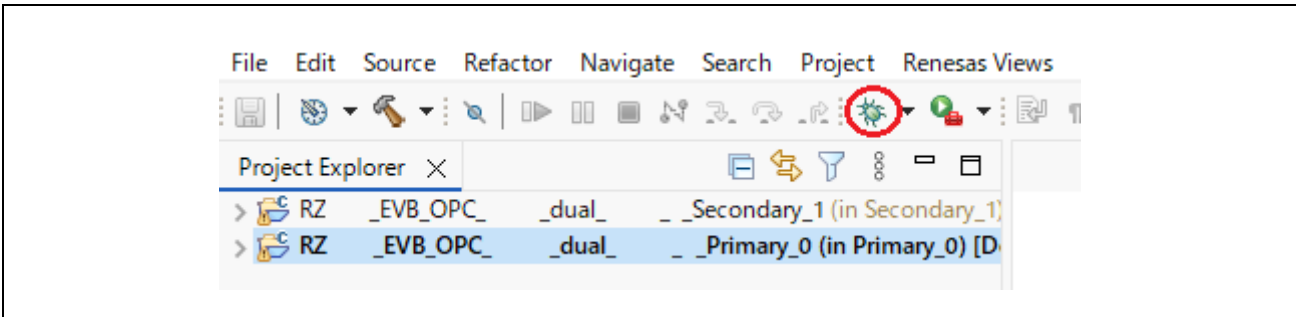


Figure 6.37. Debug Primary

Figure 6.38 のウィンドウが表示されるので、No を選択してください。

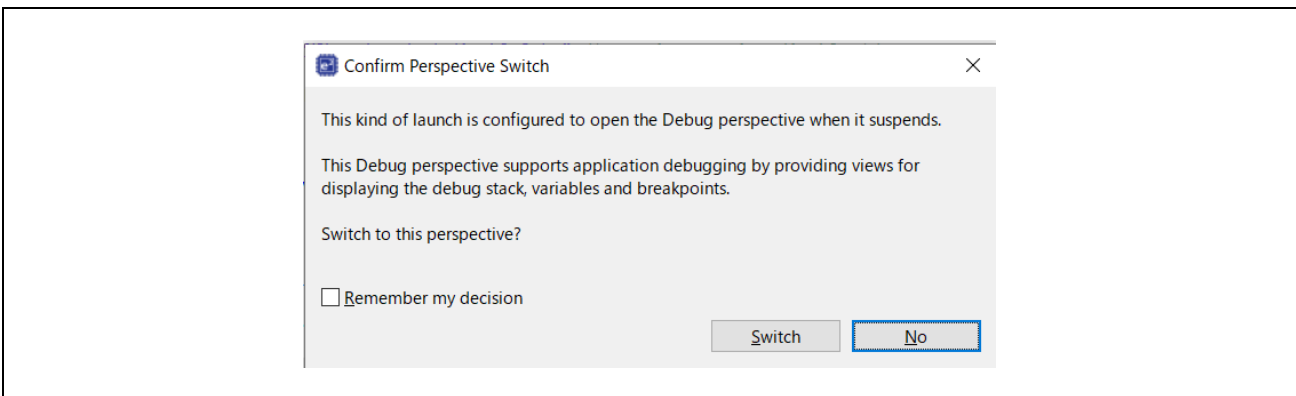


Figure 6.38. Perspective Switch

デバッガを使用せず評価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、下記手順の通り実施してください。

(1) For CR52_dual project

セカンダリプロジェクトを選択しデバッグアイコンをクリックします。

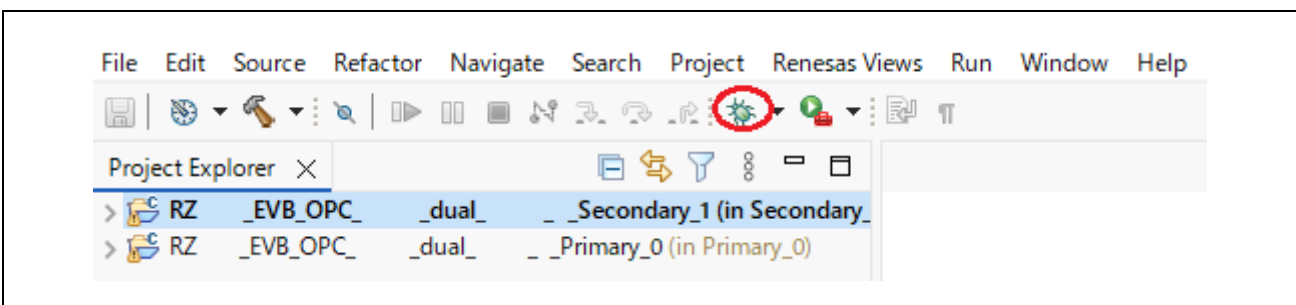


Figure 6.39. Debug Secondary

Figure 6.40 が表示されるので No をクリックしてください。

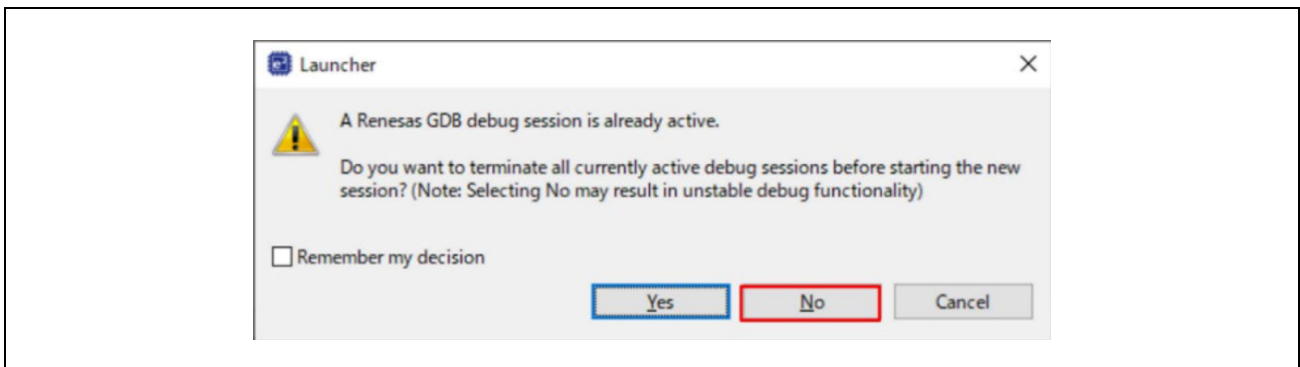


Figure 6.40. Warning Window of Starting Debug Session

Figure 6.41 が表示されるので Yes をクリックしてください。

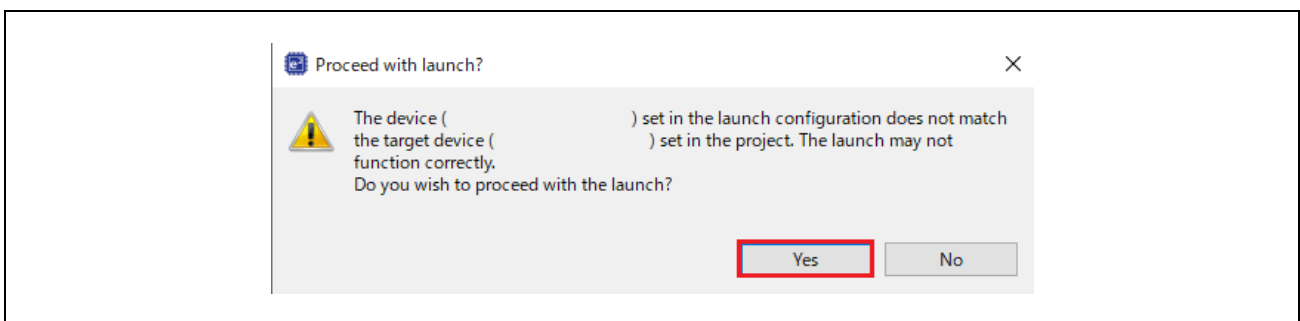


Figure 6.41. Warning Window of Device Name

Figure 6.42 が表示されるので Switch をクリックしてください。

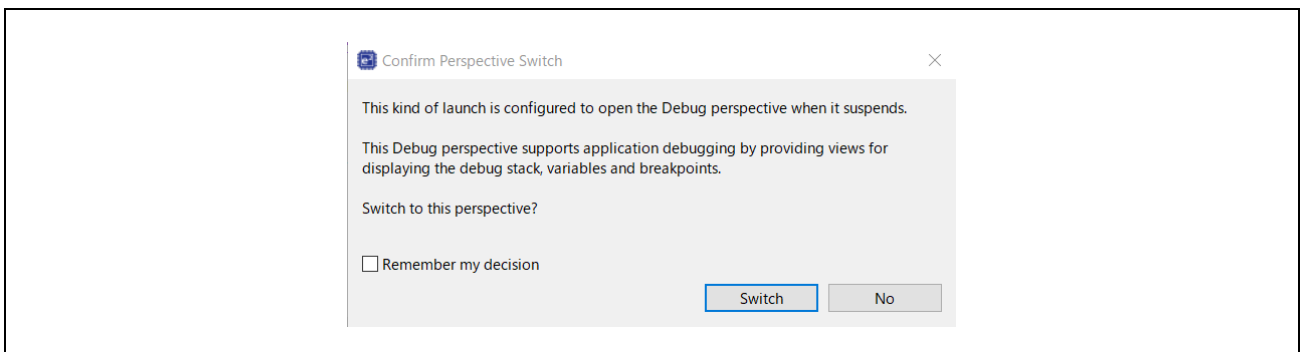


Figure 6.42. Perspective Switch

Debug タグから各 CPU 用プロジェクトを選択後、Resume アイコンをクリックすることで、プログラムが動作し、main()で break します。再度、各プロジェクトで Resume をすることで、それぞれの CPU でプログラムが動作し始めます。

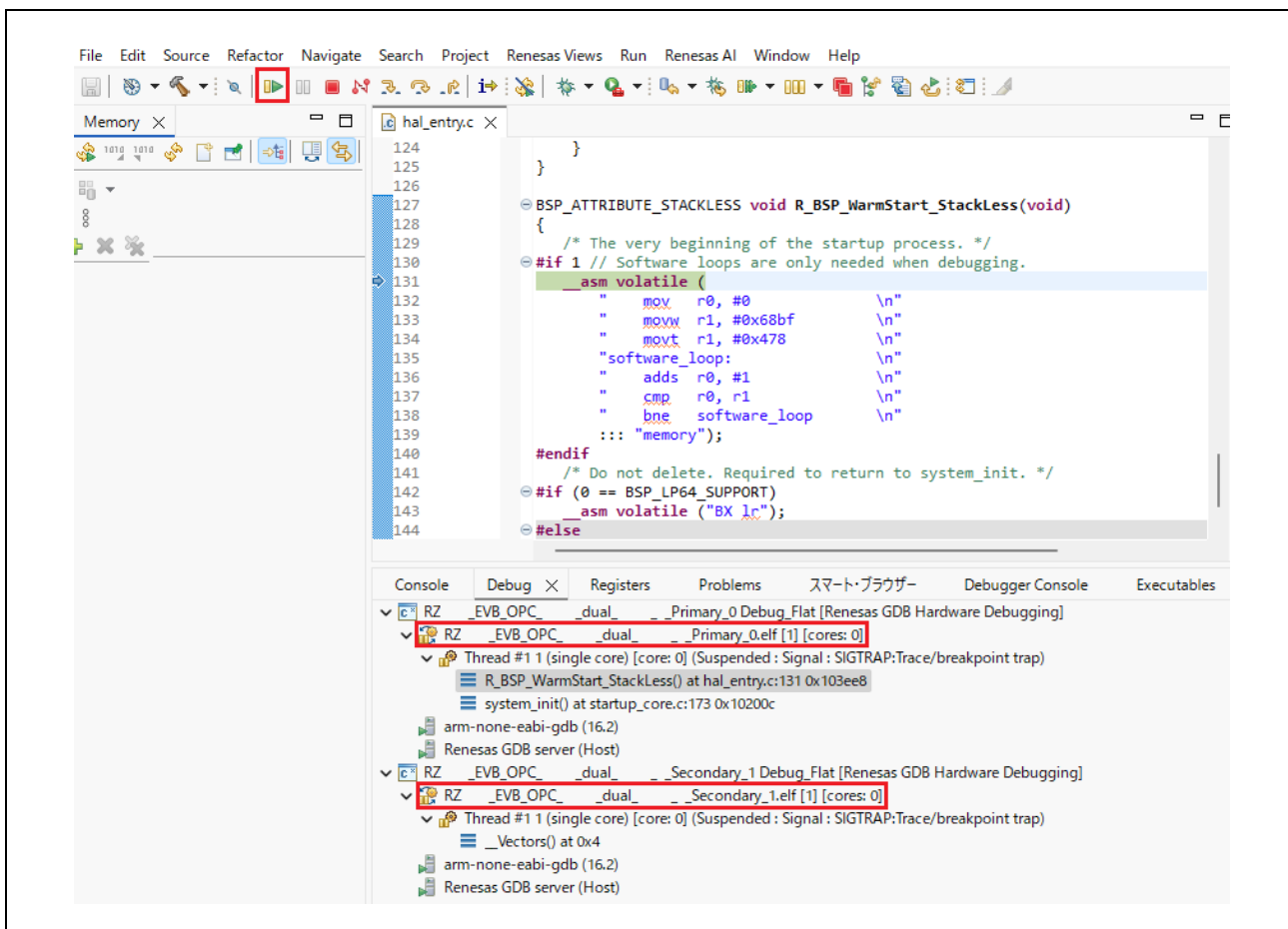


Figure 6.43. Debug

(2) For CA55_dual project

Resume アイコンをクリックするとプライマリプロジェクトのプログラムが動作し、main()で break します。

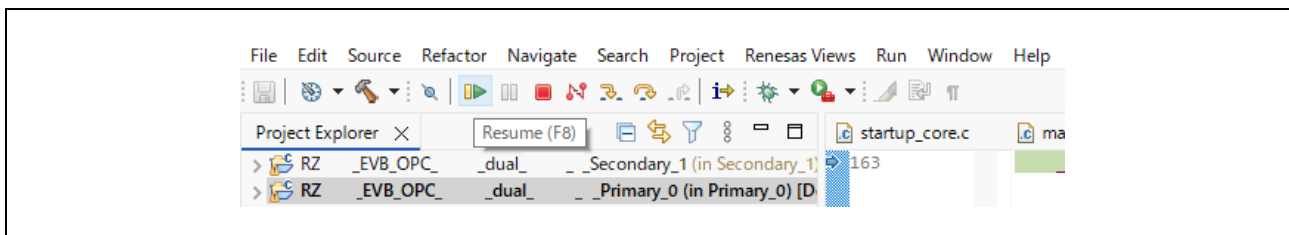


Figure 6.44. Resume Primary project

セカンダリプロジェクトを選択しデバッグアイコンをクリックします。

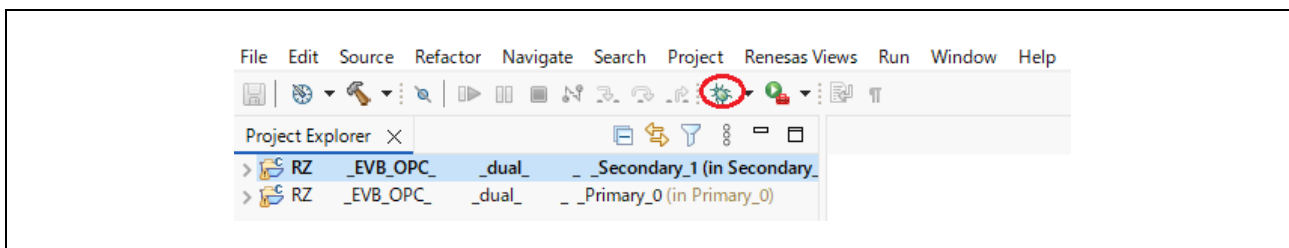


Figure 6.45. Debug Secondary project

Figure 6.46 が表示されるので No をクリックしてください。

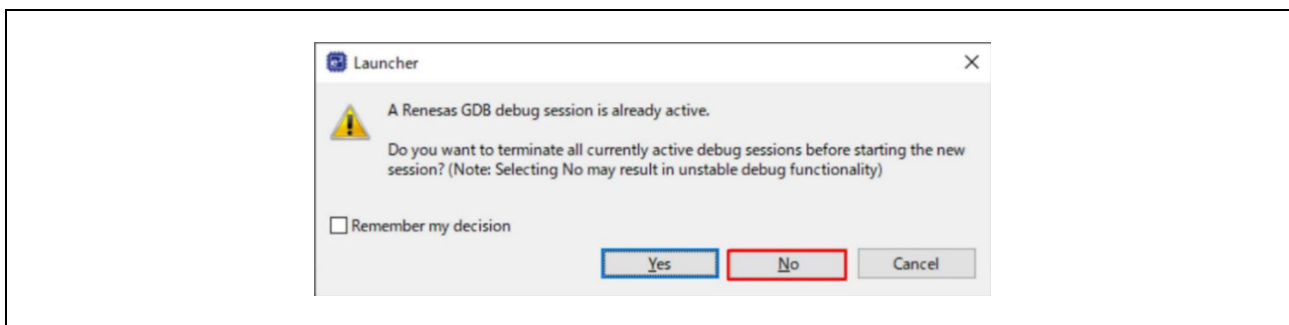


Figure 6.46. Warning Window of Starting Debug Session

Figure 6.47 が表示されるので Yes をクリックしてください。

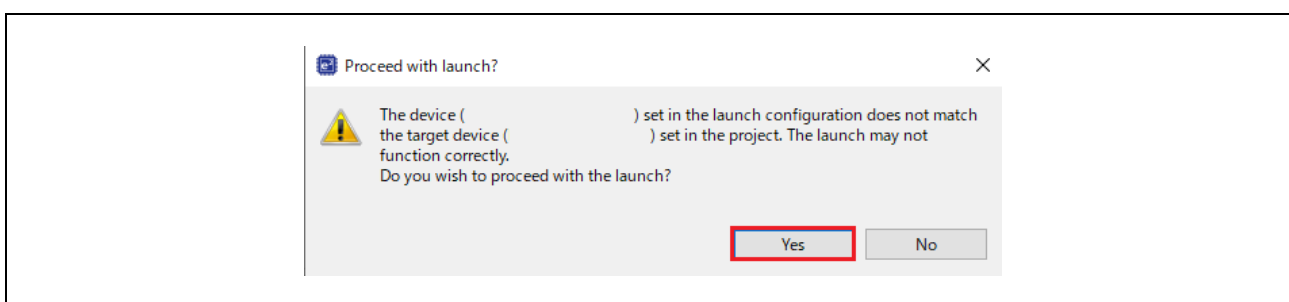


Figure 6.47. Warning Window of Device Name

Figure 6.48 が表示されるので Switch をクリックしてください。

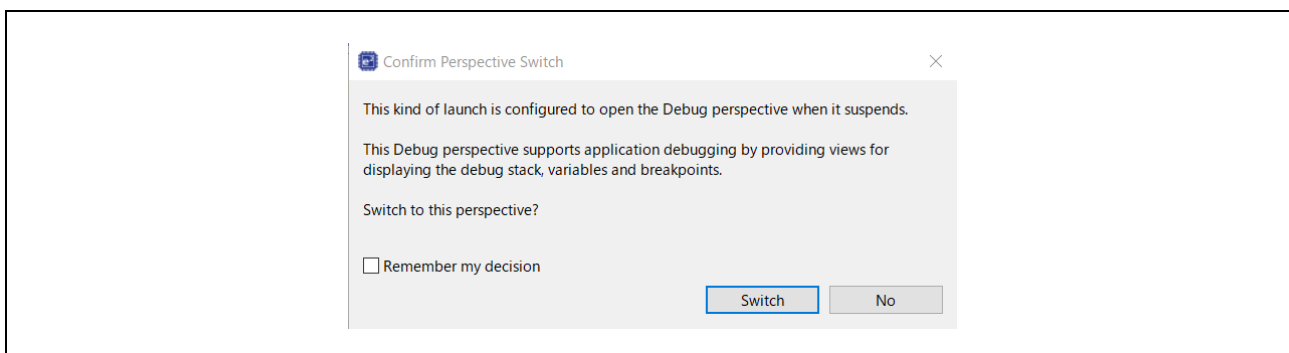


Figure 6.48. Perspective Switch

Debug タグから各 CPU 用プロジェクトを選択後、Resume アイコンをクリックすることで、プログラムが動作します。

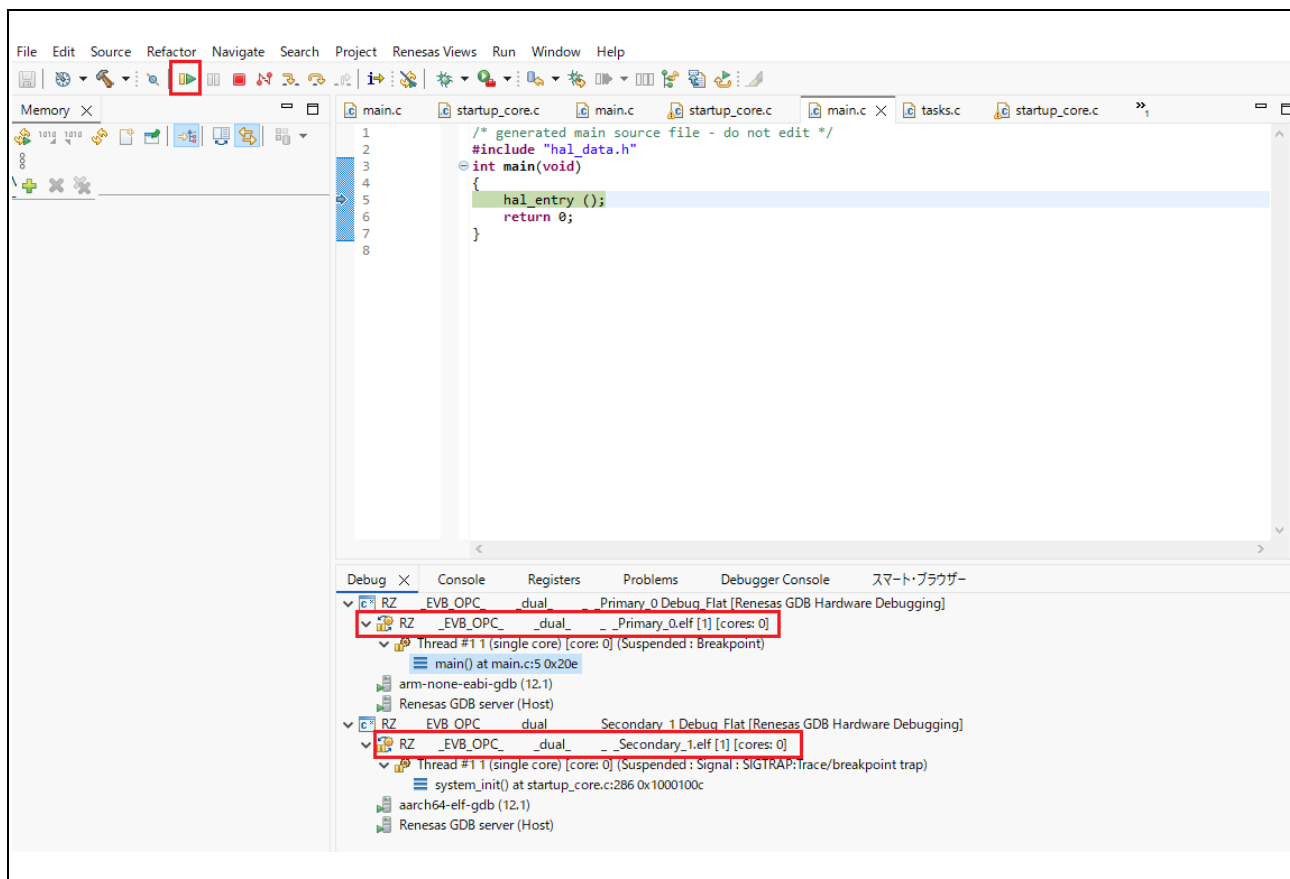


Figure 6.49. Debug

6.2.2 For Single core project

6.2.2.1 Build

Project Explorer ウィンドウから、configuration.xml を開きます。

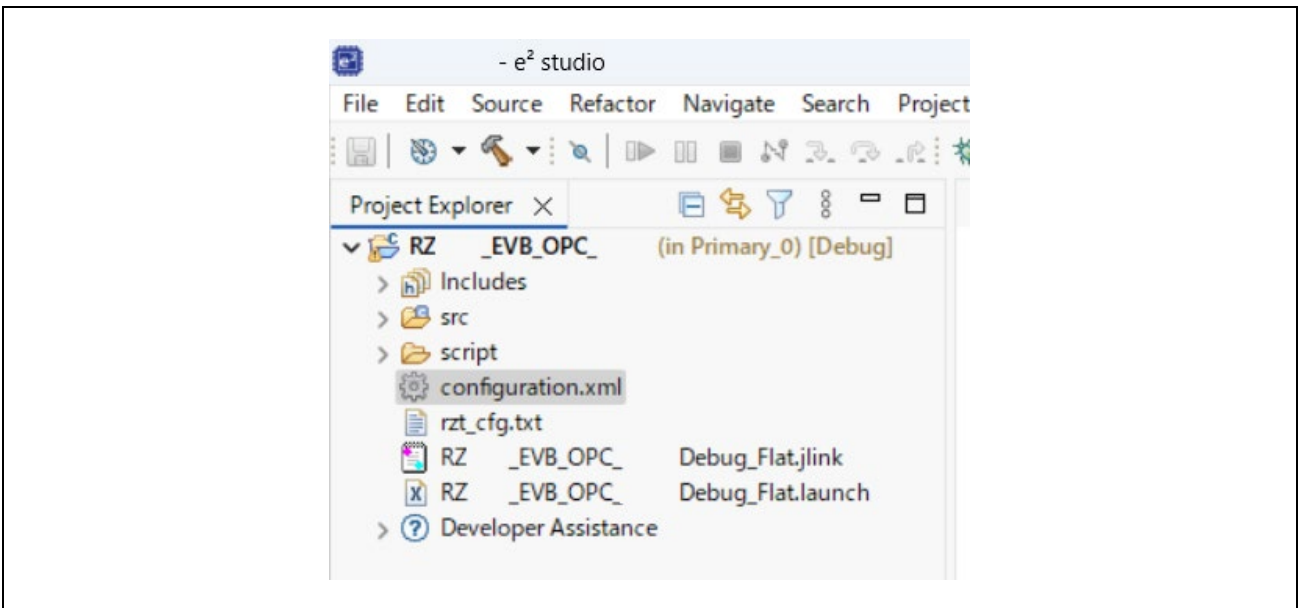


Figure 6.50. Open Smart configurator

※サンプルソフトウェアの各種設定をデフォルトから変更したい場合は、このタイミングで、9.6章を参考に設定を行ってください。

Generate Project Content をクリックします。

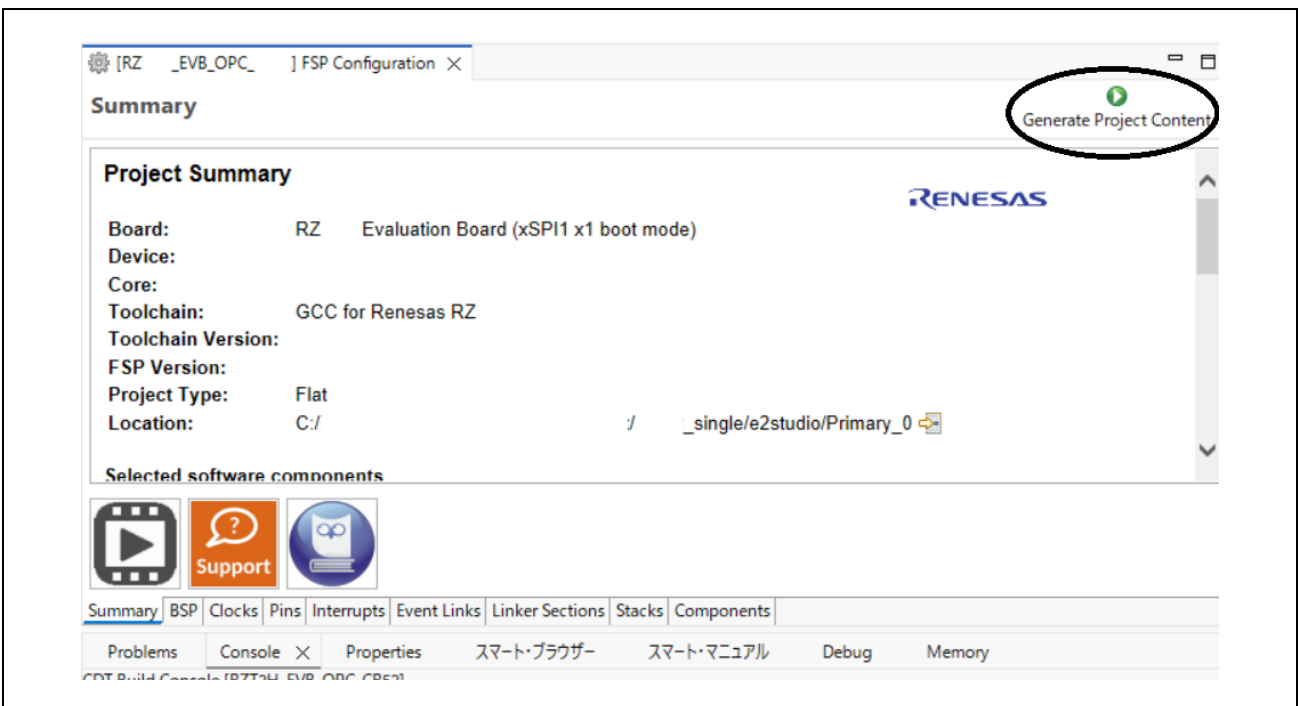


Figure 6.51. Generate Project Content

デバッガを使用してデバッグを行う場合は、下記のループ処理を追加してください。

“(Project Location)\src\hal_entry.c “

```
BSP_ATTRIBUTE_STACKLESS void R_BSP_WarmStart_StackLess(void)
{
  /* The very beginning of the startup process. */
  #if 1 // Software loops are only needed when debugging.
    __asm volatile (
      "  mov  r0, #0      \n"
      "  movw r1, #0x68bf \n"
      "  movt r1, #0x478  \n"
      "software_loop:   \n"
      "  adds r0, #1      \n"
      "  cmp  r0, r1      \n"
      "  bne  software_loop \n"
      ::: "memory");
  #endif
  /* Do not delete. Required to return to system_init. */
  #if (0 == BSP_LP64_SUPPORT)
    __asm volatile ("BX lr");
  #else
    __asm volatile ("BR lr");
  #endif
}
```

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

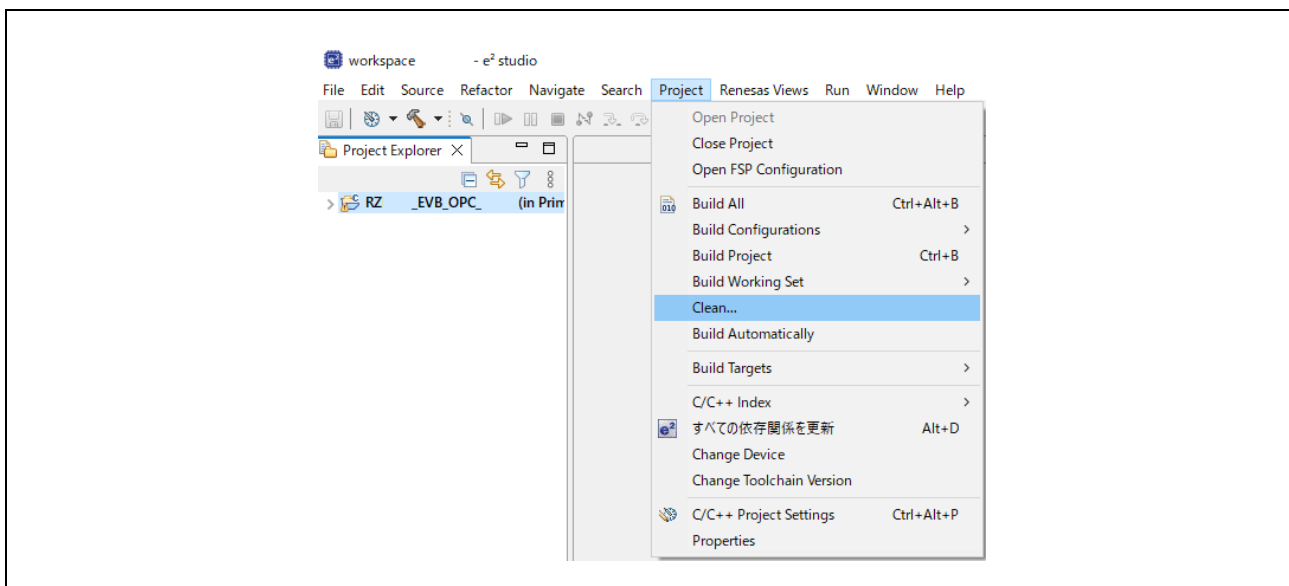


Figure 6.52. Open project Clean...

以下のように選択して、Clean をクリックするとビルドを開始します。

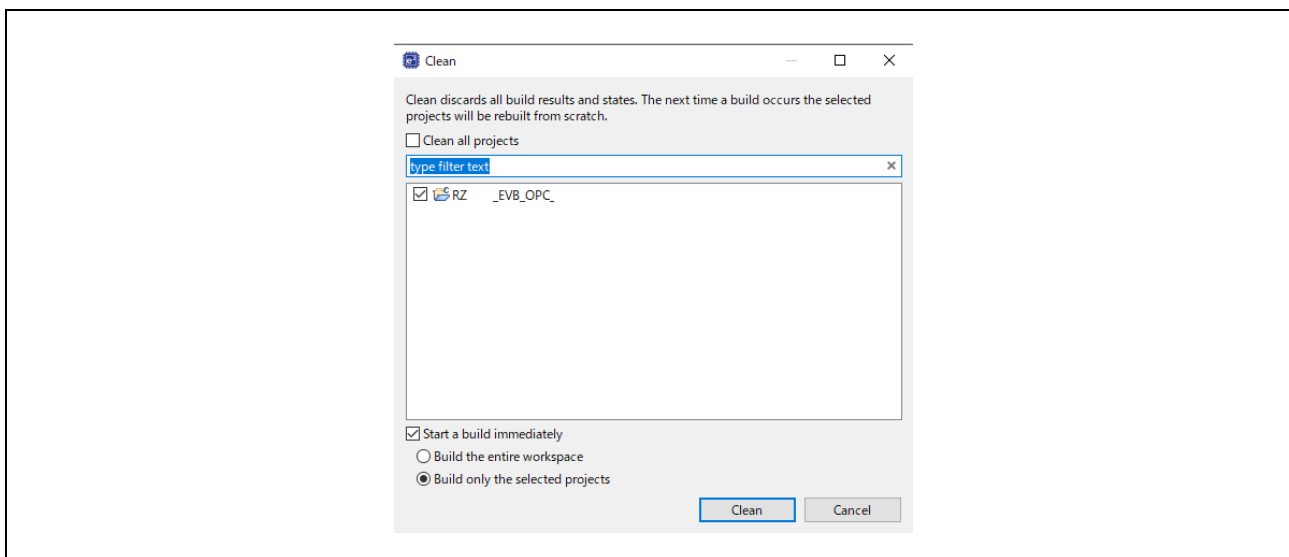


Figure 6.53. Clean and rebuild

6.2.2.2 Flash writing, debug

ビルド終了後のダウンロード手順を以下に示します。

Project Explore からプロジェクトを選択し、デバッグアイコンをクリックするとフラッシュメモリにプログラムをダウンロードします。

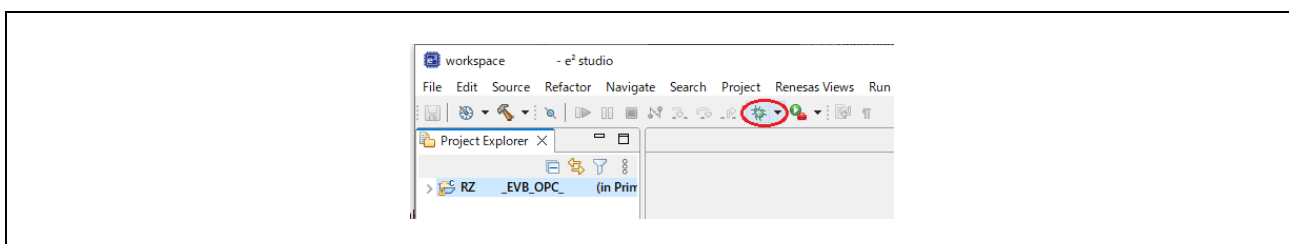


Figure 6.54. Run menu Debug As

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。

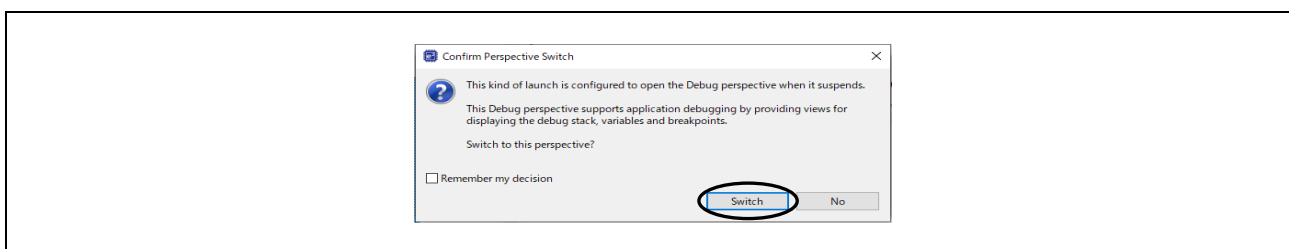



Figure 6.55. Perspective Switch

デバッガを使用せず評価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、resume  をクリックしてください。

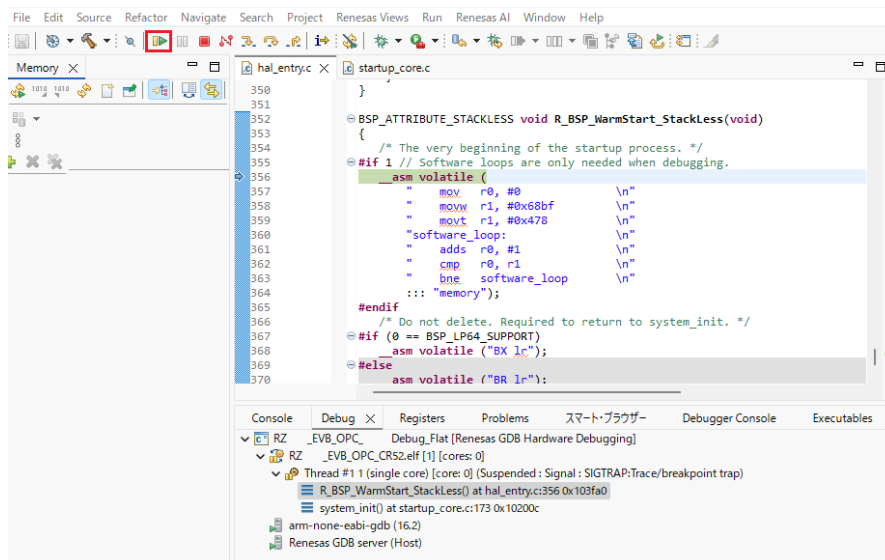


Figure 6.56. Debug view

7. Demonstration of the Sample Application

7.1 Application Behavior

本サンプルソフトは、OPC UA Server と、その Node を周期的に読み書きし動作するサンプルアプリケーションで構成されます。ソフトウェア構成の詳細仕様については 8.1 章を参照ください。

7.2 OPC UA Operation

本章では OPC UA 機能の動作確認方法を説明します。

7.2.1 Server

OPC UA Server 動作確認時の、それぞれのデバイスの役割は Figure 7.1 に示す通りです。

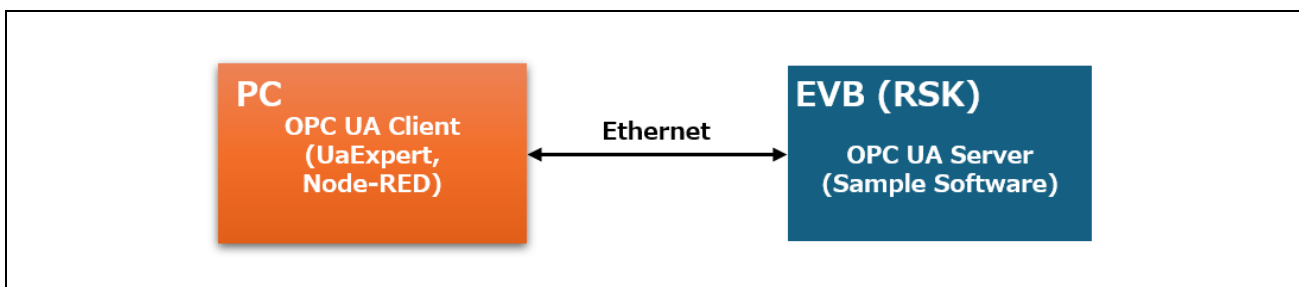


Figure 7.1. Connection Block Diagram for OPC UA Server

7.2.1.1 Without security function

(1) When using UaExpert

評価ボードの電源 ON 後、NTP 通信による時刻同期が完了するまで待ちます。電源 ON から 15 秒ほどで完了します。時刻同期が完了すると評価ボードの緑 LED (T2M, N2L, T2H:LED0, N2H:LED3) が点灯します。

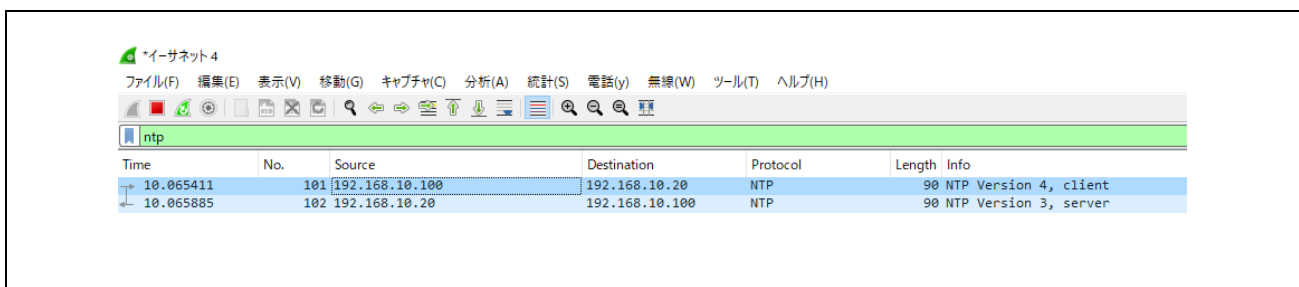


Figure 7.2. NTP Communication

UaExpert を起動し、ツールバーの  をクリックします。

Advanced タブの Endpoint Url に”opc.tcp://192.168.10.100:4840”を設定し、Security Setting はどちらも None を、Authentication Settings は”Anonymous”を選択します。”Connect Automatically”にチェックを入れて、最後に OK をクリックします。

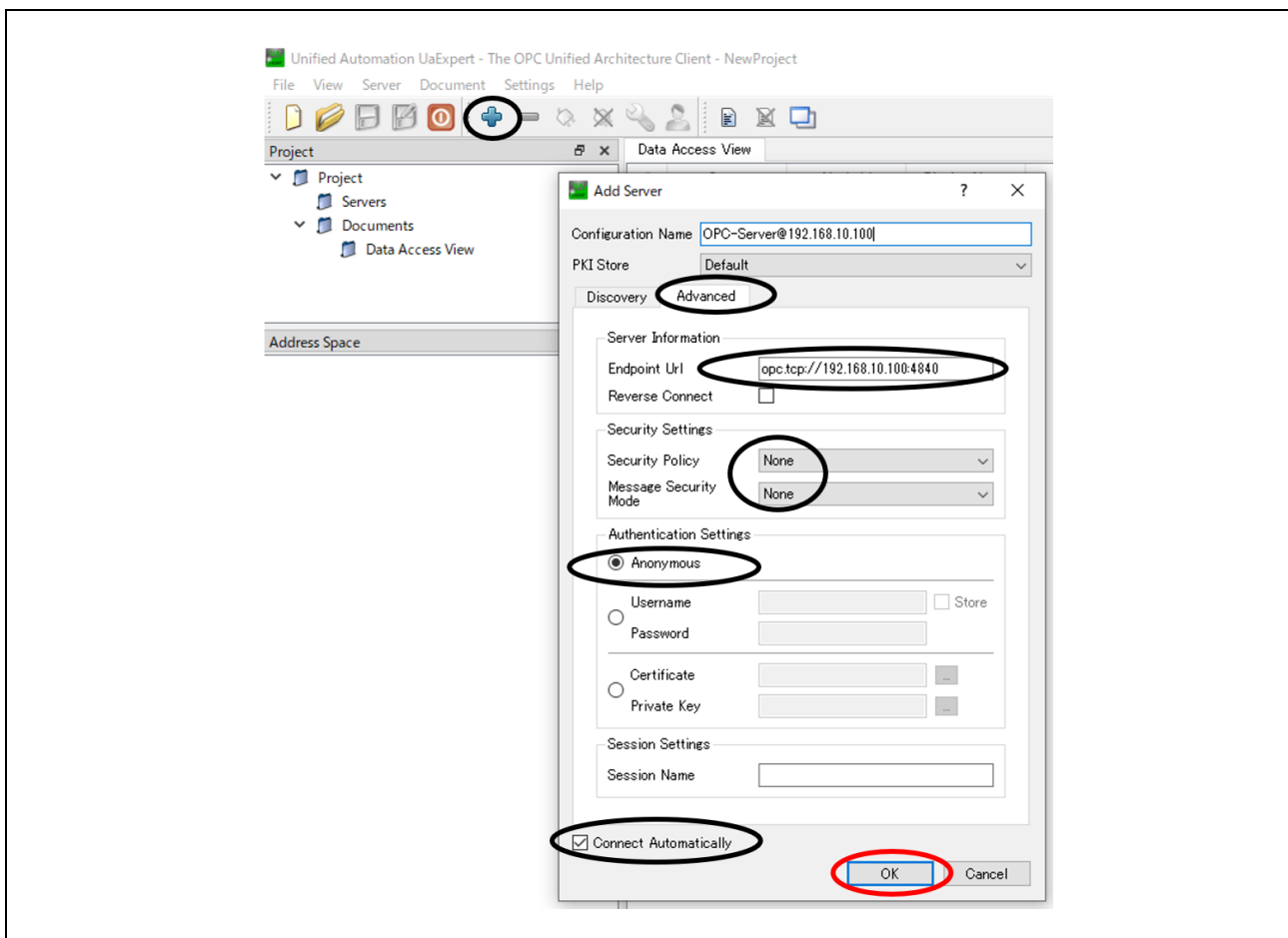



Figure 7.3. UaExpert Add server

OPC UA Server が接続されると、Project ウィンドウに接続されたことを示すアイコン  が表示されま
す。Address Space ウィンドウの Object ツリーに表示された Renesas_RZ は OPC UA Server のノードで
す。

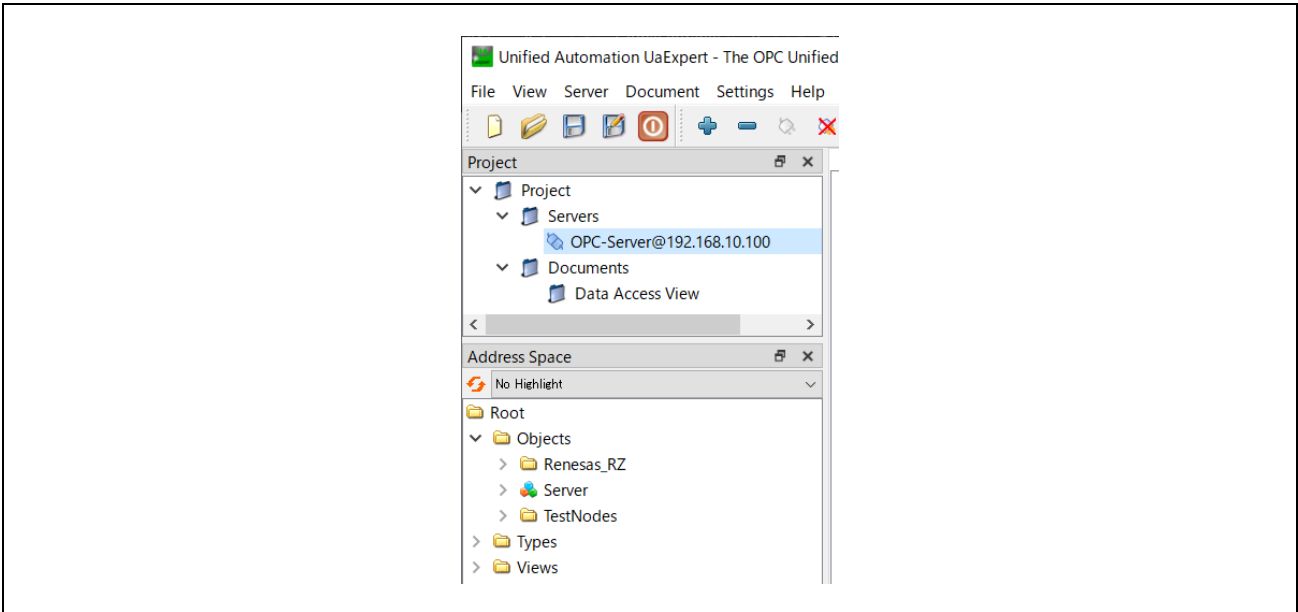


Figure 7.4. UaExpert OPC UA Server connection

Figure 7.5 の通り、Address Space ウィンドウの Root>Objects>Renesas_RZ を Data Access View ウィ
ンドウにドラッグアンドドロップすると、各種 Node の値が読み出しできます。それぞれの Node の仕様につ
いては、8.1 章を参照ください。

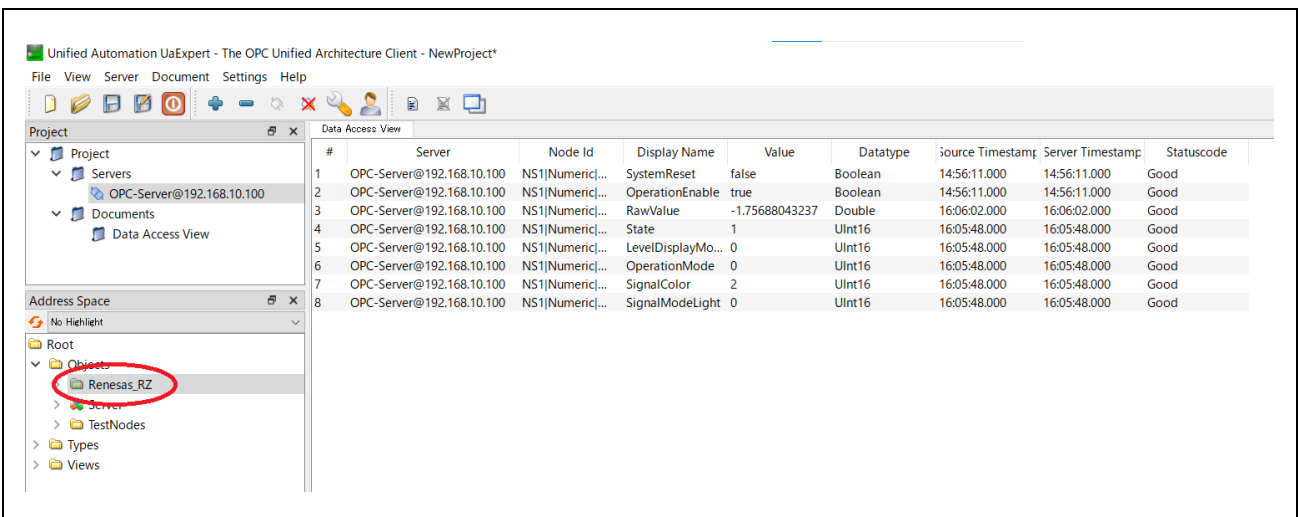


Figure 7.5. Reading Nodes

(2) When using Node-RED

5.8 章と同様の手順で、Node-RED を立ち上げます。メニューボタンから Import をクリックします。

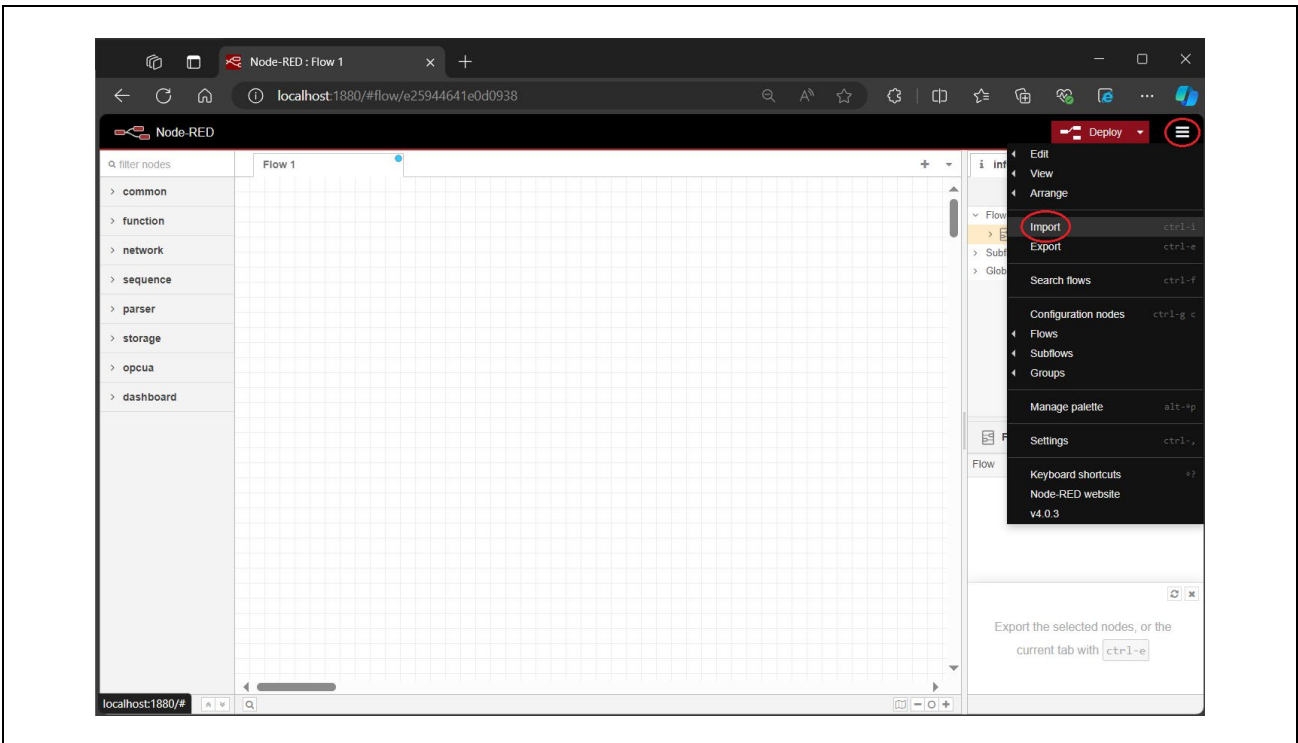


Figure 7.6. Import

“select a file to import”をクリックし、パッケージに同梱された “node-red_opcua_v2.1.json” を選択後、Import ボタンをクリックします。

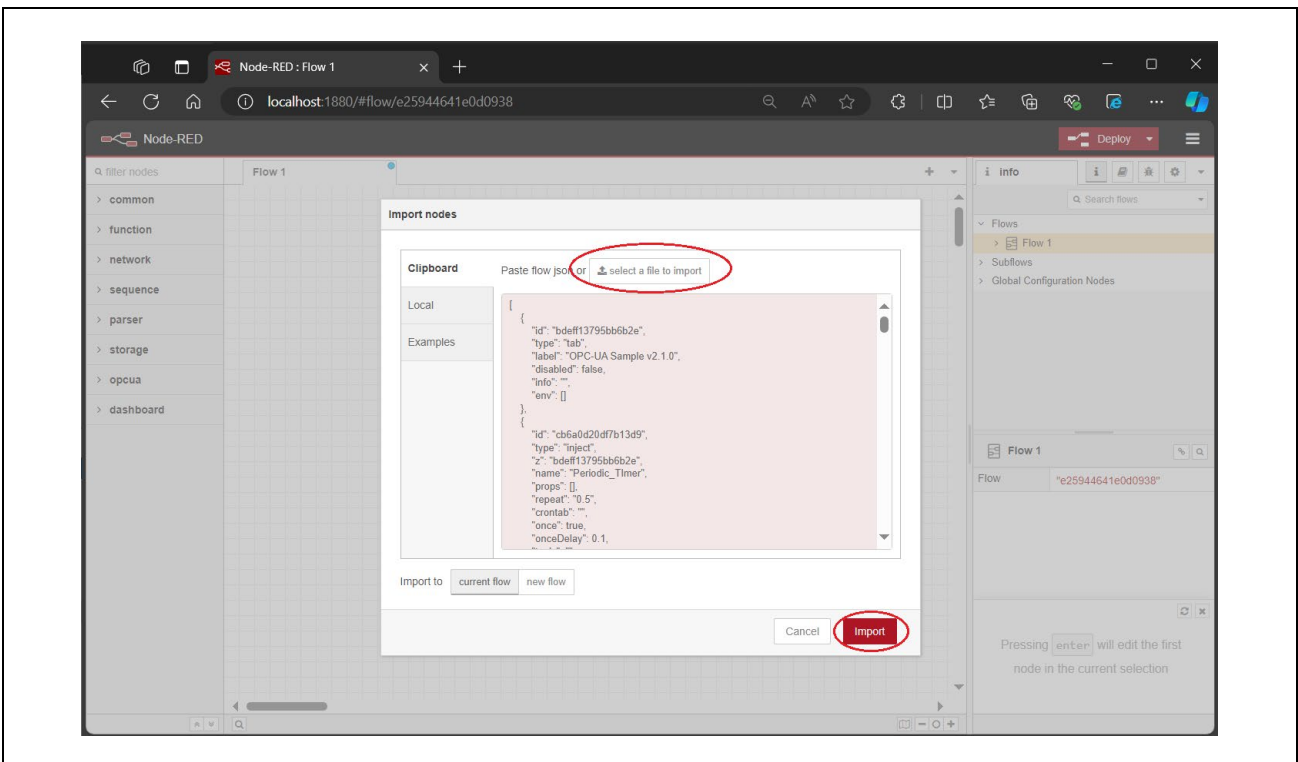


Figure 7.7. Import .json file

フローをインポート後に、画面右上の Deploy をクリックします。

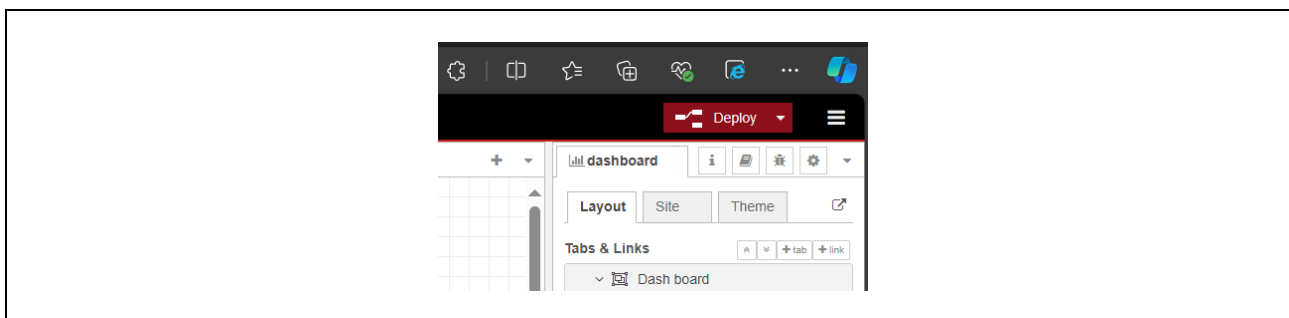


Figure 7.8. Deploy

OPC UA Server とのコネクトに成功すると、OPC UA Client Node の下に”active reading”と表示されます。

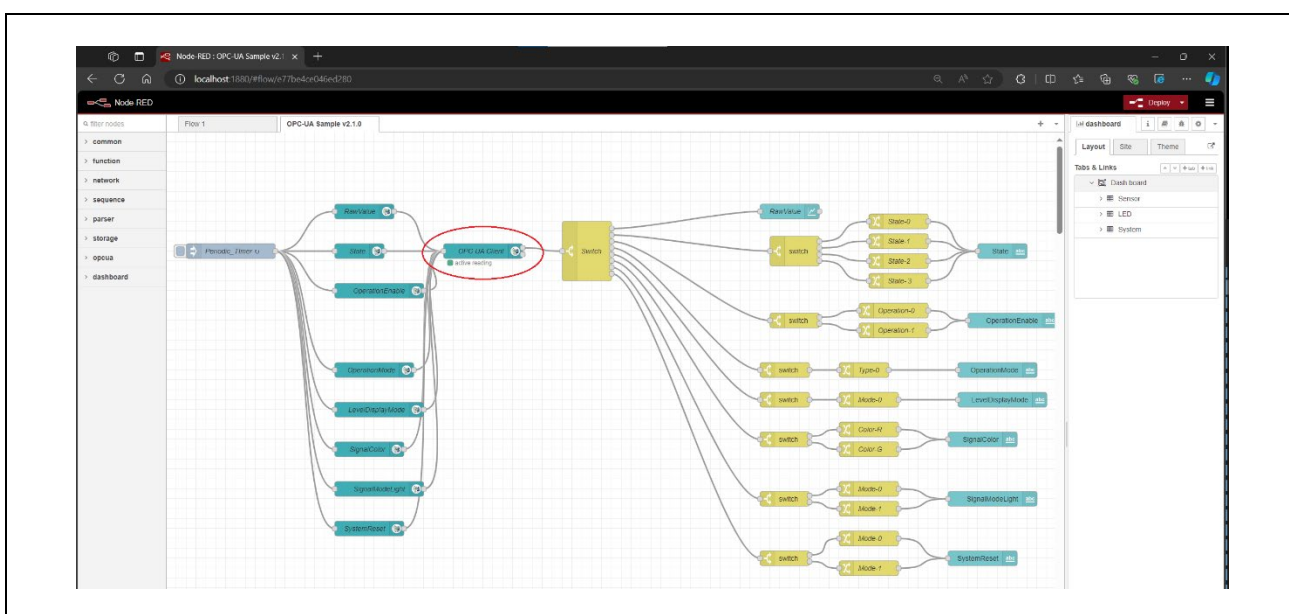


Figure 7.9. Connection Successful

画面右上の ▾ アイコンから Dashboard をクリックします。

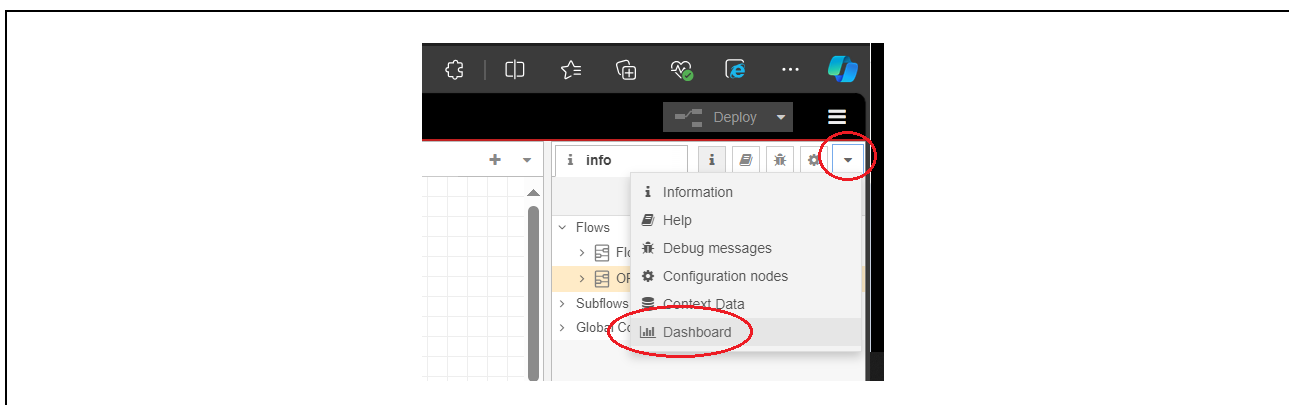



Figure 7.10. Open Dashboard (1)

画面右上の  アイコンをクリックします。

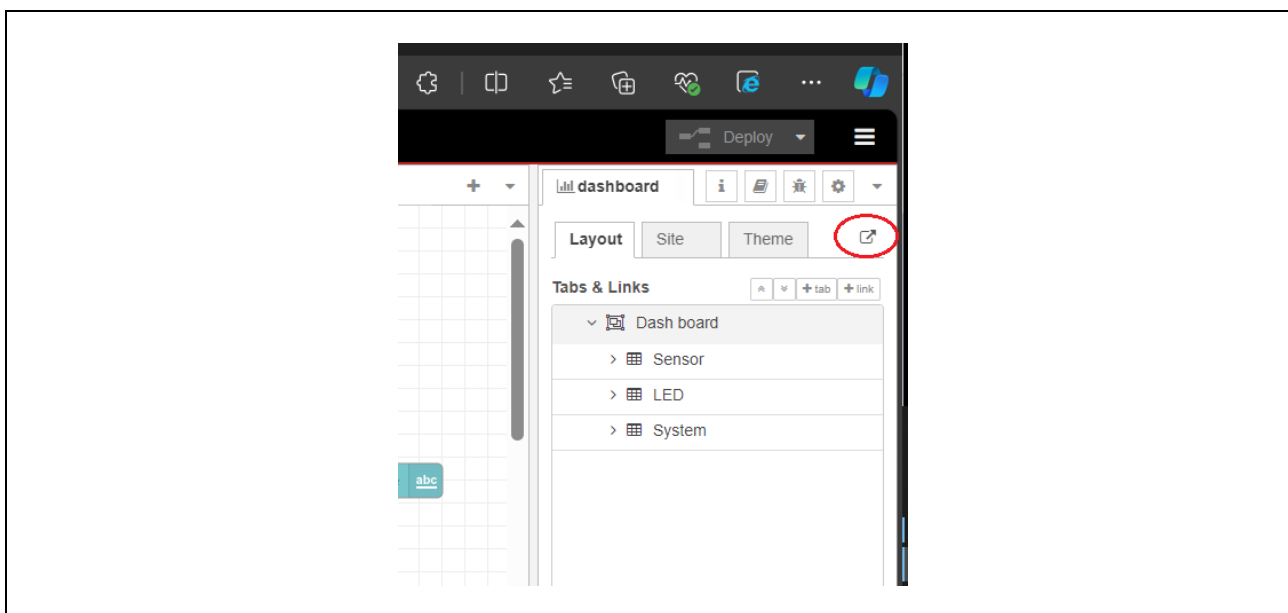


Figure 7.11. Open Dashboard (2)

Dashboard が別ウィンドウで開き、それぞれの Node の値が表示されます。

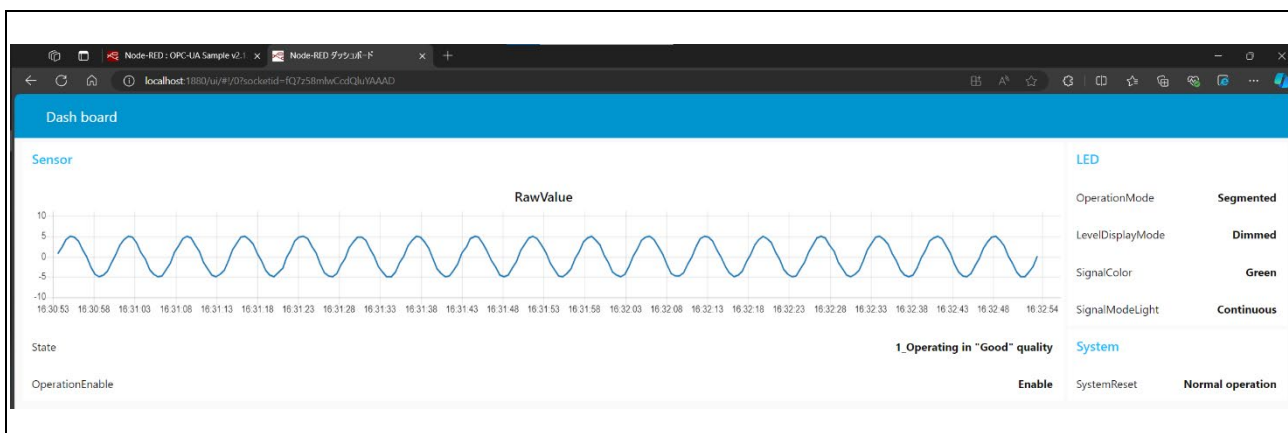


Figure 7.12. Dashboard

7.2.1.2 With security function

セキュリティ機能を使用する場合、サーバの秘密鍵と公開鍵証明書、クライアントの公開鍵証明書を保存した USB フラッシュドライブを評価ボードに接続するか、ENABLE_FREERTOS_PLUS_FAT:0 に設定しビルドしたソフトを使用してください。

USB フラッシュドライブに書き込みが必要なファイルは、“cert_server.der”, “pkey_server.der”, “cert_client_1.der”, “cert_c_usr_1.der” の 4 つです。ファイルは “<USB_ROOT>/passw0rd” フォルダに保存してください。本プロジェクトの TFTP 機能を使用して USB フラッシュドライブにファイル書き込みを行う場合は、9.2 章を参照してください。書き込み後はシステムリセットか、評価ボードの電源 OFF→ON を実施してください。

評価ボードの電源 ON 後、NTP 通信による時刻同期が完了するまで待ちます。電源 ON から 15 秒ほどで完了します。時刻同期が完了すると評価ボードの緑 LED (T2M, N2L, T2H:LED0, N2H:LED3) が点灯します。

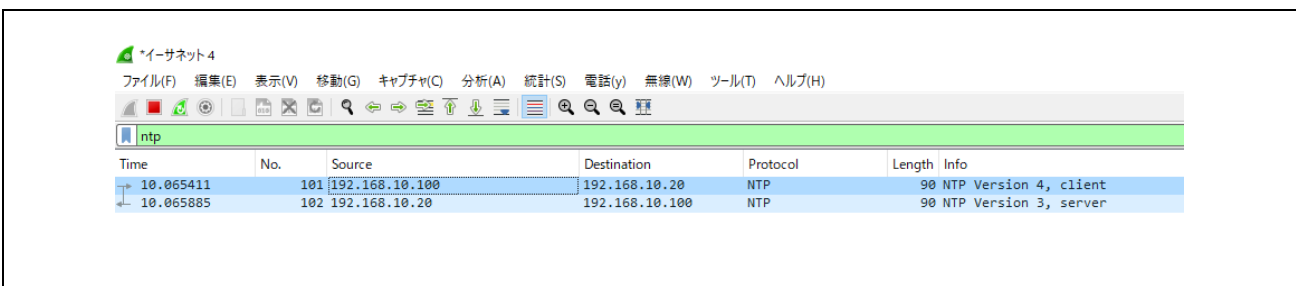



Figure 7.13. NTP Communication

パッケージに同梱されている“keyfiles.zip”から、“uaexpert.der”を \unifiedautomation\uaexpert\PK\own\certs フォルダに、“uaexpert_key.pem”を \unifiedautomation\uaexpert\PK\own\private フォルダにそれぞれコピーします。

UaExpert で、ツールバーの  をクリックします。Advanced タブの Endpoint Url に“opc.tcp://192.168.10.100:4840”を設定し、Security Setting は任意の Security Policy, Security Mode を選択してください。Authentication Settings は 任意のものを選択してください。Username/Password を選択した場合は、Table 2.1 に記載されたユーザ名とパスワードのペアを入力してください。Certificate を選択した場合は、パッケージに同梱されている“keyfiles.zip”から、“uaexpert_usr.der” と “uaexpert_key_usr.pem”を選択します。“Connect Automatically” にチェックを入れて、最後に OK をクリックします。

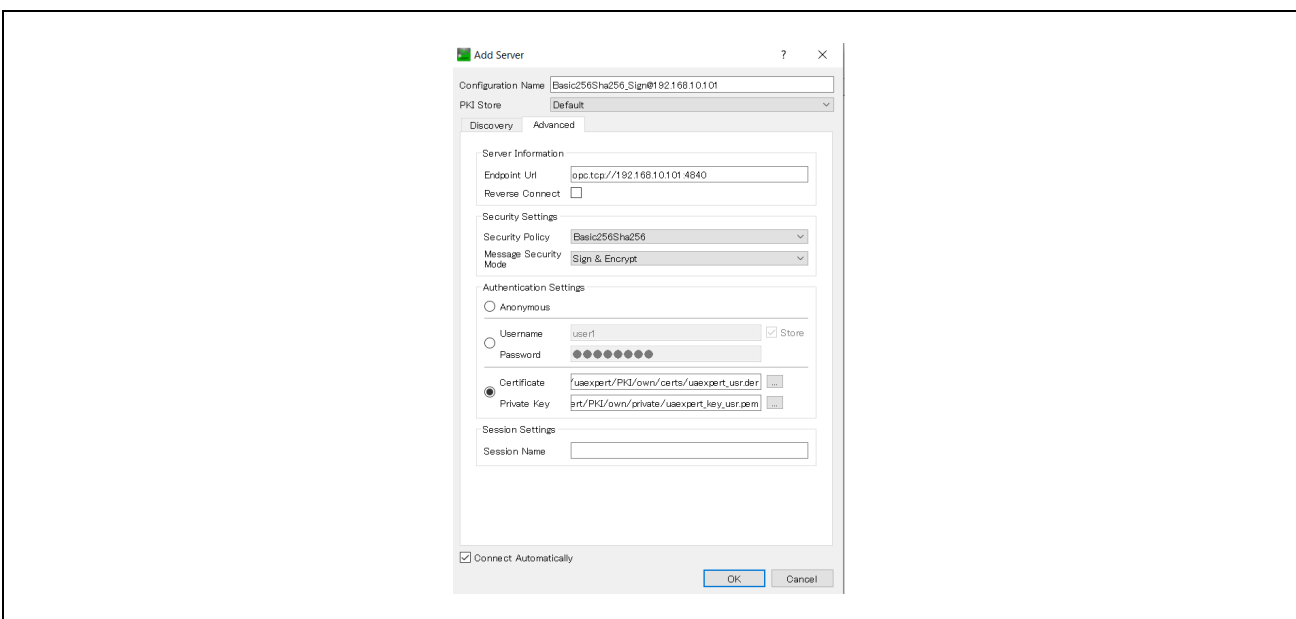


Figure 7.14. UaExpert Add server

初回接続時のみ、Figure 7.15 のポップアップが表示されます。"Trust Server Certificate"、"Continue"をクリックしてください。

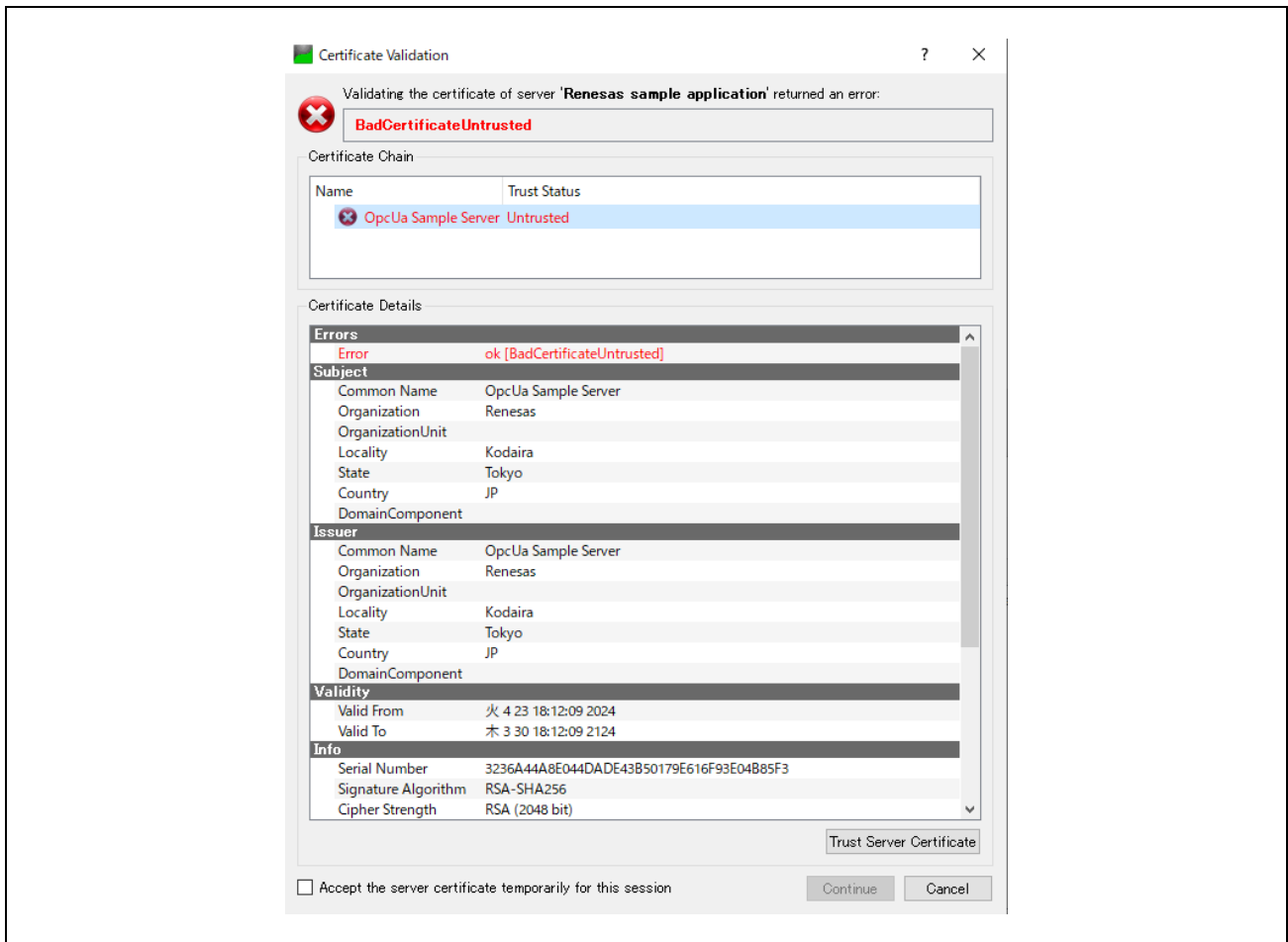


Figure 7.15. Certificate Validation

Connect時に、タイムアウトが発生する場合、"Settings" > "Configure UaExpert..."から、"General.ConnectTimeout"の値を大きくしてください。

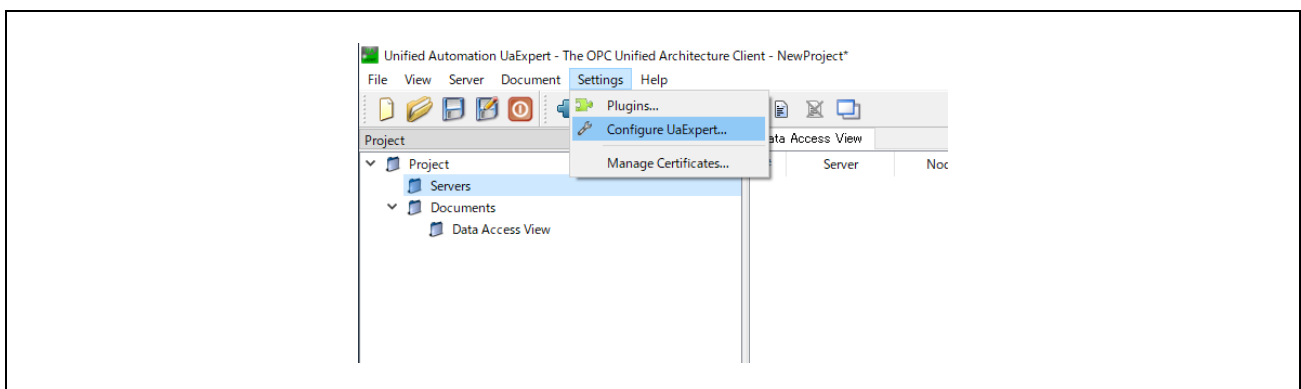


Figure 7.16. Configure UaExpert

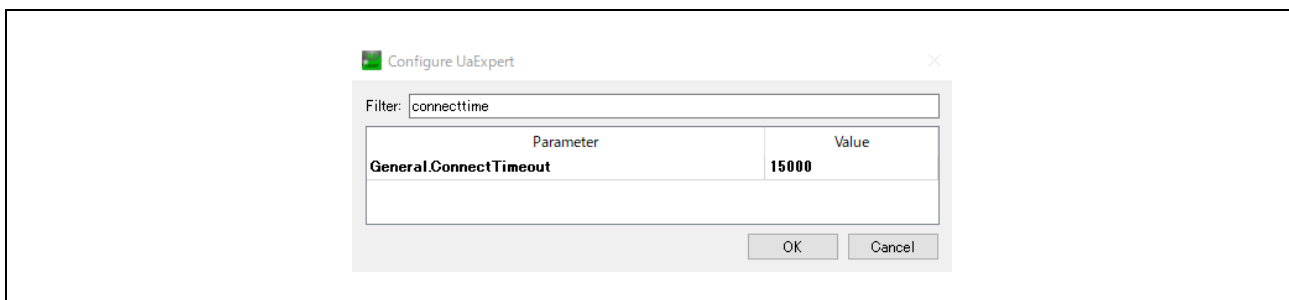


Figure 7.17. UaExpert ConnectTimeout Setting

接続後の動作確認は、“7.2.1.1 章 Without security function” と同様です。

7.2.2 Publisher

Layout: Dynamic Layout, Security Policy: PubSub-Aes128-CTR の場合を例として動作確認を行います。OPC UA Publisher 動作確認時の、それぞれのデバイスの役割は Figure 7.18 に示す通りです。

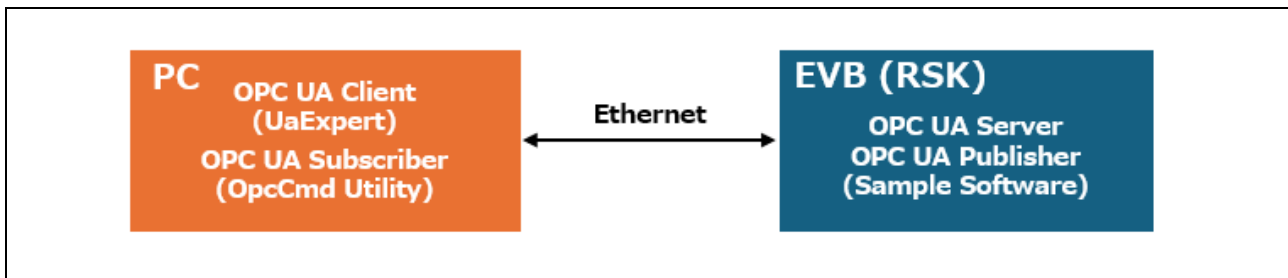


Figure 7.18. Connection Block Diagram for OPC UA Publisher

評価ボードを電源 ON した後、コマンドプロンプトを立ち上げ、OpcCmd Utility の展開先に移動します。そこで下記コマンドを入力してください。本コマンドは、AES128 で暗号化され”opc.udp://239.0.0.1”宛てに Publish されたメッセージを Subscribe、指定した鍵を使用して復号化するためのものです。コマンド内の”key”で指定された鍵は、上位 32 Byte が SigningKey、次の 16 Byte が EncryptingKey、次の 4 Byte が KeyNonce を表しています。

```
OpcCmd uaSubscriber subscribeDataSet opc.udp://239.0.0.1:4840 --SecurityKeyServiceUri
static:?key=000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223
2425262728292A2B2C2D2E2F30313233 --SecurityGroupId TestGroup --SecurityMode
SecuritySignAndEncrypt -cni "(ネットワークインターフェース名)"
```

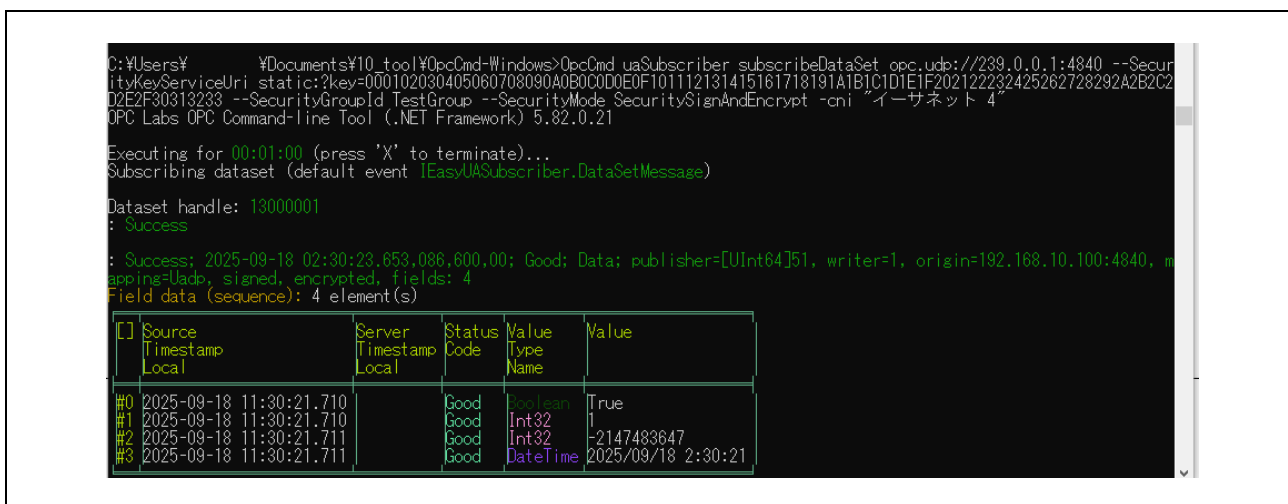


Figure 7.19. Command Execution Result

7.2.3 Subscriber

Layout: Dynamic Layout, Security Policy: PubSub-Aes128-CTR の場合を例として動作確認を行います。OPC UA Subscriber 動作確認時の、それぞれのデバイスの役割は Figure 7.20 に示す通りです。

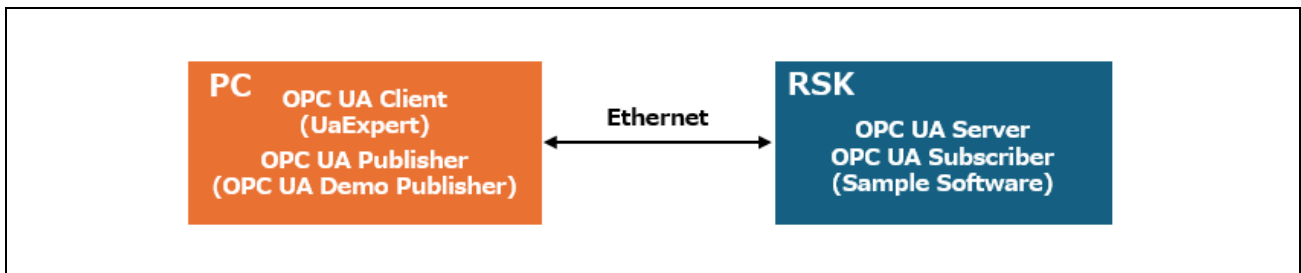


Figure 7.20. Connection Block Diagram for OPC UA Subscriber

評価ボードを電源 ON した後、コマンドプロンプトを立ち上げ、UADemoPublisher の展開先に移動します。そこで下記コマンドを入力してください。本コマンドは、指定した鍵を使用して AES128 で暗号化されたメッセージを"opc.tcp://239.0.0.1"宛てに Publish するためのものです。コマンド内の"key"で指定された鍵は、上位 32 Byte が SigningKey、次の 16 Byte が EncryptingKey、次の 4 Byte が KeyNonce を表しています。

```
UADemoPublisher publish -cru opc.tcp://239.0.0.1 -ewg DynamicLayoutGroup -edsw SimpleWriter --
SecurityKeyServiceUri
static:?key=000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223
2425262728292A2B2C2D2E2F30313233 --SecurityGroupId TestGroup --SecurityMode SignAndEncrypt -
cni "(ネットワークインターフェース名)"
```

```
C:\Users% ¥Documents¥10_tool¥UADemoPublisher>UADemoPublisher publish -cru opc.tcp://239.0.0.1 -ewg DynamicLayoutGroup -edsw SimpleWriter --SecurityKeyServiceUri static:?key=000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F30313233 --SecurityGroupId TestGroup --SecurityMode SignAndEncrypt -cni "イーサネット 4"
OPC Labs OPC-UA Demo Publisher (.NET Framework) 5.73.0.5
Loading configuration...
TransportProfileUri: https://opcfoundation.org/UA-Profile/Transport/pubsub-udp-udp
PublishedDataSets:
Simple: DataSetClassId=aaa79794-1af7-4f96-8401-4096cd1d8908, 4 field(s)
Simple-Proxy: DataSetClassId=aaa79794-1af7-4f96-8401-4096cd1d8908, 4 field(s)
AllTypes: DataSetClassId=511be198-9ade-4a72-8d75-1f1bebe173d6, 8 field(s)
MassTest: DataSetClassId=9b976b7b-0cb7-46c3-a715-b979884b55ae, 100 field(s)
AllTypes-Dynamic: DataSetClassId=cc7cb5f4-4272-45c2-9a4d-f85a8b331f6a, 16 field(s)
Connections (enabled objects only):
FixedLayoutConnection: [opc.tcp://239.0.0.1 | イーサネット 4] PublisherId=[UInt16]30
DynamicLayoutConnection: [opc.tcp://239.0.0.1 | イーサネット 4] PublisherId=[UInt64]31
DynamicLayoutGroup: WriterGroupId=102, MaxNetworkMessageSize=65507, PublishingInterval=500, KeepAliveTime=2000
SimpleWriter: DataSetWriterId=1, DataSetName=Simple, KeyFrameCount=25
FlexibleLayoutConnection: [opc.tcp://239.0.0.1 | イーサネット 4] PublisherId=[String]32
Executing for Infinite (press 'X' to terminate)...
Starting to publish...
Progress: 00:03:48 Messages produced: DynamicLayoutConnection 445, FixedLayoutConnection 0, FlexibleLayoutConnection 0
```

Figure 7.21. Command Execution Result

その後、UaExpert から"Subscribed_Variables_Simple" Node が存在することを確認します。

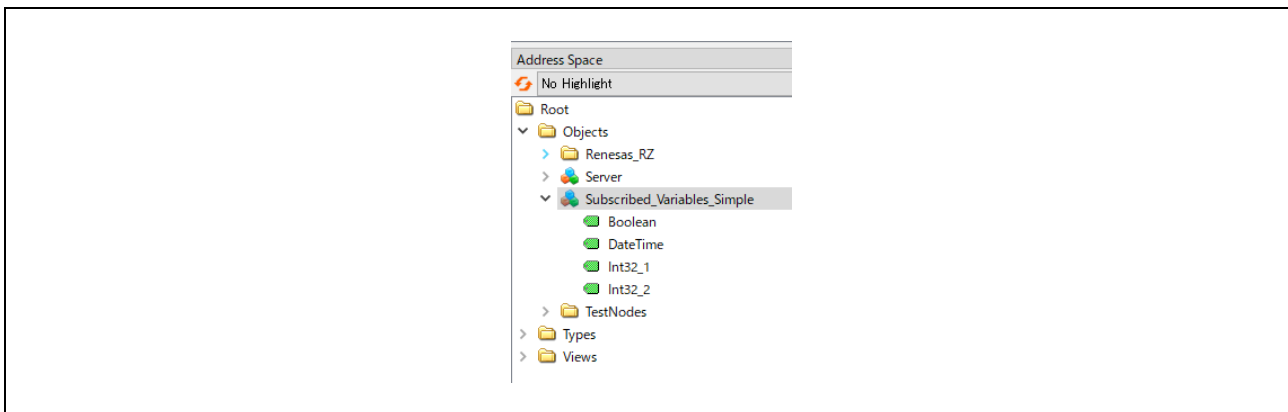


Figure 7.22. “Subscribed_Variables_Simple” Node

この Node を Data Access View で確認してみると、定期的に値が更新されており、正常に Subscribe できていることが確認できます。

Data Access View						
#	Server	Node Id	Display Name	Value		
1	SecurityNone@...	NS1 Numeric 50000	Boolean	true		Boo
2	SecurityNone@...	NS1 Numeric 50003	DateTime	2025-09-18T02:31:34.606Z		Da
3	SecurityNone@...	NS1 Numeric 50001	Int32_1	13		Int:
4	SecurityNone@...	NS1 Numeric 50002	Int32_2	2787		Int:

Figure 7.23. Check the Node Values

8. Software Specifications

8.1 Software Structure

8.1.1 Overview of Software Components

Figure 8.1 は本サンプルソフトのアーキテクチャ図を示します。本サンプルソフトは OS に FreeRTOS を使用しています。

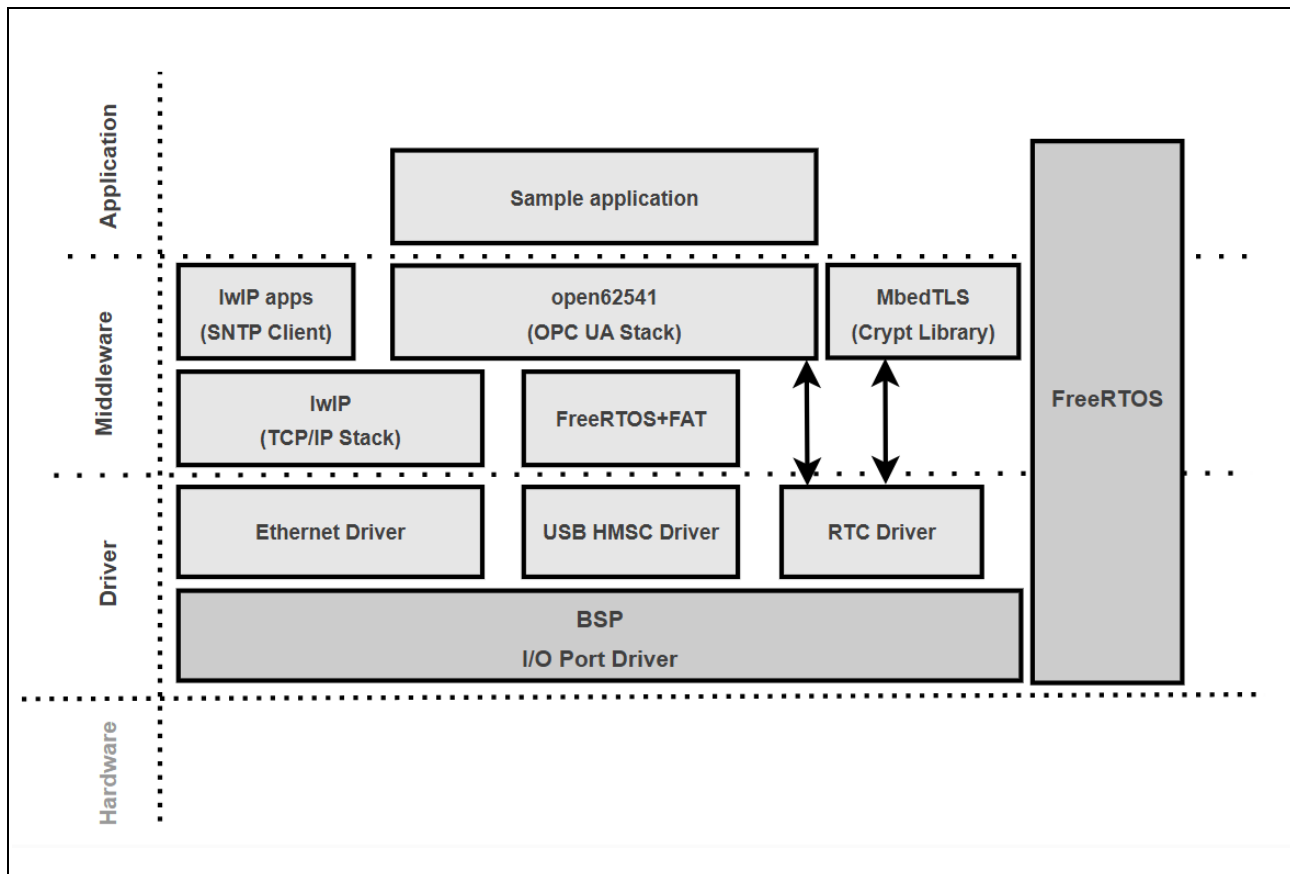


Figure 8.1. Software Architecture Diagram

8.1.2 Overview of Each Component

8.1.2.1 Driver

本サンプルソフトで使用されているドライバはFSPによって生成されています。使用しているFSPスタックはTable 8.1に示す通りです。各々の詳細な設定についてはSmart configurator 上でご確認ください。

Table 8.1. FSP Stacks

Stack name	Purpose
g_ioport I/O Port Driver	Pin Settings, Control
g_ether0 Ethernet	Ethernet communication
g_uart0 UART	Serial communication for debug
FreeRTOS Heap 4	Allocating heap for FreeRTOS
g_rtc0 Realtime Clock	Realtime clock
FreeRTOS+FAT (USB Driver)	USB communication and Filesystem

8.1.2.2 lwIP (TCP/IP Stack)

本サンプルソフトはTCP/IPスタックとしてlwIPを使用しています。使用しているベースバージョンは、STABLE-2_1_0にコミットリビジョン"79cd89f99d1032cc5375569e5b24c375b9d230fa"までの変更を反映したものです。Table 8.2は、デフォルト値から変更しているオプションを示します。

Table 8.2. lwIP settings

Option name	Option value	Description
LWIP_NETIF_API	1	Support netif api
LWIP_POSIX_SOCKETS_IO_NAMES	0	Disable POSIX-style sockets functions names
LWIP_IGMP	1	Turn on IGMP module
LWIP_MULTICAST_TX_OPTIONS	1	Enable multicast TX support
LWIP_TCP_KEEPALIVE	1	Enable TCP KeepAlive
TCP_KEEPIDLE_DEFAULT	30000	Set default KeepAlive timer 30 sec.
TCP_KEEPINTVL_DEFAULT	5000	Set default KeepAlive interval 5 sec.
TCP_KEEPCNT_DEFAULT	3	Set default counter for KeepAlive probes 3 times

8.1.2.3 Mbed TLS (Crypt Library)

本サンプルソフトは暗号ライブラリとしてMbed TLSを使用しています。使用しているベースバージョンはv3.6.4です。詳細は[Releases · Mbed-TLS/mbedtls](#)をご確認ください。

8.1.2.4 open62541 (OPC UA Stack)

本サンプルソフトでは、OPC UA Server のプロトコルスタックとして、open62541 を使用しています。open62541 の詳細は下記 Link をご参照ください。

[open62541](#)

・バージョン

本サンプルソフトで使用している open62541 の Base Version は以下です。

Base Version : v1.3.15

(Commit 3eed1a6d5c5b207c531b2d35ed88aa0a4a4541e5)

・ライセンス

open62541 のライセンス条件は、MPL v2.0 です。詳細は <https://www.mozilla.org/en-US/MPL/2.0/> 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。

・open62541 ファイル

open62541 を FreeRTOS + lwIP の環境で実行するため、下記 Link 先で紹介されている CMake を利用して open62541.c と open62541.h ファイルを生成する方法を用いています。詳細は、Appendix の 9.1 章をご参照ください。

[Building open62541 — open62541 1.3.0-dirty documentation](#)

・情報モデル

ベンダ間、業界間の相互運用を実現するため、OPC UA では情報モデル(Information Model) と呼ばれる統一されたデータモデルが xml ファイル形式で用意されています。OPC UA で共通に利用するビルトインモデルと、業界・団体毎に利用するコンパニオンモデル、更にベンダ毎に利用することが可能なベンダ固有のモデルが含まれます。(Figure 8.2)

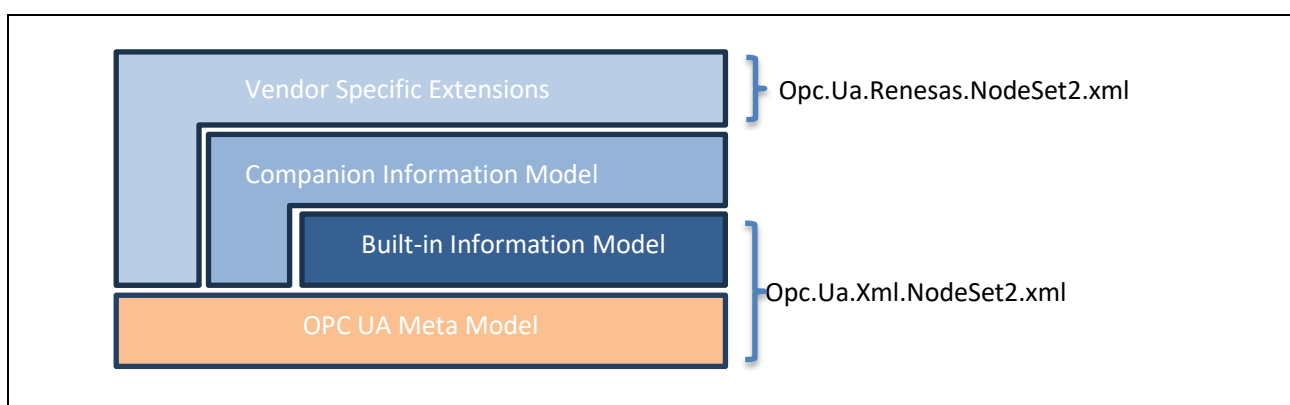


Figure 8.2. Information Model

8.1.2.5 Sample Application

本プロジェクトは疑似センサとして動作し、周期的に OPC UA Node に書き込み、読み出しを行います。状態に応じて疑似センサの出力値は制御され、評価ボードに搭載された LED の点灯状況も変化します。

評価ボードの起動後、NTP による時刻同期が行われると State0 から State1 に遷移し、疑似センサの処理が開始されます。疑似センサの取得値は、State1 の際は正弦波状に変化し、State2 の際は正弦波にランダムなノイズを乗せた値となり、“RawValue” Node に書き込みされます。State1 と State2 間の状態遷移は評価ボードのスイッチを押すことで行われます。

OPC UA Client から、“OperationEnable” Node に“0:Disable”が書きこまれると、State3 に移行し疑似センサの処理は停止します。再度“1:Enable”を書き込むことで直前の状態に戻り、疑似センサの処理が再開されます。

状態ごとの LED の点灯状況は下記の通りで、それぞれの値が“SignalColor” Node と“SignalModeLight” Node に書き込みされます。

State0 : 赤 LED 点灯、State1 : 緑 LED 点灯、State2 : 緑 LED 点滅、State3 : 赤 LED 点滅

これらの状態遷移は Figure 8.4 に示す通りです。

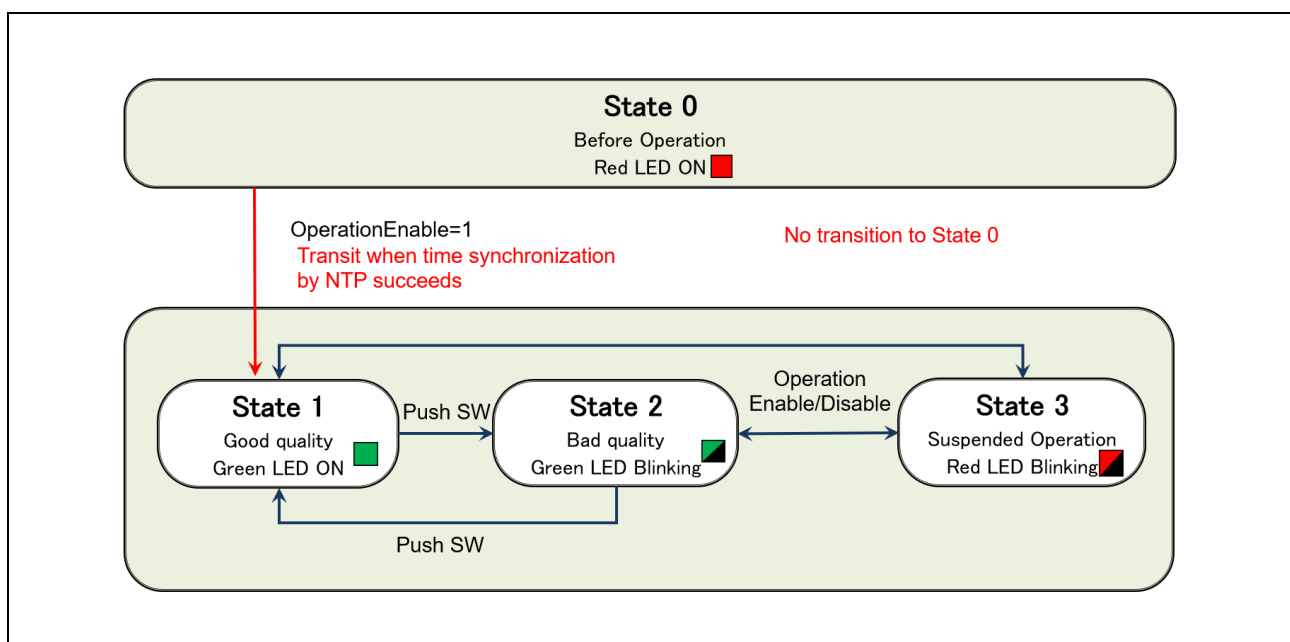


Figure 8.4. State Transition

また、“SystemReset” Node に“1:Execute SystemReset”を書き込まれるとシステムリセットが実行されま

す。上記のスイッチと LED はデバイスごとに Table 8.3 に示す通りです。

Table 8.3. Hardware Information

	Switch	Red LED	Green LED
RZ/T2M, N2L	SW1	LED2	LED0
RZ/T2H	SW11	LED5	LED0
RZ/N2H	SW3	LED4	LED3

8.2 Task Structure

この章では、サンプルソフトで作成・使用している RTOS タスクについて説明します。

8.2.1 Task List

Table 8.4 に本サンプルソフトにて使用しているタスクの一覧を示します。

Table 8.4. Task List

Task name	Priority	Trigger Type	Description
Main Thread	2	Startup	Initialize modules and create tasks.
Ethernet Monitor Task	3	Periodic (1sec.)	Monitor the port link status.
Ethernet Reader Task	5	Event (Receive packet)	Notify lwIP of received packets.
Ethernet Writer Task	5	Event (Packet Sending Req)	Transmit the packet upon receiving a transmission request.
OPC_TASK	5	Periodic (Dynamic control)	Initialize and run open62541.
SAMPLE_APP_TASK	2	Periodic (10msec.)	Run sample application.
lwIP	6	Event (Receive message)	Packet transmission task for lwIP.
lwIP Port Launcher Task	3	Event (Receive LinkStatus)	lwIP Link Status Management.
lwIP Port Receiver Task	5	Event (Receive message)	Packet reception task for lwIP. Notify the application of received packets.
LLDP_timer_1, 2	2	Periodic (1sec.)	Manage the LLDP agent and periodically send LLDP packets.
LLDP_receive_1, 2	2	Event (Receive packet)	Perform reception processing of LLDP frames
LLDP_get_mib_1, 2	2	Event (Read Req)	Get LLDP-MIB information
LLDP_set_mib_1, 2	2	Event (Write Req)	Set LLDP-MIB information

Table 8.5. RTOS Max Priority

Item	Value
Max Priority	10

8.2.2 Task Operation

本章では、本サンプルソフトの主な処理シーケンスについて説明します。

8.2.2.1 Startup

Figure 8.5 にスタートアップ処理のシーケンス図を示します。Main Thread タスクはスタートアップ処理にて生成される最初のタスクで、主にモジュールの初期化、他タスクの生成などを行います。

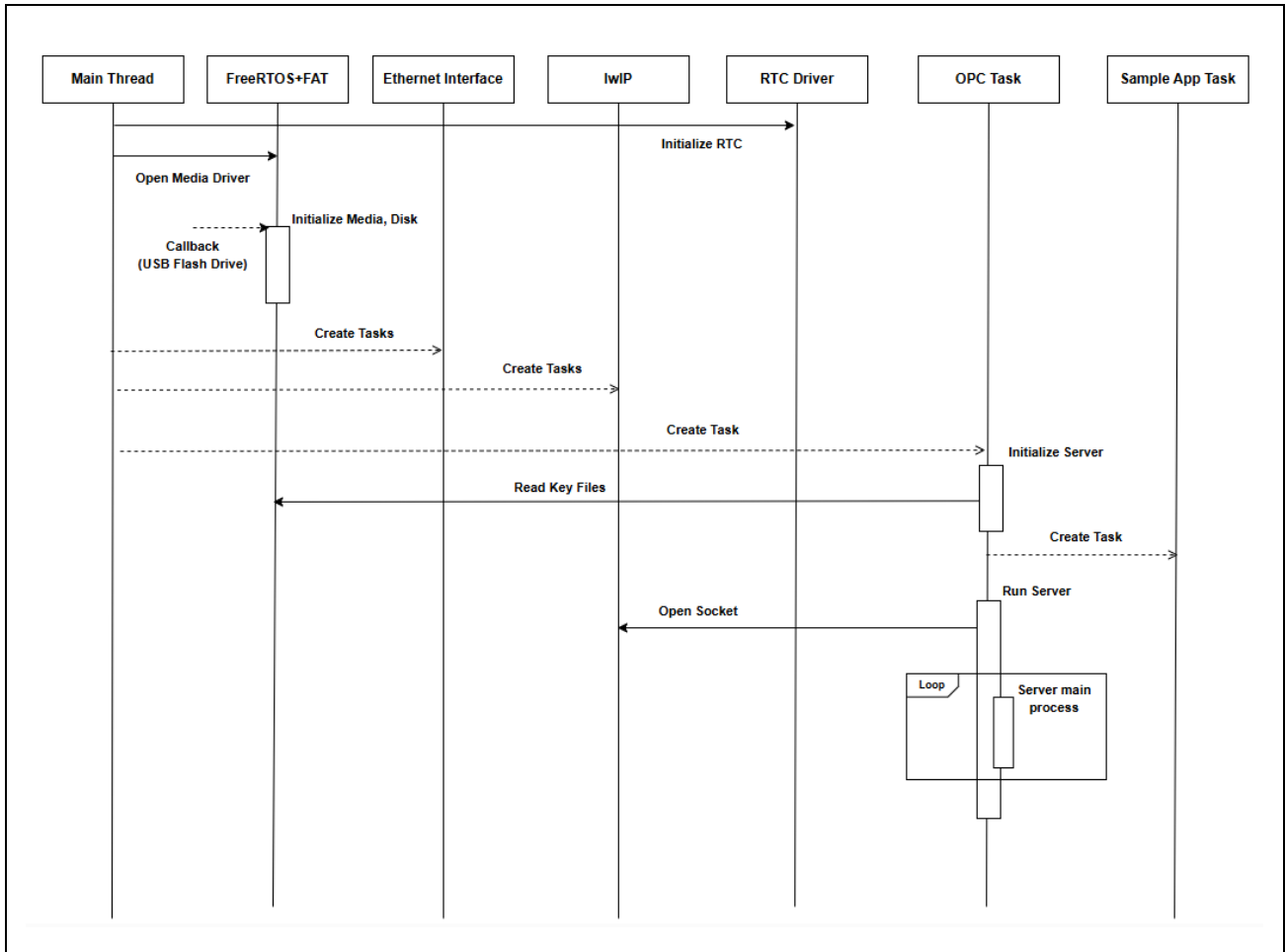


Figure 8.5. Startup

8.2.2.2 OPC UA communication

Figure 8.6に OPC UA の送受信処理のシーケンス図を示します。lwIP を通して送受信される OPC UA データは Mbed TLS を使用して暗復号化されます。

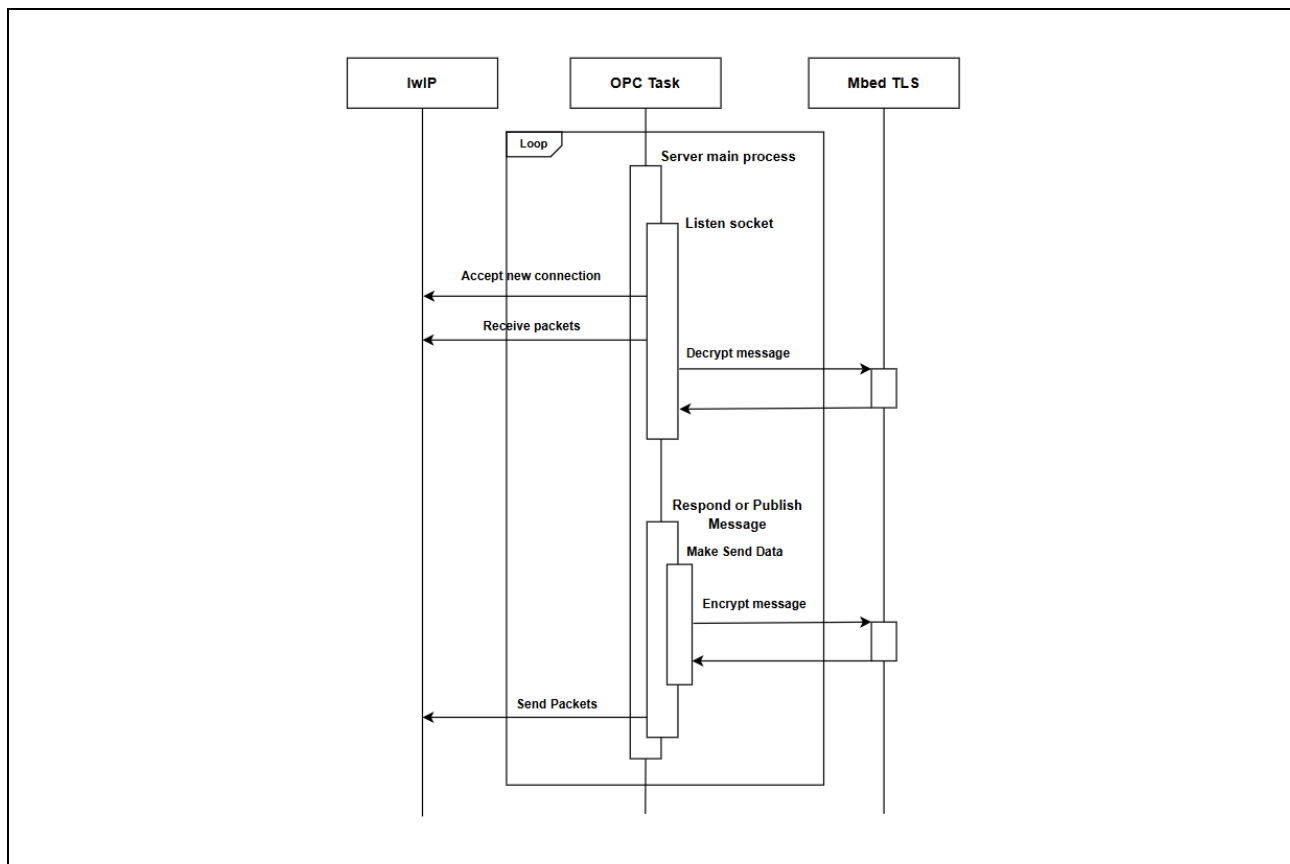


Figure 8.6. OPC UA communication

8.2.2.3 Sample application

Figure 8.7 にサンプルアプリケーションのシーケンス図を示します。サンプルアプリケーションは 8.1.2.5 章の仕様の通りにノードの値や LED 出力を制御します。

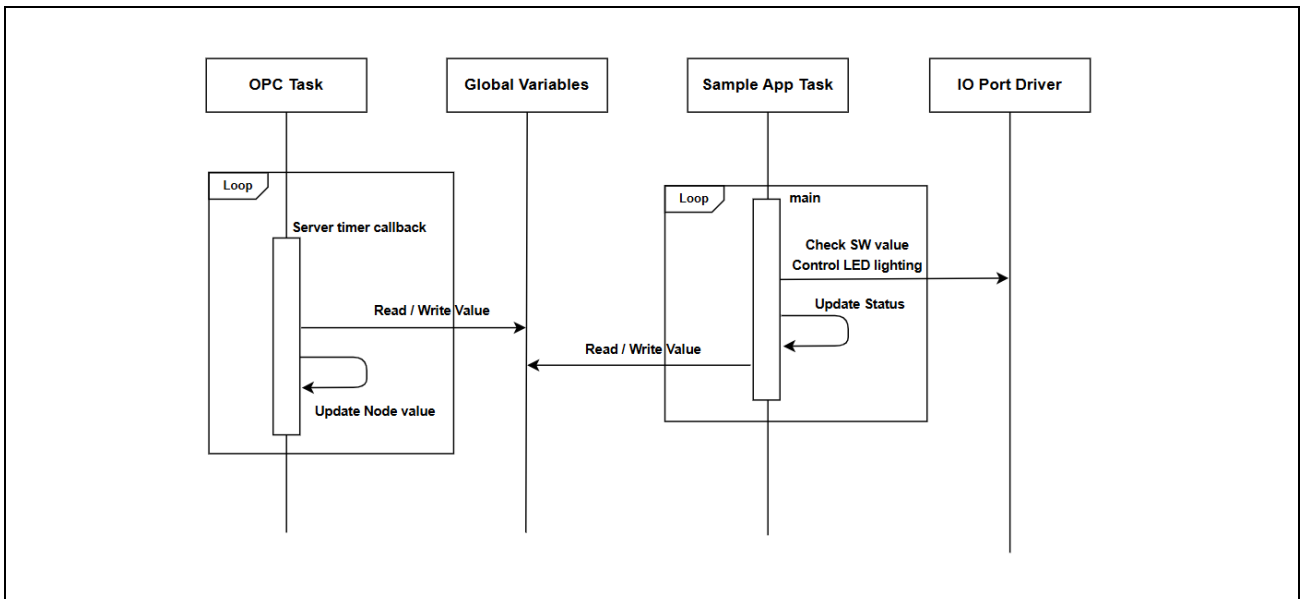


Figure 8.7. Sample application

8.2.2.4 Time synchronization

Figure 8.8 に時刻同期処理のシーケンス図を示します。SNTP 通信により NTP サーバから受け取った時刻情報は、RZ デバイスの機能である RTC によって管理され、各種機能によって使われます。

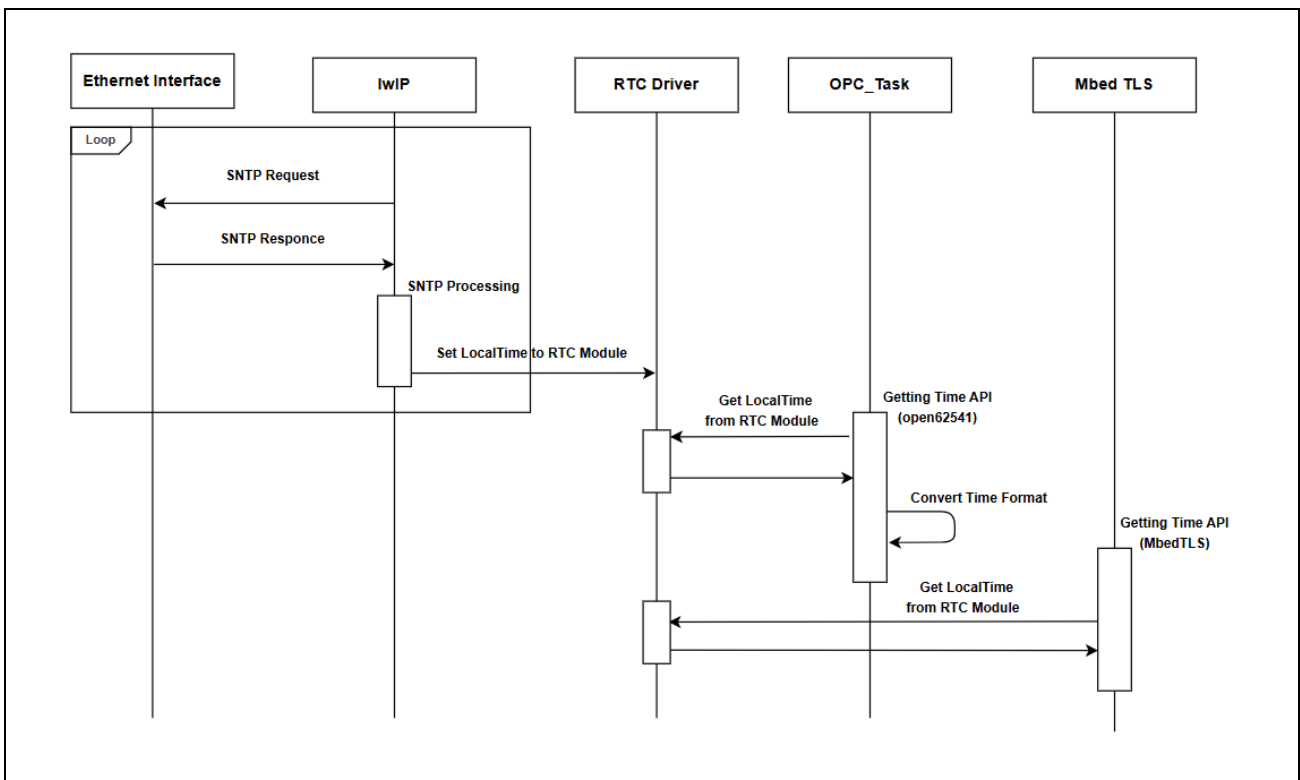


Figure 8.8. Time synchronization

8.3 Footprint (ROM and RAM Usage)

本章では、それぞれのプロジェクトのメモリ使用量を示します。

8.3.1 RZ/T2M project

8.3.1.1 e² studio

(1) Primary project

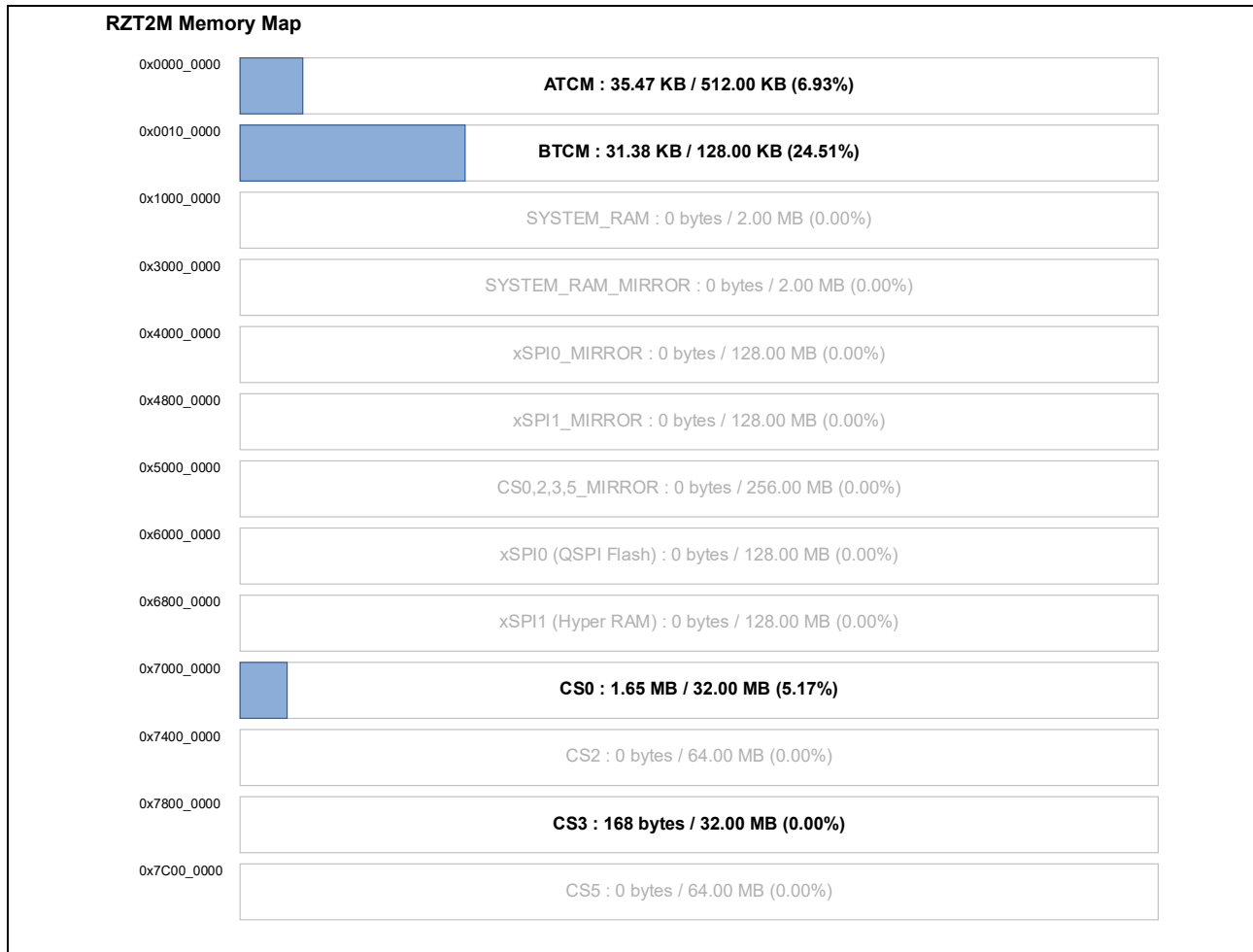


Figure 8.9. Memory footprint of the RZ/T2M Primary project (e² studio)

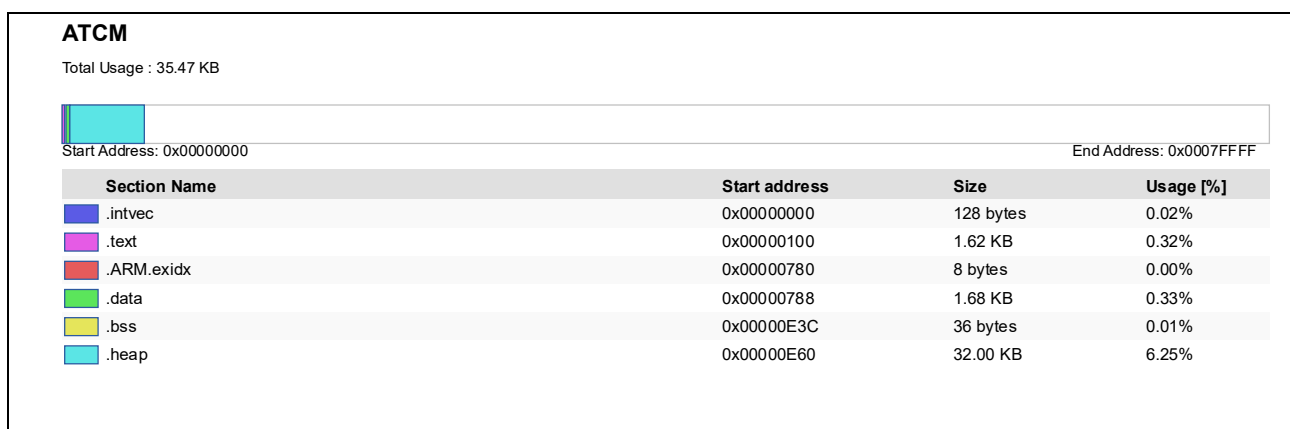


Figure 8.10. ATCM footprint of the RZ/T2M Primary project (e² studio)

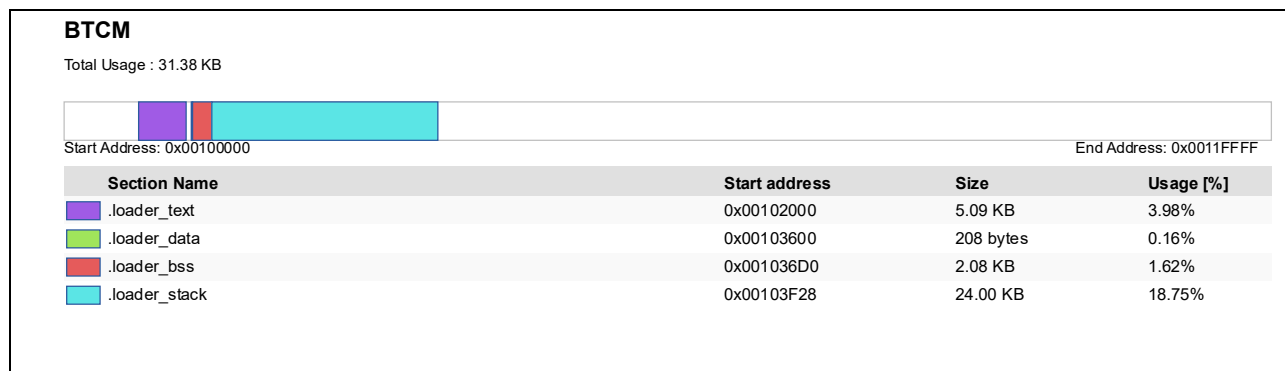


Figure 8.11. BTCM footprint of the RZ/T2M Primary project (e² studio)

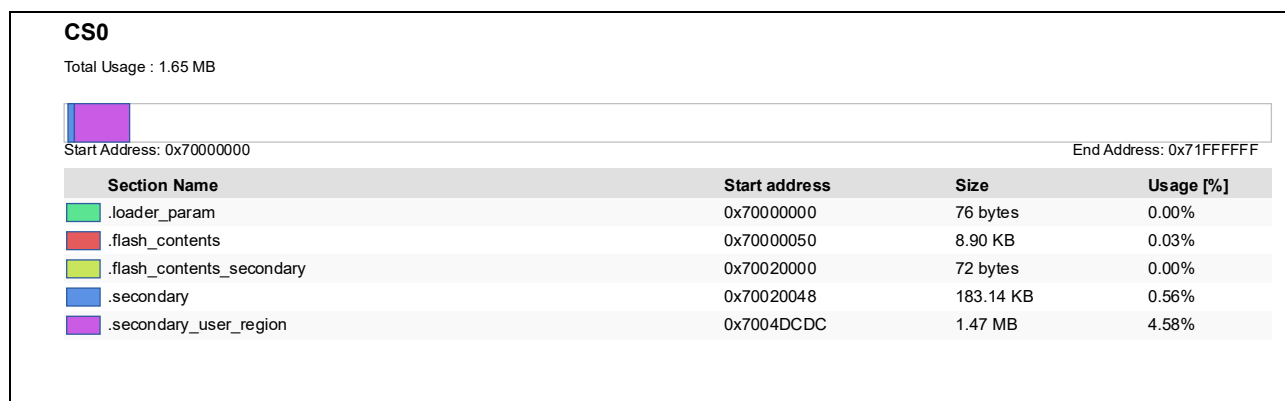


Figure 8.12. CS0 (NOR Flash) footprint of the RZ/T2M Primary project (e² studio)

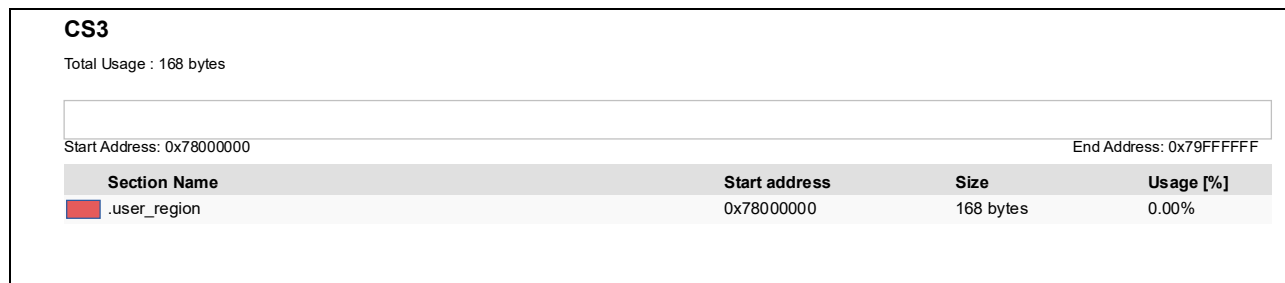


Figure 8.13. CS3 (SDRAM) footprint of the RZ/T2M Primary project (e² studio)

(2) Secondary project

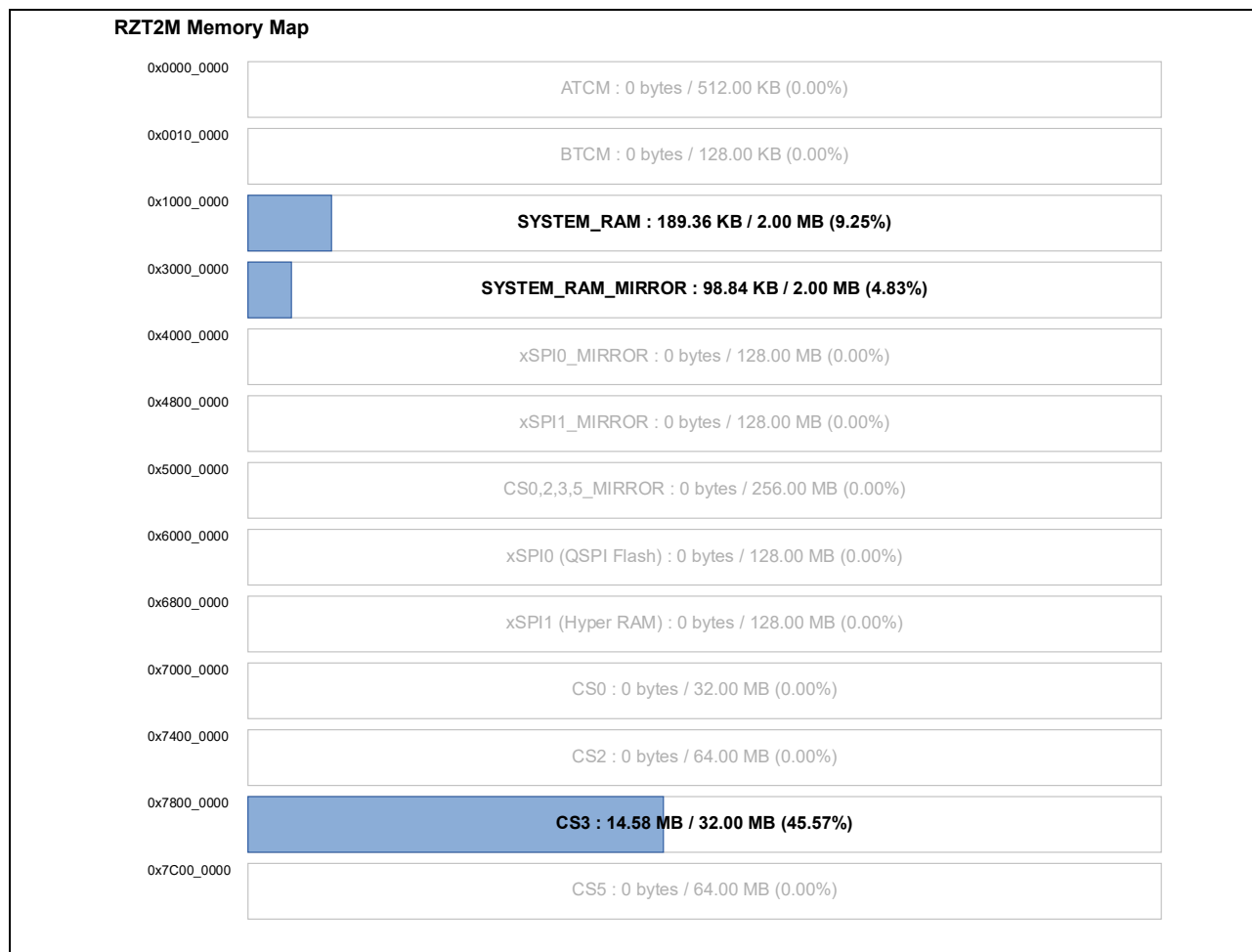


Figure 8.14. Memory footprint of the RZ/T2M Secondary project (e² studio)

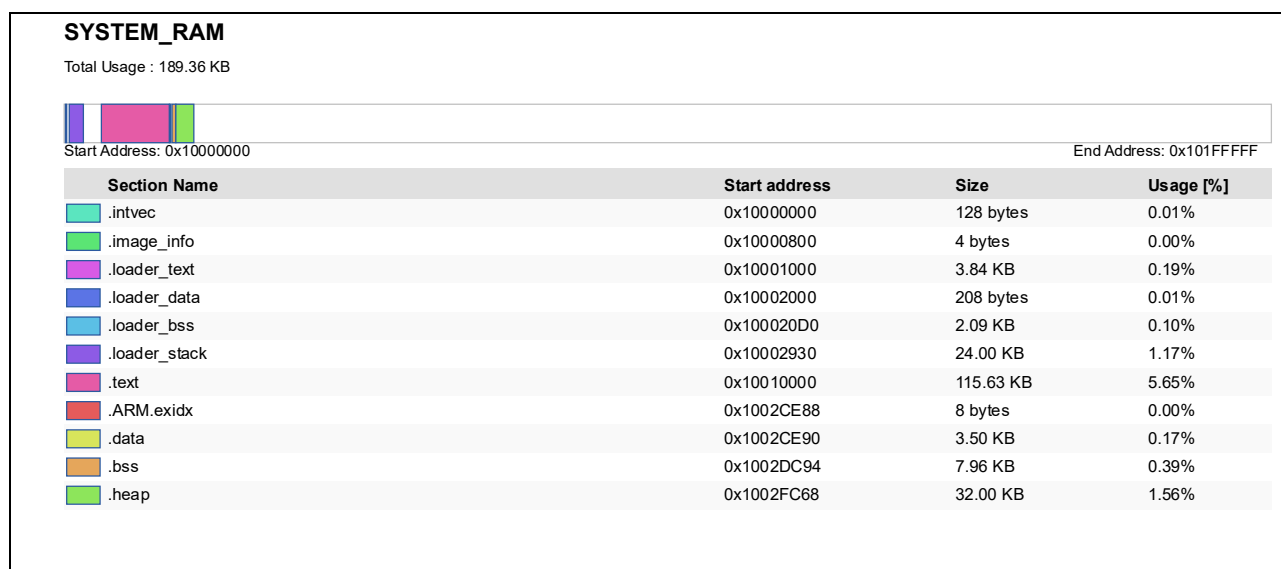


Figure 8.15. SYSTEM_RAM footprint of the RZ/T2M Secondary project (e² studio)



Figure 8.16. SYSTEM_RAM_MIRROR footprint of the RZ/T2M Secondary project (e² studio)

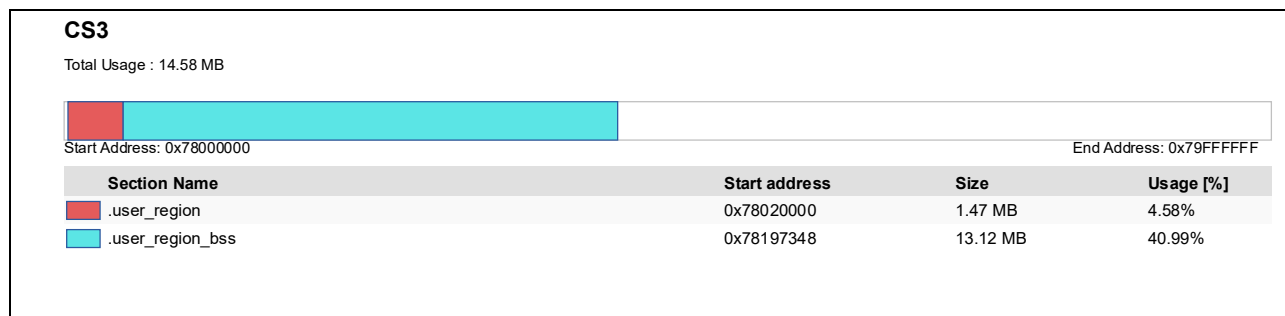


Figure 8.17. CS3 (SDRAM) footprint of the RZ/T2M Secondary project (e² studio)

8.3.1.2 EWARM

(1) Primary project

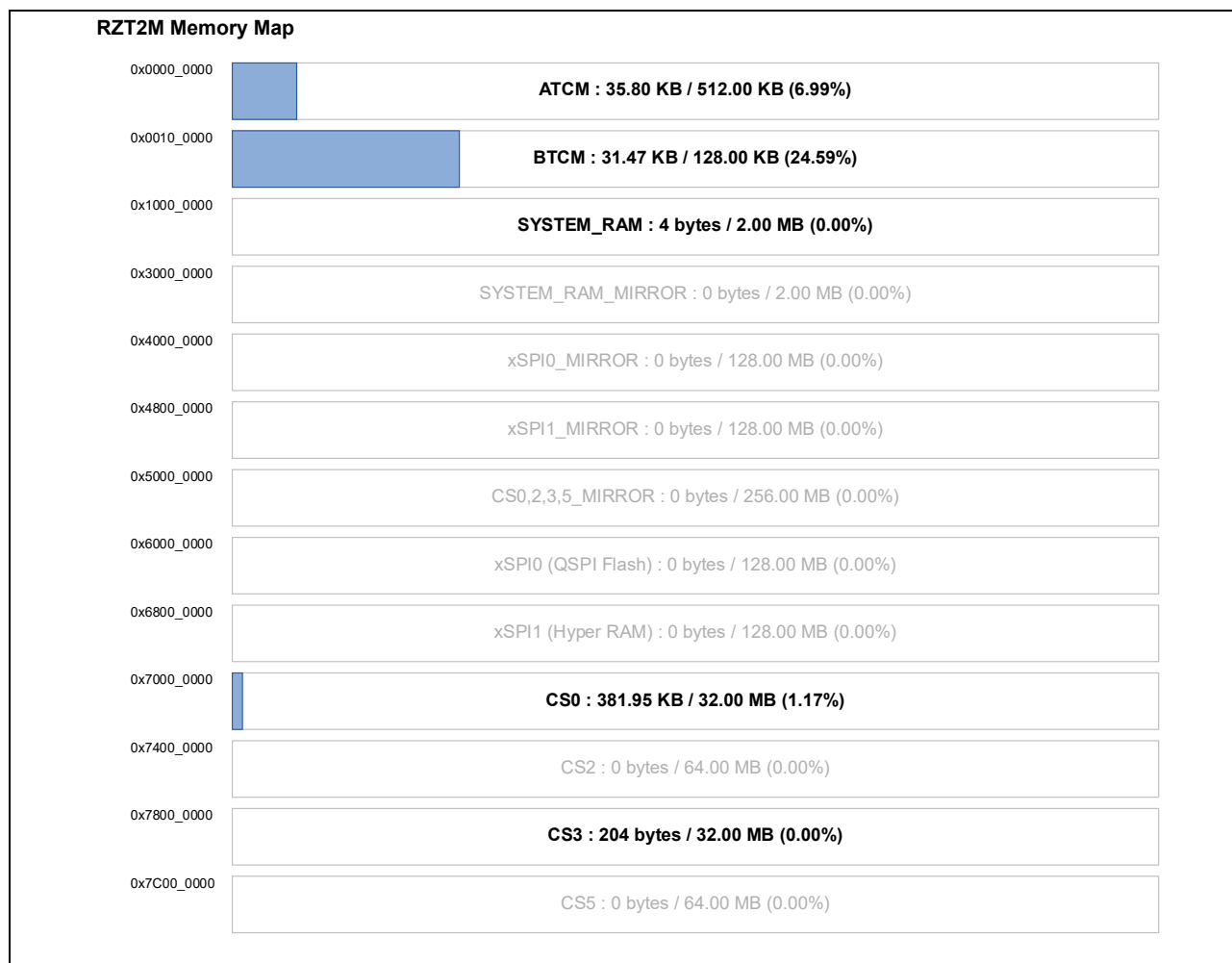


Figure 8.18. Memory footprint of the RZ/T2M Primary project (EWARM)

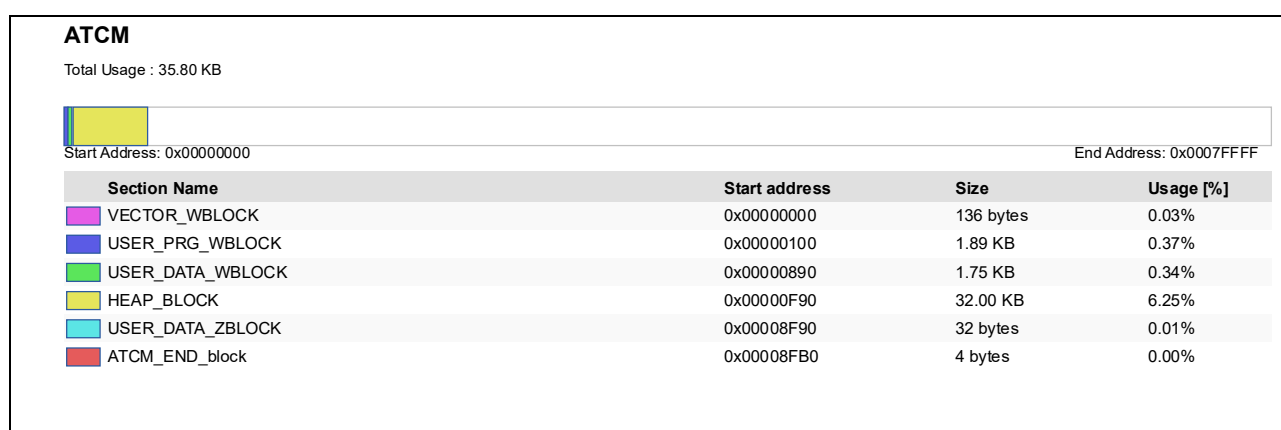


Figure 8.19. ATCM footprint of the RZ/T2M Primary project (EWARM)

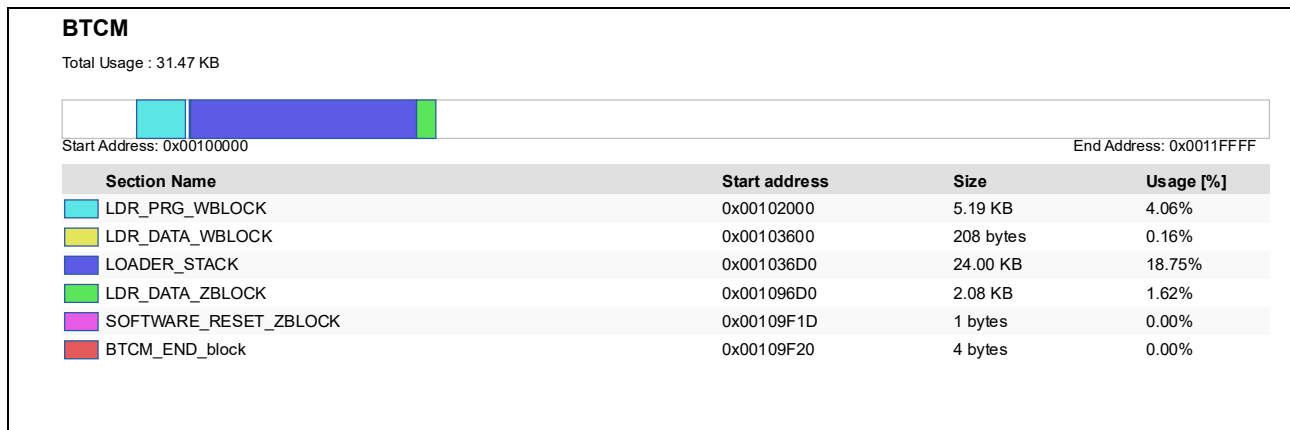


Figure 8.20. BTCM footprint of the RZ/T2M Primary project (EWARM)



Figure 8.21. SYSTEM_RAM footprint of the RZ/T2M Primary project (EWARM)

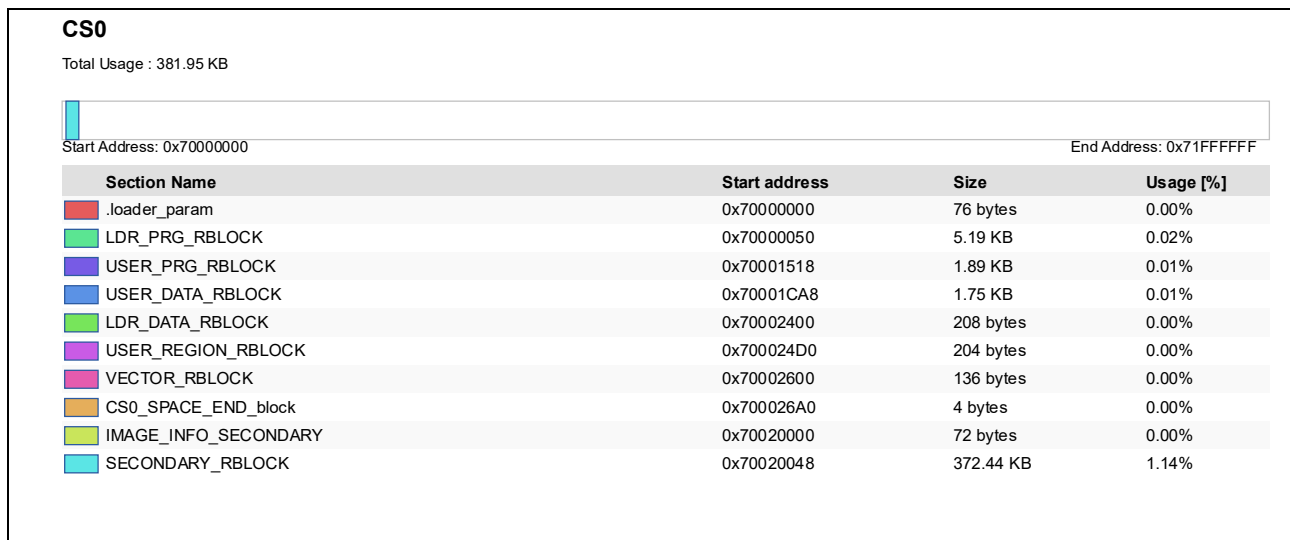


Figure 8.22. CS0 (NOR Flash) footprint of the RZ/T2M Primary project (EWARM)

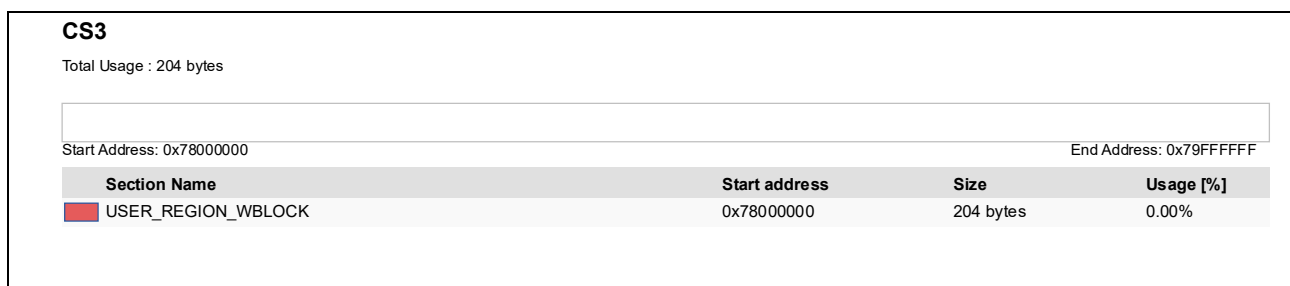


Figure 8.23. CS3 (SDRAM) footprint of the RZ/T2M Primary project (EWARM)

(2) Secondary project

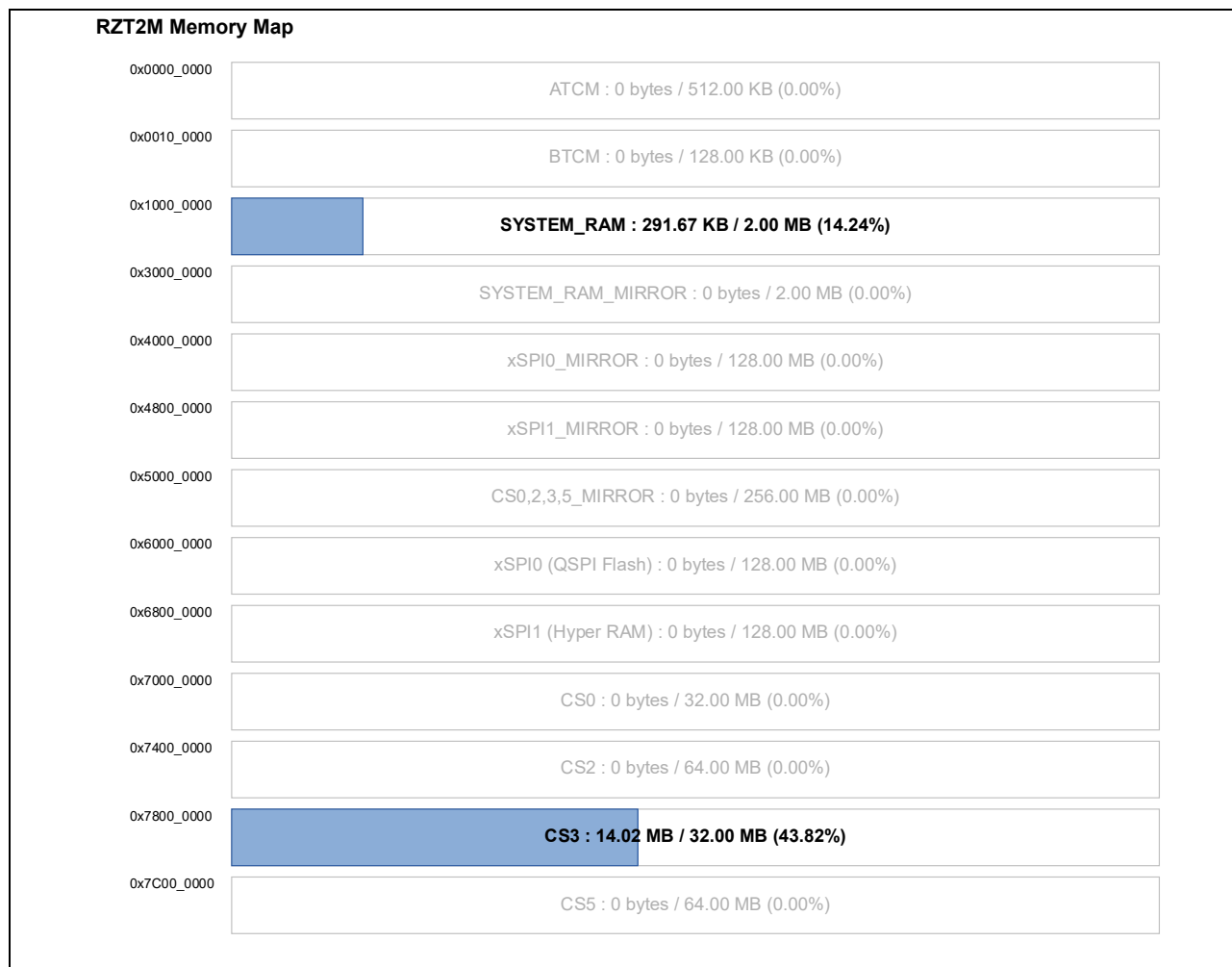


Figure 8.24. Memory footprint of the RZ/T2M Secondary project (EWARM)

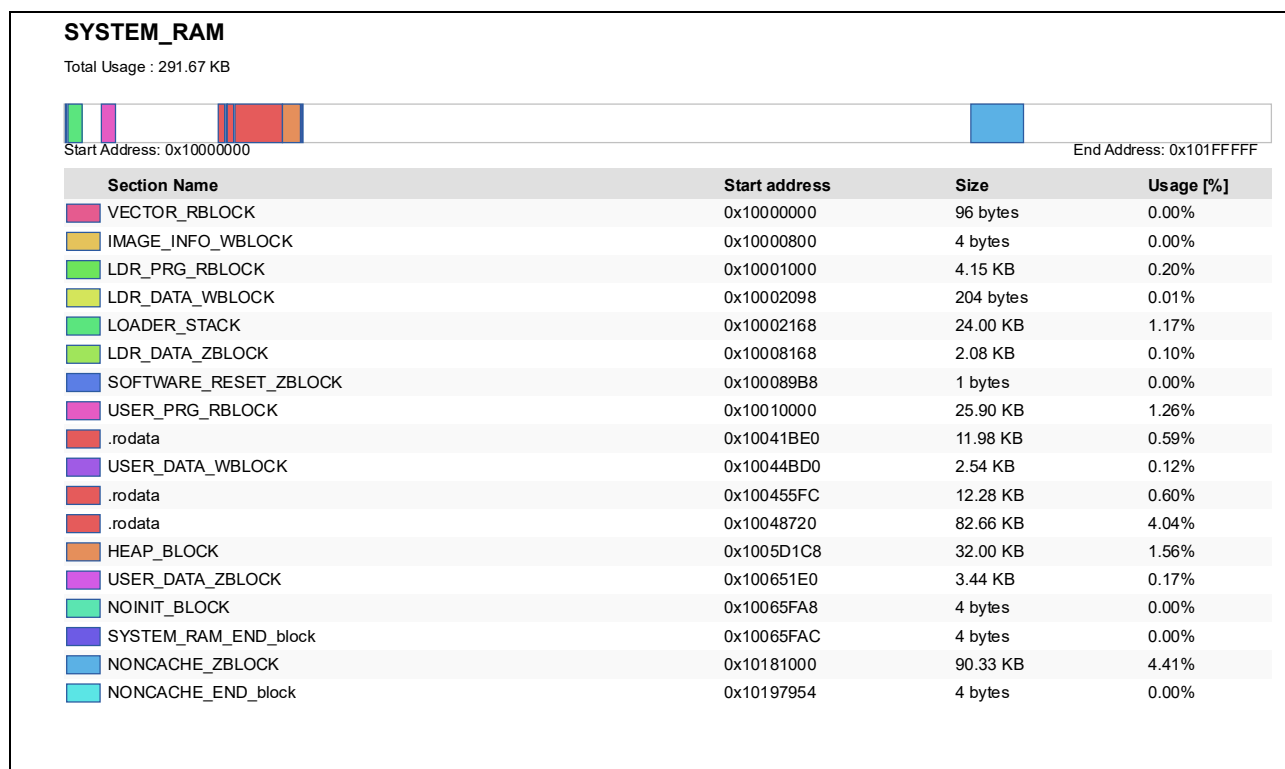


Figure 8.25. SYSTEM_RAM footprint of the RZ/T2M Secondary project (EWARM)

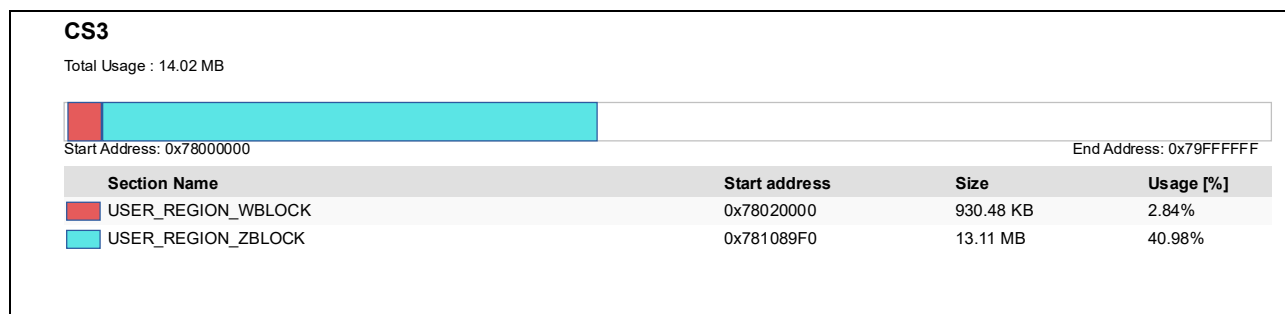


Figure 8.26. CS3 (SDRAM) footprint of the RZ/T2M Secondary project (EWARM)

8.3.2 RZ/N2L project

8.3.2.1 e² studio

(1) Primary project

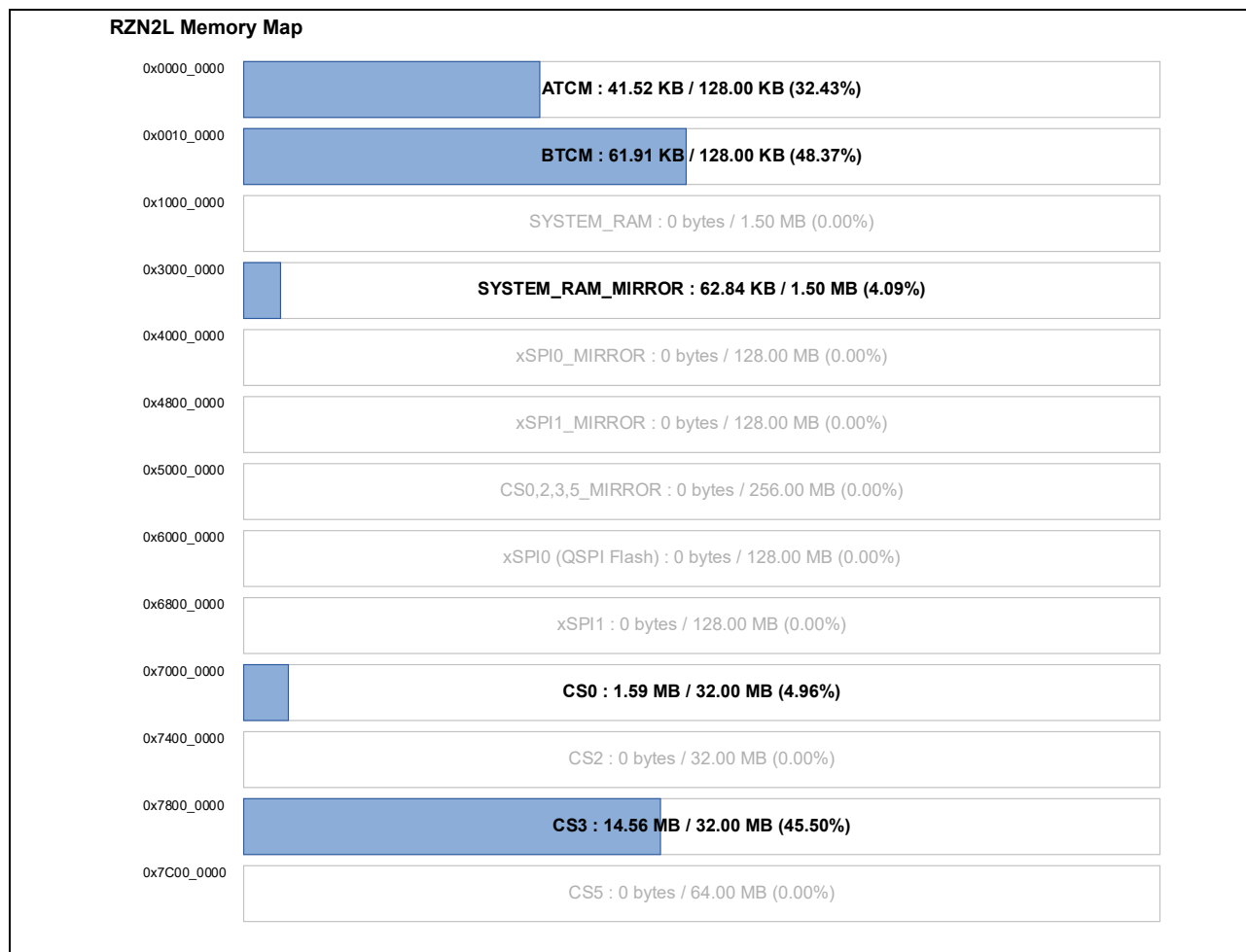


Figure 8.27. Memory footprint of the RZ/N2L Primary project (e² studio)

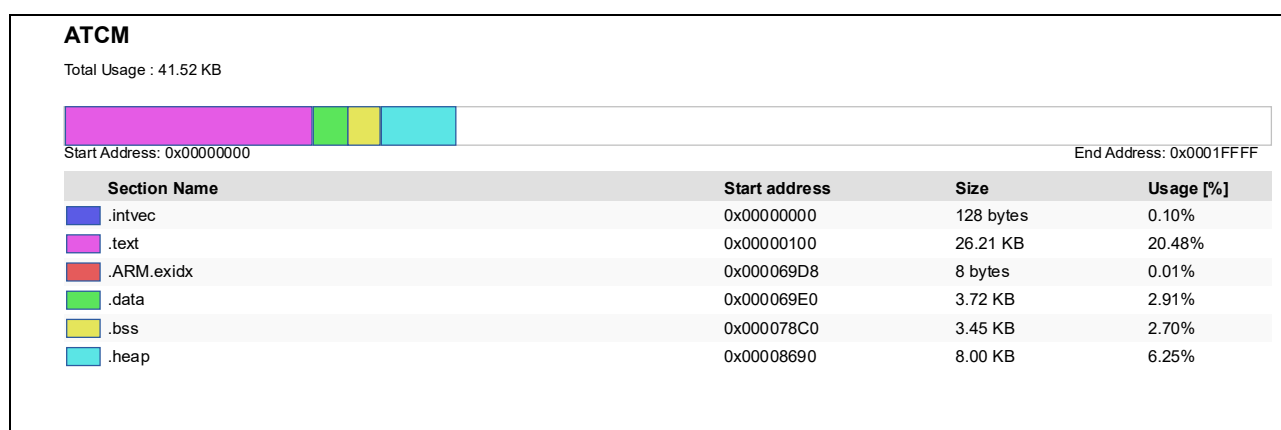


Figure 8.28. ATCM footprint of the RZ/N2L Primary project (e² studio)

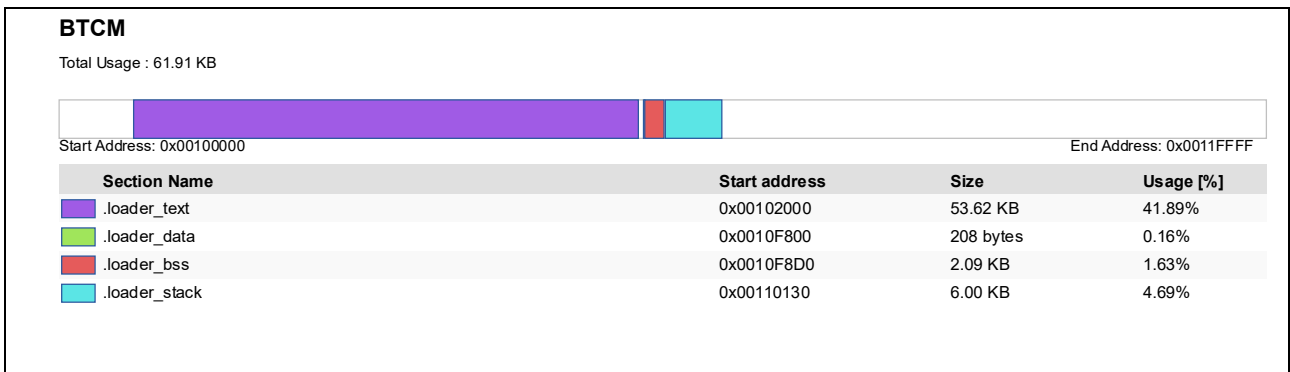


Figure 8.29. BTCM footprint of the RZ/N2L Primary project (e² studio)

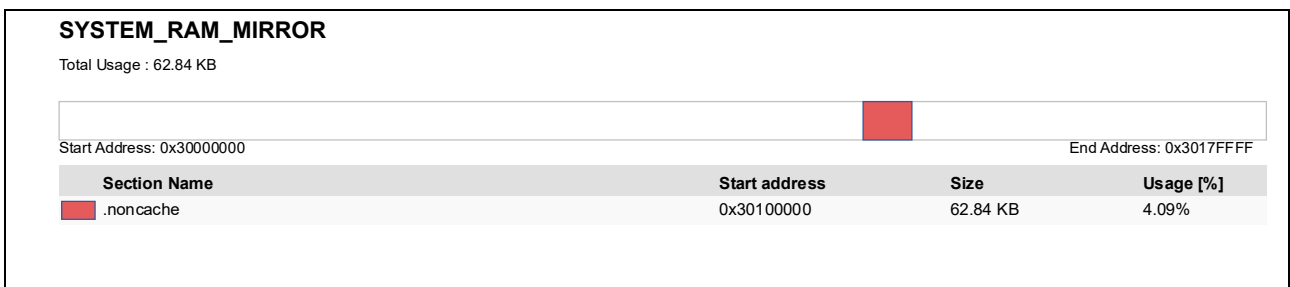


Figure 8.30. SYSTEM_RAM_MIRROR footprint of the RZ/N2L Primary project (e² studio)

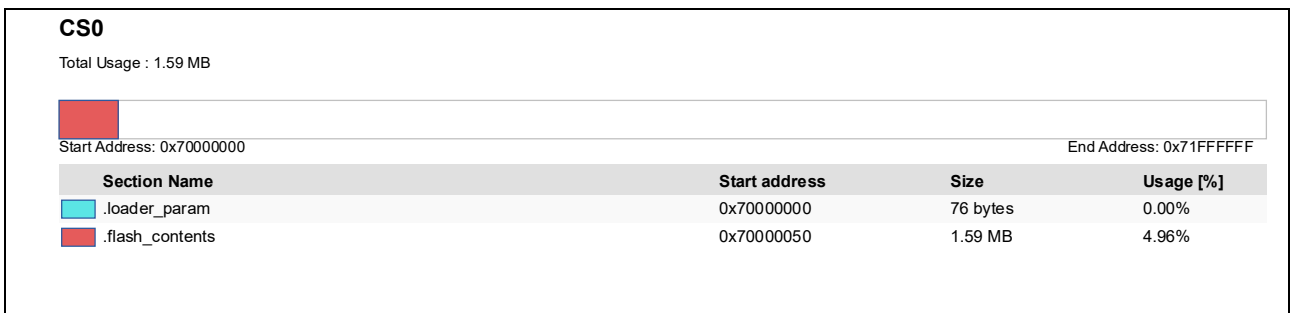


Figure 8.31. CS0 (NOR Flash) footprint of the RZ/N2L Primary project (e² studio)

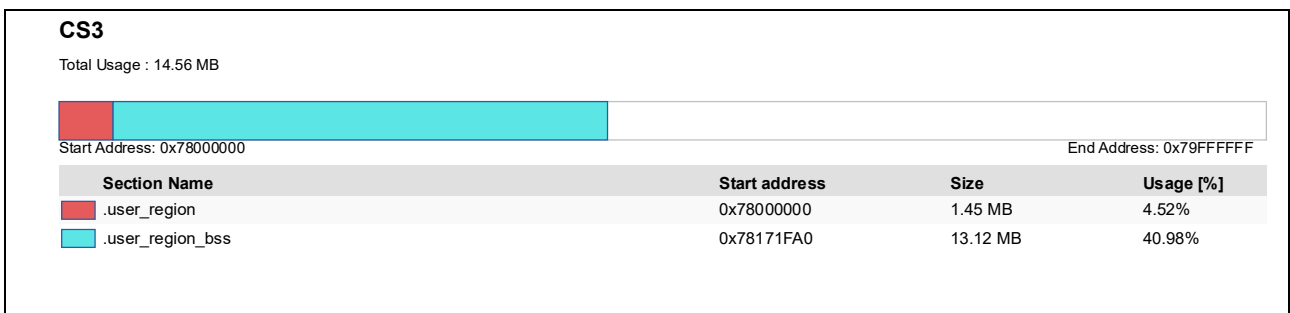


Figure 8.32. CS3 (SDRAM) footprint of the RZ/ N2L Primary project (e² studio)

8.3.2.2 EWARM

(1) Primary project

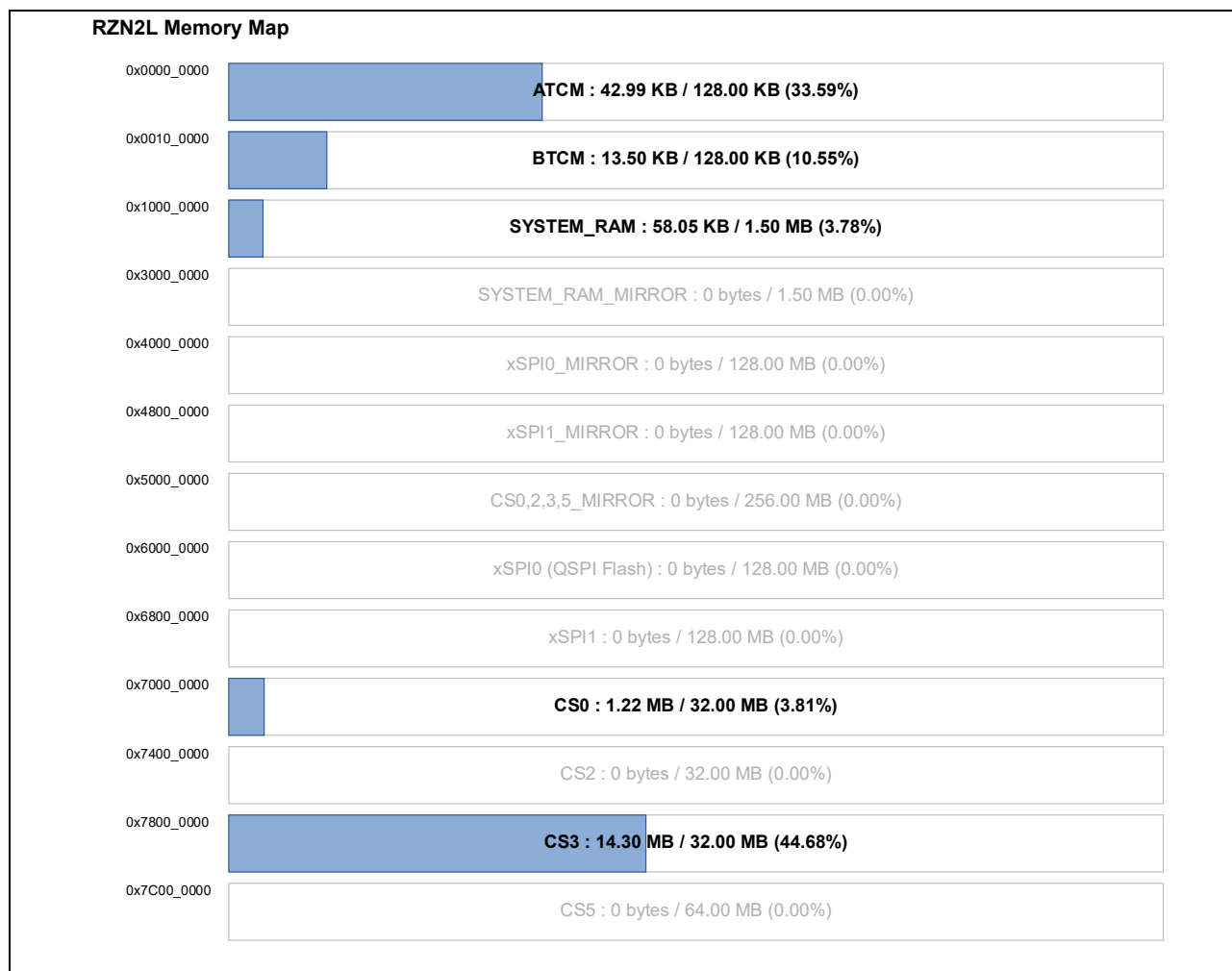


Figure 8.33. Memory footprint of the RZ/N2L Primary project (EWARM)

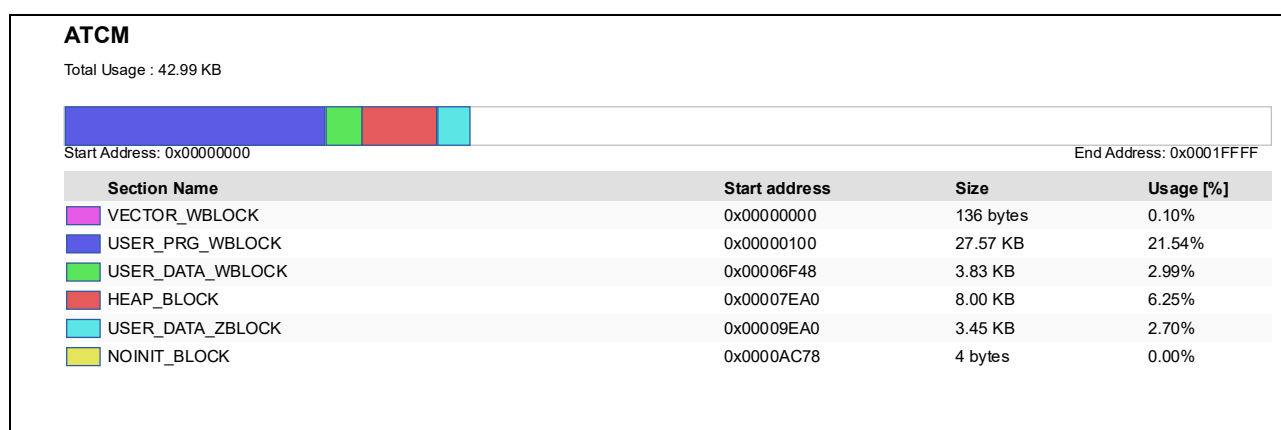


Figure 8.34. ATCM footprint of the RZ/N2L Primary project (EWARM)

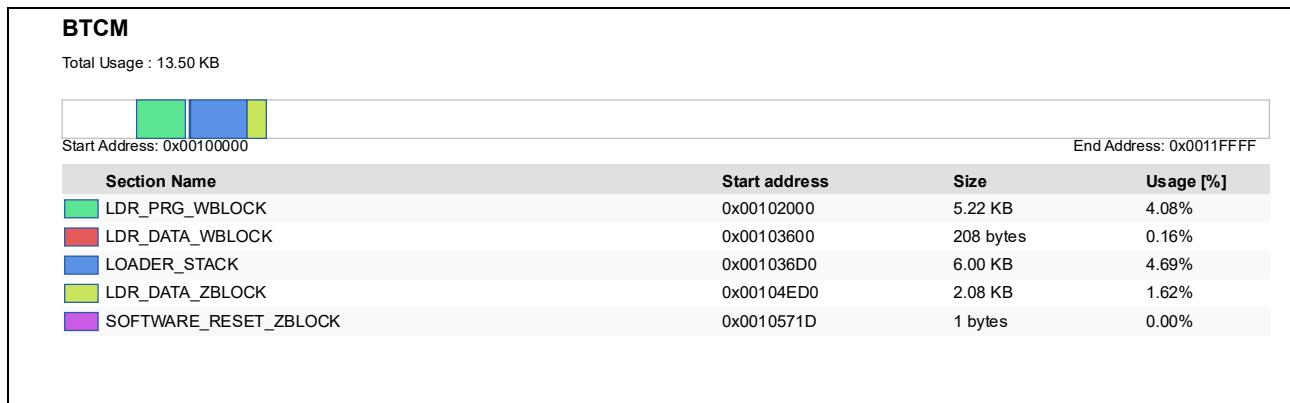


Figure 8.35. BTCM footprint of the RZ/N2L Primary project (EWARM)



Figure 8.36. SYSTEM_RAM footprint of the RZ/N2L Primary project (EWARM)

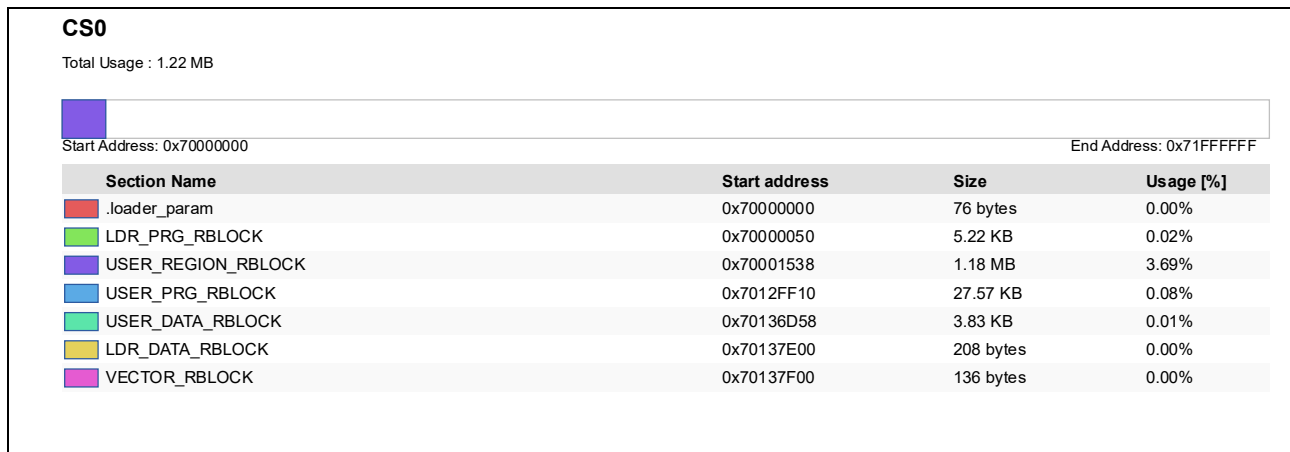


Figure 8.37. CS0 (NOR Flash) footprint of the RZ/N2L Primary project (EWARM)

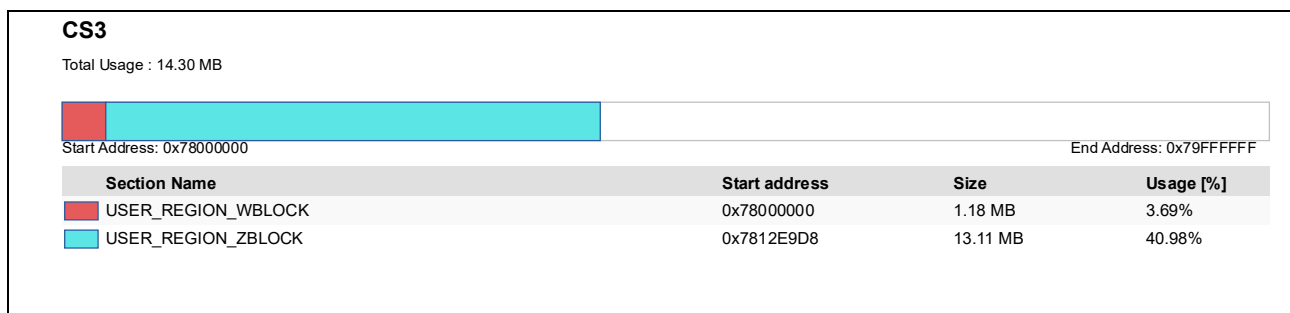


Figure 8.38. CS3 (SDRAM) footprint of the RZ/N2L Primary project (EWARM)

8.3.3 RZ/T2H, N2H project (CR52_dual)

※RZ/T2H と N2H は同じメモリ配置で実装しているため、代表して RZ/T2H での測定結果を記載していません。

8.3.3.1 e² studio

(1) Primary project

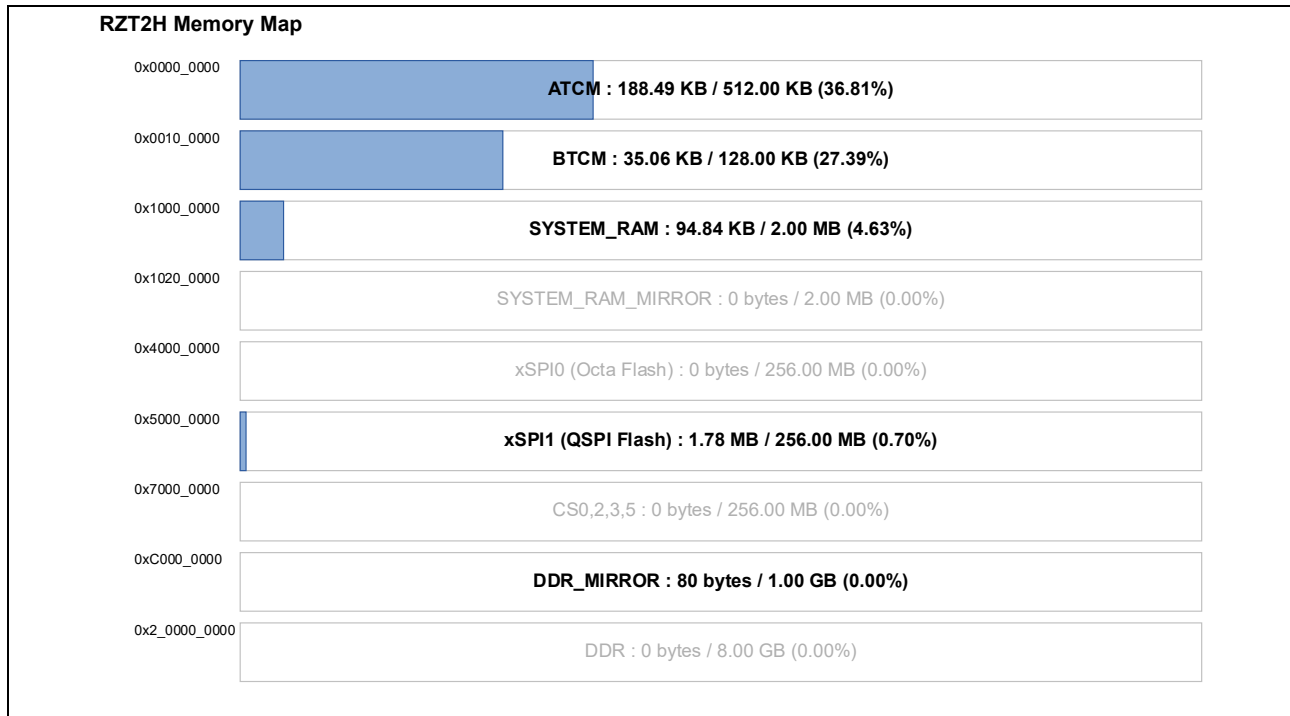


Figure 8.39. Memory footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)

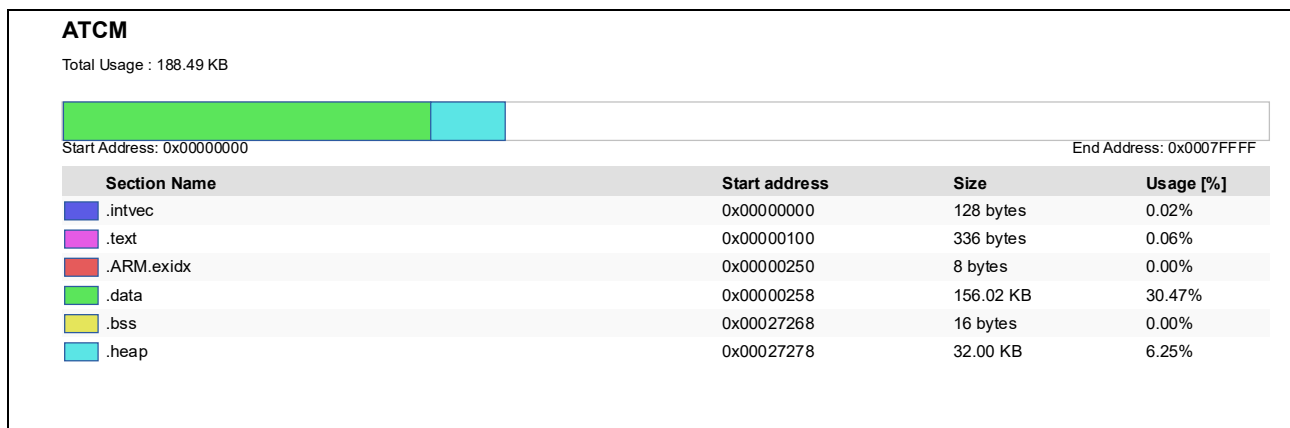


Figure 8.40. ATCM footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)

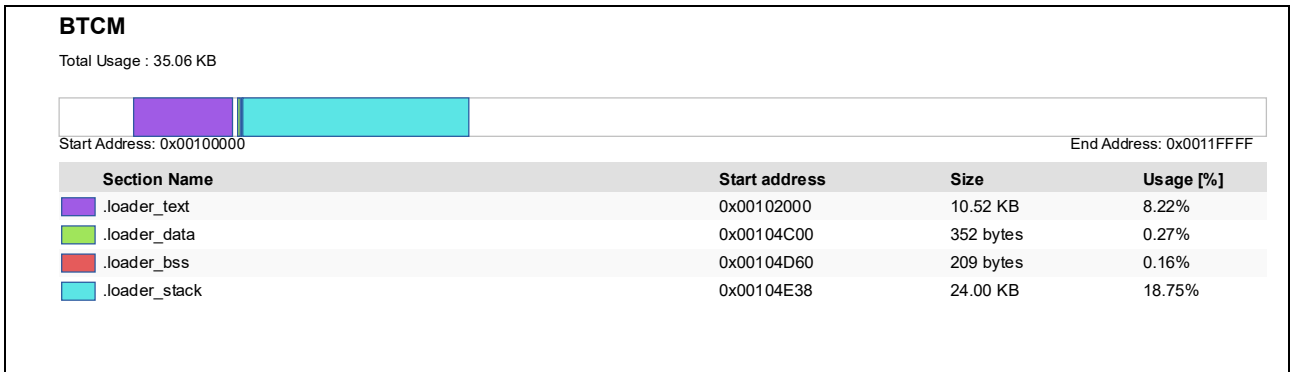


Figure 8.41. BTCM footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)



Figure 8.42. SYSTEM_RAM footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)

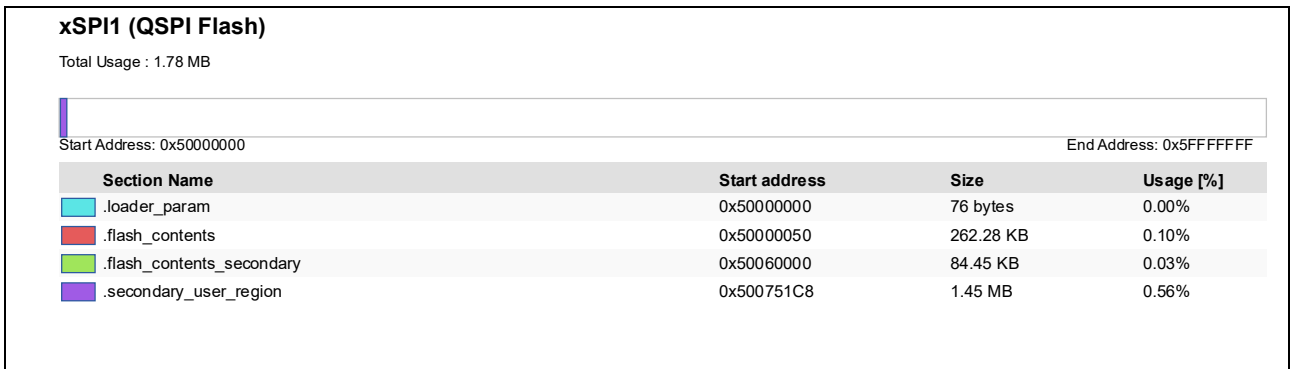


Figure 8.43. xSPI1 (QSPI Flash) footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)



Figure 8.44. DDR_MIRROR footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)

(2) Secondary project

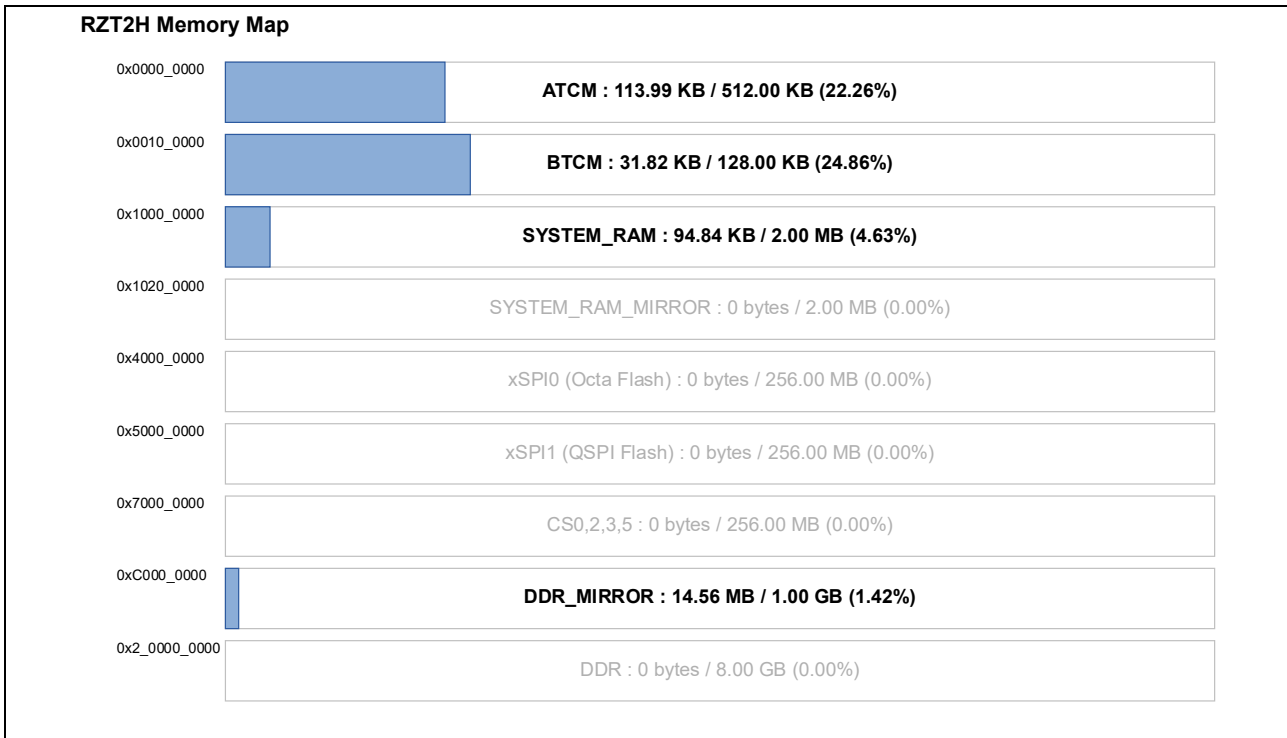


Figure 8.45. Memory footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)

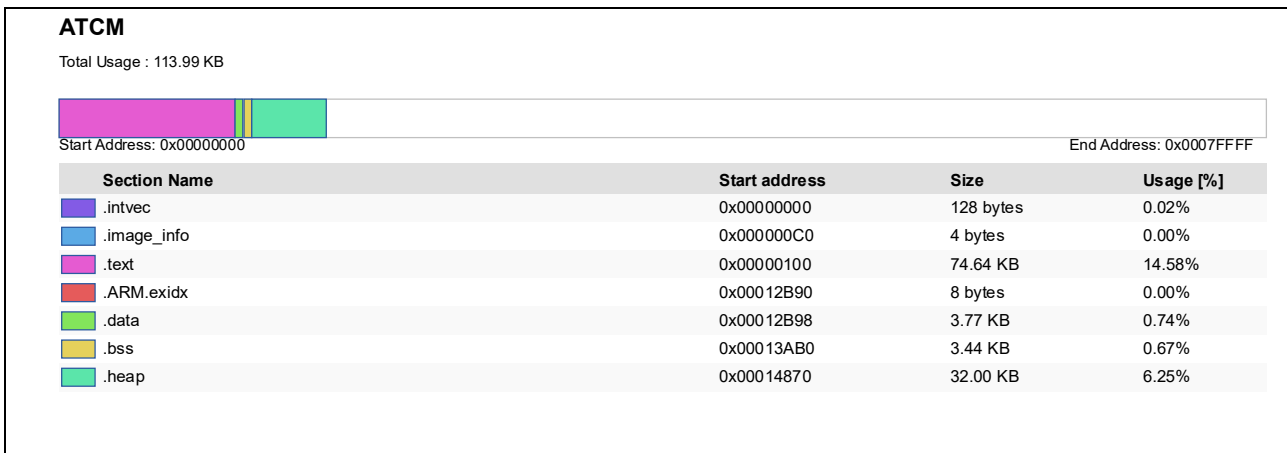


Figure 8.46. ATCM footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)

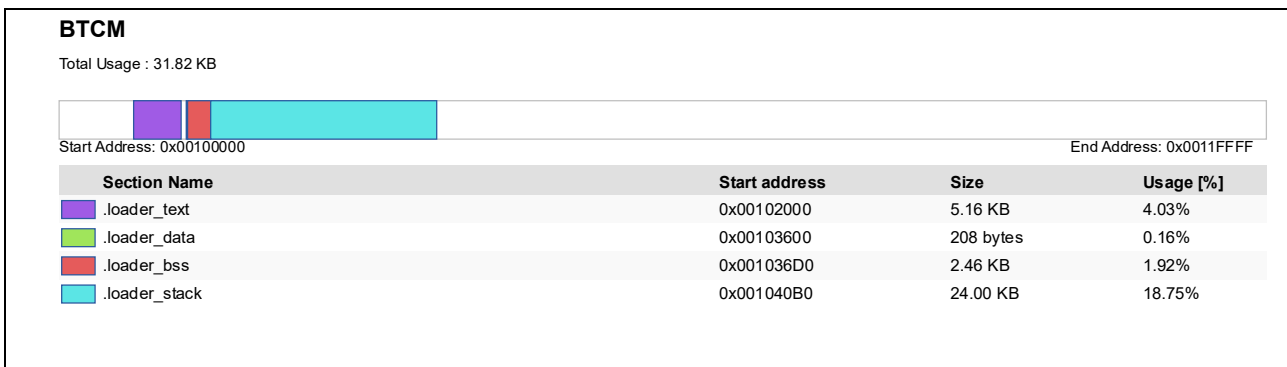


Figure 8.47. BTCM footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)



Figure 8.48. SYSTEM_RAM footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)

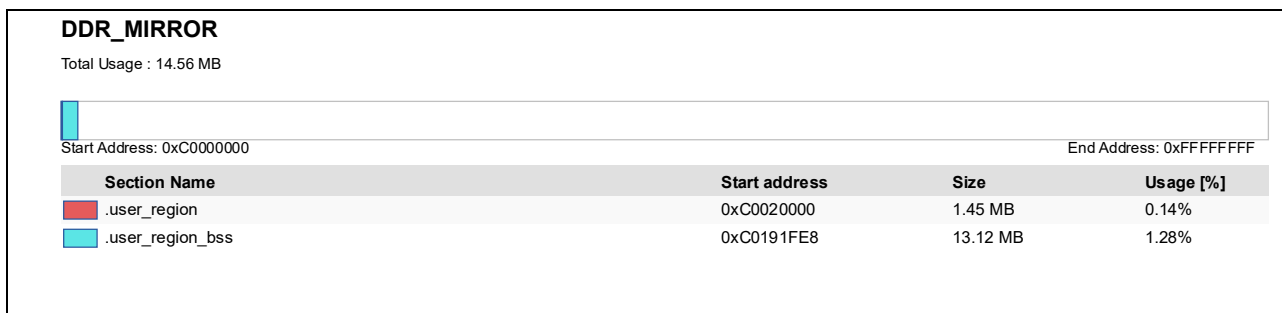


Figure 8.49. DDR_MIRROR footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)

8.3.3.2 EWARM

(1) Primary project

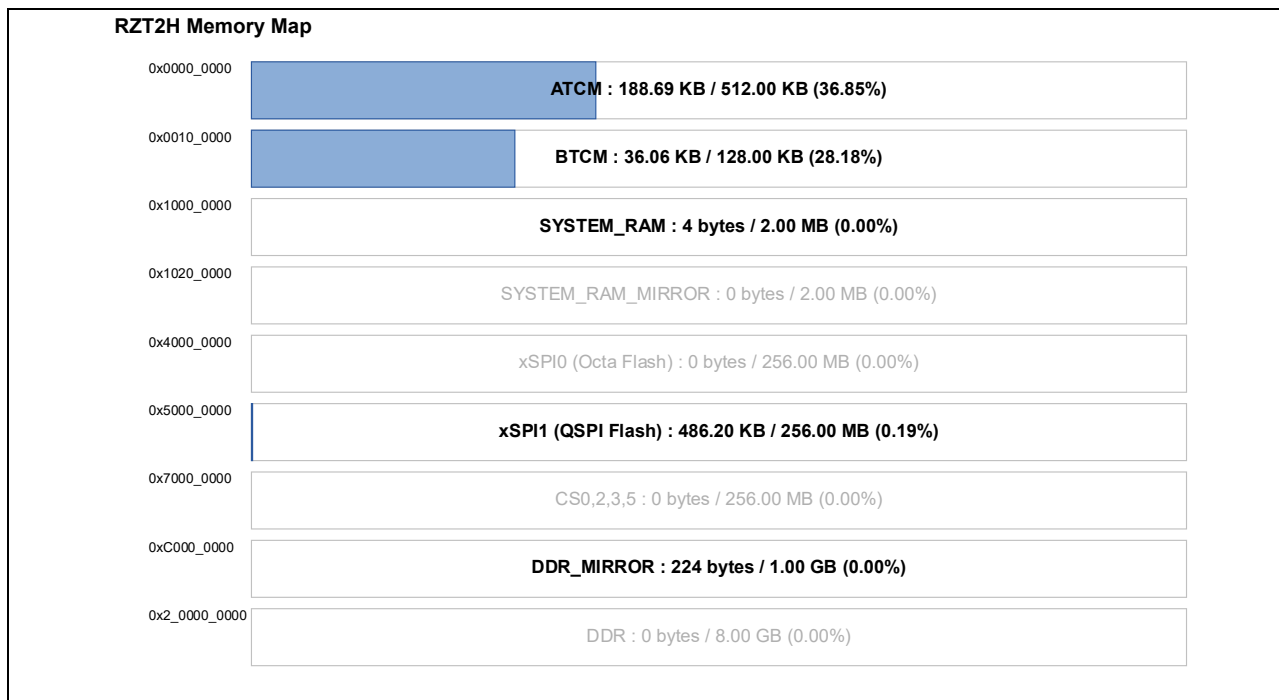


Figure 8.50. Memory footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)

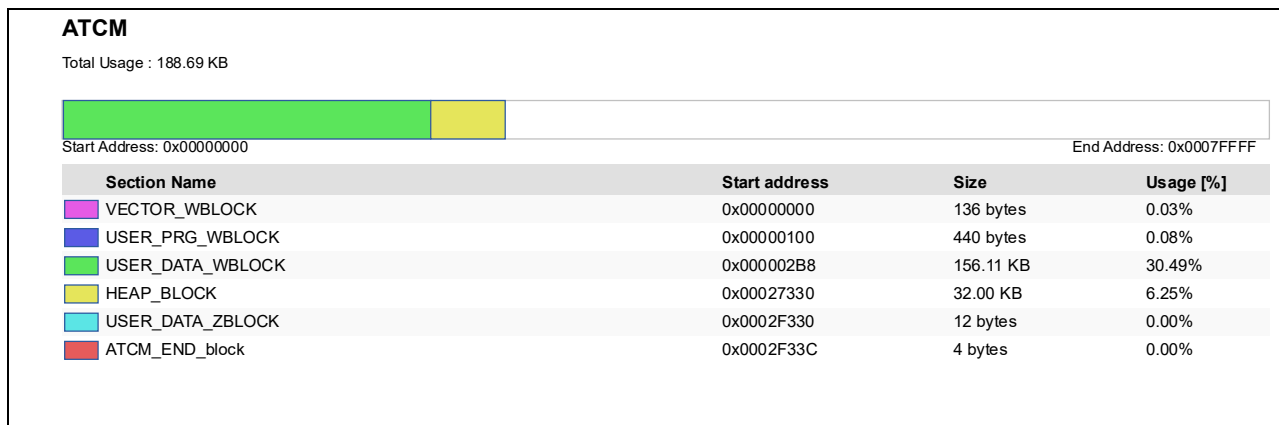


Figure 8.51. ATCM footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)

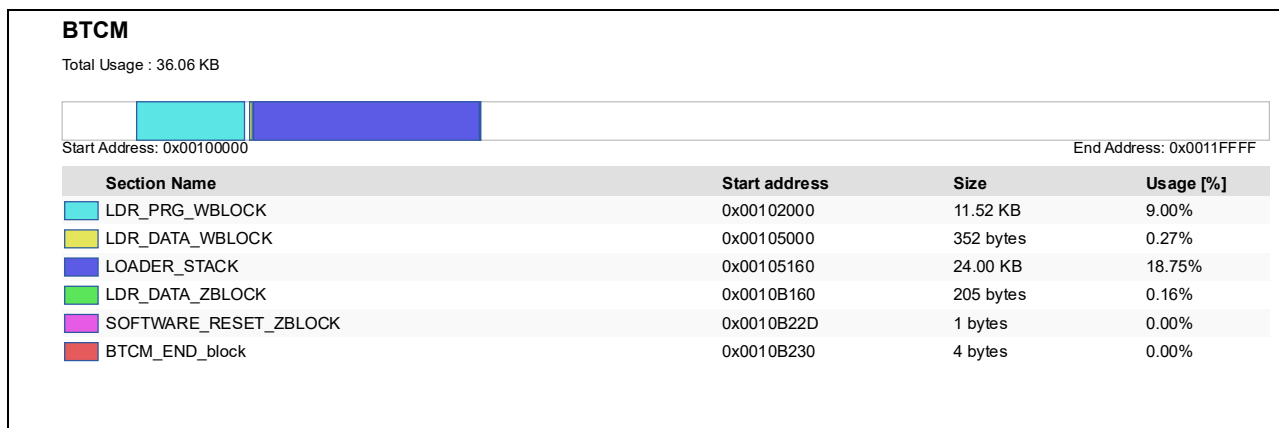


Figure 8.52. BTCM footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)

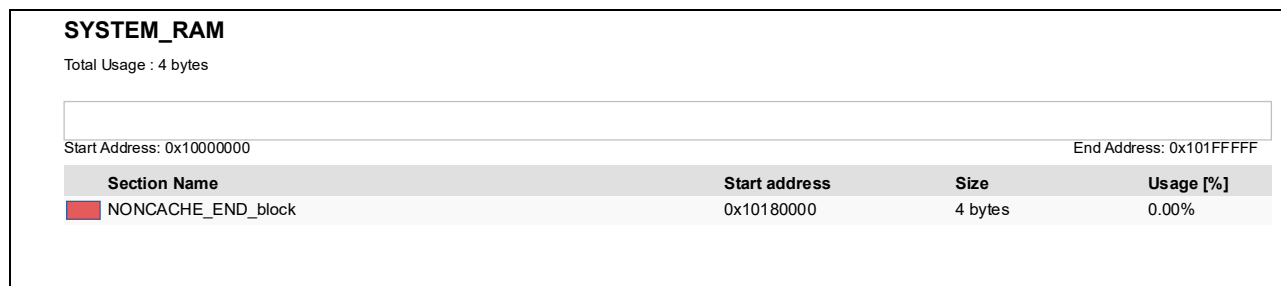


Figure 8.53. SYSTEM_RAM footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)

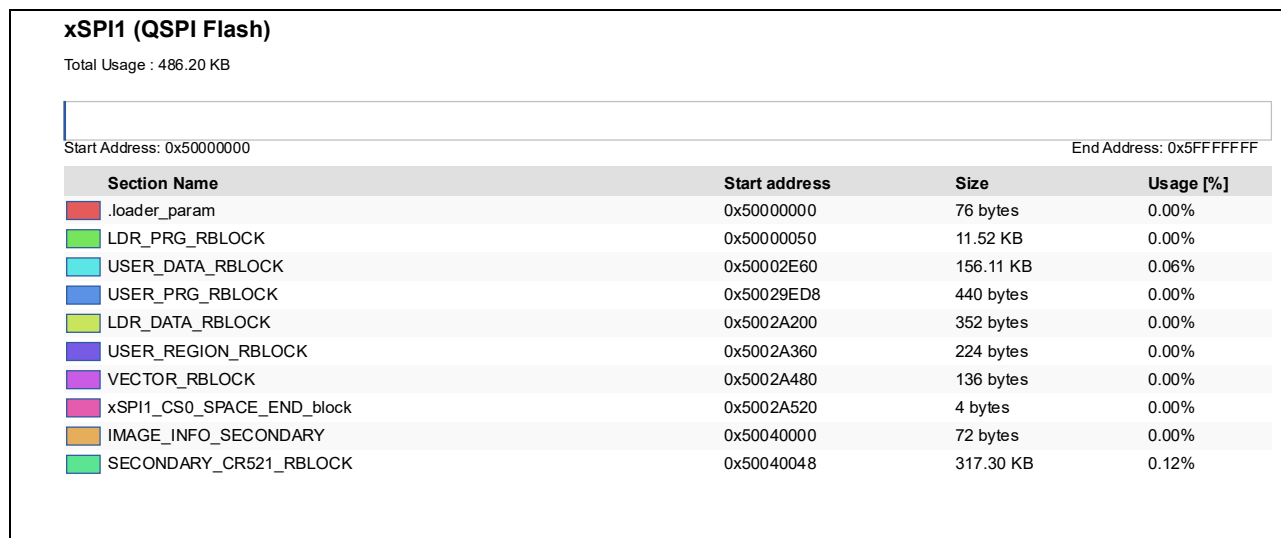


Figure 8.54. xSPI1 (QSPI Flash) footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)

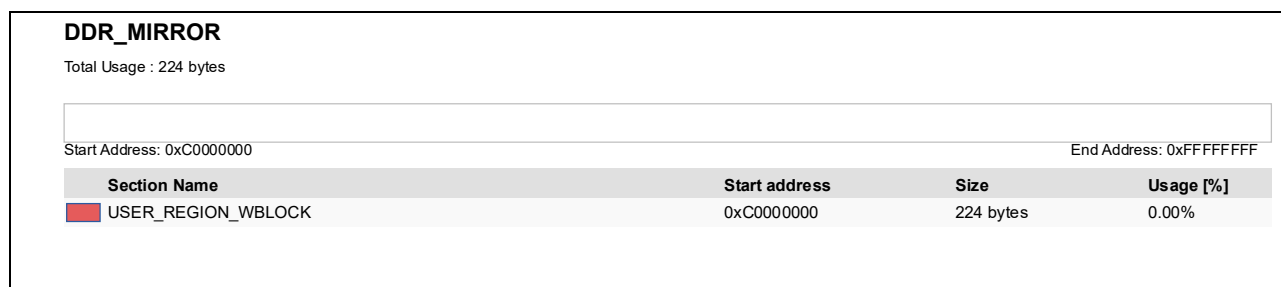


Figure 8.55. DDR_MIRROR footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)

(2) Secondary project

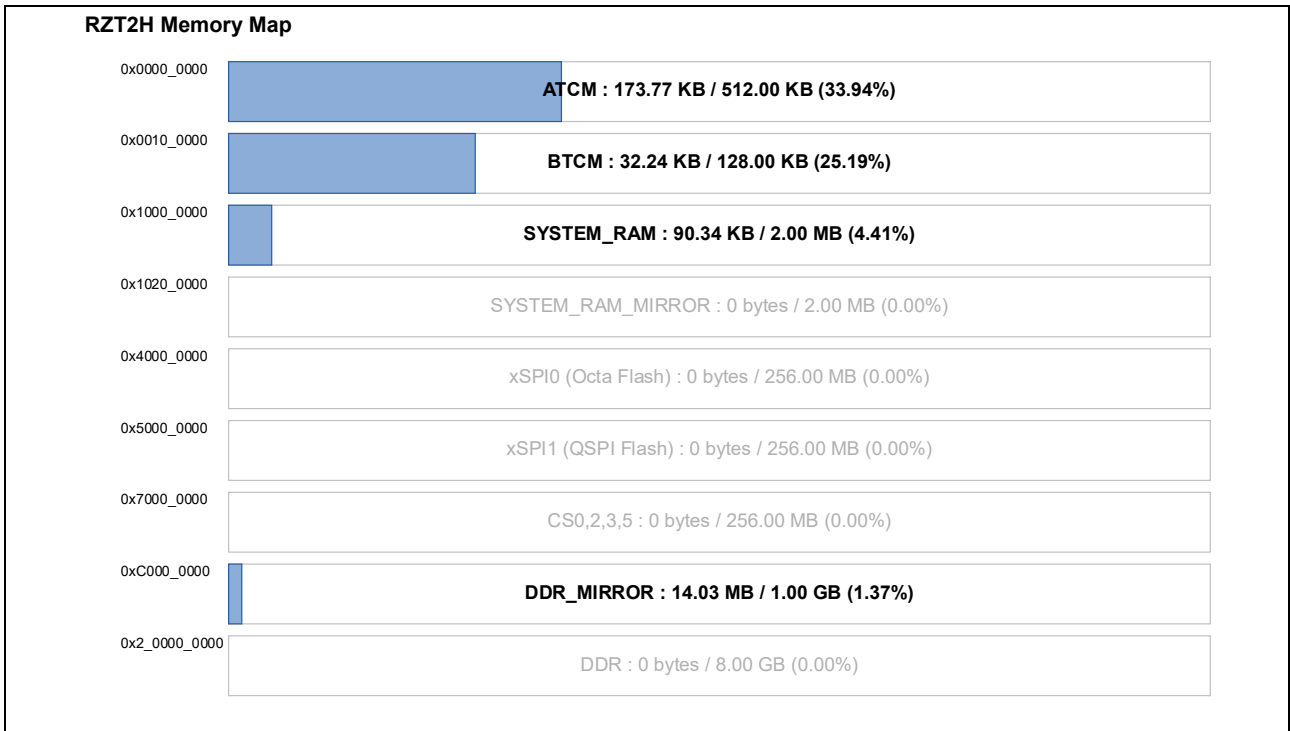


Figure 8.56. Memory footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)

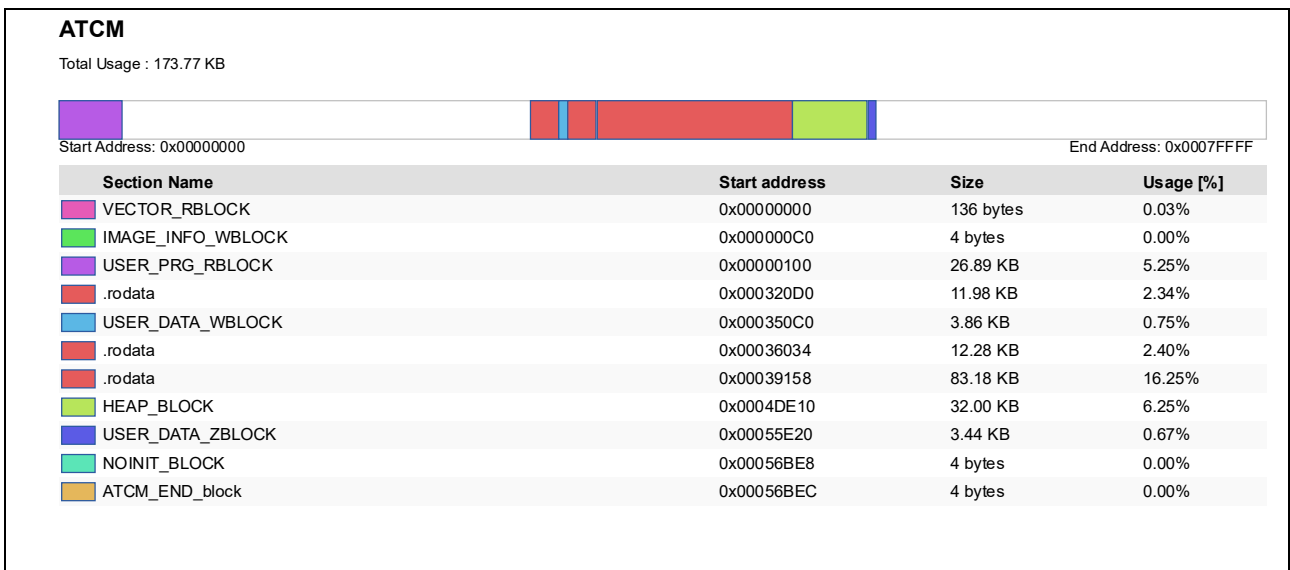


Figure 8.57. ATCM footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)

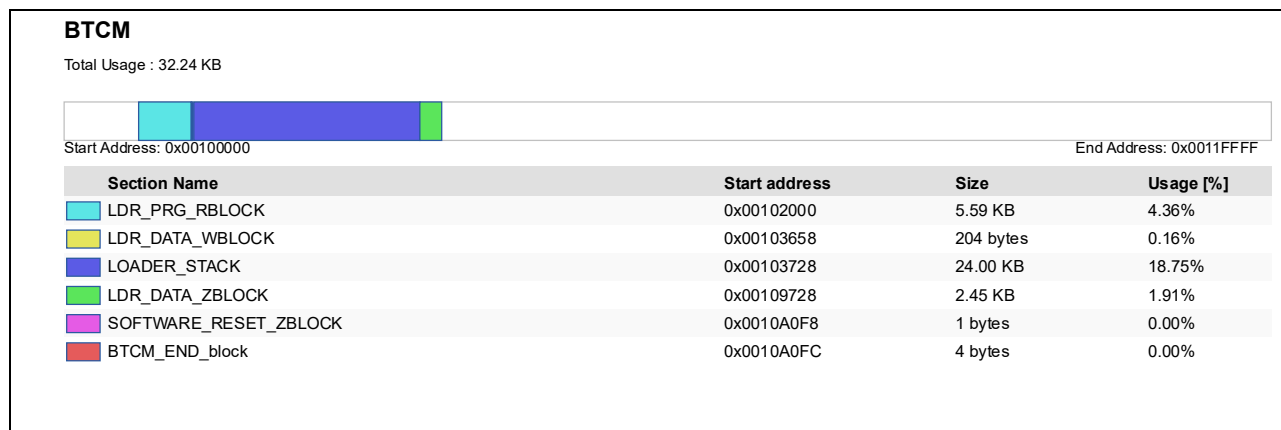


Figure 8.58. BTCM footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)



Figure 8.59. SYSTEM_RAM footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)

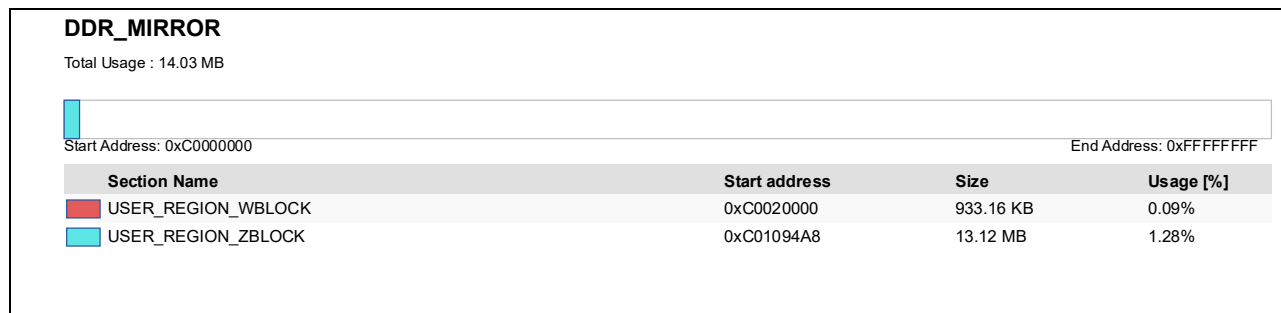


Figure 8.60. DDR_MIRROR footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)

8.3.4 RZ/T2H, N2H project (CA55_dual)

※RZ/T2H と N2H は同じメモリ配置で実装しているため、代表して RZ/T2H での測定結果を記載していません。

8.3.4.1 e² studio

(1) Primary project

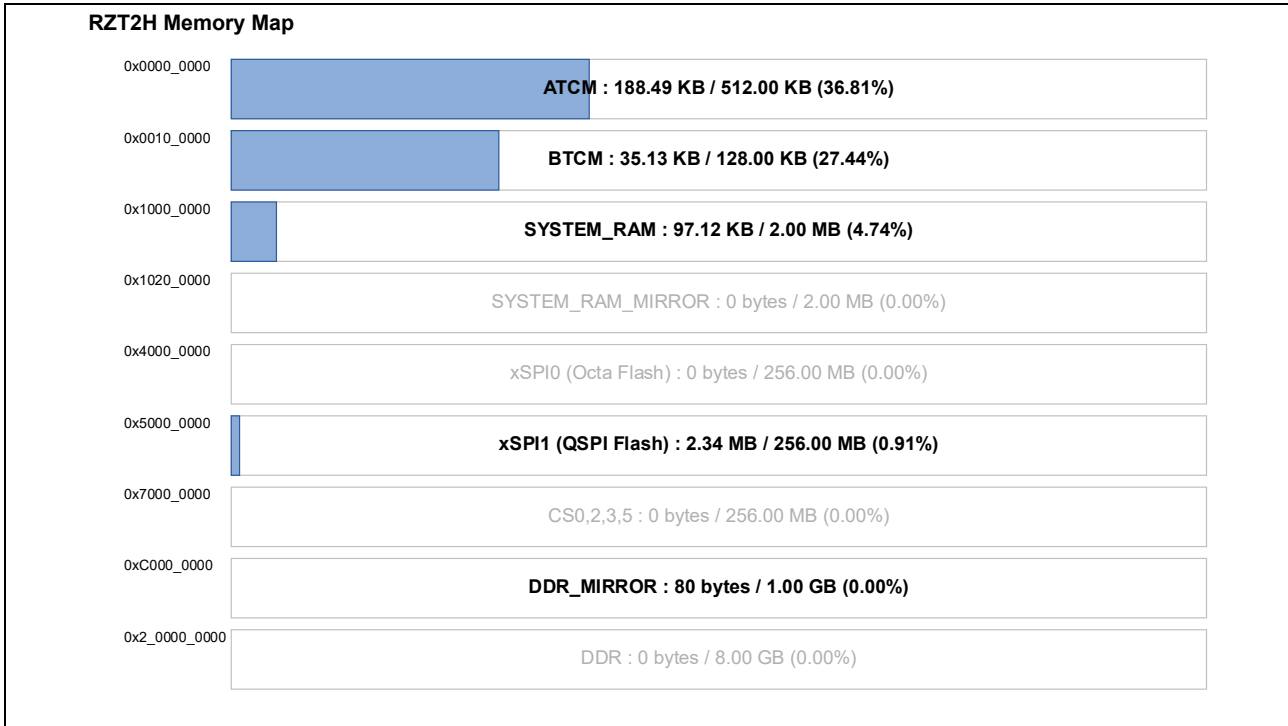


Figure 8.61. Memory footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)

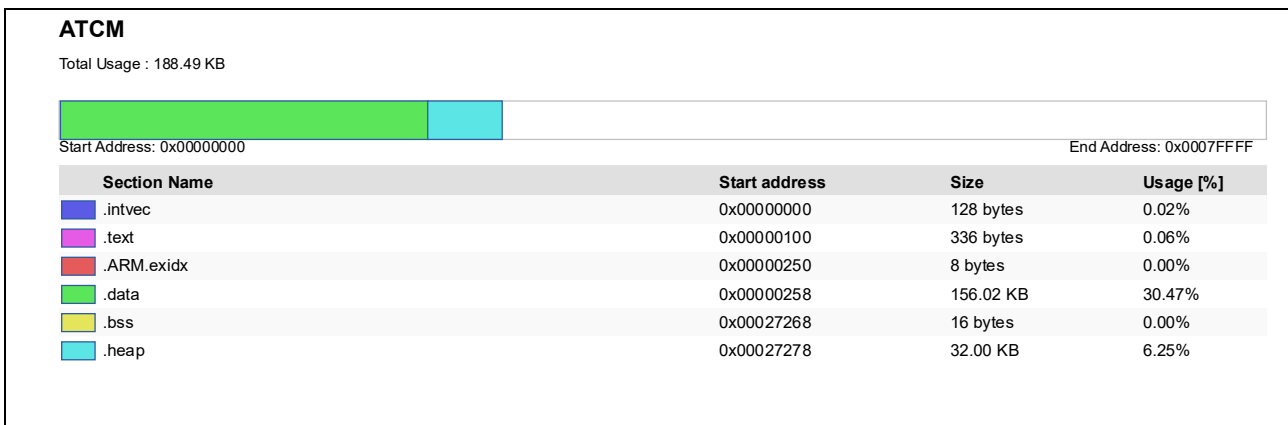
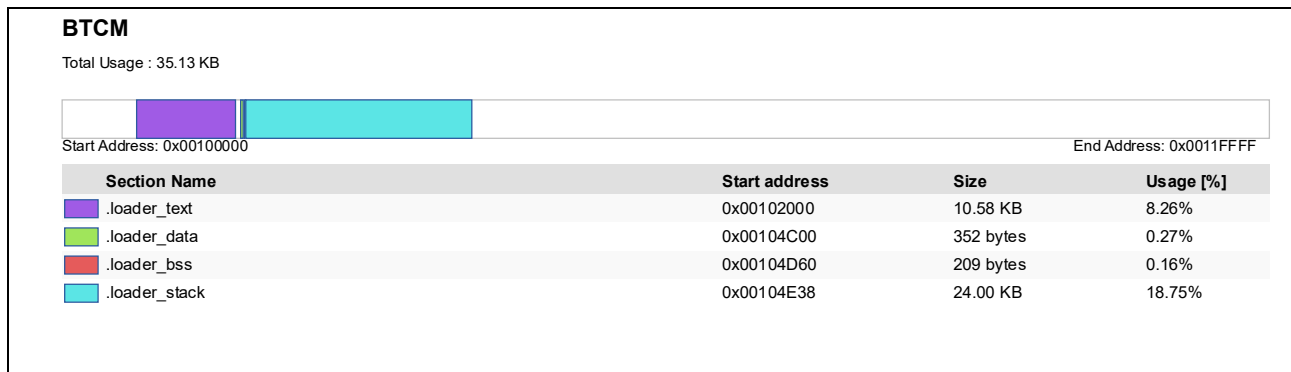
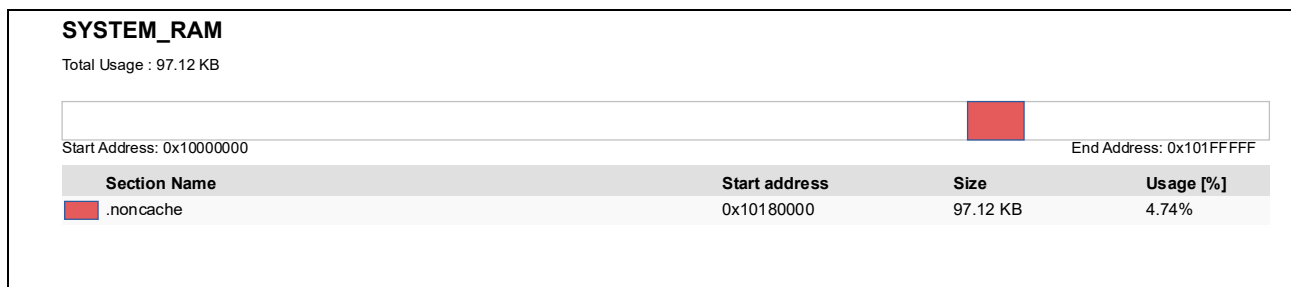
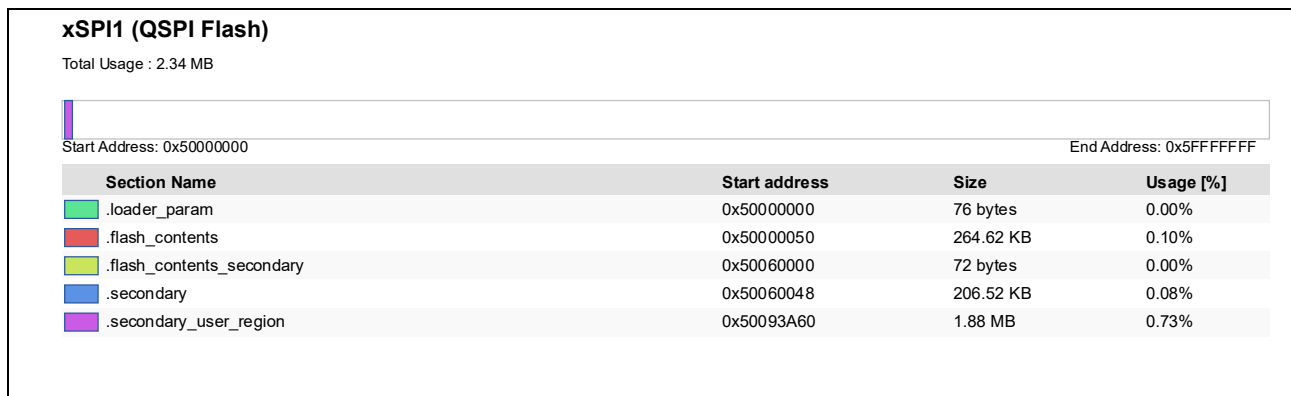


Figure 8.62. ATCM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)

Figure 8.63. BTCM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)Figure 8.64. SYSTEM_RAM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)Figure 8.65. xSPI1 (QSPI Flash) footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)Figure 8.66. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)

(2) Secondary project

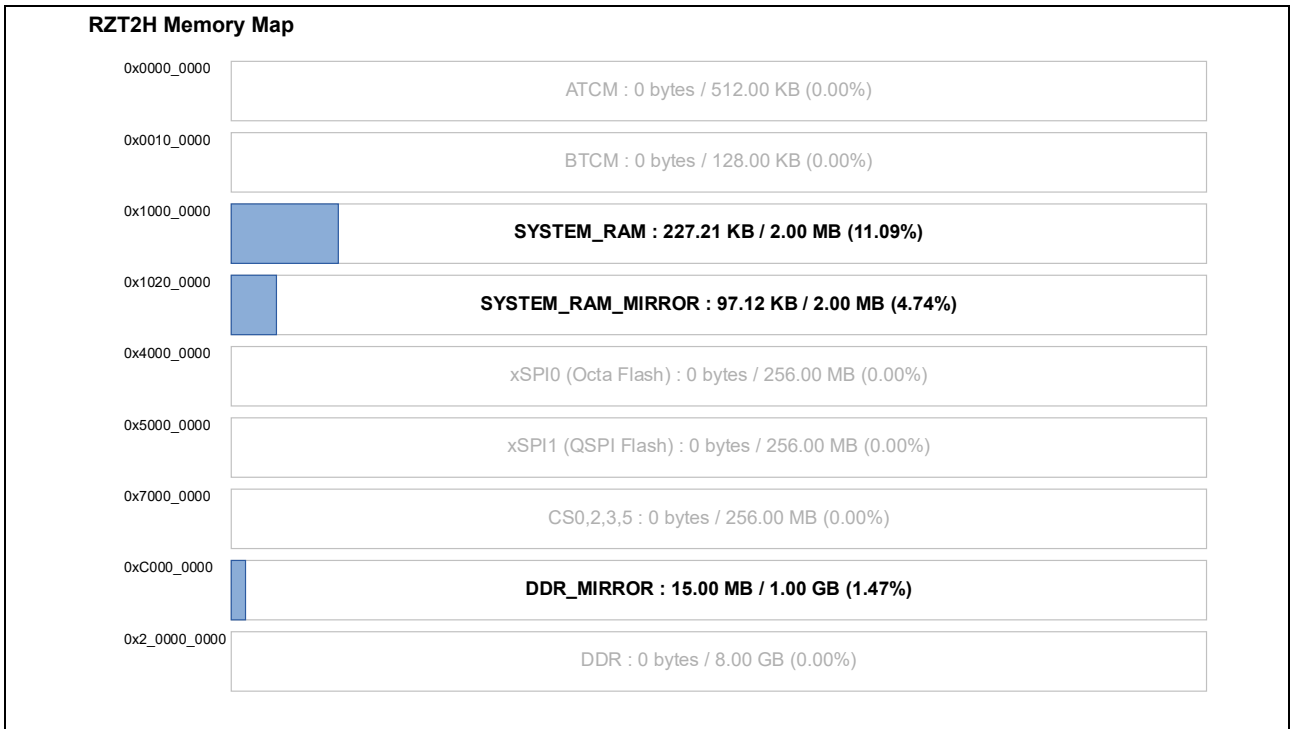


Figure 8.67. Memory footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)

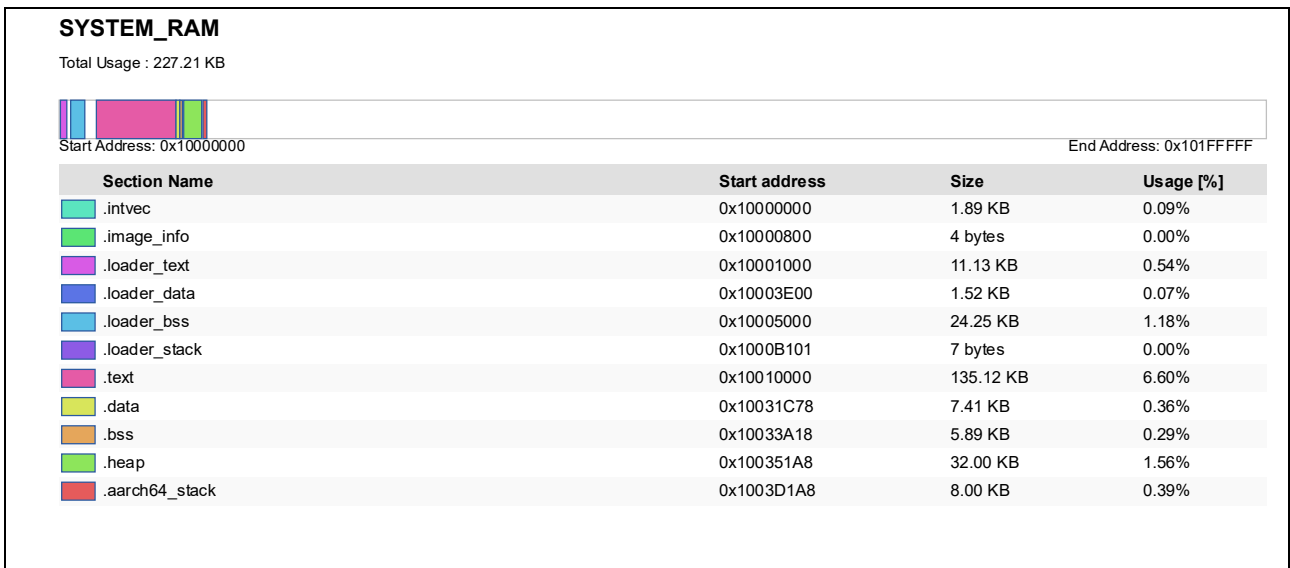


Figure 8.68. SYSTEM_RAM footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)



Figure 8.69. SYSTEM_RAM_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)

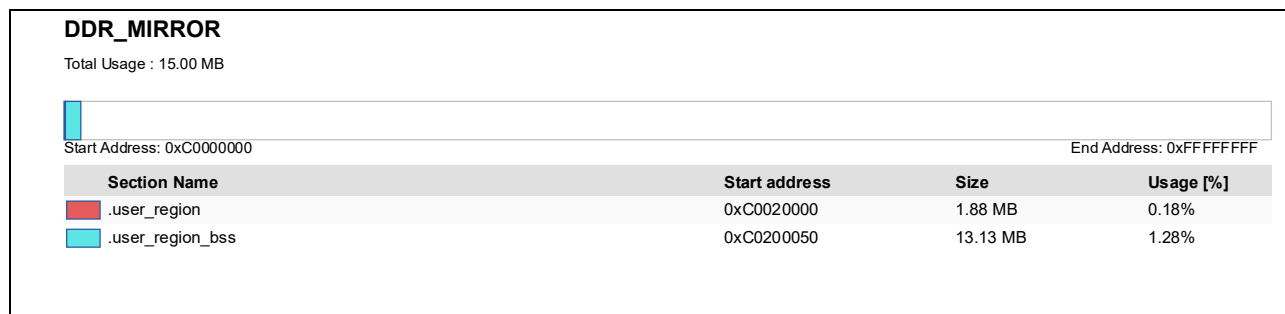


Figure 8.70. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)

8.3.4.2 EWARM

(1) Primary project

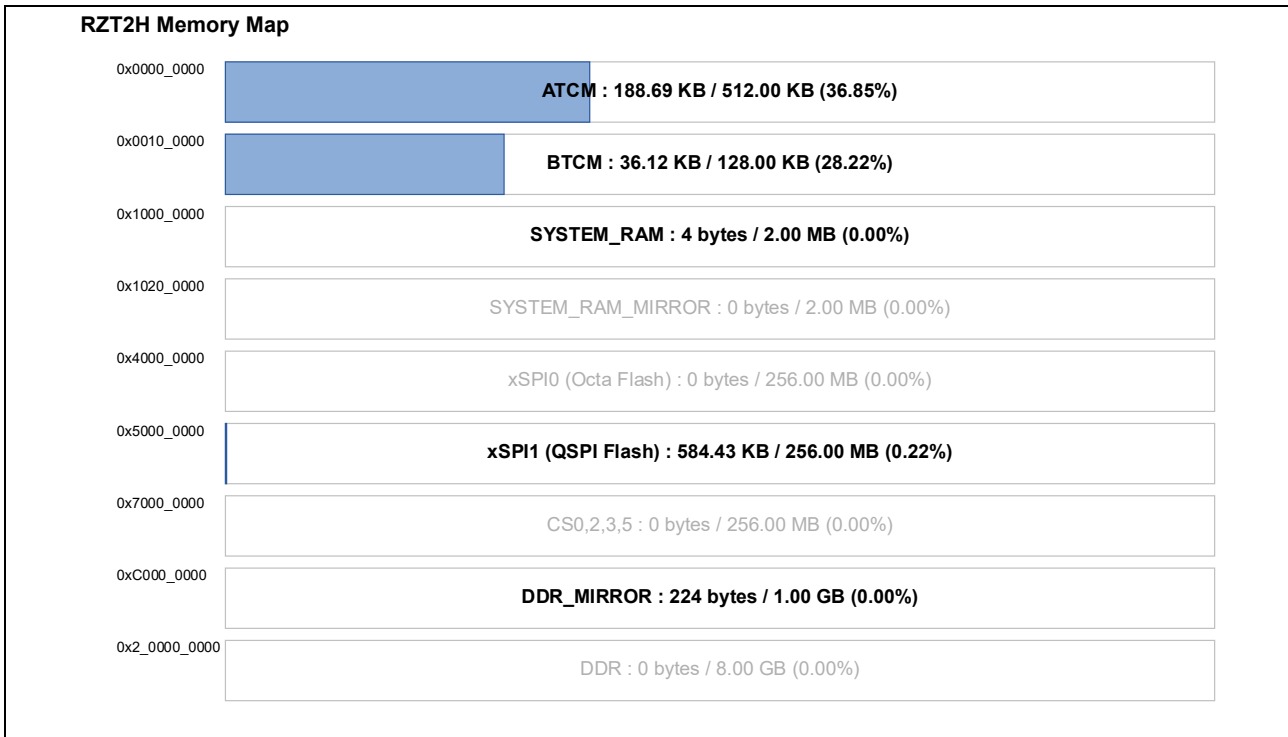


Figure 8.71. Memory footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)

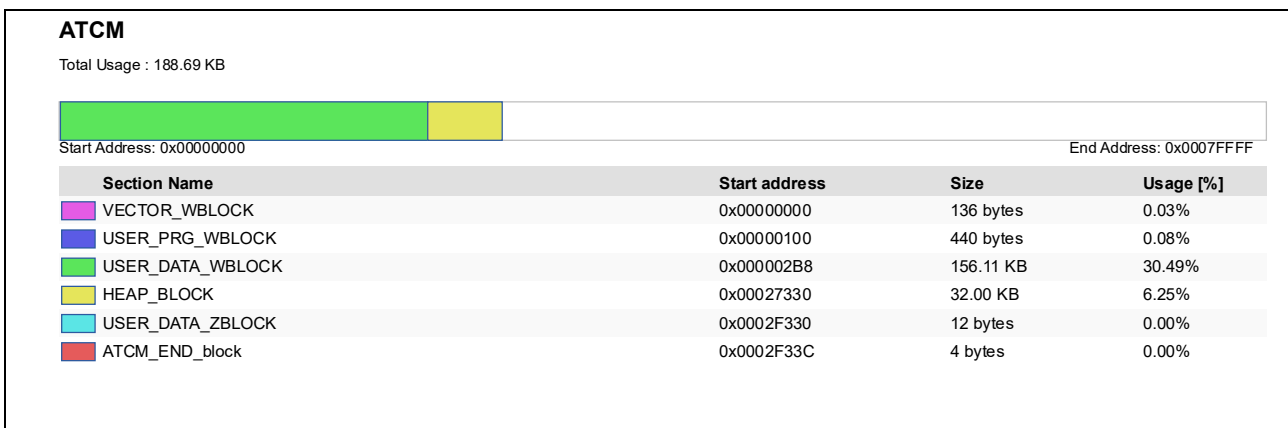


Figure 8.72. ATCM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)

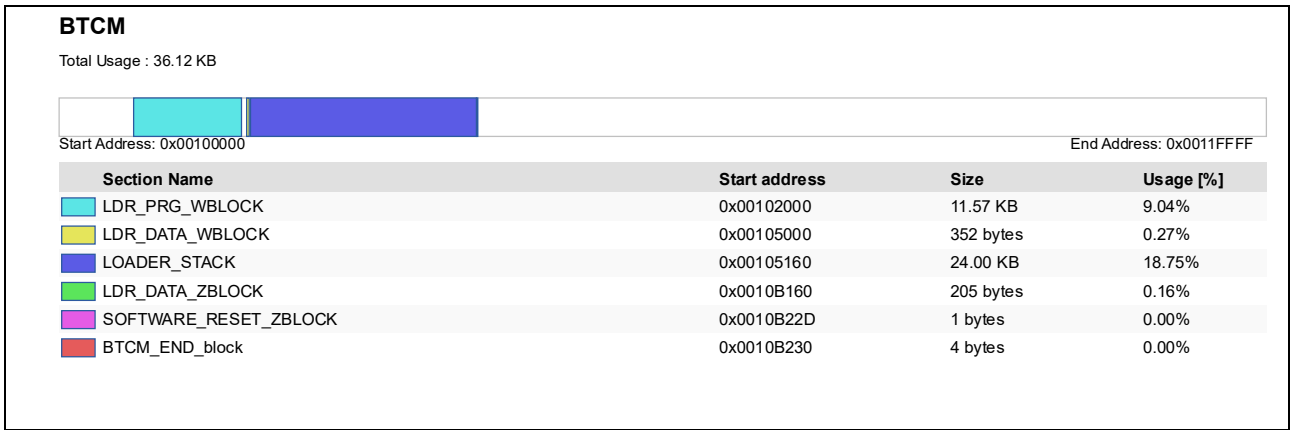


Figure 8.73. BTCM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)



Figure 8.74. SYSTEM_RAM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)

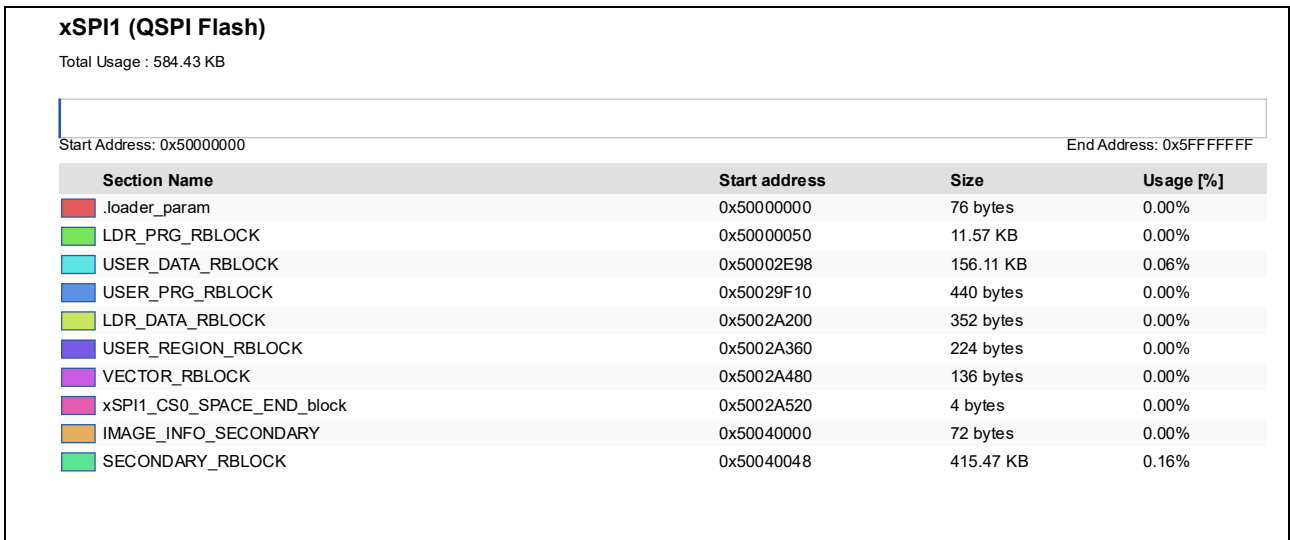


Figure 8.75. xSPI1 (QSPI Flash) footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)



Figure 8.76. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)

(2) Secondary project

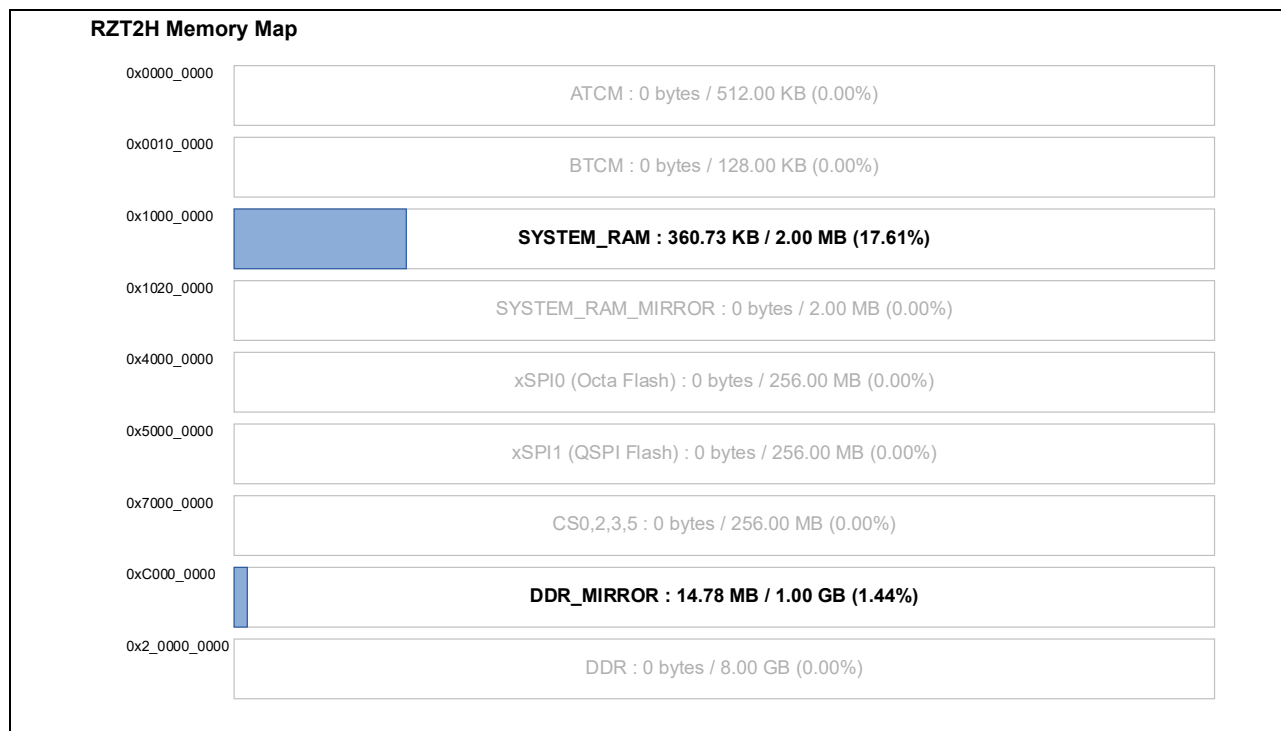


Figure 8.77. Memory footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)

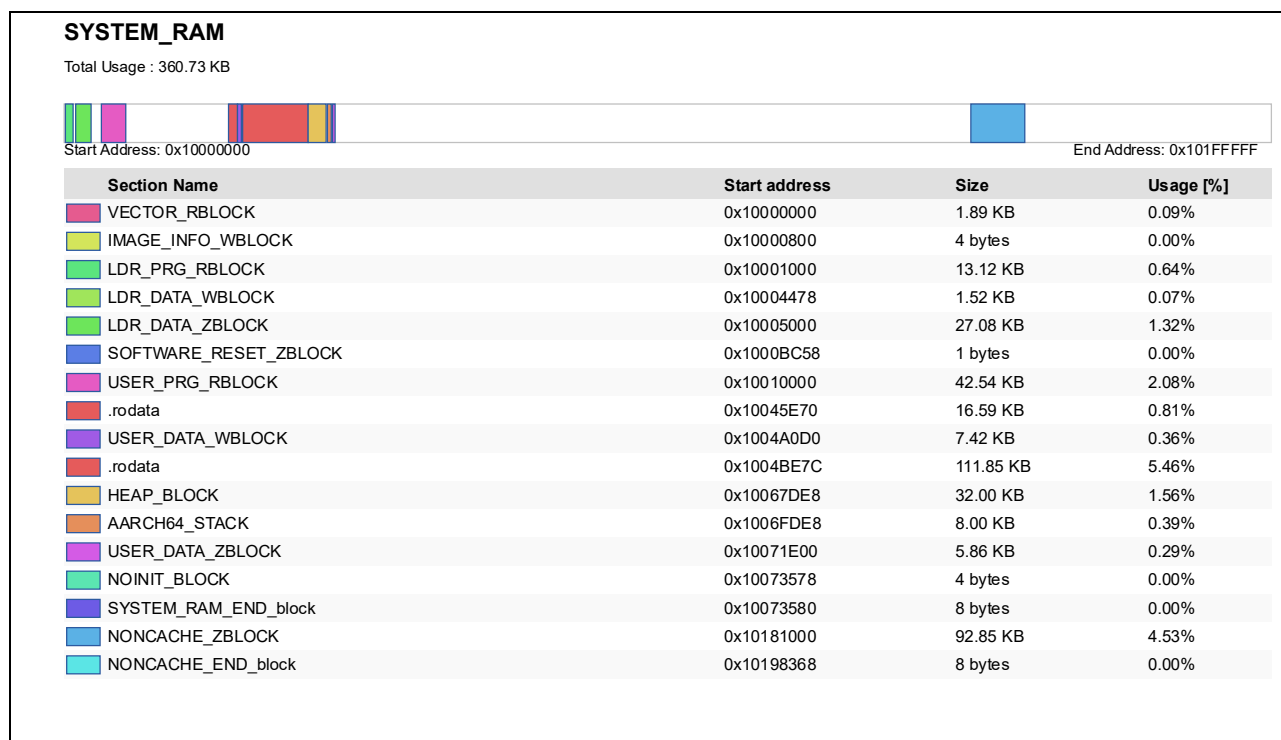


Figure 8.78. SYSTEM_RAM footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)

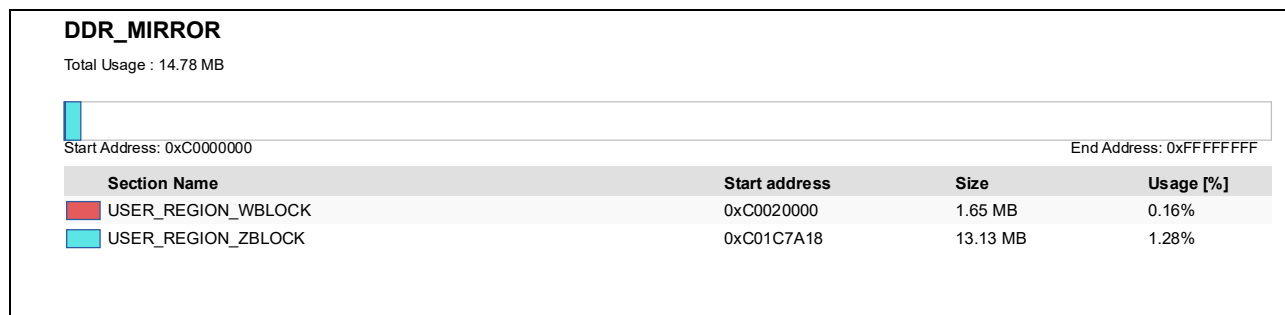


Figure 8.79. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)

9. Appendix

9.1 File Generation of open62541

本サンプルソフトの OPC UA スタックはオープンソースの open62541 を使用しています。open62541 を FreeRTOS + LwIP の環境で実行するため、下記 Link 先では CMake を使って open62541.c と open62541.h を生成する方法が推奨されており、本サンプルソフトでもこの方法を用いています。

[Building open62541 — open62541 1.3.0-dirty documentation](#)

本章では、Windows 環境で open62541 および Renesas サンプル情報モデルを e² studio もしくは EWARM で実行するためのファイルとして生成する手順について説明します。ここでは、WSL2 が実行可能な Windows10 バージョン 1903 以降(OS Build 18362.1049)、もしくは Windows11 を使用しています。

9.1.1 Linux environment Setup

CMake を実行するための Linux 環境を構築します。本書では、下記 Link 先ページを参考に WSL2 を使用して Linux (Ubuntu 18.04) をインストールした環境で CMake を実行します。

(参考) [以前のバージョンの WSL の手動インストール手順 | Microsoft Learn](#)

- 1) PowerShell を [管理者として実行] します。PowerShell を検索し、右クリックで [管理者として実行] を選択します。
- 2) 次のコマンドを入力し Linux 用 Windows サブシステムを有効にします。

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

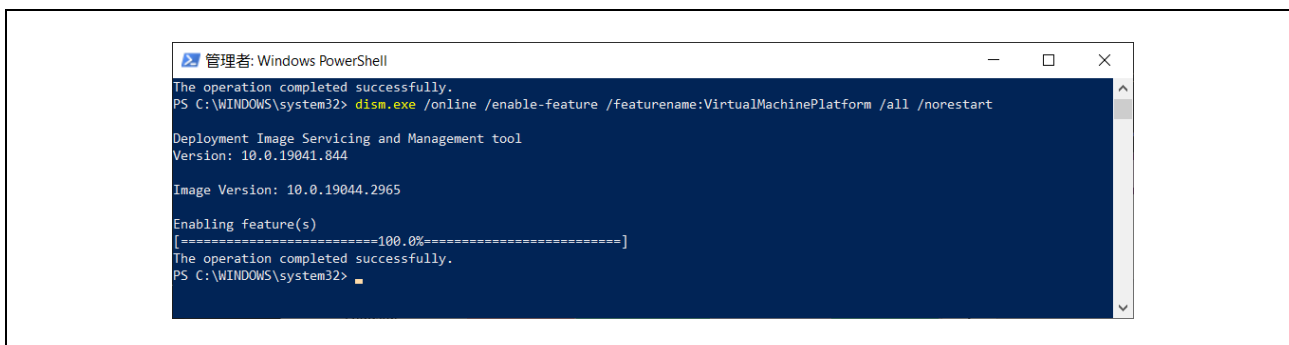


Figure 9.1. Microsoft-Windows-Subsystem-Linux

- 3) 次のコマンドを入力し仮想マシン プラットフォーム機能を有効にします。

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

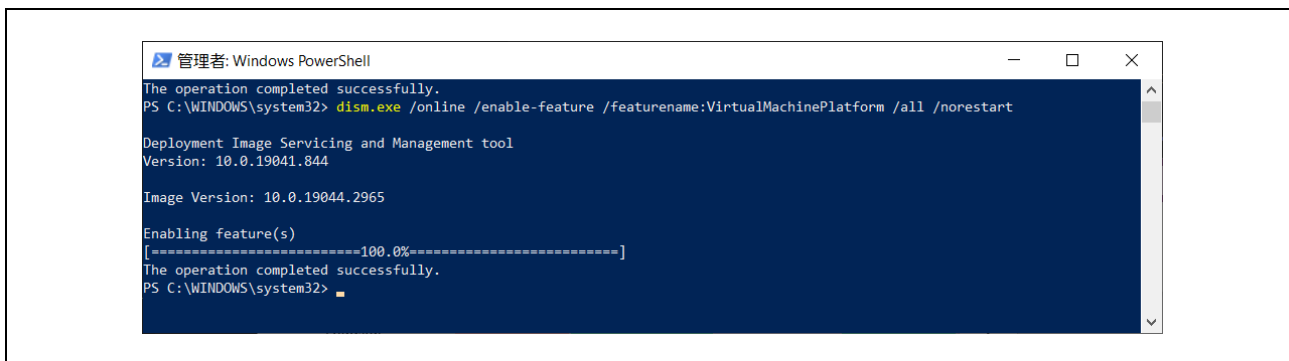


Figure 9.2. VirtualMachinePlatform

- 4) PC を再起動し、WSL のインストールを完了します。
- 5) 下記より x64 マシン用 WSL2 Linux カーネル更新プログラム パッケージをダウンロードして実行します。

[x64 マシン用 WSL2 Linux カーネル更新プログラム パッケージ](#)

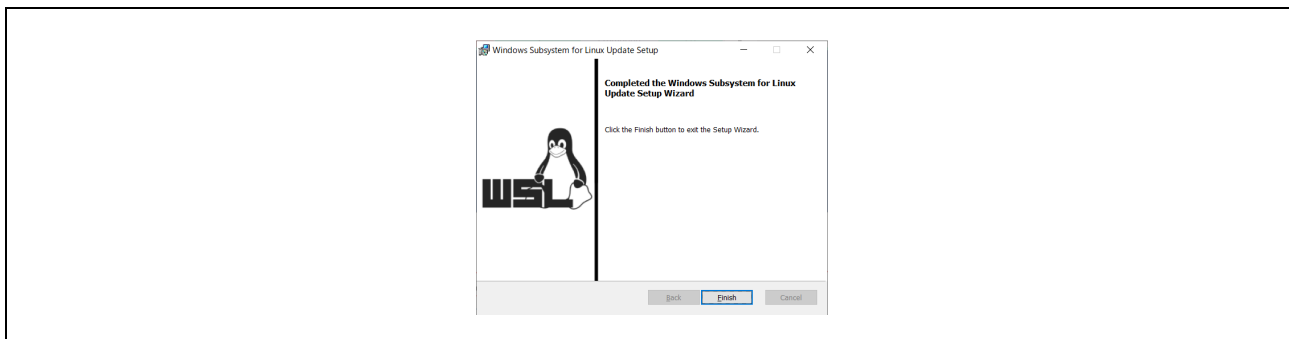


Figure 9.3. wsl_update_x64.msi

- 6) 下記コマンドを実行し、WSL 2 を既定のバージョンとして設定します。
`wsl --set-default-version 2`
- 7) Linux ディストリビューションをダウンロードします。ここでは以下より、Ubuntu 18.04 をダウンロードします。

[Ubuntu 18.04](#)

- 8) ダウンロードしたファイルを含むフォルダに移動し、下記コマンドを実行します。

`Add-AppxPackage . \Ubuntu_1804.2019.522.0_x64.appx`

- 9) Ubuntu_1804.2019.522.0_x64.appx をダブルクリックしてインストールします。

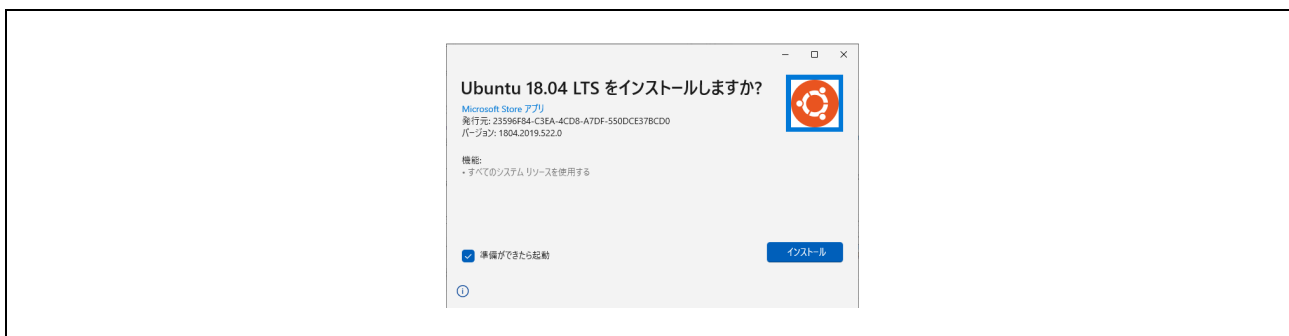


Figure 9.4. Ubuntu Install

- 10) Linux のユーザ名とパスワードを設定します。

(参考) [WSL 開発環境を設定する | Microsoft Learn](#)

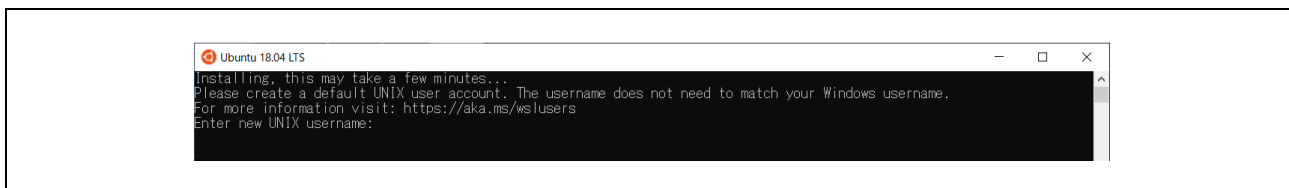


Figure 9.5. UNIX username

9.1.2 Install CMake

11) 下記 Linux コマンドを実行します。

```
sudo apt-get update
```

```

s@ubuntu-3CG3013V10:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2717 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [467 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1317 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [182 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1303 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [3045 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [308 kB]
Get:16 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.8 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [553 kB]
Get:18 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3928 B]
Get:19 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [1347 kB]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [187 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1914 kB]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [420 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [25.6 kB]
Get:24 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6088 B]
Get:25 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.3 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [14.6 kB]
Get:27 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.2 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 B]
Fetched 27.9 MB in 21s (1338 kB/s)
Reading package lists... Done

```

Figure 9.6. apt-get update

12) 下記コマンドを実行します。

```
sudo apt-get install git build-essential gcc pkg-config cmake python
```

```

s@ubuntu-3CG3013V10:~$ sudo apt-get install git build-essential gcc pkg-config cmake python
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cmake-data cpp cpp-7 dpkg-dev fakeroot g++ g++-7 gcc-7 gcc-7-base
  gcc-8-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libarchive13 libasan4 libatomic1
  libbinutils libc-dev-bin libc6 libc6-dev libcc1-0 libcilkrts5 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libgcc-7-dev libgcc1 libgomp1 libisl19 libitm1 libjsncpp1 liblsan0 libmpx2 libpython-stdlib
  libpython2.7-minimal libpython2.7-stdlib libquadmath0 librtmp1 libssl1.1 libstdc++-7-dev libstdc++6 libstdc++7
  libubsan0 linux-libc-dev make manpages-dev python-minimal python2.7 python2.7-minimal
Suggested packages:
  binutils-doc cmake-doc ninja-build cpp-doc gcc-7-locale debian-keyring g++-multilib g++-7-multilib gcc-7-doc
  libstdc++6-7-dbg gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-7-multilib libgcc1-dbg
  libgomp1-dbg liblinal-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg libstdc++6-dbg libubsan0-dbg libcilkrts5-dbg
  libmpx2-dbg libquadmath0-dbg git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn lrzip glibc-doc brr libstdc++-7-doc make-doc python-doc python-tk python2.7-doc
  binfmt-support
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cmake cmake-data cpp cpp-7 dpkg-dev fakeroot g++

```

Figure 9.7. install

途中で下記画面がでたら OK を選択します。

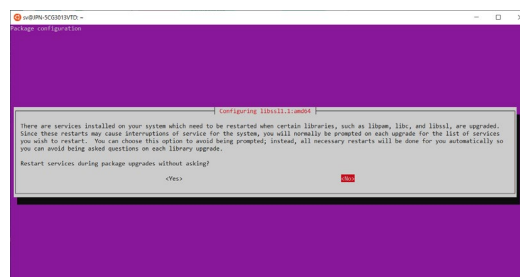


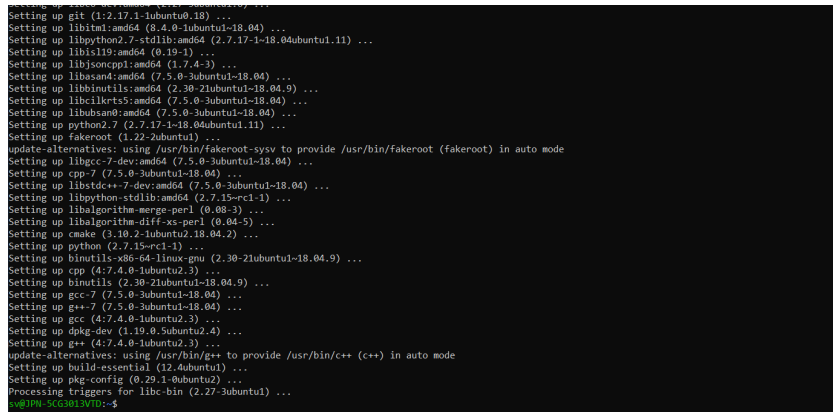
Figure 9.8. restart

13) 下記コマンドをそれぞれ実行します。

```

sudo apt-get install cmake-curses-gui           # Needed for CMAKE GUI
sudo apt-get install libmbdtdls-dev            # For encryption
sudo apt-get install liburcu-dev              # For multithreading
sudo apt-get install check                     # For unit tests
sudo apt-get install python-sphinx graphviz   # For doc generation
sudo apt-get install python-sphinx-rtd-theme  # For doc's style

```



```

Setting up git (1:2.17.1-1ubuntu0.18) ...
Setting up libitm1:amd64 (8.4.0-1ubuntu1-18.04) ...
Setting up libpython2.7-stdeb:amd64 (2.7.17-1-18.04ubuntu1.11) ...
Setting up libisl19:amd64 (0.19-1) ...
Setting up libjsoncpp1:amd64 (1.7.4-3) ...
Setting up libasan4:amd64 (7.5.0-3ubuntu1-18.04) ...
Setting up libbinutils:amd64 (2.30-21ubuntu1-18.04.9) ...
Setting up libckit5:amd64 (7.5.0-3ubuntu1-18.04) ...
Setting up libubsan0:amd64 (7.5.0-3ubuntu1-18.04) ...
Setting up python2.7 (2.7.17-1-18.04ubuntu1.11) ...
Setting up fakeroot (1.22-2ubuntu1) ...
update-alternatives: using /usr/bin/fakeroot-sysv to provide /usr/bin/fakeroot (fakeroot) in auto mode
Setting up libgcc-7-dev:amd64 (7.5.0-3ubuntu1-18.04) ...
Setting up cpp-7 (7.5.0-3ubuntu1-18.04) ...
Setting up libstdc++-7-dev:amd64 (7.5.0-3ubuntu1-18.04) ...
Setting up libpython2.7-dev:amd64 (2.7.15rc1-1) ...
Setting up libalgorithm-diff-perl (0.08-3) ...
Setting up libalgorithm-diff-xs-perl (0.04-5) ...
Setting up cmake (3.10.2-1ubuntu2.18.04.2) ...
Setting up python (2.7.15rc1-1) ...
Setting up binutils-x86-64-linux-gnu (2.30-21ubuntu1-18.04.9) ...
Setting up cpp (4:7.4.0-1ubuntu2.3) ...
Setting up binutils (2.30-21ubuntu1-18.04.9) ...
Setting up gcc-7 (7.5.0-3ubuntu1-18.04) ...
Setting up g++-7 (7.5.0-3ubuntu1-18.04) ...
Setting up gcc (4:7.4.0-1ubuntu2.3) ...
Setting up dpkg-dev (1.19.0-ubuntu2.4) ...
Setting up g++ (4:7.4.0-1ubuntu2.3) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.4ubuntu1) ...
Setting up pkg-config (0.29.1-0ubuntu2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
sv@JPN-5CG3013VTD:~$

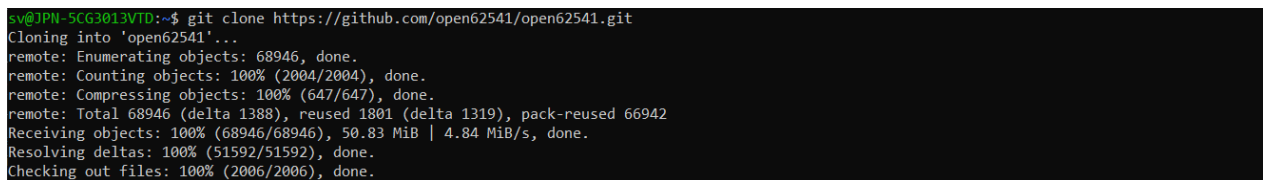
```

Figure 9.9. install

9.1.3 open62541 File Generation

14) open62541 を任意のフォルダにクローンを生成します。

```
git clone https://github.com/open62541/open62541.git
```



```

sv@JPN-5CG3013VTD:~$ git clone https://github.com/open62541/open62541.git
Cloning into 'open62541'...
remote: Enumerating objects: 68946, done.
remote: Counting objects: 100% (2004/2004), done.
remote: Compressing objects: 100% (647/647), done.
remote: Total 68946 (delta 1388), reused 1801 (delta 1319), pack-reused 66942
Receiving objects: 100% (68946/68946), 50.83 MiB | 4.84 MiB/s, done.
Resolving deltas: 100% (51592/51592), done.
Checking out files: 100% (2006/2006), done.
sv@JPN-5CG3013VTD:~$

```

Figure 9.10. git clone

15) /open62541 ディレクトリに移動し、Git submodule で特定のコミットバージョン(ここでは、v1.3.15) を参照します。

```

cd open62541
git log -1
git checkout 3eed1a6d5c5b207c531b2d35ed88aa0a4a4541e5
git submodule init
git submodule update

```

```

sv@IPM-5CG3013VM:~$ cd open62541
sv@IPM-5CG3013VM:~/open62541$ git log -1
commit 6287f35545e397b1a7384906859f5b504dbdc25 (HEAD -> master, origin/master, origin/HEAD)
Merge: 25806dad84 aab096ca5
Author: Julius Pfromen <jpfr@users.noreply.github.com>
Date: Fri Jul 14 12:13:36 2023 +0200

Merge pull request #5877 from open62541/1.4

Merge 1.4 to master
sv@IPM-5CG3013VM:~/open62541$ git checkout b7e5e49f32d00490be74c2eacef892c7fbd0be60
Checking out files: 100% (3358/3358), done.
Note: checking out 'b7e5e49f32d00490be74c2eacef892c7fbd0be60'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

git checkout -b <new-branch-name>

HEAD is now at b7e5e49f3 [ci skip] Pack with inline submodules
sv@IPM-5CG3013VM:~/open62541$ git submodule init
Submodule 'deps/mqtt-c' (https://github.com/LiamBindle/MQTT-C.git) registered for path 'deps/mqtt-c'
sv@IPM-5CG3013VM:~/open62541$ git submodule update
Cloning into '/home/sv/open62541/deps/mqtt-c'...
Submodule path 'deps/mqtt-c': checked out 'f69ce1e7f54f3b1834c9c9137ce0ec5d703cb4d'
sv@IPM-5CG3013VM:~/open62541$ git log -1
commit b7e5e49f32d00490be74c2eacef892c7fbd0be60 (HEAD, origin/pack/v1.3.4)
Author: github-actions[bot] <41898282+github-actions[bot]@users.noreply.github.com>
Date: Mon Nov 14 12:28:23 2022 +0800

[ci skip] Pack with inline submodules

```

Figure 9.11. git submodule

- 16) File Explorer から Linux フォルダを開きます。/home/(username)/open62541 ディレクトリに CMakeLists.txt があることを確認します。サンプルソフトに添付の patch_open62541.zip を解凍すると得られる下記 3 つのファイルをここにコピーします。

```

diff.patch
Opc.Ua.Renesas.NodeSet2.xml
patch.sh

```

- 17) /open62541 ディレクトリで下記コマンドを実行します。

```
bash patch.sh
```

```

xxxxx @IPM-5CG3013VM:~/home/xxxxx/tst/open62541$ bash patch.sh
patching file CMakeLists.txt
patching file tools/schema/Opc.Ua.NodeSet2.Reduced.xml

```

Figure 9.12. command result

- 18) cmake プロジェクトの標準的な手順に従いライブラリをコンパイルします。/open62541/build ディレクトリを作成し、cmake .. を実行します。(Failed になる項目もありますが問題ありません)

```
mkdir build && cd build
cmake ..
```

```

xxxxx @IPM-5CG3013VM:~/home/xxxxx/tst/open62541/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Found Python3: /usr/bin/python3.8 (found version "3.8.10") found components: Interpreter
-- Found Git: /usr/bin/git (found version "2.29.1")
-- open62541 Version: v1.3.4-1-ab7e49f3-dirty
-- CMAKE_BUILD_TYPE not given; setting to 'Debug'
-- The selected architecture is: posix
-- Test CC flag -std=c99
-- Performing Test flag_supported
-- Performing Test flag_supported - Success
-- Test CC flag -pipe
-- Performing Test flag_supported
-- Performing Test flag_supported - Success
-- Test CC flag -Wall
-- Performing Test flag_supported
-- Performing Test flag_supported - Success

```

Figure 9.13. cmake

19) 下記コマンドを実行し `ccmake` の設定 Window を起動します。

```
ccmake ..
```

20) Figure 9.14 の通り設定を変更した後、[t] to toggle advanced mode を押し、"UA_ENABLE_PUBSUB_ENCRYPTION"を ON に変更します。[c] to configure を実行後、[q] to quit without generating で閉じます。

```

BUILD_SHARED_LIBS           OFF
CLANG_FORMAT_EXE           CLANG_FORMAT_EXE-NOTFOUND
CMAKE_BUILD_TYPE           Debug
CMAKE_INSTALL_PREFIX       /usr/local
UA_ARCHITECTURE            freertosLWIP
UA_BUILD_EXAMPLES         OFF
UA_BUILD_TOOLS             OFF
UA_BUILD_UNIT_TESTS       OFF
UA_ENABLE_AMALGAMATION     ON
UA_ENABLE_DA               OFF
UA_ENABLE_DIAGNOSTICS     OFF
UA_ENABLE_DISCOVERY       OFF
UA_ENABLE_ENCRYPTION      MBEDTLS
UA_ENABLE_ENCRYPTION_TPM2 OFF
UA_ENABLE_HISTORIZING     OFF
UA_ENABLE_JSON_ENCODING   ON
UA_ENABLE_METHODCALLS    ON
UA_ENABLE_PUBSUB          ON
UA_ENABLE_PUBSUB_DELTAFRAMES OFF
UA_ENABLE_PUBSUB_ETH_UADP OFF
UA_ENABLE_PUBSUB_INFORMATIONMO ON
UA_ENABLE_PUBSUB_INFORMATIONMO ON
UA_ENABLE_SUBSCRIPTIONS   ON
UA_ENABLE_SUBSCRIPTIONS_EVENTS ON
UA_FORCE_WARNOR           ON
UA_LOGLEVEL               500
UA_MULTITHREADING         0
UA_NAMESPACE_ZERO        REDUCED
  
```

Figure 9.14. `ccmake` for OPC UA Server, PubSub

```

UA_ENABLE_PUBSUB          ON
UA_ENABLE_PUBSUB_BUFMALLOC OFF
UA_ENABLE_PUBSUB_DELTAFRAMES OFF
UA_ENABLE_PUBSUB_ENCRYPTION ON
UA_ENABLE_PUBSUB_ETH_UADP OFF
UA_ENABLE_PUBSUB_FILE_CONFIG OFF
UA_ENABLE_PUBSUB_INFORMATIONMO ON
UA_ENABLE_PUBSUB_INFORMATIONMO ON
UA_ENABLE_PUBSUB_INFORMATIONMODEL: Enable PubSub information model twin
Press [enter] to edit option Press [d] to delete an entry
Press [c] to configure
Press [h] for help          Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently On)
  
```

Figure 9.15. `ccmake` for OPC UA PubSub Encryption

21) 下記コマンドで `make` します。 `make` 自体はエラー終了しますが、問題ありません。

```
make -j
```

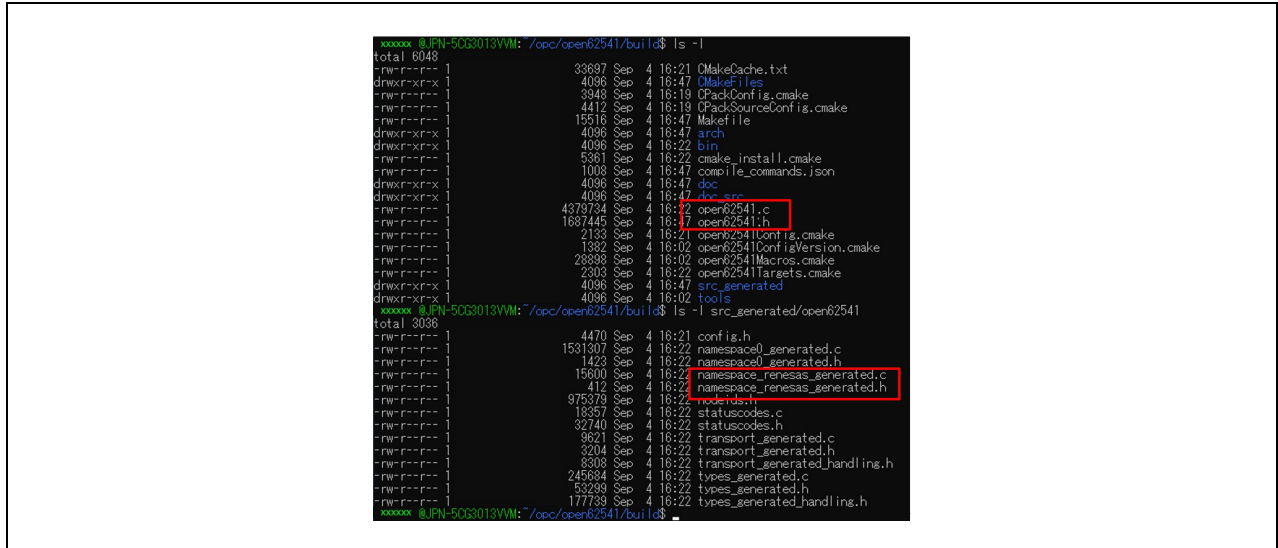
```

Configuring done
Generating done
-- Build files have been written to: //home/xxxxxx/tst/oper62541/build
Scanning dependencies of target oper62541-generator-types
Scanning dependencies of target oper62541-generator-statuscode
Scanning dependencies of target oper62541-generator-transport
 44] Generating src_generated/oper62541/types_generated.c, src_generated/oper62541/types_generated.h, src_generated/oper62541/types_generated_handling.h
 82] Generating src_generated/oper62541/nodesids.h
102] Generating src_generated/oper62541/transport_generated.c, src_generated/oper62541/transport_generated.h, src_generated/oper62541/transport_generated_handling.h
102] Generating src_generated/oper62541/statuscodes.h, src_generated/oper62541/statuscodes.c
102] Built target oper62541-generator-statuscode
102] Built target oper62541-generator-transport
102] Built target oper62541-generator-types
Scanning dependencies of target oper62541-generator-namespaces
202] Generating src_generated/oper62541/namespaces_generated.c, src_generated/oper62541/namespaces_generated.h
INFO: main: Preprocessing //home/xxxxxx/tst/oper62541/tools/schema/OpC.Ua.NodeSet2.Reduced.xml
INFO: main: Preprocessing //home/xxxxxx/tst/oper62541/tools/schema/OpC.Ua.NodeSet2.EventMinimal.xml
INFO: main: Skipping Nodeset since it is already loaded: //home/mashiko/tst/oper62541/tools/schema/OpC.Ua.NodeSet2.EventMinimal.xml
INFO: main: Skipping Nodeset since it is already loaded: //home/mashiko/tst/oper62541/tools/schema/OpC.Ua.NodeSet2.Part0_Subset.xml
INFO: main: Generating code for Nodeset oper62541
INFO: main: Nodeset generation code successfully printed
202] Built target oper62541-generator-namespaces
Scanning dependencies of target oper62541-amalgamation-source
202] Generating src_generated/oper62541/namespaces_renesas_generated.c, src_generated/oper62541/namespaces_renesas_generated.h
202] Generating oper62541.h
Starting amalgamation file //home/xxxxxx/tst/oper62541/build/oper62541.h
  
```

Figure 9.16. `make`

22) `ls` コマンドで `/open62541/build` ディレクトリおよび `./src_generated/open62541` ディレクトリに下記ファイルが生成されていることを確認します。

- `open62541.c`
- `open62541.h`
- `namespace_renesas_generated.c`
- `namespace_renesas_generated.h`



```
xxxxxx @IPN-50G3013WMM: /opc/open62541/build$ ls -l
total 6048
-rw-r--r-- 1 33687 Sep  4 16:21 OMakeCache.txt
drwxr-xr-x 1 4096 Sep  4 16:47 OMakeFiles
-rw-r--r-- 1 3948 Sep  4 16:19 OPackConfig.cmake
-rw-r--r-- 1 4412 Sep  4 16:19 OPackSourceConfig.cmake
-rw-r--r-- 1 15518 Sep  4 16:47 Makefile
drwxr-xr-x 1 4096 Sep  4 16:47 arch
drwxr-xr-x 1 4096 Sep  4 16:22 bin
-rw-r--r-- 1 5361 Sep  4 16:22 cmake_install.cmake
-rw-r--r-- 1 1008 Sep  4 16:47 compile_commands.json
drwxr-xr-x 1 4096 Sep  4 16:47 doc
drwxr-xr-x 1 4096 Sep  4 16:47 doc_src
-rw-r--r-- 1 4379734 Sep  4 16:22 open62541.c
-rw-r--r-- 1 1687445 Sep  4 16:17 open62541.h
-rw-r--r-- 1 2133 Sep  4 16:21 open62541Config.cmake
-rw-r--r-- 1 1392 Sep  4 16:02 open62541ConfigVersion.cmake
-rw-r--r-- 1 28886 Sep  4 16:02 open62541Macros.cmake
-rw-r--r-- 1 2303 Sep  4 16:22 open62541Targets.cmake
drwxr-xr-x 1 4096 Sep  4 16:47 src_generated
drwxr-xr-x 1 4096 Sep  4 16:02 tools
xxxxxx @IPN-50G3013WMM: /opc/open62541/build$ ls -l src_generated/open62541
total 3036
-rw-r--r-- 1 4470 Sep  4 16:21 config.h
-rw-r--r-- 1 1531307 Sep  4 16:22 namespace0_generated.c
-rw-r--r-- 1 1423 Sep  4 16:22 namespace0_generated.h
-rw-r--r-- 1 15600 Sep  4 16:22 namespace_renesas_generated.c
-rw-r--r-- 1 412 Sep  4 16:22 namespace_renesas_generated.h
-rw-r--r-- 1 975379 Sep  4 16:22 nodes.h
-rw-r--r-- 1 18357 Sep  4 16:22 statuscodes.c
-rw-r--r-- 1 32740 Sep  4 16:22 statuscodes.h
-rw-r--r-- 1 3621 Sep  4 16:22 transport_generated.c
-rw-r--r-- 1 3204 Sep  4 16:22 transport_generated.h
-rw-r--r-- 1 8308 Sep  4 16:22 transport_generated_handling.h
-rw-r--r-- 1 245634 Sep  4 16:22 types_generated.c
-rw-r--r-- 1 53239 Sep  4 16:22 types_generated.h
-rw-r--r-- 1 177739 Sep  4 16:22 types_generated_handling.h
xxxxxx @IPN-50G3013WMM: /opc/open62541/build$
```

Figure 9.17. Generated Files

23) 生成されたファイルをプロジェクトの `common/oss/OPC_UA_SERVER` 内に格納してください。

9.1.4 Changes in Generated Files

本プロジェクトでは、本手順で生成した `open62541.c, h` のファイルに対して、一部変更を行っています。変更前のファイルは `"open62541_before.zip"` に格納されているため、変更内容を確認する場合はプロジェクト内のファイルとの比較をしてください。

9.2 TFTP

本サンプルソフトは TFTP Server 機能を持っています。動作確認では、評価ボードに接続する Windows PC を TFTP Client として使用しています。Windows の TFTP Client 動作はデフォルトでは無効になっているのでご注意ください。

TFTP はパスワード設定ができないため、保存先のフォルダ名を簡易的なパスワードとして使用していません。初期パスワードは"passw0rd"です。tftp_client.h 内に TFTP_PASSWORD で定義されており、変更可能です。

9.2.1 File Writing

コマンドプロンプトを立ち上げ、書き込みするファイルが存在するフォルダまで移動します。例として、cert_server.der を書き込みする場合は下記のようにコマンドを打ちます。"Transfer successful" と表示されれば書き込み成功です。評価ボードに接続された USB フラッシュドライブの<USB_ROOT>/passw0rd フォルダに cert_server.der が書き込みされます。

```
tftp -i 192.168.10.100 put cert_server.der passw0rd/cert_server.der
```

9.2.2 File Reading

コマンドプロンプトを立ち上げます。例として、USB フラッシュドライブ内の <USB_ROOT>/passw0rd/cert_server.der を読み出しする場合は下記のようにコマンドを打ちます。読み出しに成功すると、"Transfer successful" と表示され、カレントディレクトリに cert_server.der が格納されます。

```
tftp -i 192.168.10.100 get passw0rd/cert_server.der
```

9.3 LLDP, SNMP MIB

Table 9.1 は、本サンプルソフトの LLDP 機能と、対応する MIB を表しています。

Table 9.1. Supported LLDP and MIB

TLV Type	TLV name	TLV Variable	Transmit Contents	LLDP Local System MIB object	LLDP Remote System MIB object
0	End of LLDPDU	-	-	-	-
1	Chassis ID	chassis ID subtype	MAC address	lldpLocChassisIdSubtype	lldpRemChassisSubtype
		chassis ID	"00:11:22:33:44:55"	lldpLocChassisId	lldpRemChassisId
2	Port ID	port ID subtype	Interface name	lldpLocPortIdSubtype	lldpRemPortIdSubtype
		port ID	"PORT 1" or "PORT 2"	lldpLocPortId	lldpRemPortId
3	Time To Live	Seconds	120	-	-
4	Port description	Port description	"PORT 1" or "PORT 2"	lldpLocPortDesc	lldpRemPortDesc
5	System name	System name	"OPC UA Server/PubSub"	lldpLocSysName	lldpRemSysName
6	System description	System description	"Renesas Electronics Corporation OPC UA Sample"	lldpLocSysDesc	lldpRemSysDesc
7	System capabilities	system capabilities	Station only (bit7=1)	lldpLocSysCapSupported	lldpRemSysCapSupported
		enabled capabilities	Station only (bit7=1)	lldpLocSysCapEnabled	lldpRemSysCapEnabled
8	Management address	management address length	5	lldpLocManAddrLen	-
		management address subtype	IPv4	lldpLocManAddrSubtype	lldpRemManAddrSubtype
		management address	192.168.10.100	lldpLocManAddr	lldpRemManAddr
		interface numbering subtype	systemPortNumber	lldpLocManAddrIfSubtype	lldpRemManAddrIfSubtype
		interface number	1 or 2	lldpLocManAddrIfId	lldpRemManAddrIfId
		OID	1.3.6.1.4.1.26381	lldpLocManAddrOID	lldpRemManAddrOID

9.4 When configuring OPC UA PubSub settings using the Method

本章では、Method を使用して OPC UA PubSub 設定を行う場合の説明をします。まず Method を使用しての PubSub 設定の場合、本ソフトウェアの制限事項として、Pub もしくは Sub のレイアウトは自由に設定することができず、特定のカスタムレイアウトとなります。そのため、他デバイスと通信する際は、通信相手側でレイアウトを合わせる必要があります。また、本機能による PubSub 通信はセキュリティ None のみサポートします。

9.4.1 Publisher

OPC UA Publisher 動作確認時の、それぞれのデバイスの役割は Figure 9.18 に示す通りです。

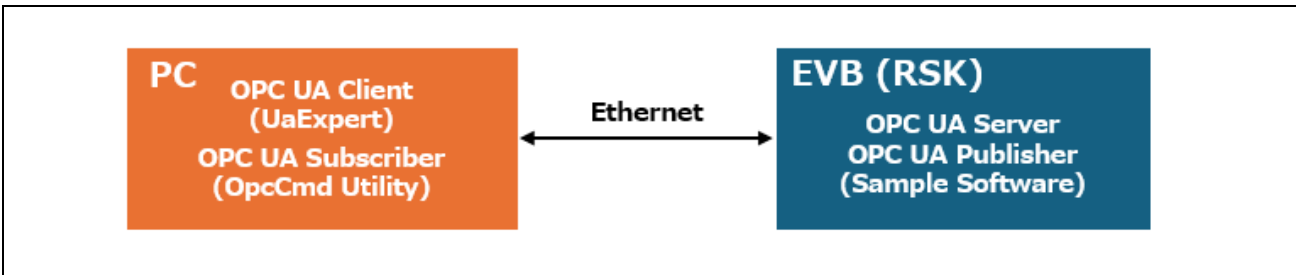


Figure 9.18. Connection Block Diagram for OPC UA Publisher

4 章の通りに接続した後、下記の手順で動作確認を行ってください。

- ・ UaExpert の立ち上げ、接続

7.2.1 章と同様の手順で、UaExpert と OPC UA Server を接続します。

Address Space ウィンドウの Object ツリーに表示された
Root>Objects>Server>PublishSubscribe>PublishedDataSets>AddPublishedDataItems Node を右クリックし、表示された“Call...”をクリックします。

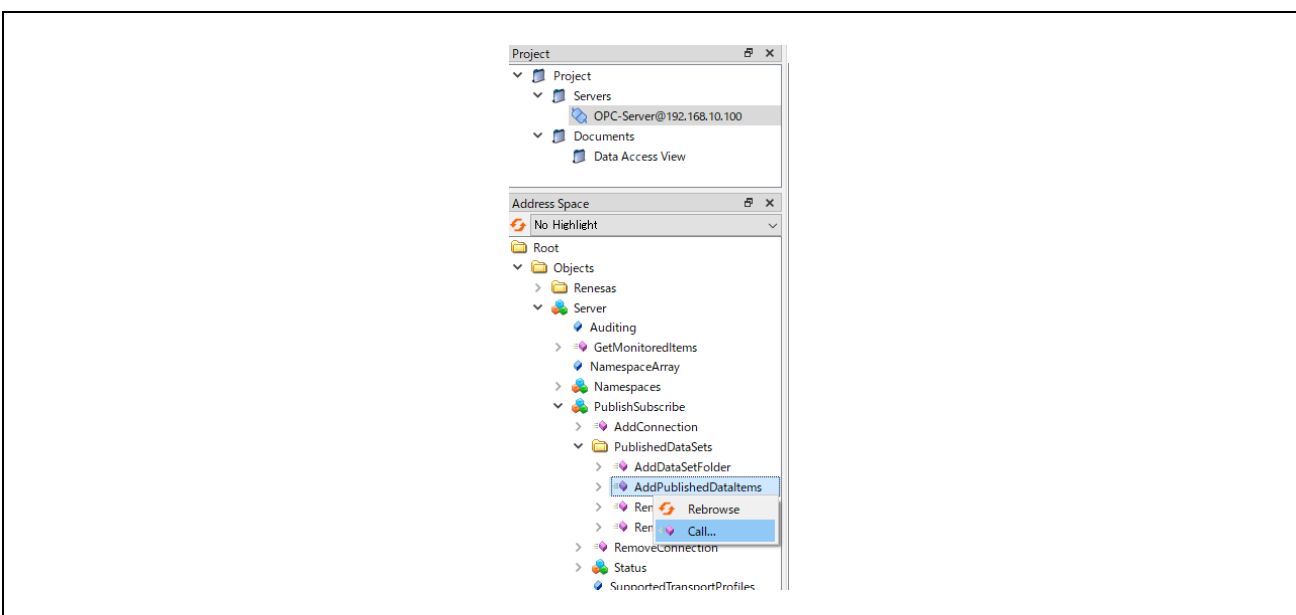


Figure 9.19. Address Space Window

表示されたウィンドウに、Publish したい Node の情報を入力し、Call をクリックします。Figure 9.20 は例として、Boolean(NS=2, ID=6049), Int32_1(NS=2, ID=6050), Int32_2(NS=2, ID=6051), DateTime(NS=2, ID=6052)の 4 つを設定しています。

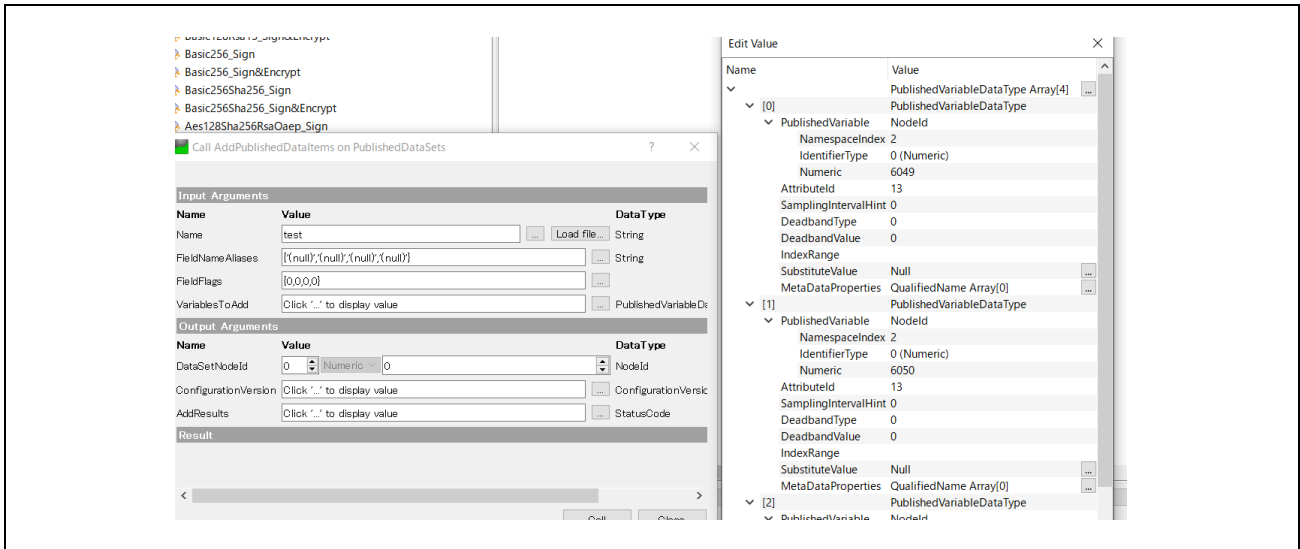


Figure 9.20. Call AddPublishedDataItems

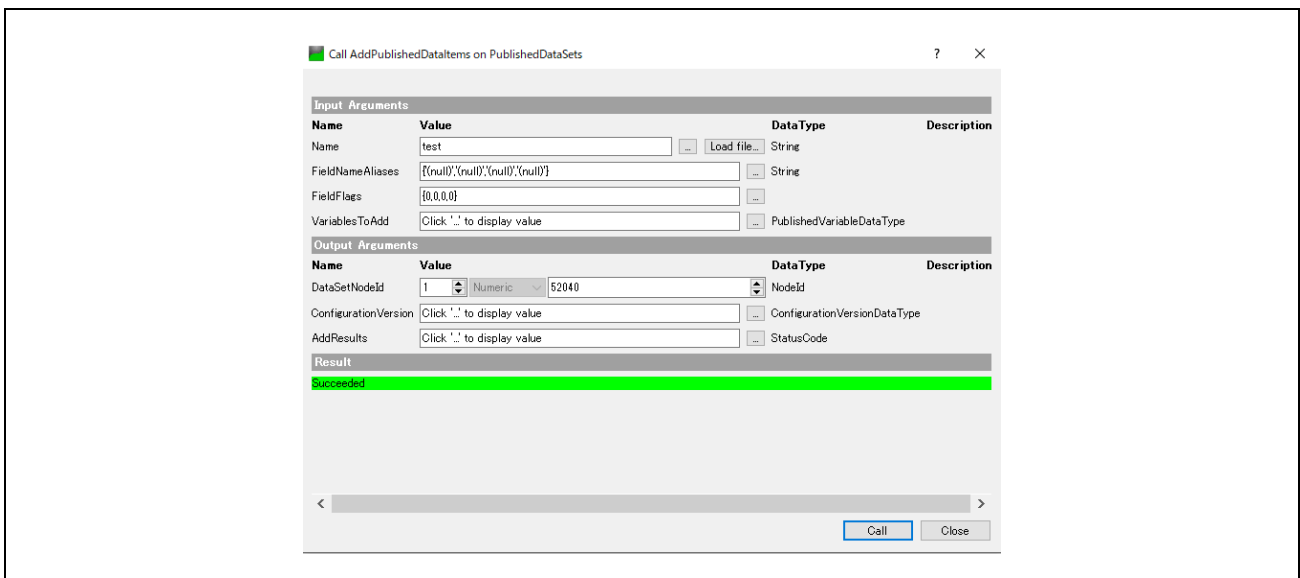


Figure 9.21. Call Result

・ OPC UA Client for PubSub Connection の実行

パッケージに含まれている“pub.exe”を、Linux 上で実行します。実行が成功すると、Publish が開始されます。

`./pub.exe`

```

~/client_build$ ./pub.exe
Endpoint URI:          opc.tcp://192.168.10.100:4840
NetworkAddressURL:    opc.udp://239.0.0.1:4840/
TransportProfile:     http://opcfoundation.org/UA-Profile/Transport/pubsub-udp-uadb
PublishingInterval:   100.0ms
PublisherID:          2234
WriterGroupID:        100
DataSetWriterID:      62541
PublishedDataSetName: test
Connection,WriterGroup Setting : Success
ConnectionID : Namespace:1, ID:50758

```

Figure 9.22 Result of “./pub.exe”

・ OpcCmd Utility の立ち上げ

コマンドプロンプトを立ち上げ、OpcCmd Utility の展開先に移動します。そこで下記コマンドを入力してください。本コマンドは、“opc.udp://239.0.0.1”宛てに Publish された全てのメッセージを Subscribe するためのものです。Subscribe は 1 分間続きます。

OpcCmd uaSubscriber subscribeDataSet -cru opc.udp://239.0.0.1

```

OPC Labs OPC Command-Line Tool (.NET Framework) 5.82.0.21
Executing for 00:01:00 (press 'X' to terminate)...
Subscribing dataset (default event IEasyUASubscriber.DataSetMessage)
Dataset handle: 13000001
: Success
: Success; 2025-09-18 03:06:08.493,863,600,00; Good; Data; publisher=[UInt32]2234, group=100, writer=62541, origin=192.168.10.100:4840, mapping=Uadb, fields: 4
Field data (sequence): 4 element(s)

```

[]	Source Timestamp Local	Server Timestamp Local	Status Code	Value Type Name	Value
#0	2025-09-18 12:06:08.493		Good	Boolean	True
#1	2025-09-18 12:06:08.493		Good	Int32	231
#2	2025-09-18 12:06:08.493		Good	Int32	-2147483417
#3	2025-09-18 12:06:08.493		Good	DateTime	2025/09/18 3:06:08

Figure 9.23. Command Execution Result

9.4.2 Subscriber

Subscribe 機能の確認には、本サンプルソフトに対応した評価ボードが2枚必要になります。接続については Figure 9.24 もしくは Figure 9.25 を参照ください。本章では、PC (OPC UA Client)と接続されている評価ボードを EVB1(RSK1)、もう一枚の評価ボードを EVB2(RSK2)と表現し、EVB2(RSK2)を OPC UA Publisher として使用します。

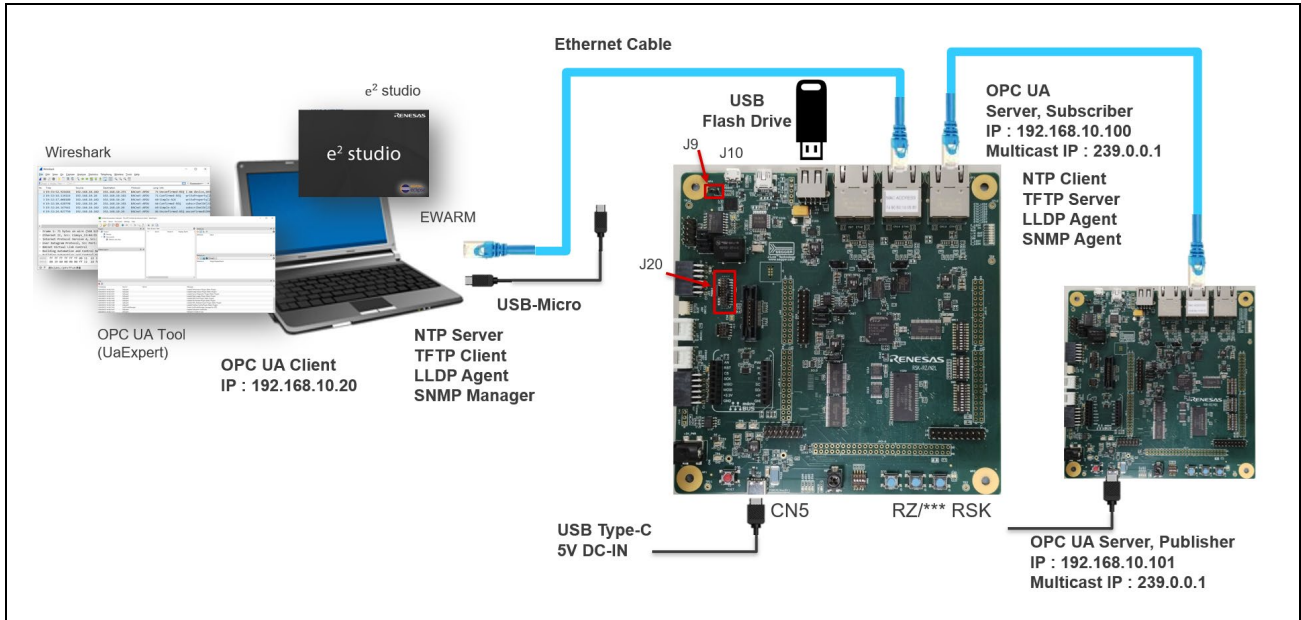


Figure 9.24. OPC UA Server Hardware Diagram for Confirming Subscriber (RZ/N2L, T2M)

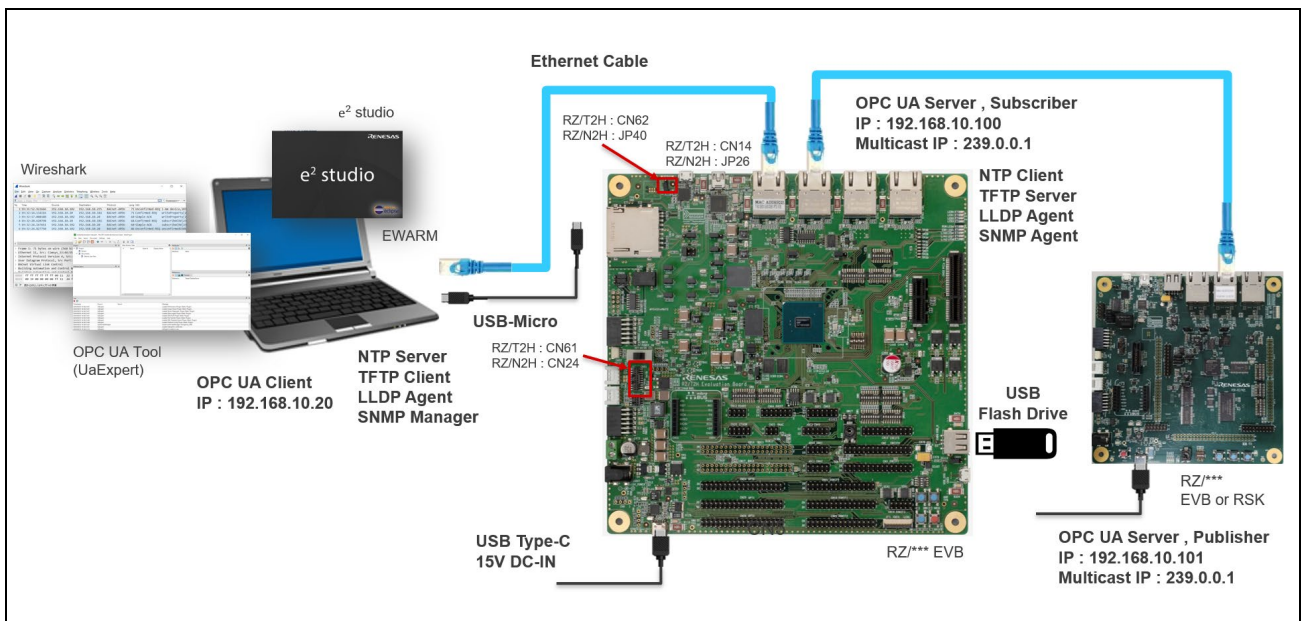


Figure 9.25. OPC UA Server Hardware Diagram for Confirming Subscriber (RZ/T2H, N2H)

OPC UA Subscriber 動作確認時の、それぞれのデバイスの役割は Figure 9.26 に示す通りです。

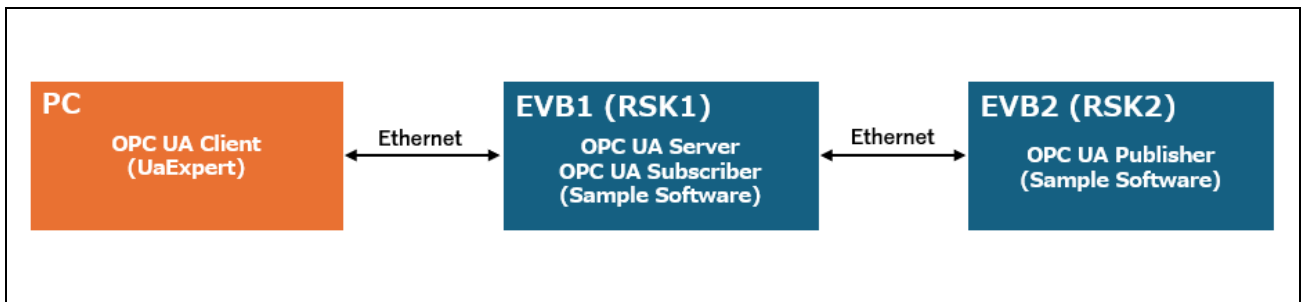


Figure 9.26. Connection Block Diagram for OPC UA Subscriber

・ 事前準備

本サンプルソフトを下記の設定でビルドし、EVB2(RSK2)に書き込みします。

- ・ #IPADR4 : 101
- ・ MAC address : 00:11:22:33:44:56

・ 動作確認

UaExpert と EVB2(RSK2)の OPC UA Server を接続します。

Address Space ウィンドウの Object ツリーに表示された Root>Objects>Server>PublishSubscribe>PublishedDataSets>AddPublishedDataItems Node を右クリックし、表示された”Call...”をクリックします。

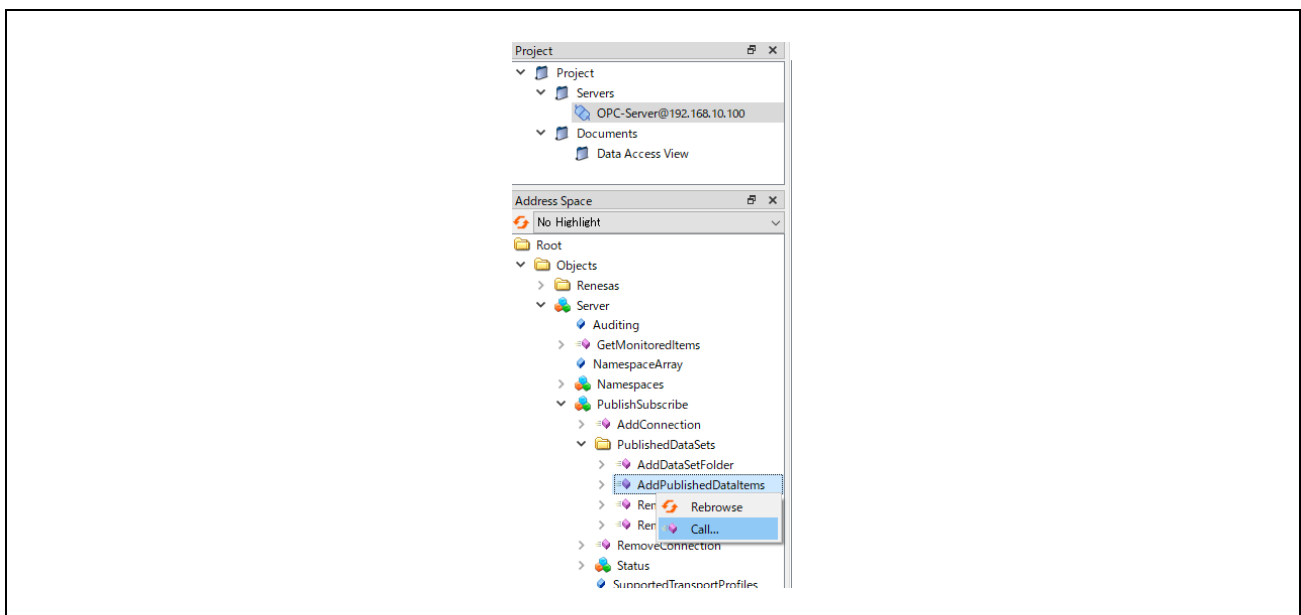


Figure 9.27. Address Space Window

表示されたウィンドウに、Publish したい Node の情報を入力し、Call をクリックします。Figure 9.28 は Publish する Node 情報として、Boolean(NS=2, ID=6049), Int32_1(NS=2, ID=6050), Int32_2(NS=2, ID=6051), DateTime(NS=2, ID=6052)の 4 つを設定しています。

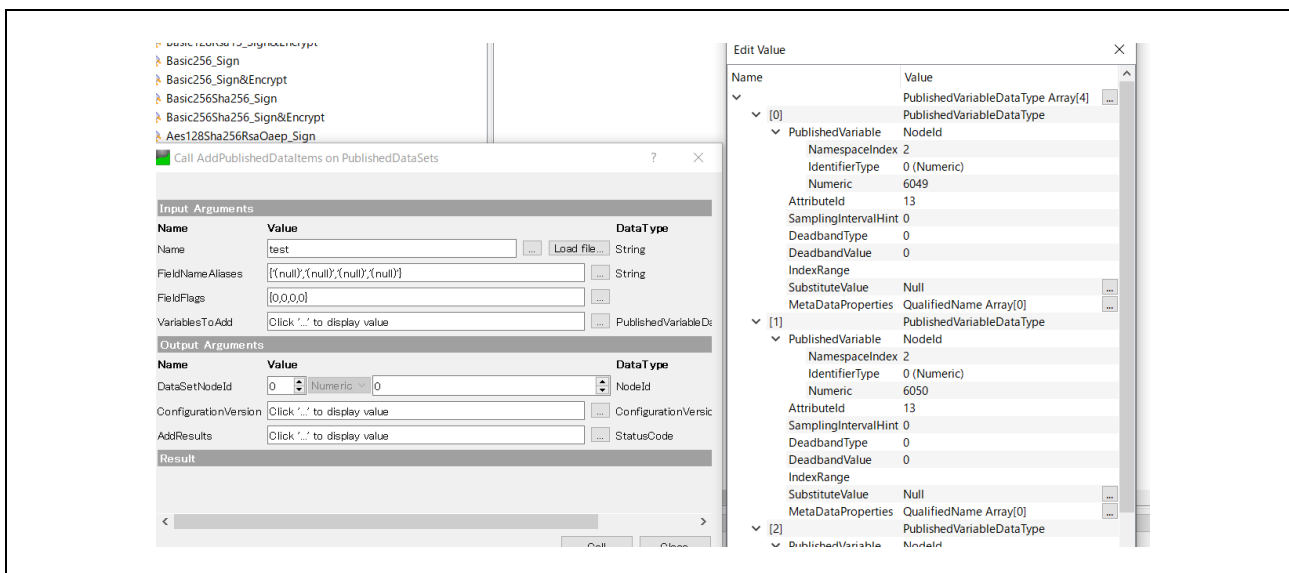


Figure 9.28. Call AddPublishedDataItems

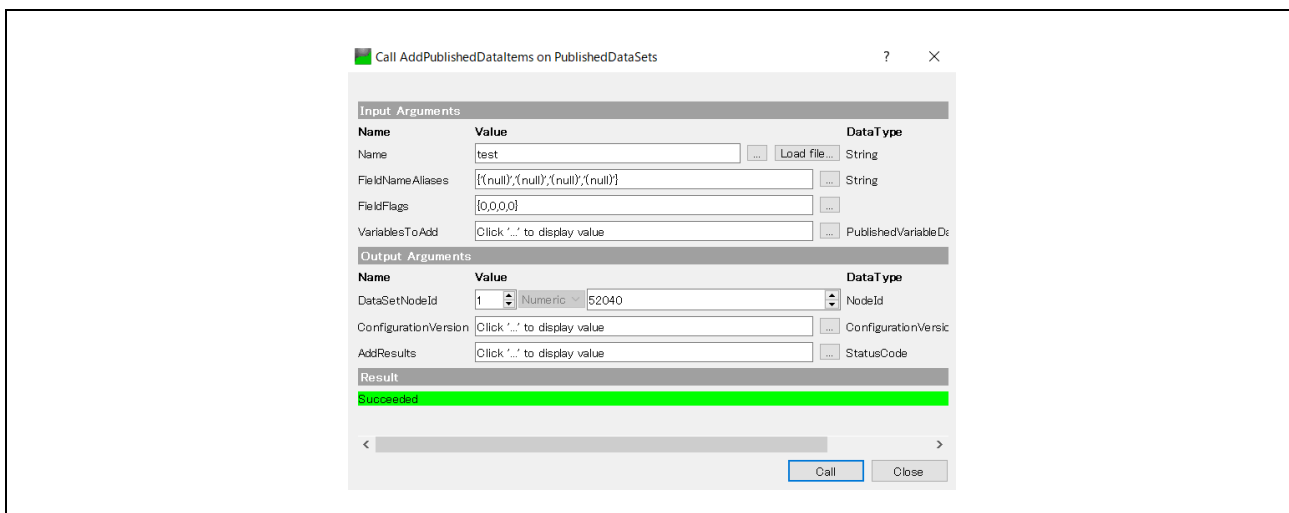


Figure 9.29. Call Result

パッケージに含まれている”pub_board2.exe”を、Linux 上で実行します。これにより、EVB2(RSK2)で pub が開始されます。

`./pub_board2.exe`

次に”sub.exe”を、Linux 上で実行します。これにより、EVB1(RSK1)で Sub が開始されます。

`./sub.exe`

実行が成功した後に、UaExpert の Address Space Window の”Root”を右クリックし、Rebrowse をクリックします。すると、Figure 9.31 のように、新しい Node が追加されていることが確認できます。これは、Subscribe した情報を格納するための Node です。

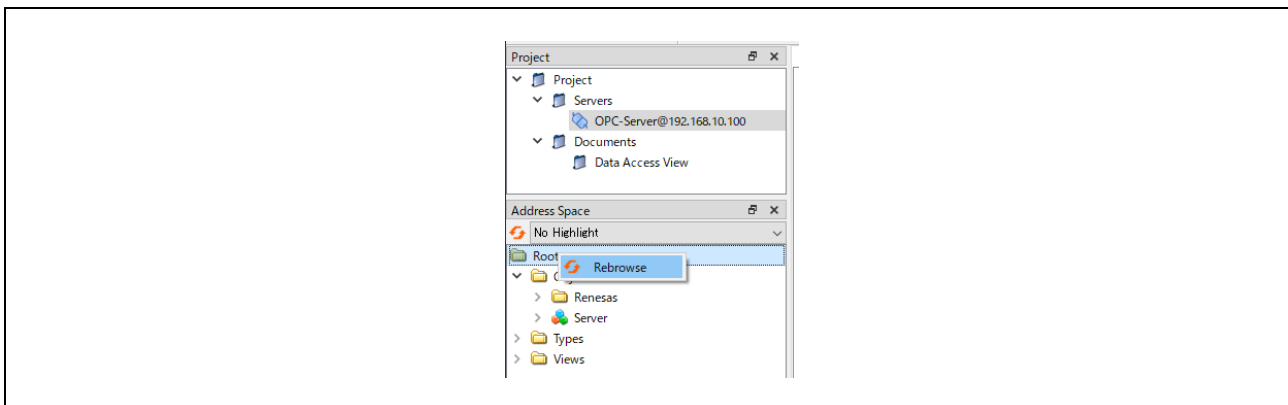


Figure 9.30. Rebrowse on UaExpert

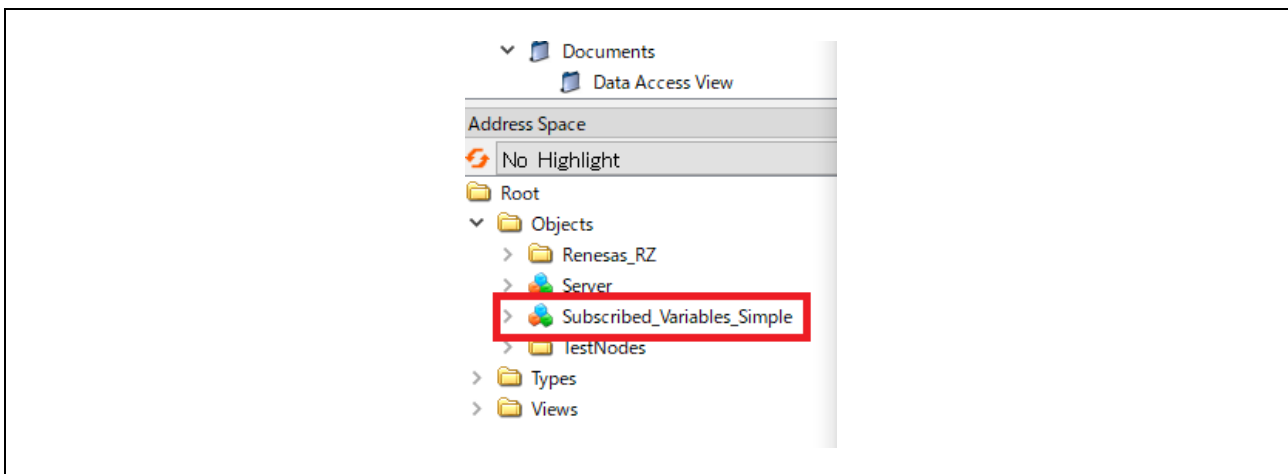


Figure 9.31. “Subscribed_Variables_Simple” Node

この Node を Data Access View で確認してみると、定期的に値が更新されており、正常に Subscribe できていることが確認できます。

Data Access View					
#	Server	Node Id	Display Name	Value	Dataty
1	SecurityNone@192.1...	NS1 Numeric 50000	Boolean	true	Boolean
2	SecurityNone@192.1...	NS1 Numeric 50003	DateTime	2025-09-19T08:13:36.596Z	DateTime
3	SecurityNone@192.1...	NS1 Numeric 50001	Int32_1	109	Int32
4	SecurityNone@192.1...	NS1 Numeric 50002	Int32_2	-2147483539	Int32

Figure 9.32. Check the added Node Values

9.4.3 OPC UA Client for PubSub Connection

本章では、pub.exe, sub.exe にて行っている PubSub 設定の変更方法を説明します。デフォルトの設定値は Table 9.2 に示す通りです。

Table 9.2. Configuration of PubSub

Executable file	Setting items	Setting item details	Setting locations in the source code	Set value
pub.exe, sub.exe	PubSubConnection	Network Address	UADP_NETWORK_ADDRESS_URL	opc.udp://239.0.0.1:4840/
		Publisher ID	WG_PUBLISHER_ID	100
pub.exe	WriterGroup	Publishing Interval	PUBLISHING_INTERVAL	1000 [msec]
		WriterGroup ID	WG_WRITER_GROUP_ID	1
	DataSetWriter	DataSet Writer ID	WG_DATA_SET_WRITER_ID	1
		PublisherDataSet Name	PDS_NAME	test
sub.exe	DataSetReader	WriterGroup ID	RG_WRITER_GROUP_ID	1
		DataSetWriter ID	RG_DATA_SET_WRITER_ID	1

- ・ Publisher の設定を変更する場合

addconnection.h の下記の部分を、任意の値に変更します。ここで、“PDS_NAME”だけは別途設定される PublisherDataSet Name と同じものにする必要があることに注意ください。9.4.1 章の例の場合、Figure 9.20 の Name に設定されている“test”と同じ名前にしてください。

```

8 //Publisher Setting
9 #define WG_PUBLISHER_ID      2234
10 #define PUBLISHING_INTERVAL  100 // ms
11 #define WG_WRITER_GROUP_ID  100
12 #define WG_DATA_SET_WRITER_ID 62541
13 #define PDS_NAME            "test" //Need to match the name of the PublisherDataSet.

```

Figure 9.33. Change Publisher Settings

Linux 上で、ソースコードがあるディレクトリにて、下記コマンドを実行します。

```
gcc -o pub.exe client_call_method_addconnection_pub.c open62541.c
```

- Subscriber の設定を変更する場合

addconnection.h の下記の部分を、Subscribe したい DataSet に合わせて変更します。

```
15 //Subscriber Setting
16 #define RG_PUBLISHER_ID      2234
17 #define RG_WRITER_GROUP_ID  101
18 #define RG_DATA_SET_WRITER_ID 62541
```

Figure 9.34. Change Subscriber Settings

Linux 上で、ソースコードがあるディレクトリにて、下記コマンドを実行します。

```
gcc -o sub.exe client_call_method_addconnection_sub.c open62541.c
```

9.5 Configuration for launching FSP Smart Configurator from EWARM

下記の手順に従って、設定してください。

- EWARM の” Tools -> Configure Tools...”をクリックします。
- New を選択し、それぞれ下記のように入力します。
 - Menu Text: FSP Smart Configurator
 - Command: \$RASC_EXE_PATH\$
 - Argument: --compiler IAR configuration.xml
 - Initial Directory: \$PROJ_DIR\$

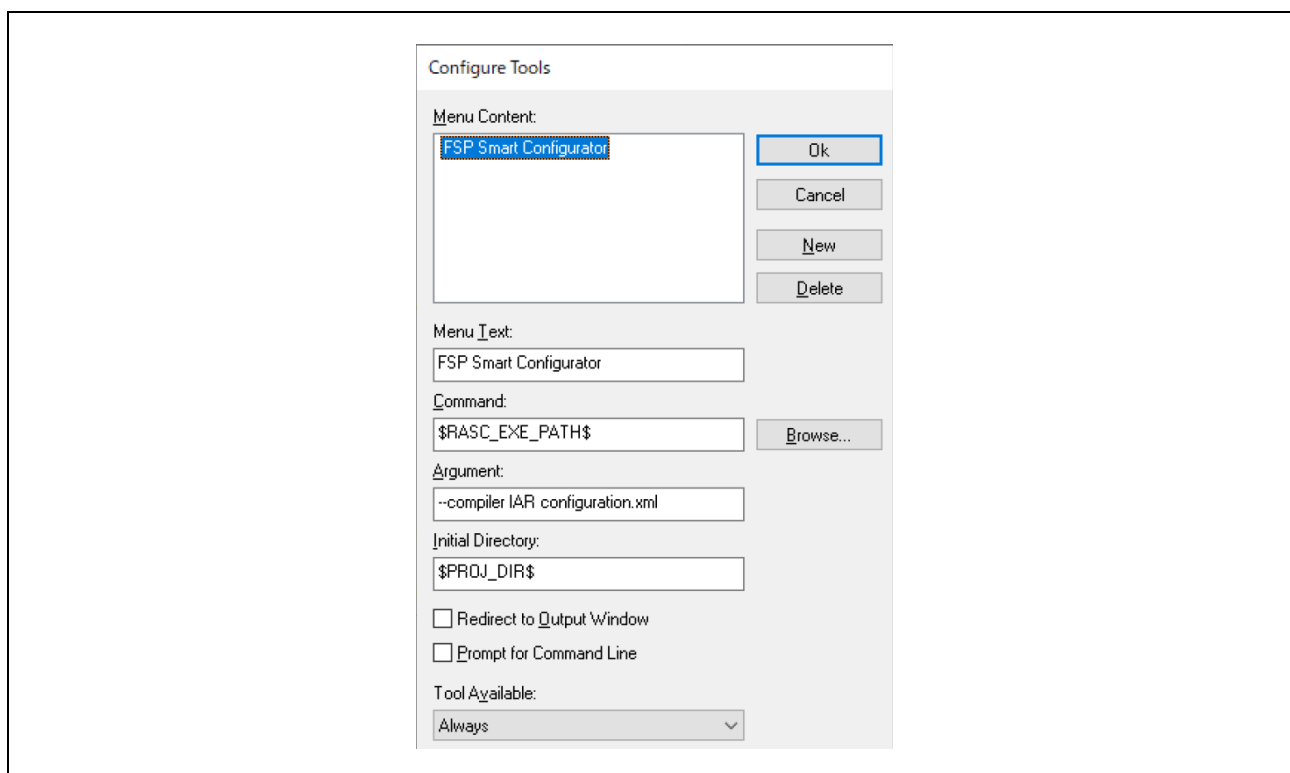


Figure 9.35. Settings “Configure Tools”

- buildinfo.ipcf ファイルを開き、“RASC_EXE_PATH”に FSP Smart Configurator のパスを入力します。

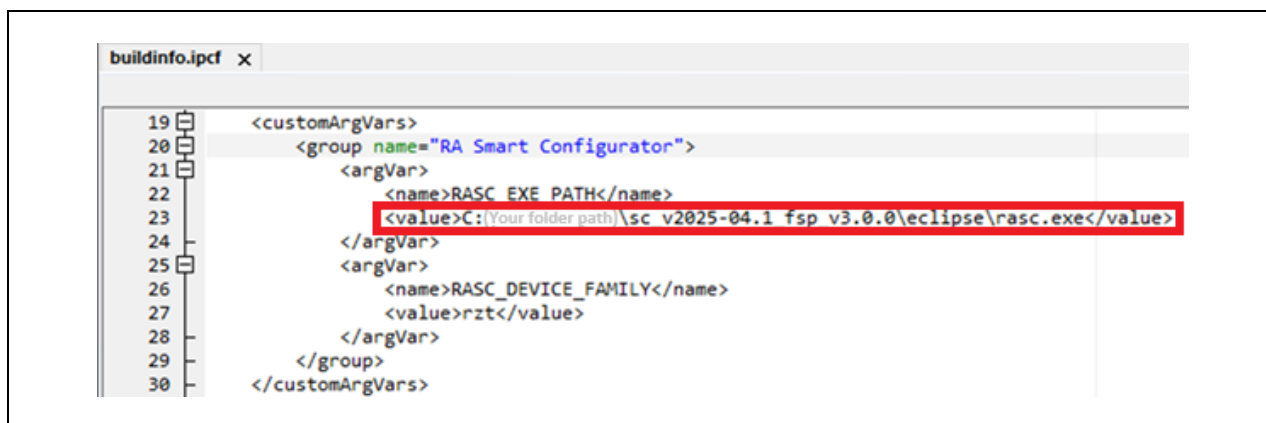


Figure 9.36. buildinfo.ipcf

9.6 How to Change Various Settings in the Sample Software

サンプルソフトウェアの各種設定をデフォルト値から変更する手順を下記に示します。

9.6.1 When using e² studio

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。ただし、Dual core 構成の場合、セカンダリプロジェクトを選択してください。

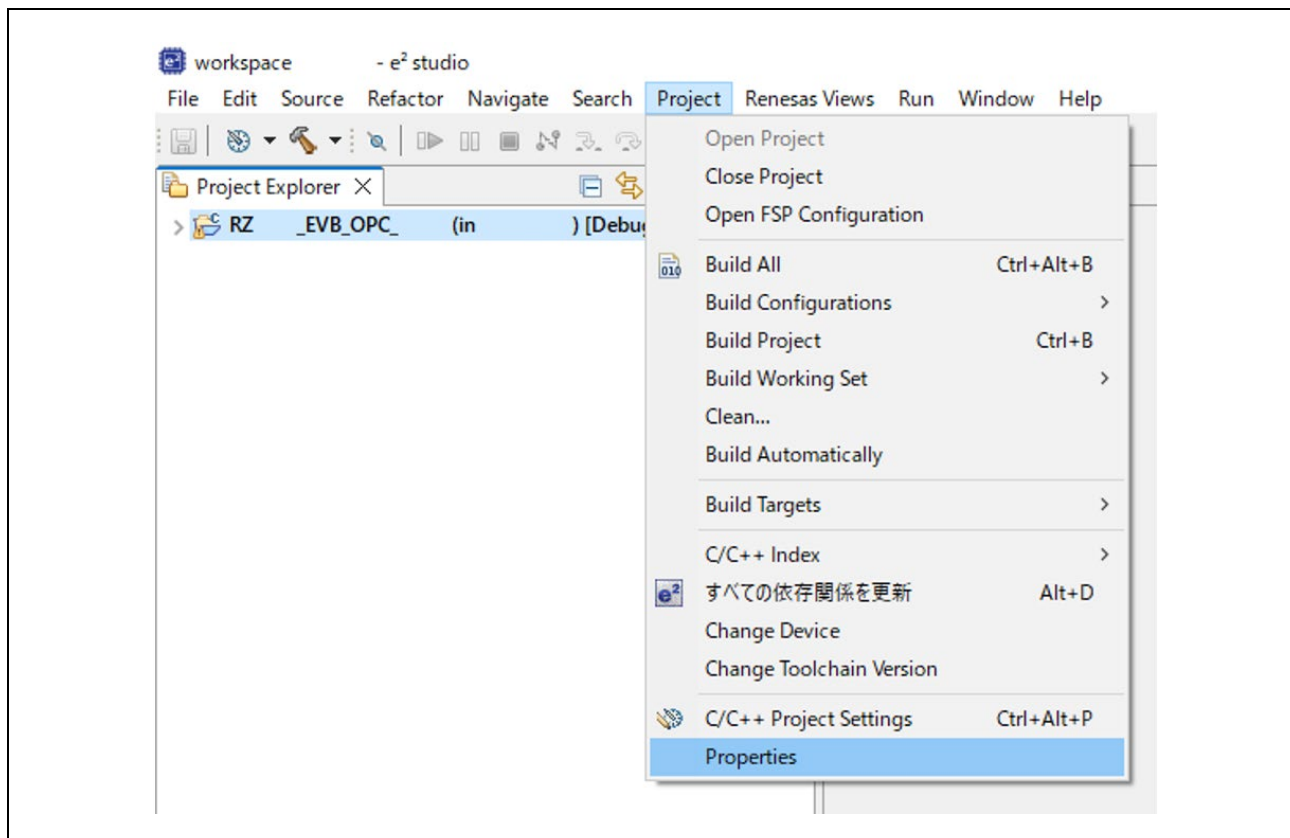


Figure 9.37. Open project properties

C/C++General > Paths and Symbols の#Symbols タグから Languages の GNU C を選択します。

ここで各種機能の有効化/無効化の切り替えや、設定を行うことができます。詳細は Table 9.3 の通りです。

Table 9.3. Symbol Settings

Symbol Name	Description	Parameter	Initial Value
IPADR1, 2, 3, 4	IP アドレス定義	任意の IP アドレス	192, 168, 10, 100
IPADR4_NTPServer	接続する NTP サーバの IP アドレスのうち、下位 8bit ※上位 24bit は IPADR1, 2, 3 と兼用します。	NTP サーバの IP アドレス	20
ENABLE_PUBSUB_OVER_MQTT	PubSub over MQTT 機能の有効化 ※現在は 0:無効 のみサポートしています。	0:無効 1:有効	0
ENABLE_PUBSUB_OVER_UADP	PubSub over UADP 機能の有効化	0:無効 1:有効	1
TRUSTLIST_SIZE_MAX	TrustList の最大数 ※接続する OPC UA Client の数に応じて変更してください。	任意の数値	32
REVOCACTIONLIST_SIZE_MAX	RevocationList の最大数 ※CRL ファイルの保存数に応じて変更してください。	任意の数値	8
ISSUERLIST_SIZE_MAX	登録する IssuerList の最大数 ※Issuer ファイルの保存数に応じて変更してください。	任意の数値	8
ENABLE_FREERTOS_PLUS_FAT	FreeRTOS+FAT, TFTP 機能の有効化 ※無効化した場合、ソースコード上 (keyfiles.h) に埋め込みされた鍵ファイルが、セキュリティ機能に使用されるようになります。	0:無効 1:有効	1

IP アドレスを変更する場合、ここで#IPADR1, 2, 3, 4 の値を変更してください。

各種設定を変更後、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”Rebuild Index”をクリックします。

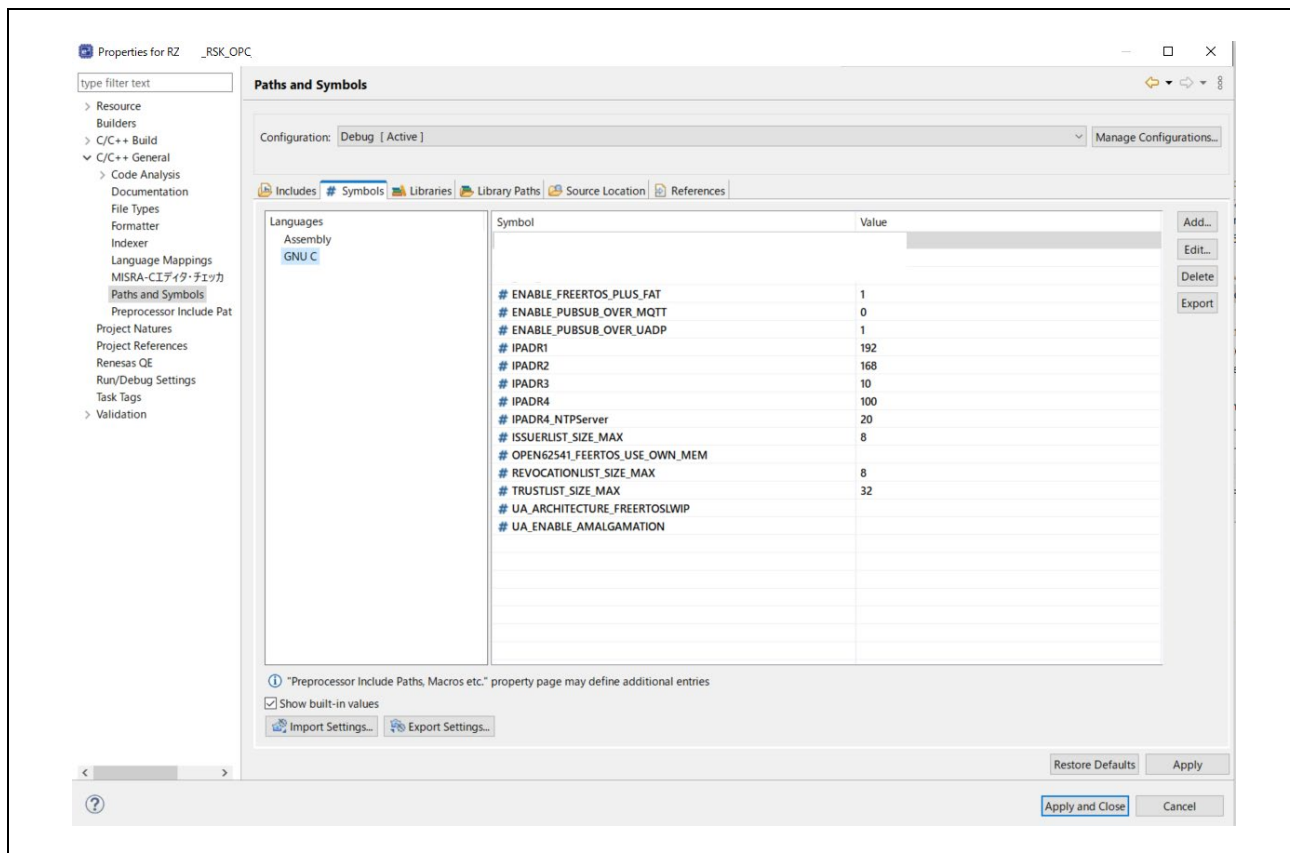


Figure 9.38. Change Build Options

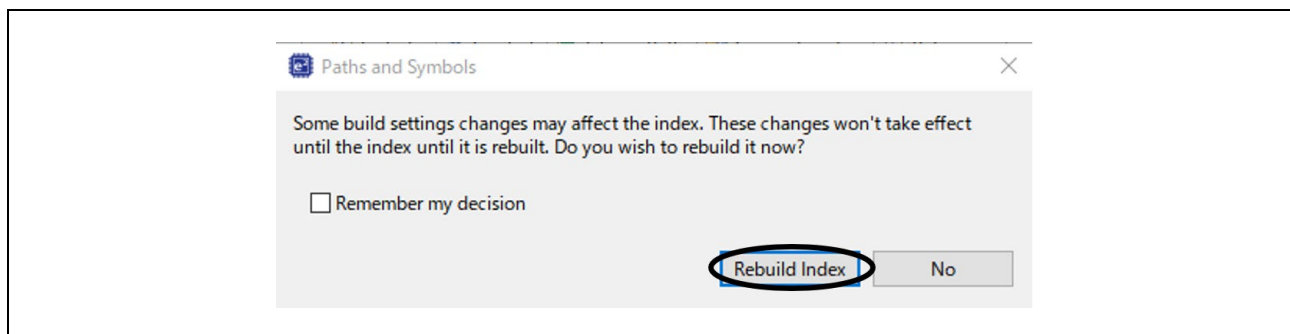


Figure 9.39. Click “Rebuild index”

MAC アドレスを変更する場合、下記の手順で変更ください。

Figure 9.40 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。ただし、Dual core 構成の場合、セカンダリプロジェクトの configuration.xml を選択してください。

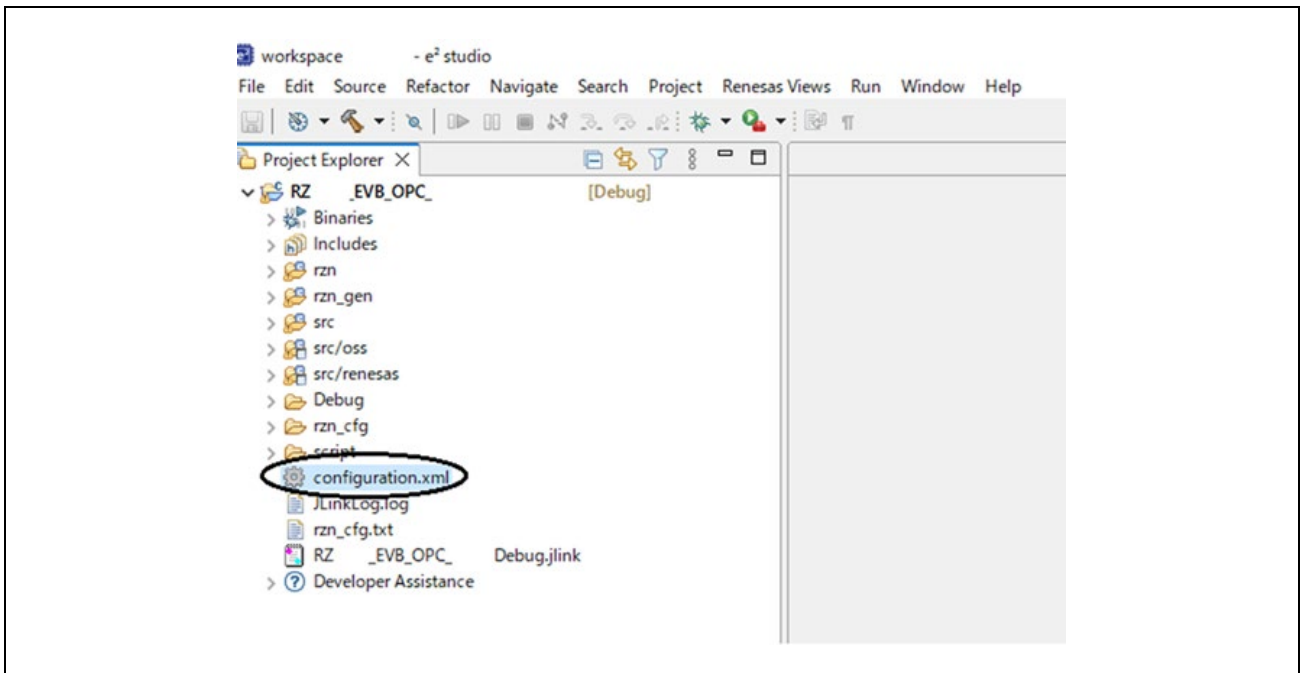


Figure 9.40. Double click configuration.xml

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

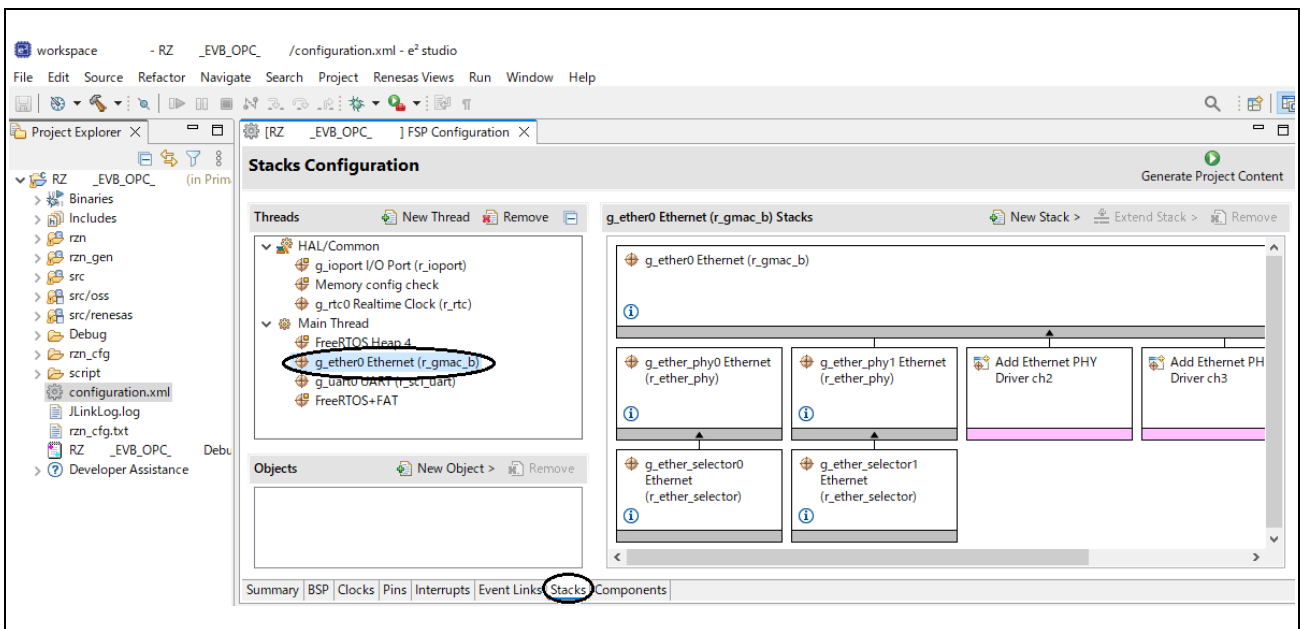


Figure 9.41. Click g_ether0

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

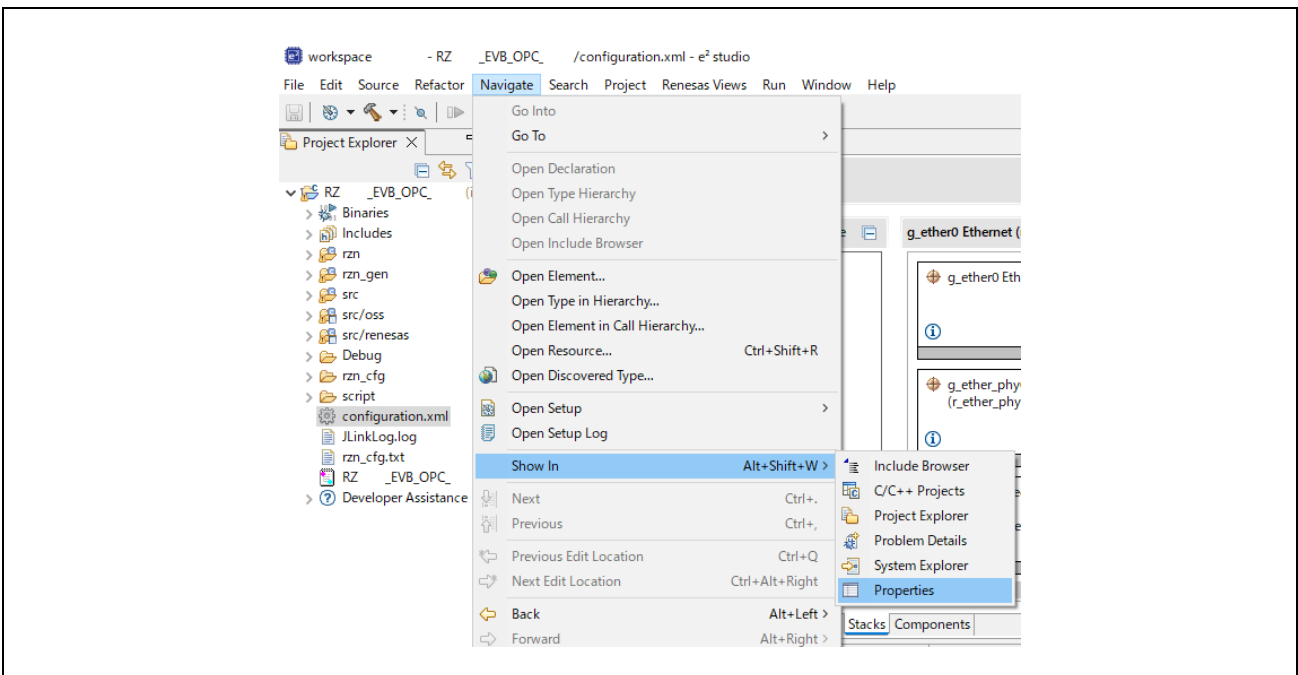


Figure 9.42. Click Properties

Properties タブを開き、General>MAC address 0(例 : 74:90:50:10:F9:ED)を入力します。
 入力後 Generate Project Content をクリックします。

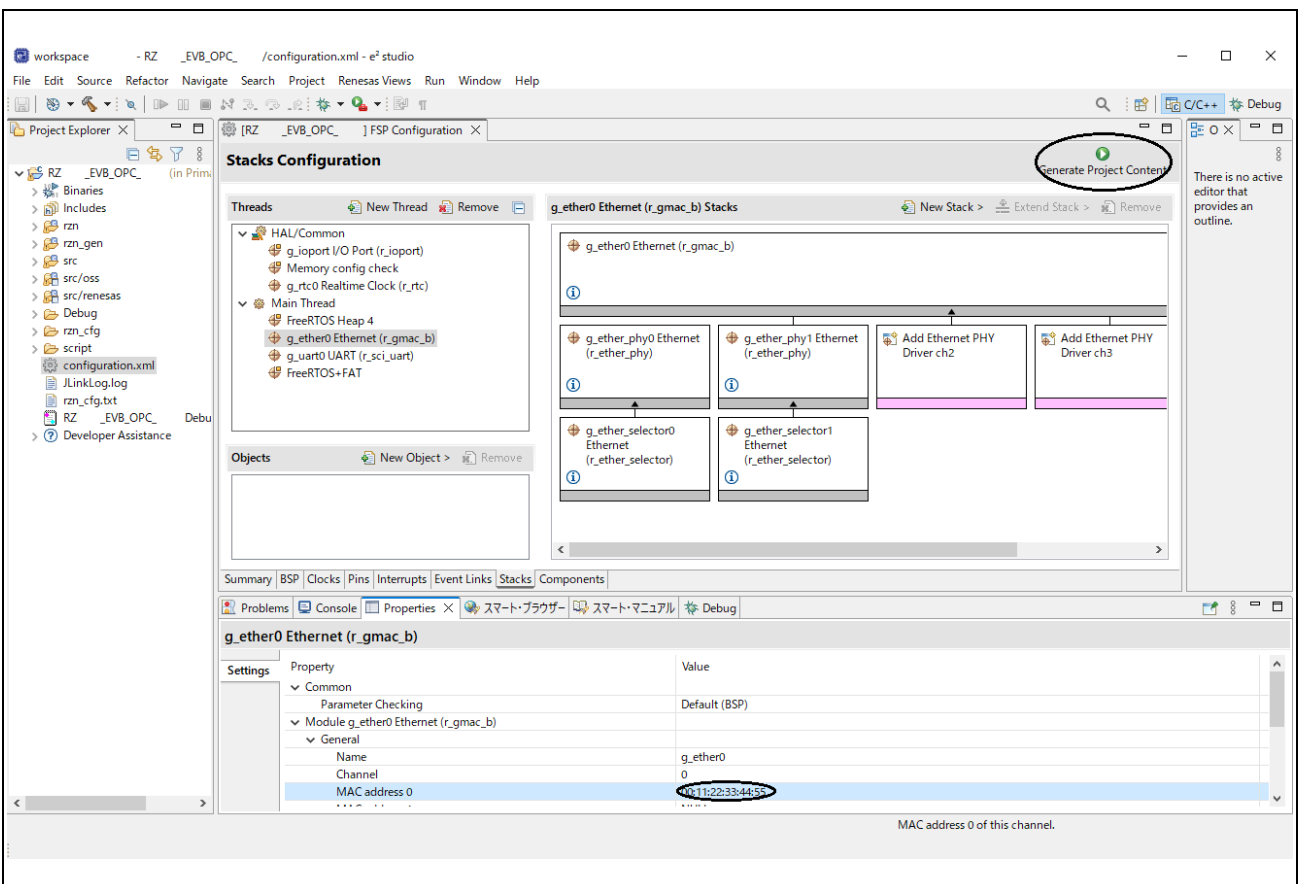


Figure 9.43. Enter MAC address

また、PubSub 機能を使用する場合、opc_pubsub_uadp.h の定義を変更してください。

Table 9.4. opc_pubsub_uadp.h defines

Defines	Description	Parameter	Initial Value
UADP_LAYOUT	PubSub のサンプル設定のレイアウト選択	UADP_FIXED_LAYOUT: Periodic Fixed Layout UADP_DYNAMIC_LAYOUT: Dynamic Layout	UADP_FIXED_LAYOUT
ENABLE_PUBSUB_EXAMPLE_SETTING	PubSub のサンプル設定の有効化	PUBSUB_EXAMPLE_DISABLE: 無効 PUBSUB_EXAMPLE_ENABLE: 有効、暗号化無し PUBSUB_EXAMPLE_ENABLE_AES128: 有効、暗号化あり(AES128-CTR) PUBSUB_EXAMPLE_ENABLE_AES256: 有効、暗号化あり(AES256-CTR)	PUBSUB_EXAMPLE_DISABLE

```

1  #define UADP_FIXED_LAYOUT 1
2  #define UADP_DYNAMIC_LAYOUT 2
3  #define UADP_LAYOUT UADP_FIXED_LAYOUT /* Select Fixed or Dynamic */
4
5  #define PUBSUB_EXAMPLE_DISABLE 0 /* Diable pubsub example config */
6  #define PUBSUB_EXAMPLE_ENABLE 1 /* Enable pubusub example config without security */
7  #define PUBSUB_EXAMPLE_ENABLE_AES128 2 /* Enable pubusub example config with security (PubSub-Aes128-CTR) */
8  #define PUBSUB_EXAMPLE_ENABLE_AES256 3 /* Enable pubusub example config with security (PubSub-Aes256-CTR) */
9
10 #define ENABLE_PUBSUB_EXAMPLE_SETTING PUBSUB_EXAMPLE_DISABLE /* Select 0-3 */

```

Figure 9.44. opc_pubsub_uadp.h

9.6.2 When using EWARM

Workspace 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 Options... を開きます。ただし、Dual core 構成の場合、セカンダリプロジェクトから設定してください。

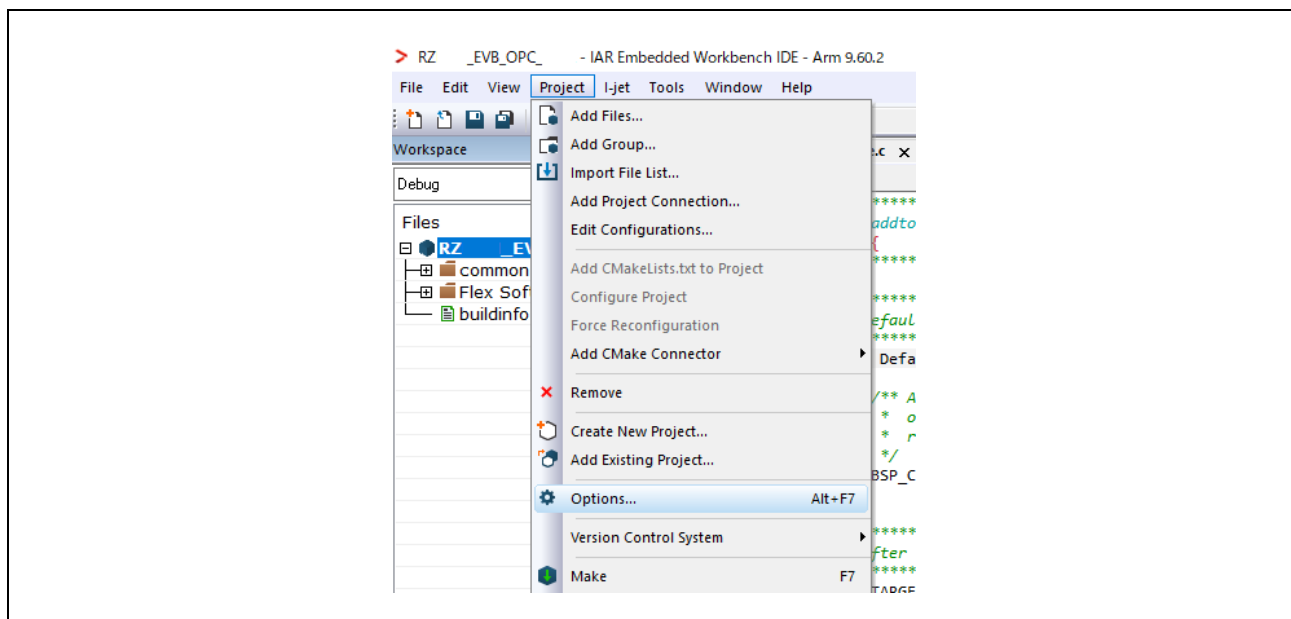


Figure 9.45. Open Options

C/C++ Compiler > Preprocessor の Defined symbols で各種機能の有効化/無効化の切り替えや、設定を行うことができます。詳細は Table 9.5 の通りです。

Table 9.5. Symbol Settings

Symbol Name	Description	Parameter	Initial Value
IPADR1, 2, 3, 4	IP アドレス定義	任意の IP アドレス	192, 168, 10, 100
IPADR4_NTPServer	接続する NTP サーバの IP アドレスのうち、下位 8bit ※上位 24bit は IPADR1, 2, 3 と兼用します。	NTP サーバの IP アドレス	20
ENABLE_PUBSUB_OVER_MQTT	PubSub over MQTT 機能の有効化 ※現在は 0:無効 のみサポートしています。	0:無効 1:有効	0
ENABLE_PUBSUB_OVER_UADP	PubSub over UADP 機能の有効化	0:無効 1:有効	1
TRUSTLIST_SIZE_MAX	TrustList の最大数 ※接続する OPC UA Client の数に応じて変更してください。	任意の数値	32
REVOCACTIONLIST_SIZE_MAX	RevocationList の最大数 ※CRL ファイルの保存数に応じて変更してください。	任意の数値	8
ISSUERLIST_SIZE_MAX	登録する IssuerList の最大数 ※Issuer ファイルの保存数に応じて変更してください。	任意の数値	8
ENABLE_FREERTOS_PLUS_FAT	FreeRTOS+FAT, TFTP 機能の有効化 ※無効化した場合、ソースコード上 (keyfiles.h) に埋め込みされた鍵ファイルが、セキュリティ機能に使用されるようになります。	0:無効 1:有効	1

IP アドレスを変更する場合、ここで #IPADR1, 2, 3, 4 の値を変更してください。

各種設定を変更後、OK をクリックします。

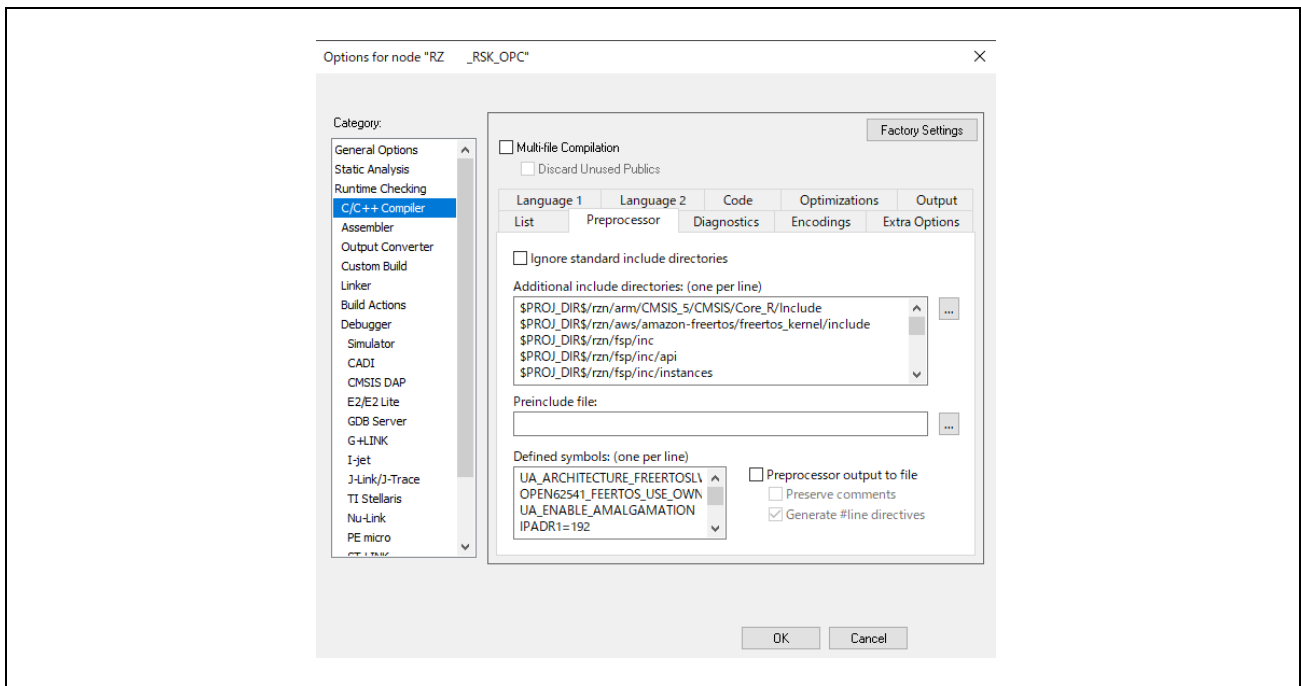


Figure 9.46. Change Build Options

MAC アドレスを変更する場合、ここで、下記の手順で変更ください。

FSP Smart Configurator で Stacks タブを開き、g_ether0 Ethernet をクリックして選択します。

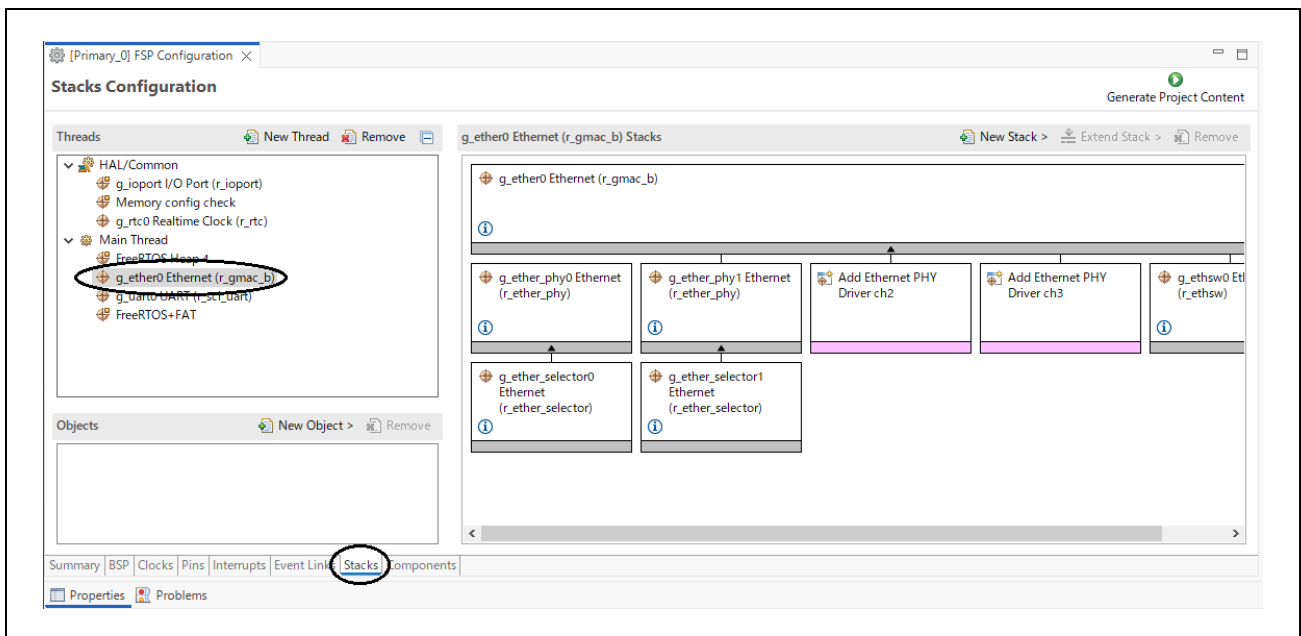


Figure 9.47. Click g_ether0

Properties タブを開き、General>MAC address 0 を入力します。

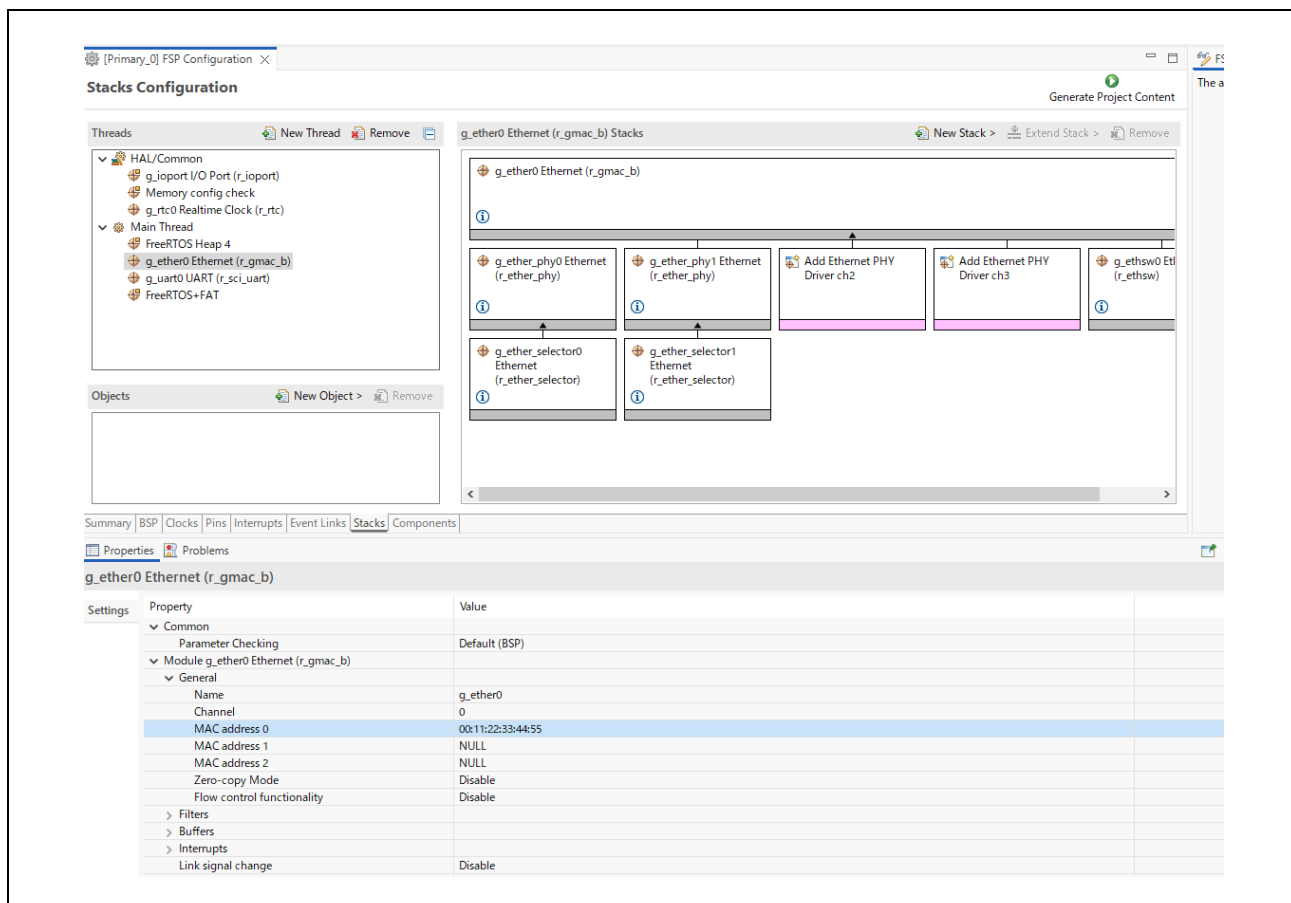


Figure 9.48. Enter MAC address

MAC アドレスの設定が完了した後に、“Generate Project Content”ボタンをクリックするとコードが生成されます。

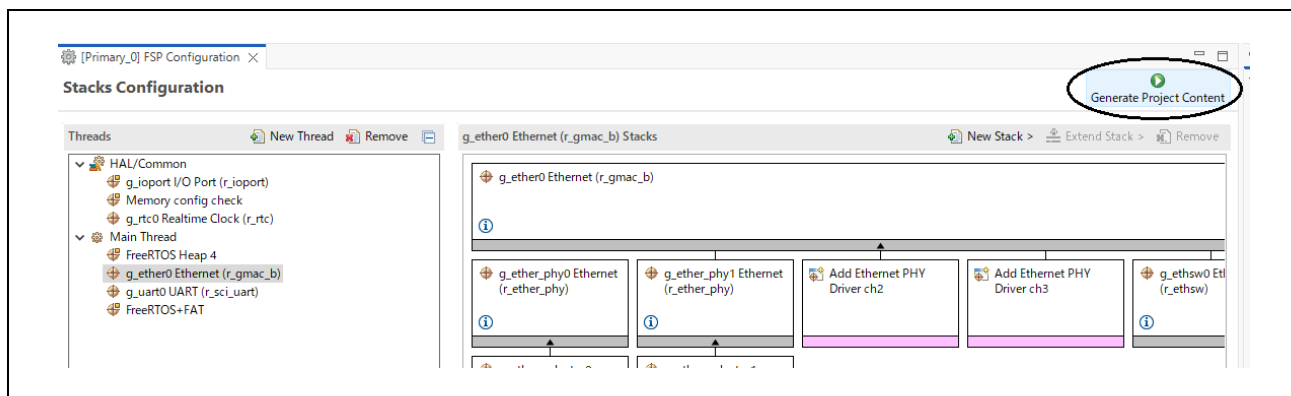


Figure 9.49. Generate Project Content

また、PubSub 機能を使用する場合、opc_pubsub_uadp.h の定義を変更してください。

Table 9.6. opc_pubsub_uadp.h defines

Defines	Description	Parameter	Initial Value
UADP_LAYOUT	PubSub のサンプル設定のレイアウト選択	UADP_FIXED_LAYOUT: Periodic Fixed Layout UADP_DYNAMIC_LAYOUT: Dynamic Layout	UADP_FIXED_LAYOUT
ENABLE_PUBSUB_EXAMPLE_SETTING	PubSub のサンプル設定の有効化	PUBSUB_EXAMPLE_DISABLE: 無効 PUBSUB_EXAMPLE_ENABLE: 有効、暗号化無し PUBSUB_EXAMPLE_ENABLE_AES128: 有効、暗号化あり(AES128-CTR) PUBSUB_EXAMPLE_ENABLE_AES256: 有効、暗号化あり(AES256-CTR)	PUBSUB_EXAMPLE_DISABLE

```

5
6 #define UADP_FIXED_LAYOUT 1
7 #define UADP_DYNAMIC_LAYOUT 2
8 #define UADP_LAYOUT UADP_FIXED_LAYOUT /* Select Fixed or Dynamic */
9
10 #define PUBSUB_EXAMPLE_DISABLE 0 /* Diable pubsub example config */
11 #define PUBSUB_EXAMPLE_ENABLE 1 /* Enable pubusub example config without security */
12 #define PUBSUB_EXAMPLE_ENABLE_AES128 2 /* Enable pubusub example config with security (PubSub-Aes128-CTR) */
13 #define PUBSUB_EXAMPLE_ENABLE_AES256 3 /* Enable pubusub example config with security (PubSub-Aes256-CTR) */
14
15 #define ENABLE_PUBSUB_EXAMPLE_SETTING PUBSUB_EXAMPLE_DISABLE /* Select 0-3 */
16

```

Figure 9.50. opc_pubsub_uadp.h

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2026/04/15	-	RZ/T2M, N2L 向けアプリケーションノート (R01AN7048xxxxxx)と RZ/T2H, N2H 向けアプリケーションノート(R01AN7755xxxxxx)を統合し、新規作成
		-	RZ FSPv4.0.0 に対応
		p.80	8章 Software Specifications を新規作成

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

- Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
- Ethernet is a registered trademark of Fuji Xerox Co. Ltd.
- J-Link is a trademark of SEGGER Microcontroller GmbH.
- IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc
- Additionally all product names and service names in this document are a trademark or a registered trademark which belongs to the respective owners. a trademark or a registered trademark which belongs to the respective OWNERS.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
 - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

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

Contact information

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

www.renesas.com/contact/.