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。

RZ/T2M と RZ/T2ME は互換性があるため、本書ではこれらのモデル名を同義語とみなしてください。

Target Device

RZ/T シリーズ: RZ/T2M (RZ/T2ME), 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 File System 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 e² studio environment 34
5.4 UaExpert 35
5.5 Wireshark 35
5.6 OpcCmd 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 e² 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

9.2 TFTP 123
9.2.1 File Writing 123
9.2.2 File Reading 123
9.3 LLDP, SNMP MIB 124
9.4 When configuring OPC UA PubSub settings using the Method 125
9.4.1 Publisher 125
9.4.2 Subscriber 129
9.5 Configuration for launching FSP Smart Configurator from EWARM 132
9.6 How to Change Various Settings in the Sample Software 133
9.6.1 Settings before building 133
9.6.2 Settings after building 144
9.7 Notes when using newer FSP version 145

Revision History 146

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幅広い産業分野で利用されています。

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	BSP	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	SKS	Security Key Service
18	SNMP	Simple Network Management Protocol
19	SNTP	Simple Network Time Protocol
20	TFTP	Trivial File Transfer Protocol
21	TLV	Type Length Value
22	UACTT	OPC UA Compliance Test Tool
23	UADP	UA Datagram Protocol
24	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。本プロジェクトは、サンプルソフト v3.0.0 ベースで作成した RZ/T2M, RZ/T2H, RZ/N2L 向けソフトウェアにより Table 2.2 に示す内容で認証を取得しています。本サンプルソフトはそれらのソフトから機能を拡充したもので、UACTT (v1.04.11-01.00.510)を使用して動作確認済です。

Table 2.1. OPC UA Server Specifications

Items	
Supported Profiles and Facets	
Application Profile	Embedded 2017 UA Server [UACore 1.04]
Additional Facets	Method Server Facet
Security Policies	None Basic256 (Deprecated) Basic128Rsa15 (Deprecated) Basic256Sha256 Aes128-Sha256-RsaOaep
User Tokens	Anonymous Server Facet User Name Password Server Facet X509 Certificate Server Facet
Server Capabilities	
Max sessions	2
Max subscriptions per session	4
Max monitored items per subscription	100
Min publishing interval	500 ms
Min sampling interval	250 ms

Table 2.2. OPC UA Server Previously Certified Specifications

Application Profile	Micro Embedded Device 2017 Server [UACore 1.04]
Additional Facets	Method Server Facet
Security Policies	None Basic256 Basic128Rsa15 Basic256Sha256 Aes128-Sha256-RsaOaep
User Tokens	Anonymous Server Facet User Name Password Server Facet

本サンプルソフトに設定されている Username と Password のペアは Table 2.3 に示す通りです。

Table 2.3. Username/Password

	Username	Password
1	user1	password
2	user2	password1

2.1.2 OPC UA PubSub

本サンプルソフトの OPC UA PubSub 仕様は Table 2.4 の通りです。SKS との通信は現時点で未対応です。

Table 2.4. 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 2.5 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 2.5. 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 File System

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を書き込むことができます。書き込み方法は 9.2 章を参照ください。

各ファイルの格納先フォルダ名は Table 2.6 に示す通りです。コンフィグファイルを除き、ファイル名に指定はありません。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。動作確認の際はあらかじめこれらのファイルを書き込みした USB フラッシュドライブを評価ボードに接続してください。

Table 2.6. USB flash drive folder structure

Item	Description	Attached to Package
/Root		
├── pki/		
│ ├── application/	Application authentication PKI directory	
│ │ ├── own/		
│ │ │ ├── certs/	Server certificate file (der format)	✓
│ │ │ └── private/	Server private key file (der format)	✓
│ │ ├── trusted/		
│ │ │ ├── certs/	Trusted client certificate files (der format)	✓
│ │ │ └── crl/	CRLs for trusted client certificates (der format)	-
│ │ └── issuers/		
│ │ ├── certs/	Client certificate issuer files (der format)	-
│ │ └── crl/	CRLs for client certificate issuers (der format)	-
│ └── user/	User authentication PKI directory	
│ ├── trusted/		
│ │ ├── certs/	User trusted certificate files (der format)	✓
│ │ └── crl/	CRLs for user trusted certificates (der format)	✓
│ └── issuers/		
│ ├── certs/	User certificate issuer files (der format)	-
│ └── crl/	CRLs for user certificate issuers (der format)	-
│		
├── pubsub/		
│ ├── keys/	PubSub symmetric key files	
│ │ ├── aes128/	AES-128 key file (bin format)	✓
│ │ └── aes256/	AES-256 key file (bin format)	✓
│ └── settings/pubsub.cfg	PubSub configuration files	✓
│		
└── system/ip.cfg	Network configuration file	✓

✓はパッケージに添付、-は非添付

TrustList, CRL, IssuerList にはそれぞれ読み出し可能な最大ファイル数が定義されています。変更する場合は opc_load_files.h に定義された下記のマクロを変更してください。

Table 2.7. Macro definitions

Item	Definition	Default value
TrustList	TRUSTLIST_SIZE_MAX	32
CRL	REVOCATIONLIST_SIZE_MAX	8
IssuerList	ISSUERLIST_SIZE_MAX	8

コンフィグファイル（pubsub.cfg、ip.cfg）は、ファイルの内容を編集することで PubSub 設定と IP アドレス設定を変更できます。詳細については、9.6.2 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動作することも可能です。その場合は、9.6.1 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 2.8. RZ/T2M Folder structure

Item	Description
RZT2M_OPCUA_RSK_rev0420	
├── common	Common files for all projects
└── project	Projects
└── CR52_dual	Dual core project
├── e2studio	e² 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 2.9. RZ/N2L Folder structure

Item	Description
RZN2L_OPCUA_RSK_rev0420	
├── common	Common files for all projects
└── project	Projects
└── CR52_single	Single core project
├── e2studio	e² 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 2.10. RZ/T2H Folder structure

Item	Description
RZT2H_OPCUA_EVB_rev0420	
├── common	Common files for all projects
└── project	Projects
├── CA55_dual	Dual core project
│ ├── e2studio	e² 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	e² 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 2.11. RZ/N2H Folder structure

Item	Description
RZN2H_OPCUA_EVB_rev0420	
├── 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 2.12. Common folder structure

Item	Description
common	Common files for all projects
├── oss	Open Source Software
│ ├── mbedtls	Mbed TLS
│ ├── open62541	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 RSK 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
	e² studio	2025-12	RZ FSP GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Configurator	FSP Smart Configurator	2025-12	
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 RSK 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
	e² studio	2025-12	RZ FSP GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Configurator	FSP Smart Configurator	2025-12	
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
	e² studio	2026-04.1	RZ FSP GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Configurator	FSP Smart Configurator	2026-04.1	
Flexible Software Package	FSP	4.1.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.1.arm-13-24	
	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
	e² studio	2026-04.1	RZ FSP GitHub - renesas/rz-fsp: Flexible Software Package (FSP) for Renesas RZ series · GitHub
Configurator	FSP Smart Configurator	2026-04.1	
Flexible Software Package	FSP	4.1.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.1.arm-13-24	
	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	2.0.1	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	
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

本書では、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

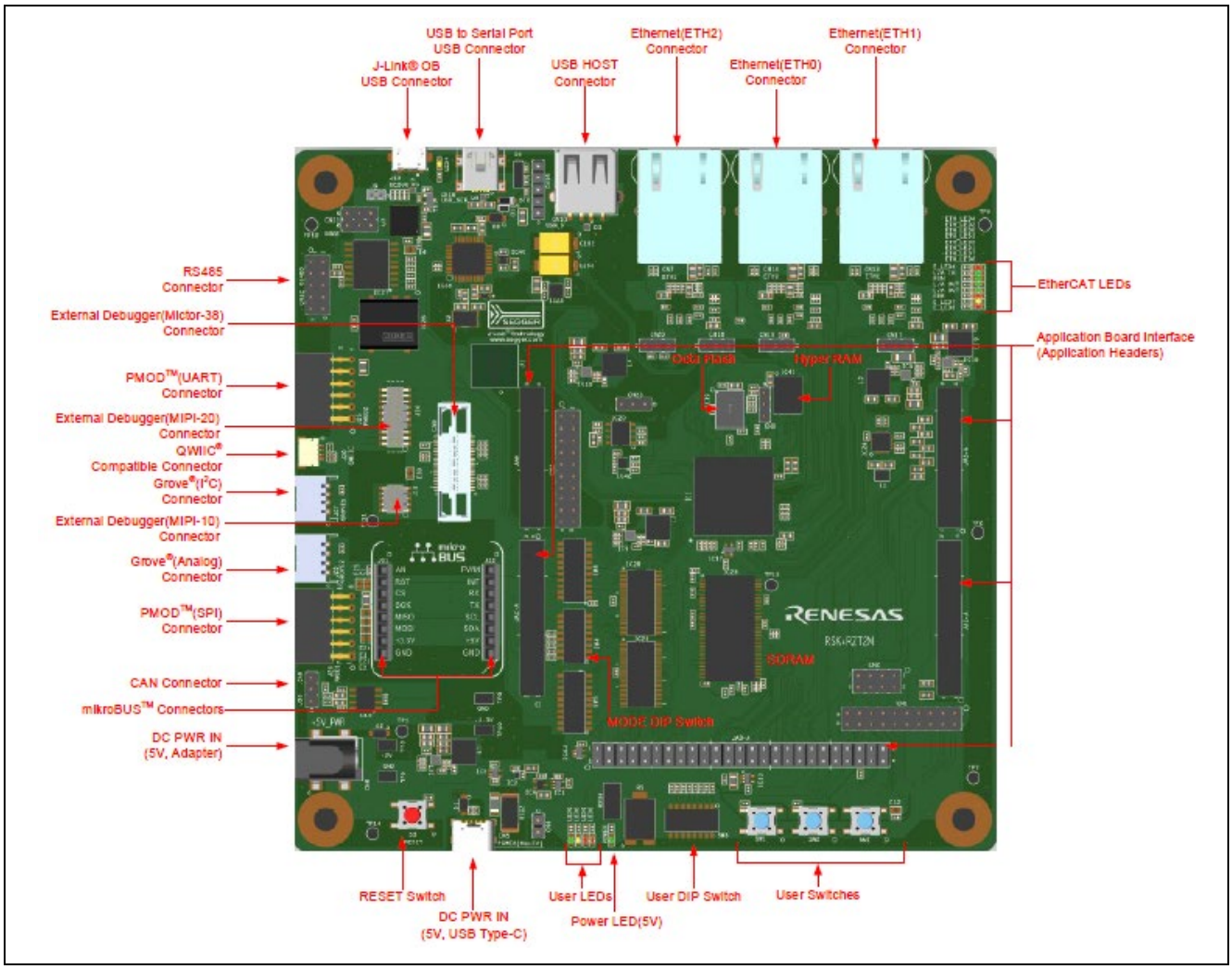


Figure 4.1. RZT2M RSK board layout

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 Settings

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 Settings

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。

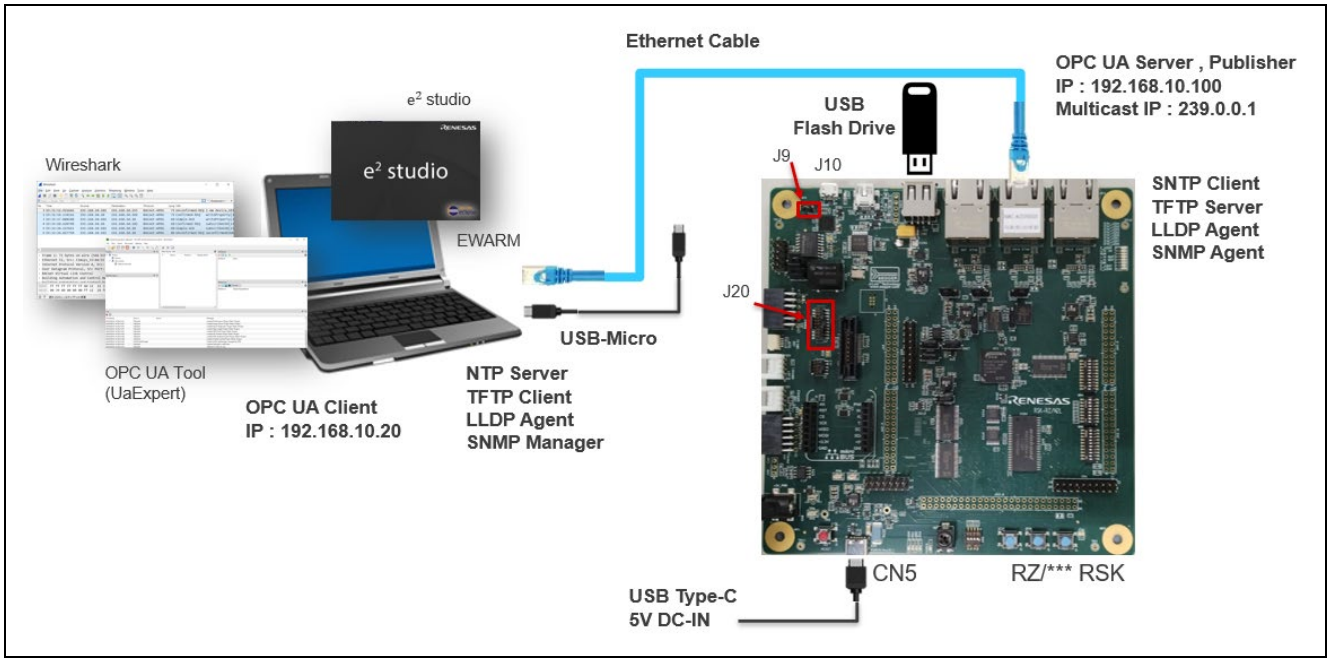


Figure 4.2. RZ/T2M RSK board OPC UA Server hardware connection diagram

4.2 RZ/N2L RSK Board

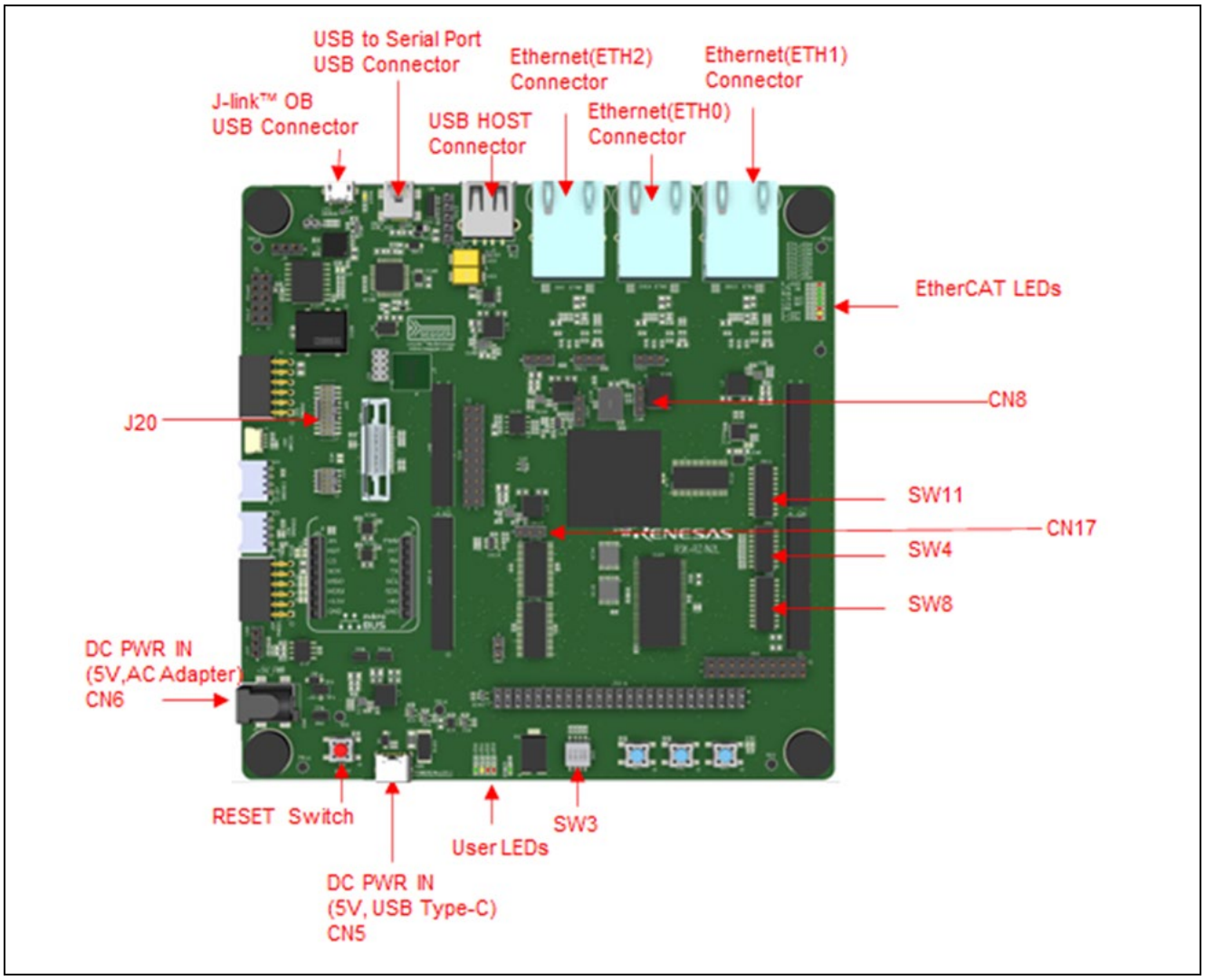


Figure 4.3. RZN2L RSK board layout

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 Settings

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 Settings

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。

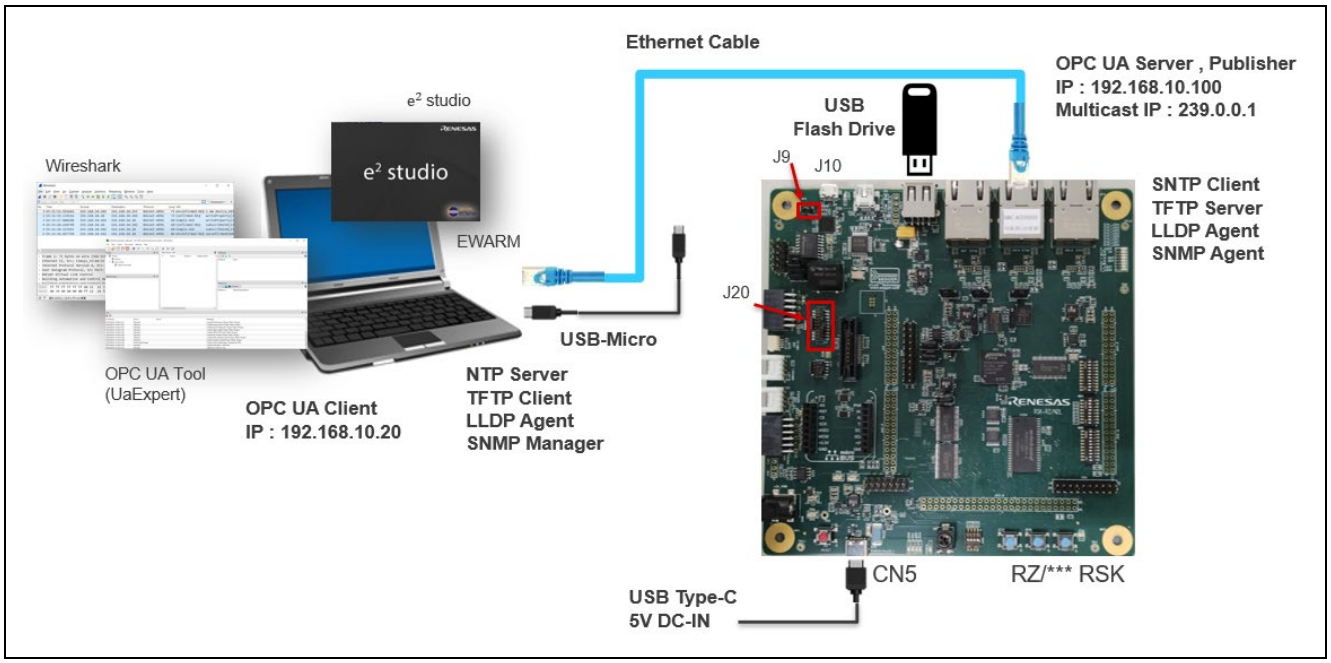


Figure 4.4. RZ/N2L RSK board OPC UA Server hardware connection diagram

4.3 RZ/T2H Evaluation Board

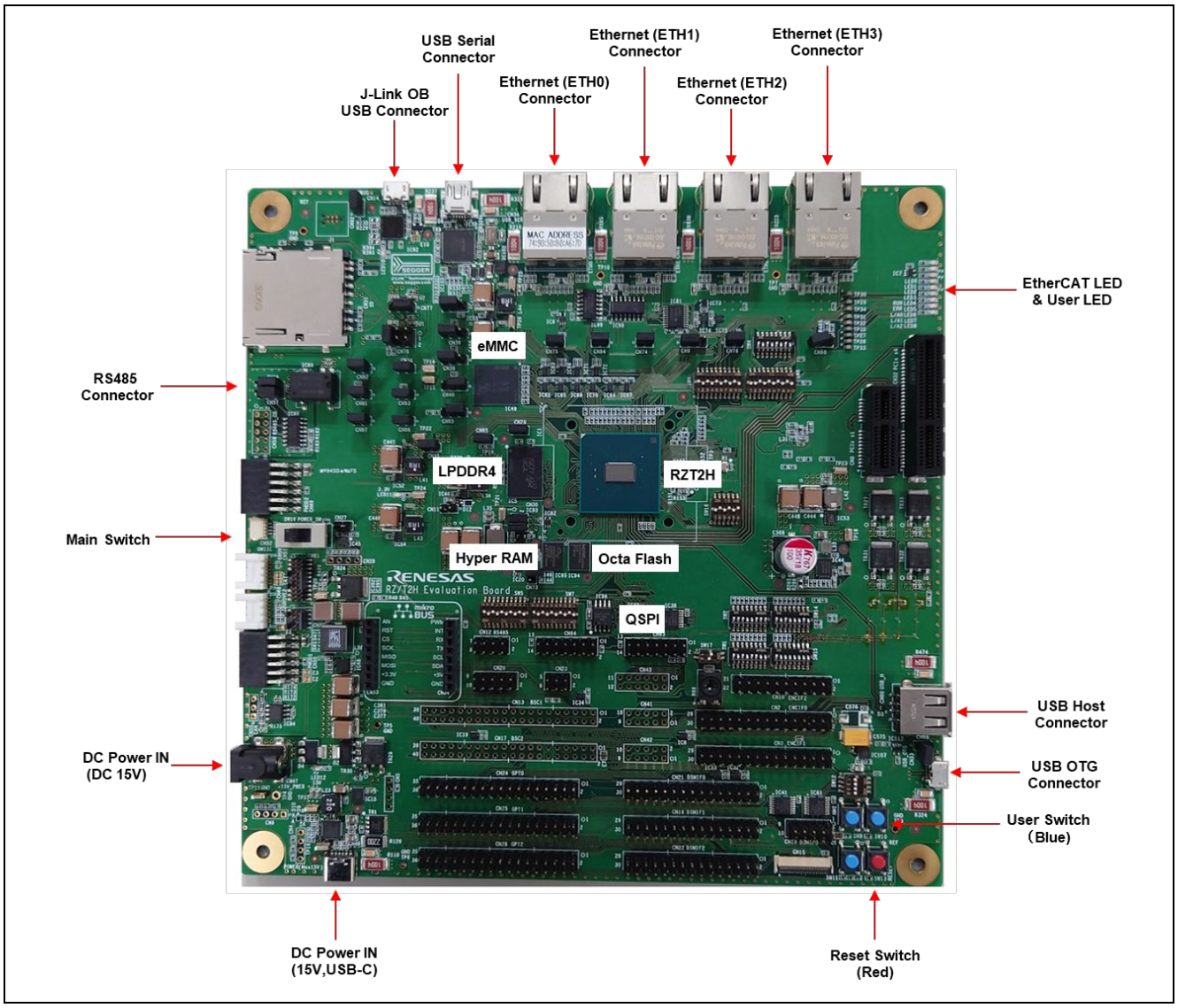


Figure 4.5. RZT2H Evaluation board layout

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 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。

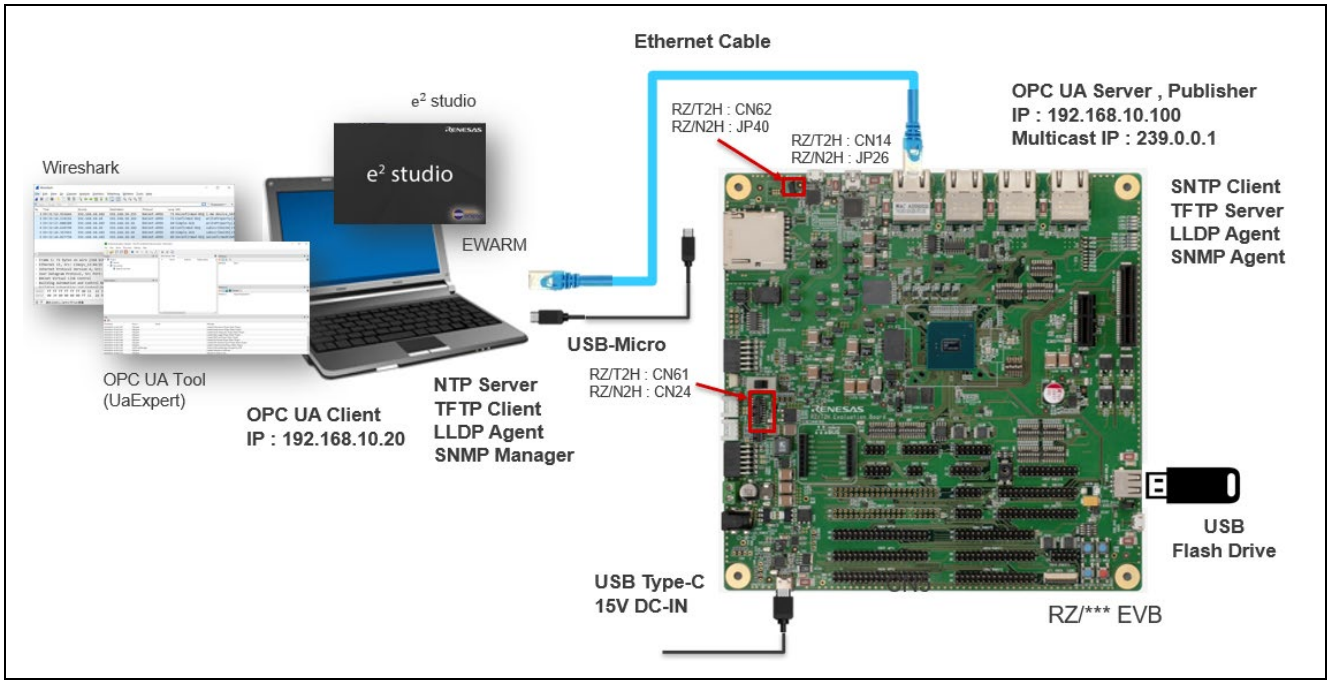


Figure 4.6. RZ/T2H EVB OPC UA Server hardware connection diagram

4.4 RZ/N2H Evaluation Board

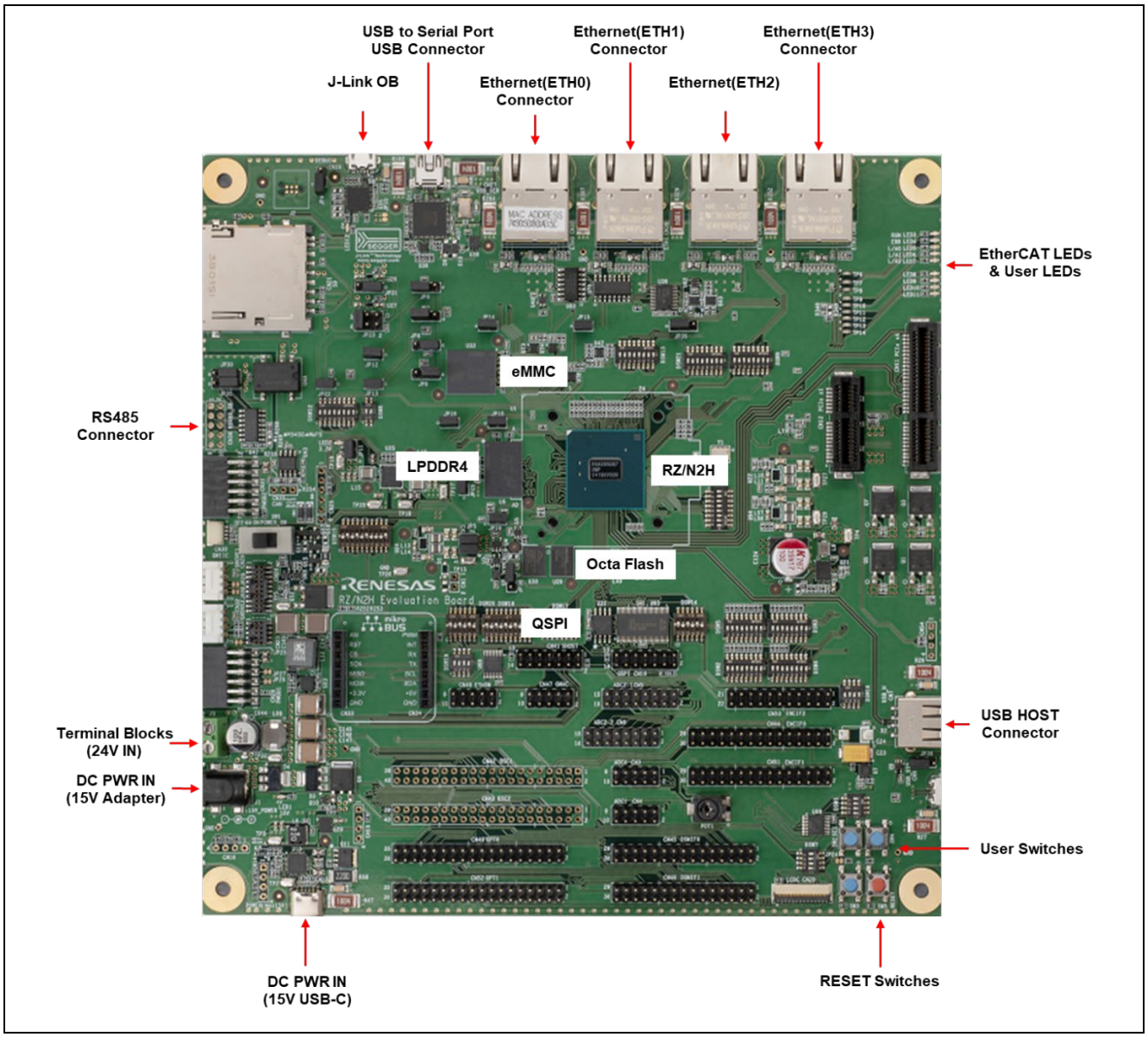


Figure 4.7. RZN2H Evaluation board layout

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_2. (for Ethernet Port 2)**
	Short 2-3	3.3-V power is supplied to VCC1833_2. (for Ethernet Port 2)
JP9	**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)
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_CRS 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_LEDRUN(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。

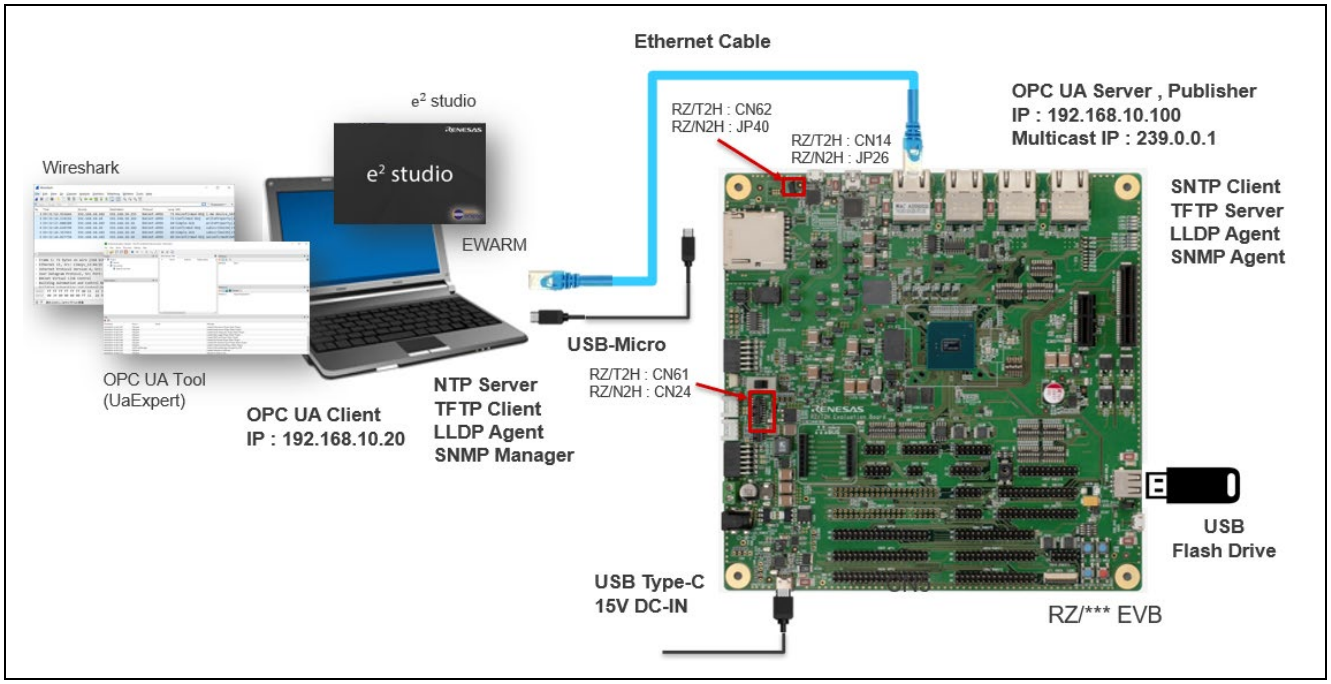


Figure 4.8. RZ/N2H EVB OPC UA Server hardware connection diagram

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

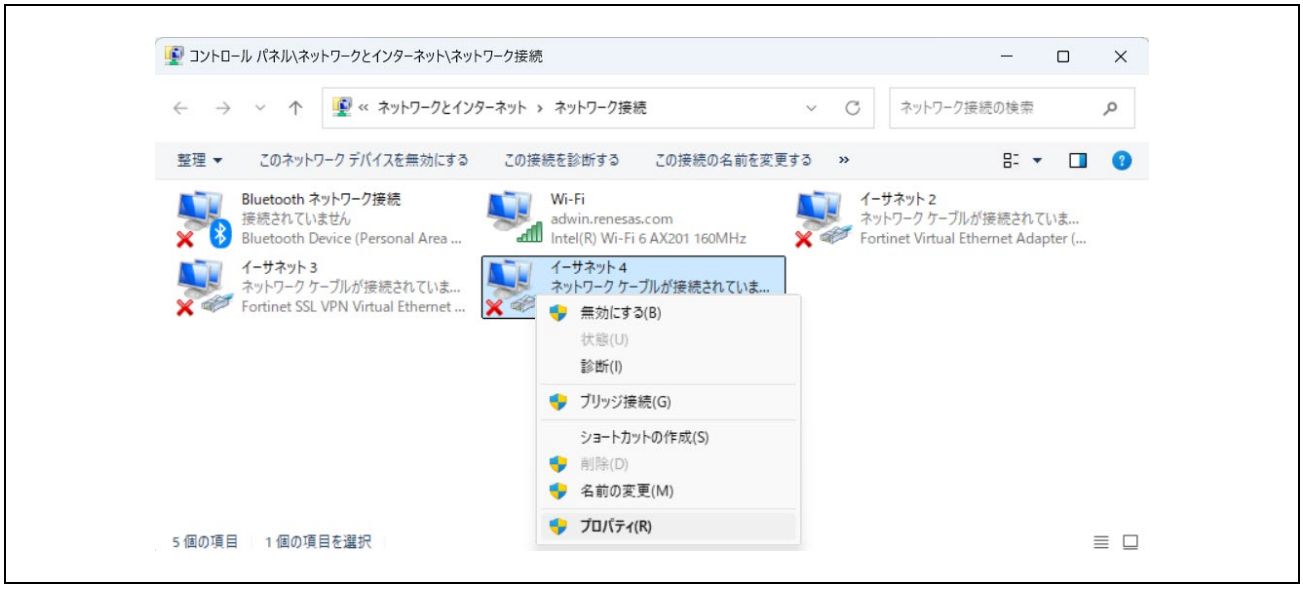


Figure 5.1. Network connection

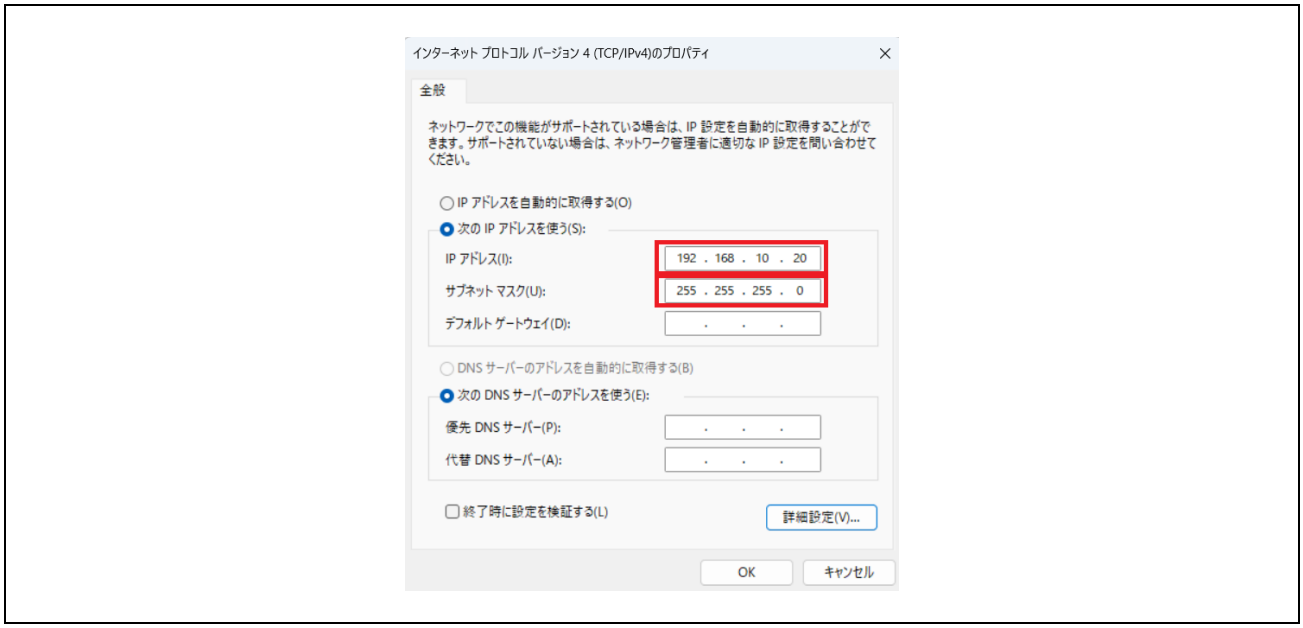


Figure 5.2. TCP/IPv4 properties

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.*** (100 以外の 1 から 254 の任意の値)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。

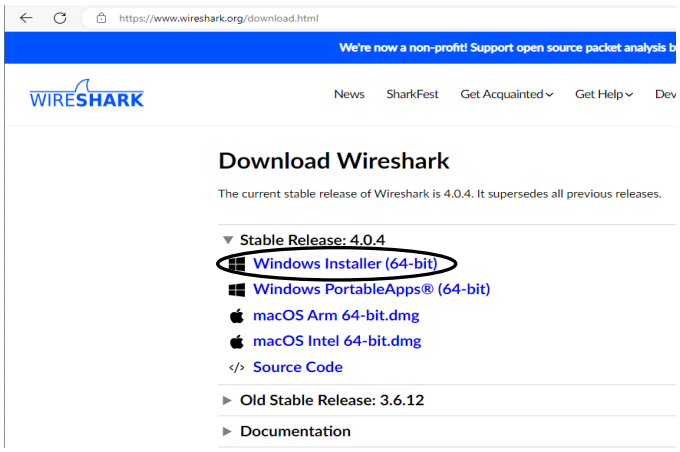


Figure 5.4. Download Wireshark

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

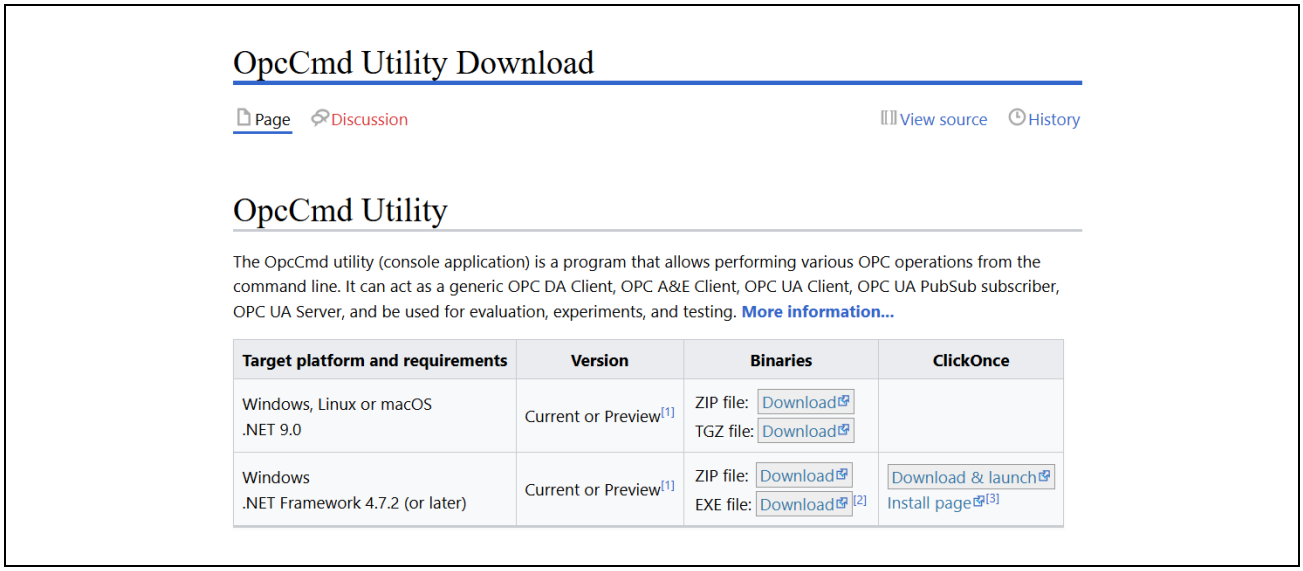


Figure 5.5. Download OpcCmd Utility

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

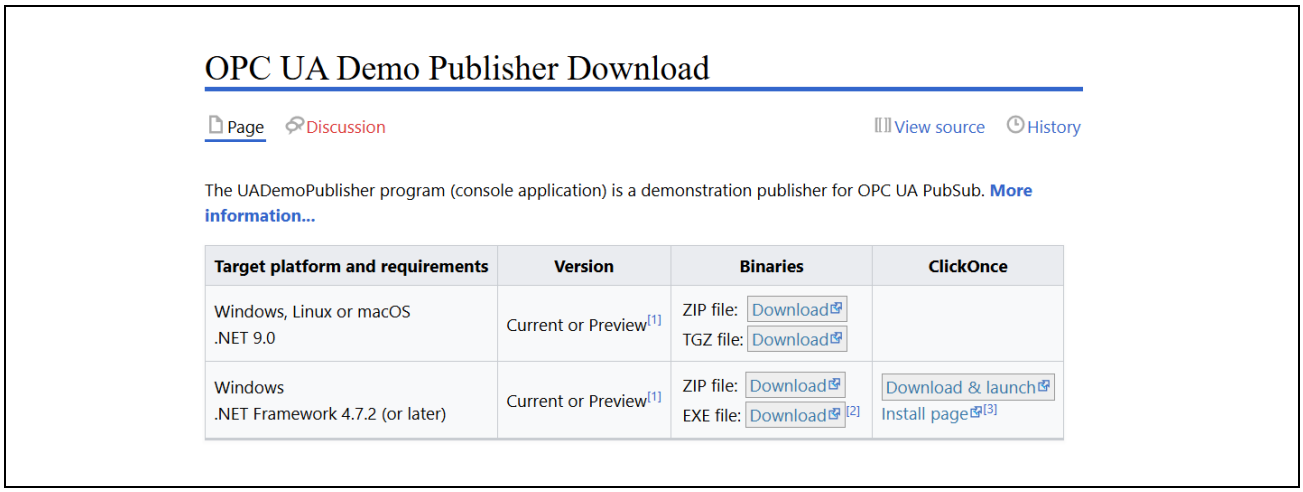


Figure 5.6. Download OPC UA Demo Publisher

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。

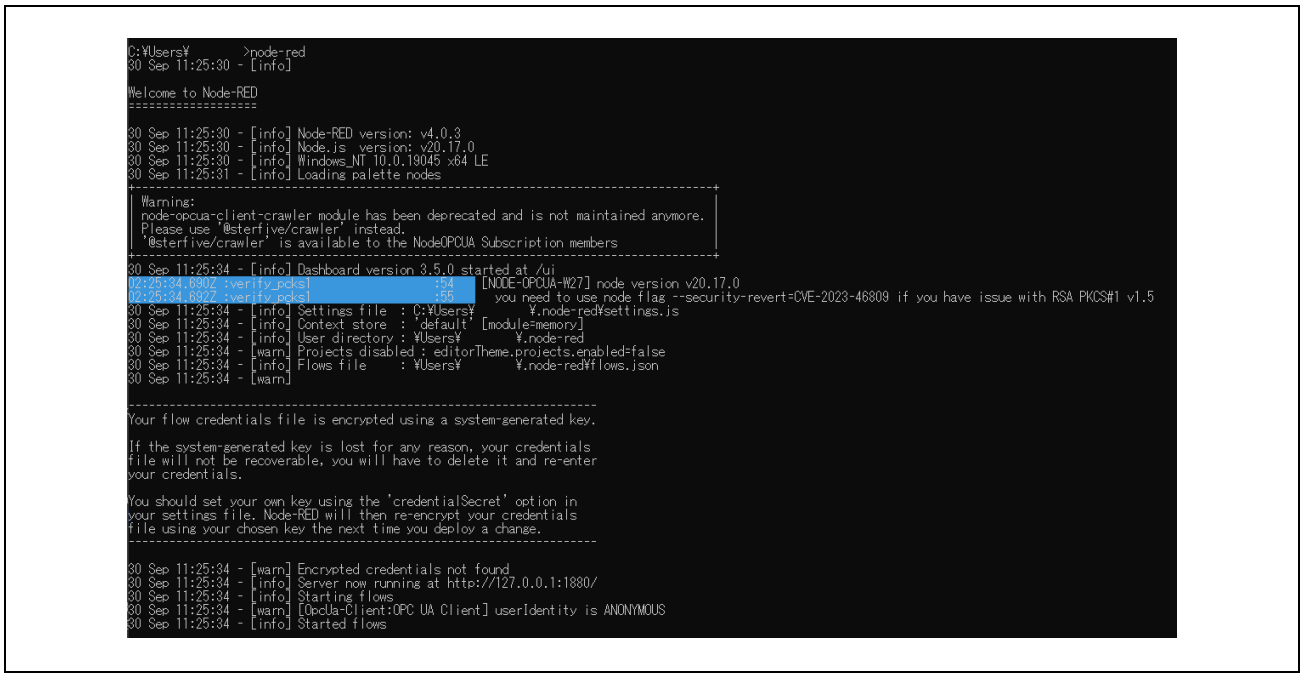


Figure 5.7. Run Node-RED

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

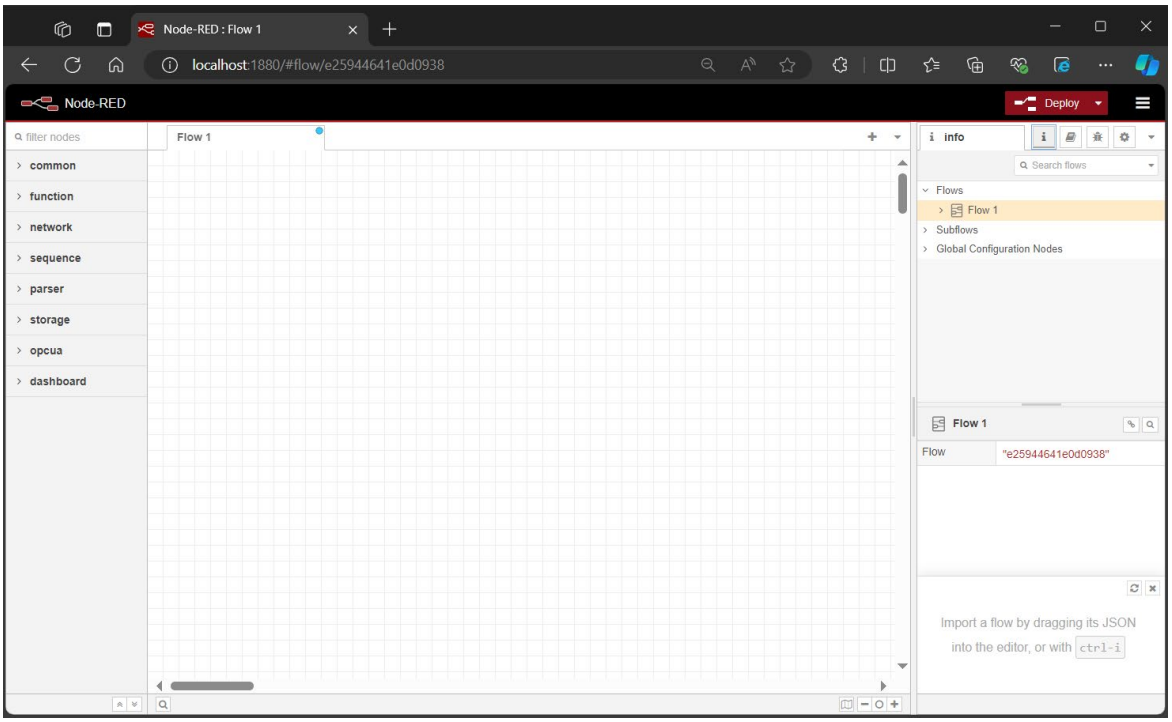


Figure 5.8. Start Node-RED

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\

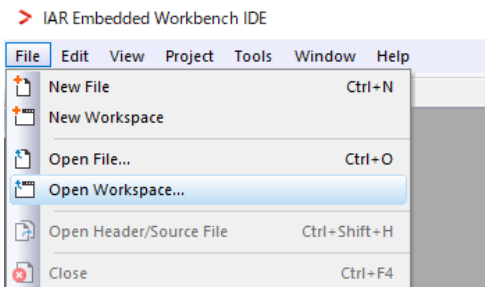


Figure 6.1. Open Workspace

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。

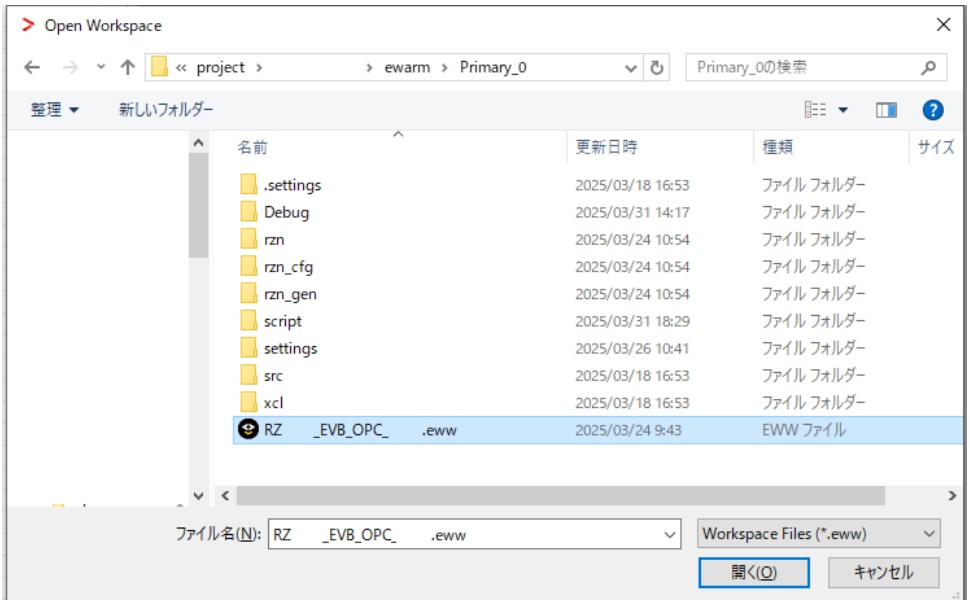


Figure 6.2. Open project file

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

※RZ/T2M で FSP v4.1.0 以降を使用時、もしくは RZ/T2H, RZ/N2H で FSP v4.2.0 以降を使用時は 9.7 章のガイドを参照ください。

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。

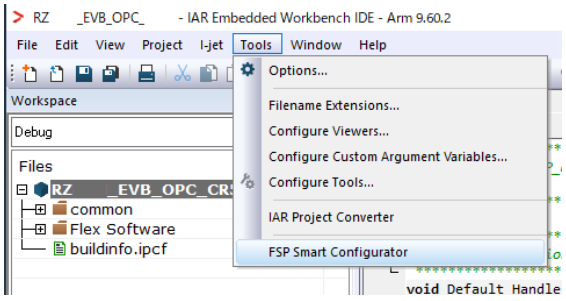


Figure 6.3. Tools tab

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。

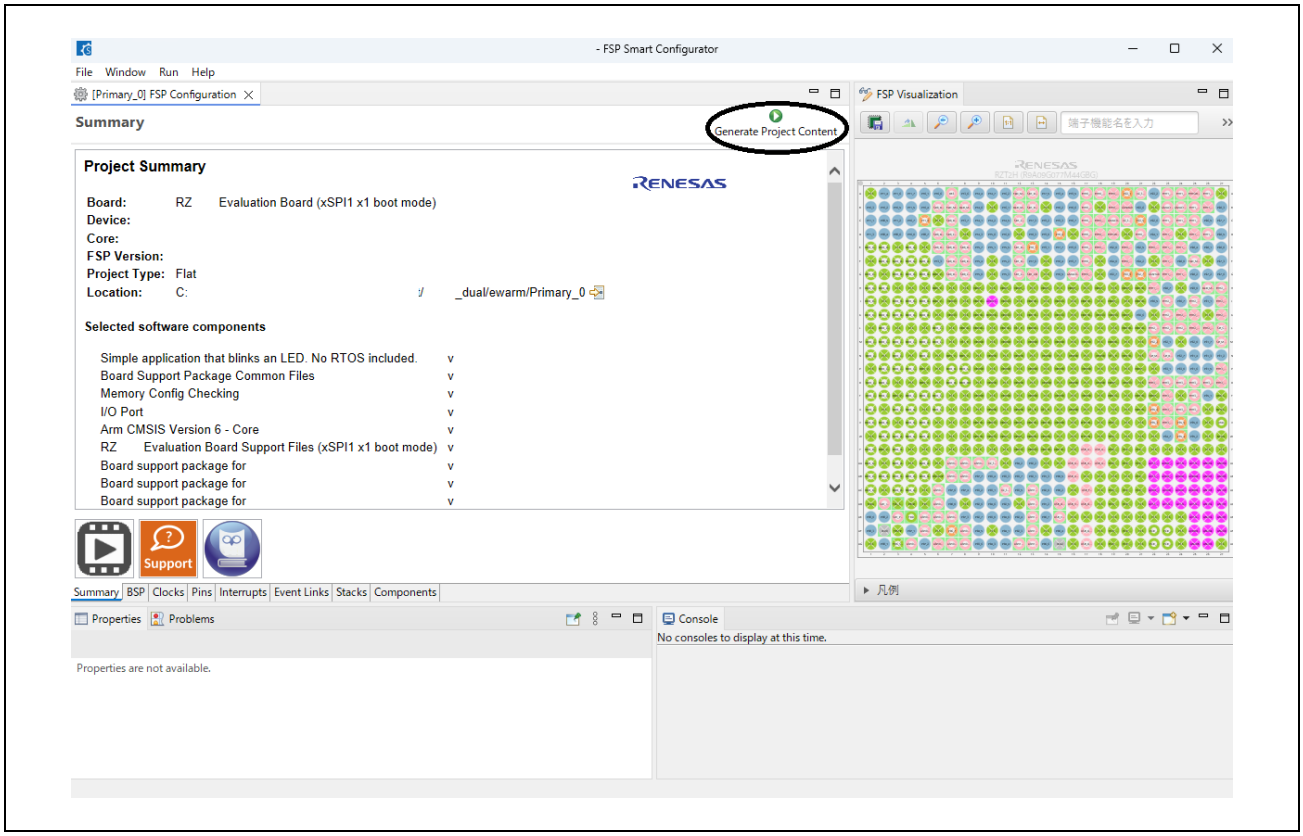


Figure 6.4. Generate Project Content

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

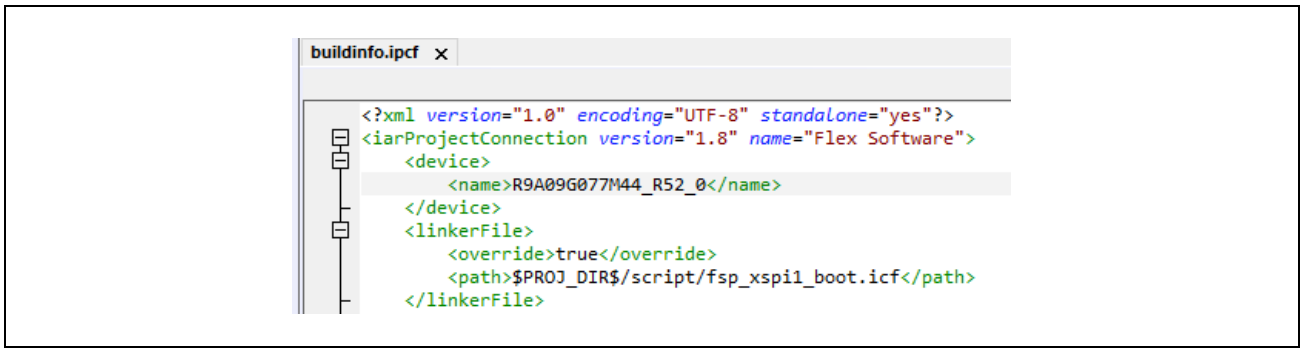


Figure 6.5. IAR EWARM Project File

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。

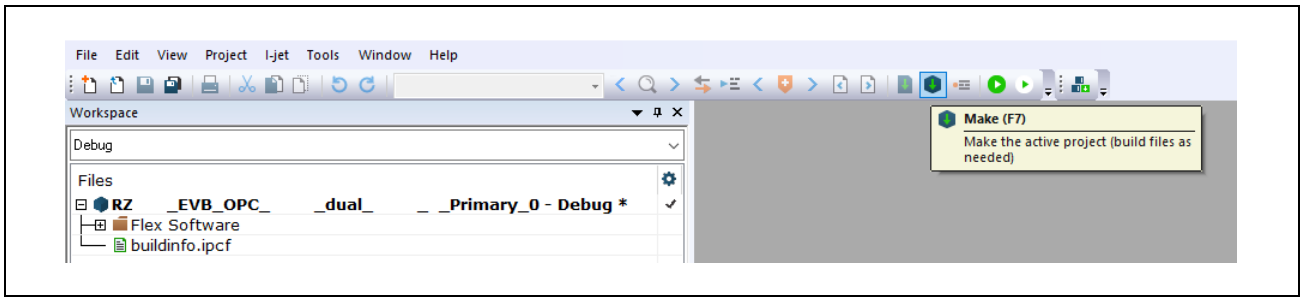


Figure 6.6. Make Primary project

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。

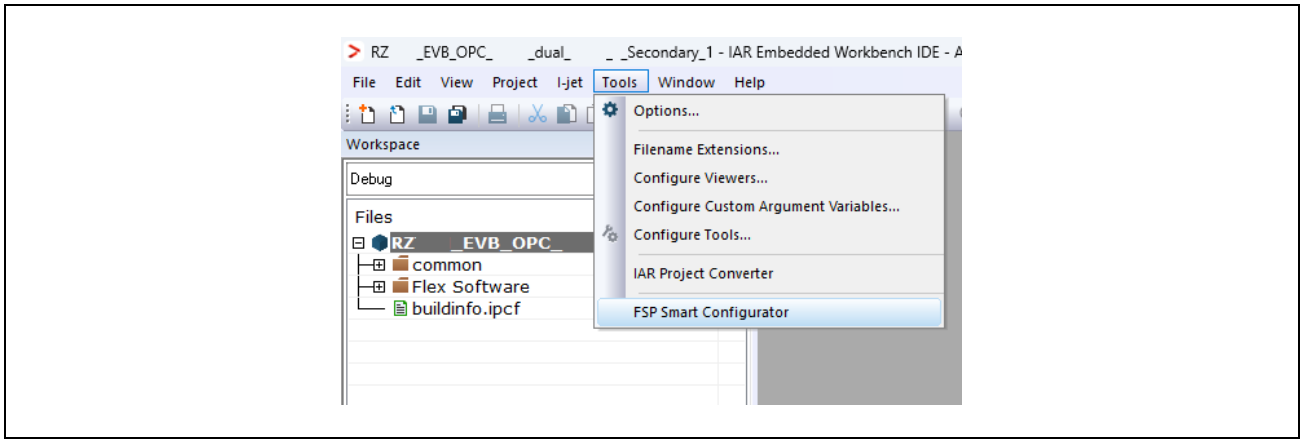


Figure 6.7. Open FSP SC

※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、9.6.1 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。

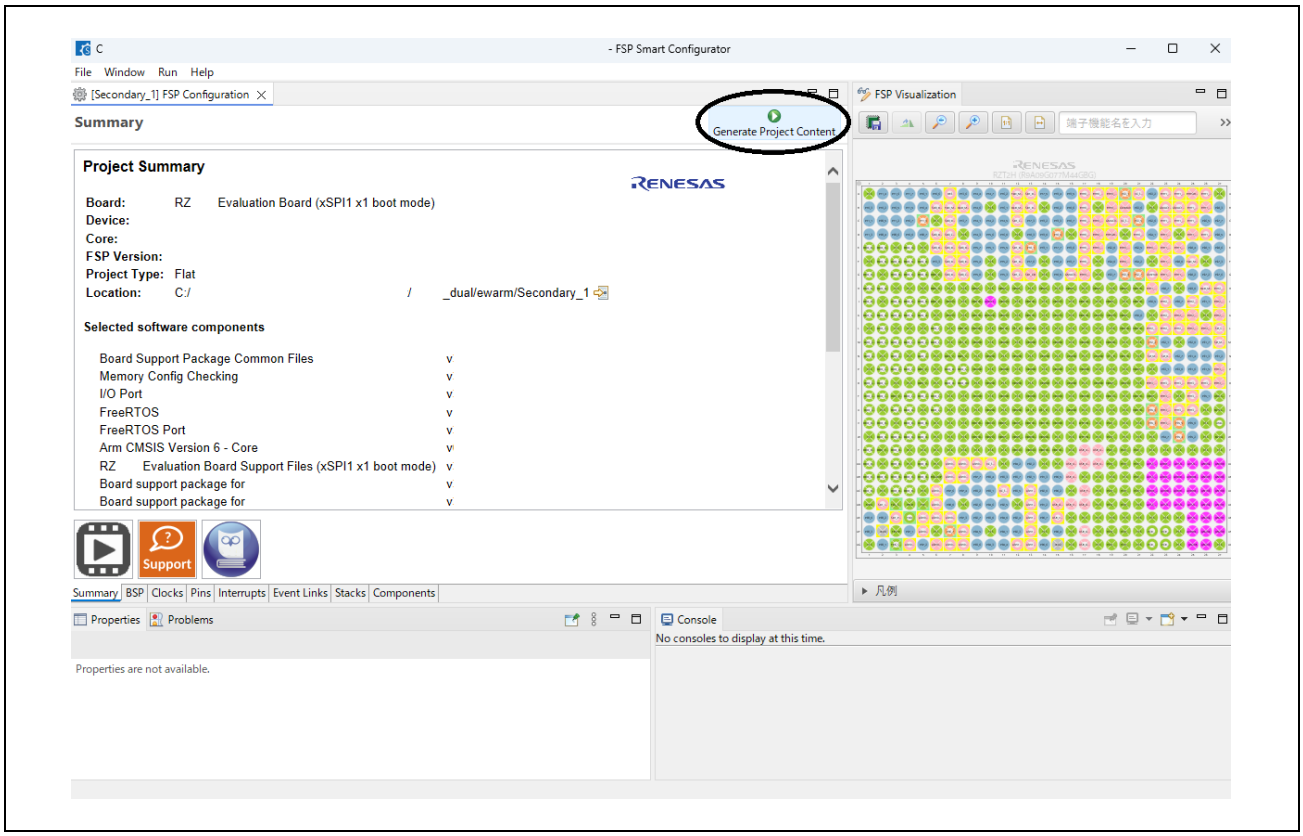


Figure 6.8. Generate Project Content

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

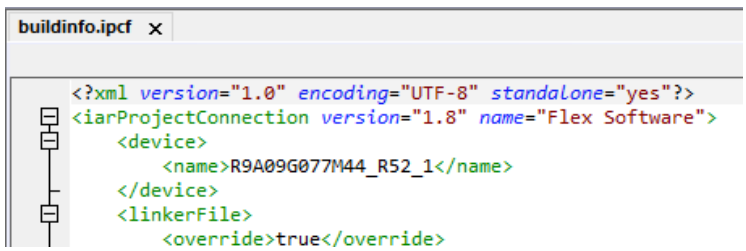


Figure 6.9. IAR EWARM Project File

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。

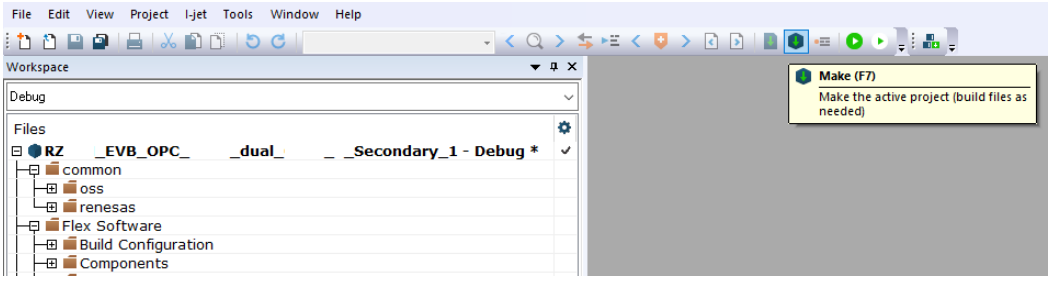


Figure 6.10. Make Secondary project

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。

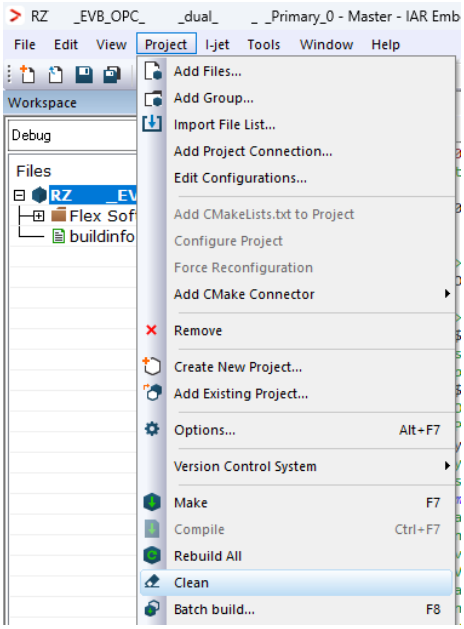


Figure 6.11. Clean Primary project

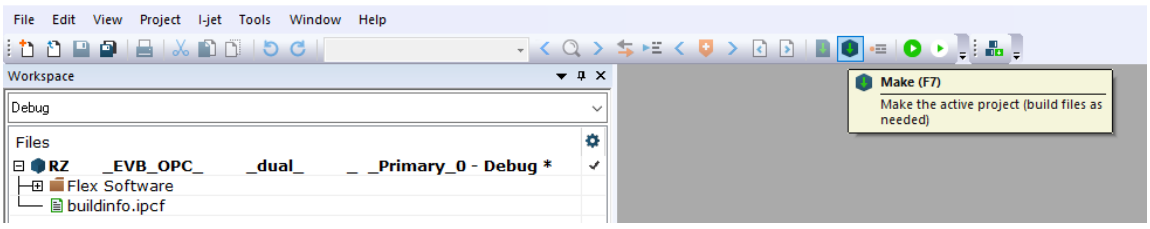


Figure 6.12. Make Primary project

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。

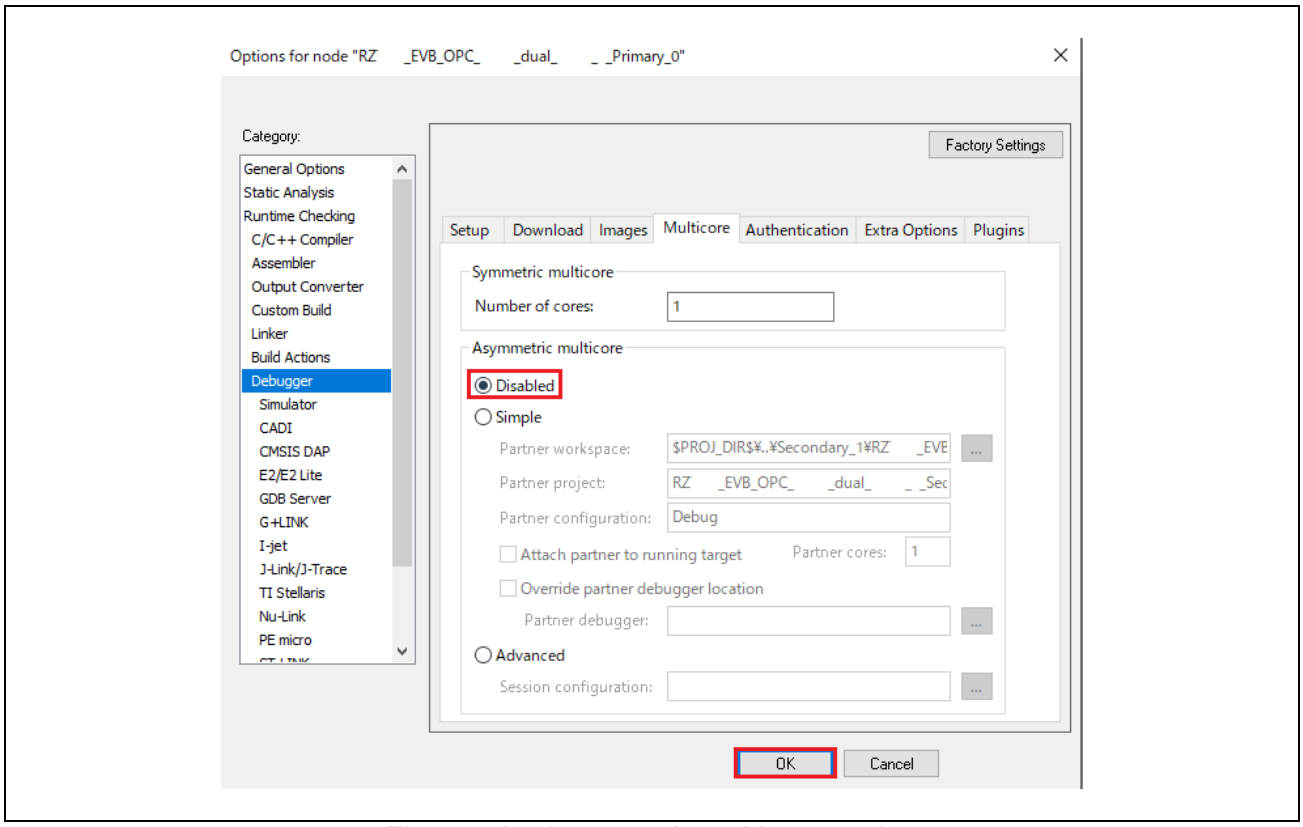


Figure 6.13. Asymmetric multicore setting

Project タブから Download＞Download file…をクリックします。

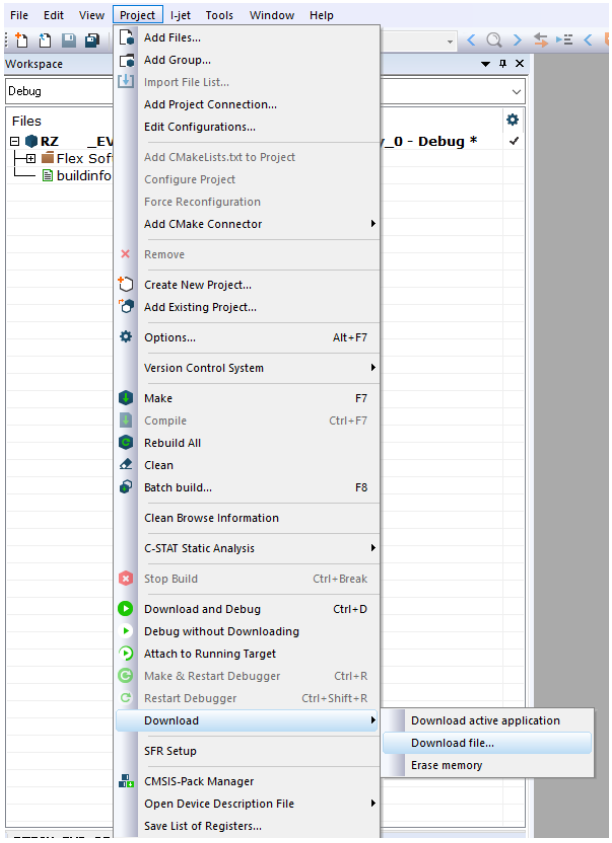


Figure 6.14. Download file…

\(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。

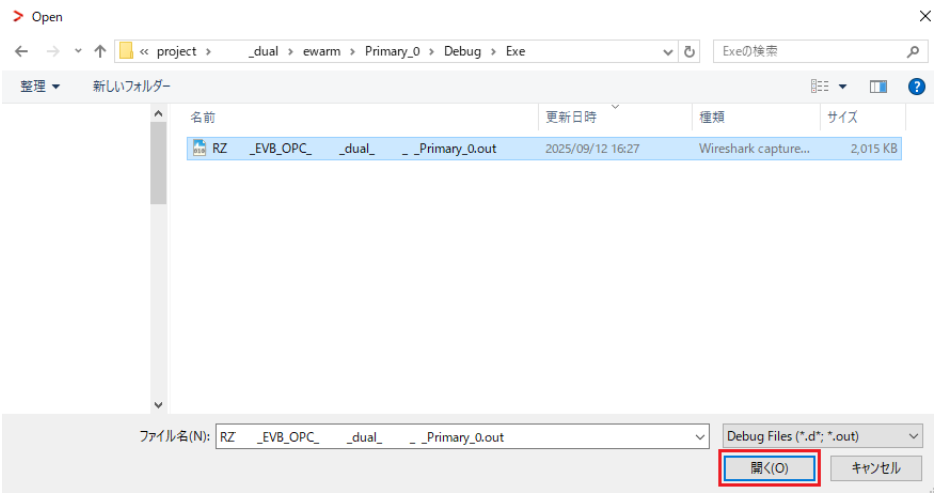


Figure 6.15. Open .out file

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 タブから、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

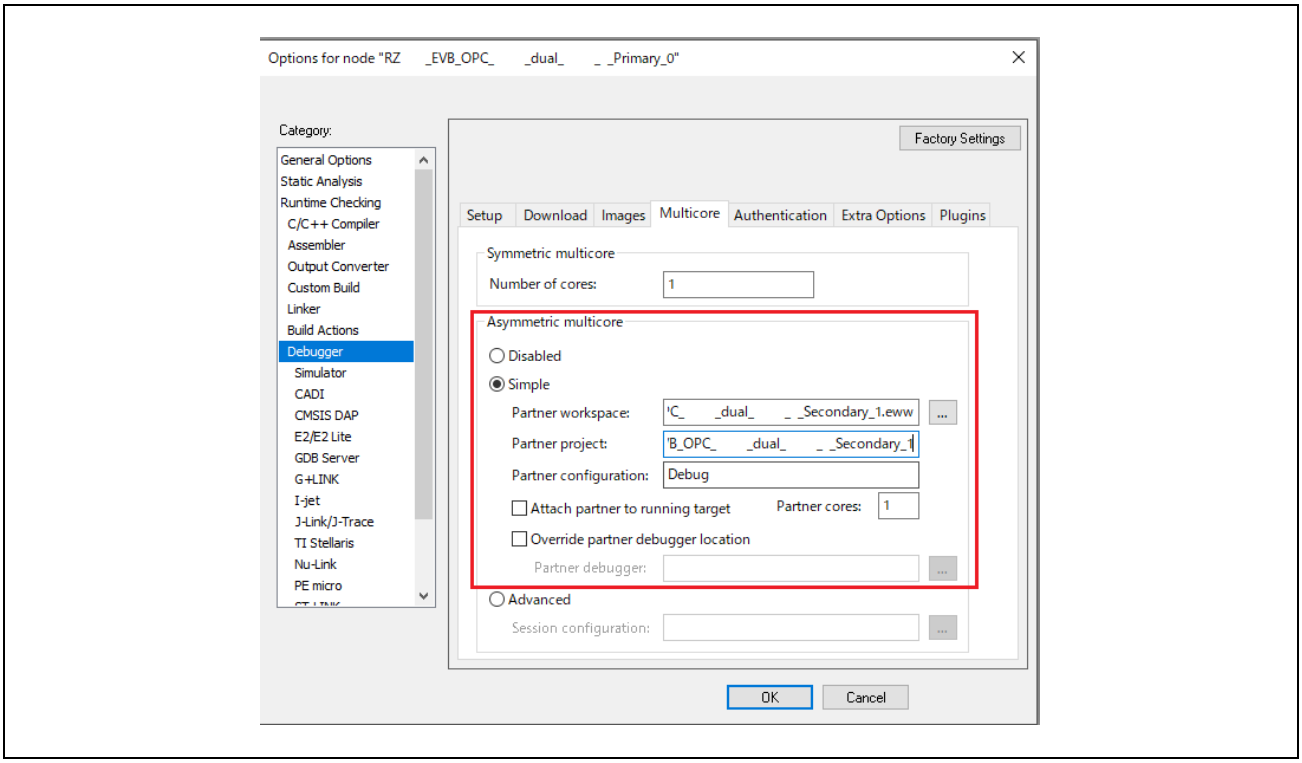


Figure 6.16. Asymmetric multicore setting

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。

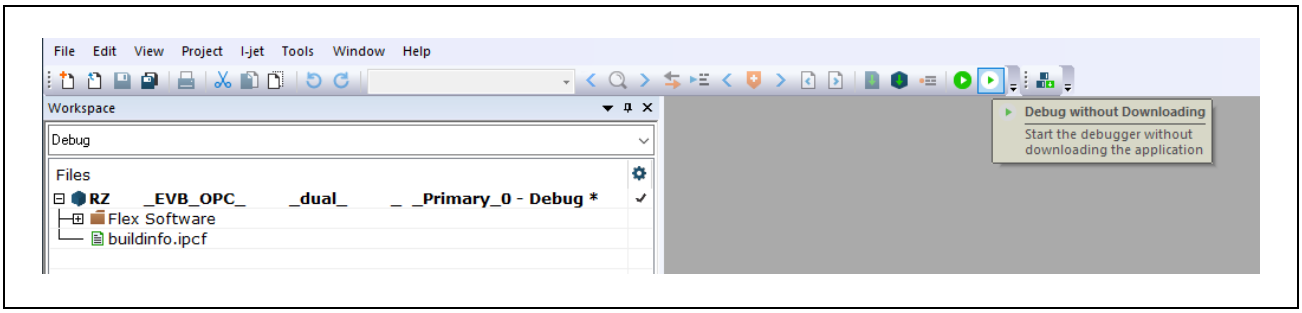


Figure 6.17. Debug without Downloading

6.1.2 For Single core project

6.1.2.1 Build

※RZ/N2L で FSP v4.1.0 以降を使用時は 9.7 章のガイドを参照ください。

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。

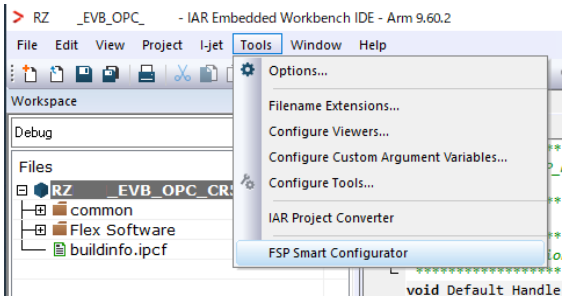


Figure 6.18. Tools tab

※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、9.6.1 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。

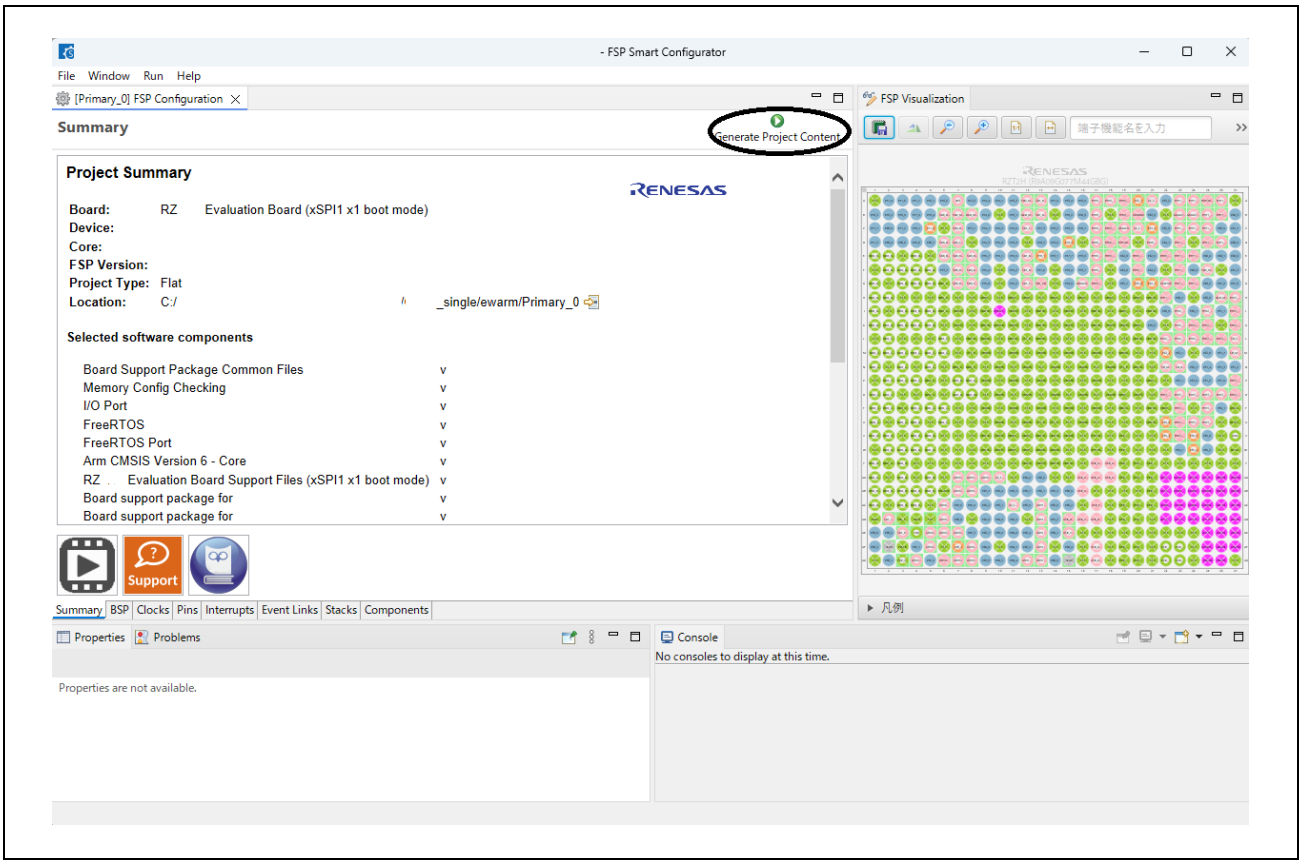


Figure 6.19. Generate Project Content

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。

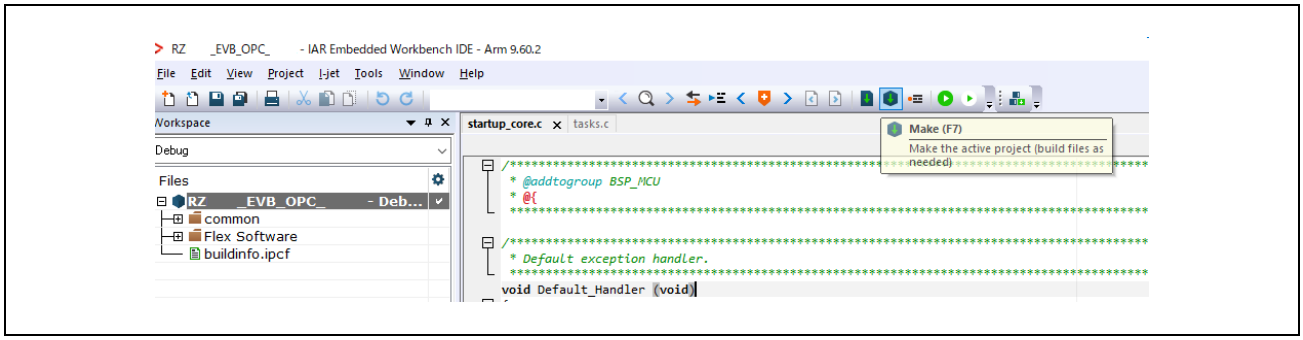


Figure 6.20. Make button

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。

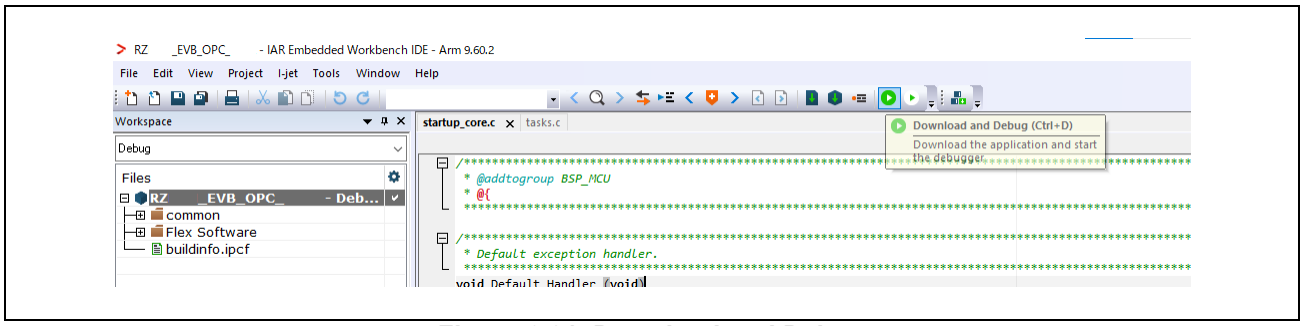


Figure 6.21. Download and Debug

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。

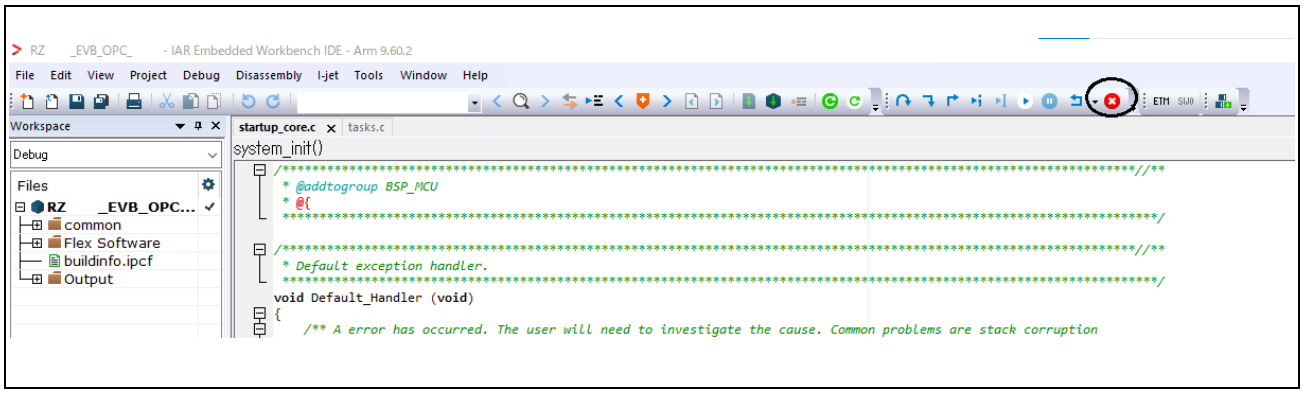


Figure 6.22. Stop Debugging

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\e2studio_v20**-**_fsp_v*.*.*\eclipse\e2studio.exe

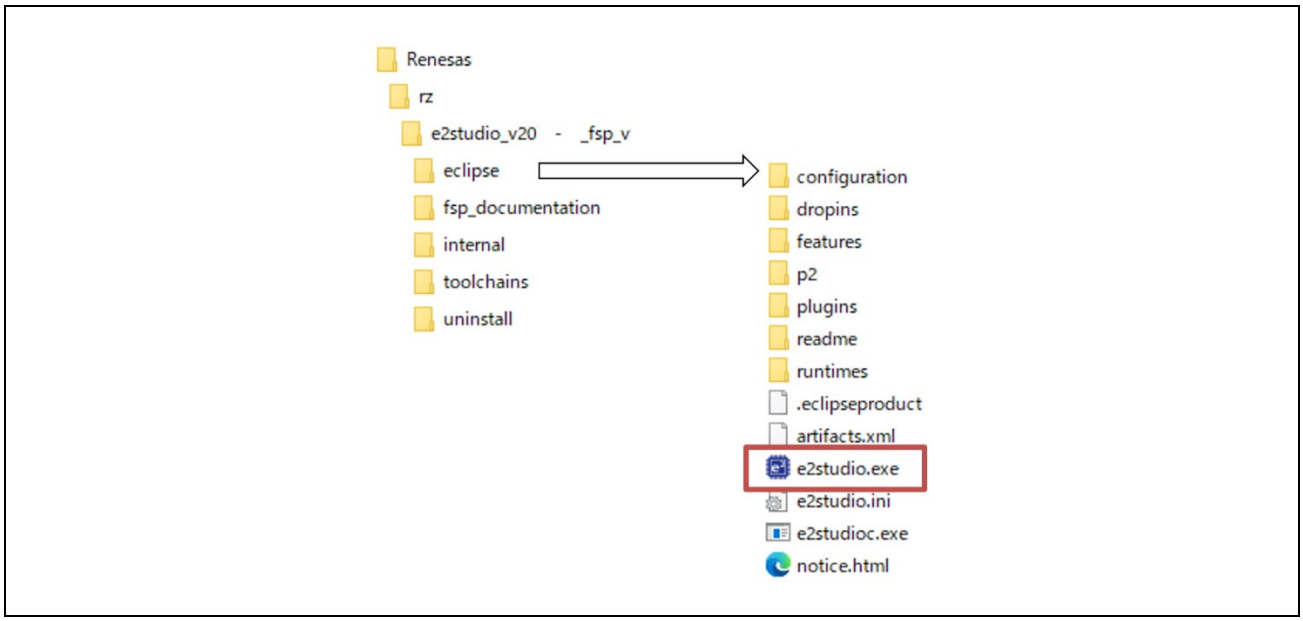


Figure 6.23. Launch project (1)

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。

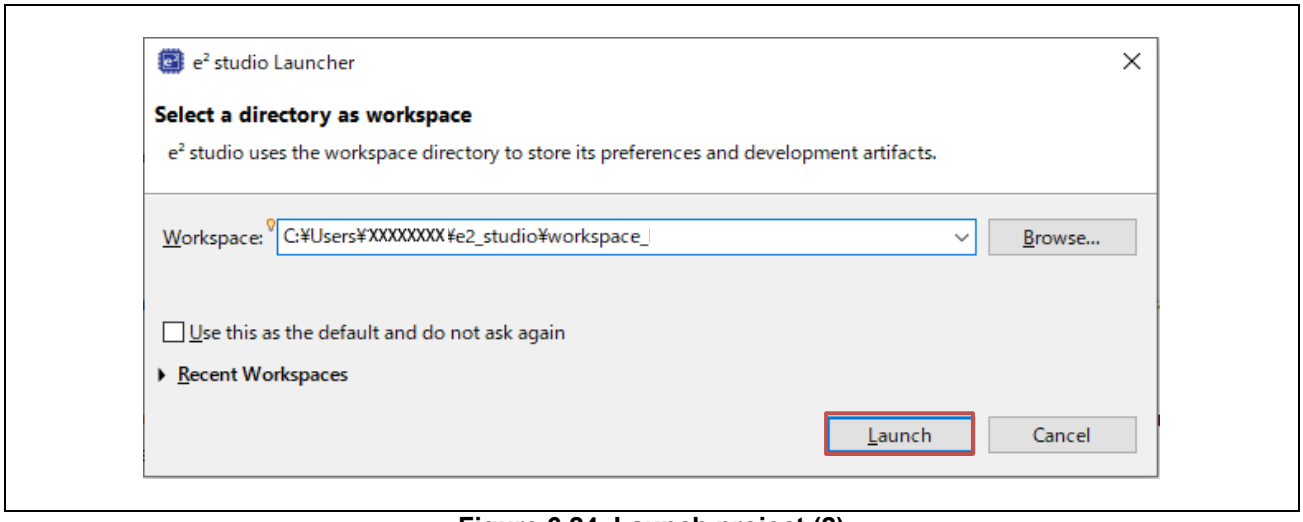


Figure 6.24. Launch project (2)

・Import existing projects をクリックします。

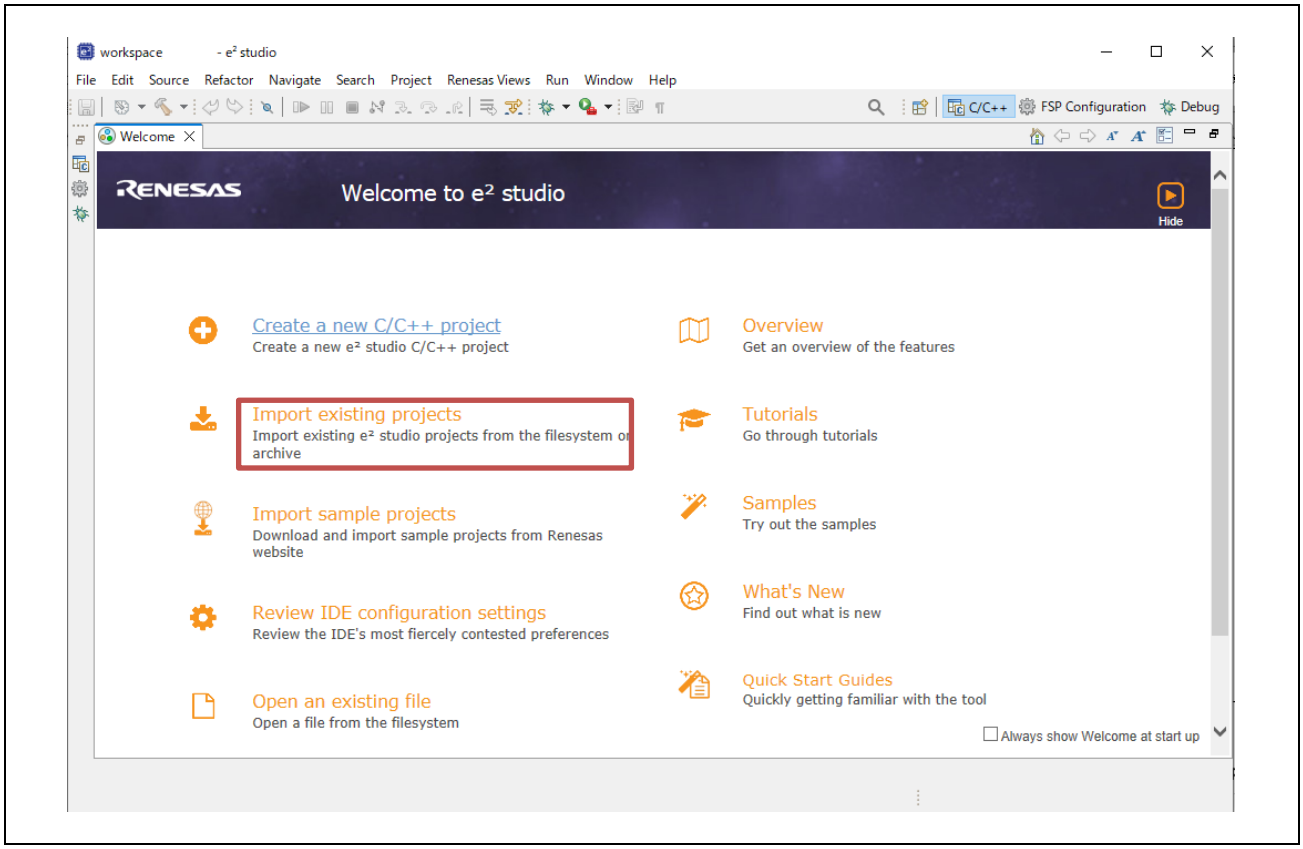


Figure 6.25. Launch project (3)

・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\

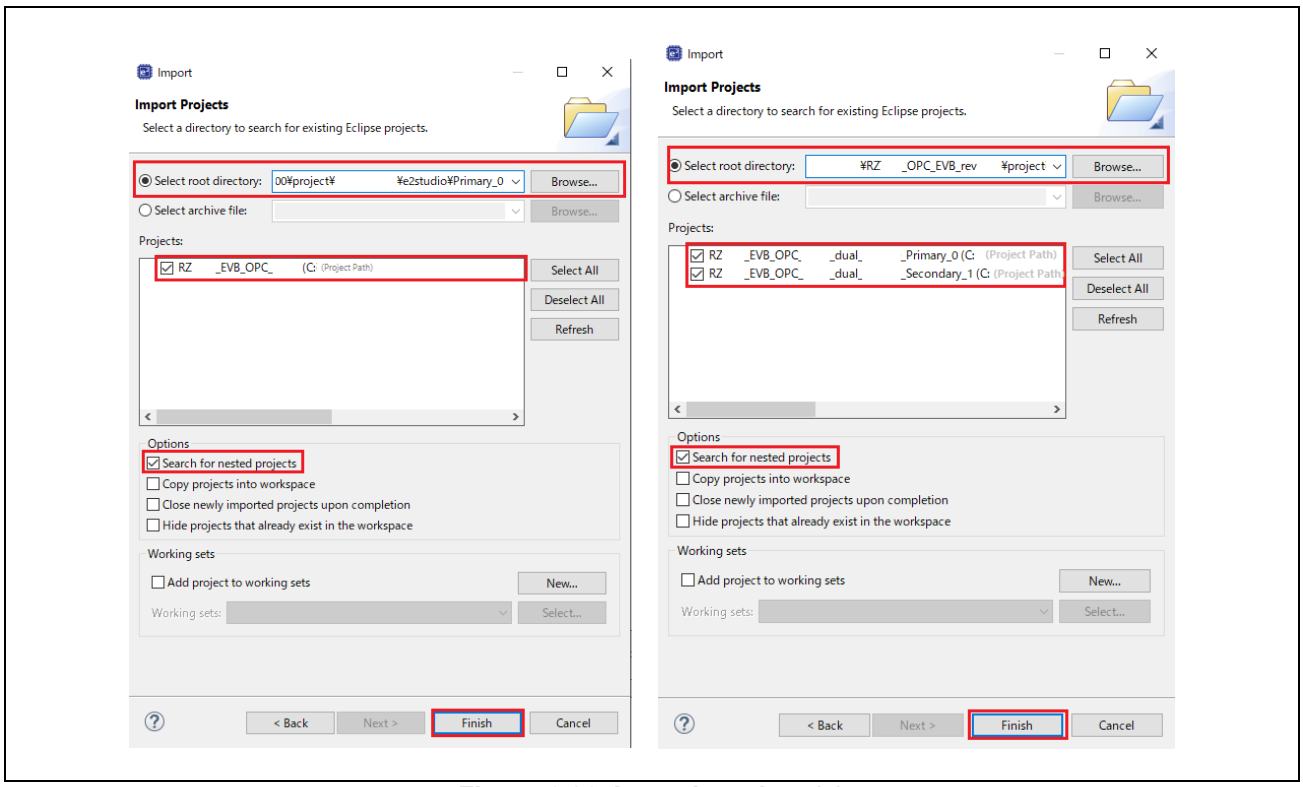


Figure 6.26. Launch project (4)

・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

※RZ/T2M で FSP v4.1.0 以降を使用時、もしくは RZ/T2H, RZ/N2H で FSP v4.2.0 以降を使用時は 9.7 章のガイドを参照ください。

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。

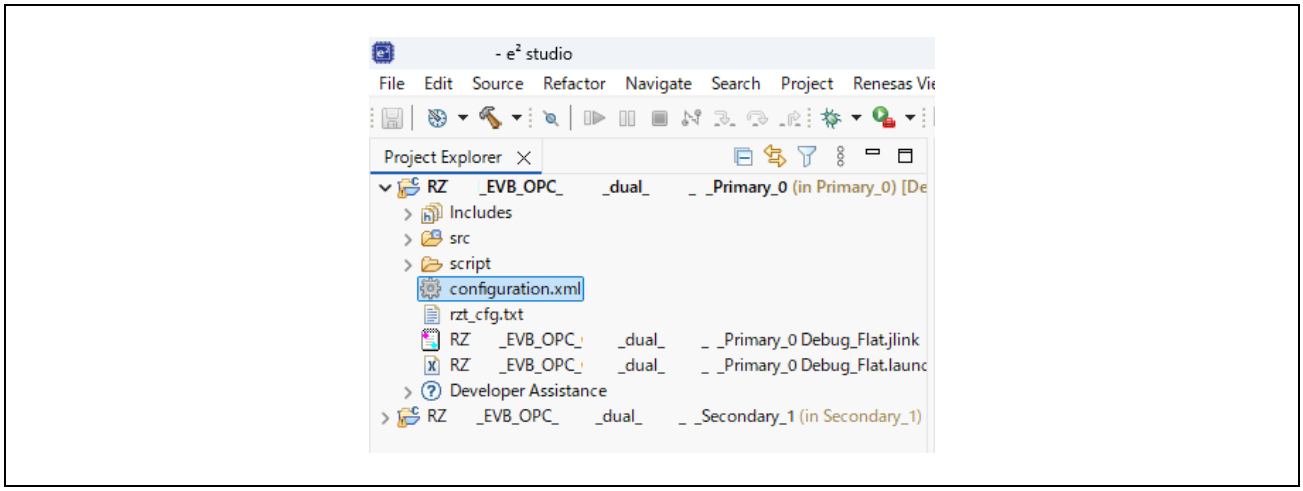


Figure 6.27. Open Smart configurator

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

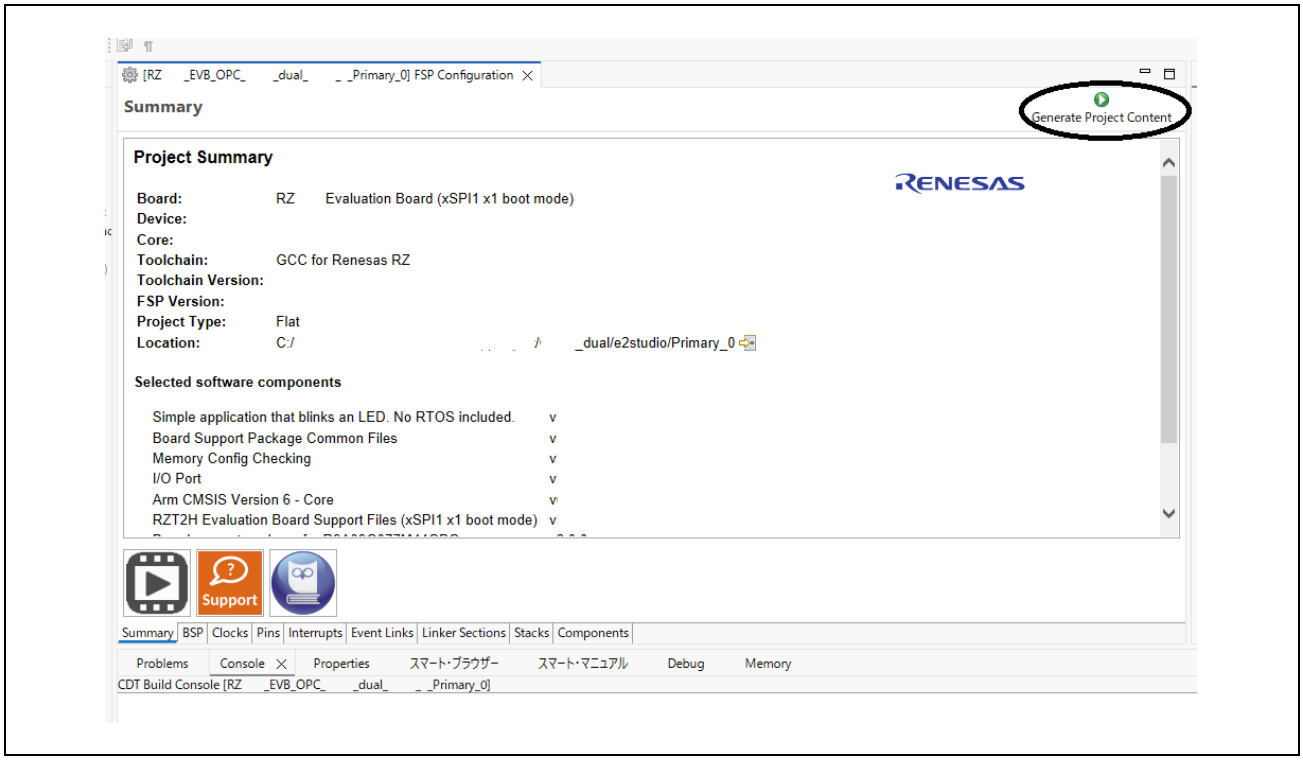


Figure 6.28. Generate Project Content

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。

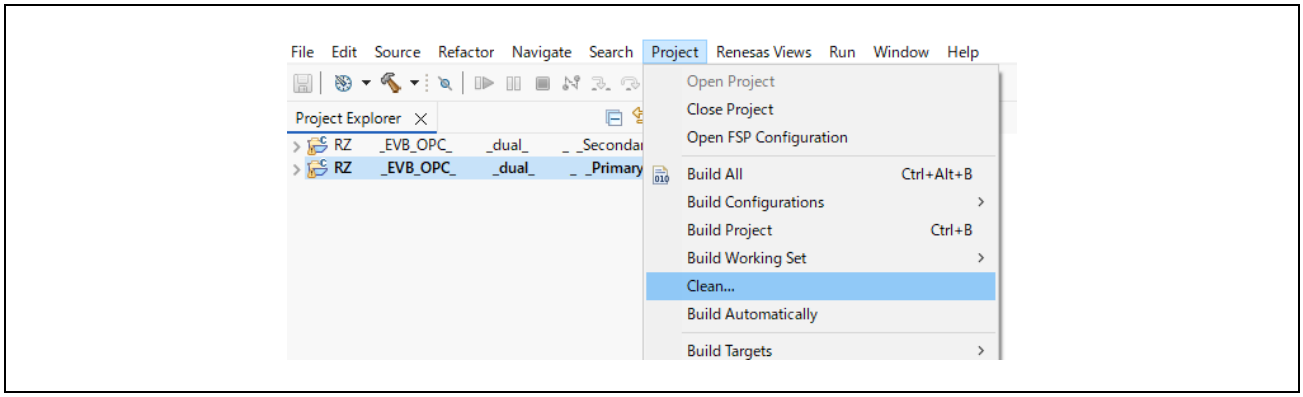


Figure 6.29. Open project Clean…

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。

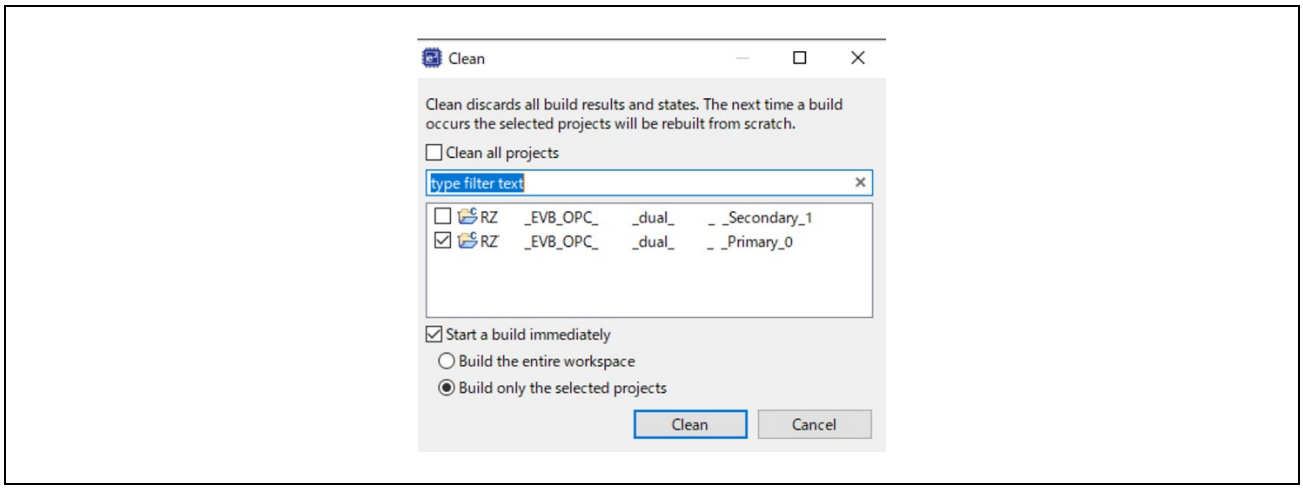


Figure 6.30. Clean and rebuild Primary project

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。

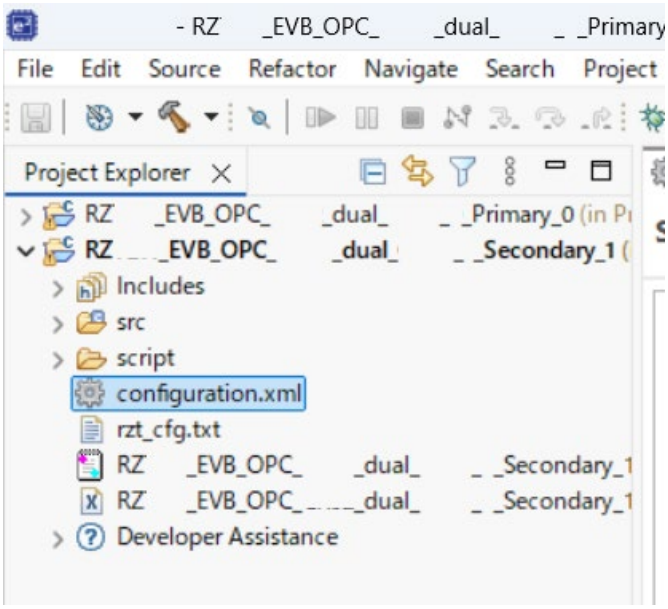


Figure 6.31. Open Smart configurator

※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、9.6.1 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。

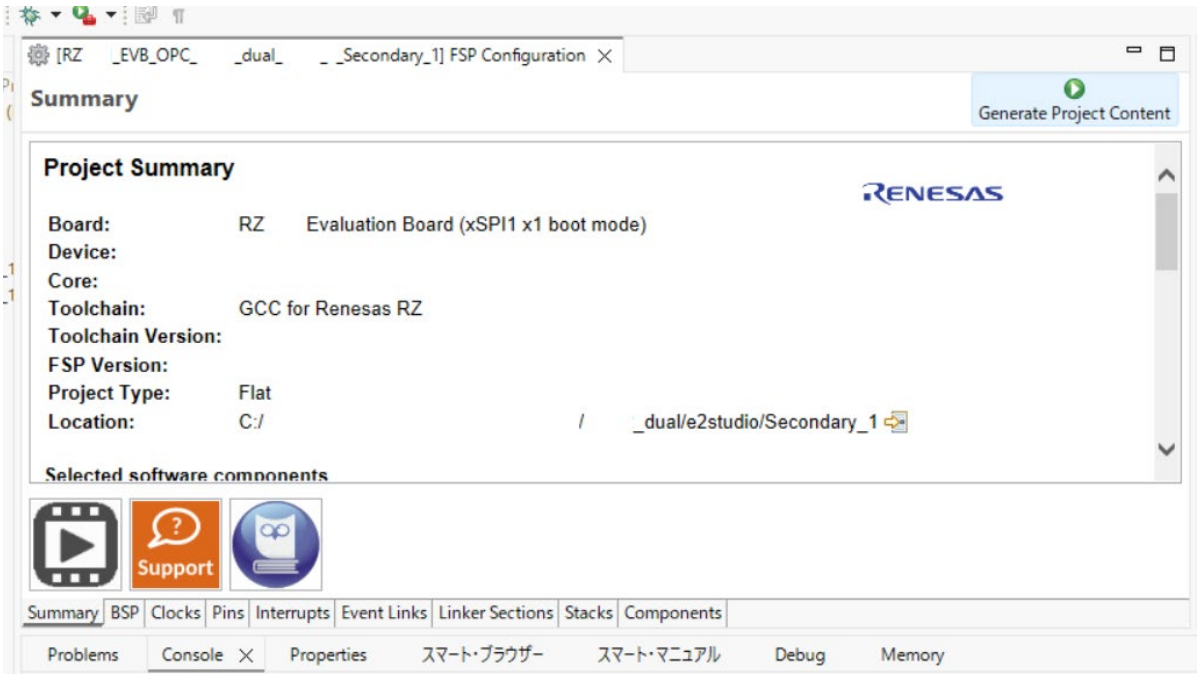


Figure 6.32. Generate Project Content

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

```
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
}
```

CA55

```
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
}
```

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。

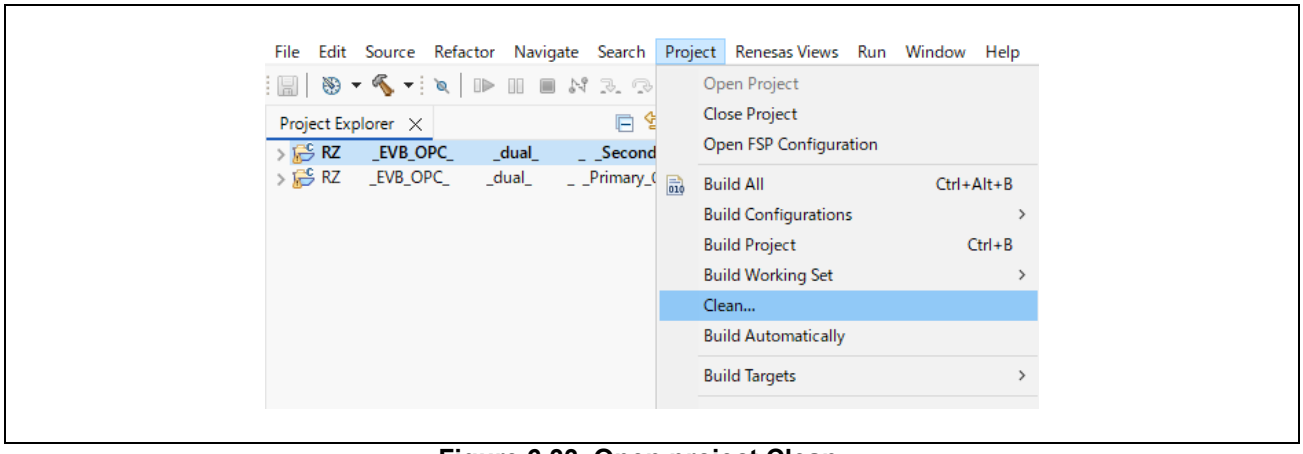


Figure 6.33. Open project Clean…

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。

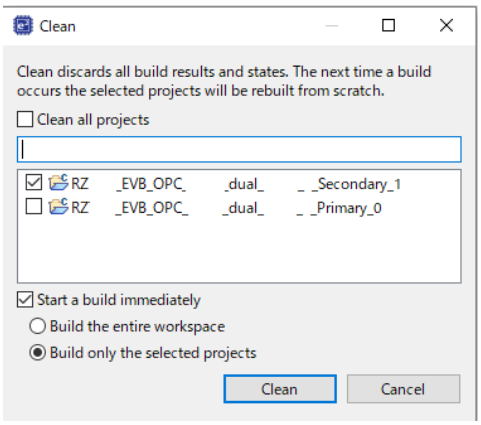


Figure 6.34. Clean and rebuild Secondary project

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。

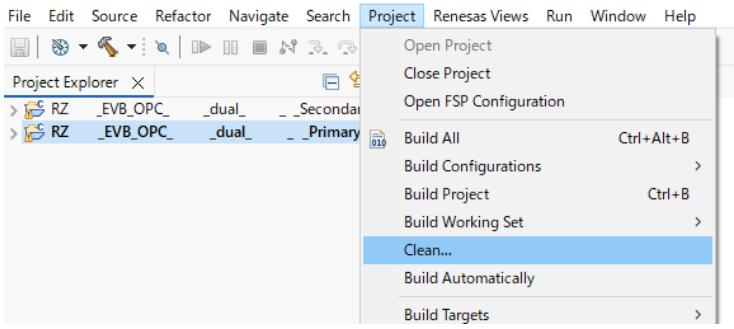


Figure 6.35. Open project Clean…

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。

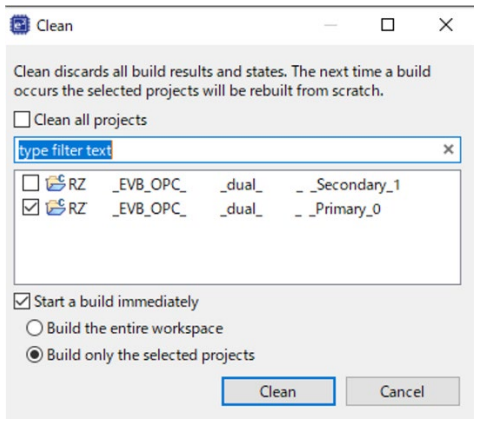


Figure 6.36. Clean and rebuild Primary project

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。

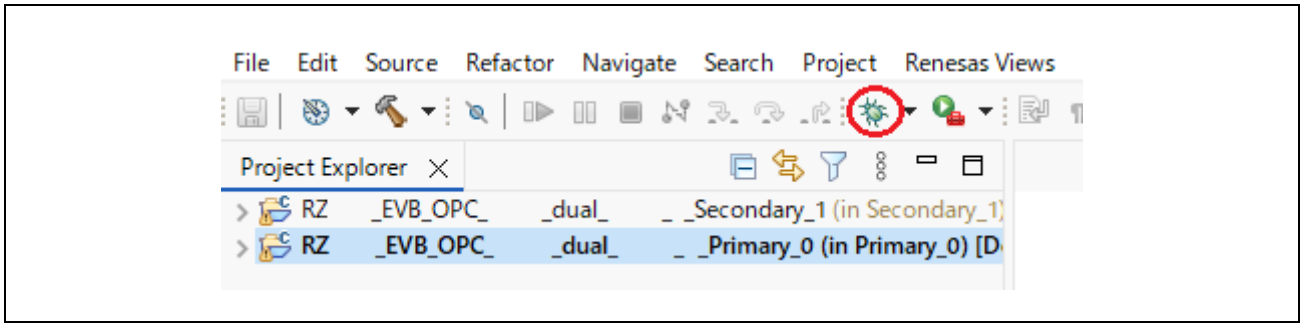


Figure 6.37. Debug Primary

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。

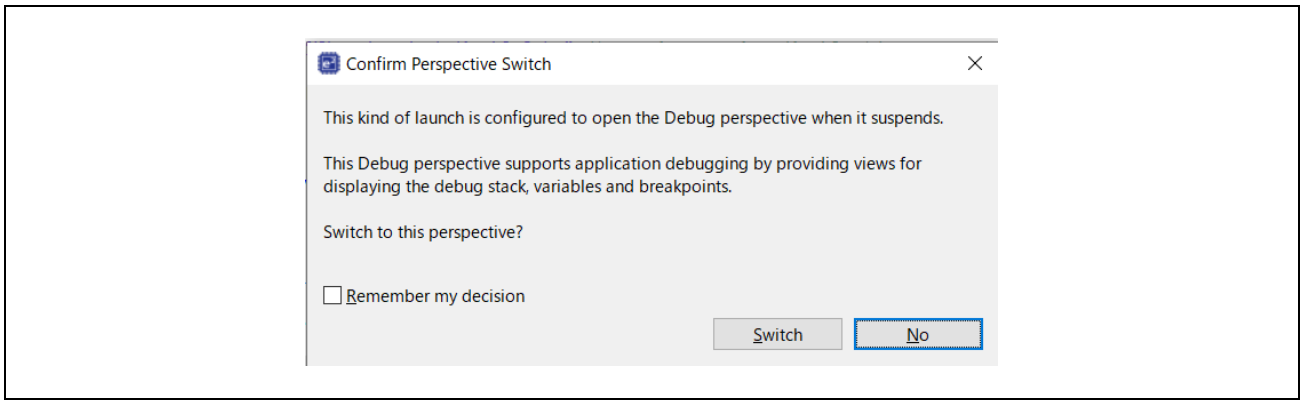


Figure 6.38. Perspective Switch

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。

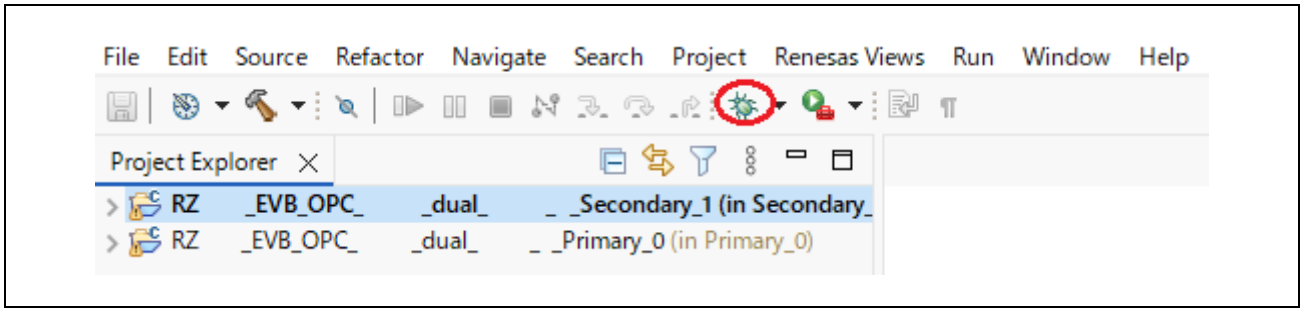


Figure 6.39. Debug Secondary

Figure 6.40 が表示されるので No をクリックしてください。

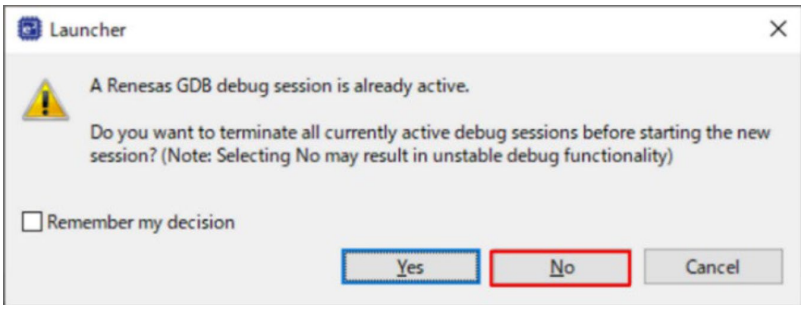


Figure 6.40. Warning Window of Starting Debug Session

Figure 6.41 が表示されるので Yes をクリックしてください。

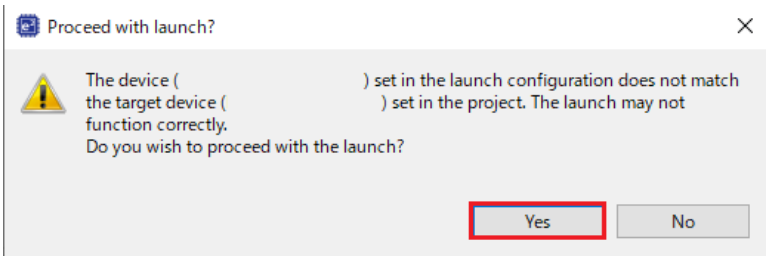


Figure 6.41. Warning Window of Device Name

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

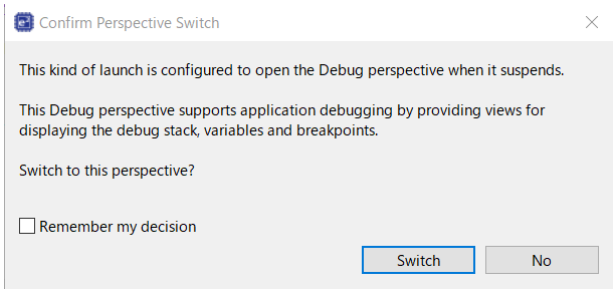


Figure 6.42. Perspective Switch

Debug タブから各 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。

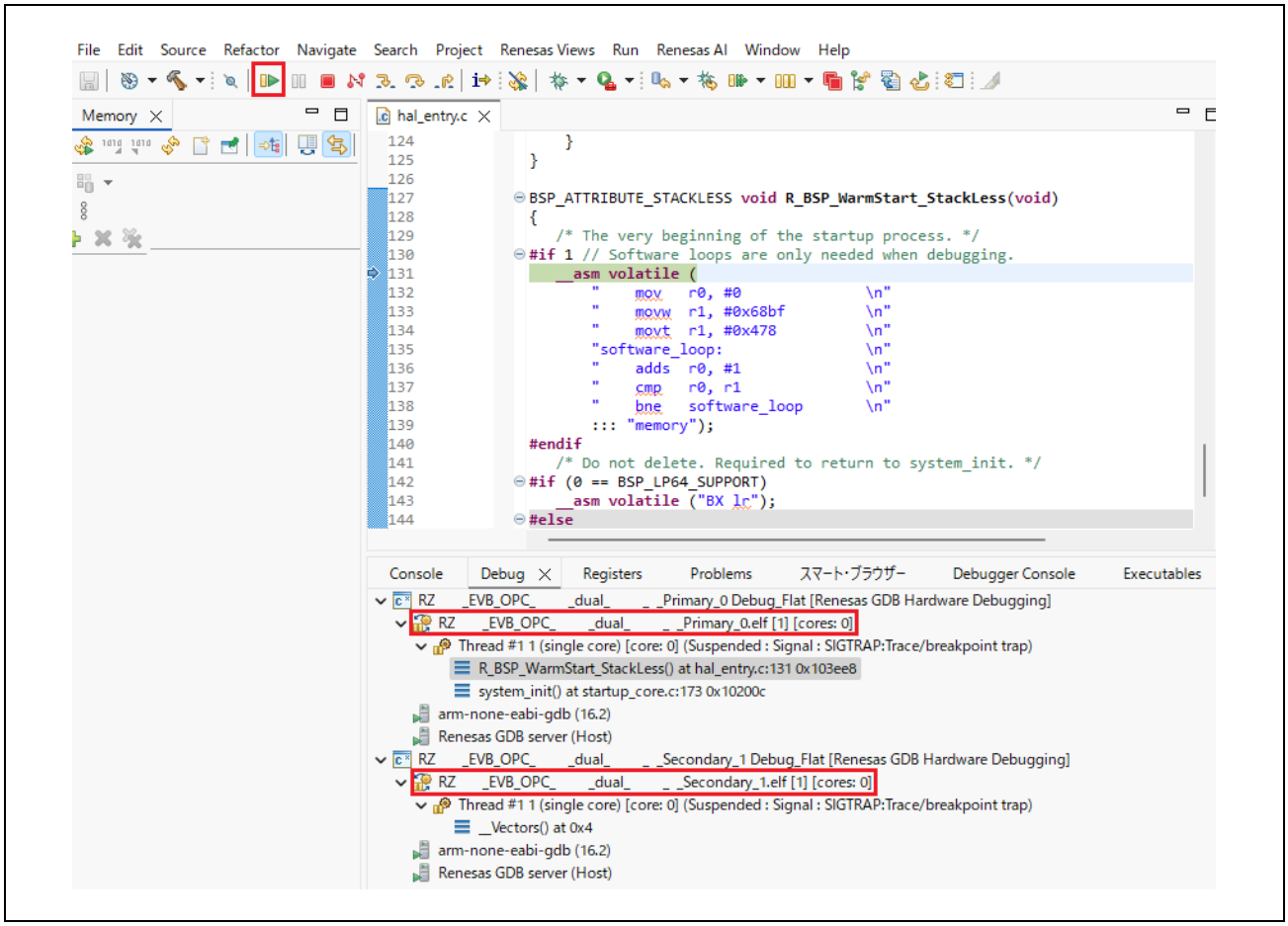


Figure 6.43. Debug

(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。

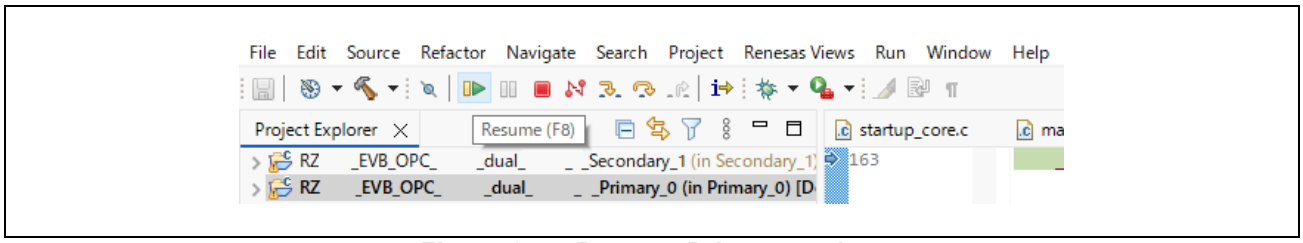


Figure 6.44. Resume Primary project

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。

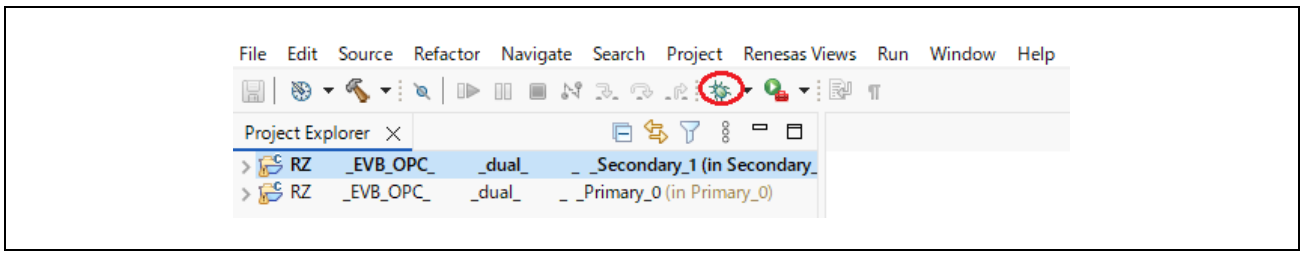


Figure 6.45. Debug Secondary project

Figure 6.46 が表示されるので No をクリックしてください。

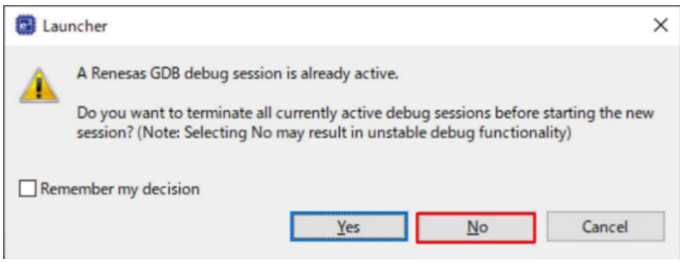


Figure 6.46. Warning Window of Starting Debug Session

Figure 6.47 が表示されるので Yes をクリックしてください。

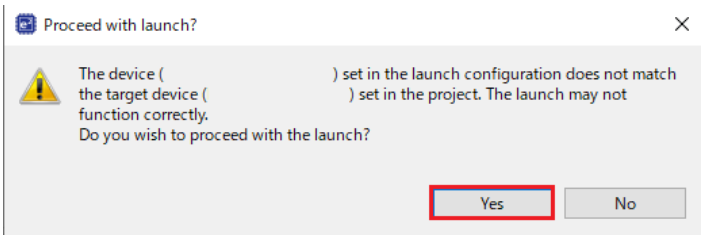


Figure 6.47. Warning Window of Device Name

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

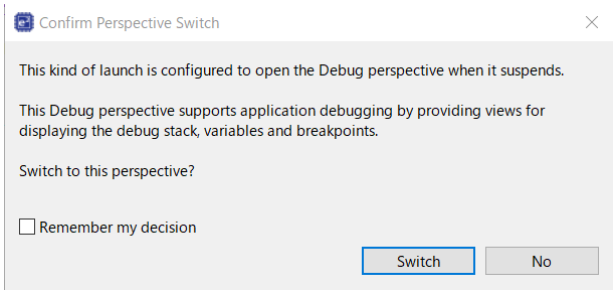


Figure 6.48. Perspective Switch

Debug タブから各 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。

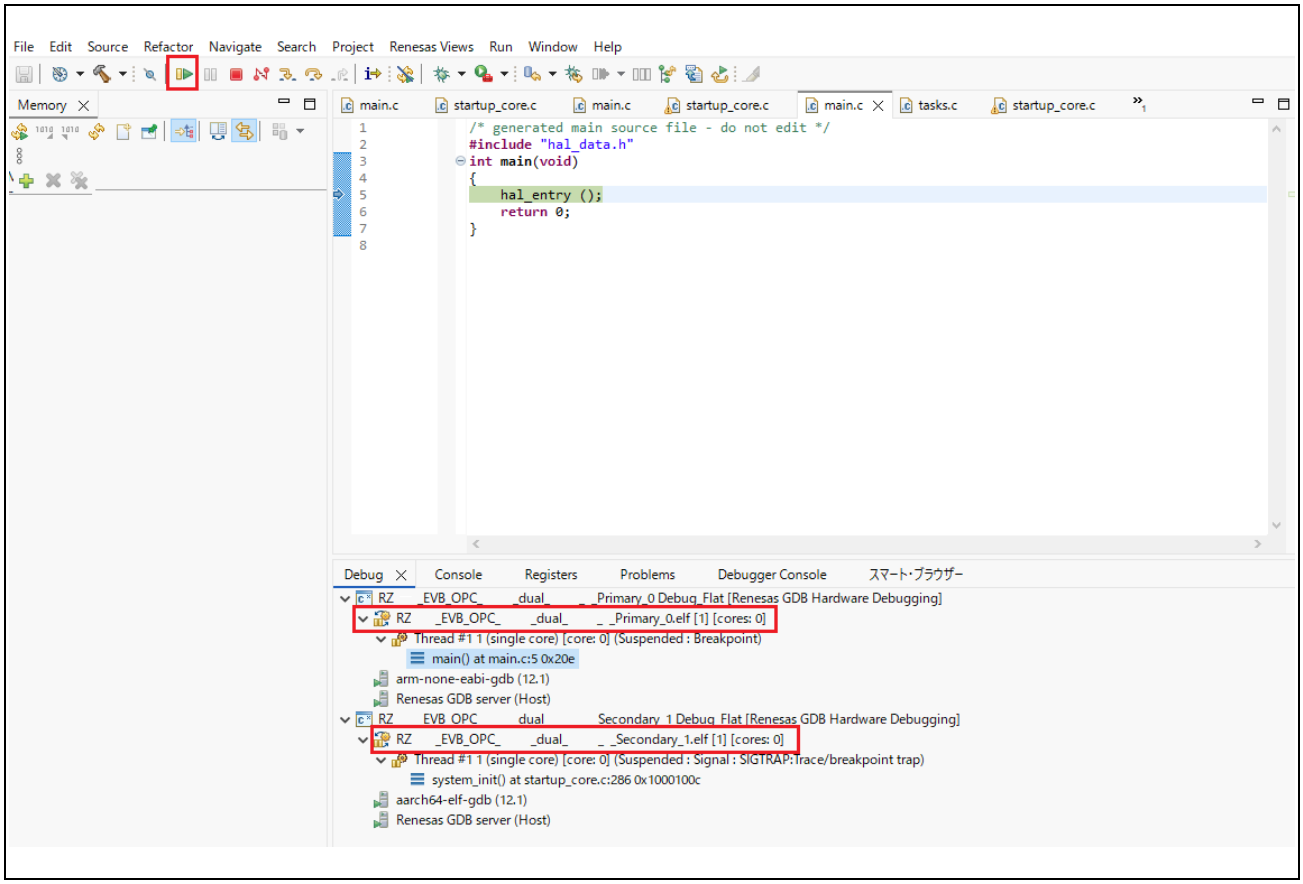


Figure 6.49. Debug

6.2.2 For Single core project

6.2.2.1 Build

※RZ/N2L で FSP v4.1.0 以降を使用時は 9.7 章のガイドを参照ください。

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

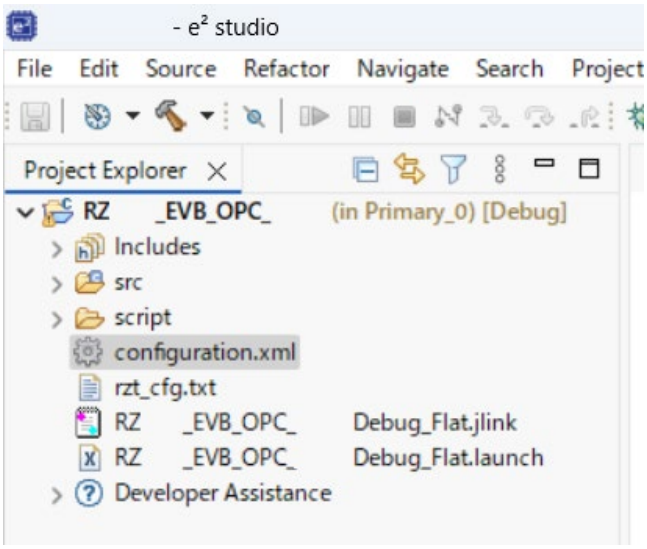


Figure 6.50. Open Smart configurator

※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、9.6.1 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。

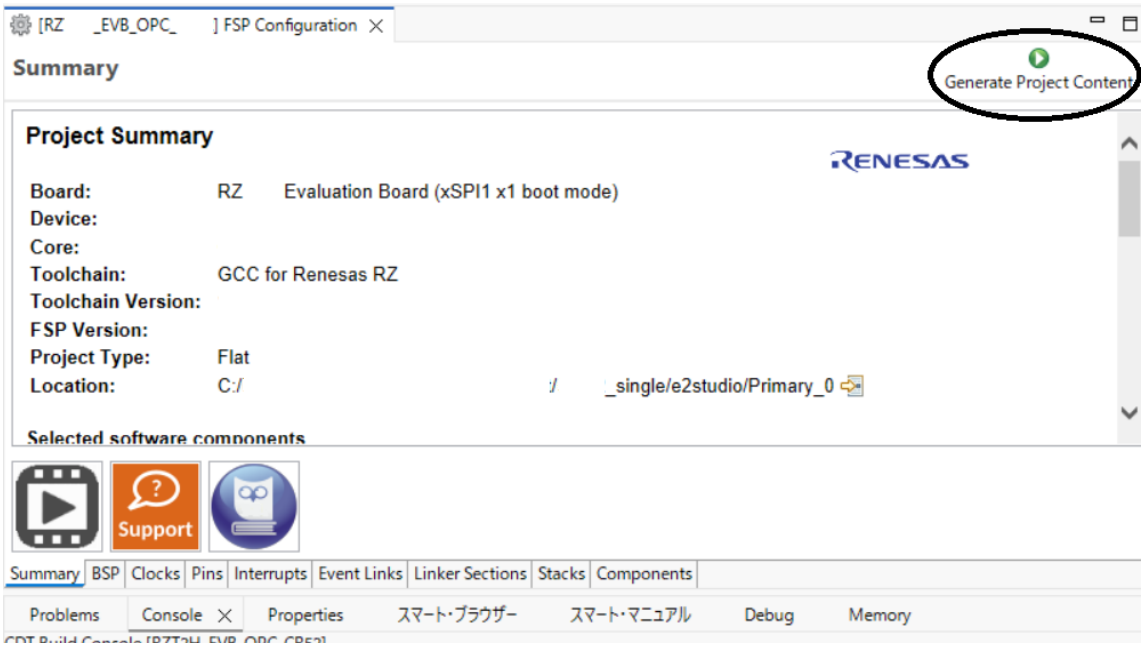


Figure 6.51. Generate Project Content

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。

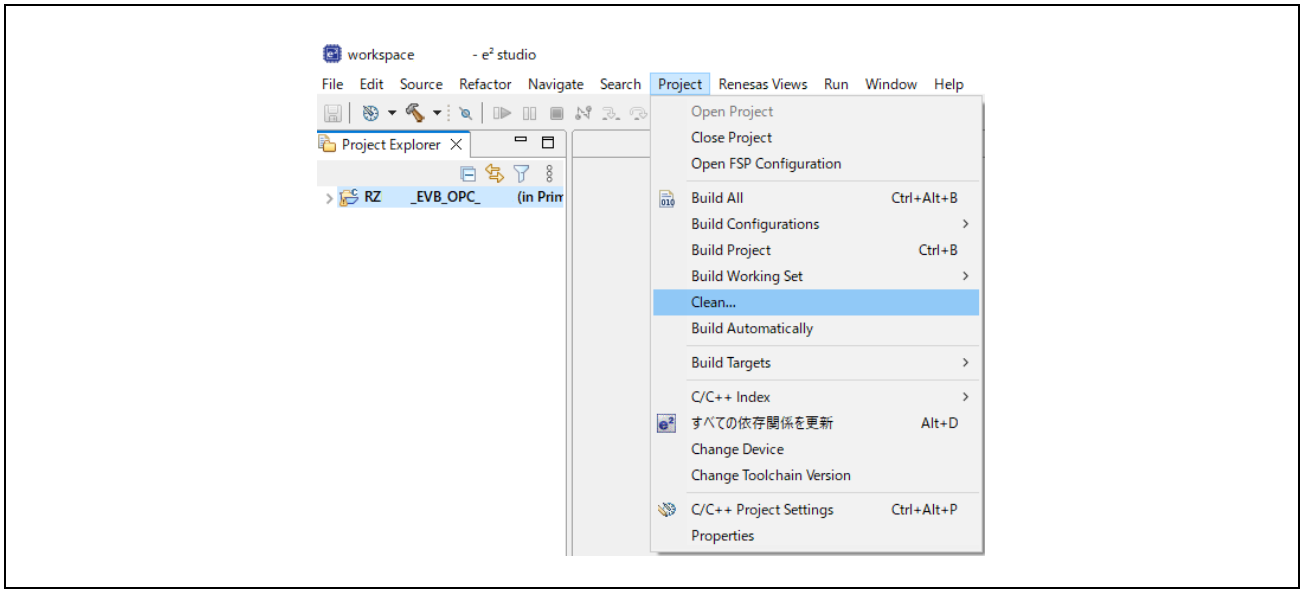


Figure 6.52. Open project Clean…

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。

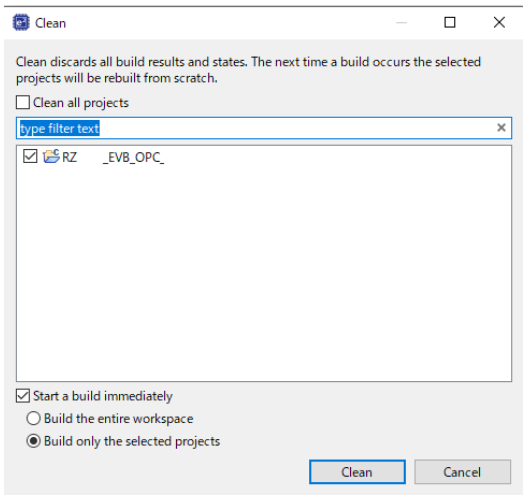


Figure 6.53. Clean and rebuild

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 Explorer 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。

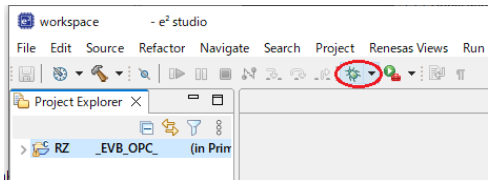


Figure 6.54. Run menu Debug As

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。

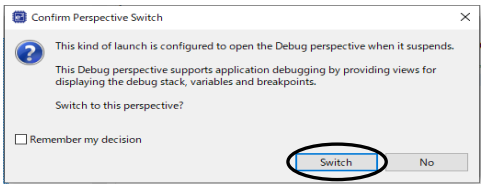


Figure 6.55. Perspective Switch

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。

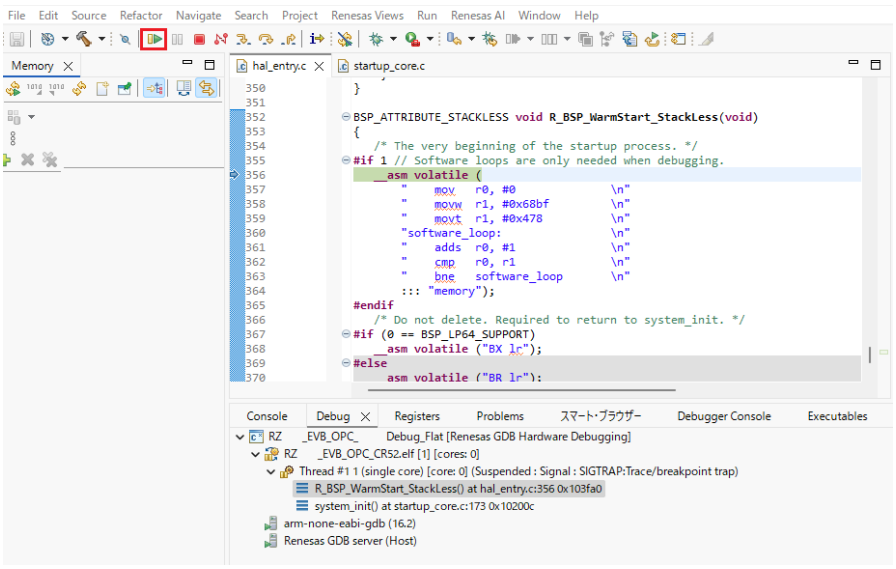


Figure 6.56. Debug view

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。

動作確認には 2.1.4 章に記載の通りにファイルを書き込みした USB フラッシュドライブを評価ボードに接続する、もしくは ENABLE_FREERTOS_PLUS_FAT:0 に設定しビルドしたソフトを使用する必要があります。

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。

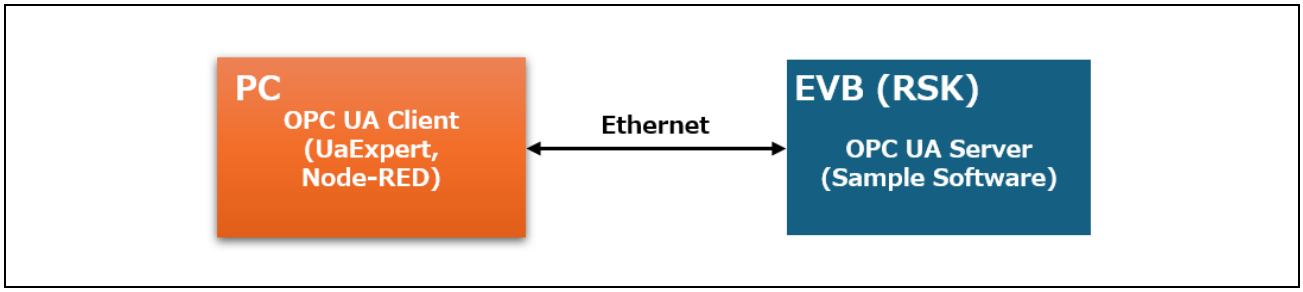


Figure 7.1. Connection Block Diagram for OPC UA Server

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と初期化処理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。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。

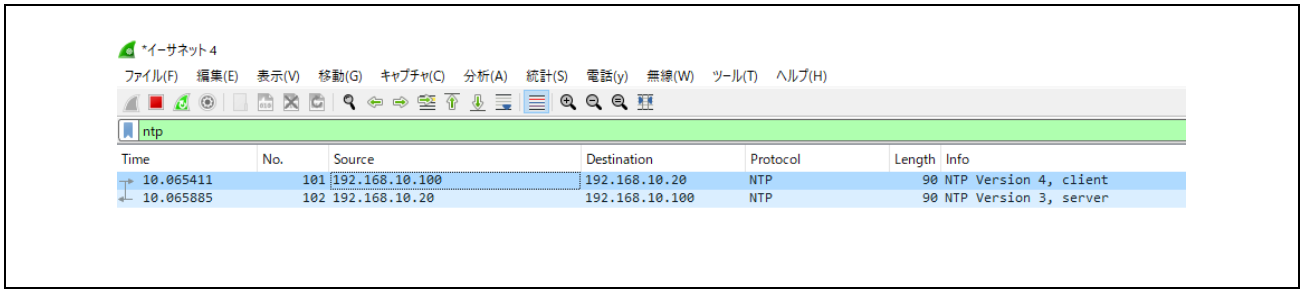


Figure 7.2. NTP Communication

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。

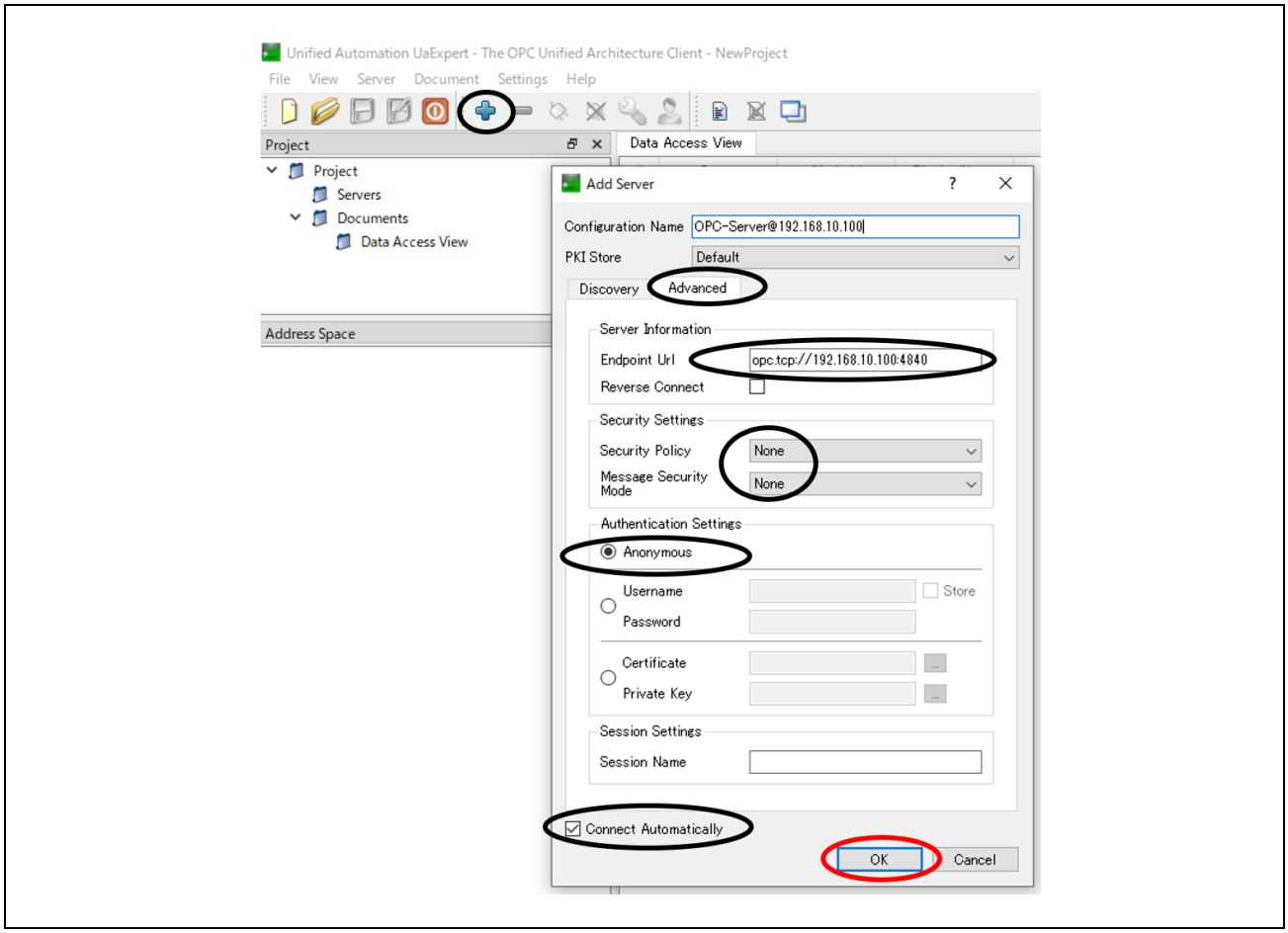


Figure 7.3. UaExpert - Add server

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。

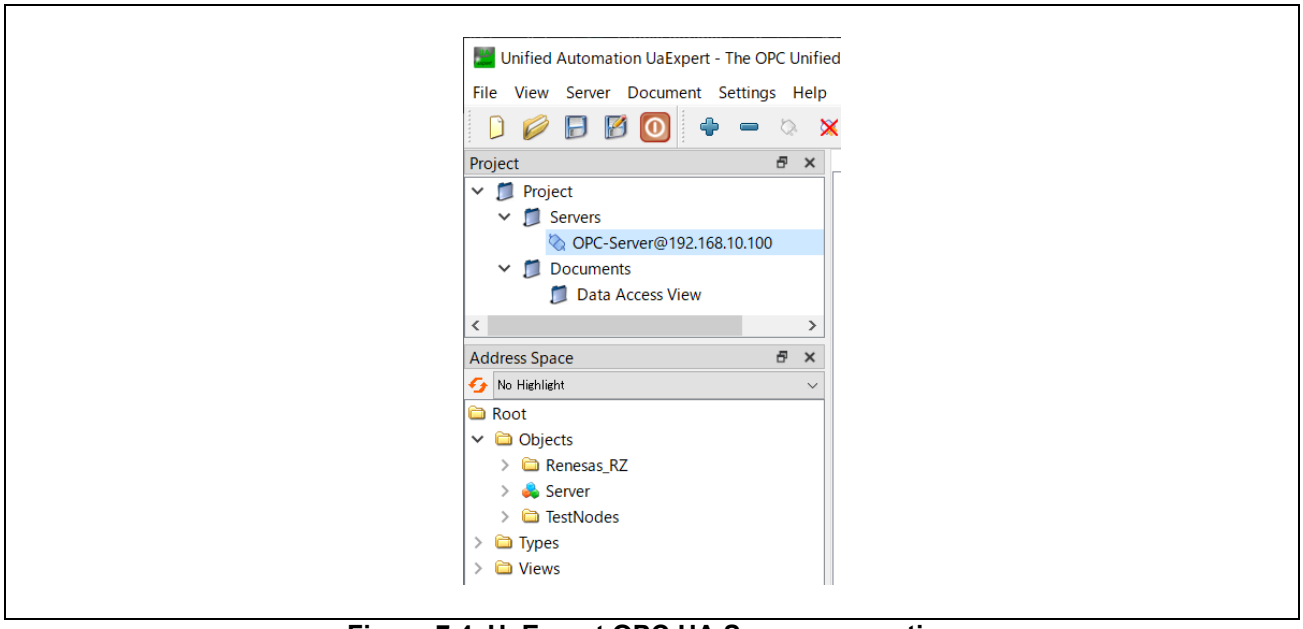


Figure 7.4. UaExpert OPC UA Server connection

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。

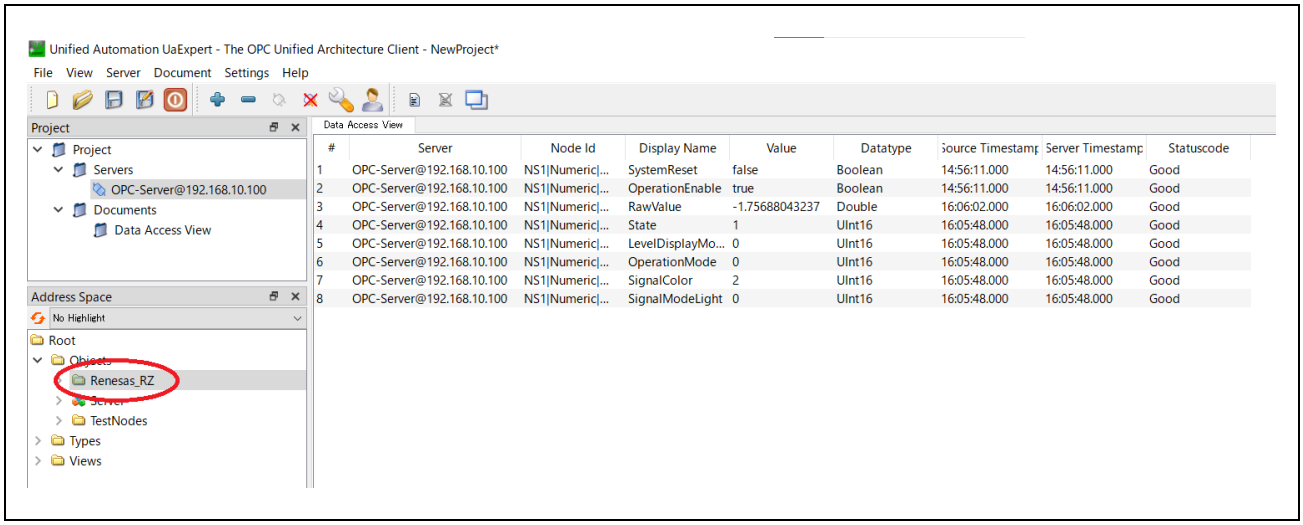


Figure 7.5. Reading Nodes

(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。

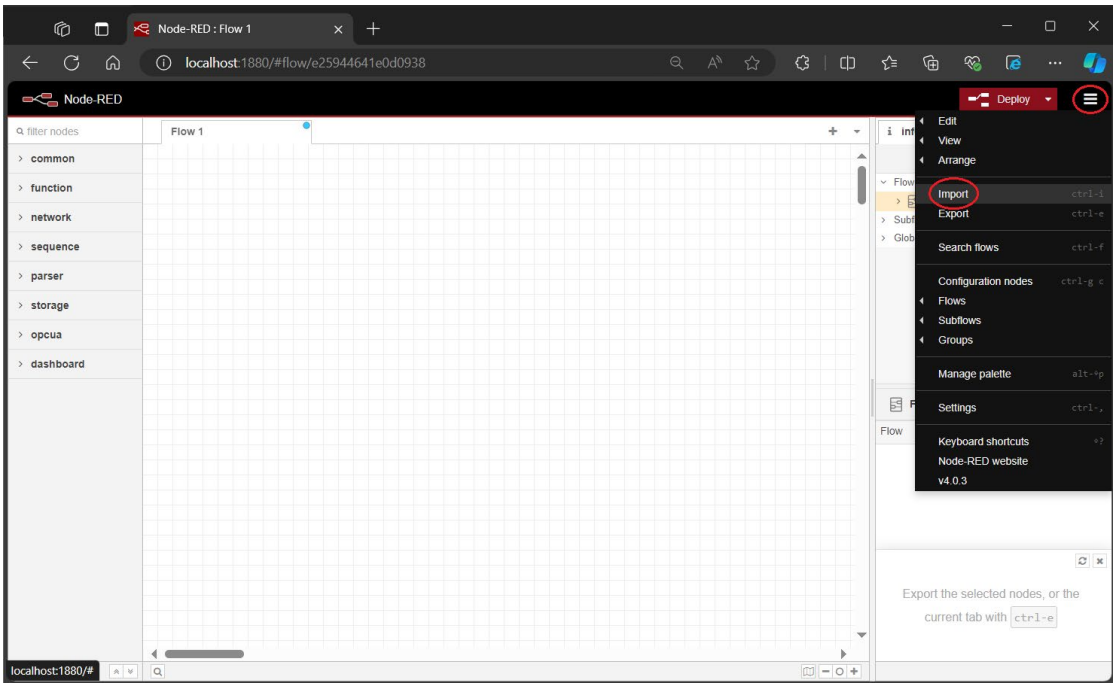


Figure 7.6. Import

“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。

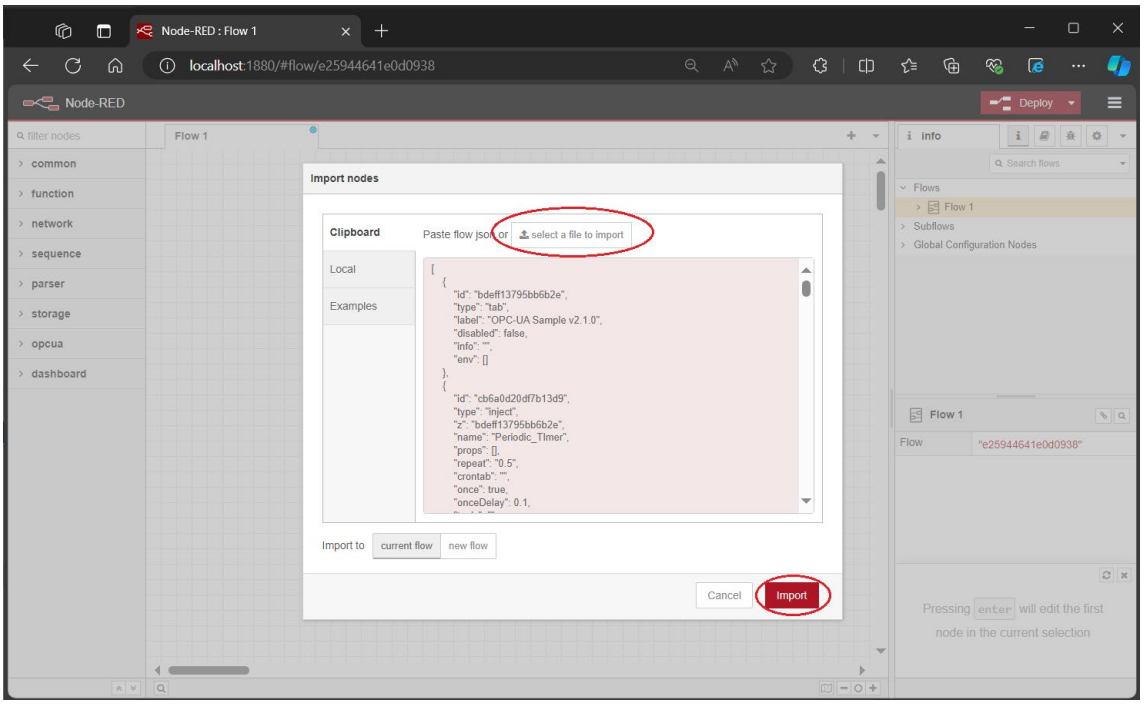


Figure 7.7. Import .json file

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。

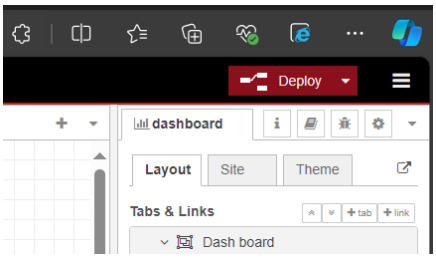


Figure 7.8. Deploy

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。

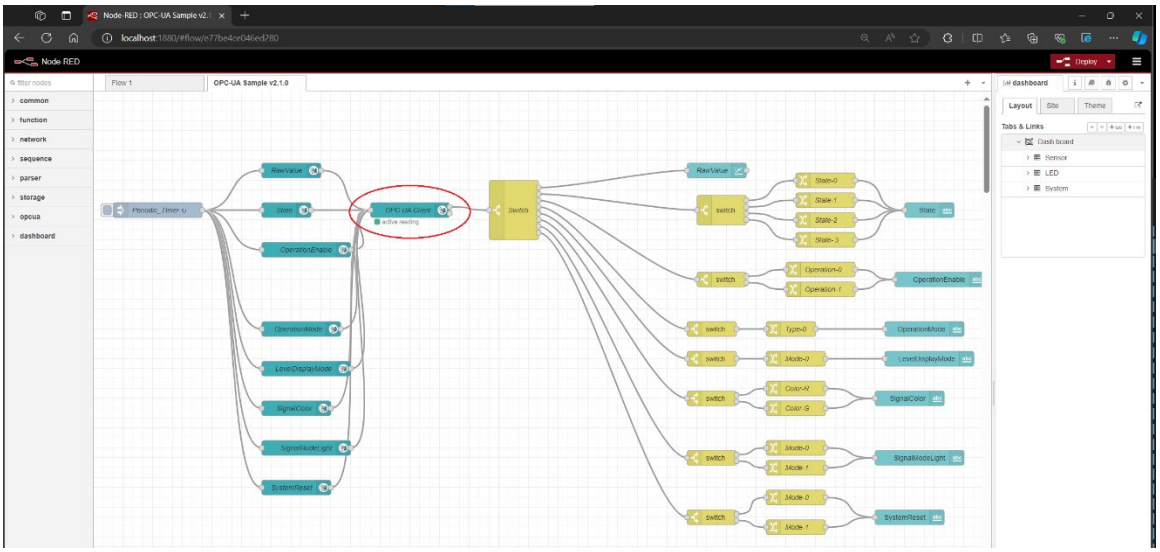


Figure 7.9. Connection Successful

画面右上の ▼ アイコンから Dashboard をクリックします。

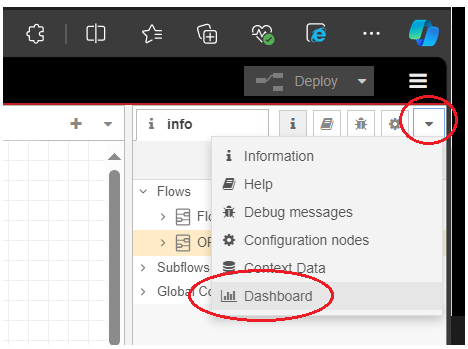


Figure 7.10. Open Dashboard (1)

画面右上の アイコンをクリックします。

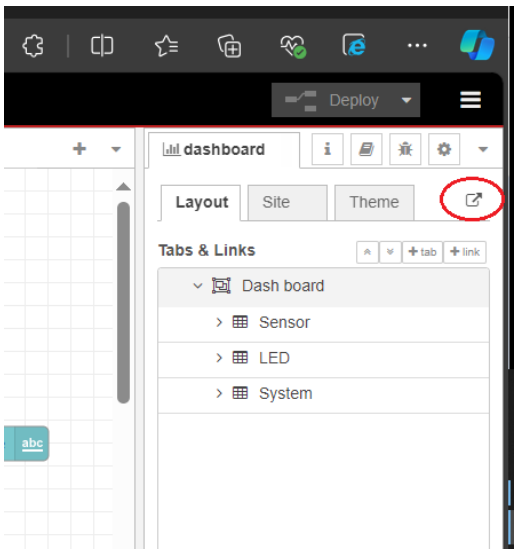


Figure 7.11. Open Dashboard (2)

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。

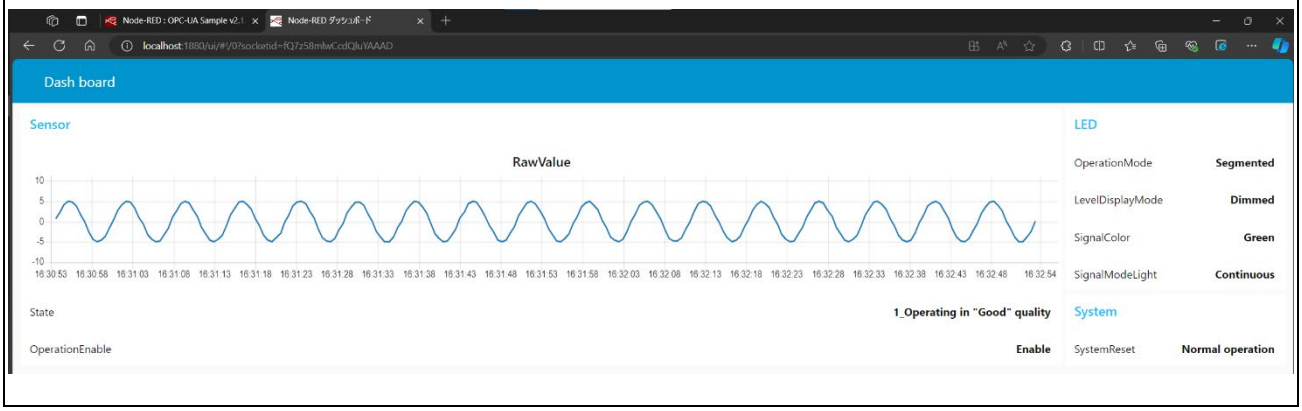


Figure 7.12. Dashboard

7.2.1.2 With security function

評価ボードの電源 ON 後、NTP 通信による時刻同期と初期化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。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。

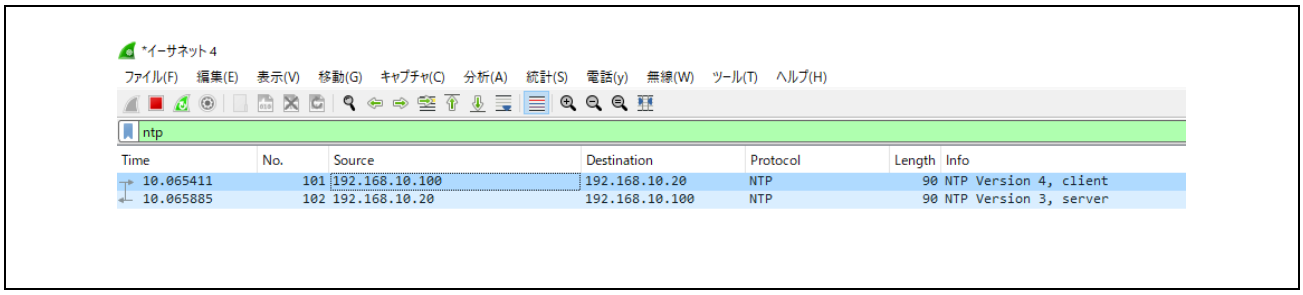


Figure 7.13. NTP Communication

パッケージに同梱されている”keyfiles.zip”から、”uaexpert.der”を PC 上の \unifiedautomation\uaexpert\PKI\own\certs フォルダに、”uaexpert_key.pem”を \unifiedautomation\uaexpert\PKI\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 も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 2.3 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/Private Key を選択した場合は、”keyfiles.zip”に含まれている”uaexpert_usr.der”と”uaexpert_key_usr.pem”を選択してください。その後、”Connect Automatically” にチェックを入れて、最後に OK をクリックします。

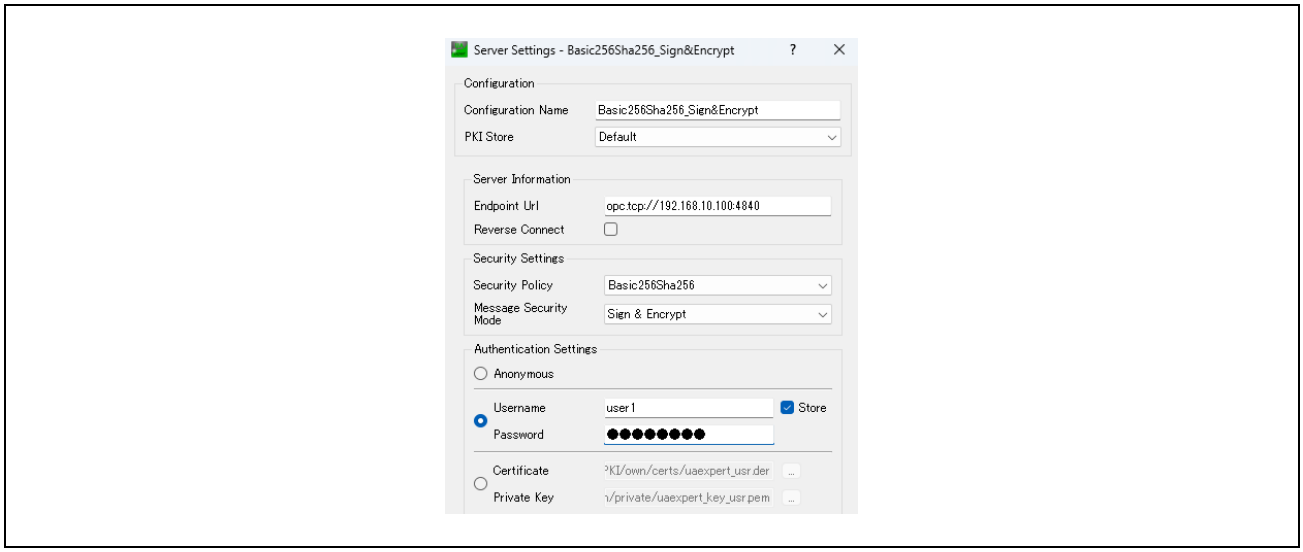


Figure 7.14. UaExpert Add server

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。

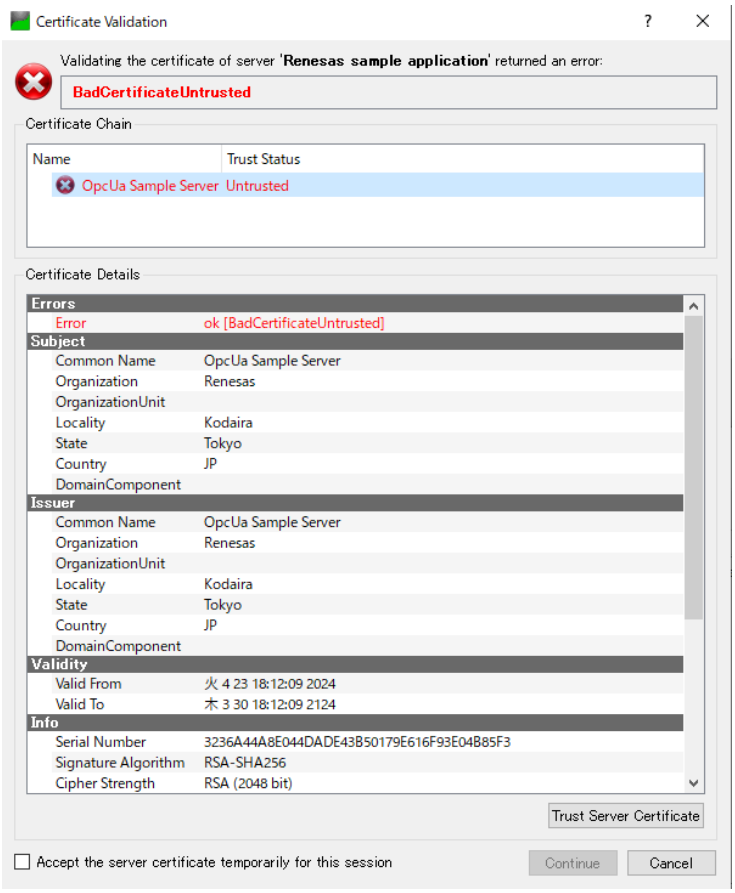


Figure 7.15. Certificate Validation

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。

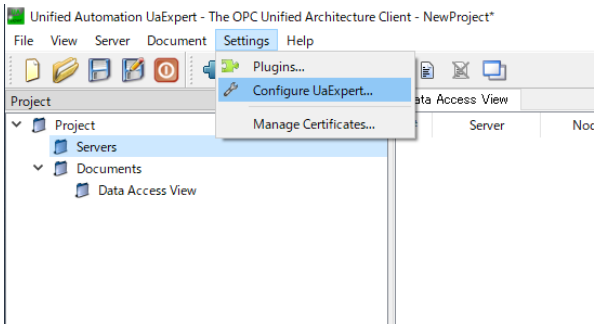


Figure 7.16. Configure UaExpert

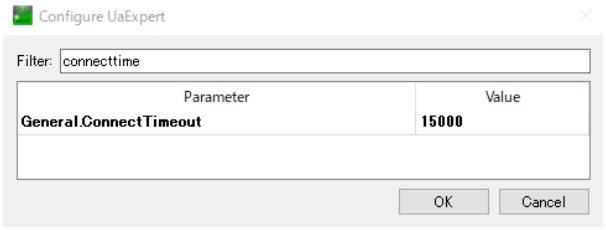


Figure 7.17. UaExpert ConnectTimeout Setting

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。

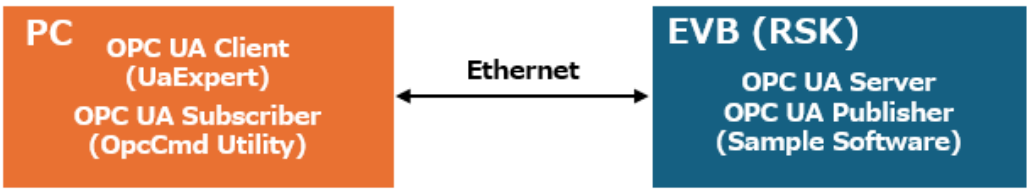


Figure 7.18. Connection Block Diagram for OPC UA Publisher

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)"
```

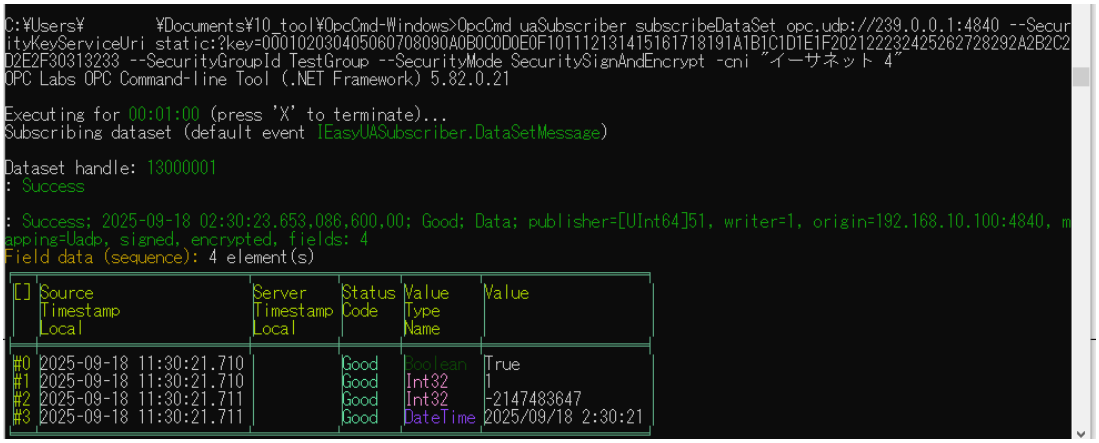


Figure 7.19. Command Execution Result

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。

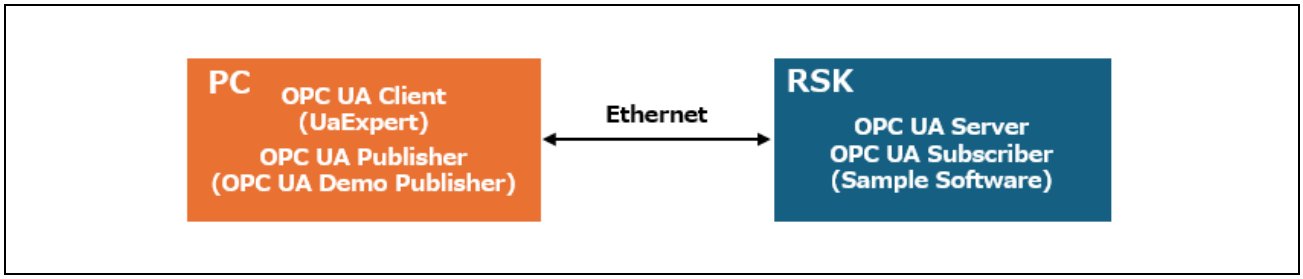


Figure 7.20. Connection Block Diagram for OPC UA Subscriber

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.udp://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.udp://239.0.0.1 -ewg DynamicLayoutGroup -edsw SimpleWriter --SecurityKeyServiceUri static:?key=000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F30313233 --SecurityGroupId TestGroup --SecurityMode SignAndEncrypt -cni "(ネットワークインターフェース名)"

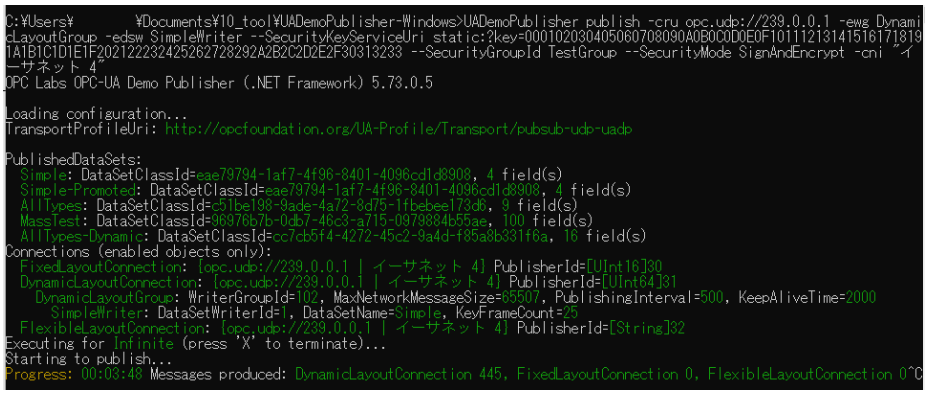


Figure 7.21. Command Execution Result

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

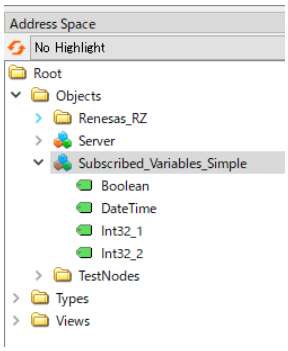


Figure 7.22. “Subscribed_Variables_Simple” Node

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。

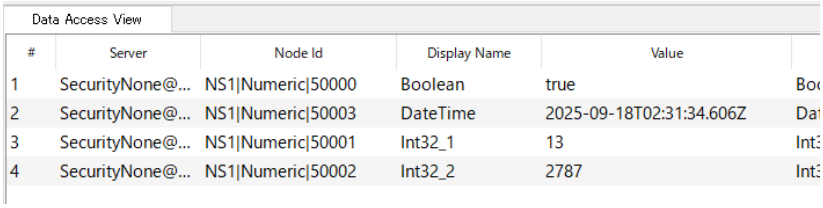


Figure 7.23. Check the Node Values

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。

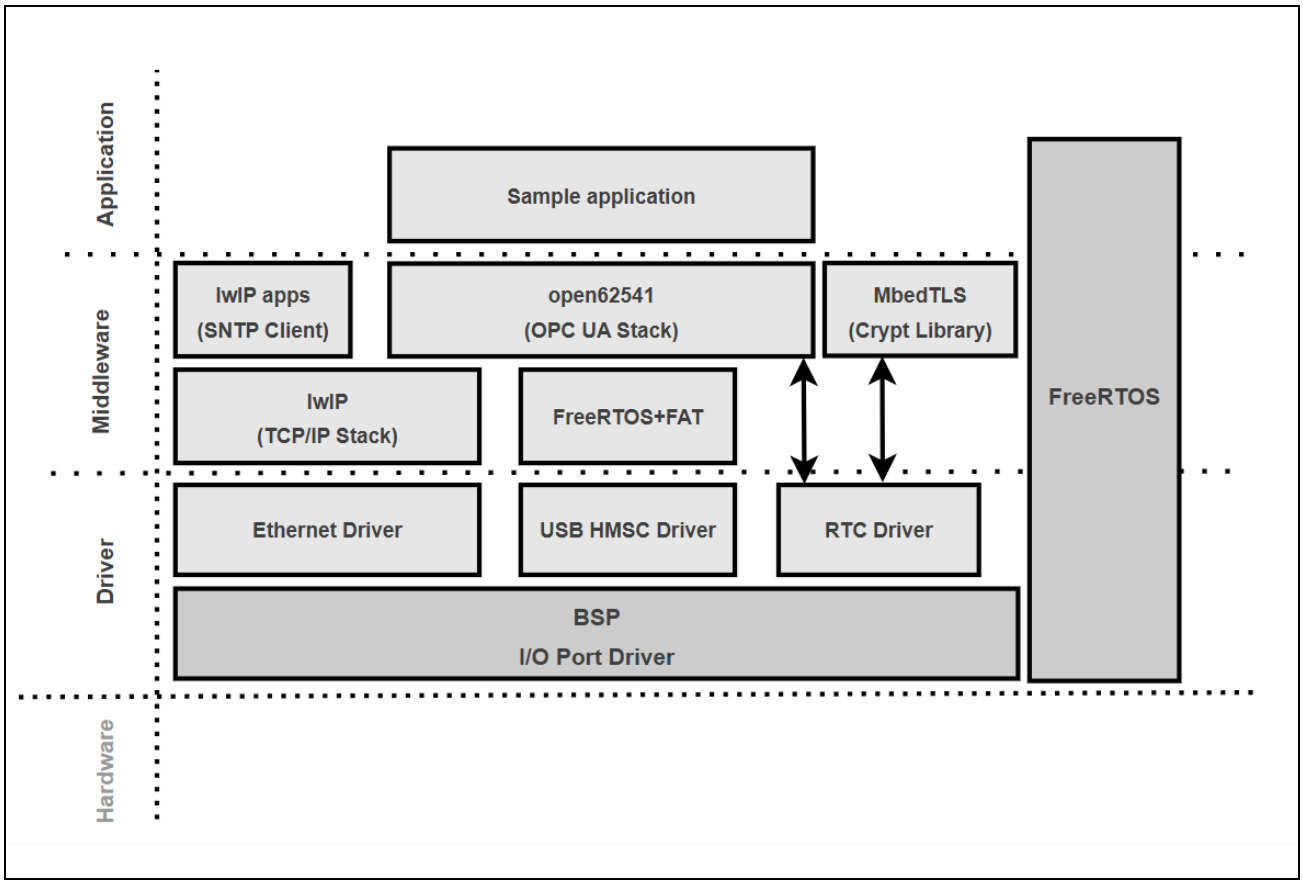


Figure 8.1. Software Architecture Diagram

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 (Cryptographic 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。使用しているバージョンは v3.6.6 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 のベースバージョンは以下です。

Base Version ：v1.3.18

(Commit db393f2c94374ee55d89ef46e5163686932222ee)

・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 のコンフィグレーションと情報モデル NodeSet のコンパイルを実施しています。詳細は、Appendix の 9.1 章をご参照ください。

open62541 Documentation — 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)

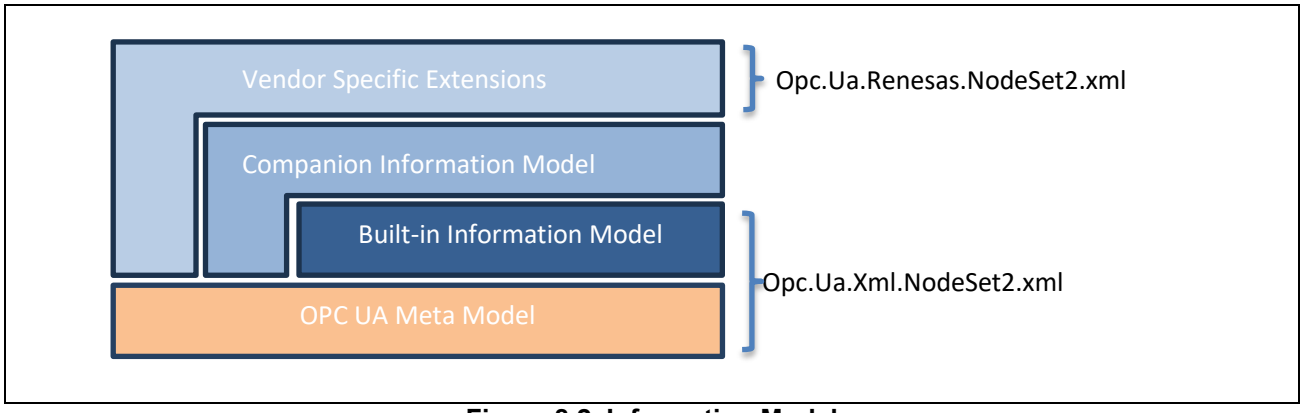


Figure 8.2. Information Model

・Opc.Ua.Xml.NodeSet2.xml

OPC UA 情報モデルを記述する為のルール(メタモデル) とメタモデルによって定義されている OPC UA の基本的な情報モデル(Built-in Information Model) が記述されています。

Opc.Ua.Xml.NodeSet2.xml (UA-Nodeset/XML/Opc.Ua.Xml.NodeSet2.xml at d1bb6a22125bd7cd986272b1ee98a18a91d76fff · OPCFoundation/UA-Nodeset · GitHub)

・Opc.Ua.Renesas.NodeSet2.xml

本サンプル用に作成した情報モデルです。作成には、OPC UA 情報モデル用エディタである SiOME を使用しています。詳細は下記 Link をご参照ください。

Siemens OPC UA Modeling Editor (SiOME) - ID: 109755133 - Industry Support Siemens

Renesas 情報モデルは Figure 8.3 の構成となっており、各 Node の読み書きは 8.1.2.5 章に記載された通りに行われます。

Folder	Node	Type	NodeId	DataType	array	AccessLevel	Mearning
Renesas_RZ		FolderType	ns=1, i=5001	-	-	-	-
LED		FolderType	ns=1, i=5002	-	-	-	-
	OperationMode	BaseDataVariableType	ns=1, i=6013	UInt16	N	Read	0:Segmented (Used as stack of individual lights) 1:Levelmeter (Used as level meter) 2:Running_Light (The whole stack acts as a running light) 3:Other 4~:NotUsed
	LevelDisplayMode	BaseDataVariableType	ns=1, i=6014	UInt16	N	Read	0:Dimmed (Uses dimming to display fractions.) 1:Blinking (Uses blinking to display fractions.) 2:Other 3~:NotUsed
	SignalColor	BaseDataVariableType	ns=1, i=6008	UInt16	N	Read	0:Off 1:Red 2:Green 3:Blue 4:Yellow 5:Purple 6:Cyan 7:White 8~:NotUsed
	SignalModeLight	BaseDataVariableType	ns=1, i=6009	UInt16	N	Read	0:Continuous 1:Blinking 2:Flashing 3~:NotUsed
Sensor		FolderType	ns=1, i=5003	-	-	-	-
	RawValue	BaseDataVariableType	ns=1, i=6010	Double	N	Read	-
	State	BaseDataVariableType	ns=1, i=6011	UInt16	N	Read	0:Before Operation 1:Operating in "Good" quality 2:Operating in "Bad" quality 3:Suspended Operation 4~:NotUsed
	OperationEnable	BaseDataVariableType	ns=1, i=6012	Boolean	N	Read&Write	0:Disable 1:Enable
System		FolderType	ns=1, i=5004	-	-	-	-
	SystemReset	BaseDataVariableType	ns=1, i=6015	Boolean	N	Read&Write	0:Normal operation 1:Execute System Reset

Figure 8.3. Renesas 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。

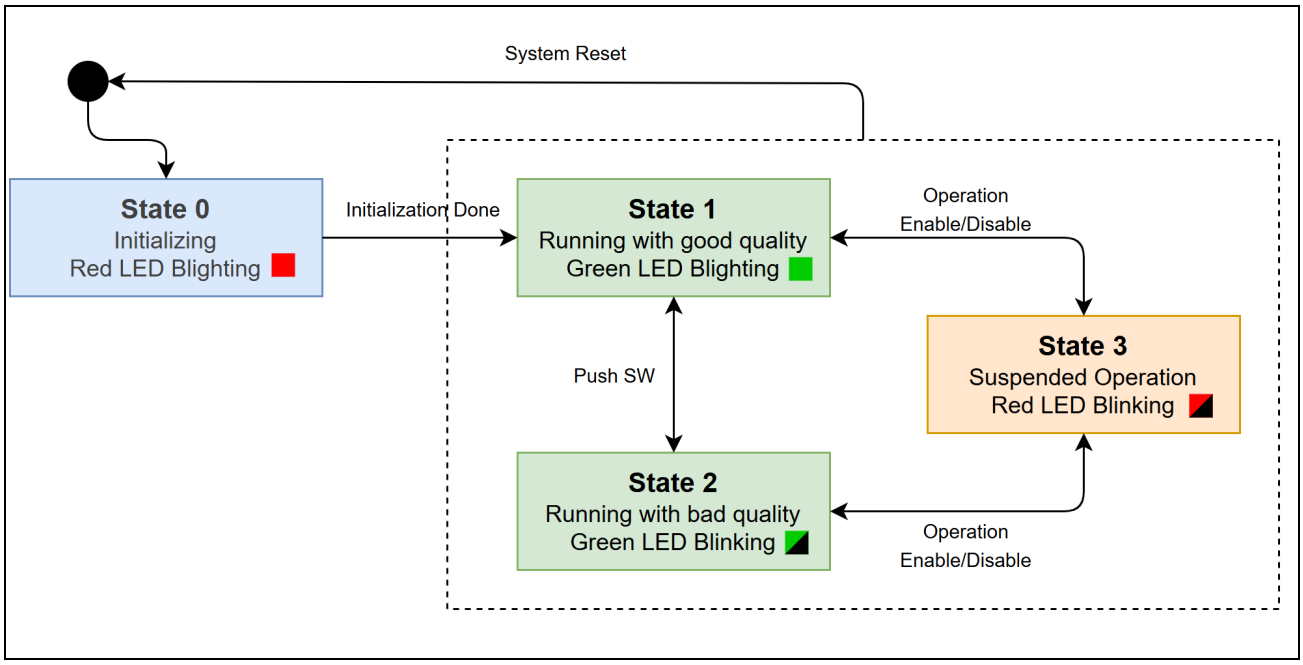


Figure 8.4. State Transition

また、”SystemReset” Node に”1:Execute SystemReset”を書き込まれると、”Status” Node (i=2259) が 4:Shutdown になり、”SecondsTillShutdown” Node (i=2992) で 15 秒のカウントダウンが始まります。その後、値が 0 にな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。

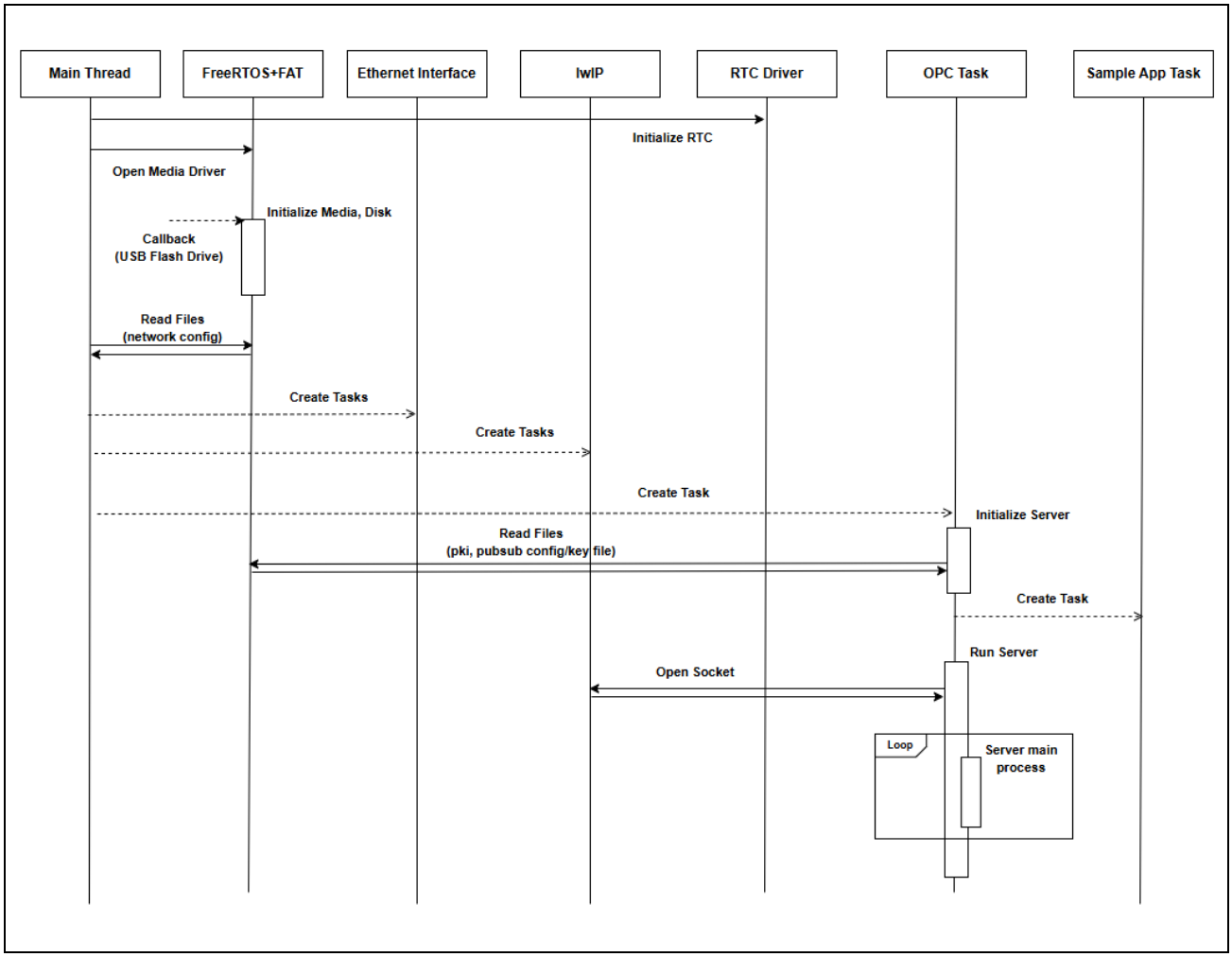


Figure 8.5. Startup

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。

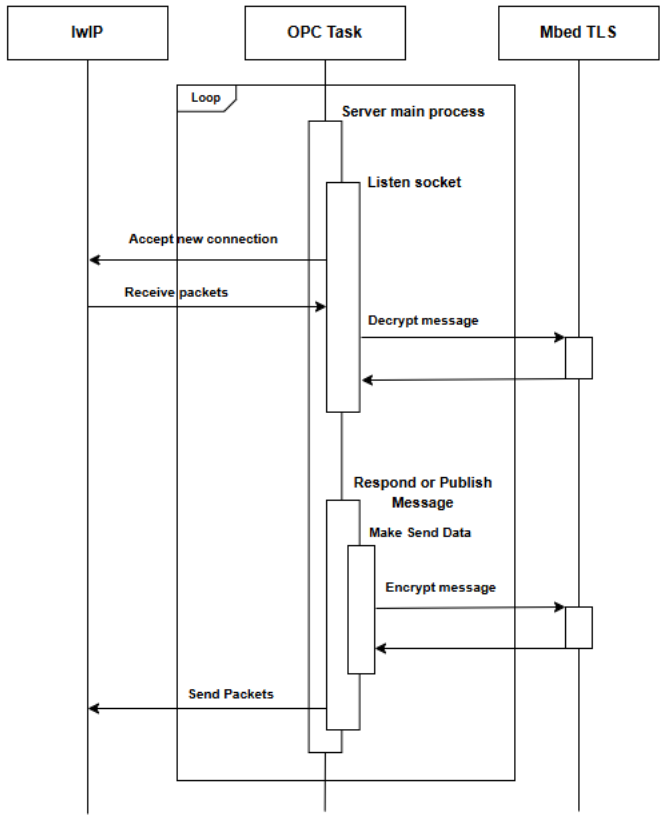


Figure 8.6. OPC UA communication

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。

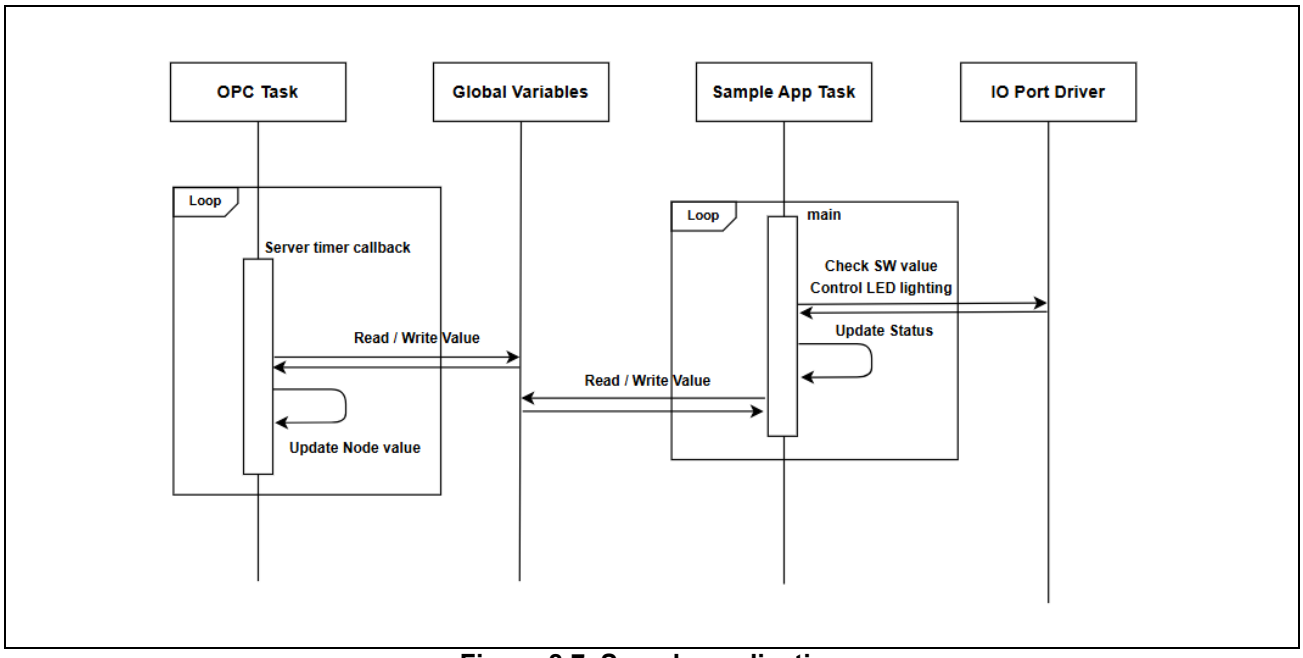


Figure 8.7. Sample application

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。

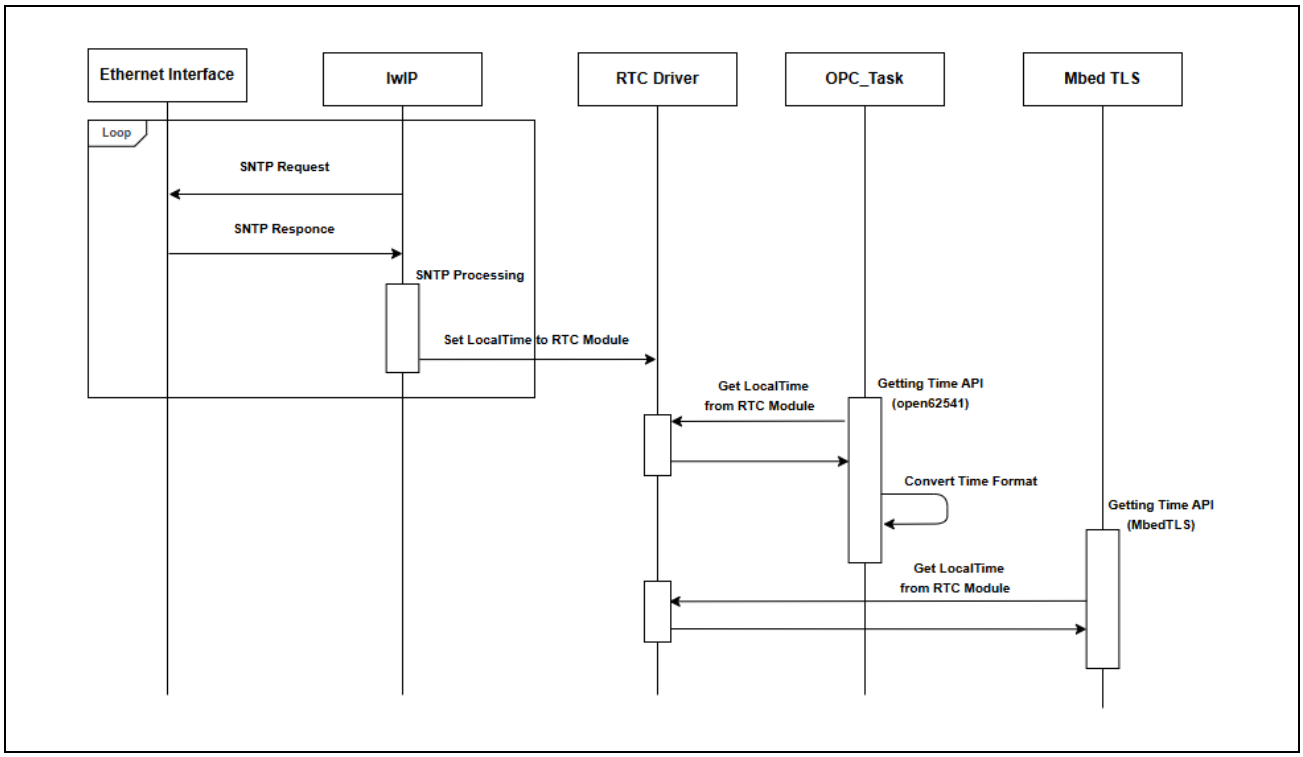


Figure 8.8. Time synchronization

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

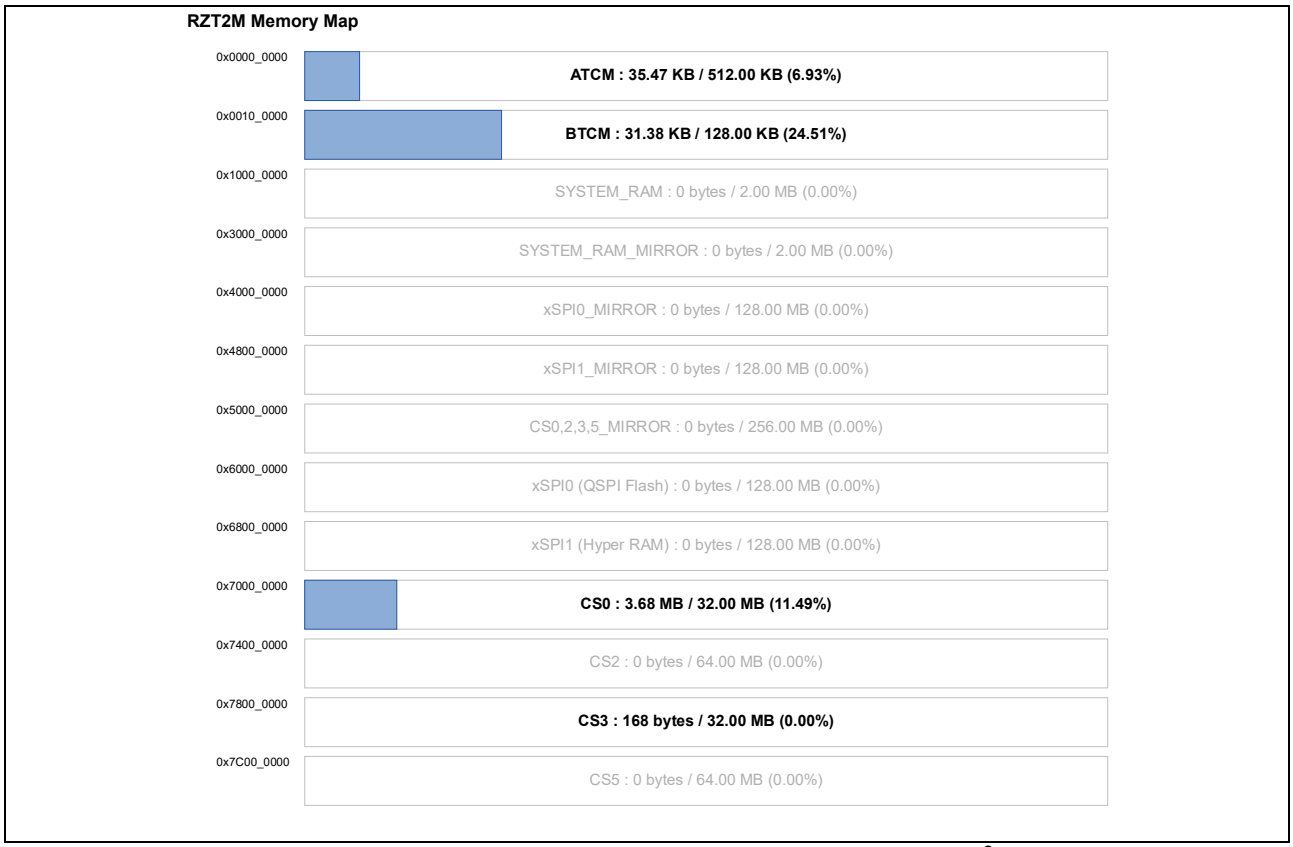
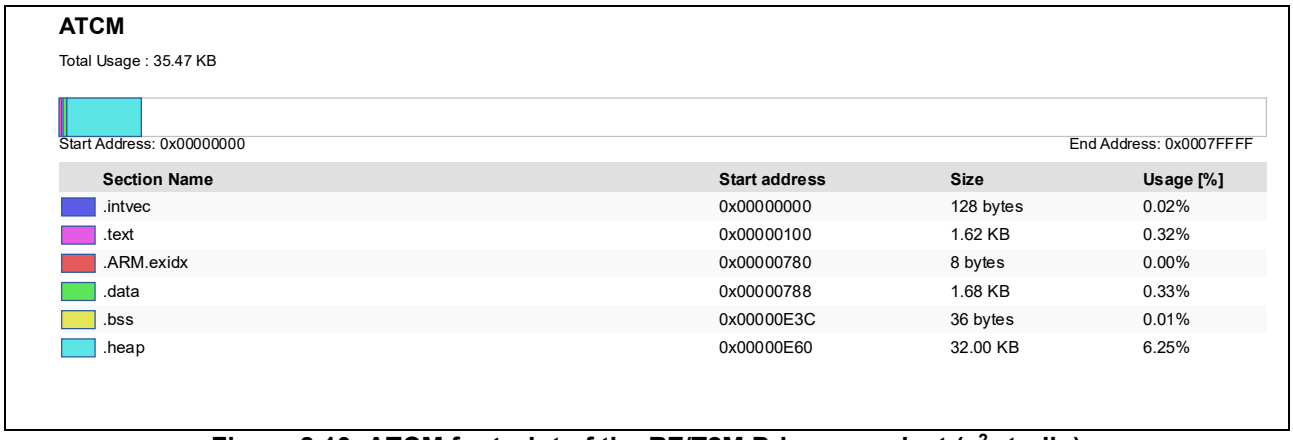
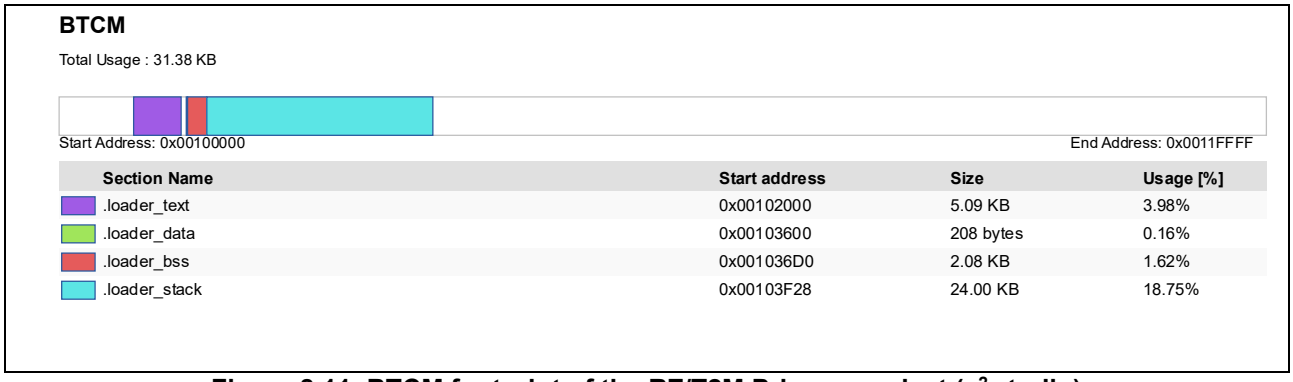


Figure 8.9. Memory footprint of the RZ/T2M Primary project (e² studio)



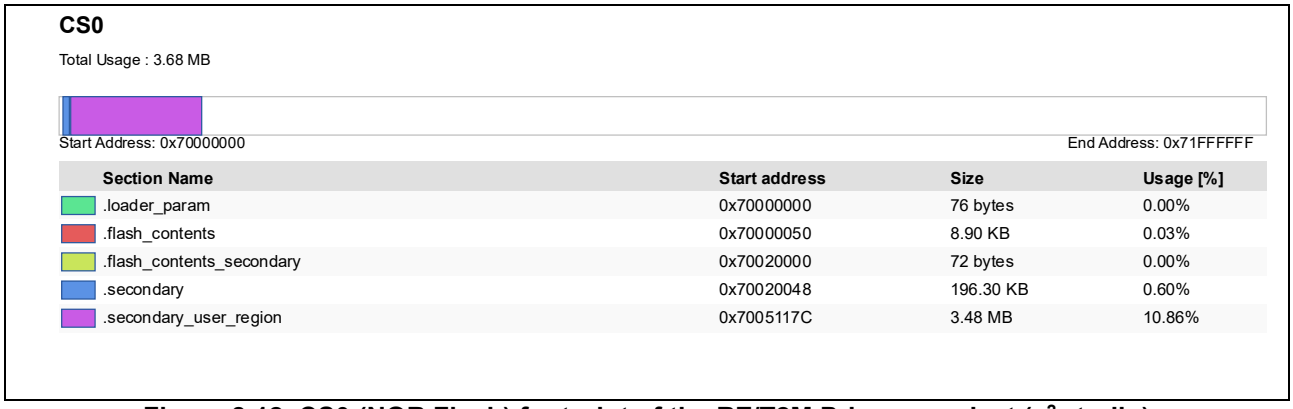
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.text	0x00000100	1.62 KB	0.32%
.ARM.exidx	0x00000780	8 bytes	0.00%
.data	0x00000788	1.68 KB	0.33%
.bss	0x00000E3C	36 bytes	0.01%
.heap	0x00000E60	32.00 KB	6.25%

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



Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	5.09 KB	3.98%
.loader_data	0x00103600	208 bytes	0.16%
.loader_bss	0x001036D0	2.08 KB	1.62%
.loader_stack	0x00103F28	24.00 KB	18.75%

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



Section Name	Start address	Size	Usage [%]
.loader_param	0x70000000	76 bytes	0.00%
.flash_contents	0x70000050	8.90 KB	0.03%
.flash_contents_secondary	0x70020000	72 bytes	0.00%
.secondary	0x70020048	196.30 KB	0.60%
.secondary_user_region	0x7005117C	3.48 MB	10.86%

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



Section Name	Start address	Size	Usage [%]
.user_region	0x78000000	168 bytes	0.00%

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

(2) Secondary project

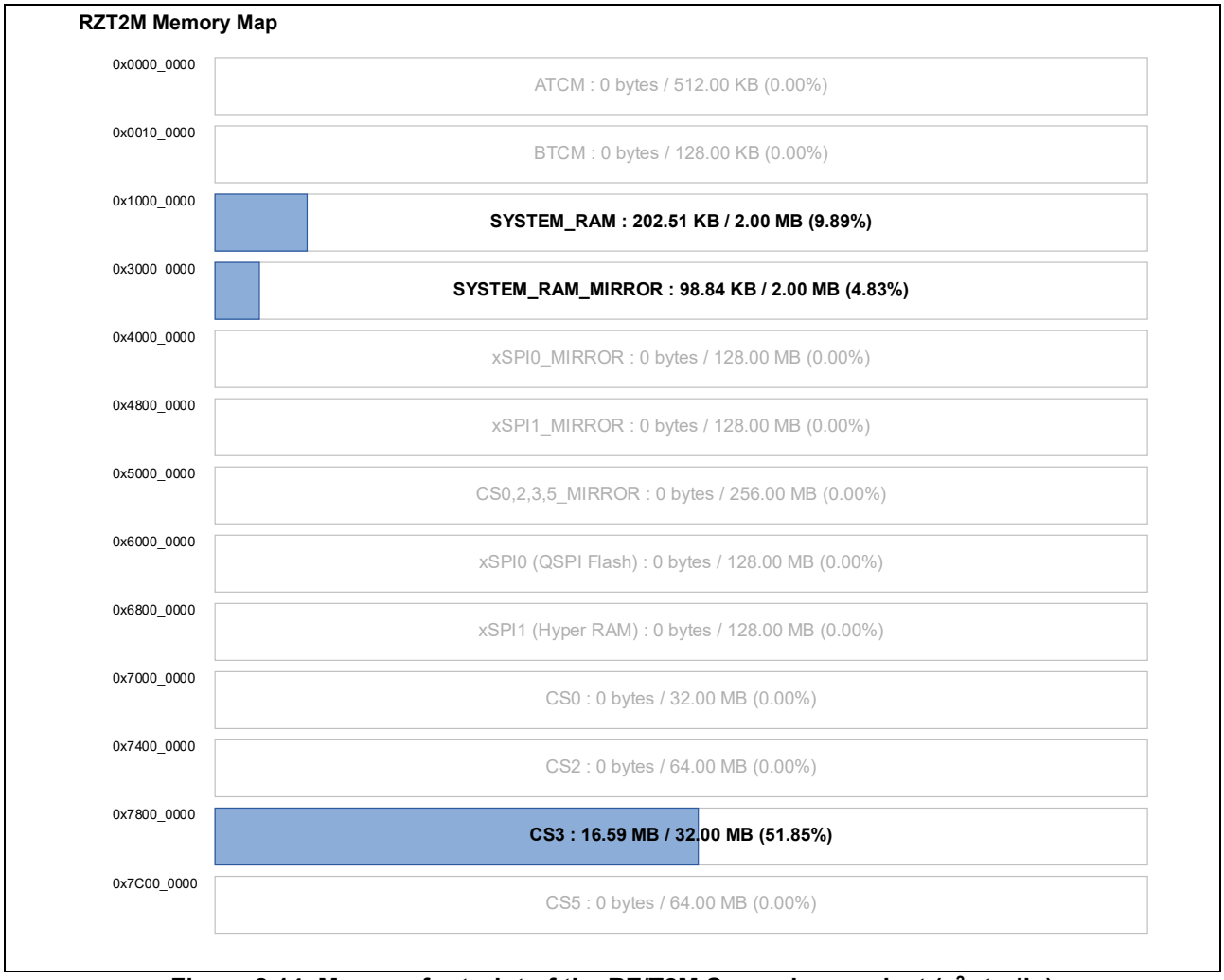
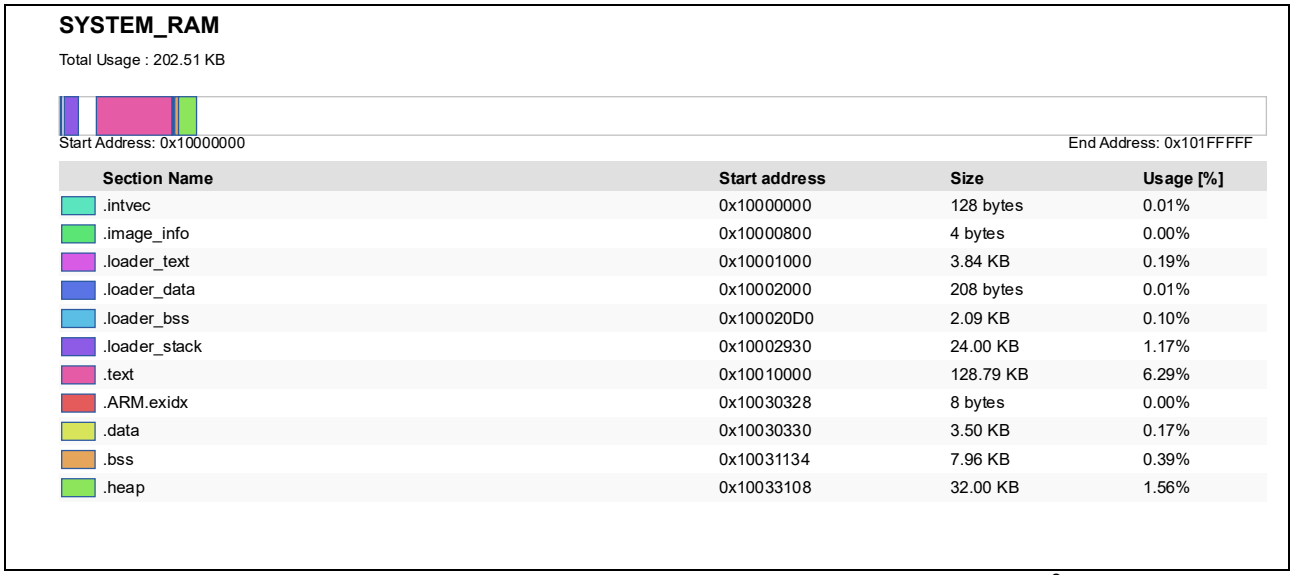


Figure 8.14. Memory footprint of the RZ/T2M Secondary project (e² studio)



Section Name	Start address	Size	Usage [%]
.intvec	0x10000000	128 bytes	0.01%
.image_info	0x10000800	4 bytes	0.00%
.loader_text	0x10001000	3.84 KB	0.19%
.loader_data	0x10002000	208 bytes	0.01%
.loader_bss	0x100020D0	2.09 KB	0.10%
.loader_stack	0x10002930	24.00 KB	1.17%
.text	0x10010000	128.79 KB	6.29%
.ARM.exidx	0x10030328	8 bytes	0.00%
.data	0x10030330	3.50 KB	0.17%
.bss	0x10031134	7.96 KB	0.39%
.heap	0x10033108	32.00 KB	1.56%

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)

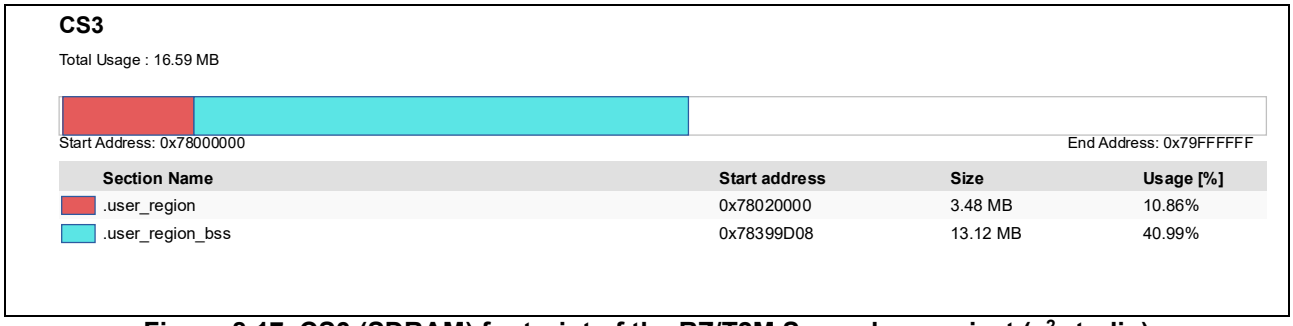


Figure 8.17. CS3 (SDRAM) footprint of the RZ/T2M Secondary project (e² studio)

8.3.1.2 EWARM

(1) Primary project

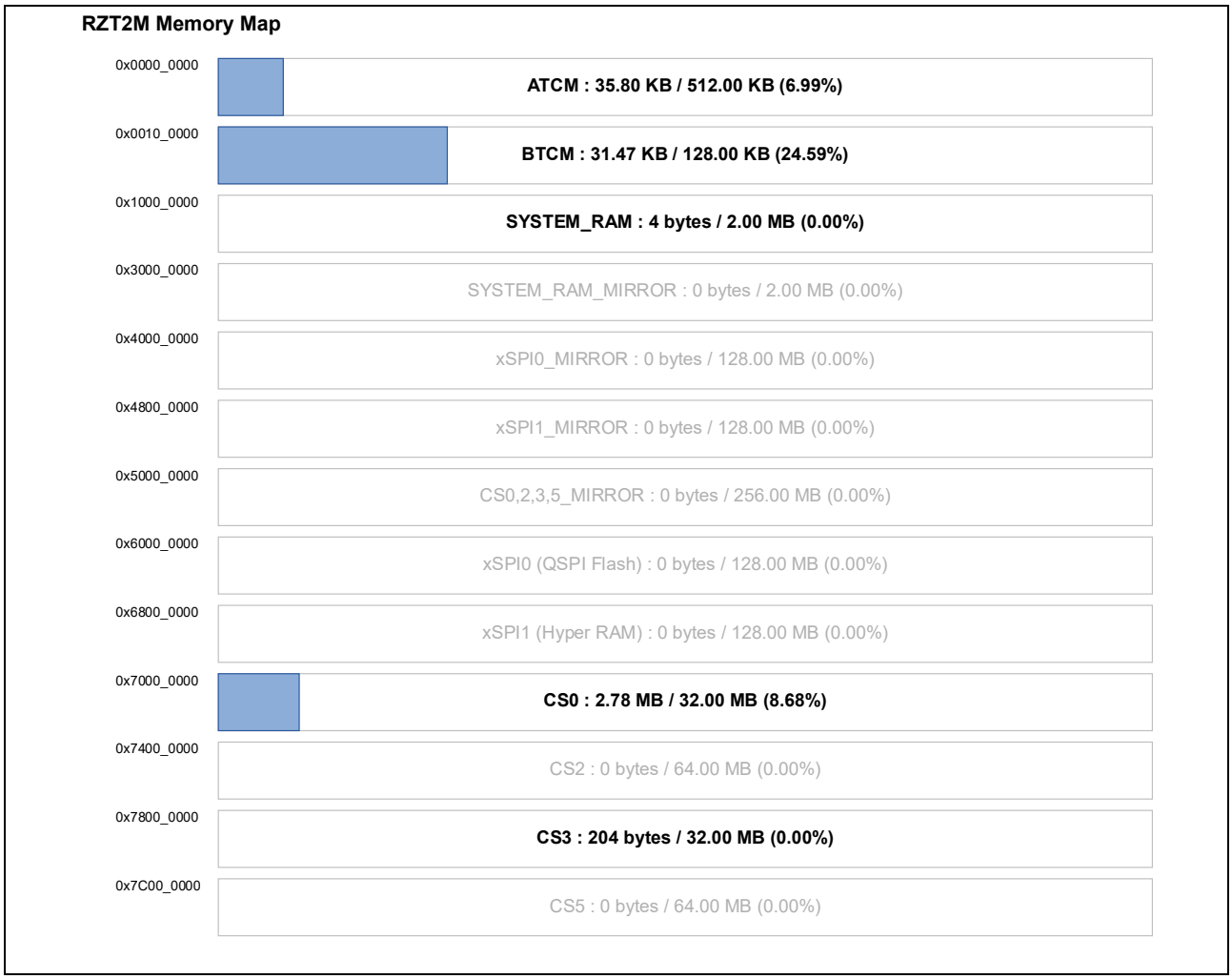
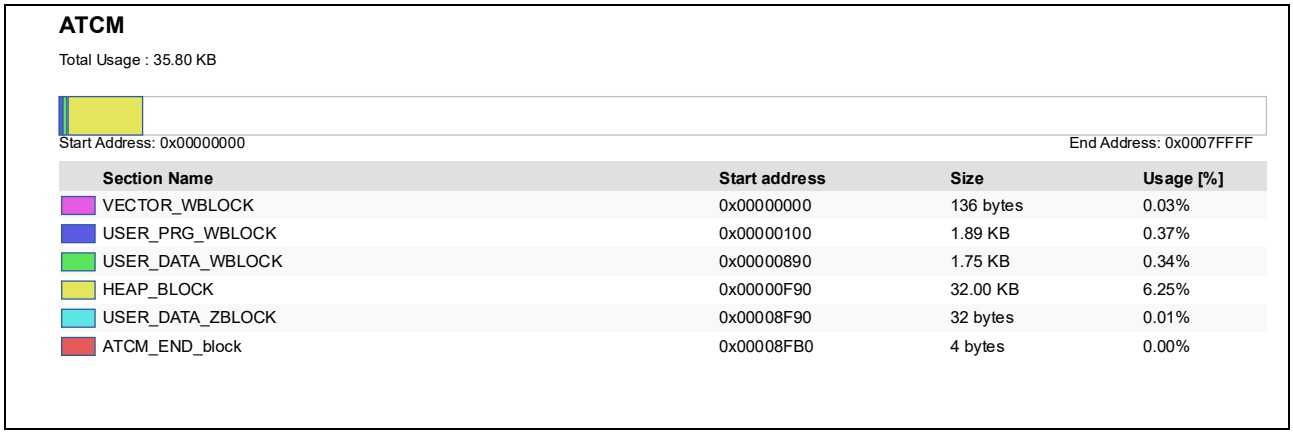
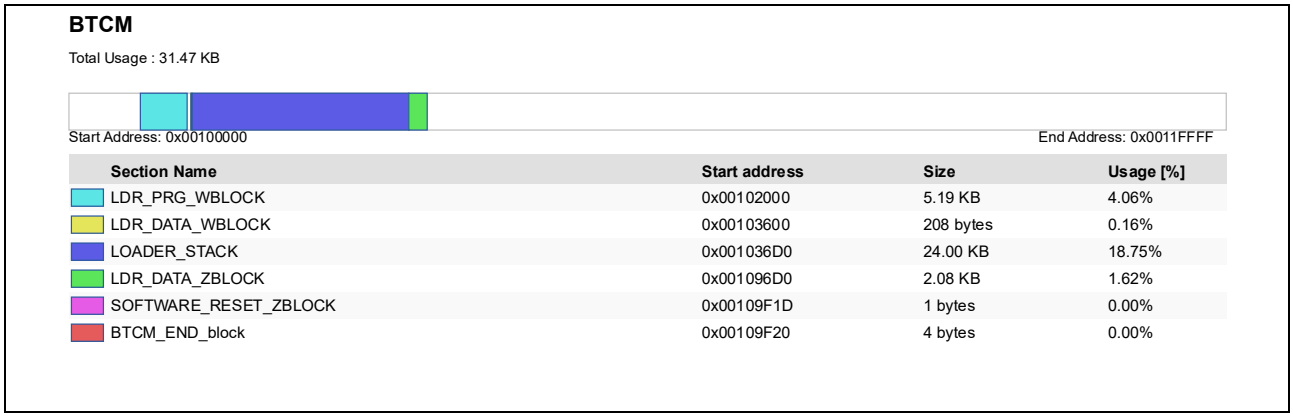


Figure 8.18. Memory footprint of the RZ/T2M Primary project (EWARM)



Section Name	Start address	Size	Usage [%]
VECTOR_WBLOCK	0x00000000	136 bytes	0.03%
USER_PRG_WBLOCK	0x00000100	1.89 KB	0.37%
USER_DATA_WBLOCK	0x00000890	1.75 KB	0.34%
HEAP_BLOCK	0x00000F90	32.00 KB	6.25%
USER_DATA_ZBLOCK	0x00008F90	32 bytes	0.01%
ATCM_END_block	0x00008FB0	4 bytes	0.00%

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



BTCM

Total Usage : 31.47 KB

Start Address: 0x00100000 End Address: 0x0011FFFF

Section Name	Start address	Size	Usage [%]
LDR_PRG_WBLOCK	0x00102000	5.19 KB	4.06%
LDR_DATA_WBLOCK	0x00103600	208 bytes	0.16%
LOADER_STACK	0x001036D0	24.00 KB	18.75%
LDR_DATA_ZBLOCK	0x001096D0	2.08 KB	1.62%
SOFTWARE_RESET_ZBLOCK	0x00109F1D	1 bytes	0.00%
BTCM_END_block	0x00109F20	4 bytes	0.00%

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



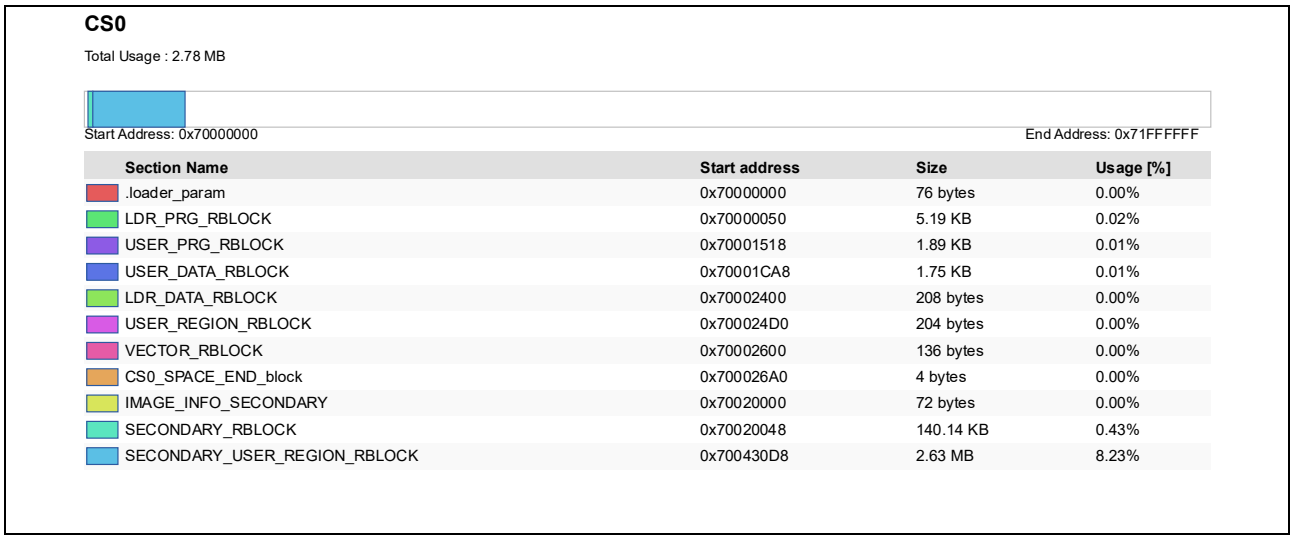
SYSTEM_RAM

Total Usage : 4 bytes

Start Address: 0x10000000 End Address: 0x101FFFFF

Section Name	Start address	Size	Usage [%]
NONCACHE_END_block	0x10180000	4 bytes	0.00%

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



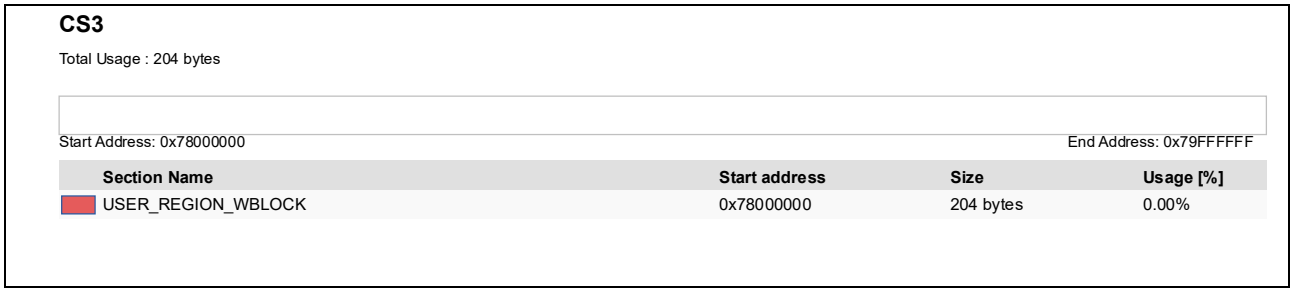
CS0

Total Usage : 2.78 MB

Start Address: 0x70000000 End Address: 0x71FFFFFF

Section Name	Start address	Size	Usage [%]
.loader_param	0x70000000	76 bytes	0.00%
LDR_PRG_RBLOCK	0x70000050	5.19 KB	0.02%
USER_PRG_RBLOCK	0x70001518	1.89 KB	0.01%
USER_DATA_RBLOCK	0x70001CA8	1.75 KB	0.01%
LDR_DATA_RBLOCK	0x70002400	208 bytes	0.00%
USER_REGION_RBLOCK	0x700024D0	204 bytes	0.00%
VECTOR_RBLOCK	0x70002600	136 bytes	0.00%
CS0_SPACE_END_block	0x700026A0	4 bytes	0.00%
IMAGE_INFO_SECONDARY	0x70020000	72 bytes	0.00%
SECONDARY_RBLOCK	0x70020048	140.14 KB	0.43%
SECONDARY_USER_REGION_RBLOCK	0x700430D8	2.63 MB	8.23%

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



CS3

Total Usage : 204 bytes

Start Address: 0x78000000 End Address: 0x79FFFFFF

Section Name	Start address	Size	Usage [%]
USER_REGION_WBLOCK	0x78000000	204 bytes	0.00%

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

(2) Secondary project

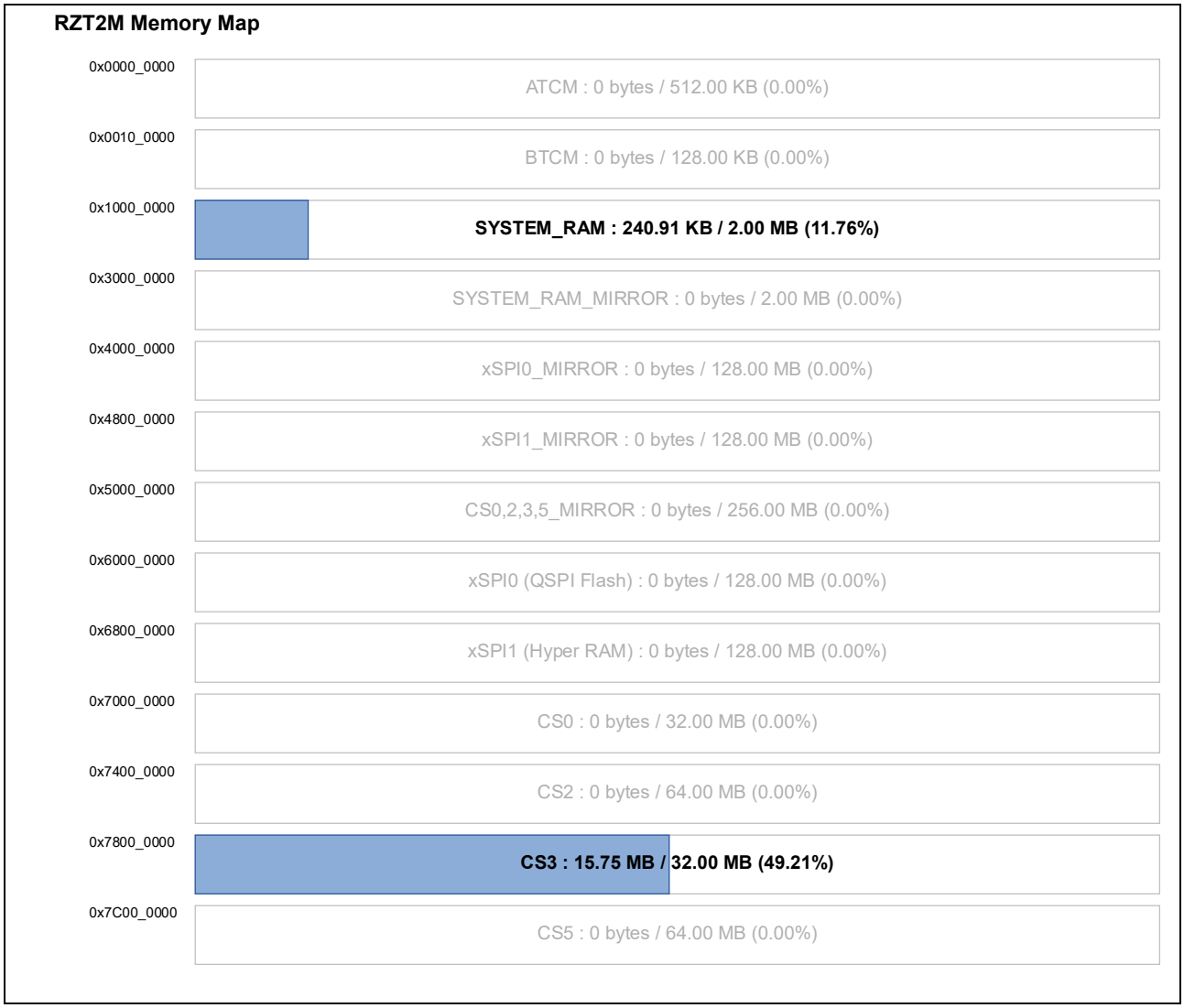
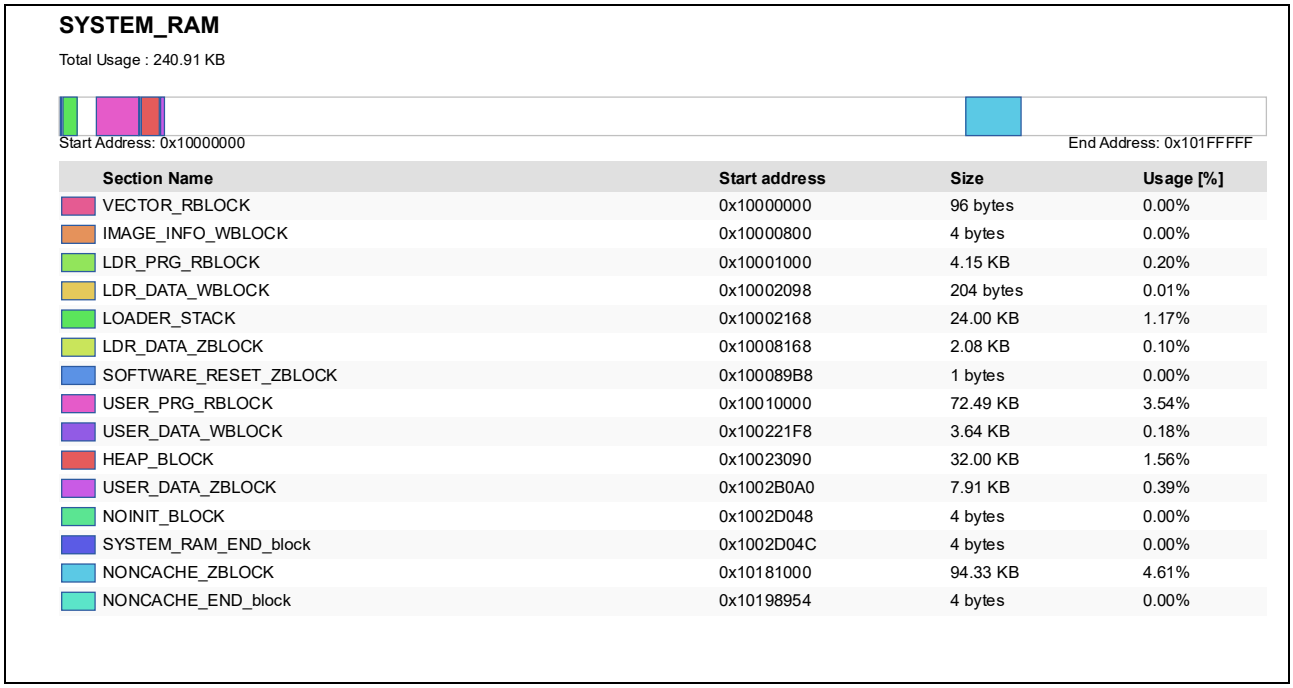
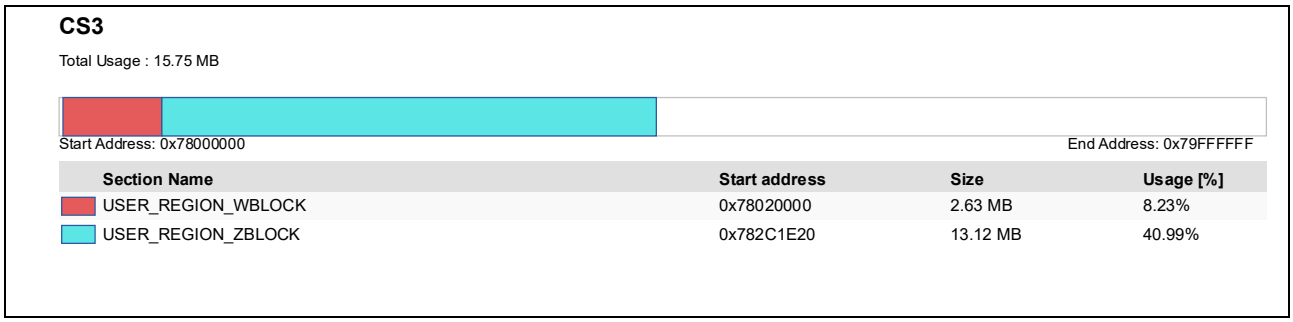


Figure 8.24. Memory footprint of the RZ/T2M Secondary project (EWARM)



Section Name	Start address	Size	Usage [%]
VECTOR_RBLOCK	0x10000000	96 bytes	0.00%
IMAGE_INFO_WBLOCK	0x10000800	4 bytes	0.00%
LDR_PRG_RBLOCK	0x10001000	4.15 KB	0.20%
LDR_DATA_WBLOCK	0x10002098	204 bytes	0.01%
LOADER_STACK	0x10002168	24.00 KB	1.17%
LDR_DATA_ZBLOCK	0x10008168	2.08 KB	0.10%
SOFTWARE_RESET_ZBLOCK	0x100089B8	1 bytes	0.00%
USER_PRG_RBLOCK	0x10010000	72.49 KB	3.54%
USER_DATA_WBLOCK	0x100221F8	3.64 KB	0.18%
HEAP_BLOCK	0x10023090	32.00 KB	1.56%
USER_DATA_ZBLOCK	0x1002B0A0	7.91 KB	0.39%
NOINIT_BLOCK	0x1002D048	4 bytes	0.00%
SYSTEM_RAM_END_block	0x1002D04C	4 bytes	0.00%
NONCACHE_ZBLOCK	0x10181000	94.33 KB	4.61%
NONCACHE_END_block	0x10198954	4 bytes	0.00%

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



Section Name	Start address	Size	Usage [%]
USER_REGION_WBLOCK	0x78020000	2.63 MB	8.23%
USER_REGION_ZBLOCK	0x782C1E20	13.12 MB	40.99%

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

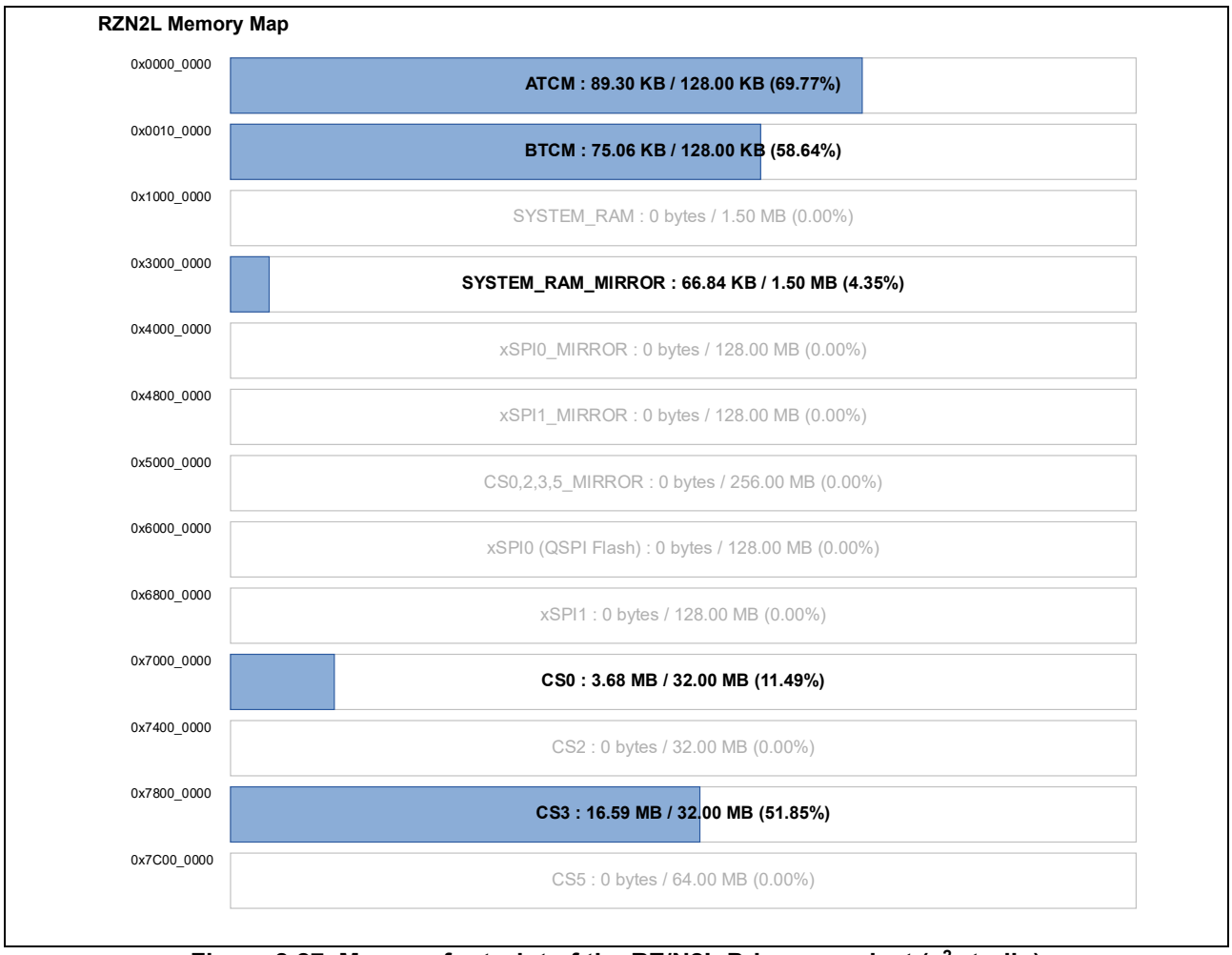
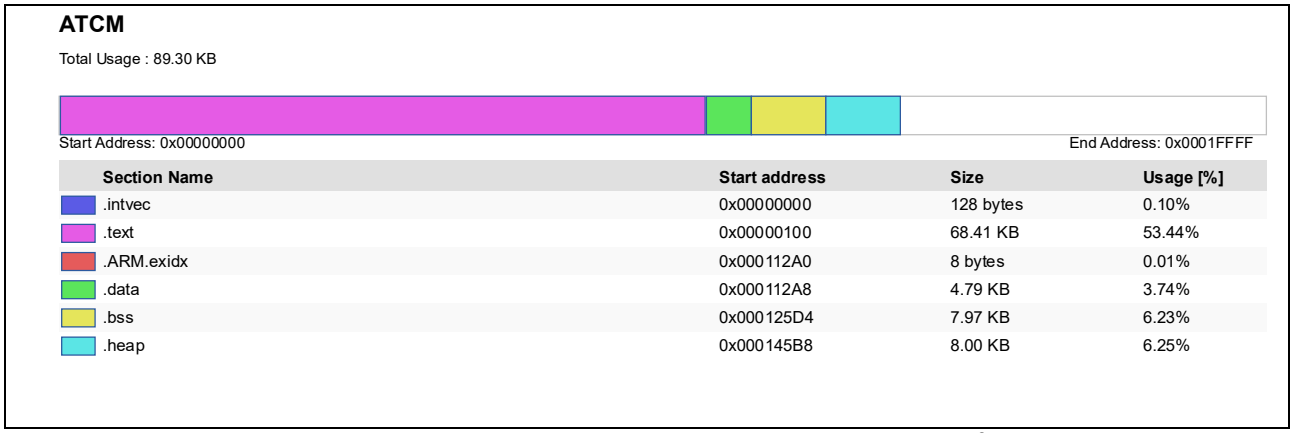


Figure 8.27. Memory footprint of the RZ/N2L Primary project (e² studio)



Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.10%
.text	0x00000100	68.41 KB	53.44%
.ARM.exidx	0x000112A0	8 bytes	0.01%
.data	0x000112A8	4.79 KB	3.74%
.bss	0x000125D4	7.97 KB	6.23%
.heap	0x000145B8	8.00 KB	6.25%

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

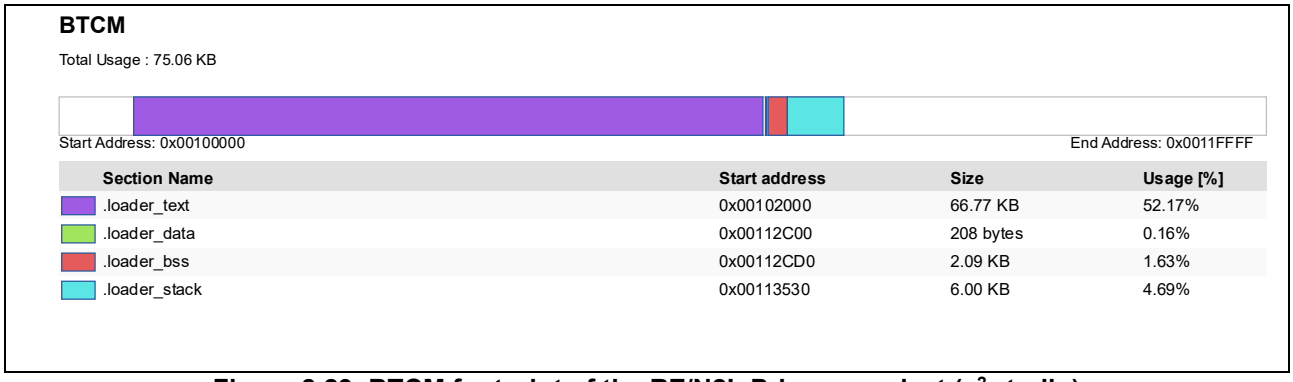


Figure 8.29. BTCM footprint of the RZ/N2L Primary project (e² studio)



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

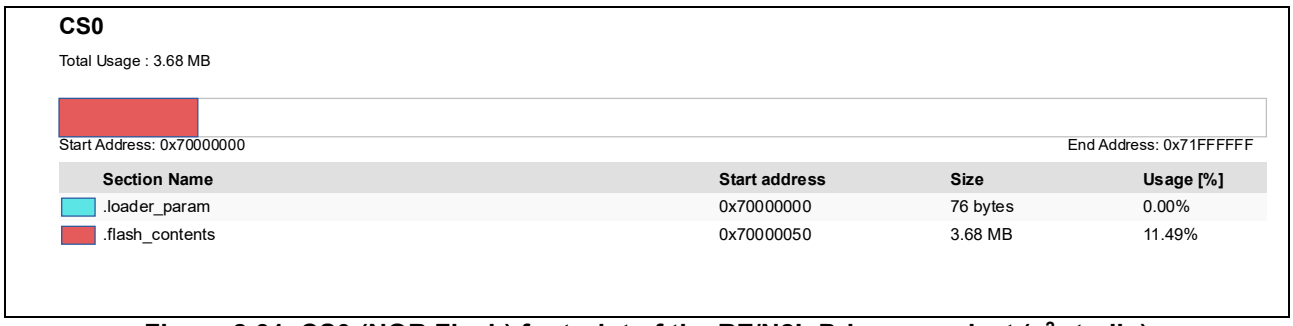


Figure 8.31. CS0 (NOR Flash) footprint of the RZ/N2L Primary project (e² studio)

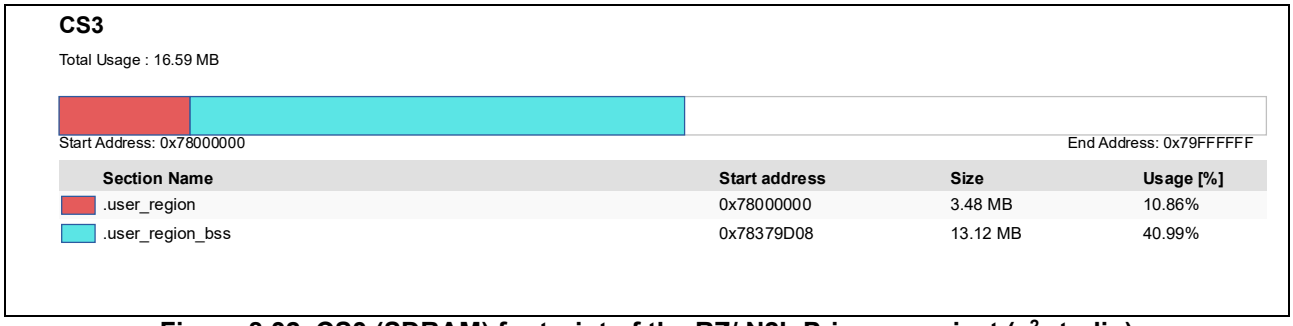


Figure 8.32. CS3 (SDRAM) footprint of the RZ/ N2L Primary project (e² studio)

8.3.2.2 EWARM

(1) Primary project

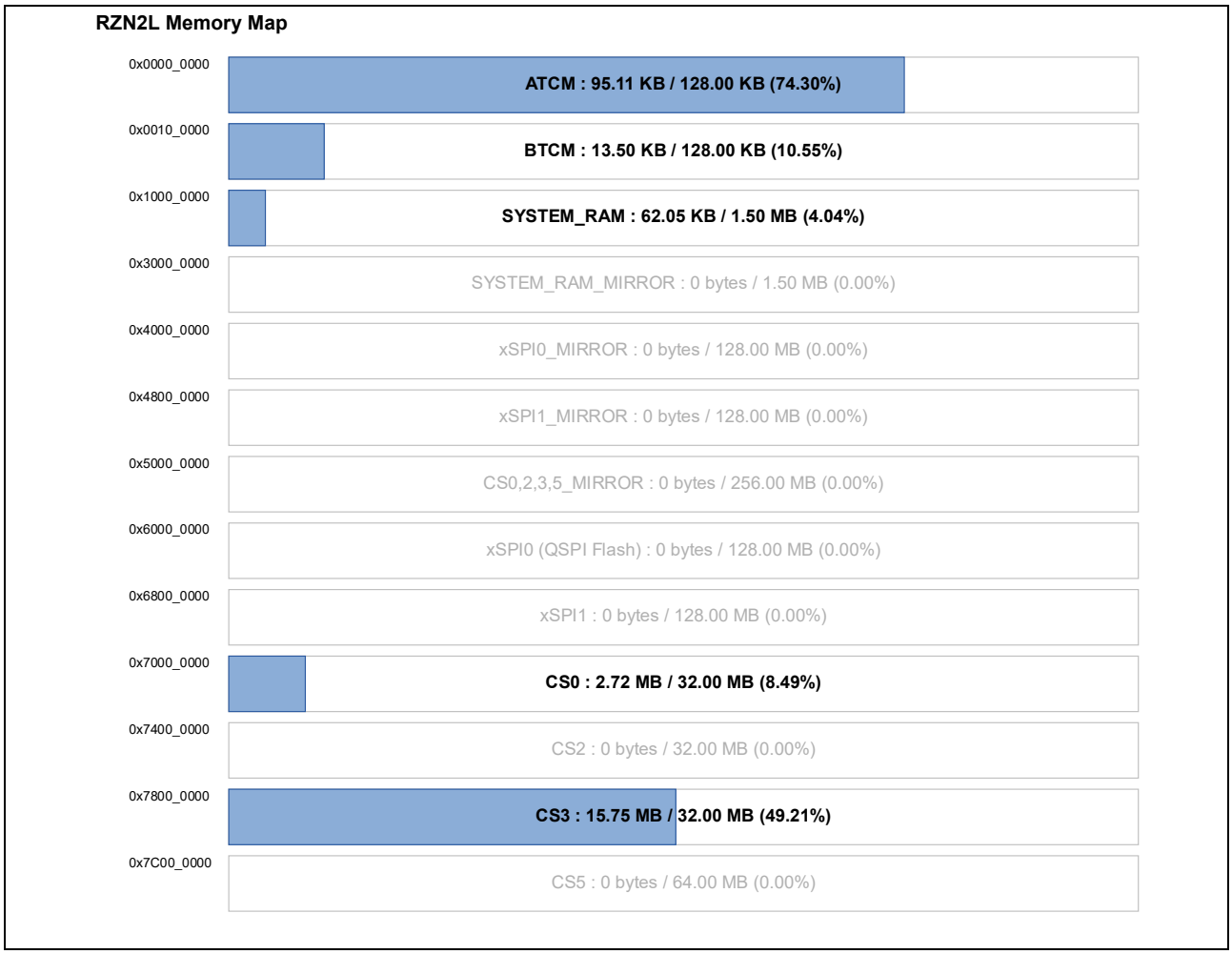
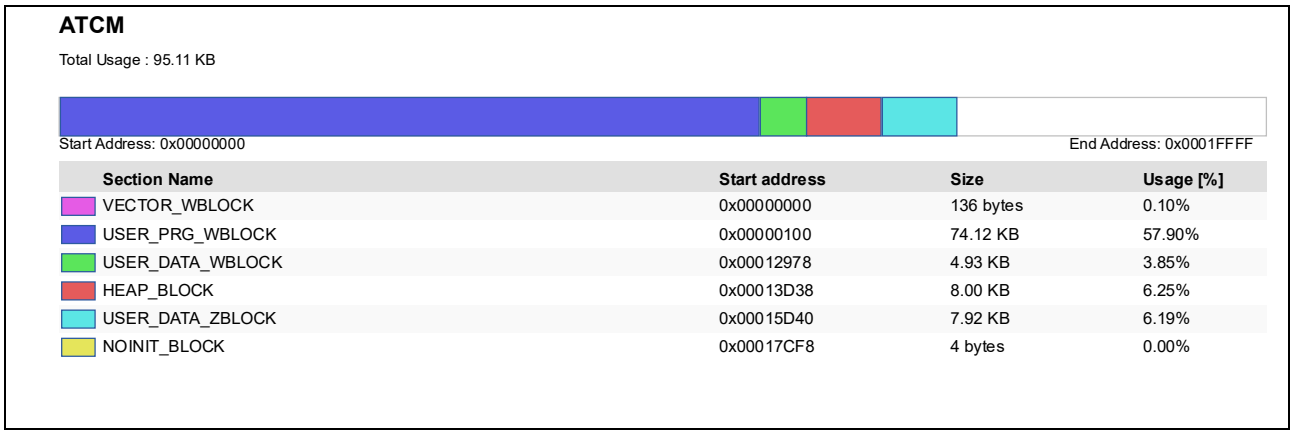
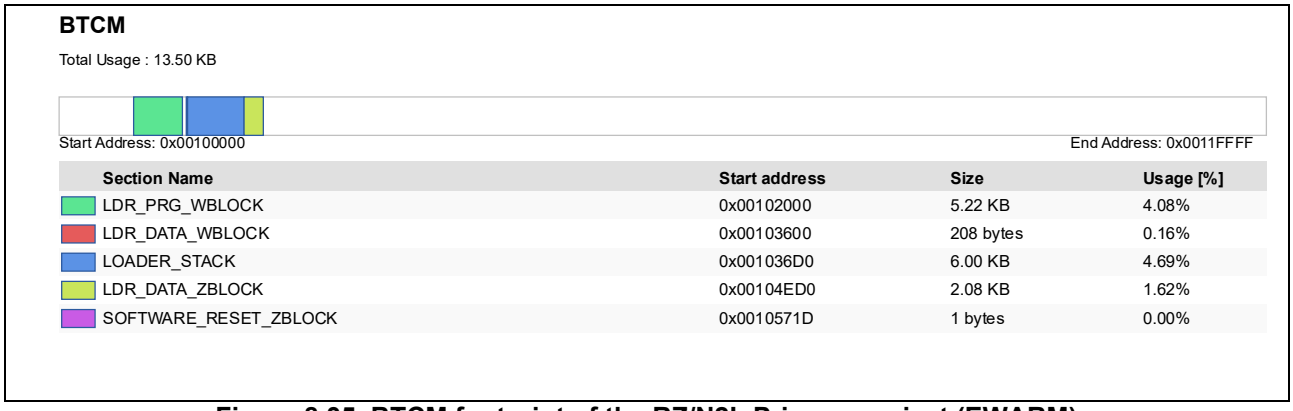


Figure 8.33. Memory footprint of the RZ/N2L Primary project (EWARM)



Section Name	Start address	Size	Usage [%]
VECTOR_WBLOCK	0x00000000	136 bytes	0.10%
USER_PRG_WBLOCK	0x00000100	74.12 KB	57.90%
USER_DATA_WBLOCK	0x00012978	4.93 KB	3.85%
HEAP_BLOCK	0x00013D38	8.00 KB	6.25%
USER_DATA_ZBLOCK	0x00015D40	7.92 KB	6.19%
NOINIT_BLOCK	0x00017CF8	4 bytes	0.00%

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



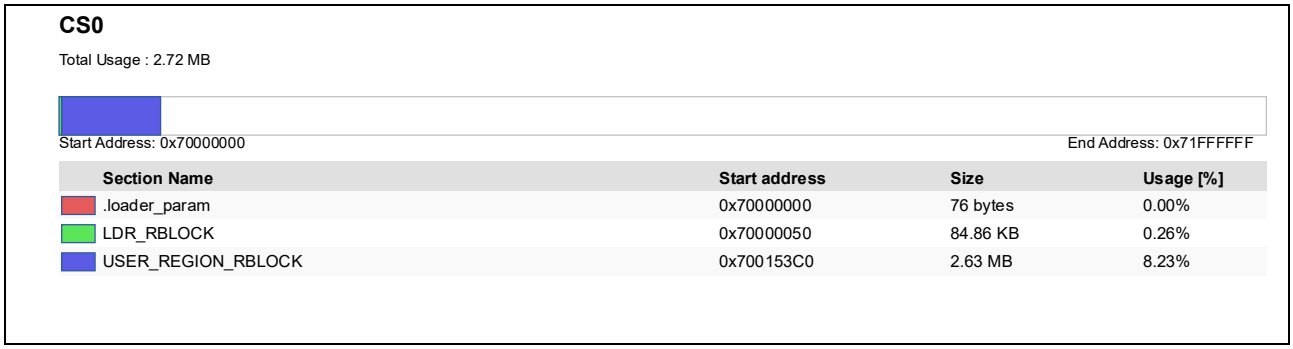
Section Name	Start address	Size	Usage [%]
LDR_PRG_WBLOCK	0x00102000	5.22 KB	4.08%
LDR_DATA_WBLOCK	0x00103600	208 bytes	0.16%
LOADER_STACK	0x001036D0	6.00 KB	4.69%
LDR_DATA_ZBLOCK	0x00104ED0	2.08 KB	1.62%
SOFTWARE_RESET_ZBLOCK	0x0010571D	1 bytes	0.00%

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



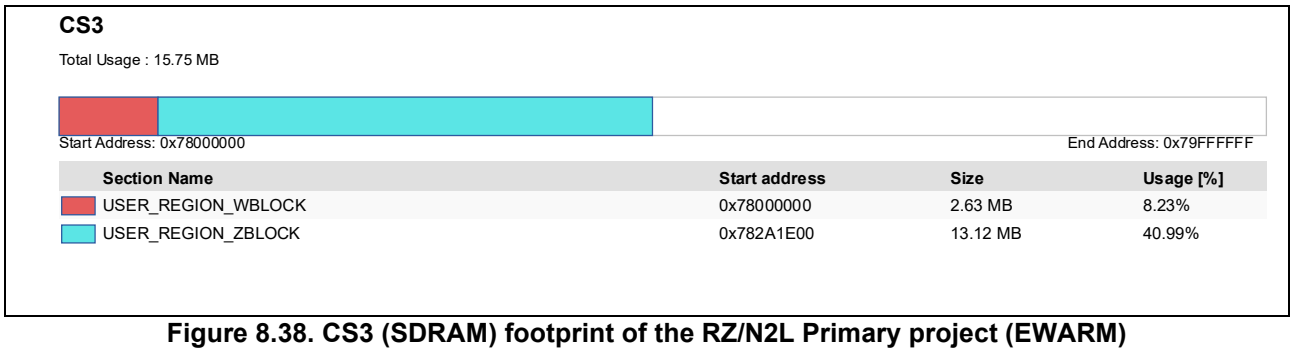
Section Name	Start address	Size	Usage [%]
NONCACHE_ZBLOCK	0x10100000	62.05 KB	4.04%

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



Section Name	Start address	Size	Usage [%]
.loader_param	0x70000000	76 bytes	0.00%
LDR_RBLOCK	0x70000050	84.86 KB	0.26%
USER_REGION_RBLOCK	0x700153C0	2.63 MB	8.23%

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



Section Name	Start address	Size	Usage [%]
USER_REGION_WBLOCK	0x78000000	2.63 MB	8.23%
USER_REGION_ZBLOCK	0x782A1E00	13.12 MB	40.99%

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す。

8.3.3.1 e² studio

(1) Primary project

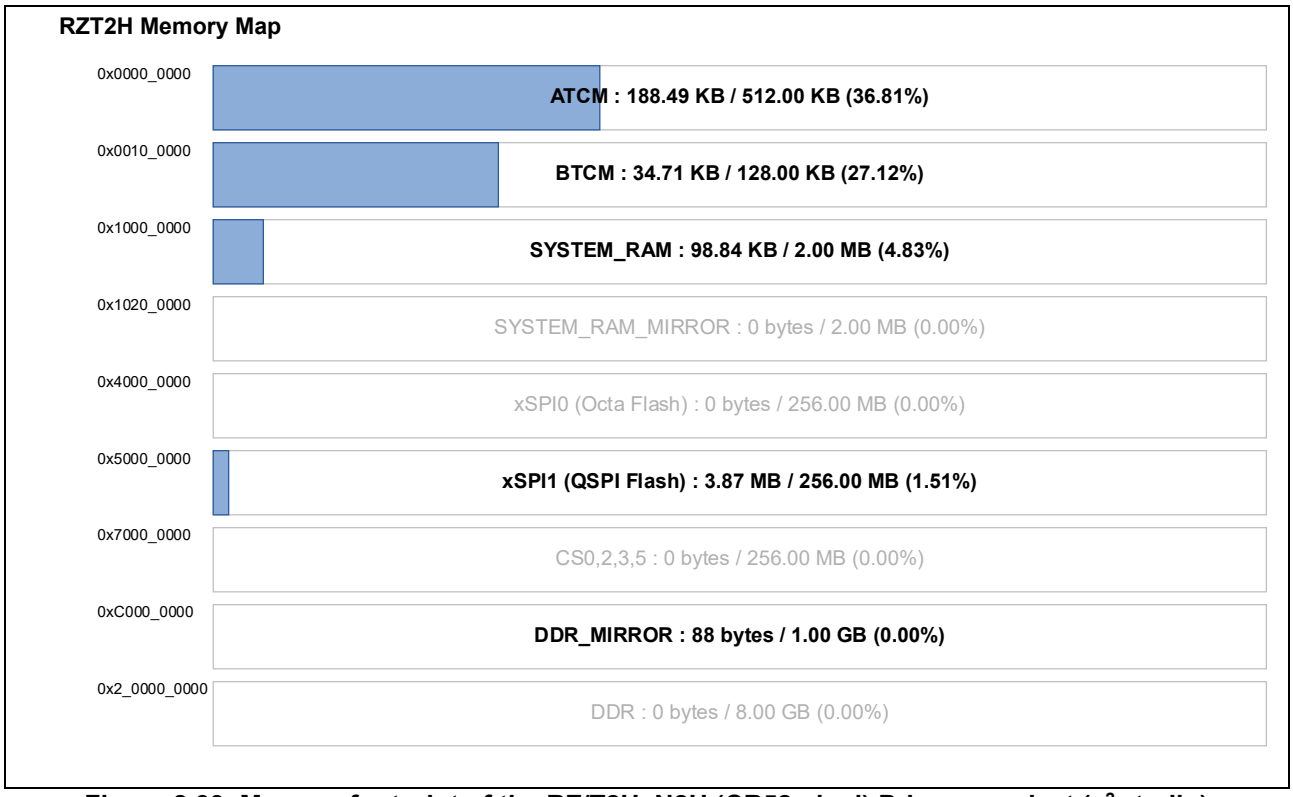
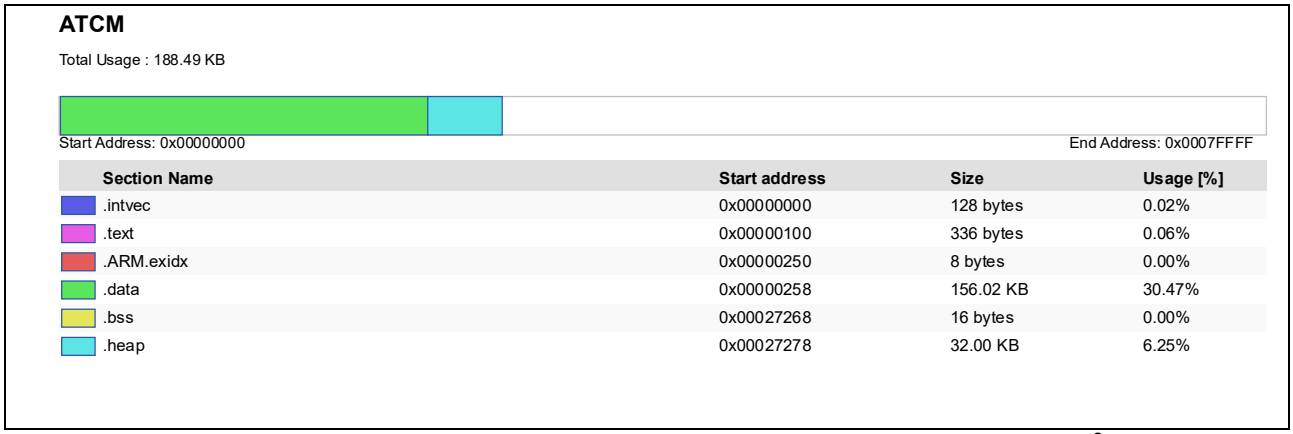
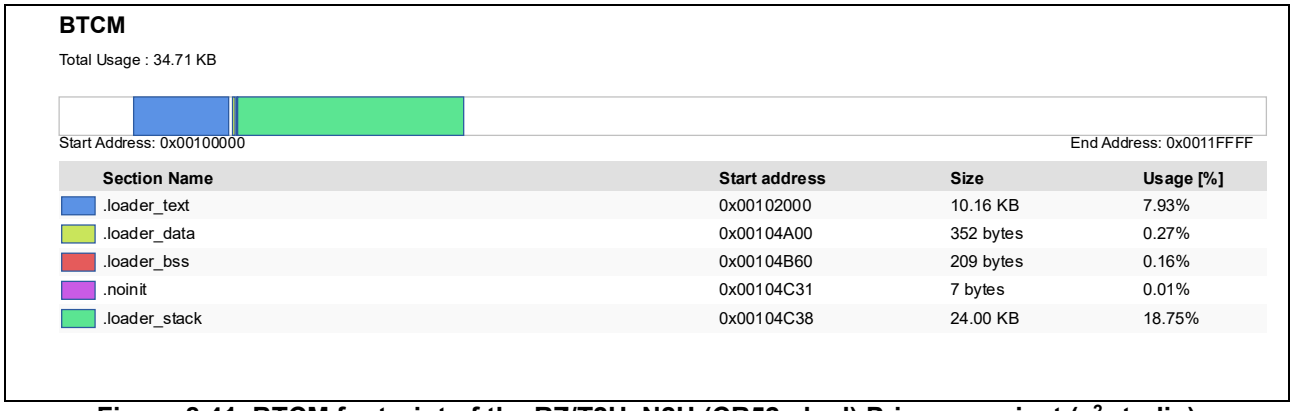


Figure 8.39. Memory footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)



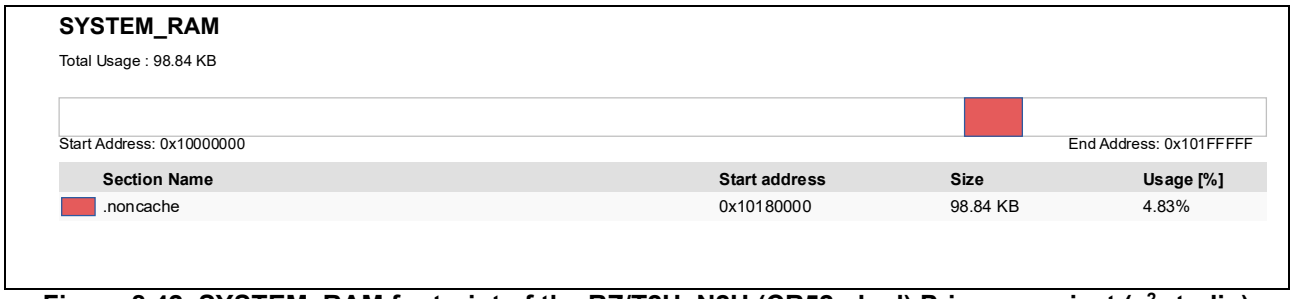
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.text	0x00000100	336 bytes	0.06%
.ARM.exidx	0x00000250	8 bytes	0.00%
.data	0x00000258	156.02 KB	30.47%
.bss	0x00027268	16 bytes	0.00%
.heap	0x00027278	32.00 KB	6.25%

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



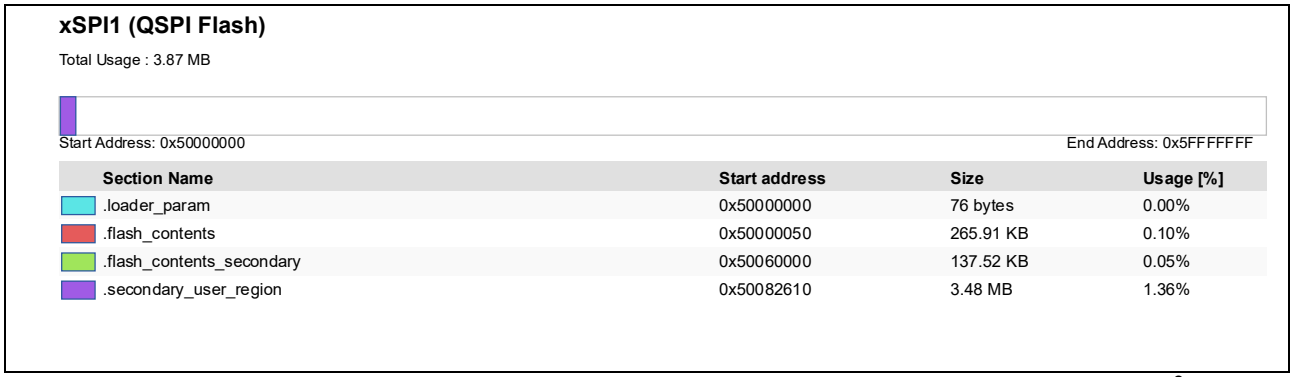
Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	10.16 KB	7.93%
.loader_data	0x00104A00	352 bytes	0.27%
.loader_bss	0x00104B60	209 bytes	0.16%
.noinit	0x00104C31	7 bytes	0.01%
.loader_stack	0x00104C38	24.00 KB	18.75%

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



Section Name	Start address	Size	Usage [%]
.noncache	0x10180000	98.84 KB	4.83%

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



Section Name	Start address	Size	Usage [%]
.loader_param	0x50000000	76 bytes	0.00%
.flash_contents	0x50000050	265.91 KB	0.10%
.flash_contents_secondary	0x50060000	137.52 KB	0.05%
.secondary_user_region	0x50082610	3.48 MB	1.36%

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



Section Name	Start address	Size	Usage [%]
.user_region	0xC0000000	88 bytes	0.00%

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

(2) Secondary project

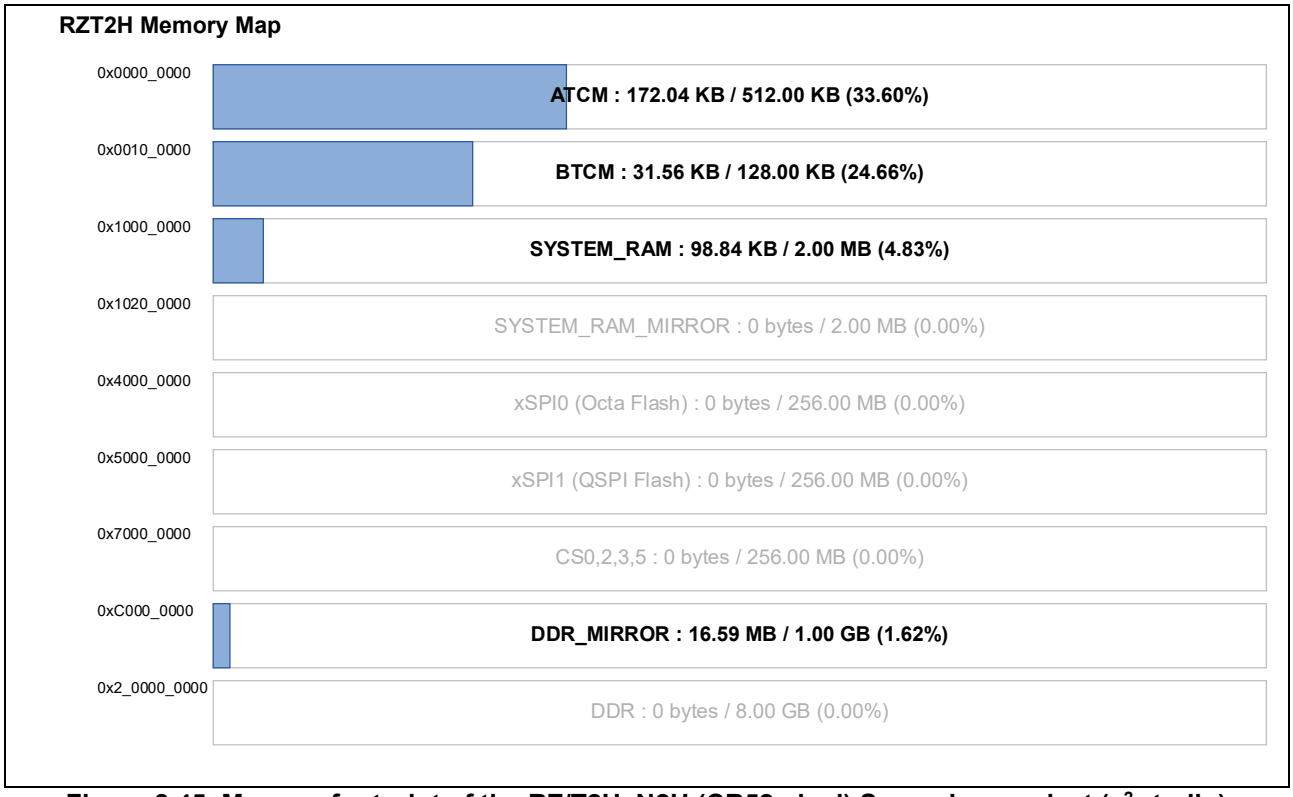
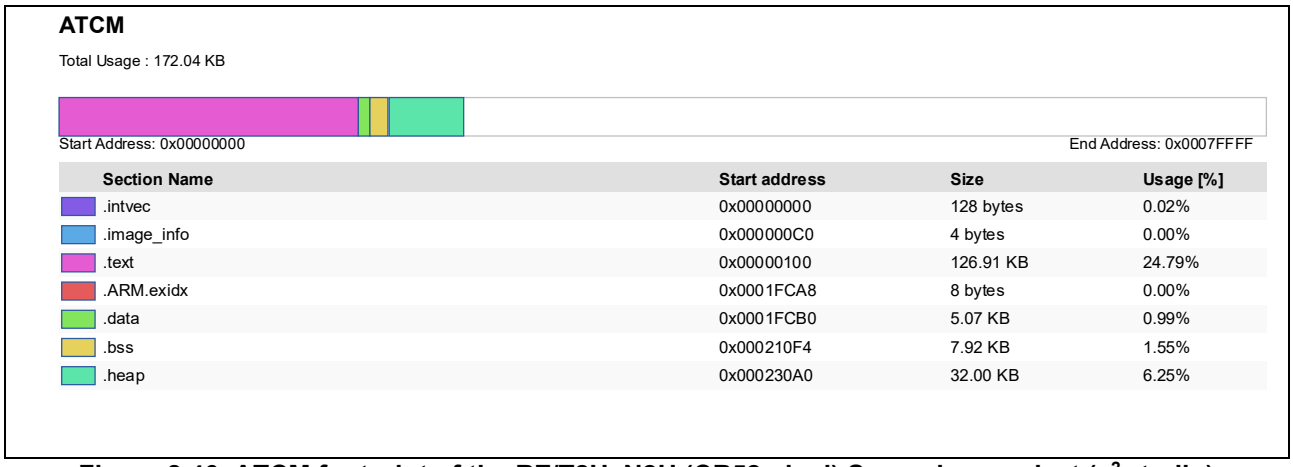
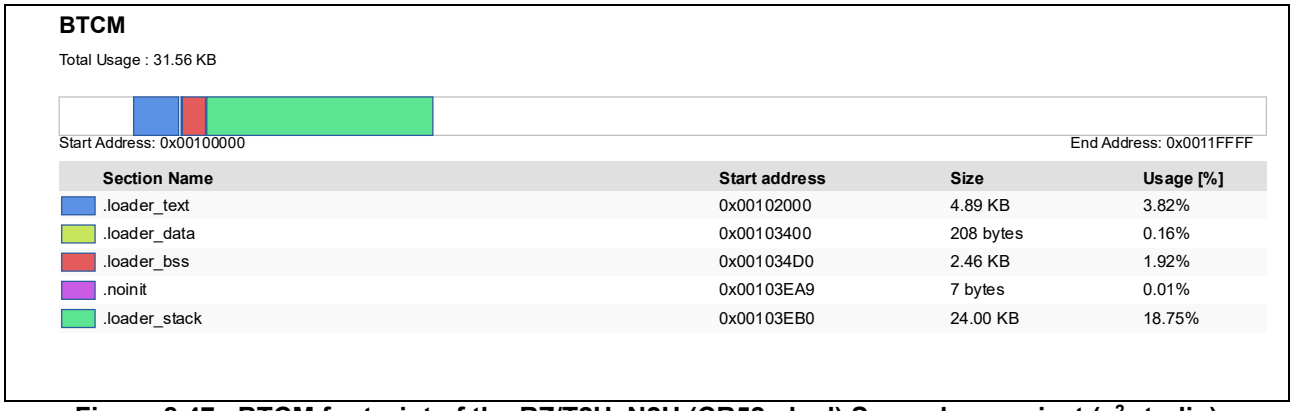


Figure 8.45. Memory footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)



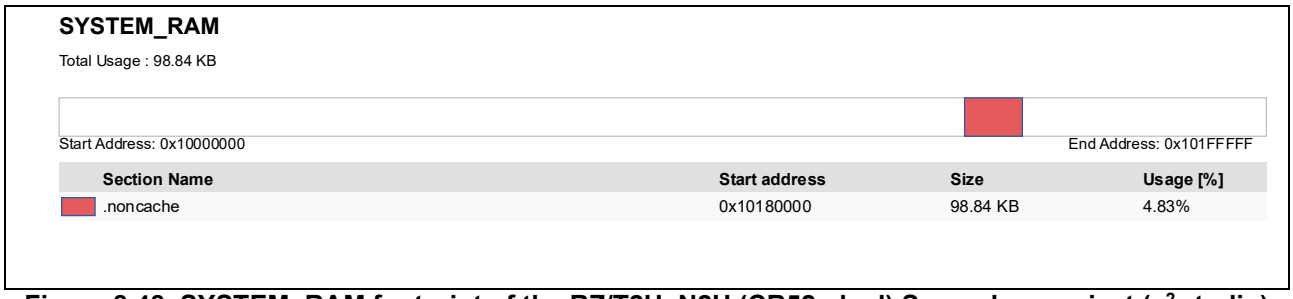
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.image_info	0x000000C0	4 bytes	0.00%
.text	0x00000100	126.91 KB	24.79%
.ARM.exidx	0x0001FCA8	8 bytes	0.00%
.data	0x0001FCB0	5.07 KB	0.99%
.bss	0x000210F4	7.92 KB	1.55%
.heap	0x000230A0	32.00 KB	6.25%

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



Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	4.89 KB	3.82%
.loader_data	0x00103400	208 bytes	0.16%
.loader_bss	0x001034D0	2.46 KB	1.92%
.noinit	0x00103EA9	7 bytes	0.01%
.loader_stack	0x00103EB0	24.00 KB	18.75%

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



Section Name	Start address	Size	Usage [%]
.noncache	0x10180000	98.84 KB	4.83%

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



Section Name	Start address	Size	Usage [%]
.user_region	0xC0020000	3.48 MB	0.34%
.user_region_bss	0xC0399D00	13.12 MB	1.28%

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

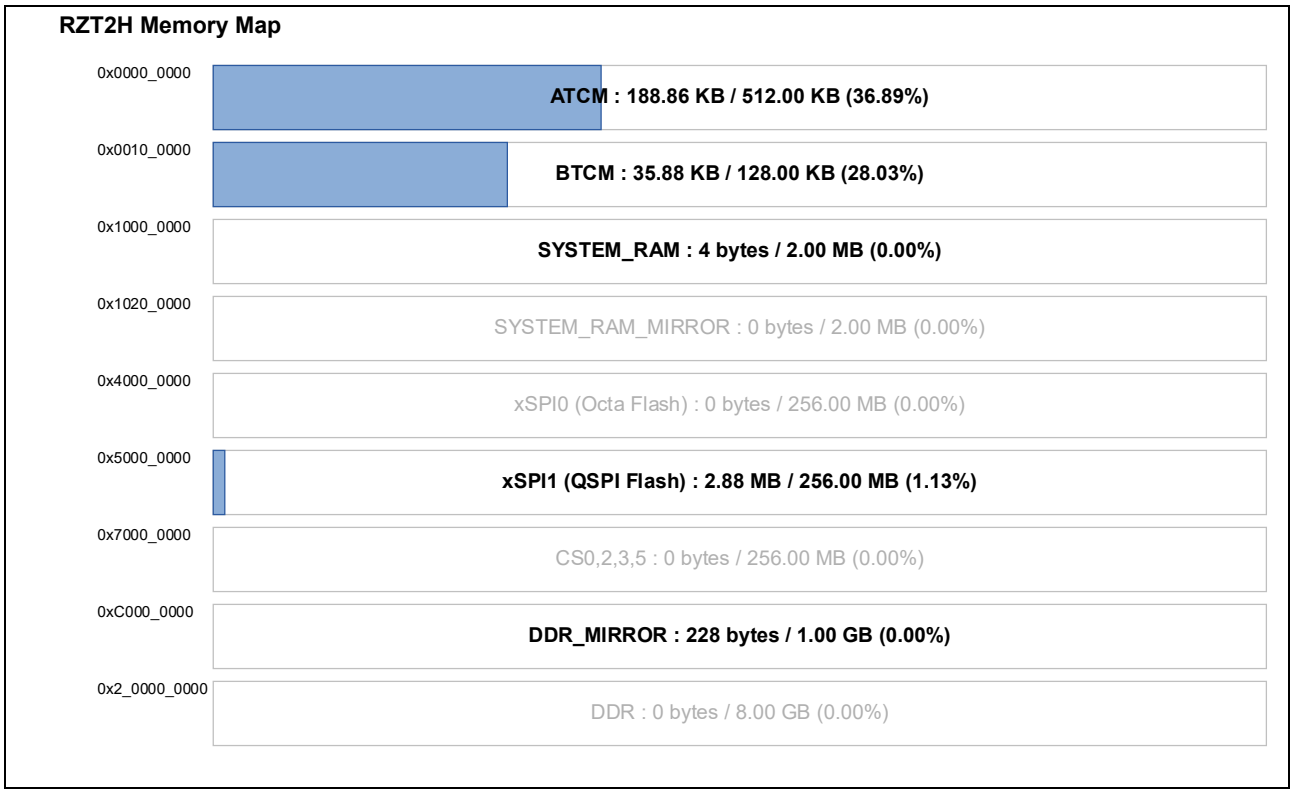
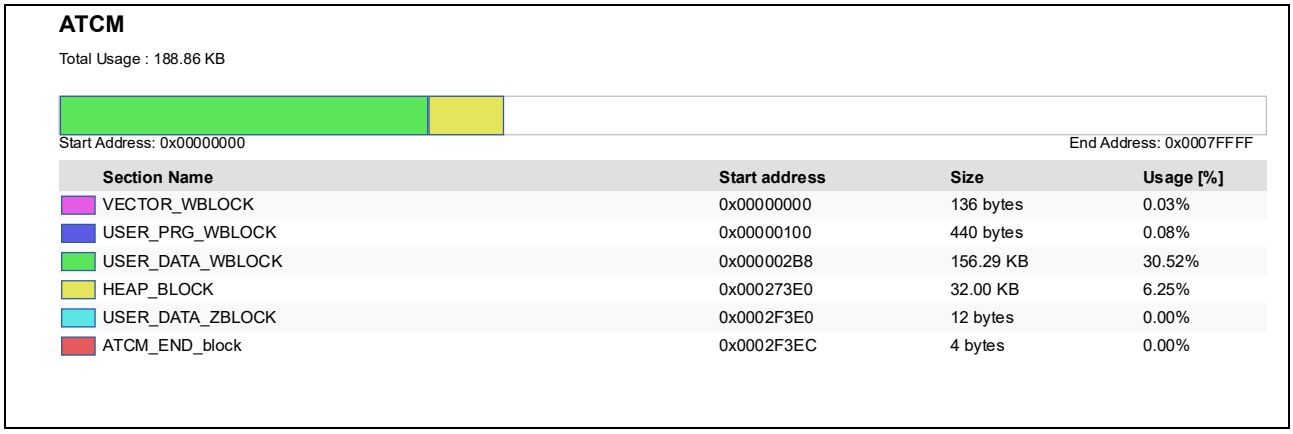
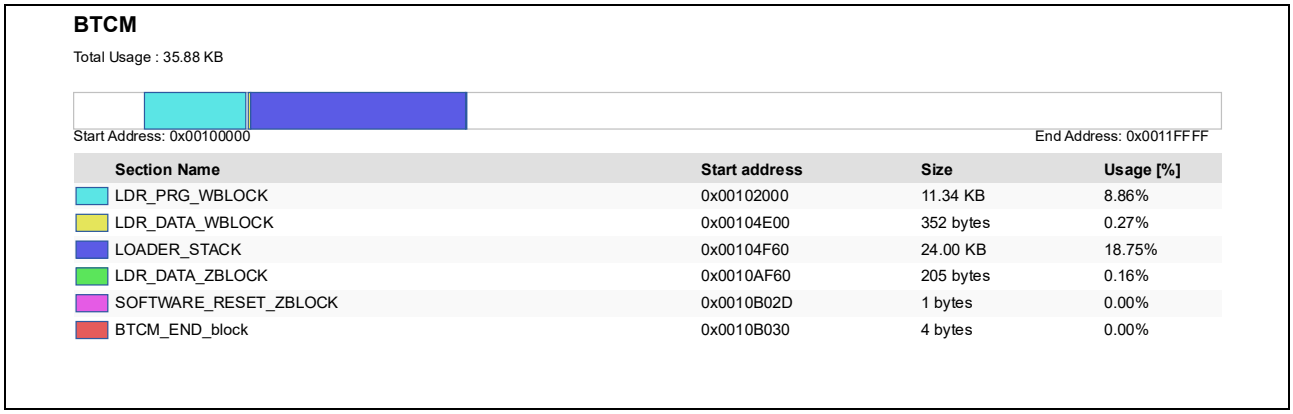


Figure 8.50. Memory footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)



Section Name	Start address	Size	Usage [%]
VECTOR_WBLOCK	0x00000000	136 bytes	0.03%
USER_PRG_WBLOCK	0x00000100	440 bytes	0.08%
USER_DATA_WBLOCK	0x000002B8	156.29 KB	30.52%
HEAP_BLOCK	0x000273E0	32.00 KB	6.25%
USER_DATA_ZBLOCK	0x0002F3E0	12 bytes	0.00%
ATCM_END_block	0x0002F3EC	4 bytes	0.00%

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



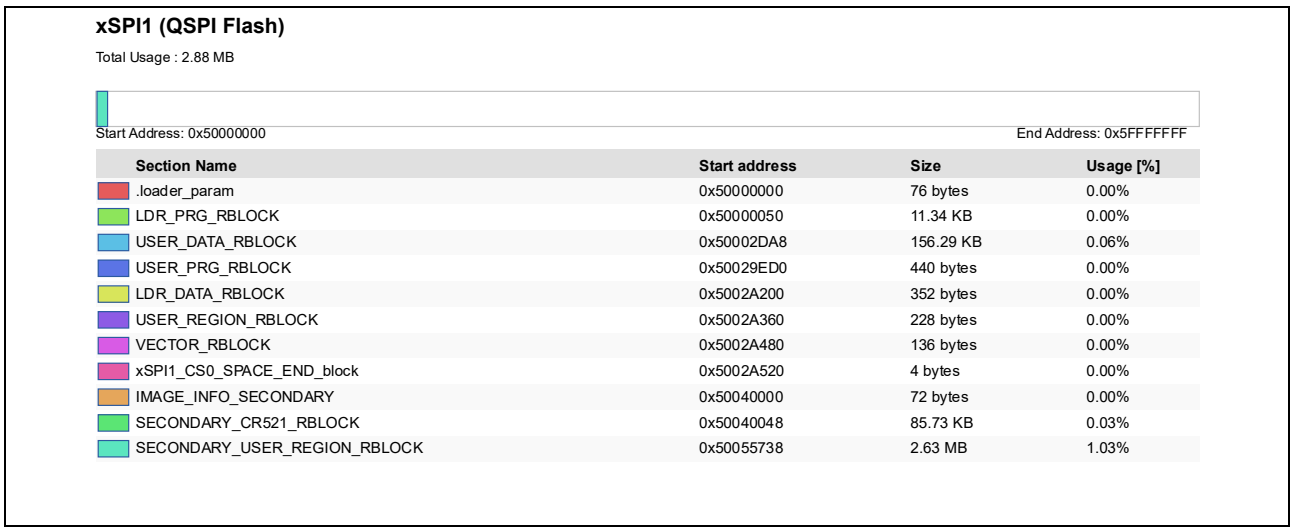
Section Name	Start address	Size	Usage [%]
LDR_PRG_WBLOCK	0x00102000	11.34 KB	8.86%
LDR_DATA_WBLOCK	0x00104E00	352 bytes	0.27%
LOADER_STACK	0x00104F60	24.00 KB	18.75%
LDR_DATA_ZBLOCK	0x0010AF60	205 bytes	0.16%
SOFTWARE_RESET_ZBLOCK	0x0010B02D	1 bytes	0.00%
BTCM_END_block	0x0010B030	4 bytes	0.00%

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



Section Name	Start address	Size	Usage [%]
NONCACHE_END_block	0x10180000	4 bytes	0.00%

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



Section Name	Start address	Size	Usage [%]
.loader_param	0x50000000	76 bytes	0.00%
LDR_PRG_RBLOCK	0x50000050	11.34 KB	0.00%
USER_DATA_RBLOCK	0x50002DA8	156.29 KB	0.06%
USER_PRG_RBLOCK	0x50029ED0	440 bytes	0.00%
LDR_DATA_RBLOCK	0x5002A200	352 bytes	0.00%
USER_REGION_RBLOCK	0x5002A360	228 bytes	0.00%
VECTOR_RBLOCK	0x5002A480	136 bytes	0.00%
xSPI1_CS0_SPACE_END_block	0x5002A520	4 bytes	0.00%
IMAGE_INFO_SECONDARY	0x50040000	72 bytes	0.00%
SECONDARY_CR521_RBLOCK	0x50040048	85.73 KB	0.03%
SECONDARY_USER_REGION_RBLOCK	0x50055738	2.63 MB	1.03%

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



Section Name	Start address	Size	Usage [%]
USER_REGION_WBLOCK	0xC0000000	228 bytes	0.00%

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

(2) Secondary project

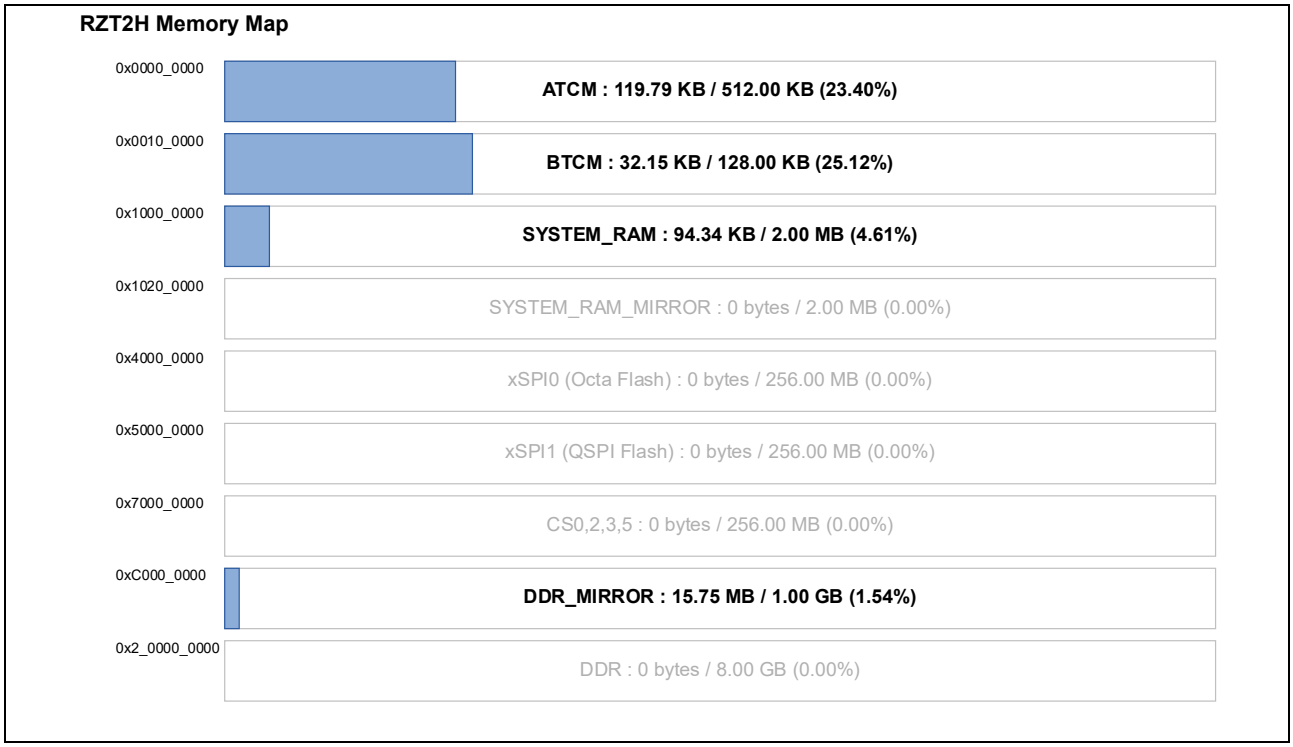
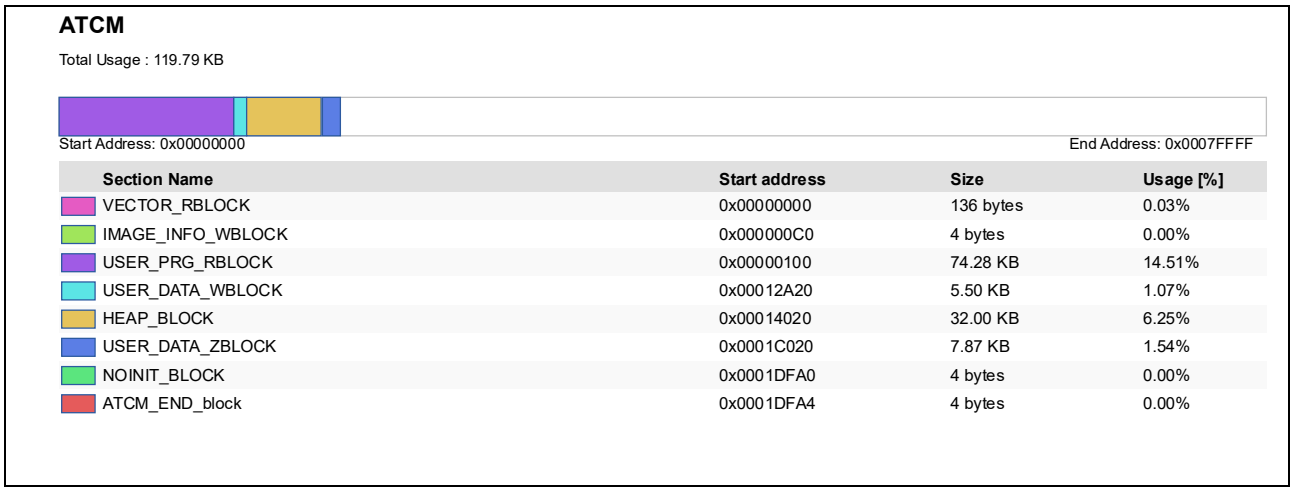
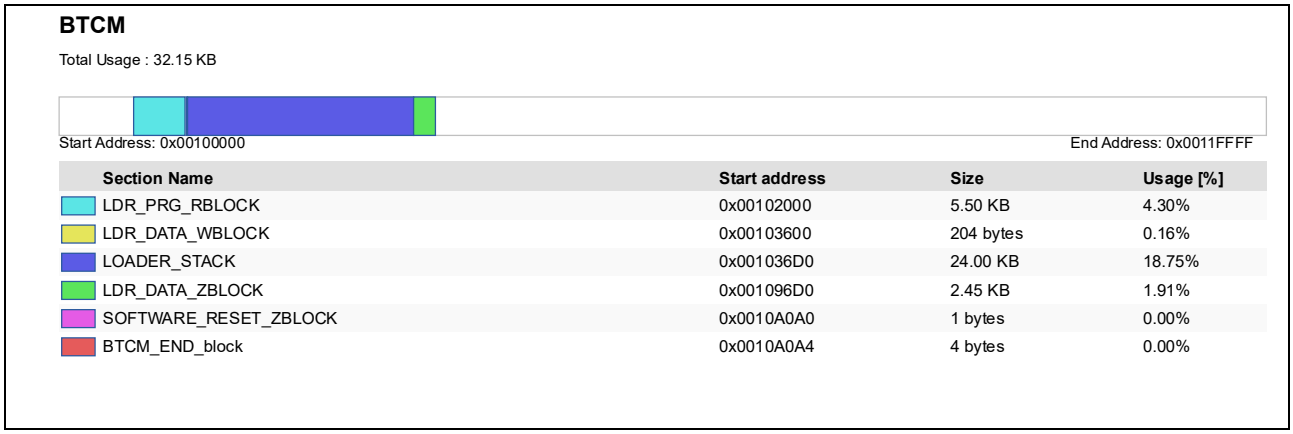


Figure 8.56. Memory footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)



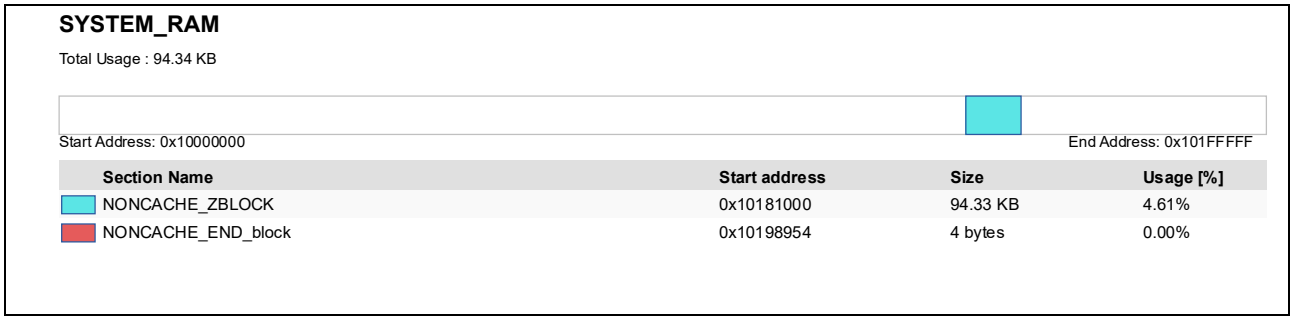
Section Name	Start address	Size	Usage [%]
VECTOR_RBLOCK	0x00000000	136 bytes	0.03%
IMAGE_INFO_WBLOCK	0x000000C0	4 bytes	0.00%
USER_PRG_RBLOCK	0x00000100	74.28 KB	14.51%
USER_DATA_WBLOCK	0x00012A20	5.50 KB	1.07%
HEAP_BLOCK	0x00014020	32.00 KB	6.25%
USER_DATA_ZBLOCK	0x0001C020	7.87 KB	1.54%
NOINIT_BLOCK	0x0001DFA0	4 bytes	0.00%
ATCM_END_block	0x0001DFA4	4 bytes	0.00%

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



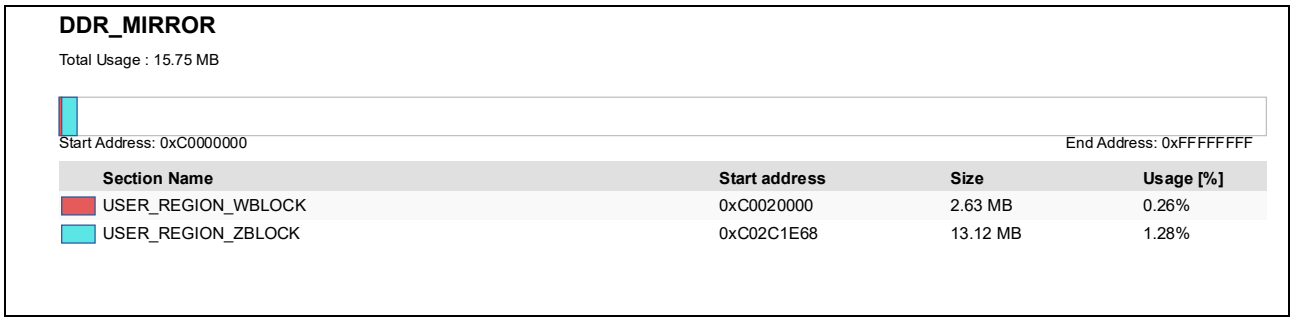
Section Name	Start address	Size	Usage [%]
LDR_PRG_RBLOCK	0x00102000	5.50 KB	4.30%
LDR_DATA_WBLOCK	0x00103600	204 bytes	0.16%
LOADER_STACK	0x001036D0	24.00 KB	18.75%
LDR_DATA_ZBLOCK	0x001096D0	2.45 KB	1.91%
SOFTWARE_RESET_ZBLOCK	0x0010A0A0	1 bytes	0.00%
BTCM_END_block	0x0010A0A4	4 bytes	0.00%

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



Section Name	Start address	Size	Usage [%]
NONCACHE_ZBLOCK	0x10181000	94.33 KB	4.61%
NONCACHE_END_block	0x10198954	4 bytes	0.00%

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



Section Name	Start address	Size	Usage [%]
USER_REGION_WBLOCK	0xC0020000	2.63 MB	0.26%
USER_REGION_ZBLOCK	0xC02C1E68	13.12 MB	1.28%

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す。

8.3.4.1 e² studio

(1) Primary project

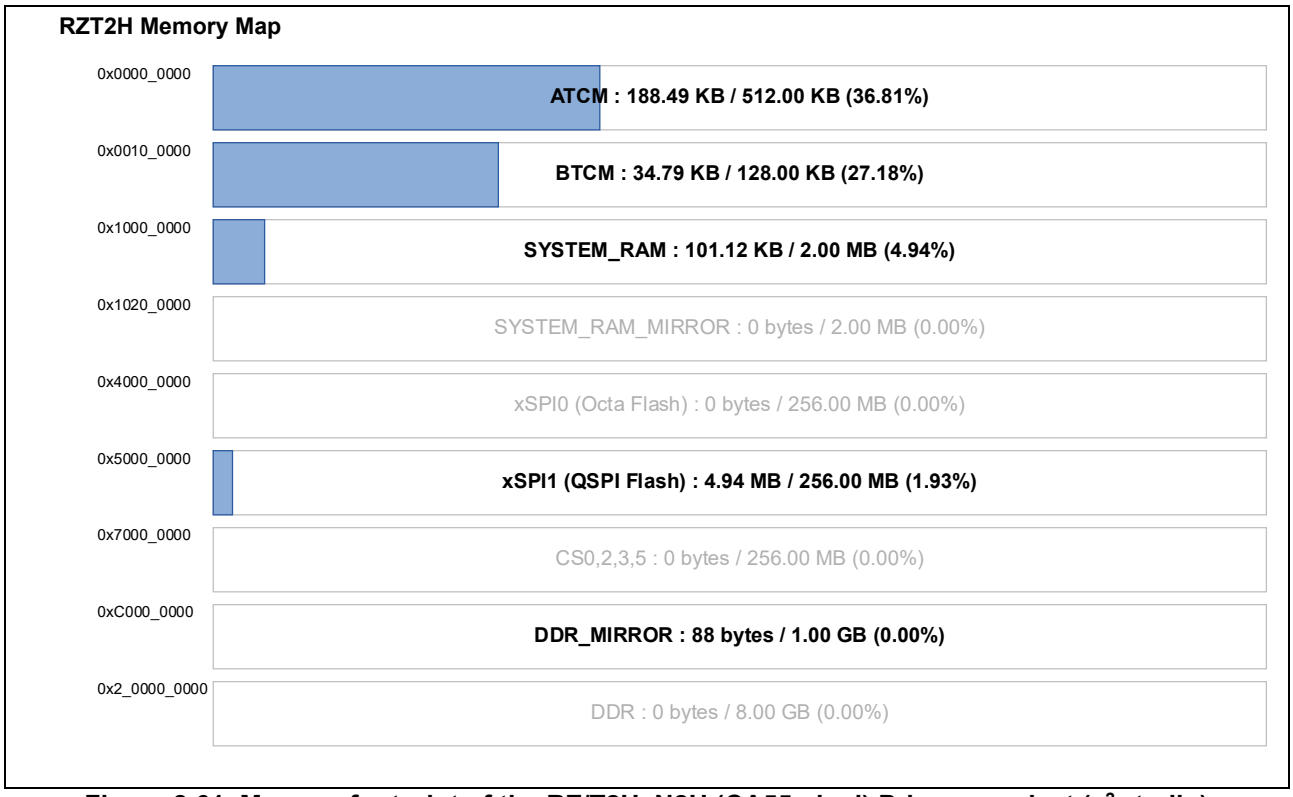
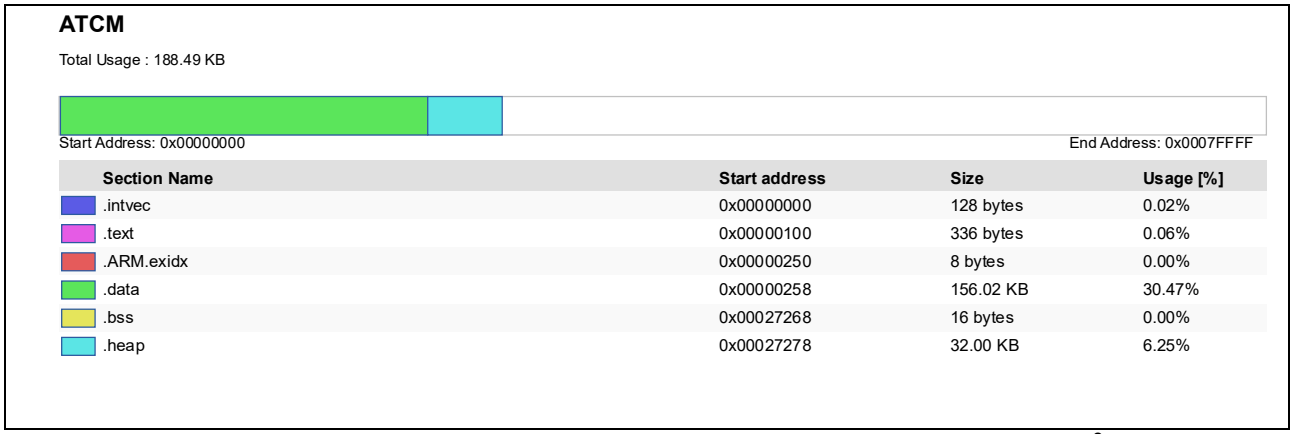
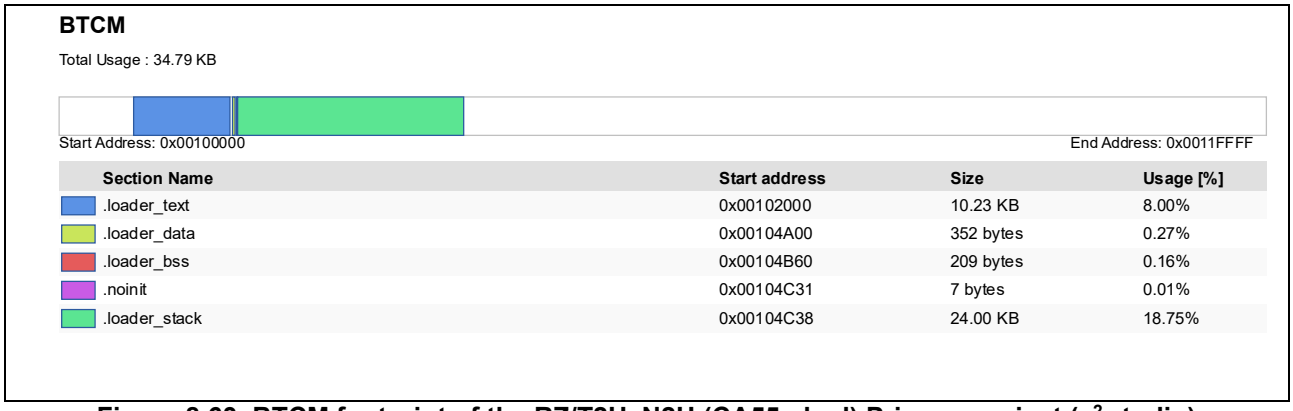


Figure 8.61. Memory footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)



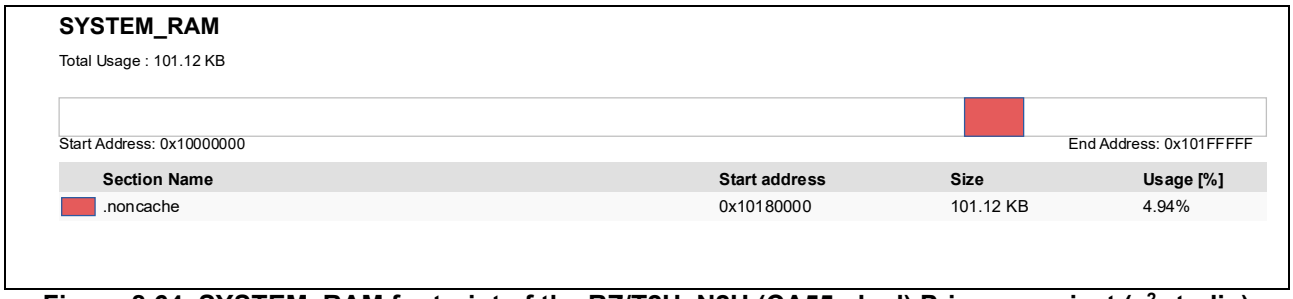
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.text	0x00000100	336 bytes	0.06%
.ARM.exidx	0x00000250	8 bytes	0.00%
.data	0x00000258	156.02 KB	30.47%
.bss	0x00027268	16 bytes	0.00%
.heap	0x00027278	32.00 KB	6.25%

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



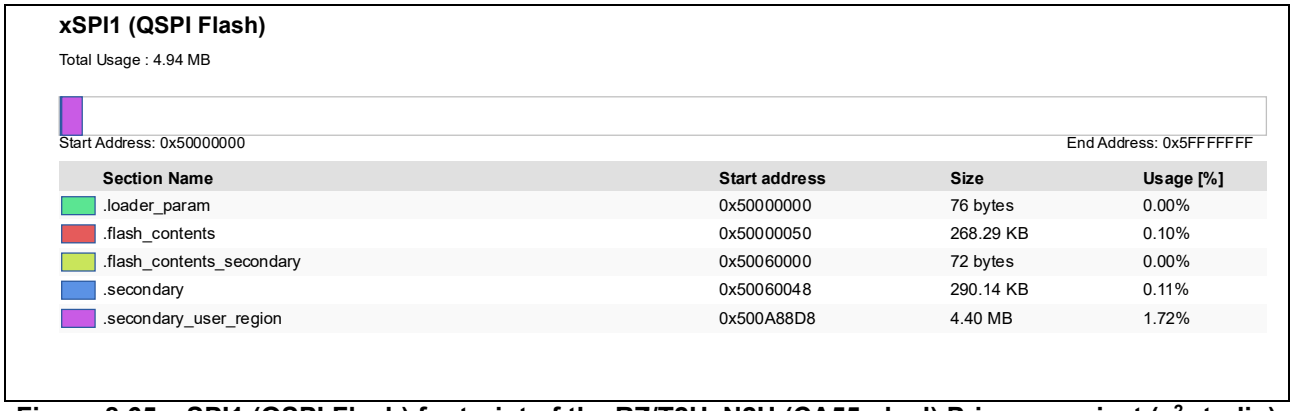
Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	10.23 KB	8.00%
.loader_data	0x00104A00	352 bytes	0.27%
.loader_bss	0x00104B60	209 bytes	0.16%
.noinit	0x00104C31	7 bytes	0.01%
.loader_stack	0x00104C38	24.00 KB	18.75%

Figure 8.63. BTCM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)



Section Name	Start address	Size	Usage [%]
.noncache	0x10180000	101.12 KB	4.94%

Figure 8.64. SYSTEM_RAM footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)



Section Name	Start address	Size	Usage [%]
.loader_param	0x50000000	76 bytes	0.00%
.flash_contents	0x50000050	268.29 KB	0.10%
.flash_contents_secondary	0x50060000	72 bytes	0.00%
.secondary	0x50060048	290.14 KB	0.11%
.secondary_user_region	0x500A88D8	4.40 MB	1.72%

Figure 8.65. xSPI1 (QSPI Flash) footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)



Section Name	Start address	Size	Usage [%]
.user_region	0xC0000000	88 bytes	0.00%

Figure 8.66. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)

(2) Secondary project

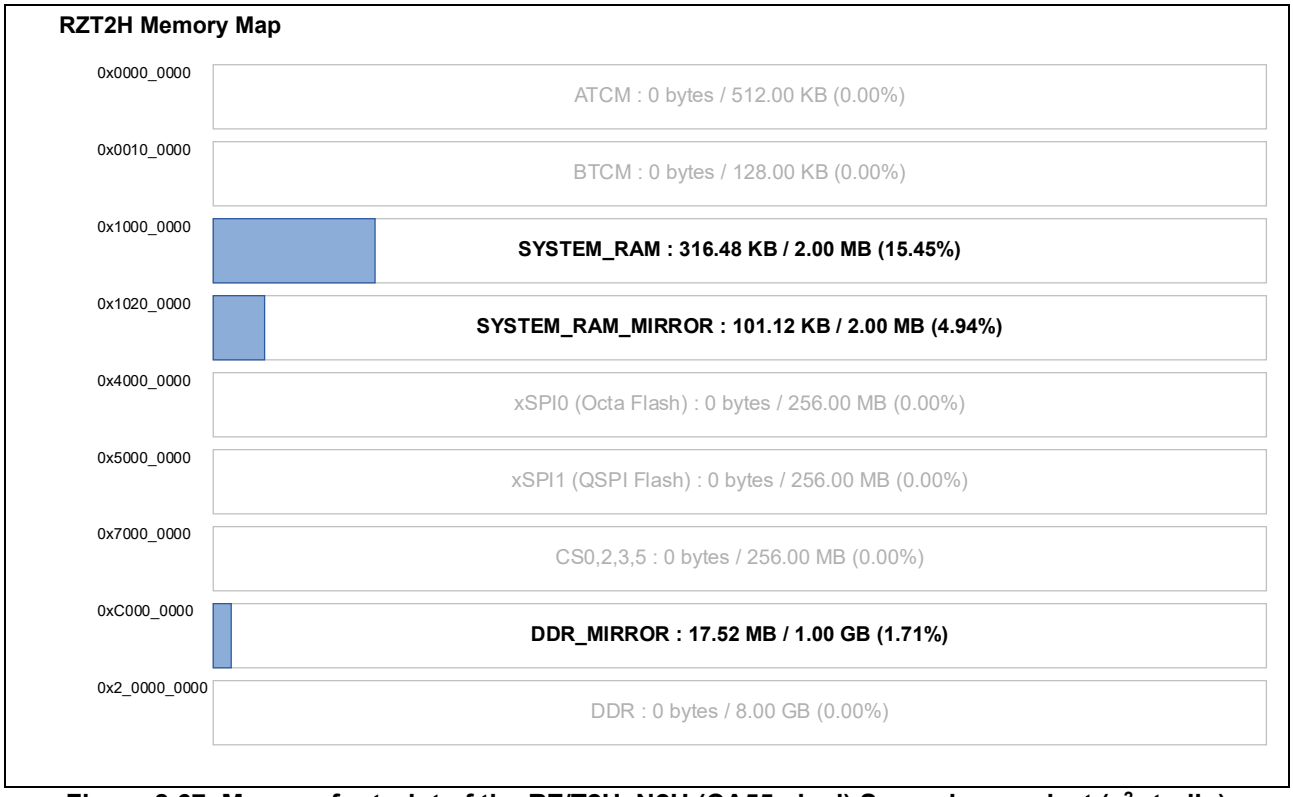
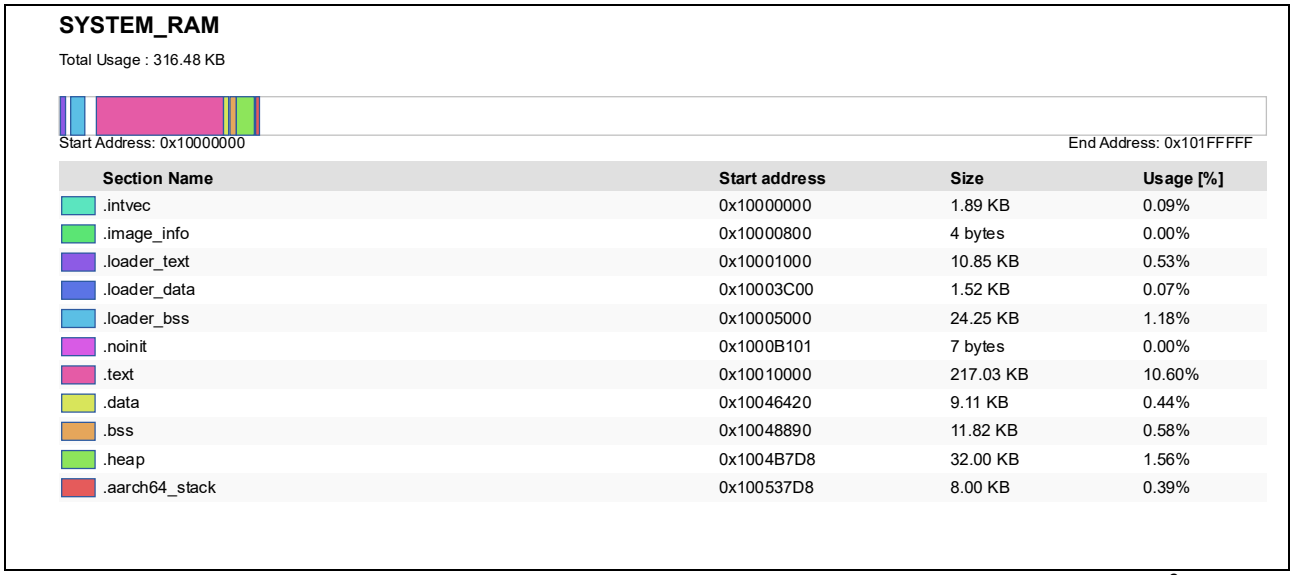


Figure 8.67. Memory footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)



Section Name	Start address	Size	Usage [%]
.intvec	0x10000000	1.89 KB	0.09%
.image_info	0x10000800	4 bytes	0.00%
.loader_text	0x10001000	10.85 KB	0.53%
.loader_data	0x10003C00	1.52 KB	0.07%
.loader_bss	0x10005000	24.25 KB	1.18%
.noinit	0x1000B101	7 bytes	0.00%
.text	0x10010000	217.03 KB	10.60%
.data	0x10046420	9.11 KB	0.44%
.bss	0x10048890	11.82 KB	0.58%
.heap	0x1004B7D8	32.00 KB	1.56%
.aarch64_stack	0x100537D8	8.00 KB	0.39%

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)

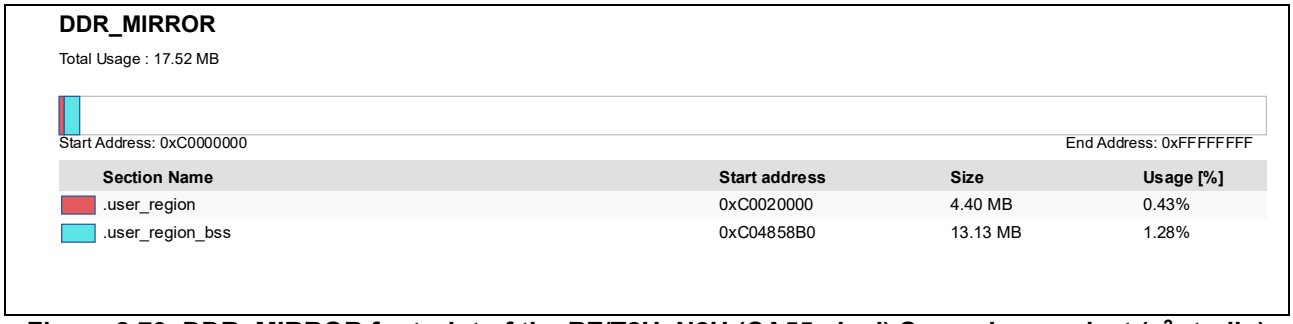


Figure 8.70. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)

8.3.4.2 EWARM

(1) Primary project

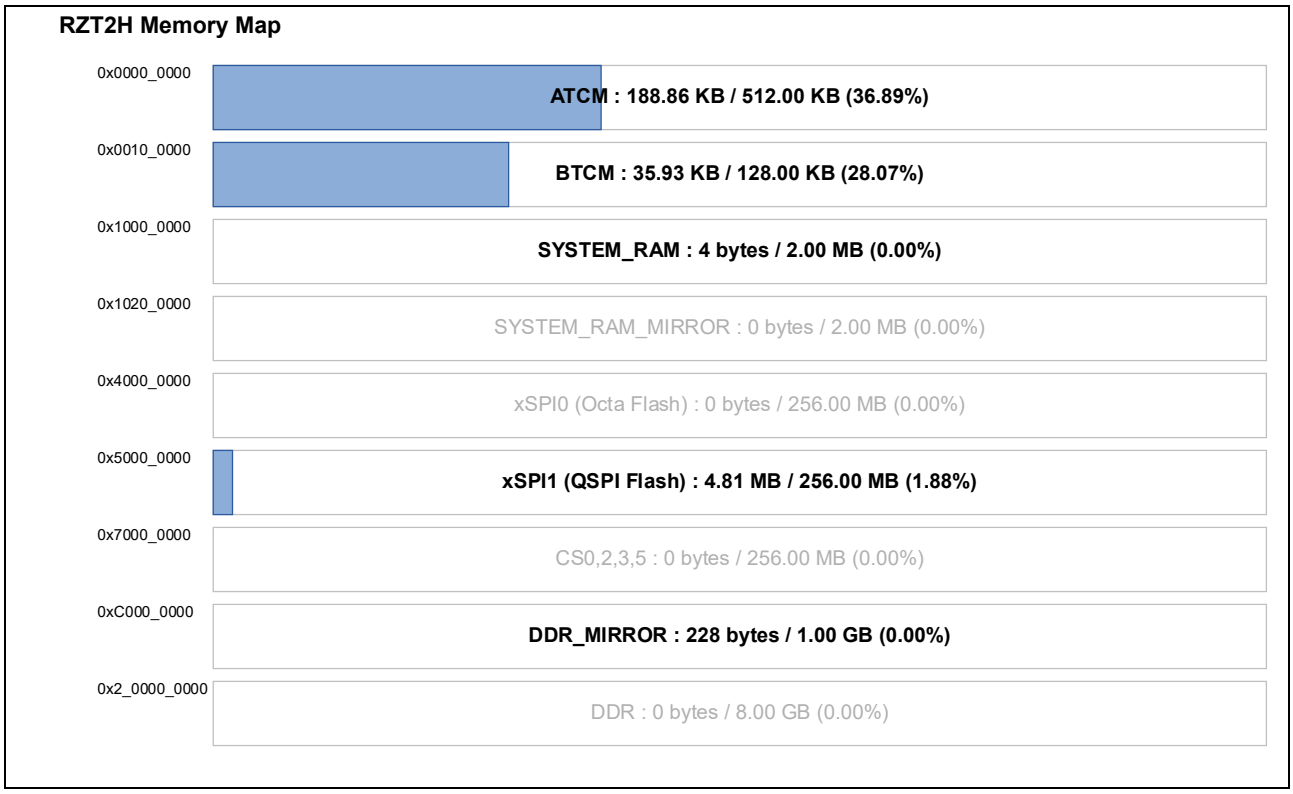
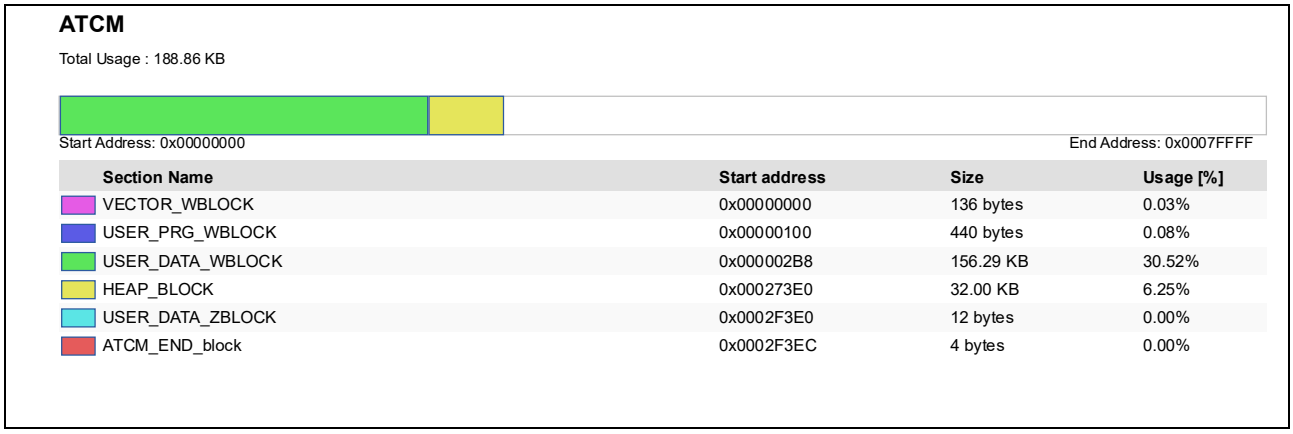
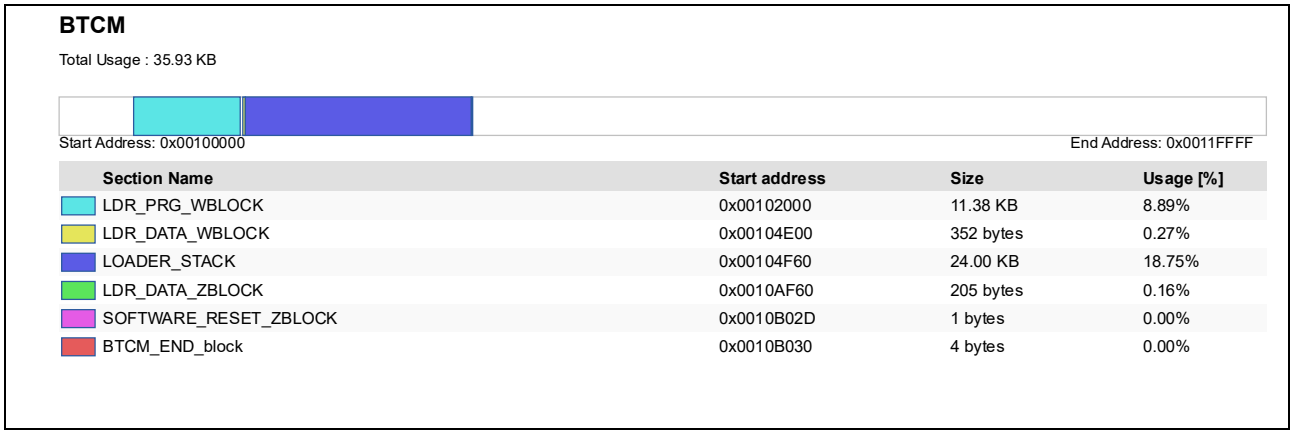


Figure 8.71. Memory footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)



Section Name	Start address	Size	Usage [%]
VECTOR_WBLOCK	0x00000000	136 bytes	0.03%
USER_PRG_WBLOCK	0x00000100	440 bytes	0.08%
USER_DATA_WBLOCK	0x000002B8	156.29 KB	30.52%
HEAP_BLOCK	0x000273E0	32.00 KB	6.25%
USER_DATA_ZBLOCK	0x0002F3E0	12 bytes	0.00%
ATCM_END_block	0x0002F3EC	4 bytes	0.00%

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



BTCM

Total Usage : 35.93 KB

Start Address: 0x00100000 End Address: 0x0011FFFF

Section Name	Start address	Size	Usage [%]
LDR_PRG_WBLOCK	0x00102000	11.38 KB	8.89%
LDR_DATA_WBLOCK	0x00104E00	352 bytes	0.27%
LOADER_STACK	0x00104F60	24.00 KB	18.75%
LDR_DATA_ZBLOCK	0x0010AF60	205 bytes	0.16%
SOFTWARE_RESET_ZBLOCK	0x0010B02D	1 bytes	0.00%
BTCM_END_block	0x0010B030	4 bytes	0.00%

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



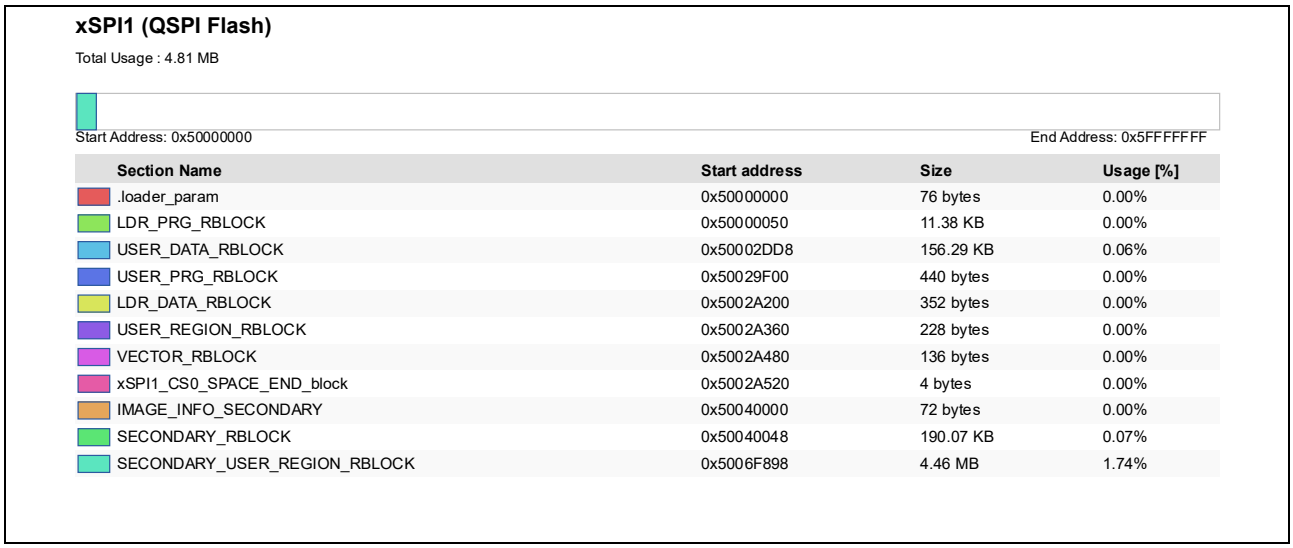
SYSTEM_RAM

Total Usage : 4 bytes

Start Address: 0x10000000 End Address: 0x101FFFFF

Section Name	Start address	Size	Usage [%]
NONCACHE_END_block	0x10180000	4 bytes	0.00%

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



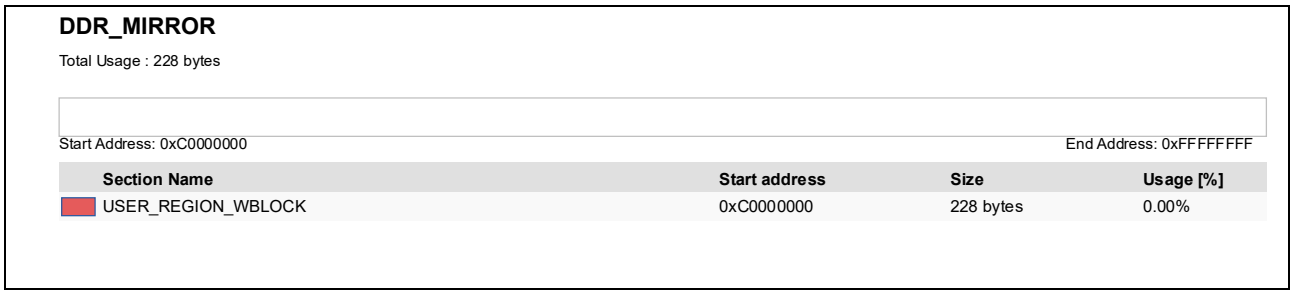
xSPI1 (QSPI Flash)

Total Usage : 4.81 MB

Start Address: 0x50000000 End Address: 0x5FFFFFFF

Section Name	Start address	Size	Usage [%]
.loader_param	0x50000000	76 bytes	0.00%
LDR_PRG_RBLOCK	0x50000050	11.38 KB	0.00%
USER_DATA_RBLOCK	0x50002DD8	156.29 KB	0.06%
USER_PRG_RBLOCK	0x50029F00	440 bytes	0.00%
LDR_DATA_RBLOCK	0x5002A200	352 bytes	0.00%
USER_REGION_RBLOCK	0x5002A360	228 bytes	0.00%
VECTOR_RBLOCK	0x5002A480	136 bytes	0.00%
xSPI1_CS0_SPACE_END_block	0x5002A520	4 bytes	0.00%
IMAGE_INFO_SECONDARY	0x50040000	72 bytes	0.00%
SECONDARY_RBLOCK	0x50040048	190.07 KB	0.07%
SECONDARY_USER_REGION_RBLOCK	0x5006F898	4.46 MB	1.74%

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



DDR_MIRROR

Total Usage : 228 bytes

Start Address: 0xC0000000 End Address: 0xFFFFFFFF

Section Name	Start address	Size	Usage [%]
USER_REGION_WBLOCK	0xC0000000	228 bytes	0.00%

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

(2) Secondary project

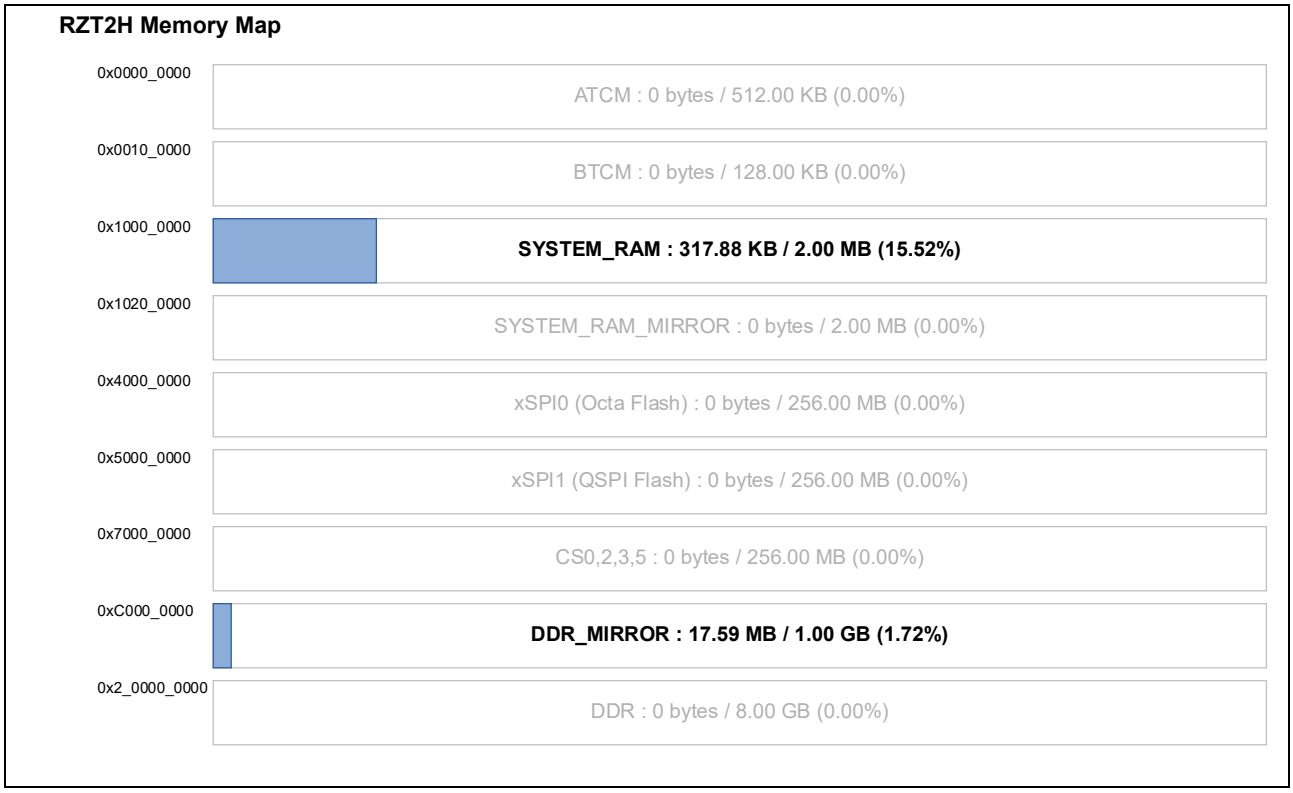
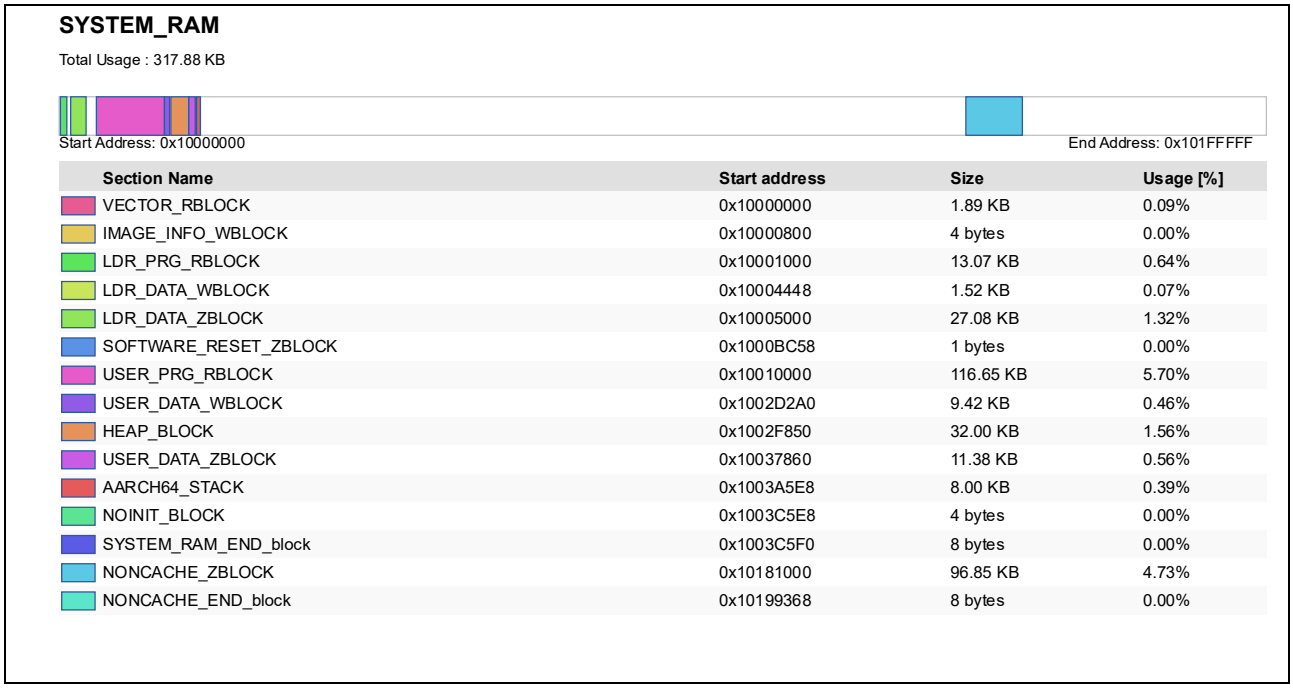


Figure 8.77. Memory footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)



Section Name	Start address	Size	Usage [%]
VECTOR_RBLOCK	0x10000000	1.89 KB	0.09%
IMAGE_INFO_WBLOCK	0x10000800	4 bytes	0.00%
LDR_PRG_RBLOCK	0x10001000	13.07 KB	0.64%
LDR_DATA_WBLOCK	0x10004448	1.52 KB	0.07%
LDR_DATA_ZBLOCK	0x10005000	27.08 KB	1.32%
SOFTWARE_RESET_ZBLOCK	0x1000BC58	1 bytes	0.00%
USER_PRG_RBLOCK	0x10010000	116.65 KB	5.70%
USER_DATA_WBLOCK	0x1002D2A0	9.42 KB	0.46%
HEAP_BLOCK	0x1002F850	32.00 KB	1.56%
USER_DATA_ZBLOCK	0x10037860	11.38 KB	0.56%
AARCH64_STACK	0x1003A5E8	8.00 KB	0.39%
NOINIT_BLOCK	0x1003C5E8	4 bytes	0.00%
SYSTEM_RAM_END_block	0x1003C5F0	8 bytes	0.00%
NONCACHE_ZBLOCK	0x10181000	96.85 KB	4.73%
NONCACHE_END_block	0x10199368	8 bytes	0.00%

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

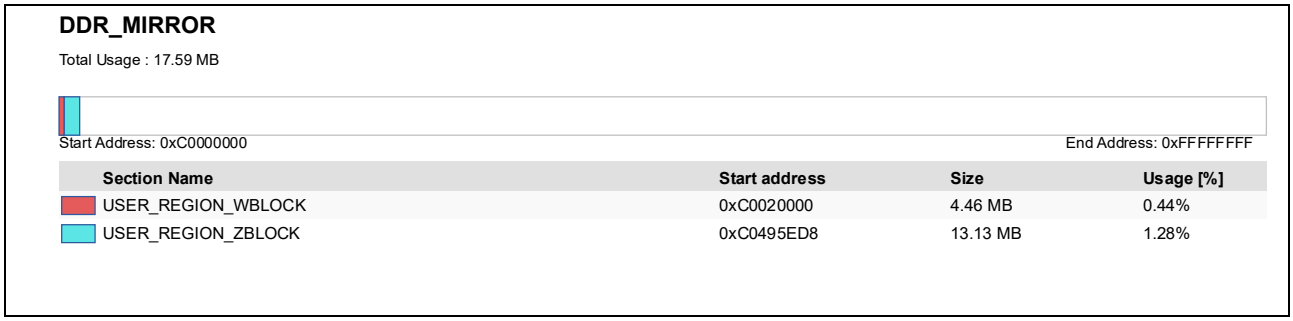


Figure 8.79. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)

9. Appendix

9.1 File Generation of open62541

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 のコンフィグレーションと情報モデル NodeSet のコンパイルを実施しています。

open62541 Documentation — open62541 1.3.0-dirty documentation

本章では、Windows 環境で open62541 および 情報モデルを e² studio もしくは EWARM で実行するために必要なファイルを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

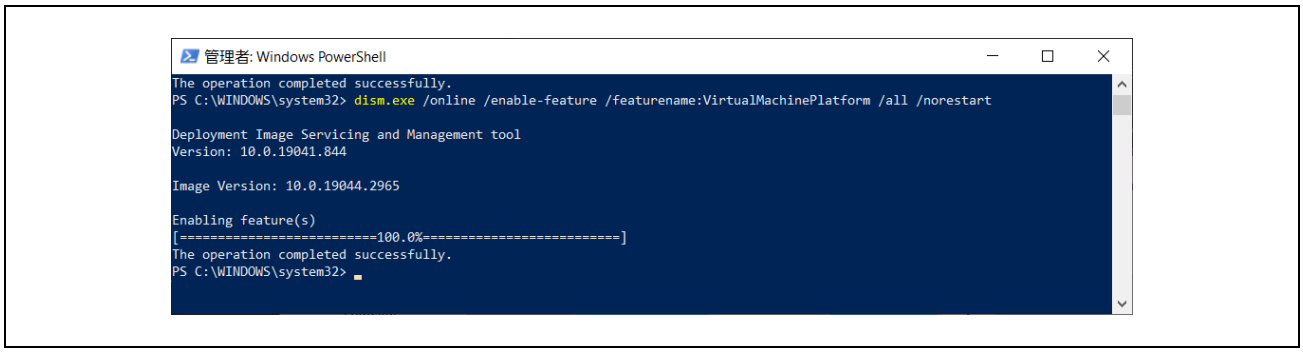


Figure 9.1. Microsoft-Windows-Subsystem-Linux

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

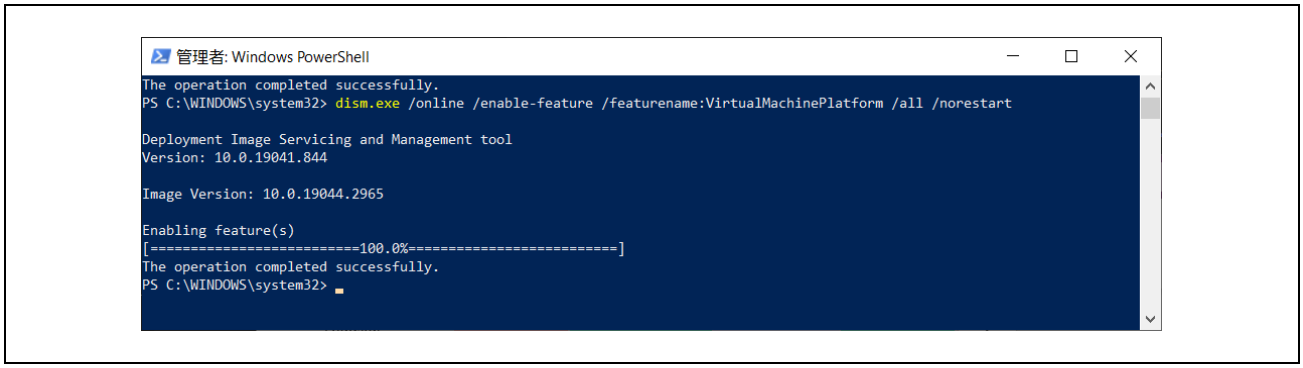


Figure 9.2. VirtualMachinePlatform

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

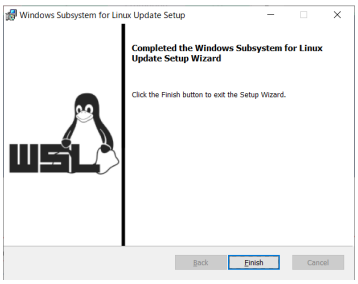


Figure 9.3. wsl_update_x64.msi

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

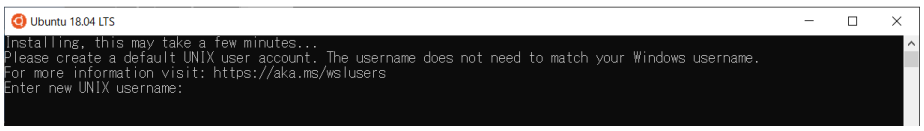


Figure 9.5. UNIX username

9.1.2 Install CMake

11) 下記 Linux コマンドを実行します。

sudo apt-get update

```
sv@JPN-5CG3013VTD:~$ 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://archive.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

```
sv@JPN-5CG3013VTD:~$ 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 libjsoncpp1 liblsan0 libmpc3 libmpx2 libpython-stdlib
  libpython2.7-minimal libpython2.7-stdlib libquadmath0 librhash0 libssl1.1 libstdc++-7-dev libstdc++6 libtsan0
  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-locales 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 libitm1-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg libtsan0-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 bzr 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。

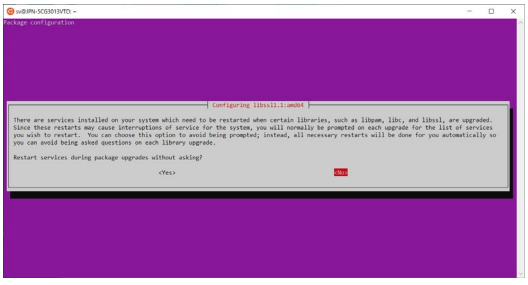


Figure 9.8. restart

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 libmbedtls-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
```

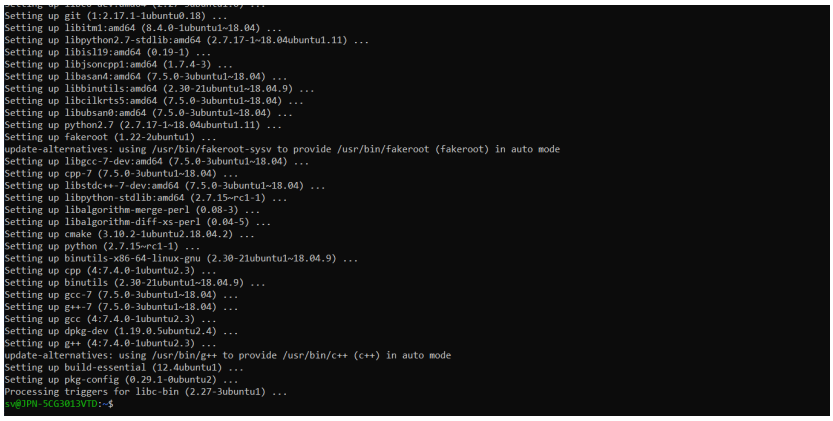


Figure 9.9. install

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
```

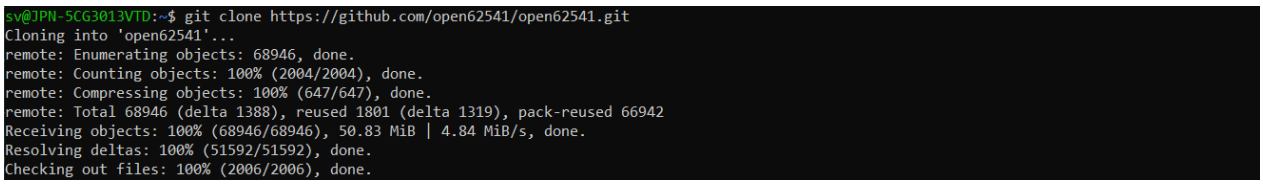


Figure 9.10. git clone

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、v1.3.18) を参照します。

```
cd open62541
git log -1
git checkout db393f2c94374ee55d89ef46e5163686932222ee
git submodule init
git submodule update
```

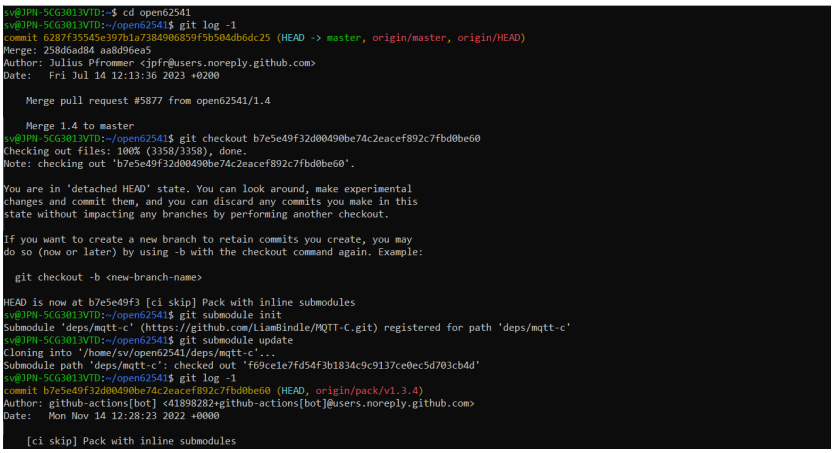


Figure 9.11. git submodule

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

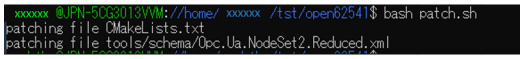


Figure 9.12. command result

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 ..

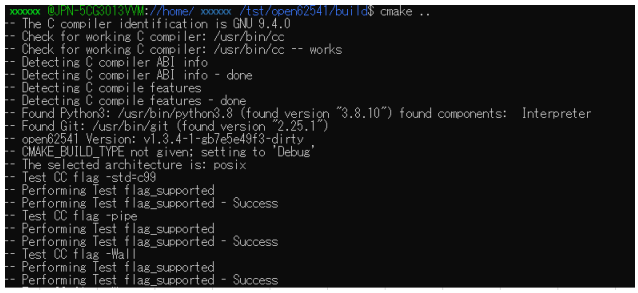


Figure 9.13. cmake

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_NODEMANAGEMENT”を OFF に、”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。

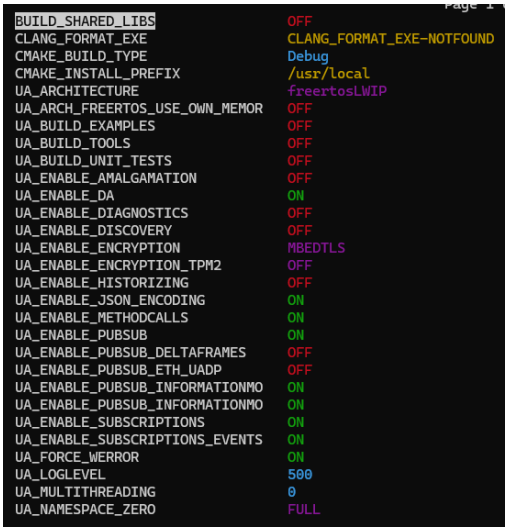


Figure 9.14. ccmake configuration

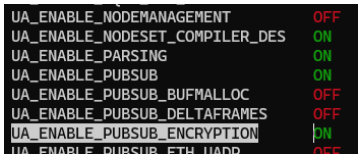


Figure 9.15. ccmake advanced configuration

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

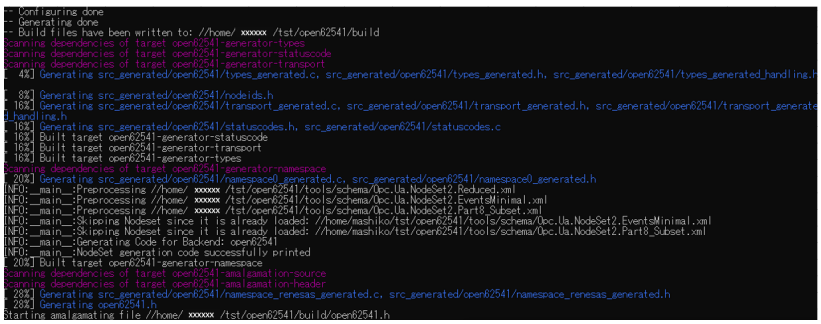


Figure 9.16. make

22) *ls* コマンドで /open62541/build/src_generated/open62541 ディレクトリにファイルが生成されていることを確認してください。

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注意ください。

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>/pki/application/own/certs 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 pki/application/own/certs/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>/pki/application/own/certs/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 pki/application/own/certs/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 9.17 に示す通りです。

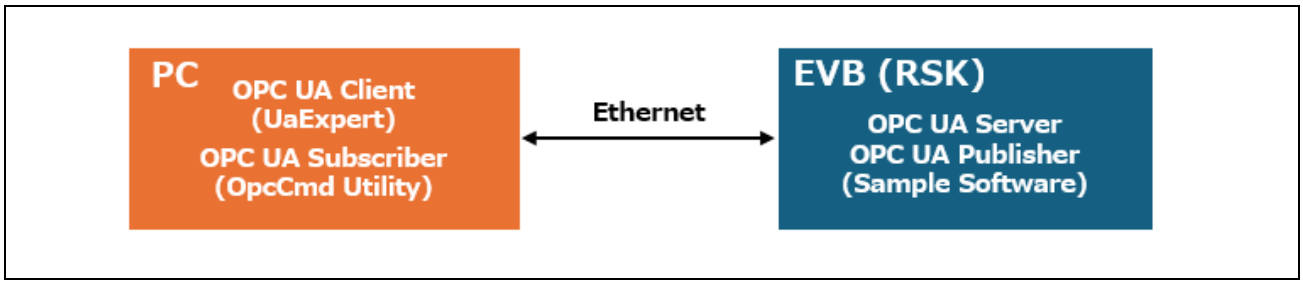


Figure 9.17. Connection Block Diagram for OPC UA Publisher

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。

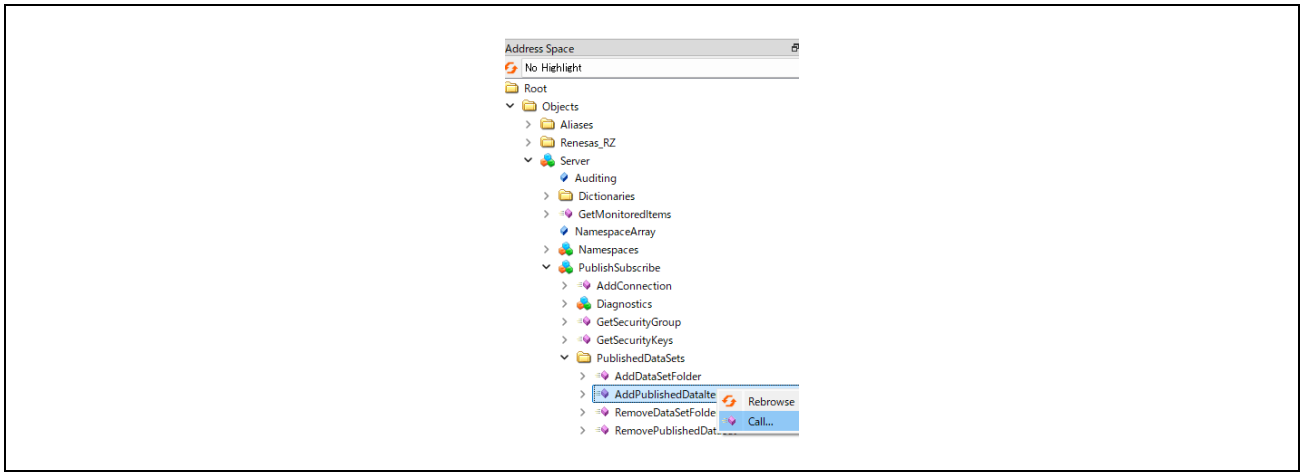


Figure 9.18. Address Space Window

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 9.19 は例として、Boolean(NS=2, ID=6049), DateTime(NS=2, ID=6052)の 2 つを設定しています。

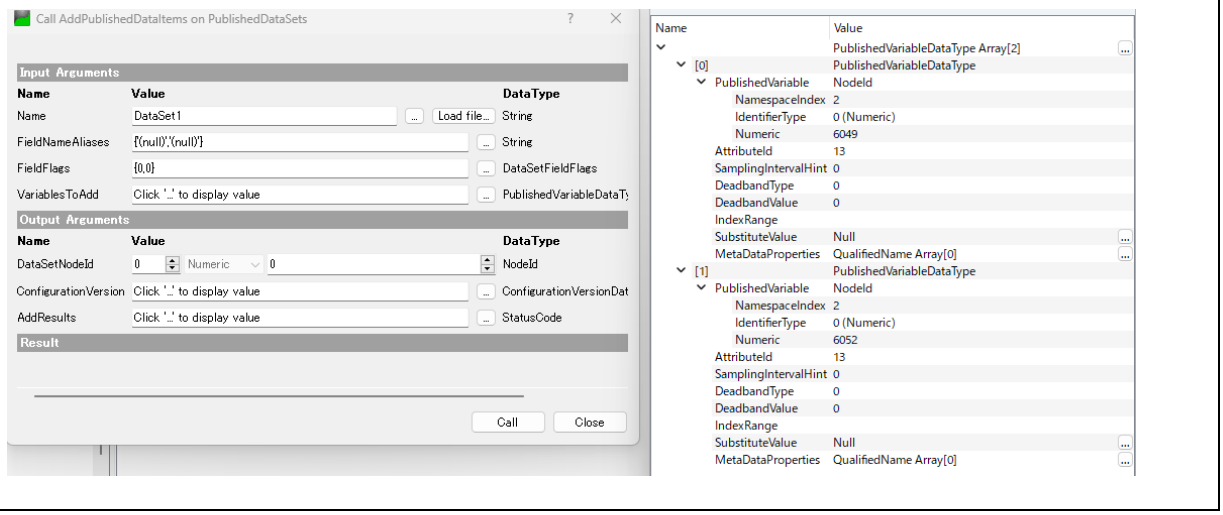


Figure 9.19. Call AddPublishedDataItems

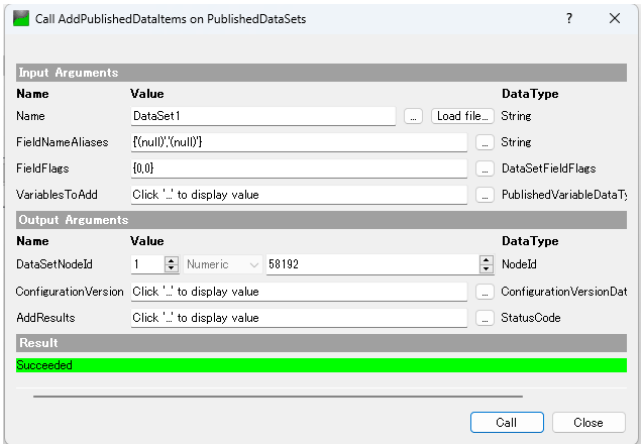


Figure 9.20. Call Result

次に、Root>Objects>Server>PublishSubscribe>AddConnection Node を右クリックし、表示された”Call…”をクリックします。

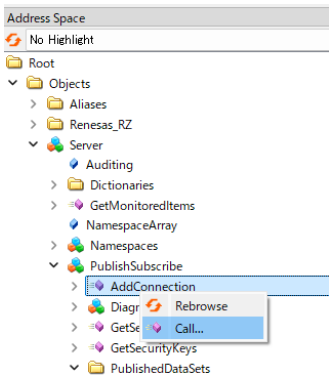


Figure 9.21. Address Space Window

表示されたウィンドウに、各項目それぞれ下記の画像のように入力し、Call をクリックします。

TransportProfileUri: http://opcfoundation.org/UA-Profile/Transport/pubsub-udp-uadp

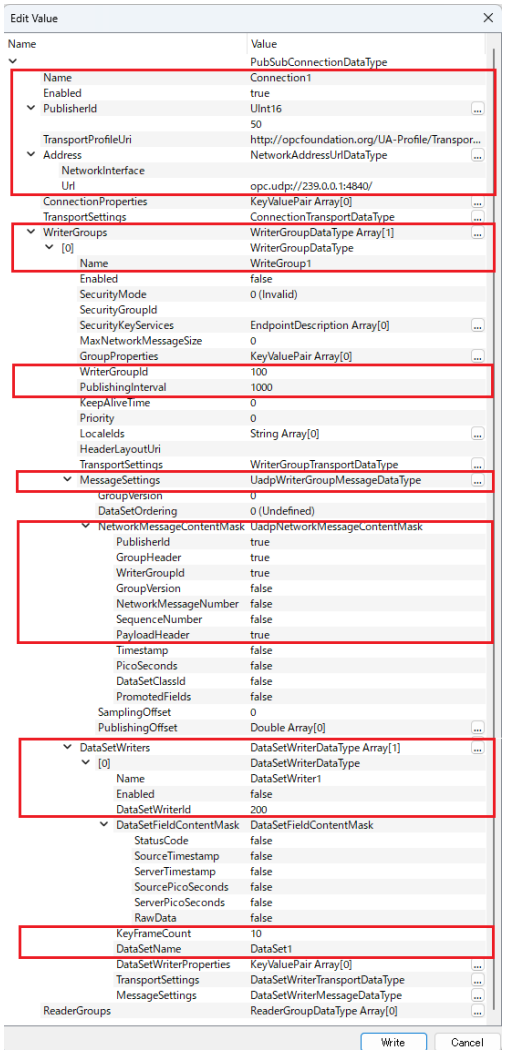


Figure 9.22. Call AddConnection

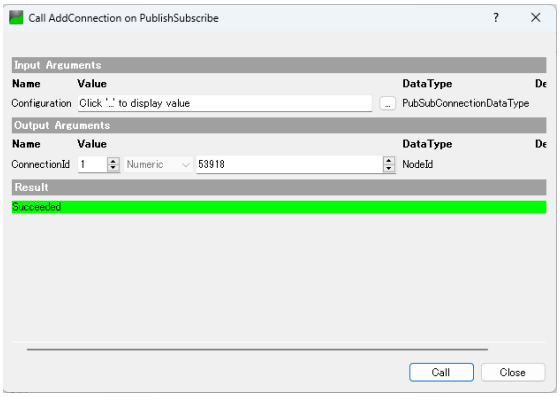


Figure 9.23. Call Result

Method の呼び出しが成功すると、Publish が開始されます。

・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 分間継続します。

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

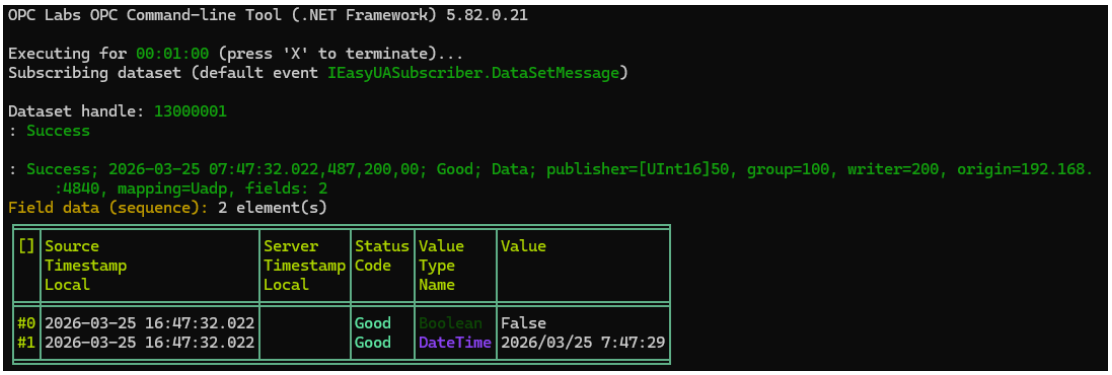


Figure 9.24. Command Execution Result

9.4.2 Subscriber

OPC UA Subscriber 動作確認時の、それぞれのデバイスの役割は Figure 9.25 に示す通りです。本テストでは 1 つの評価ボードで Publisher と Subscriber を兼用しています。

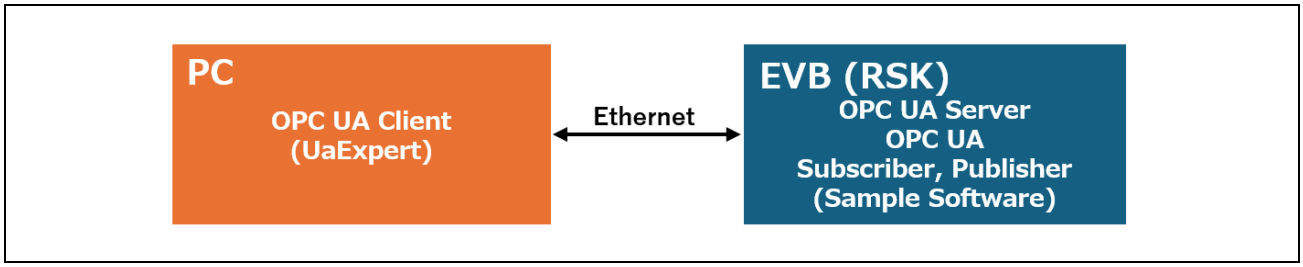


Figure 9.25. Connection Block Diagram for OPC UA Subscriber

・動作確認

9.4.1 章と同様の手順で Publish を開始させます。

次に、Root>Objects>Server>PublishSubscribe>AddConnection Node を右クリックし、表示された”Call…”をクリックします。

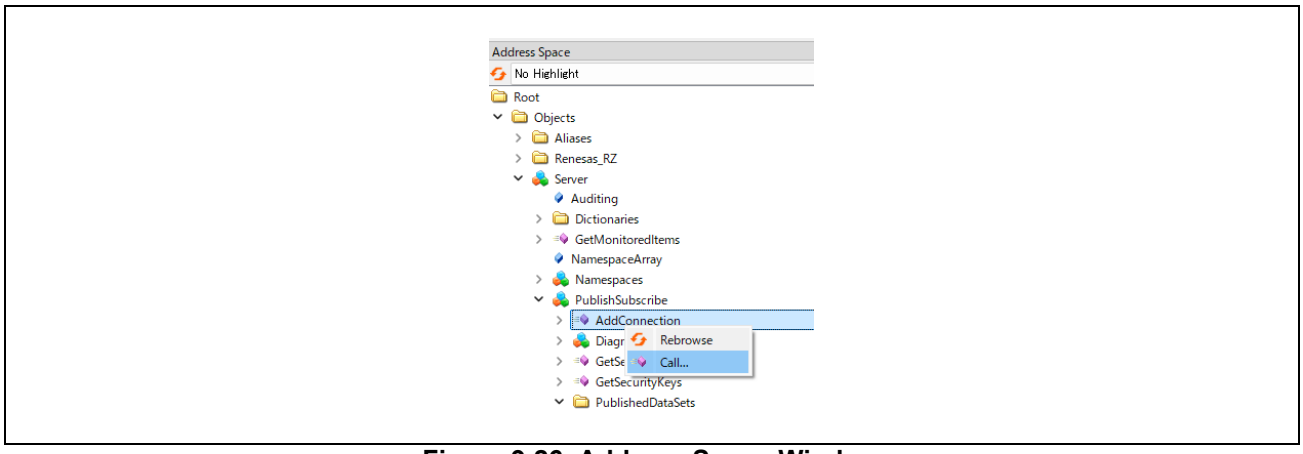


Figure 9.26. Address Space Window

表示されたウィンドウに、下記の画像のように入力し、Call をクリックします。

TransportProfileUri: http://opcfoundation.org/UA-Profile/Transport/pubsub-udp-uadp

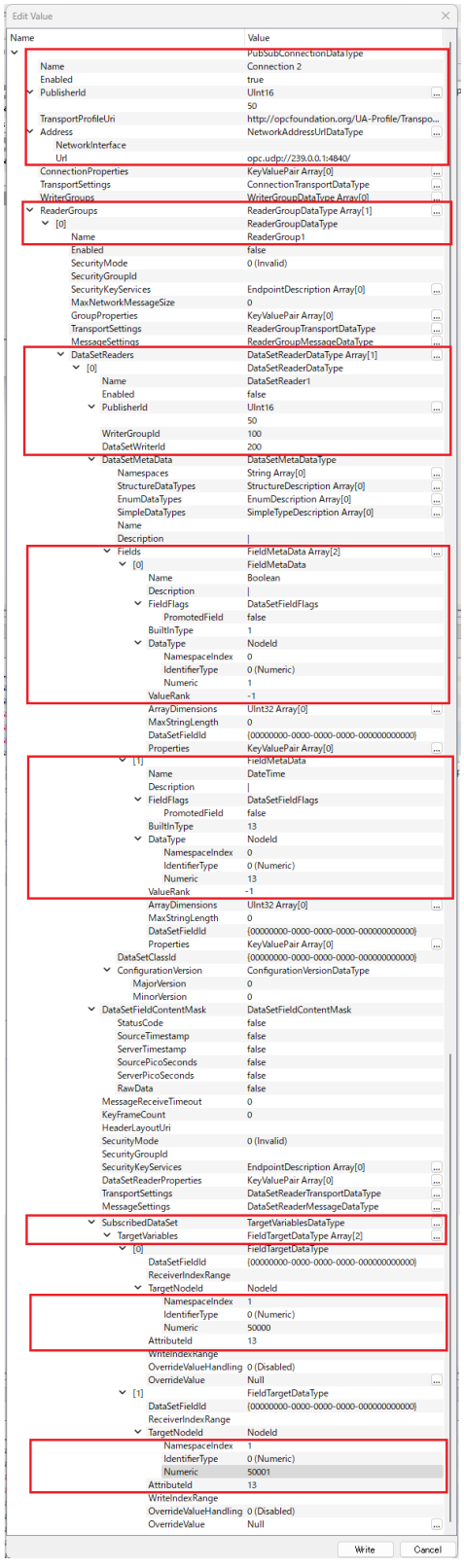


Figure 9.27. Call AddConnection

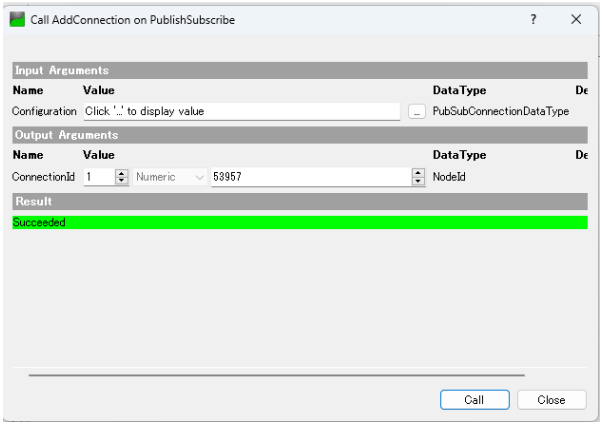


Figure 9.28. Call Result

Method の呼び出しが成功した後に、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 9.30 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。

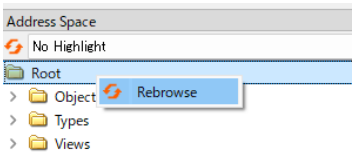


Figure 9.29. Rebrowse on UaExpert

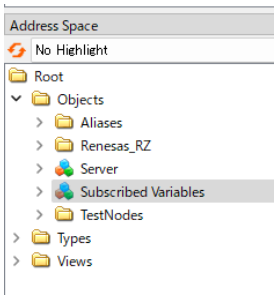
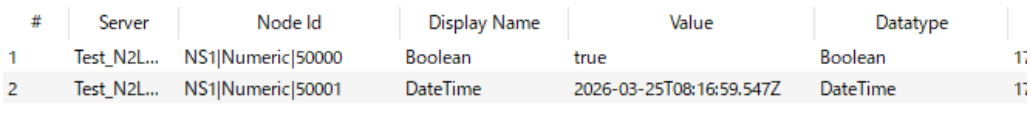


Figure 9.30. “Subscribed_Variables” Node

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。



#	Server	Node Id	Display Name	Value	Datatype	
1	Test_N2L...	NS1|Numeric|50000	Boolean	true	Boolean	17
2	Test_N2L...	NS1|Numeric|50001	DateTime	2026-03-25T08:16:59.547Z	DateTime	17

Figure 9.31. Check the added Node Values

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。

1. EWARM の” Tools -> Configure Tools…”をクリックします。
2. 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: $PROJ_DIR$

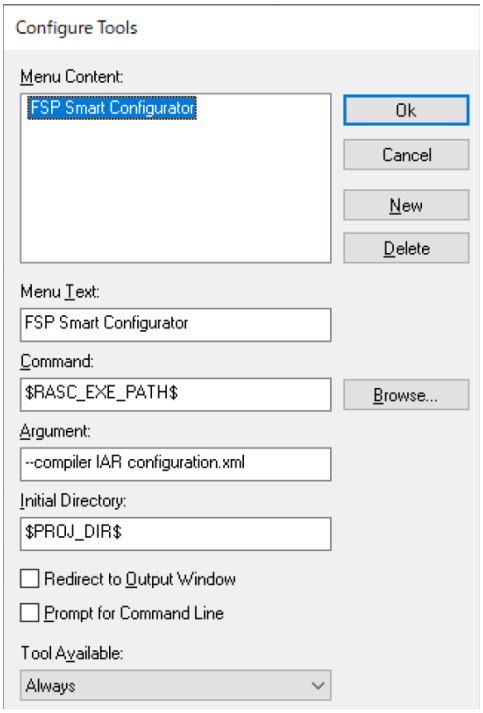


Figure 9.32. Settings “Configure Tools”

3. 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。

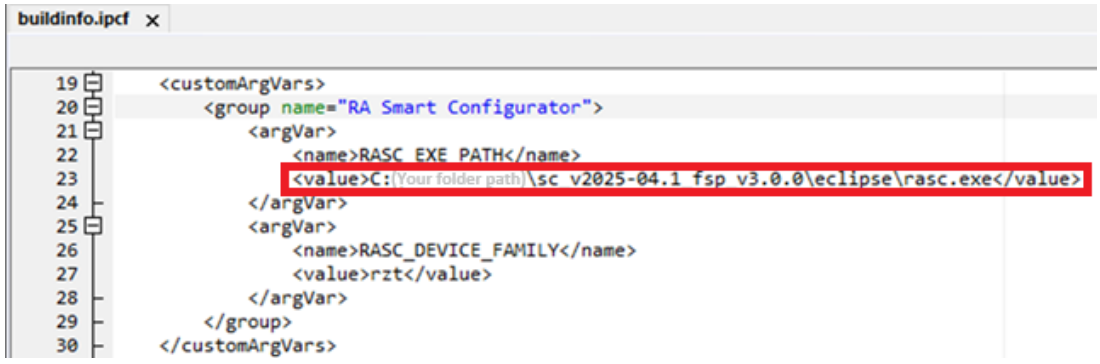


Figure 9.33. buildinfo.ipcf

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 Settings before building

9.6.1.1 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。

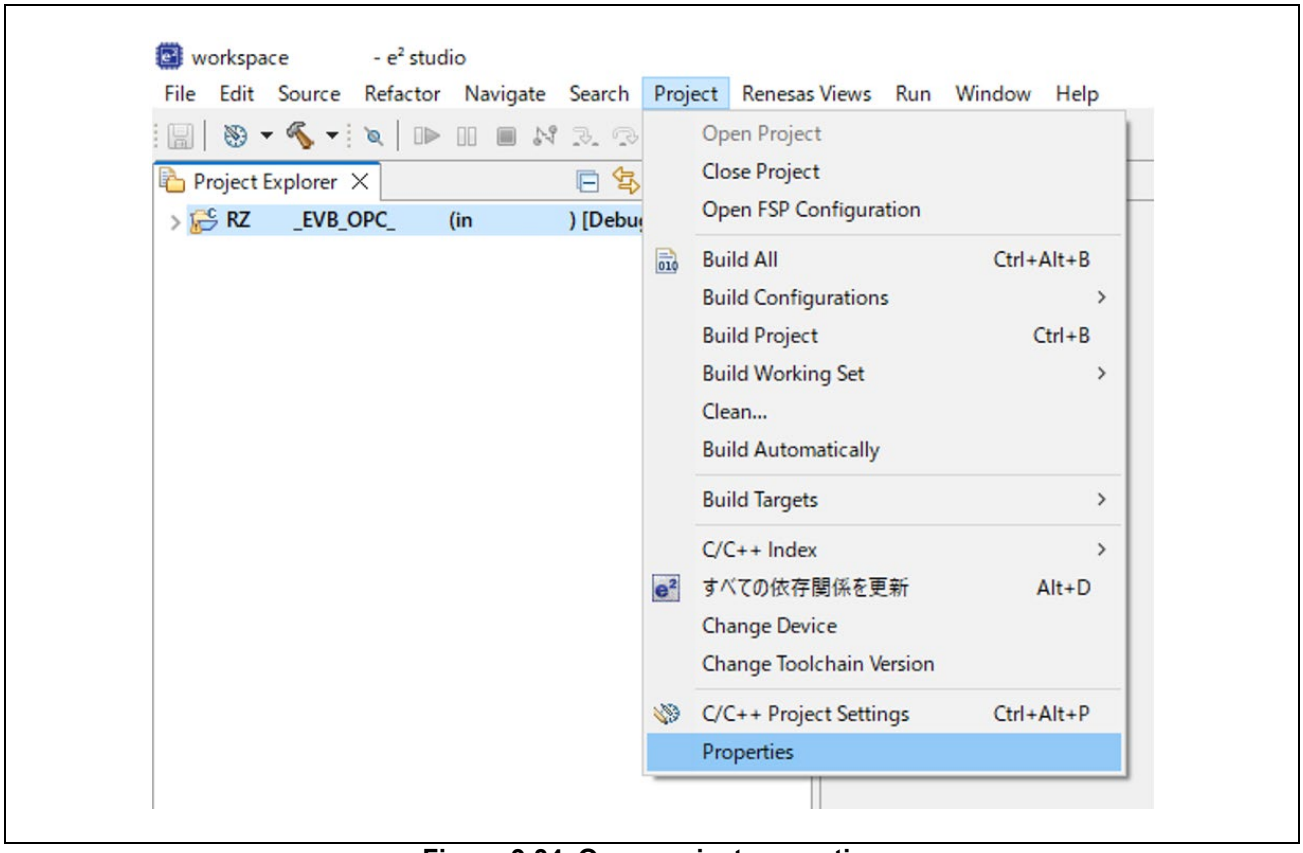


Figure 9.34. Open project properties

C/C++General > Paths and Symbols の#Symbols タブから 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 9.2 の通りです。

Table 9.2. 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
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。

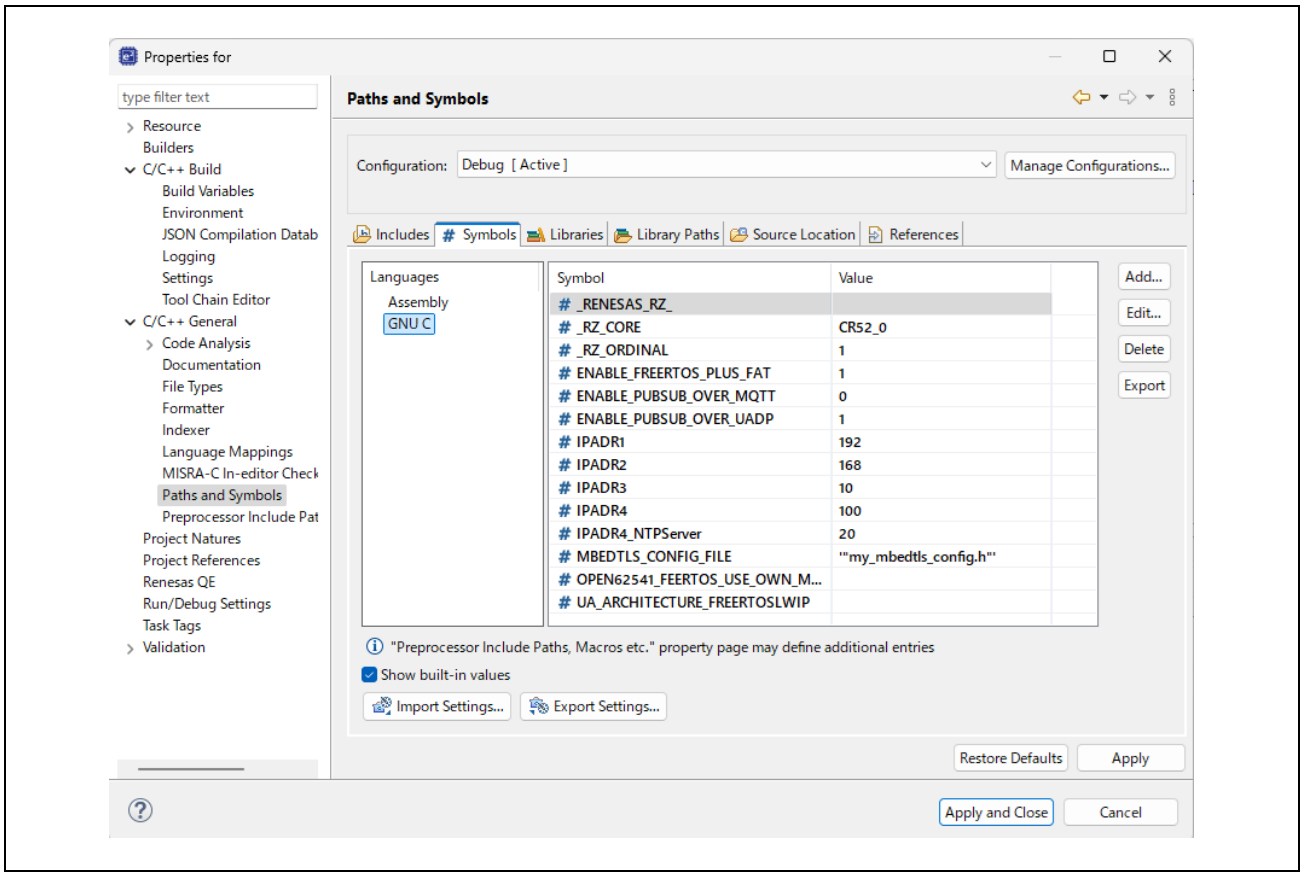


Figure 9.35. Change Build Options

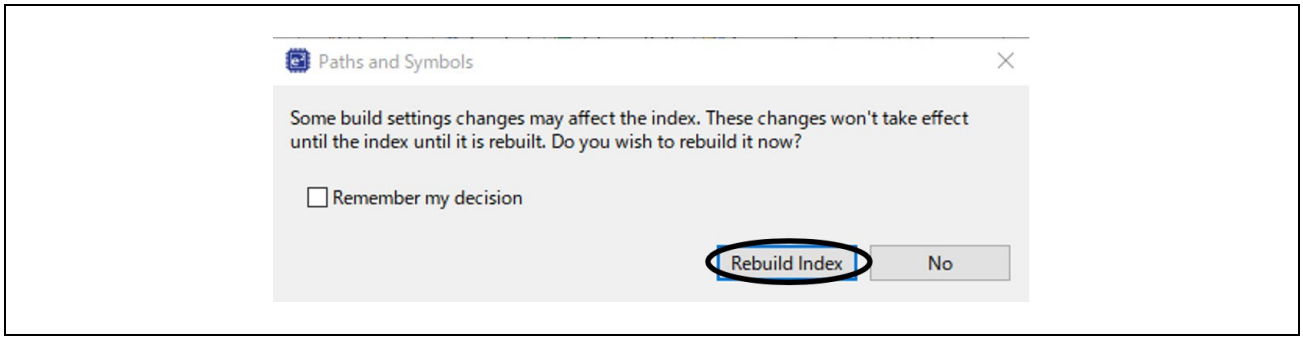


Figure 9.36. Click “Rebuild index”

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 9.37 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。

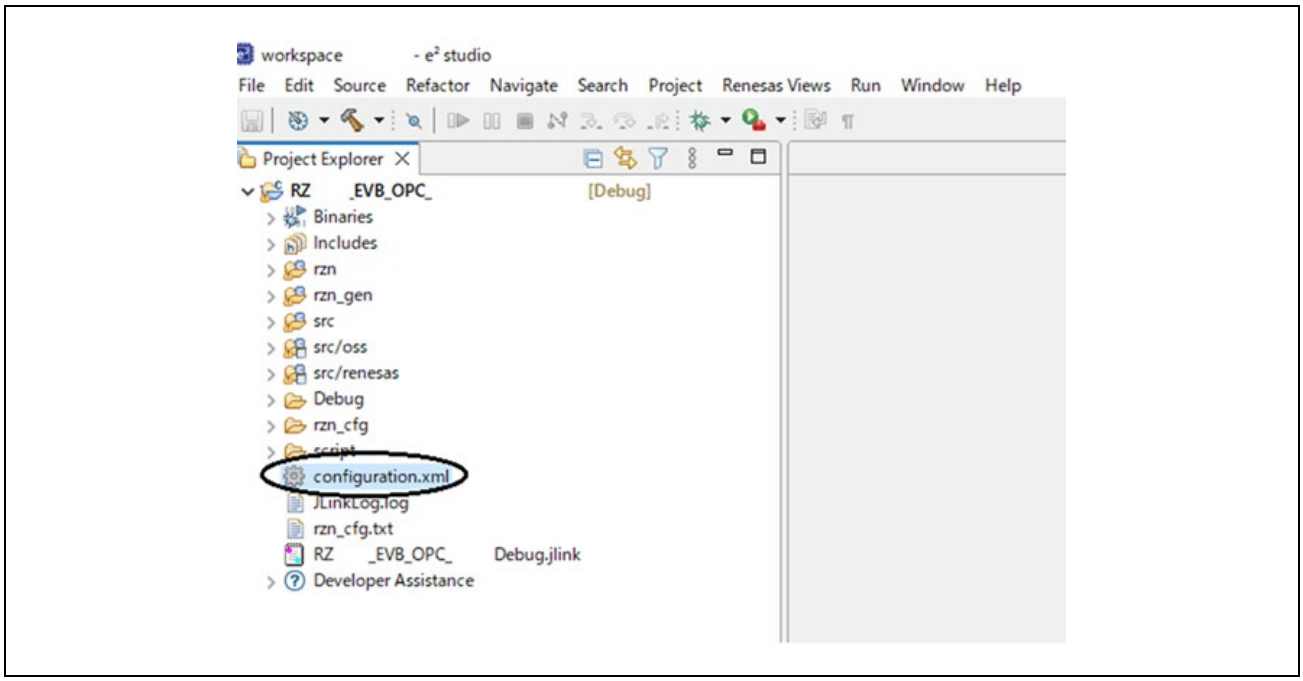


Figure 9.37. Double click configuration.xml

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

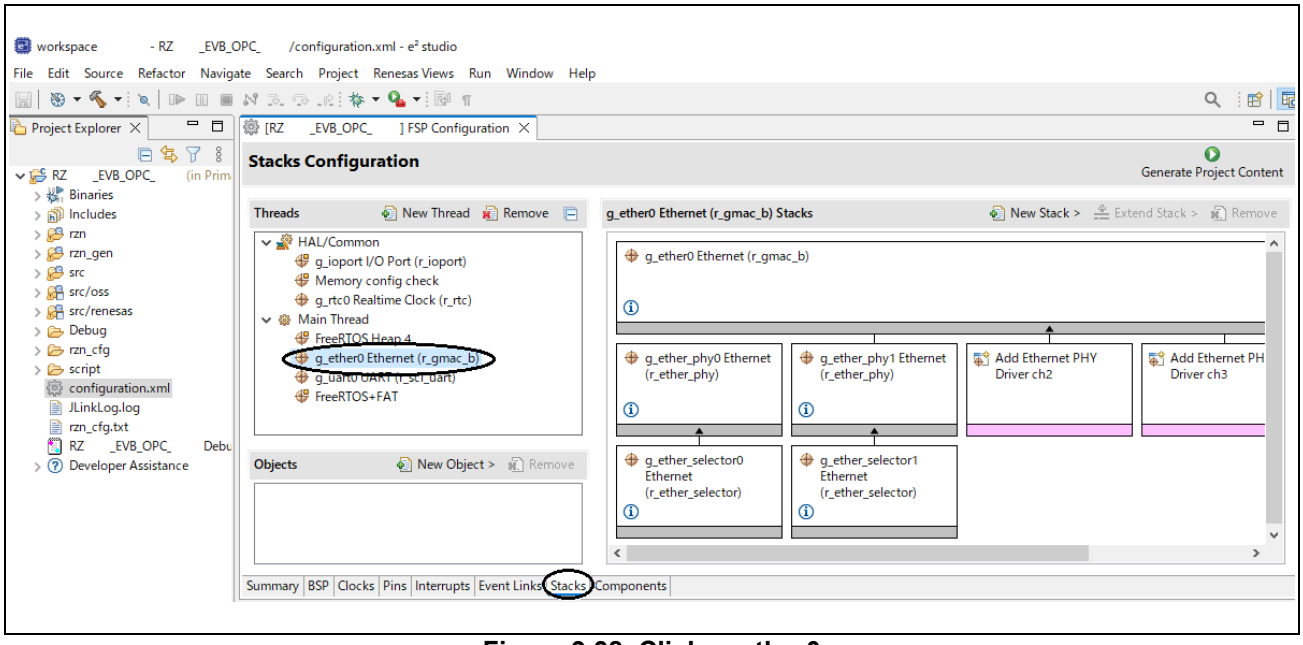


Figure 9.38. Click g_ether0

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

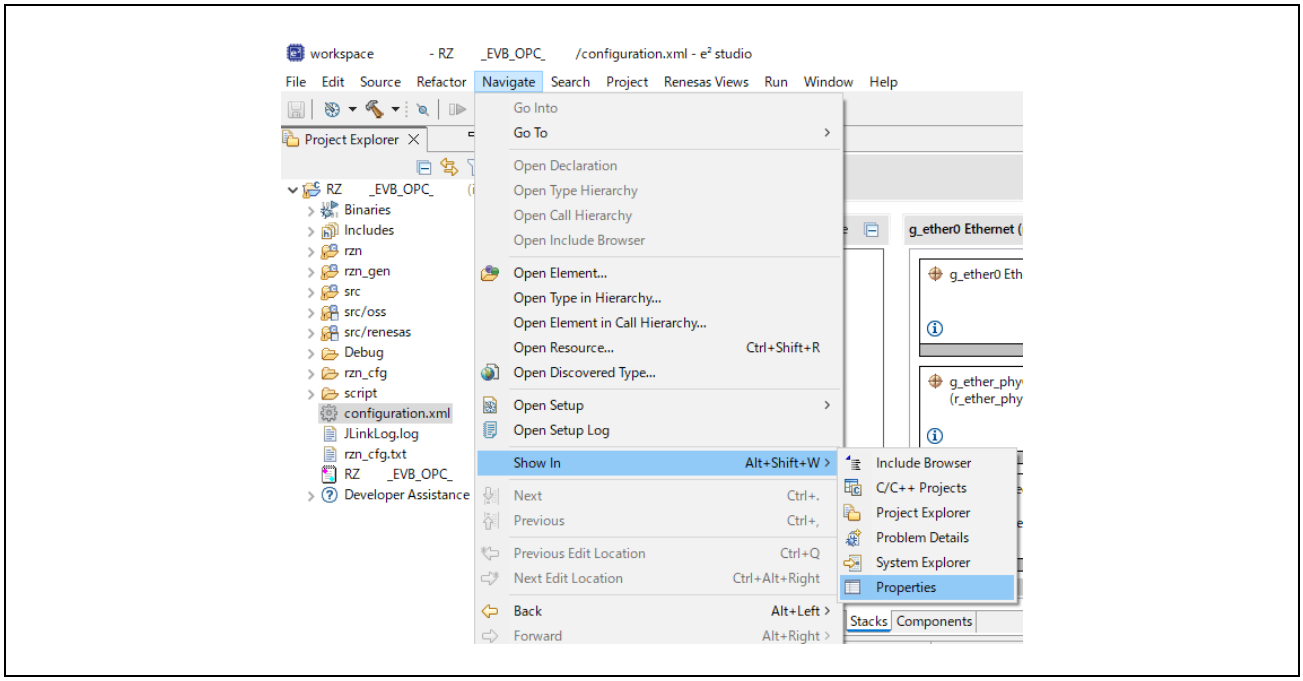


Figure 9.39. Click Properties

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。

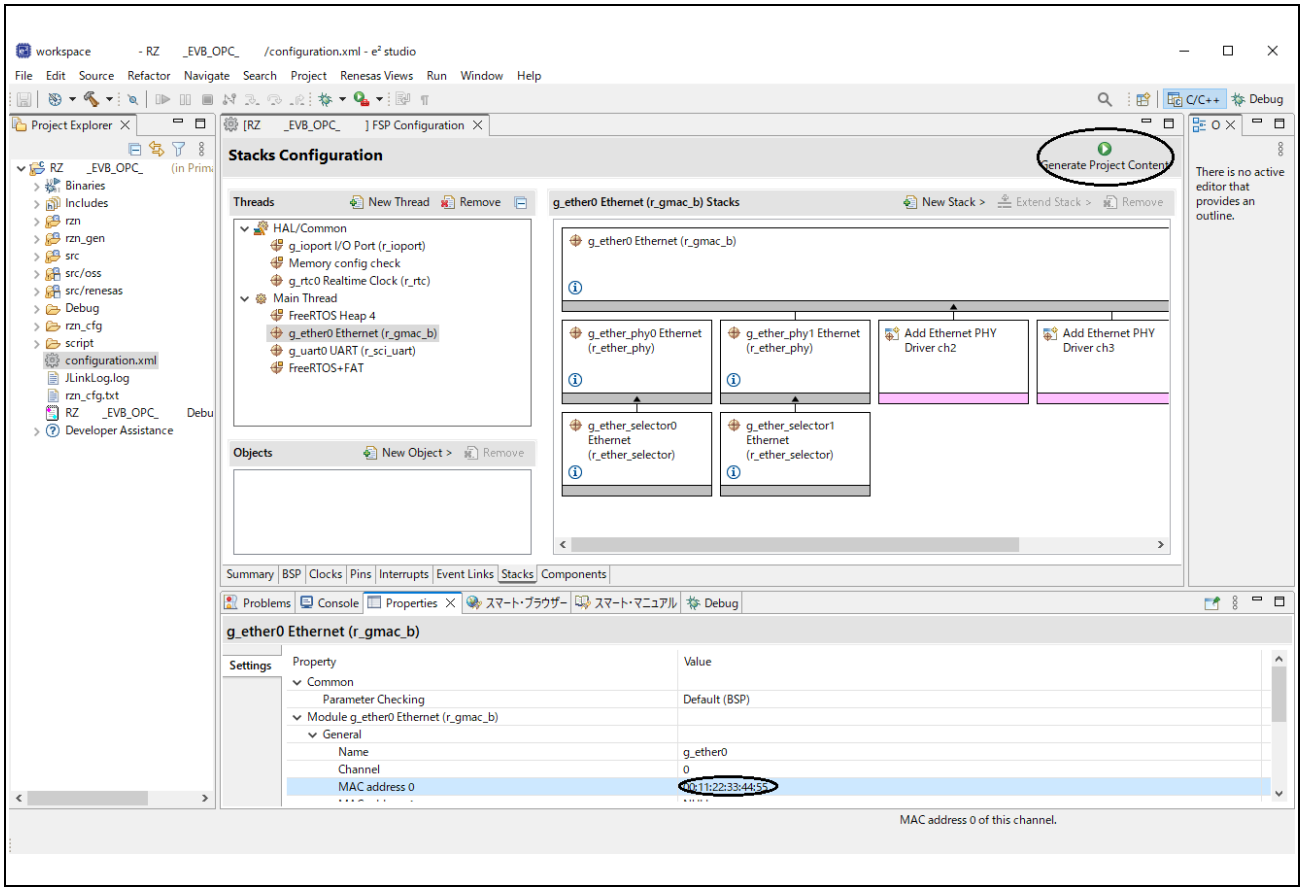


Figure 9.40. Enter MAC address

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。ただし、9.6.2 章で説明される、pubsub.cfg ファイルの設定が優先されることに注意してください。

Table 9.3. 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

```
#define UADP_FIXED_LAYOUT   1
#define UADP_DYNAMIC_LAYOUT 2
#define UADP_LAYOUT         UADP_FIXED_LAYOUT     /* Select Fixed or Dynamic */

#define PUBSUB_EXAMPLE_DISABLE          0       /* Diable pubsub example config */
#define PUBSUB_EXAMPLE_ENABLE           1       /* Enable pubusub example config without security */
#define PUBSUB_EXAMPLE_ENABLE_AES128    2       /* Enable pubusub example config with security (PubSub-Aes128-CTR) */
#define PUBSUB_EXAMPLE_ENABLE_AES256    3       /* Enable pubusub example config with security (PubSub-Aes256-CTR) */

#define ENABLE_PUBSUB_EXAMPLE_SETTING   PUBSUB_EXAMPLE_DISABLE   /* Select 0-3 */
```

Figure 9.41. opc_pubsub_uadp.h

9.6.1.2 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。

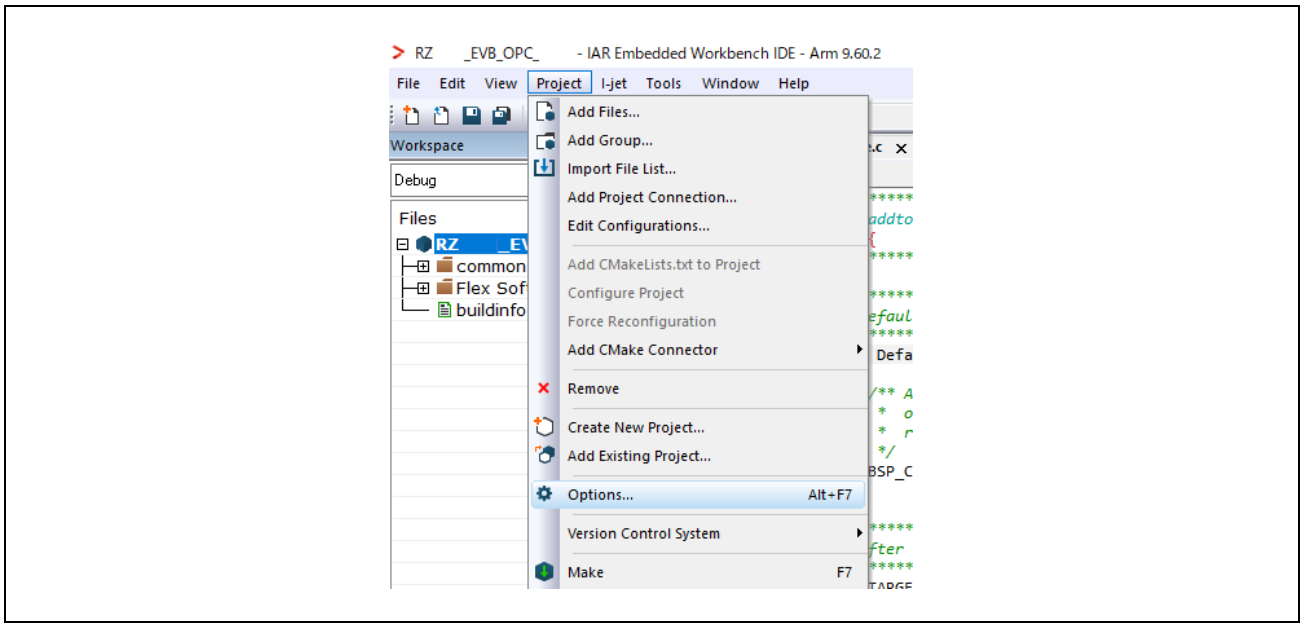


Figure 9.42. Open Options

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 9.4 の通りです。

Table 9.4. 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
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。

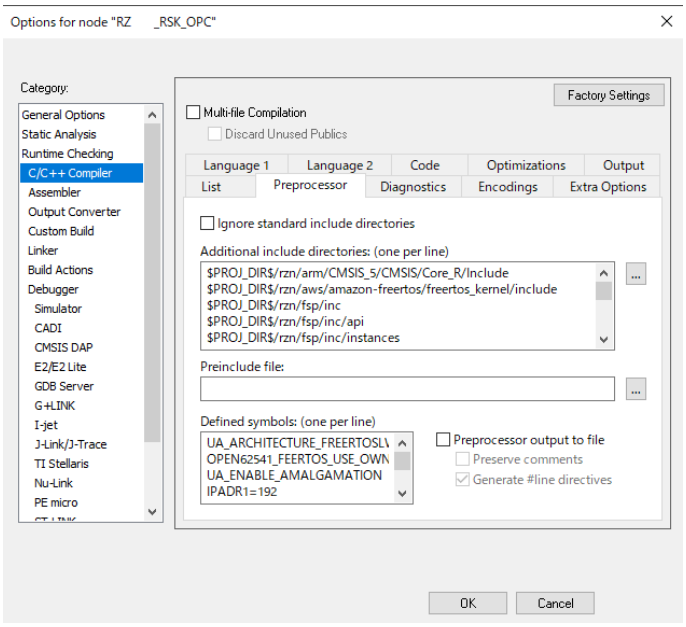


Figure 9.43. Change Build Options

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。

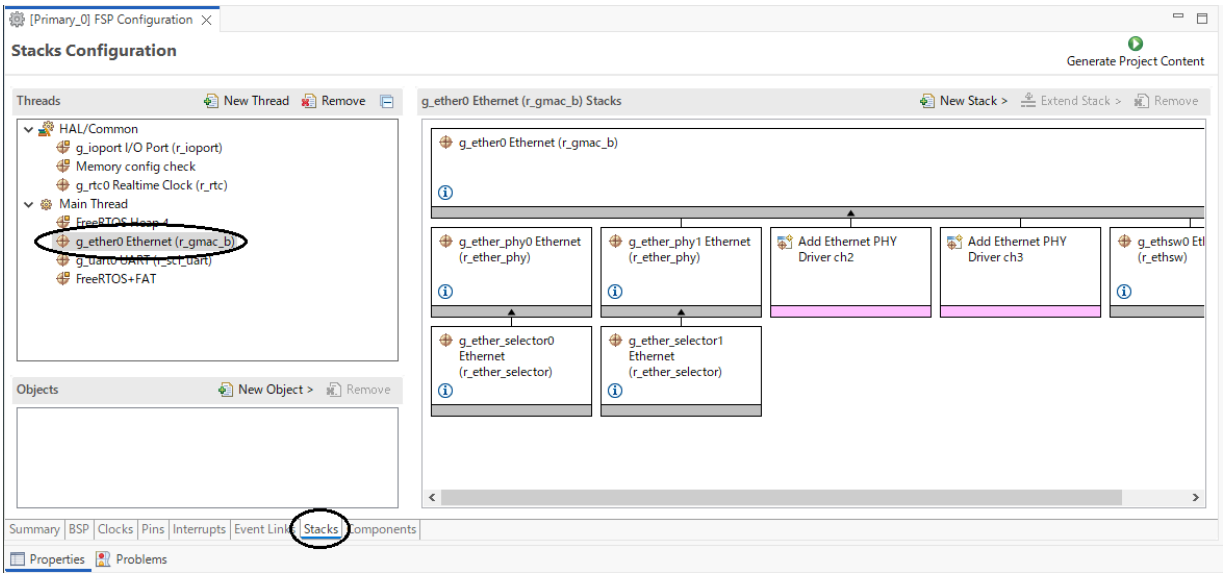


Figure 9.44. Click g_ether0

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

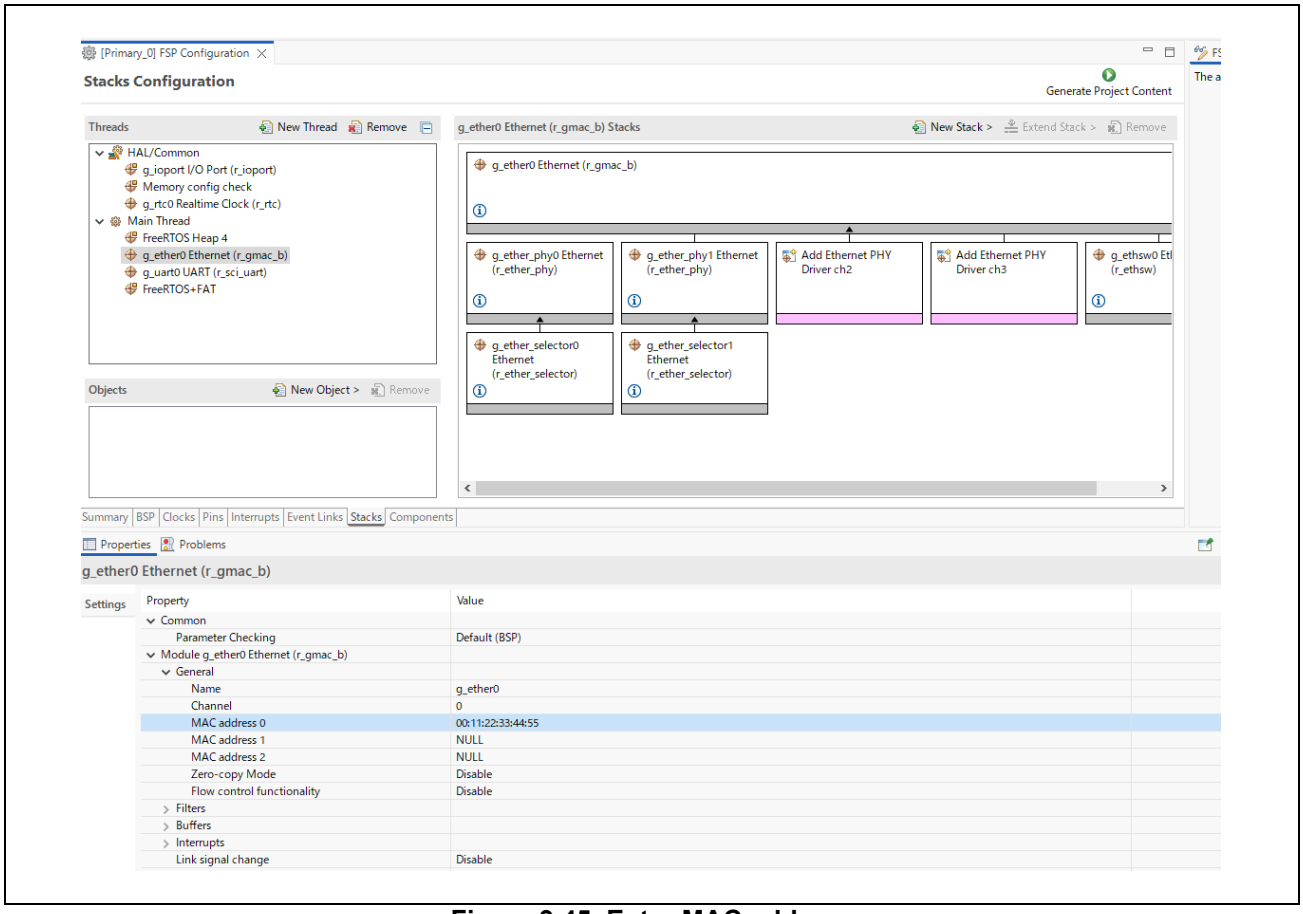


Figure 9.45. Enter MAC address

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。

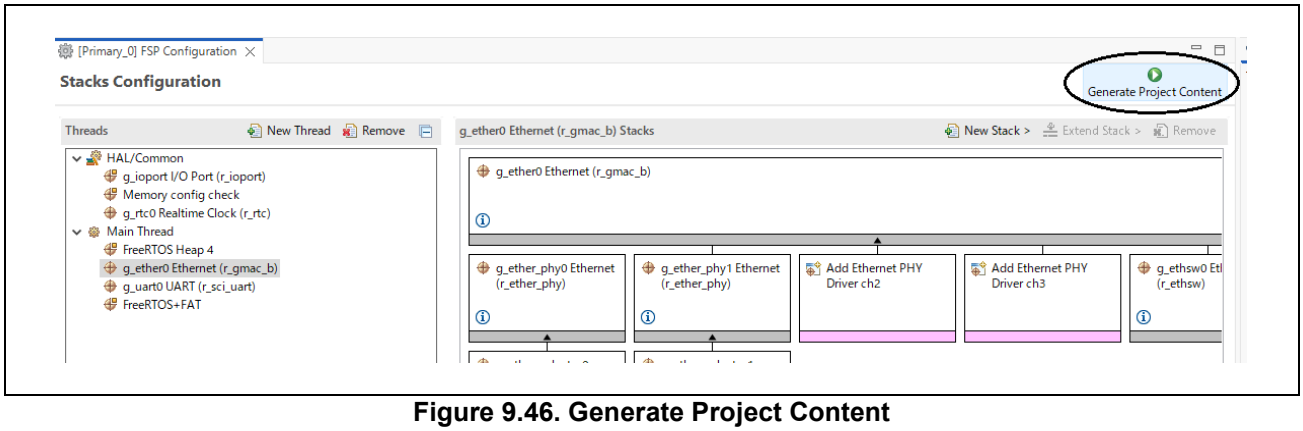


Figure 9.46. Generate Project Content

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。ただし、9.6.2 章で説明される、pubsub.cfg ファイルの設定が優先されることに注意してください。

Table 9.5. 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

```
#define UADP_FIXED_LAYOUT   1
#define UADP_DYNAMIC_LAYOUT 2
#define UADP_LAYOUT         UADP_FIXED_LAYOUT     /* Select Fixed or Dynamic */

#define PUBSUB_EXAMPLE_DISABLE          0       /* Diable pubsub example config */
#define PUBSUB_EXAMPLE_ENABLE           1       /* Enable pubusub example config without security */
#define PUBSUB_EXAMPLE_ENABLE_AES128    2       /* Enable pubusub example config with security (PubSub-Aes128-CTR) */
#define PUBSUB_EXAMPLE_ENABLE_AES256    3       /* Enable pubusub example config with security (PubSub-Aes256-CTR) */

#define ENABLE_PUBSUB_EXAMPLE_SETTING   PUBSUB_EXAMPLE_DISABLE   /* Select 0-3 */
```

Figure 9.47. opc_pubsub_uadp.h

9.6.2 Settings after building

ビルド後に、コンフィグファイル（pubsub.cfg、ip.cfg）によって機能を動的切り替える場合は、USB フラッシュドライブに保存してある各ファイルを変更後、評価ボードを再起動してください。

IP アドレスを切り替える場合は、ip.cfg の IP_MODE を FILE に、IP と NTP_SERVER_IP を任意の値に変更します。ファイルから無効な値が読み出された、もしくはファイルが無い場合は、ビルド時に設定されたデフォルトの IP アドレスが使用されます。

この機能によって IP アドレスを切り替えた場合、起動時に Server 用の証明書ファイルが生成されるため、通常よりも起動に時間がかかることに注意してください。

```
[NETWORK]
# IP address selection mode
# DEFAULT : Use compile-time default IP address
# FILE    : Use IP address defined below
IP_MODE=FILE

# Used only when IP_MODE=FILE
IP=192.168.10.101
NTP_SERVER_IP=192.168.10.21
```

PubSub の設定を動的に切り替える場合は、pubsub.cfg の” EXAMPLE”と” LAYOUT” を任意の値に変更します。9.6.1 章で説明したソースコード上のマクロ定義よりも pubsub.cfg の設定値が優先されます。ファイルから無効な値が読み出された、もしくはファイルが無い場合は、ビルド時に設定されたデフォルトの PubSub 設定が使用されます。

```
[PUBSUB]
# PubSub Example Enable or Disable
# DISABLE : Disable pubsub example config
# ENABLE : Enable pubsub example config without security
# ENABLE_AES128 : Enable pubsub example config with security (PubSub-Aes128-CTR)
# ENABLE_AES256 : Enable pubsub example config with security (PubSub-Aes256-CTR)
EXAMPLE=ENABLE

# Layout
# FIXED : Periodic Fixed Layout
# DYNAMIC : Dynamic Layout
LAYOUT=FIXED
```

9.7 Notes when using newer FSP version

RZ/T2M, RZ/N2L で FSP v4.1.0 以降を使用時、もしくは RZ/T2H, RZ/N2H で FSP v4.2.0 以降を使用時は本章のガイドを参照ください。

1. FSP のコード生成を行うと、(Project Location)/script/に格納されている下記のリンカスクリプトファイルとシェルスクリプトファイルが上書きされてしまいます。ビルドには上書きされる前のオリジナルファイルを使用してください。オリジナルファイルはパッケージに同梱されている AttachedFiles/original_scripts.zip にも格納されています。

   ・For e² studio projects:
   - ・fsp_nor_flash_boot.ld (RZ/T2M, RZ/N2L only)
   - ・fsp_xspi1_boot.ld (RZ/T2H, RZ/N2H only)
   - ・postbuild.sh (RZ/T2M, RZ/T2H, RZ/N2H only)

   ・For EWARM projects:
   - ・fsp_nor_flash_boot.icf (RZ/T2M, RZ/N2L only)
   - ・fsp_xspi1_boot.icf (RZ/T2H, RZ/N2H only)

2. e² studio かつ CR52 プロジェクトの場合、FSP コード生成後ビルドする前に Properties から下記の設定を変更してください。ただし、Dual コアプロジェクトの場合、セカンダリプロジェクト側を変更してください。

   Use newlib-nano: Unchecked

   Do not use syscalls: Checked

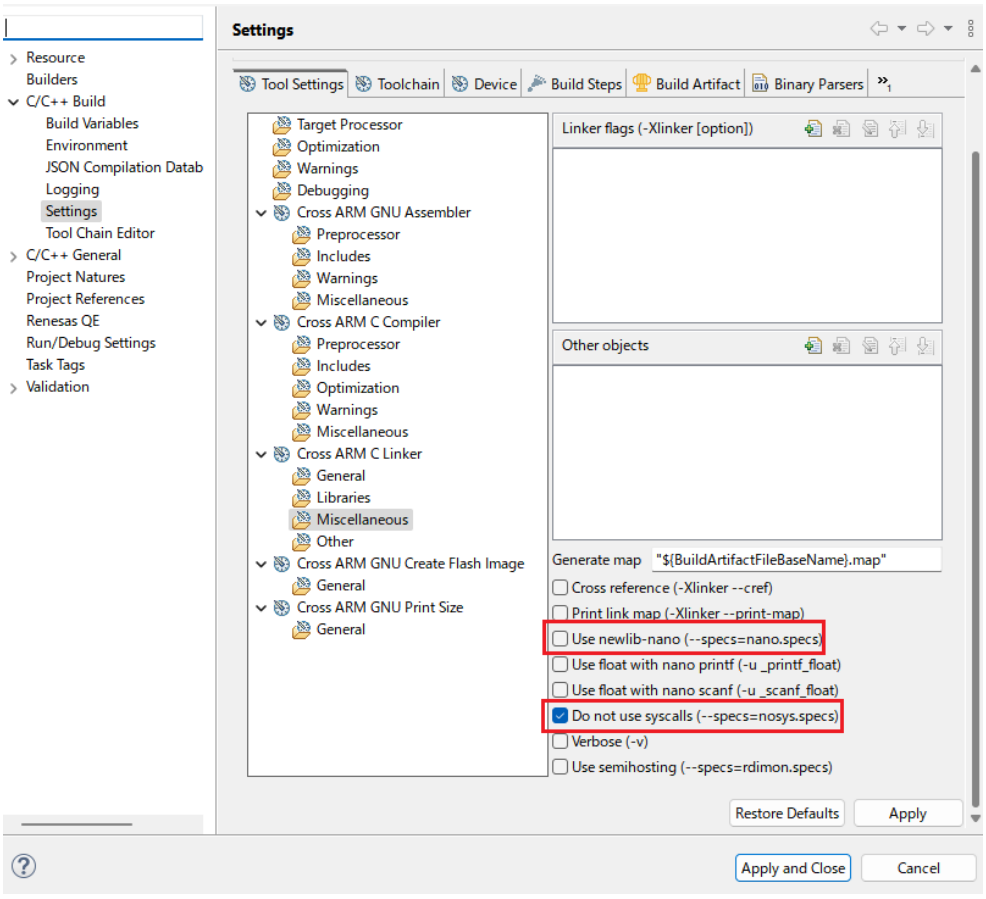


Figure 9.48 Properties

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
1.10	2026/05/29	p.6	OPC UA 認証取得に関して追記 x509 ユーザ認証を非サポートに変更
		p.8	x509 ユーザ認証用のファイルに関する記載を削除
		p.12	RZ/T2H, N2H 向け FSP バージョン更新 r_gmac_b の下記の修正取り込みのため • Fixed an issue where operation would stop after receiving or transmitting 2^{31} bytes of data via Ethernet.
		p.74	x509 ユーザ認証に関する記載を削除
		p.84	SystemReset に関する仕様を更新
		p.122	ccmake コマンドでの設定値を更新
		p.126	9.4 章 Method を使用した PubSub 設定について、動作確認手順を改善
		p.145	9.7 章 新しい FSP バージョン使用時の注意事項を記載
1.20	2026/08/28	p.6	サポート済機能について、Application Profile を Embedded 2017 UA Server に更新、User Tokens に X509 Certificate Server Facet を追加
		p.8	USB フラッシュドライブ上のファイルを用途別フォルダに分けて管理するよう変更
		p.81, 82	使用している OSS のバージョンを下記のように更新 Mbed TLS: v3.6.4 → v3.6.6 open62541: v1.3.15 → v1.3.18
		p.86	Startup シーケンス図を更新
		p.117	9.1 章 open62541 の Amalgamation オプションを OFF に変更 全般的にそれに準じた説明に変更
		p.144	9.6.2 章 コンフィグファイルについて説明した章を新規作成

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.

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/.

© 2026 Renesas Electronics Corporation. All rights reserved.