RZ/T2, RZ/N2 Group

OPC UA Server and PubSub Sample Software

Introduction

This document describes sample software for running OPC UA Server and PubSub on the RZ/T2, RZ/N2 in a real-time OS environment.

RZ/T2M and RZ/T2ME are designed to be compatible, so please use the model names in this document interchangeably.

Target Device

RZ/T Series: RZ/T2M (RZ/T2ME), RZ/T2H Group
RZ/N Series: RZ/N2L, RZ/N2H Group

Contents

1. Overview

OPC UA, which enables interoperability of industrial applications, is becoming widely used not only in factory automation but also in various industries.

This document describes the configuration and usage of sample software for implementing OPC UA Server and PubSub using a real-time OS on the RZ processors RZ/T2 and RZ/N2 for industrial networks.

1.1 Abbreviations / Definitions

The terms used in this document are defined and used as shown below.

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 User's Manual	R20UT4939******
User's Manual	Provides technical details of the RZ/T2M microprocessor.	RZ/T2M User's Manual Hardware	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 User's Manual	R20UT4984******
User's Manual	Provides technical details of the RZ/N2L microprocessor.	RZ/N2L User's Manual Hardware	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 User's Manual	R20UT5405******
User's Manual	Provides technical details of the RZ/T2H, RZ/N2H microprocessors.	RZ/T2H and RZ/N2H User's Manual Hardware	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 User's Manual	R20UT5522******
User's Manual	Provides technical details of the RZ/T2H, RZ/N2H microprocessors.	RZ/T2H and RZ/N2H User's Manual Hardware	R01UH1039******

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

Document Type	Description	Document Title	Document No.
Application Note	RZ/T2, N2 FSP startup guide	RZ/T2, RZ/N2 Getting Started with Flexible Software Package	R01AN6434******

2. Features

This package includes the firmware for the OPC UA stack on Renesas RZ/T2 and RZ/N2 series processors.

2.1 Function List

The features supported by this sample software are as follows.

2.1.1 OPC UA Server

The OPC UA Server specifications for this sample software are as shown in Table 2.1. This project has obtained certification for the software developed for the RZ/T2M, RZ/T2H, and RZ/N2L devices based on Sample Software v3.0.0, as described in Table 2.2. This sample software extends the functionality of those certified software packages and has been verified using the OPC UA Compliance Test Tool (UACTT) v1.04.11-01.00.510.

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

The username and password pairs configured in this sample software are listed in Table 2.3.

Table 2.3. Username/Password

	Username	Password
1	user1	password
2	user2	password1

2.1.2 OPC UA PubSub

The OPC UA PubSub specifications for this sample software are as shown in Table 2.4. Communication with SKS Server is currently not supported.

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

This sample software supports the protocols shown in Table 2.5. For details on each protocol, please refer to the chapter described in the Details section.

Table 2.5. Sample Software Support Protocols

Protocol	**Details**
SNTP Client	Get time information from NTP server when the device starts up
TFTP Server	Chapter 9.2
LLDP Agent	Chapter 9.3
SNMP Agent	Chapter 9.3

2.1.4 File System

Various files used for security functions are managed by the file system (FreeRTOS + FAT) function and read from a USB flash drive connected to the evaluation board. You can also write files to a USB flash drive via Ethernet using TFTP communication. For details on the file writing procedure, refer to chapter 9.2.

The destination folders for each file type are listed in Table 2.6. Except for the configuration file, there are no restrictions on file names. Files marked with a ✓ in the "Attached to Package" column are provided as examples in the “keyfiles.zip” package. Before operation verification, connect a USB flash drive containing these files to the evaluation board.

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	✓

✓: attached, -: not attached

The TrustList, CRL, and IssuerList each have a defined maximum number of files that can be loaded. To change these limits, modify the following macros defined in “opc_load_files.h”.

Table 2.7. Macro definitions

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

The configuration files (“pubsub.cfg” and “ip.cfg”) can be edited to modify the PubSub settings and IP address configuration. For details, refer to chapter 9.6.2.

It is also possible to perform security operations without a USB flash drive connected. In this case, please refer to chapter 9.6.1.

2.2 Package folder structure

The folder structure for this sample software is shown below.

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	e² 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	e² 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)

This project has been developed and tested on these environments using the following boards and tools.

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 Patch for RZ/T2H, N2H “EWARM_Patch_for_RZT2H_N2H_rev1.0.zip” is required
	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 Patch for RZ/T2H, N2H “EWARM_Patch_for_RZT2H_N2H_rev1.0.zip” is required
	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 Flash Drive	-	-	-	Used as file storage for file system Formatting to FAT32 format required *Can be operated without USB Flash Drive.
OPC UA Client Tool	UaExpert	2.0.1	OPC UA Clients - Unified Automation (unified-automation.com)	
Packet analyzer	Wireshark	4.4.7	Wireshark · Download	
OPC Communication Generic Tool	OpcCmd Utility	-	OpcCmd Utility Download - OPC Labs Knowledge Base	Used as OPC UA PubSub Subscriber
OPC UA Publisher Tool	UADemoPublisher	-	OPC UA Demo Publisher Download - OPC Labs Knowledge Base	
Low-code development tool	Node-RED	4.0.3	Node-RED (nodered.org) Node.js — Run JavaScript Everywhere (nodejs.org)	Used as OPC UA Client Graphical display of Node value

4. Hardware Setup

This document describes the major hardware. For more information about the boards, refer to the respective evaluation board user manuals and schematics.

4.1 RZ/T2M RSK Board

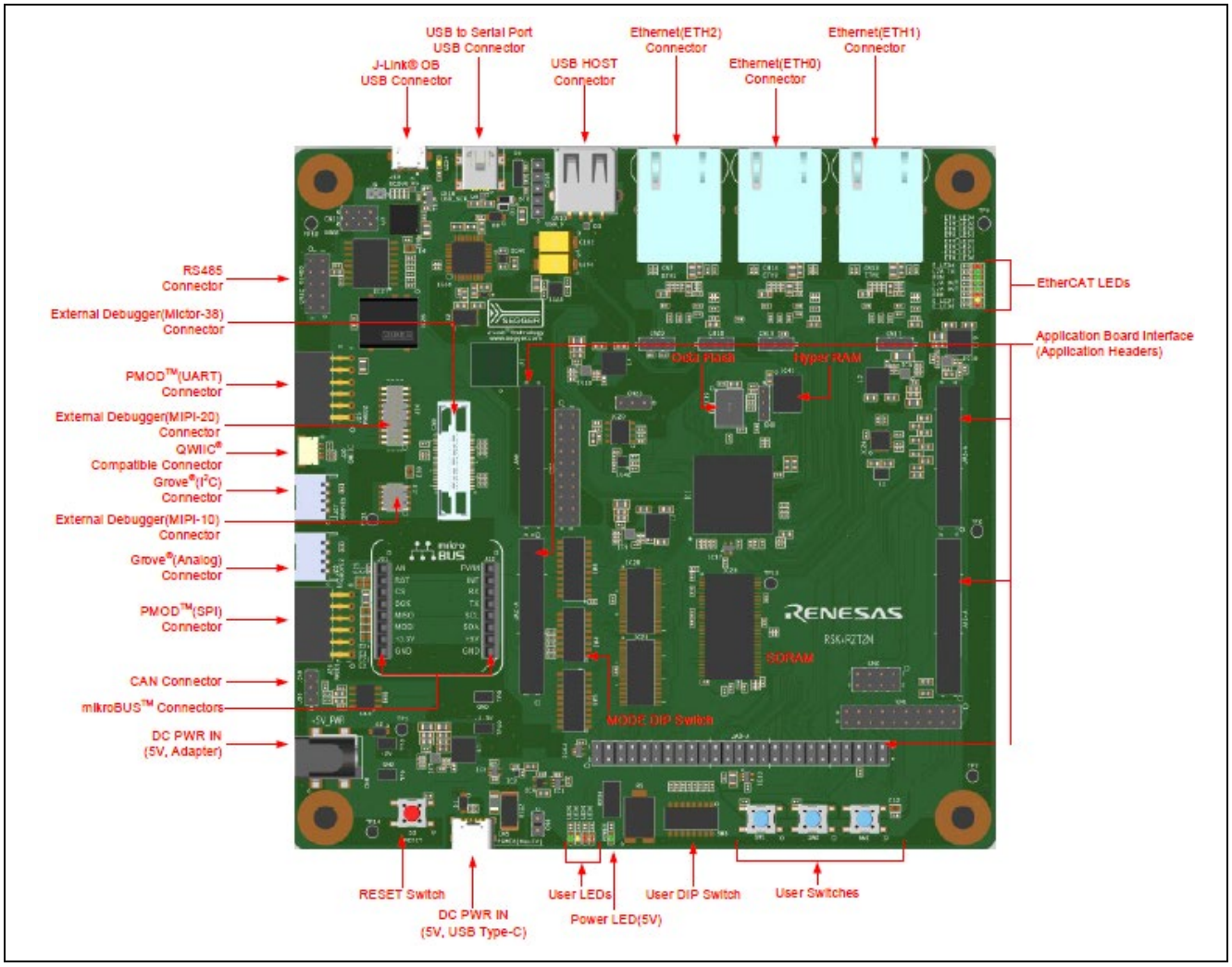


Figure 4.1. RZT2M RSK board layout

The text in red in the table indicates settings that need to be changed from the evaluation board's factory default settings.

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 shows a connection diagram when running the sample software. Connect the Ethernet cable, debugger, and 5V DC cables to the RZ/T2M RSK board.

You can connect the Ethernet cable to the ETH0 or 1 connector. When using the debugger J-Link OB on the RSK board, open J9 and connect the USB Micro cable. When using I-jet, short J9 and connect I-jet to J20.

This sample software uses SNTP for time synchronization and requires a connection to an NTP Server. A Windows PC is used as the NTP Server during the operation check.

If you use FreeRTOS+FAT for managing key files used by the OPC UA security function, a USB flash drive connection to the RSK board is required.

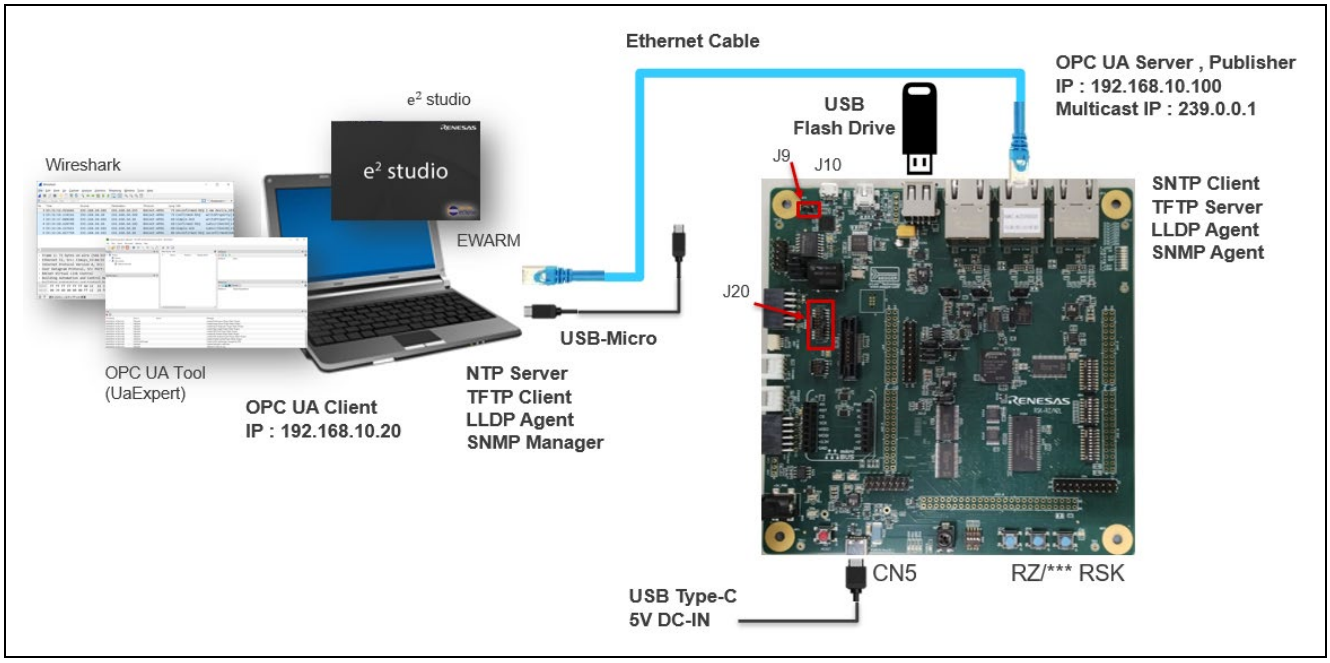


Figure 4.2. RZ/T2M RSK board OPC UA Server hardware connection diagram

4.2 RZ/N2L RSK Board

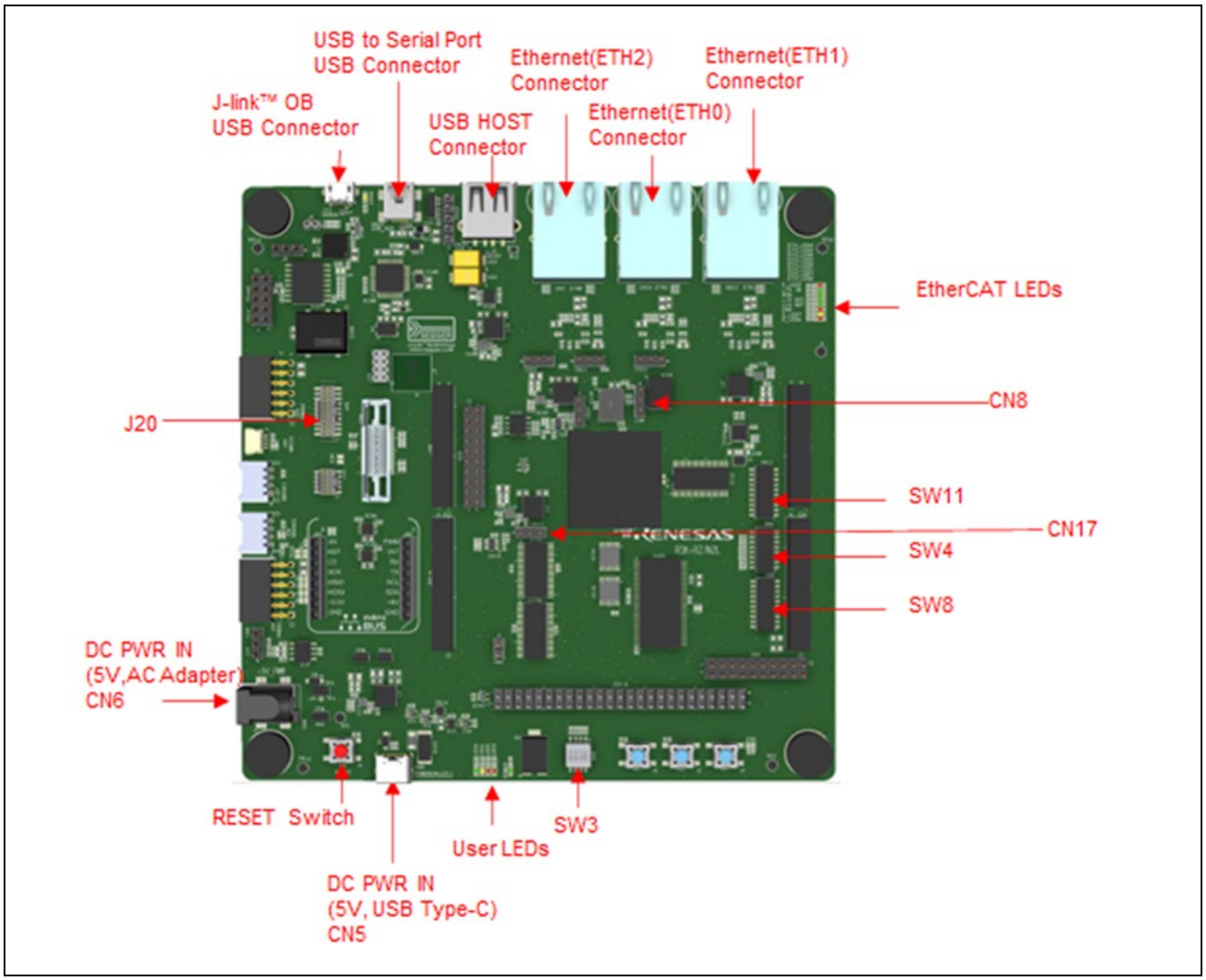


Figure 4.3. RZN2L RSK board layout

The text in red in the table indicates settings that need to be changed from the evaluation board's factory default settings.

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 shows a connection diagram when running the sample software. Connect the Ethernet cable, debugger, and 5V DC cables to the RZ/N2L RSK board.

You can connect the Ethernet cable to the ETH0 or 1 connector. When using the debugger J-Link OB on the RSK board, open J9 and connect the USB Micro cable. When using I-jet, short J9 and connect I-jet to J20.

This sample software uses SNTP for time synchronization and requires a connection to an NTP Server. A Windows PC is used as the NTP Server during the operation check.

If you use FreeRTOS+FAT for managing key files used by the OPC UA security function, a USB flash drive connection to the RSK board is required.

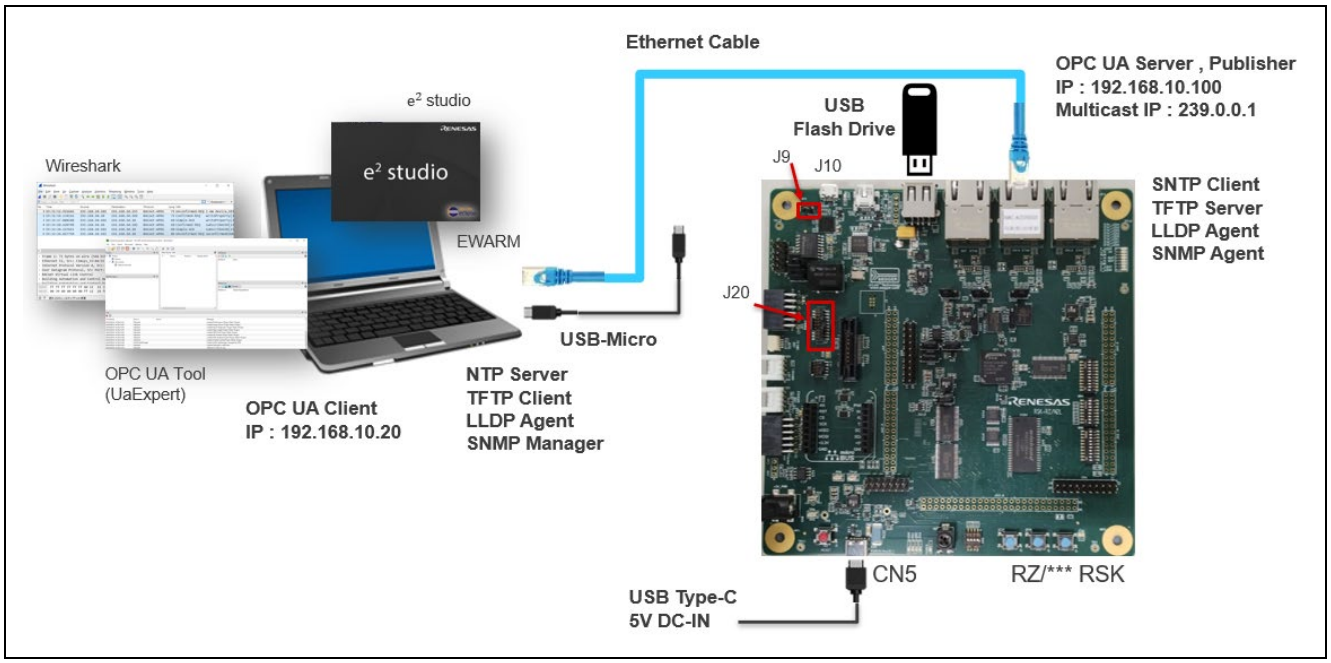


Figure 4.4. RZ/N2L RSK board OPC UA Server hardware connection diagram

4.3 RZ/T2H Evaluation Board

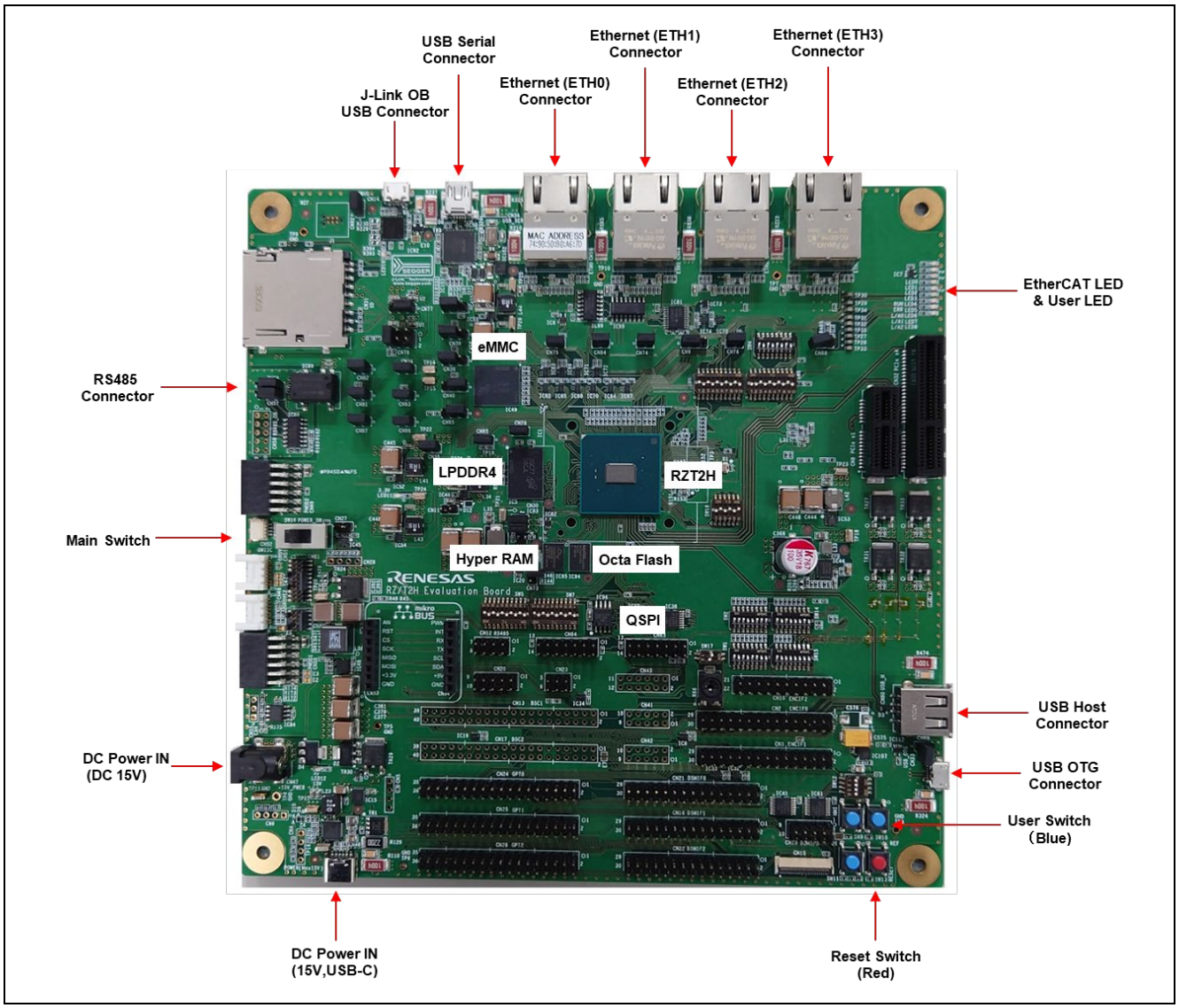


Figure 4.5. RZT2H Evaluation board layout

The text in red in the table indicates settings that need to be changed from the evaluation board's factory default settings.

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 shows a connection diagram when running the sample software. Connect the Ethernet cable, debugger, and 15V DC cables to the RZ/T2H evaluation board.

You can connect the Ethernet cable to the ETH0 or 1 connector. When using the debugger J-Link OB on the evaluation board, open CN62 and connect the USB Micro cable. When using I-jet, short CN62 and connect I-jet to CN61.

This sample software uses SNTP for time synchronization and requires a connection to an NTP Server. A Windows PC is used as the NTP Server during the operation check.

If you use FreeRTOS+FAT for managing key files used by the OPC UA security function, a USB flash drive connection to the evaluation board is required.

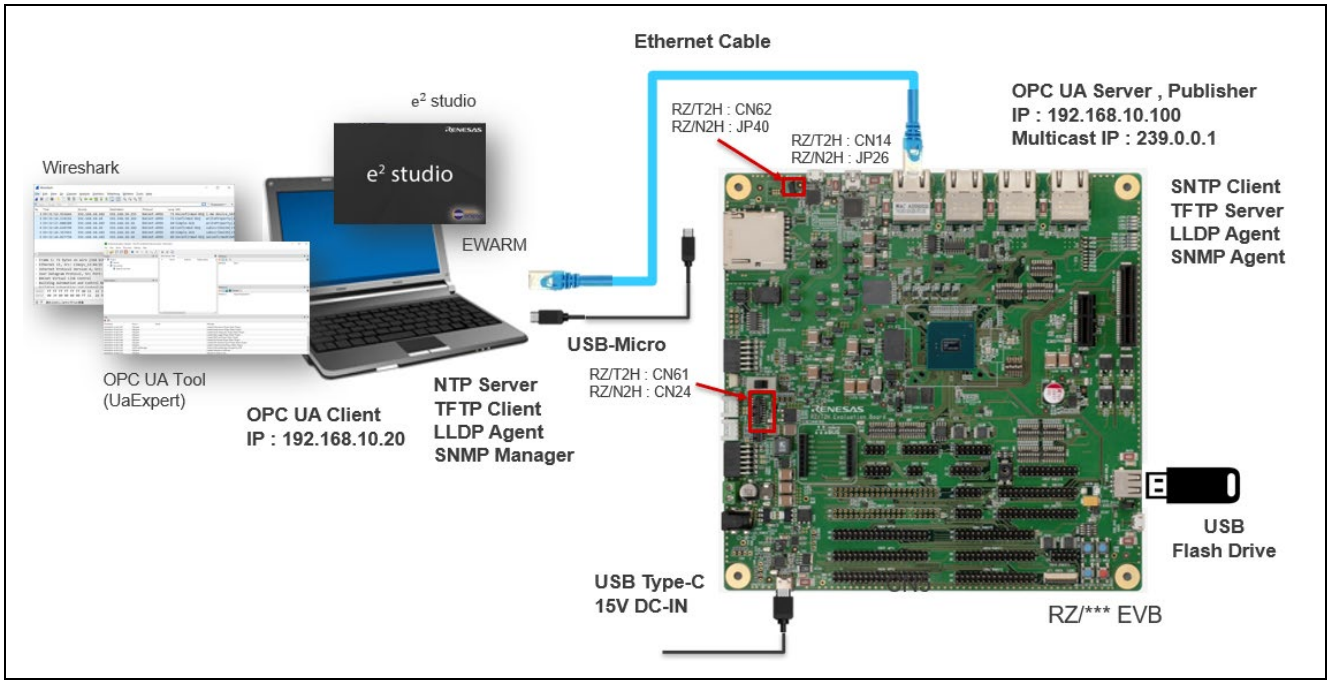


Figure 4.6. RZ/T2H EVB OPC UA Server hardware connection diagram

4.4 RZ/N2H Evaluation Board

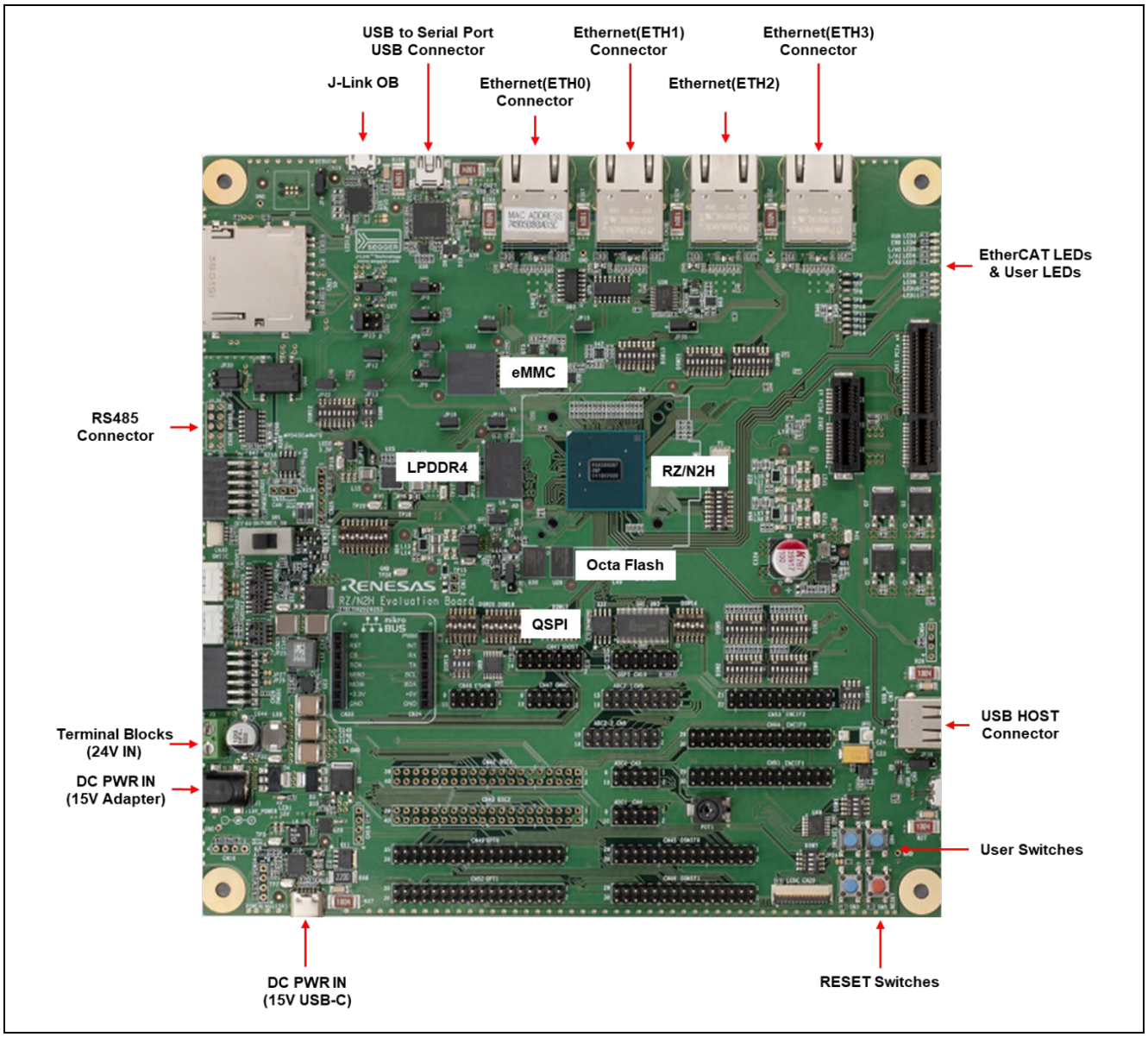


Figure 4.7. RZN2H Evaluation board layout

The text in red in the table indicates settings that need to be changed from the evaluation board's factory default settings.

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 JP31	Short 1-2	Full-duplex communication
	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 shows a connection diagram when running the sample software. Connect the Ethernet cable, debugger, and 15V DC cables to the RZ/N2H evaluation board.

You can connect the Ethernet cable to the ETH0 or 1 connector. When using the debugger J-Link OB on the evaluation board, open JP40 and connect the USB Micro cable. When using I-jet, short JP40 and connect I-jet to CN24.

This sample software uses SNTP for time synchronization and requires a connection to an NTP Server. A Windows PC is used as the NTP Server during the operation check.

If you use FreeRTOS+FAT for managing key files used by the OPC UA security function, a USB flash drive connection to the evaluation board is required.

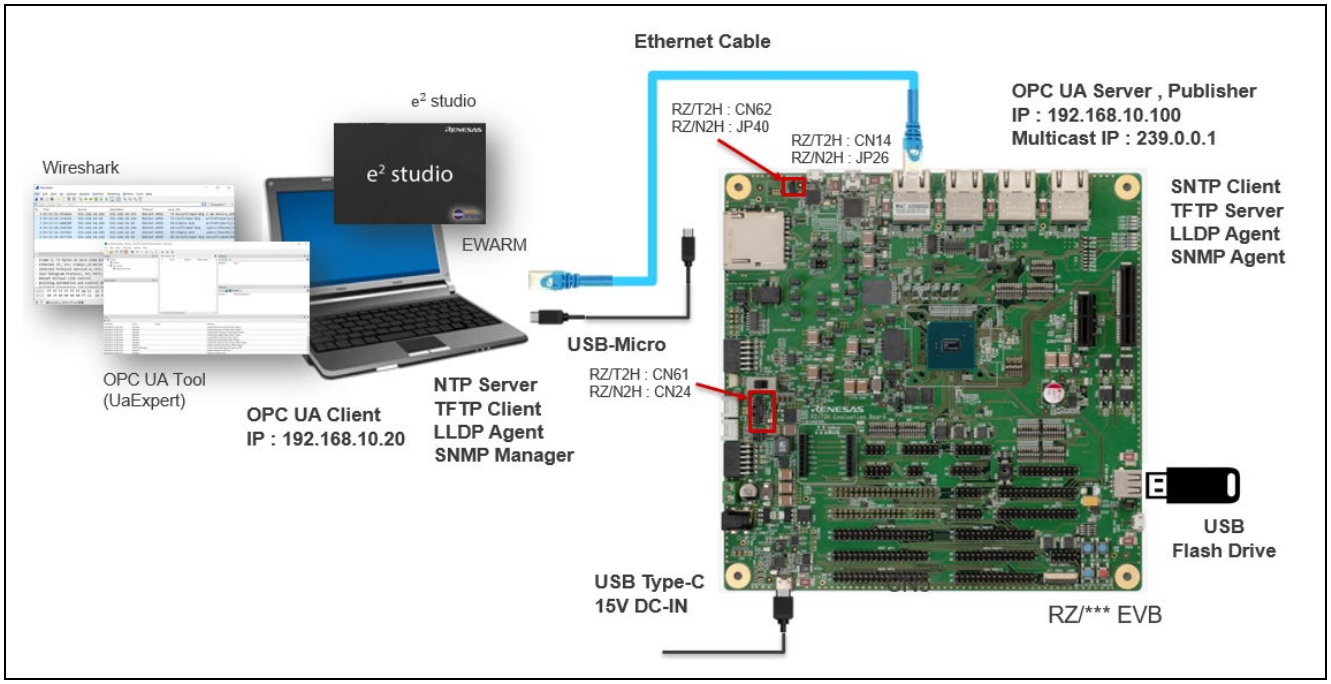


Figure 4.8. RZ/N2H EVB OPC UA Server hardware connection diagram

5. Set up the Host

5.1 Host IP Address

Set the address of the Ethernet on the PC that serves as the OPC UA Client.

Open Control panel. Configure the IP address as follows.

Control panel > Network and Internet > Change adapter options > Ethernet > Properties > Internet Protocol Version 4 (TCP/IPv4) > Properties

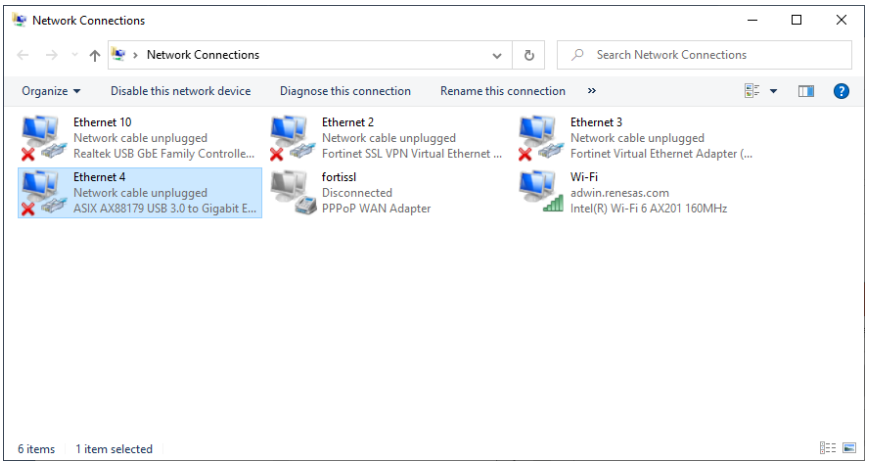


Figure 5.1. Network connection

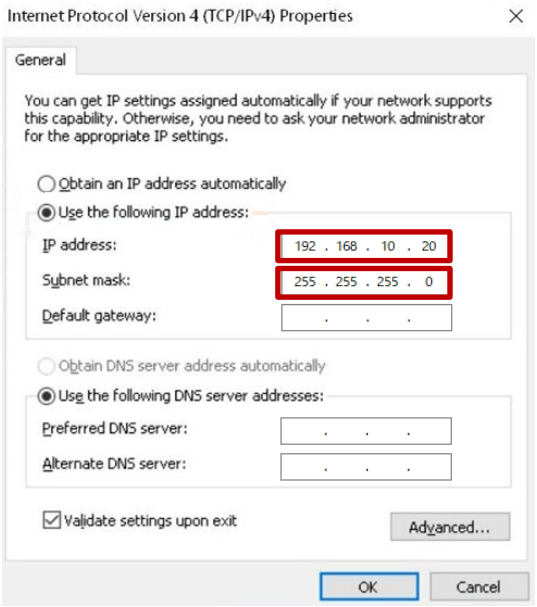


Figure 5.2. TCP/IPv4 properties

The IP address of the evaluation board set in the OPC UA server sample software is 192.168.10.100. The IP address of the PC needs to be set to 192.168.10.***. This sample software sets the NTP server IP address to 192.168.10.20 by default. Therefore, if you want to use a Windows PC as the NTP server, set the PC's IP address to 192.168.10.20.

5.2 Installation of EWARM environment

If developing in the e² studio environment, skip this chapter.

1) Download the EWARM installer file for the version listed in chapter 3.
2) Click “Install IAR Embedded Workbench® for Arm”, and follow the instructions to install.
3) Download the FSP Smart Configurator installer file for the version listed in chapter 3.
4) Run the installer file, and follow the instructions to install.

5.3 Installation of e² studio environment

If developing in the EWARM environment, skip this chapter.

1) Download the e² studio installer file for the version listed in chapter 3.
2) Run the installer file, and follow the instructions to install.

5.4 UaExpert

UaExpert is an OPC UA Client tool. In this document, it is used to connect to the OPC UA Server to access the object nodes.

Download the version listed in Table 3.5 from the website and install it on your PC. Before downloading, you must register on the Unified Automation website and activate your account. All content is provided free of charge, but by downloading or installing the software from this web page, you automatically accept the Unified Automation Software License Agreement (SLA). For license terms for software and information, please refer to the following link.

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

Please check the above conditions of use before usage.

Figure 5.3. UaExpert

5.5 Wireshark

Wireshark is a free network protocol analyzer. Download and install Wireshark from the link in Table 3.5.

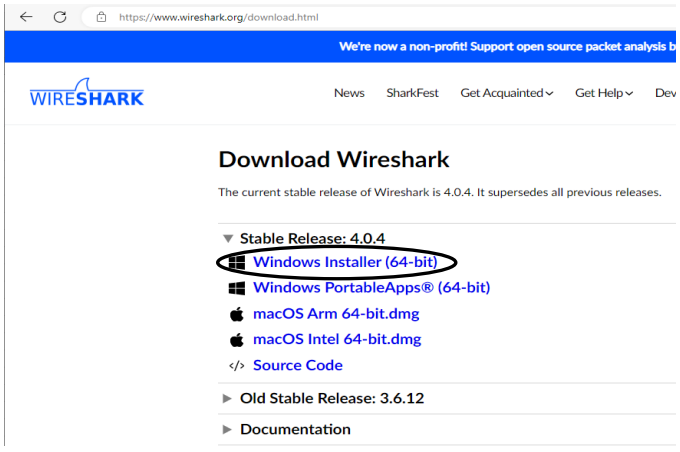


Figure 5.4. Download Wireshark

5.6 OpcCmd Utility

OpcCmd Utility is a general-purpose OPC communication tool that can be used with the command line. For this sample software, this tool is used as a Subscriber of OPC UA PubSub. Download from the link in Table 3.5 and unzip to any folder. No installation is required. License terms are Zero-Clause BSD (0BSD).

For detailed instructions on how to use this tool, please refer to the link below.

Using OpcCmd Utility as OPC UA PubSub Subscriber - OPC Labs Knowledge Base

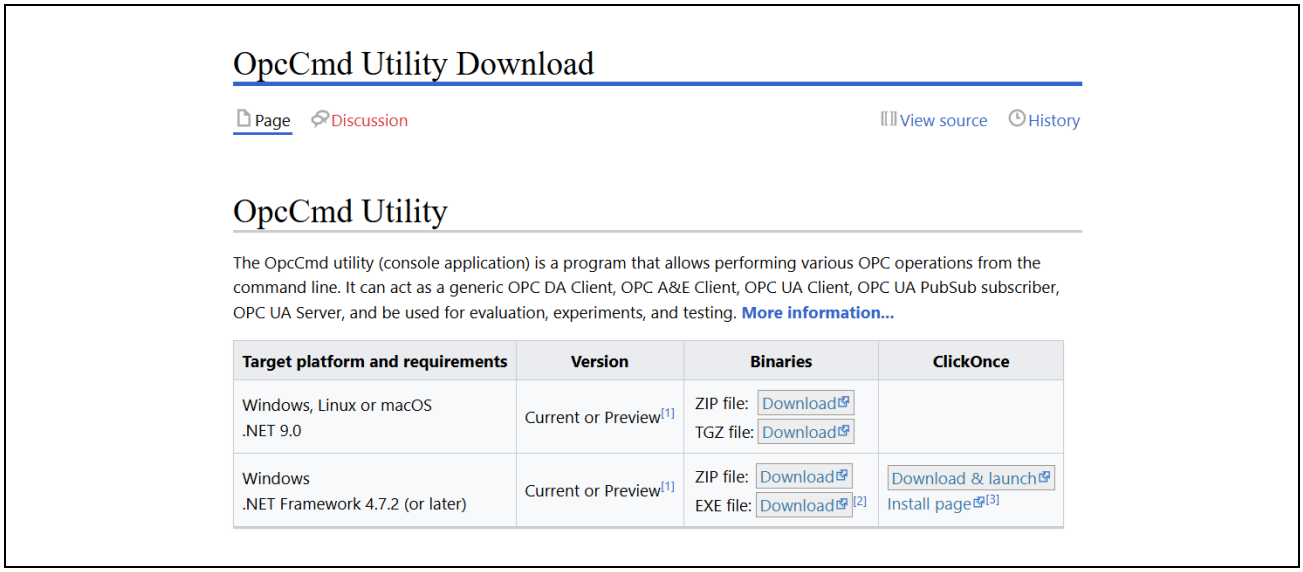


Figure 5.5. Download OpcCmd Utility

5.7 OPC UA Demo Publisher

OPC UA Demo Publisher is an OPC UA Publisher tool that can be used with the command line. Download from the link in Table 3.5 and unzip to any folder. No installation is required. License terms are Zero-Clause BSD (0BSD).

For detailed instructions on how to use this tool, please refer to the link below.

UADemoPublisher Basics - OPC Labs Knowledge Base

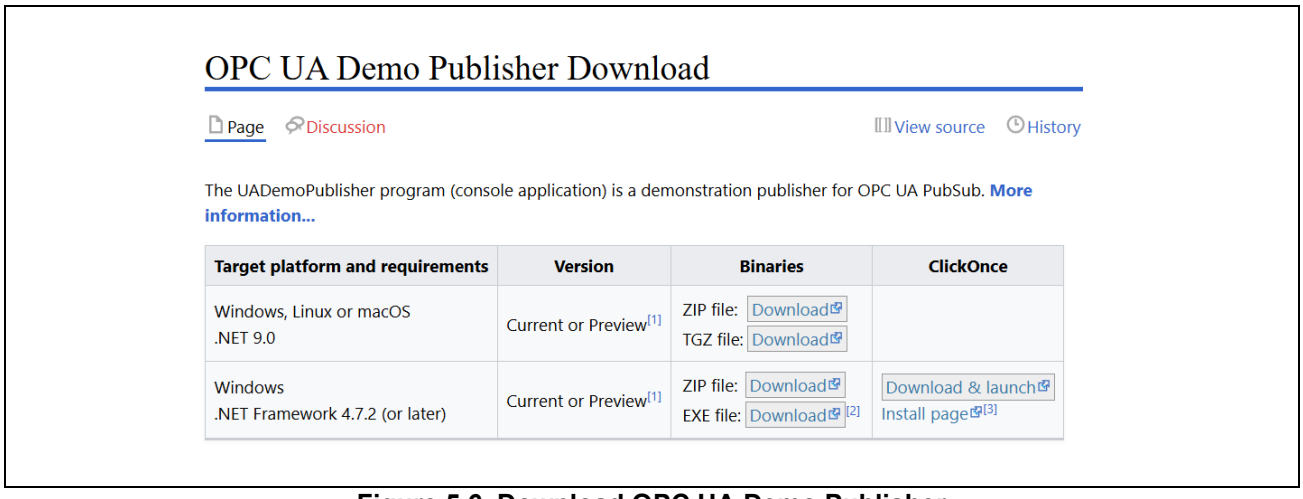


Figure 5.6. Download OPC UA Demo Publisher

5.8 Node-RED

Node-RED is a browser-based programming tool that allows users to create flows using various nodes and deploy them to the execution environment with a single click. In this project, it will be used to monitor OPC UA Node changes over time.

Install Node.js from the link listed in Table 3.5. After installation, launch Command prompt with administrator privileges and enter the following command

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

npm install node-red-contrib-opcua

npm i node-red-dashboard

Launch Command prompt again without administrator privileges and enter the following command.

node-red

If Node-RED is launched successfully, you will see a screen like Figure 5.7. Do not close this Command prompt while using Node-RED.

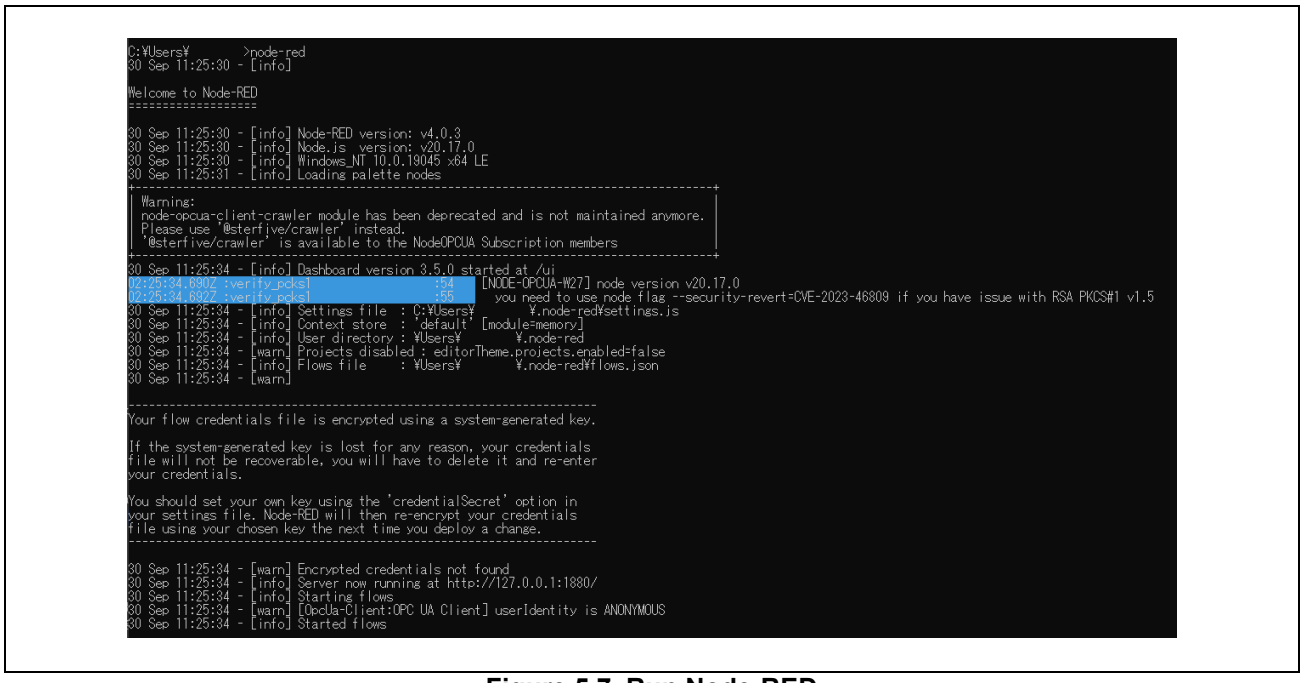


Figure 5.7. Run Node-RED

Enter the following URL into the browser, and if the screen shown in Figure 5.8 is displayed, Node-RED has been successfully launched.

http://localhost:1880

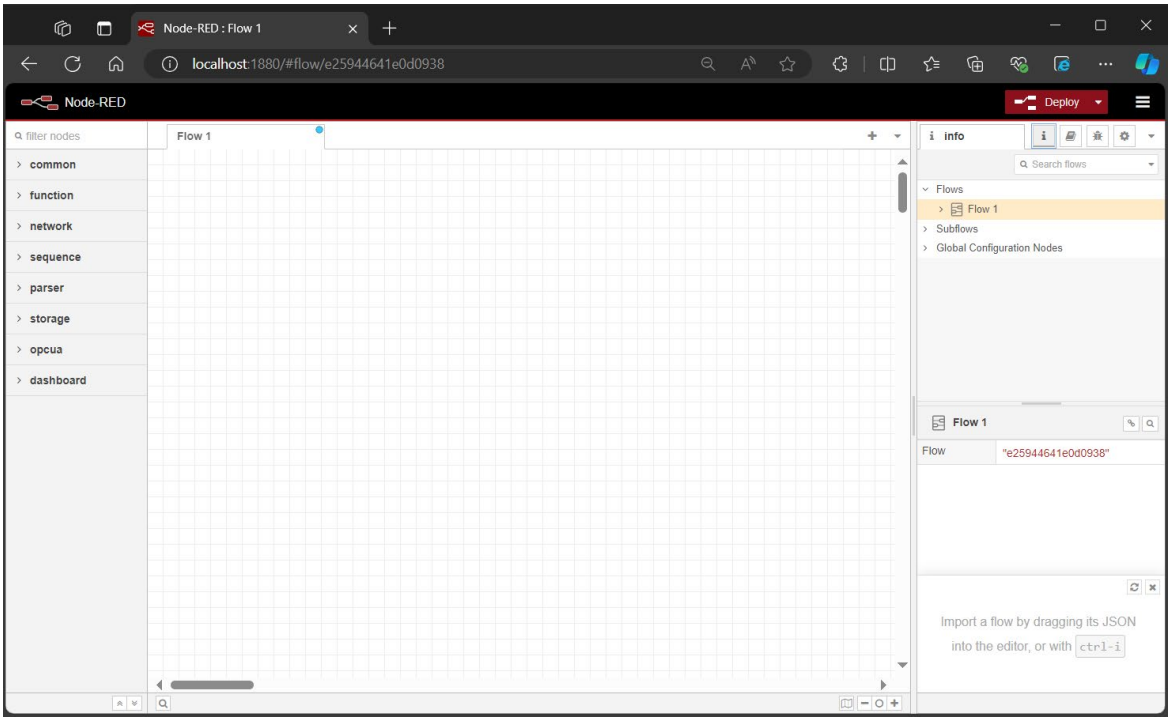


Figure 5.8. Start Node-RED

6. Running the Sample Application

This chapter describes how to download and run the program.

First, unzip the archived package of this sample software and store it in any folder. Because e^2 studio cannot recognize a project properly if the file path is too long, place it in shorter path. Also, do not use multi-byte character, such as Japanese, in the folder path.

In the Dual-core project, there are two projects: one that flashes the yellow LED on the evaluation board (RZ***_EVB(RSK)_OPC_****_*_Primary_0) and one that operates OPC UA (RZ***_EVB(RSK)_OPC_****_*_Secondary_1). Hereafter, these will be referred to as the primary project and the secondary project, respectively.

6.1 When using EWARM

Start EWARM and click "Open Workspace..." on the File tab.

CR52_single project：RZ***_OPC_EVB(RSK)_rev****\project\CR52_single\ewarm\

CR52_dual project：RZ***_OPC_EVB(RSK)_rev****\project\CR52_dual\ewarm\

CA55_dual project：RZ***_OPC_EVB(RSK)_rev****\project\CA55_dual\ewarm\

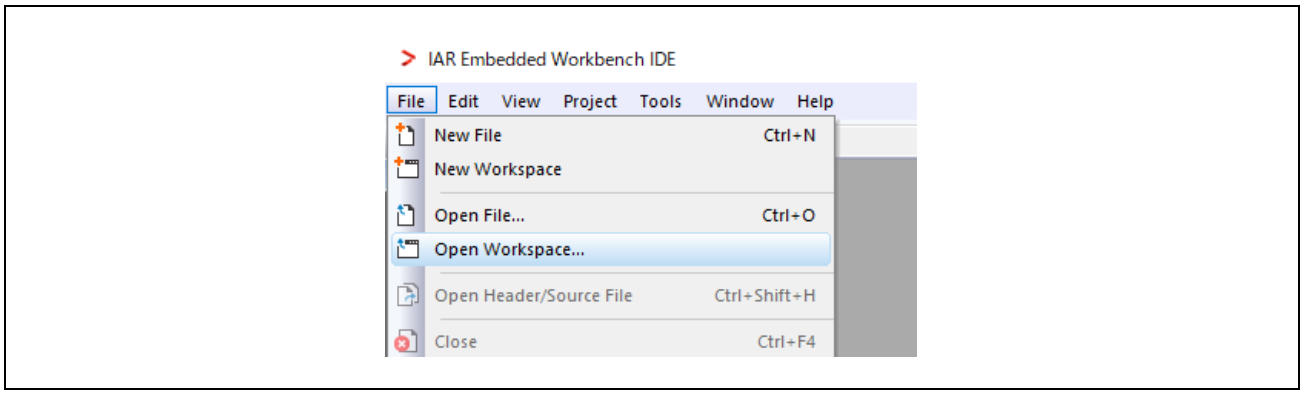


Figure 6.1. Open Workspace

Select the workspace file (.eww) and click the “Open” button.

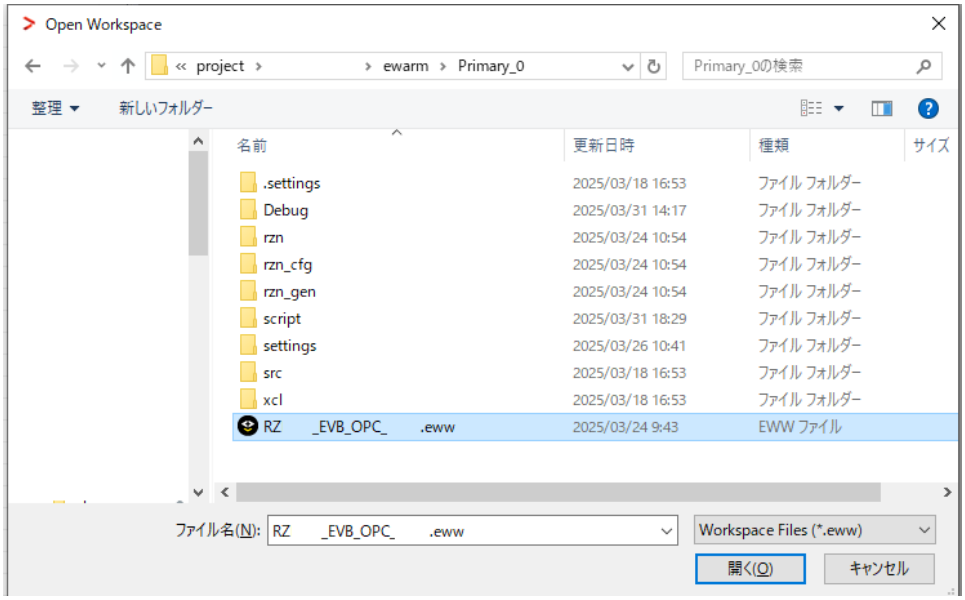


Figure 6.2. Open project file

Subsequent steps vary depending on the project used.

6.1.1 For Dual core project

6.1.1.1 Build

* Please refer to the guide in chapter 9.7 when using FSP v4.1.0 or later for RZ/T2M, or when using FSP v4.2.0 or later for RZ/T2H or RZ/N2H.

Open the primary project in EWARM and click “FSP Smart Configurator” in the Tools tab. If this button is not displayed or if an error appears after clicking it, refer to chapter 9.5.

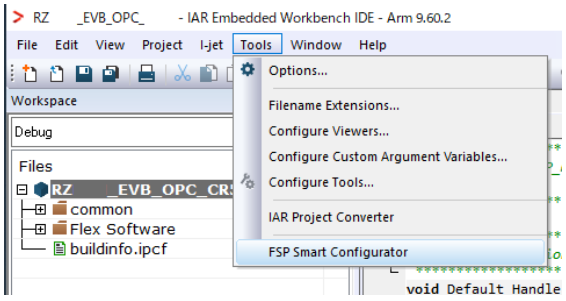


Figure 6.3. Tools tab

Clicking the “Generate Project Content” button on the FSP Smart Configurator generates the code.

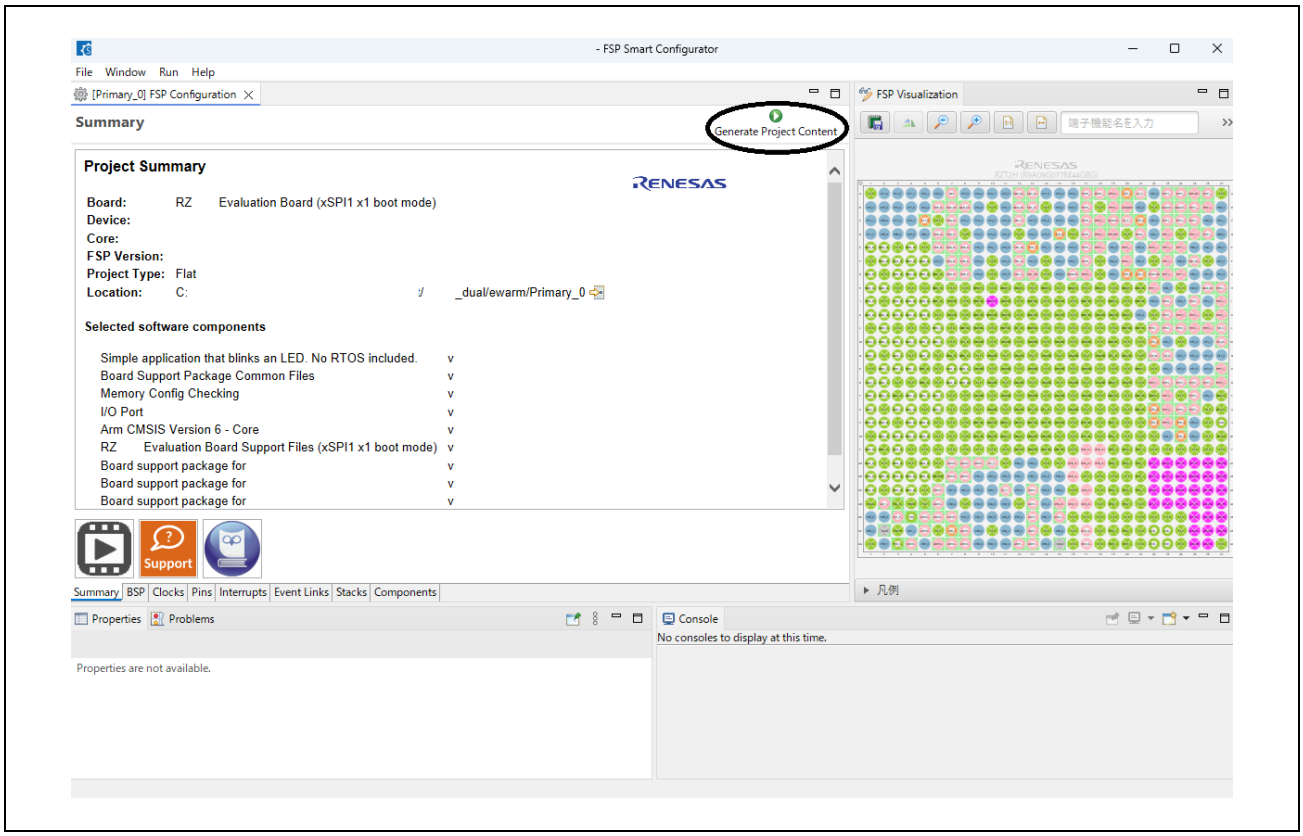


Figure 6.4. Generate Project Content

When debugging using a debugger, add loop processing at the location below.

"(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
}
```

For RZ/T2H and N2H only, change the device tag name defined in buildinfo.ipcf as follows.

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

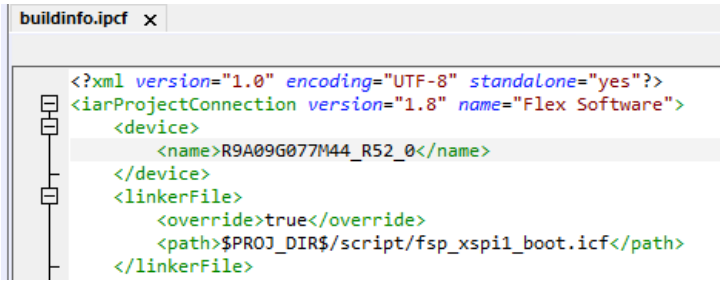


Figure 6.5. IAR EWARM Project File

Click the Make button to build. When the build is complete, a build message will appear in the build console window, showing the compile target file and the number of errors/warnings.

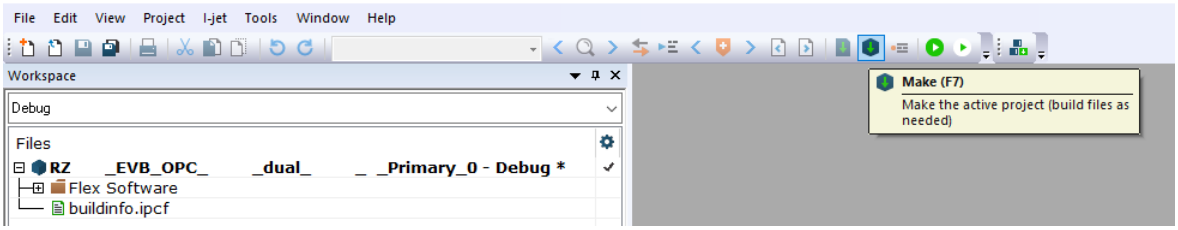


Figure 6.6. Make Primary project

Open the secondary project and click “FSP Smart Configurator” in the Tools tab.

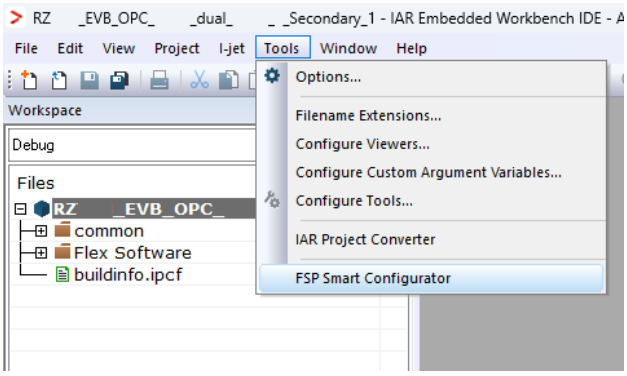


Figure 6.7. Open FSP SC

*If you wish to change any settings in the sample software from their default values, please configure them at this time by referring to chapter 9.6.1.

Clicking the “Generate Project Content” button on the FSP Smart Configurator generates the code.

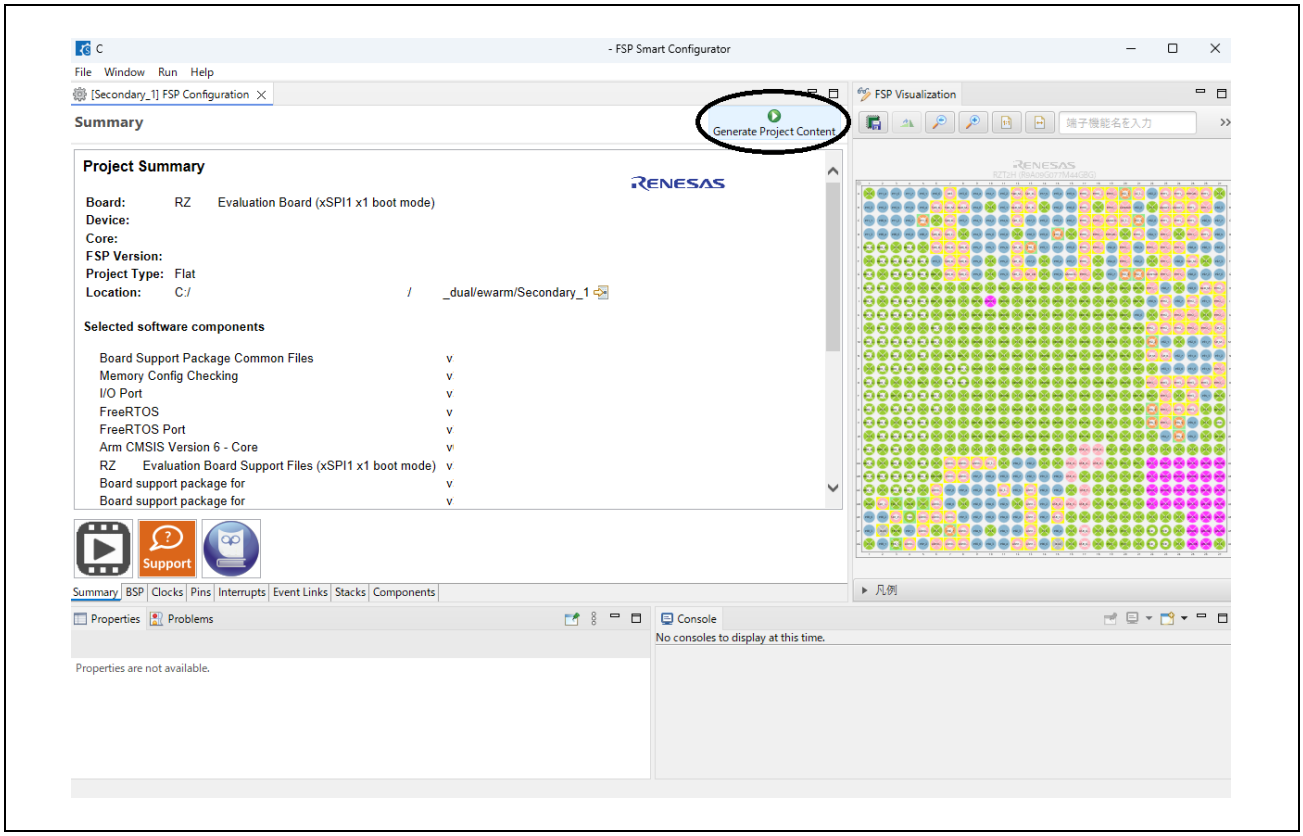


Figure 6.8. Generate Project Content

For RZ/T2H and N2H only, change the device tag name defined in buildinfo.ipcf as follows.

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

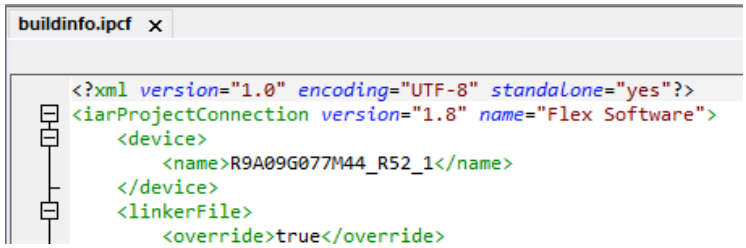


Figure 6.9. IAR EWARM Project File

Click the Make button to build. When the build is complete, a build message will appear in the build console window, showing the compile target file and the number of errors/warnings.

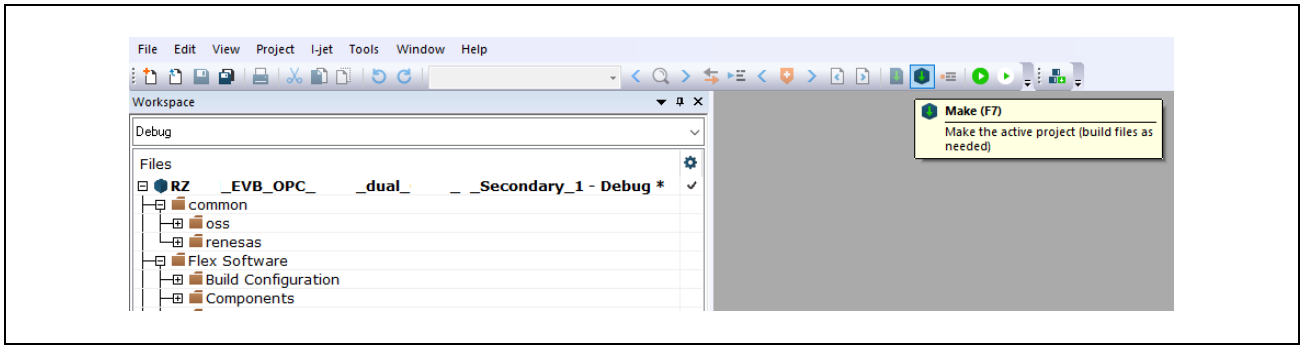


Figure 6.10. Make Secondary project

Reopen the primary project, run Clean, then click the Make button to build.

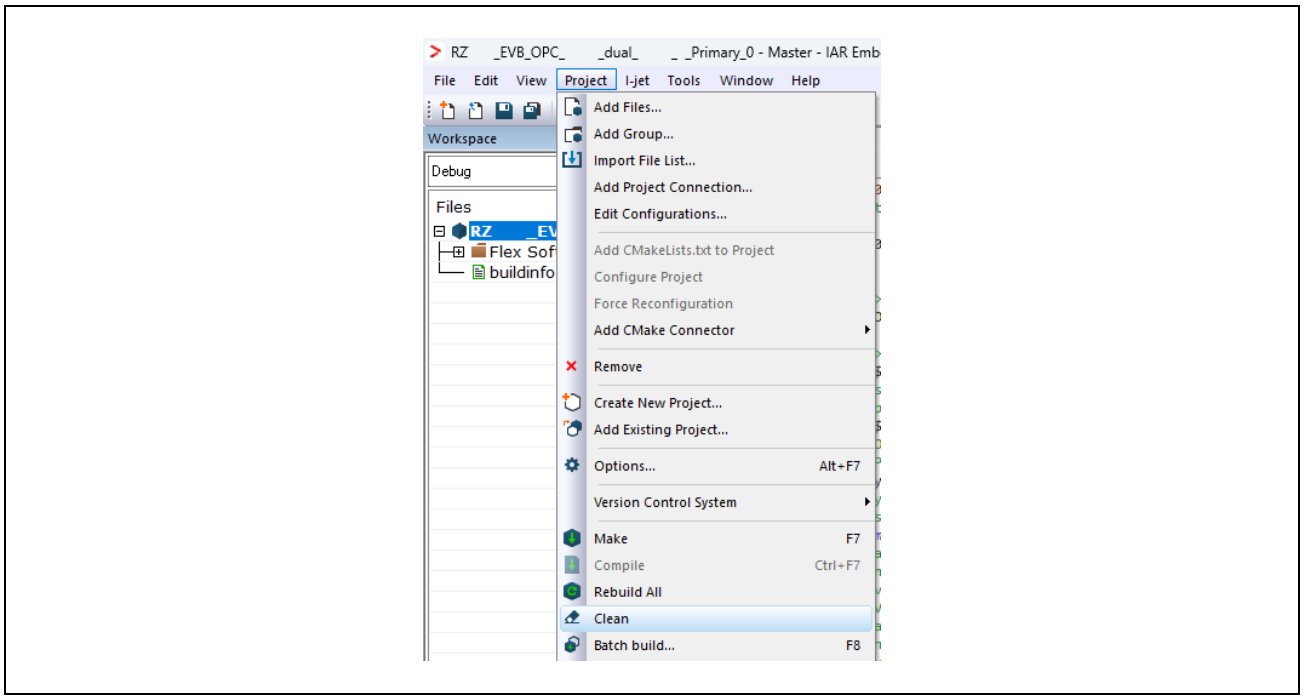


Figure 6.11. Clean Primary project

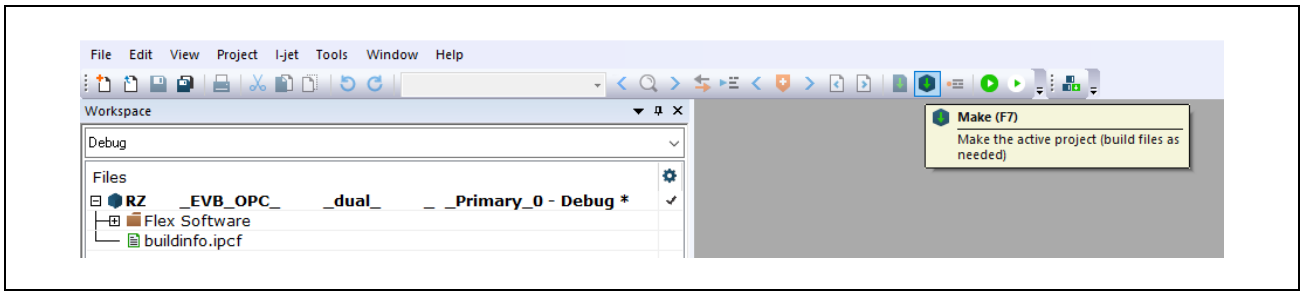


Figure 6.12. Make Primary project

6.1.1.2 Flash writing, debug

Open options of the primary project and set “Asymmetric multicore” to “Disabled” from the Multicore tab in the Debug category.

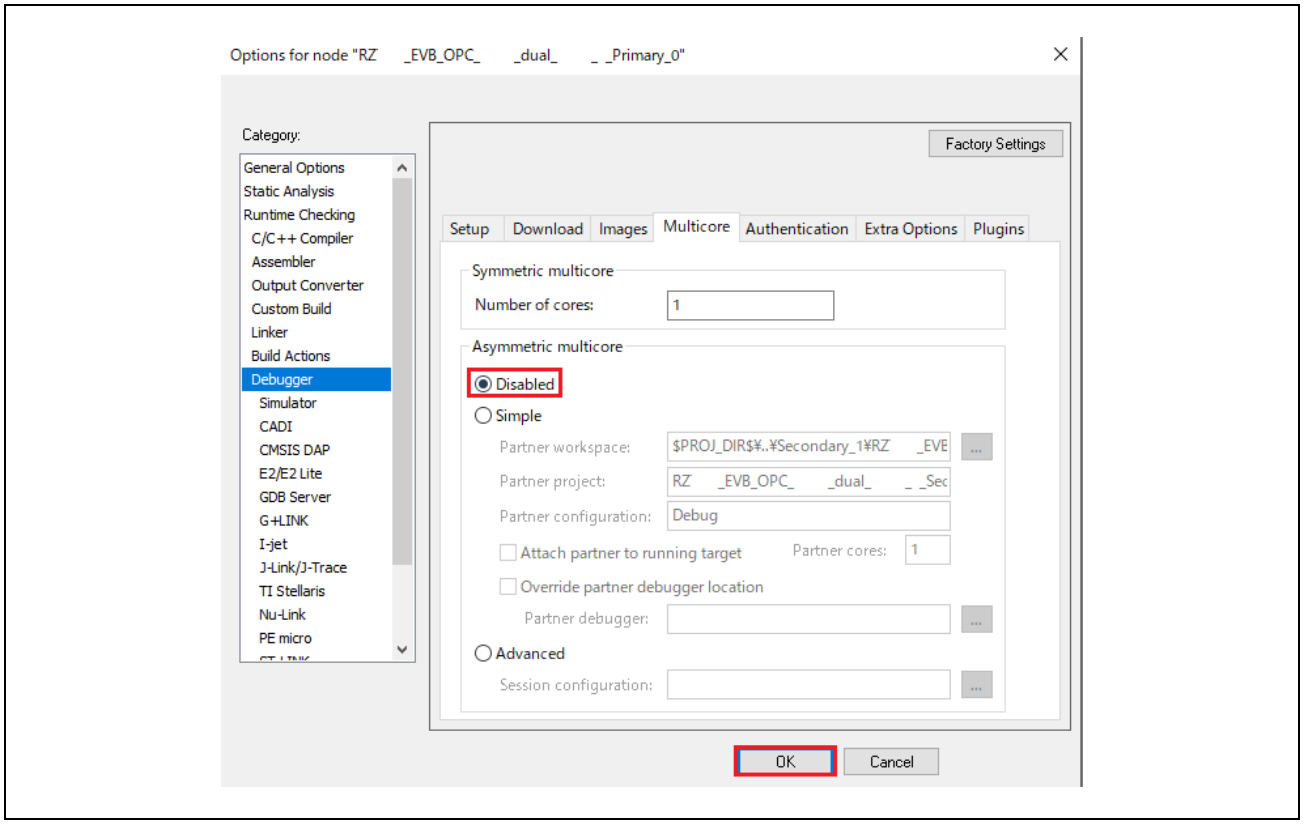


Figure 6.13. Asymmetric multicore setting

Click Download>Download file.... from the Project tab

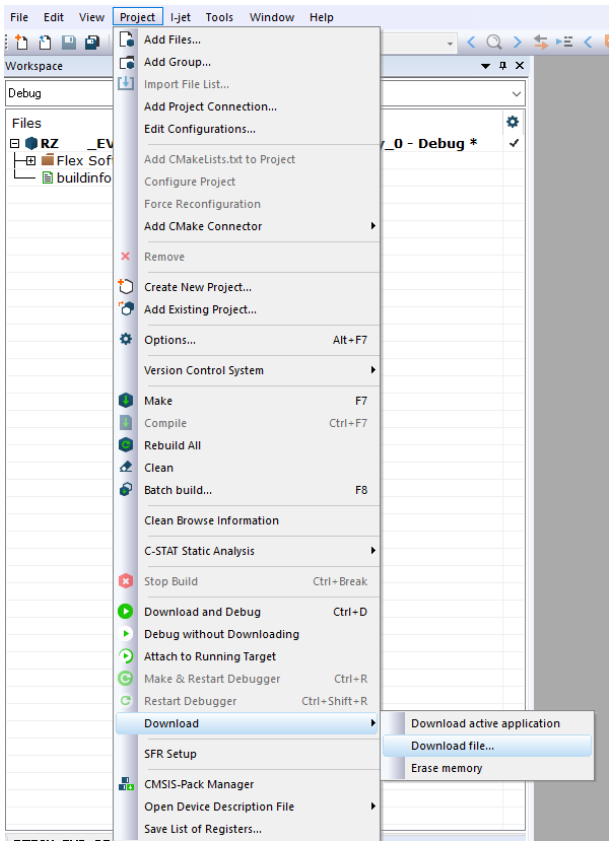


Figure 6.14. Download file…

Select \(Primary Project Location)\Debug\Exe\RZ***_EVB(RSK)_OPC_****_*_Primary_0.out to download the program to flash memory.

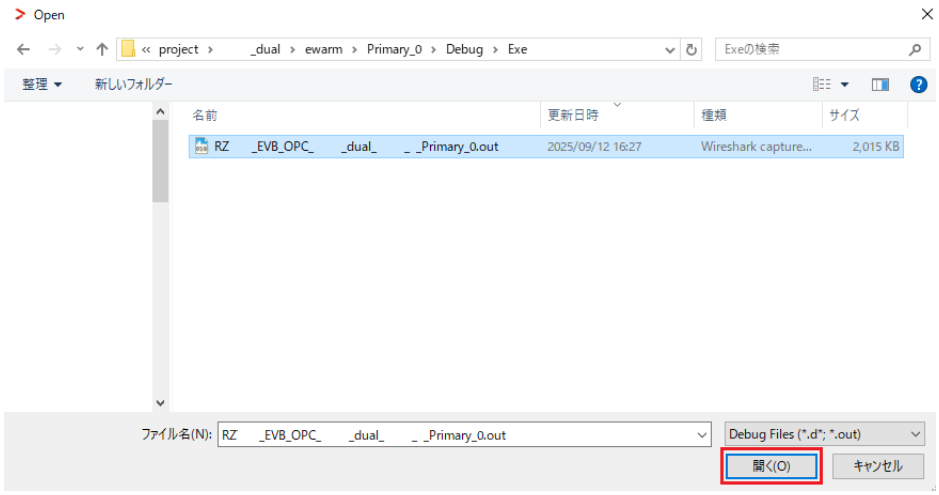


Figure 6.15. Open .out file

To operate the evaluation board alone without using the debugger, turn off the board power here, disconnect the debugger cable, and turn the power on again.

To use the debugger, open the options and select “Simple” from “Asymmetric multicore” from the Multicore tab in the Debug category, and set as follows.

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

Partner project : (Secondary project name)

Partner configuration : Debug

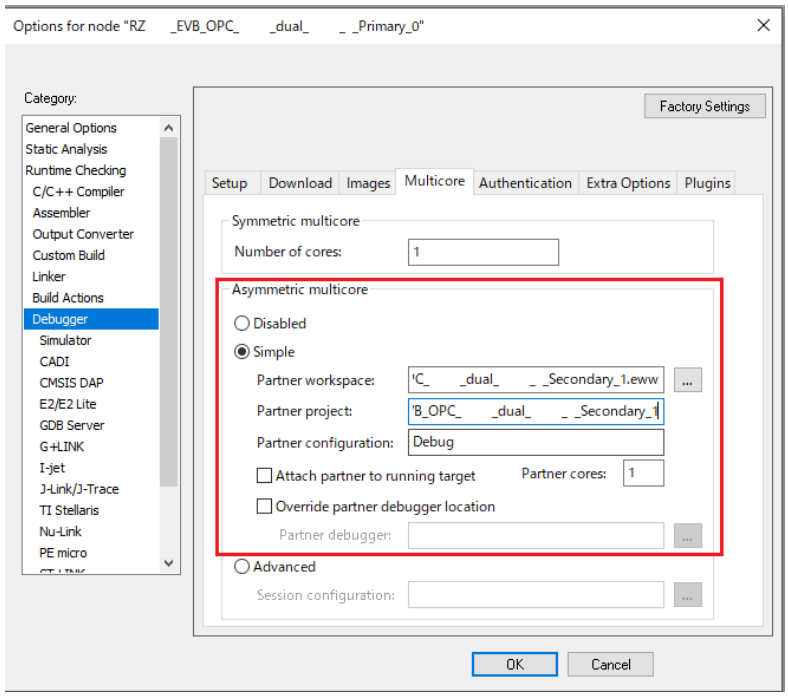


Figure 6.16. Asymmetric multicore setting

After setting, click “Debug without Downloading” to open the debugger in separate windows for primary and secondary.

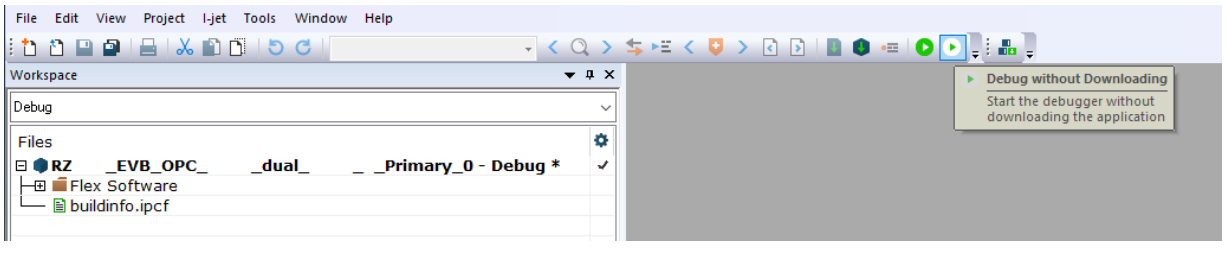


Figure 6.17. Debug without Downloading

6.1.2 For Single core project

6.1.2.1 Build

* Please refer to the guide in chapter 9.7 when using FSP v4.1.0 or later for RZ/N2L.

Open the project in EWARM and click “FSP Smart Configurator” in the Tools tab. If this button is not displayed or if an error appears after clicking it, refer to chapter 9.5.

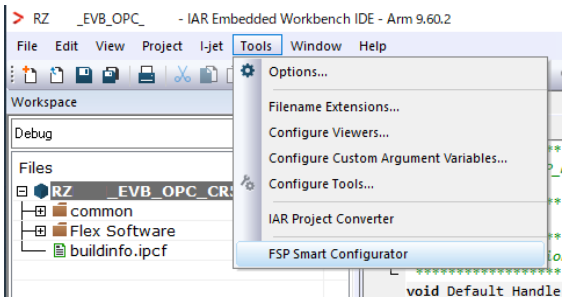


Figure 6.18. Tools tab

*If you wish to change any settings in the sample software from their default values, please configure them at this time by referring to chapter 9.6.1.

Clicking the “Generate Project Content” button on the FSP Smart Configurator generates the code.

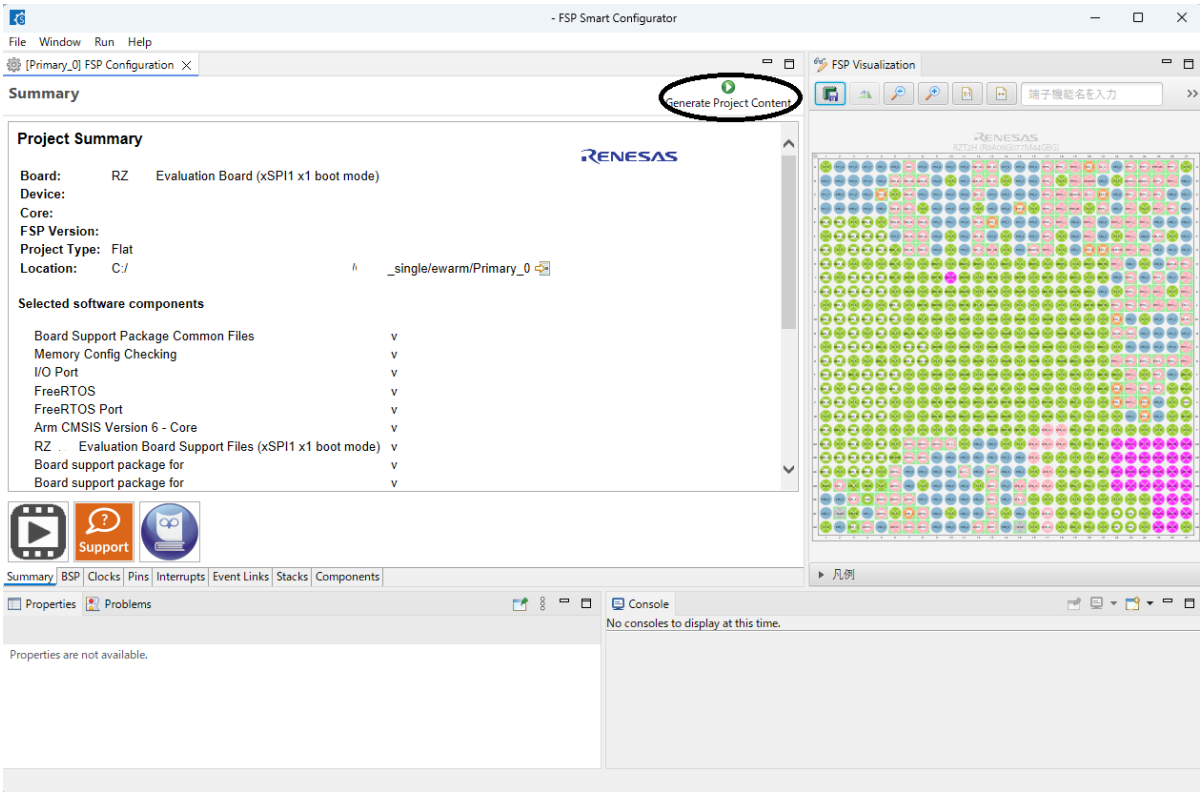


Figure 6.19. Generate Project Content

When debugging using a debugger, add loop processing appropriate for the core at the location below.

“(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
}
```

Then click the Make button in EWARM to build. When the build is complete, build messages are displayed in the build console window, showing the compile target files and the number of errors/warnings.

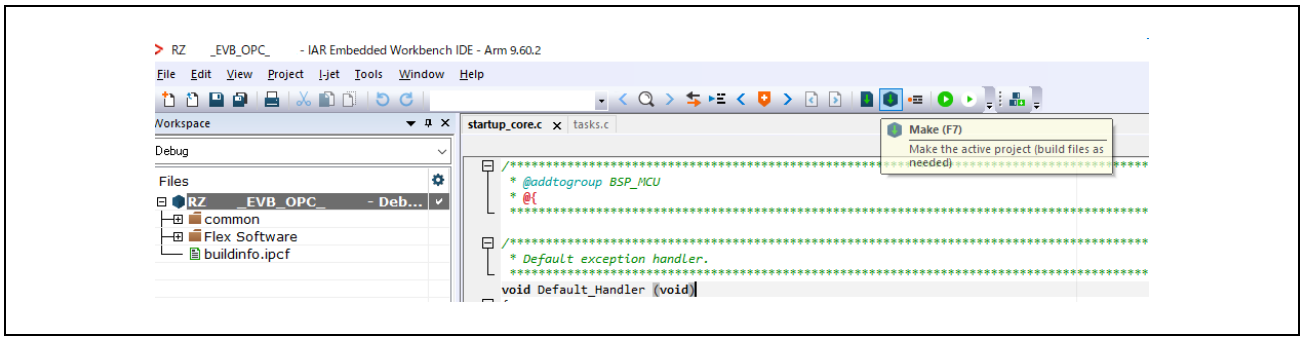


Figure 6.20. Make button

6.1.2.2 Flash writing, debug

Click the "Download and Debug" button to download the built program to the Flash memory on the evaluation board and start debugging.

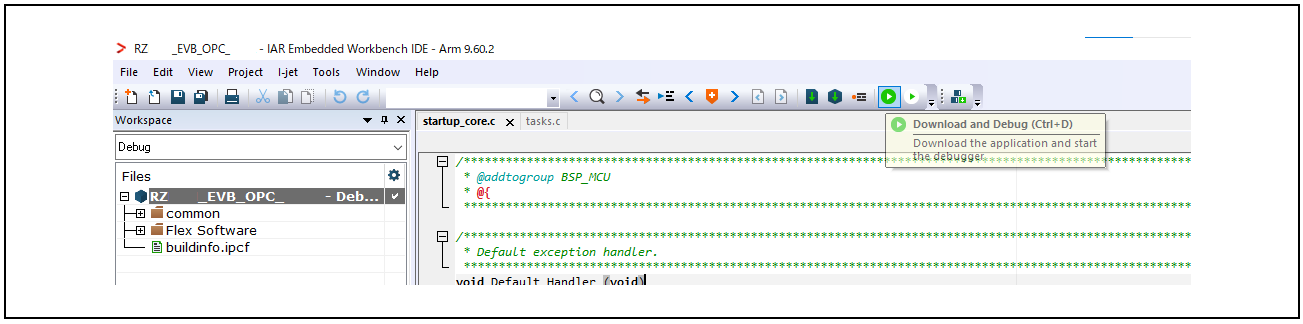


Figure 6.21. Download and Debug

To run the evaluation board alone without using the debugger, click "Stop Debugging" to disconnect the debugger, then turn off the board and turn it on again.

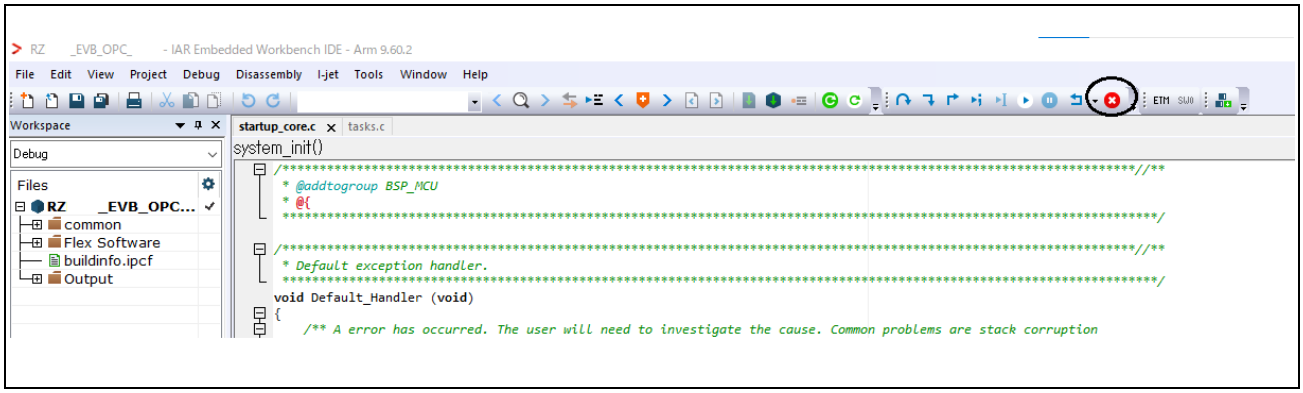


Figure 6.22. Stop Debugging

6.2 When using e² studio

Execute "e2studio.exe" to start e² studio in the following folder (default case) installed:

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

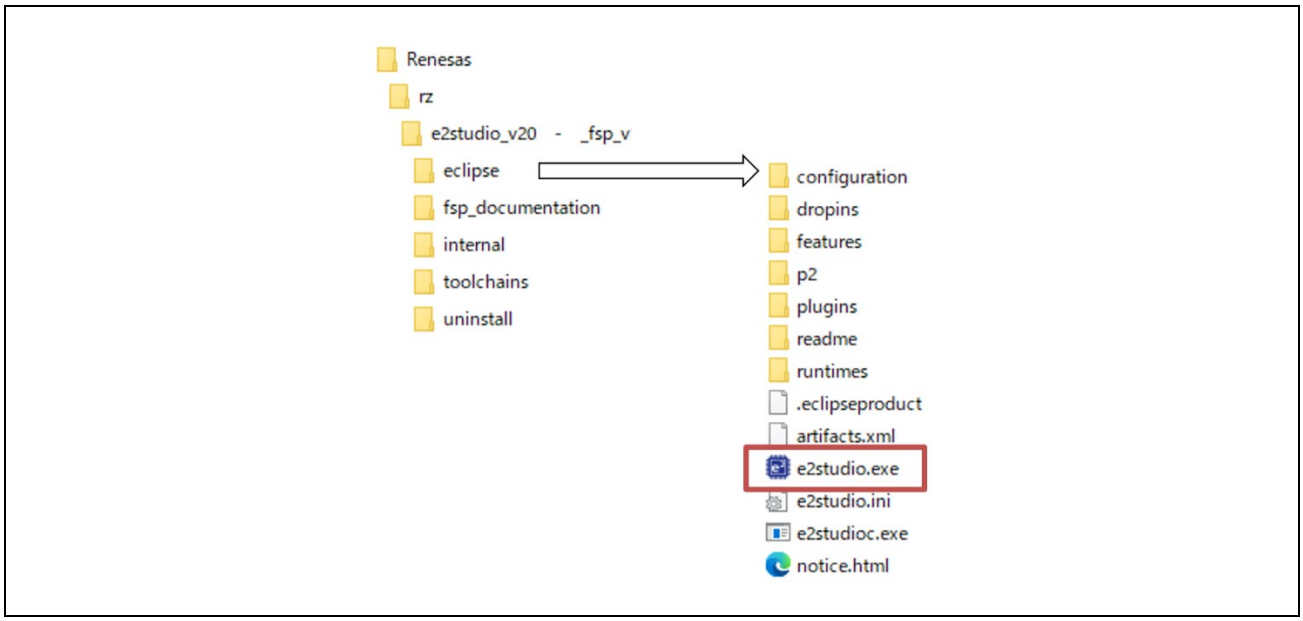


Figure 6.23. Launch project (1)

Import project

Enter any workspace directory and click “Launch”.

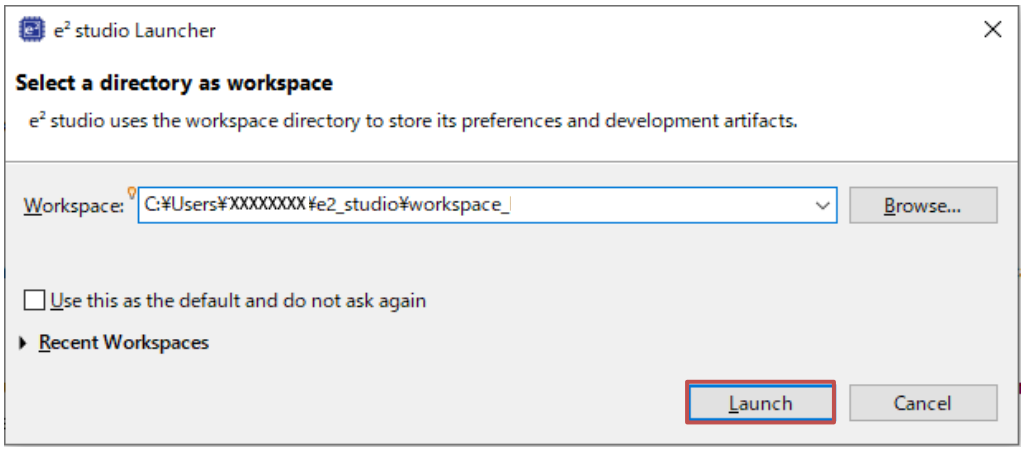


Figure 6.24. Launch project (2)

・Select “Import existing projects”

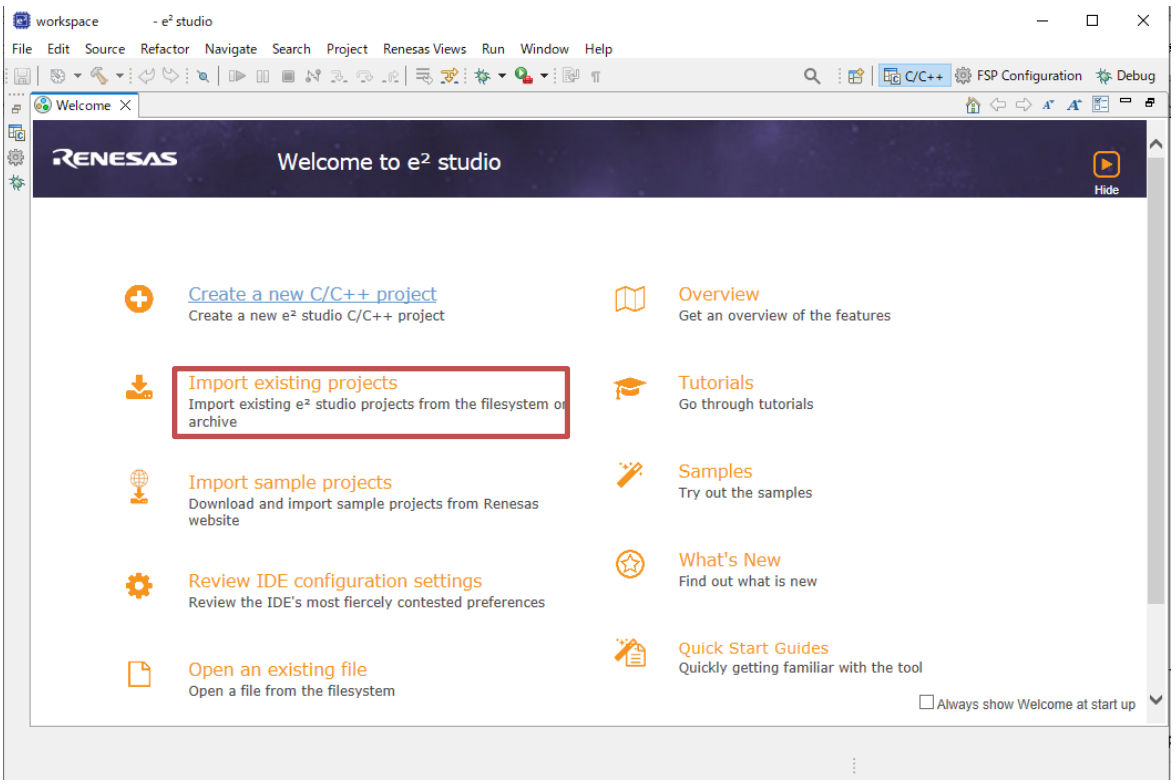


Figure 6.25. Launch project (3)

Click “Browse…” at “Select root directory” and enter the project folder to be imported. For dual-core projects, two projects are displayed.

Single core (CR52) project：RZ***_OPC_EVB(RSK)_rev****\project\CR52_single\e2studio\

Dual core (CR52-CR52) project：RZ***_OPC_EVB(RSK)_rev****\project\CR52_dual\e2studio\

Dual core (CR52-CA55) project：RZ***_OPC_EVB(RSK)_rev****\project\CA55_dual\e2studio\

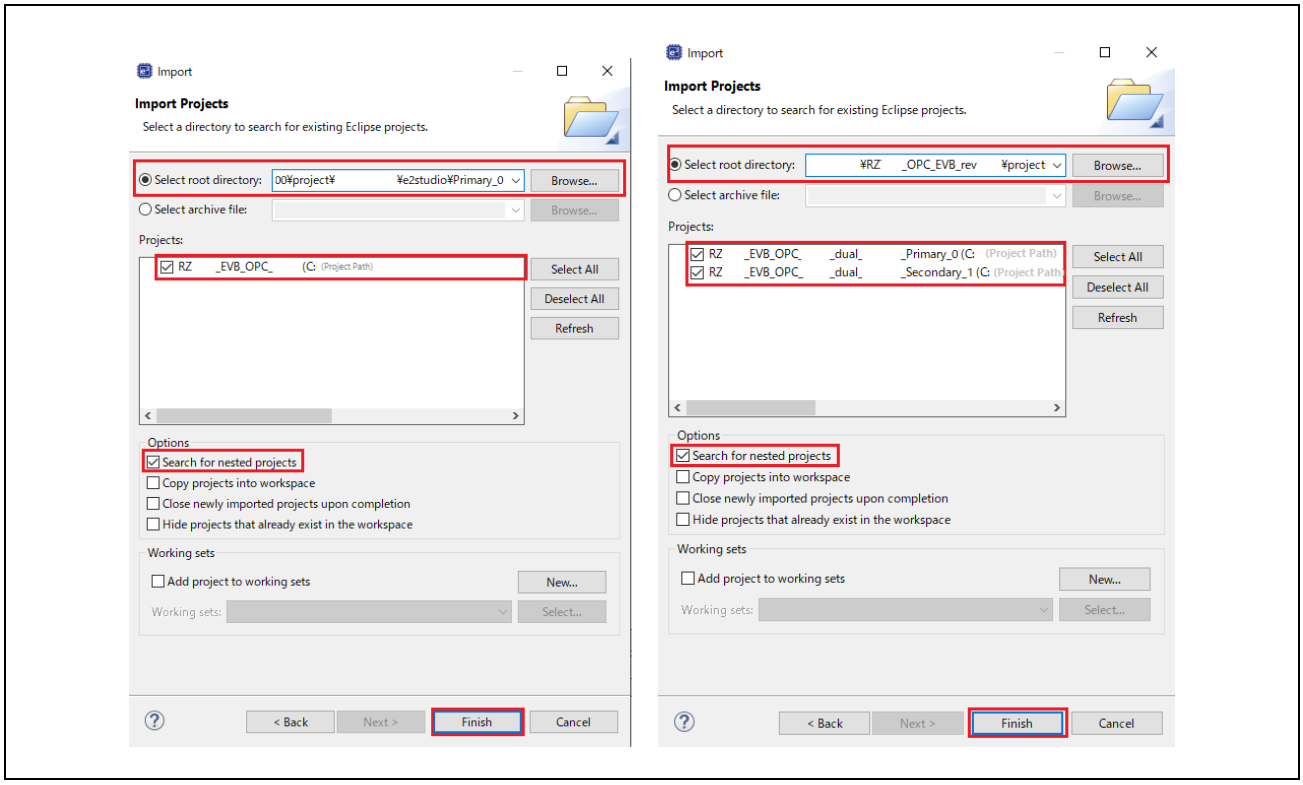


Figure 6.26. Launch project (4)

Clicking “Finish” in Figure 6.26 will start the project import, and the screen will automatically change when finished.

Subsequent steps vary depending on the project used.

6.2.1 For Dual core project

6.2.1.1 Build

* Please refer to the guide in chapter 9.7 when using FSP v4.1.0 or later for RZ/T2M, or when using FSP v4.2.0 or later for RZ/T2H or RZ/N2H.

Double-click the primary project's configuration.xml in the Project Explorer window to open the Smart Configurator.

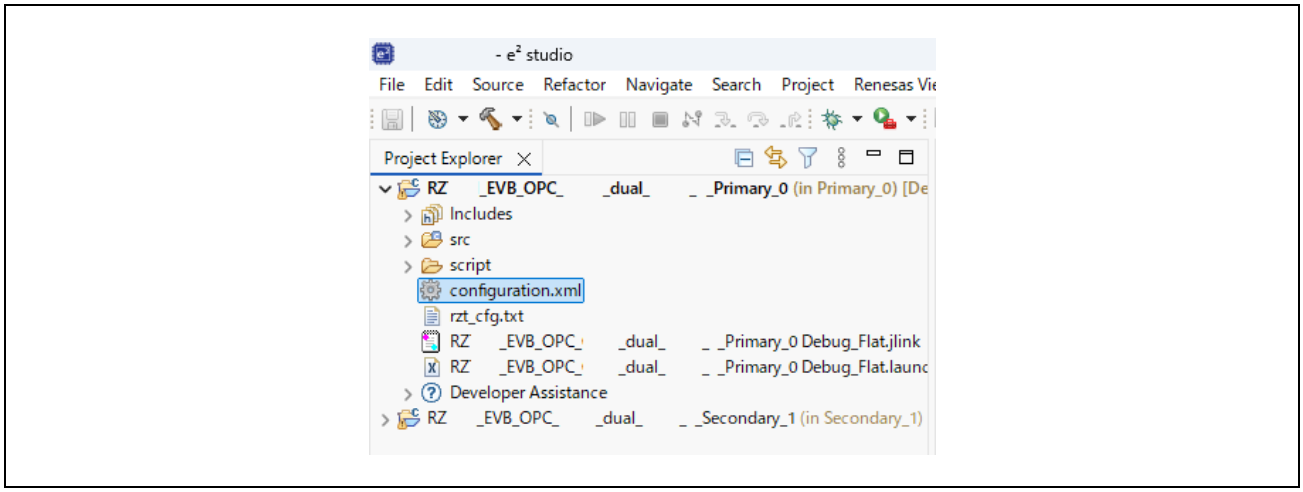


Figure 6.27. Open Smart configurator

Click “Generate Project Content” to generate the source code.

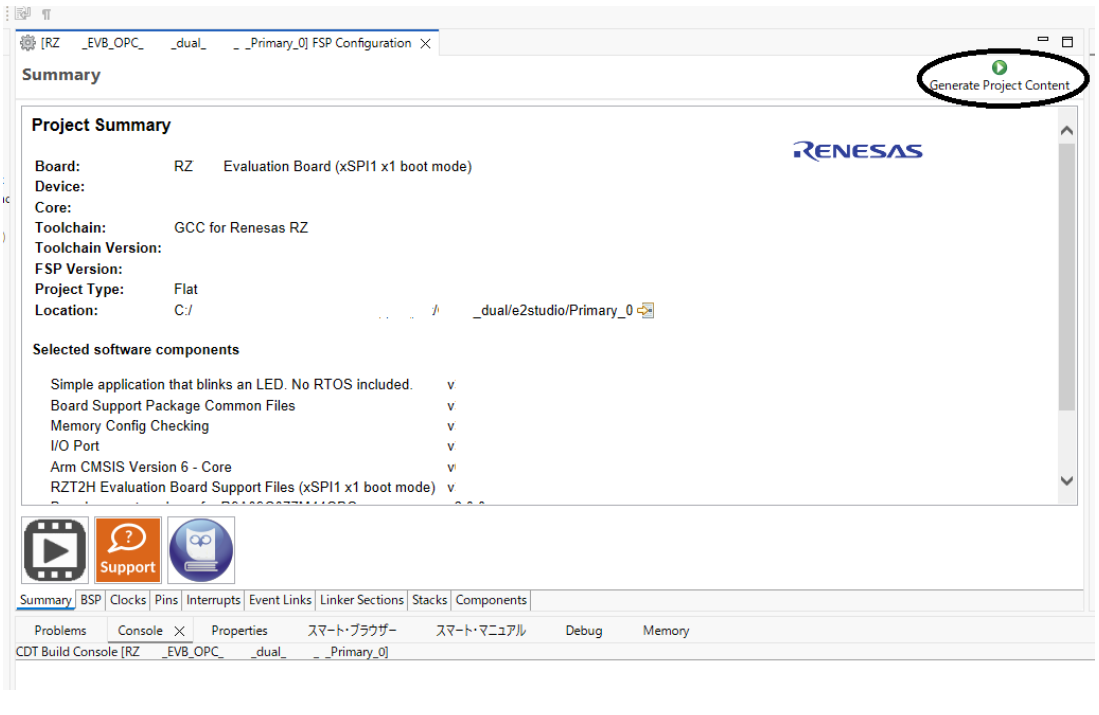


Figure 6.28. Generate Project Content

When debugging using a debugger, add a loop process to the following location.

“(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
}
```

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

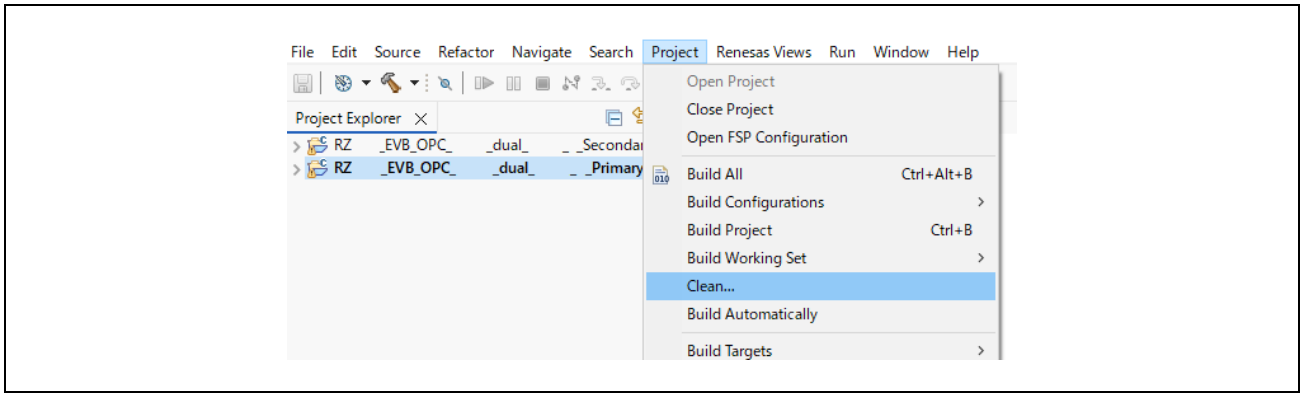


Figure 6.29. Open project Clean…

Select the followings and click Clean to start build.

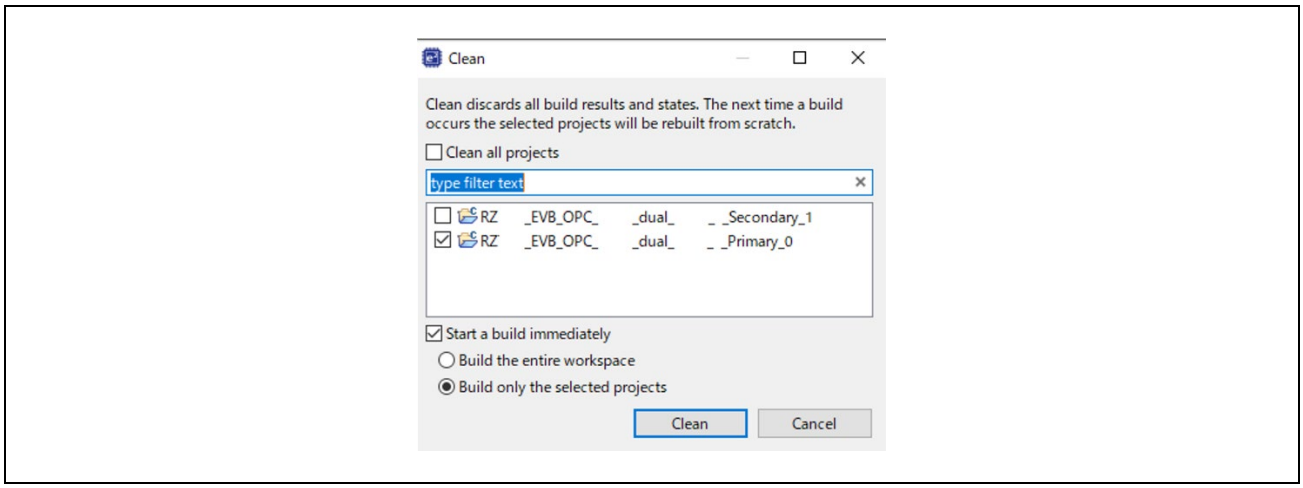


Figure 6.30. Clean and rebuild Primary project

Open the configuration.xml file for the secondary project from the Project Explorer window.

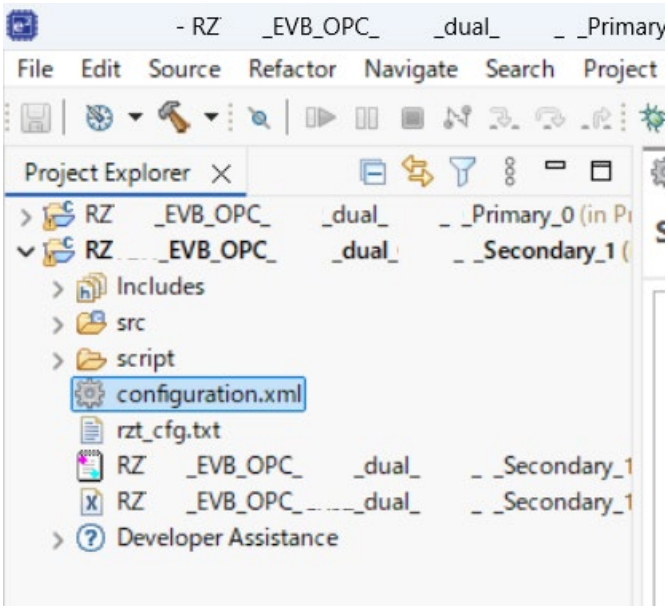


Figure 6.31. Open Smart configurator

※If you wish to change any settings in the sample software from their default values, please configure them at this time by referring to chapter 9.6.1.

Click “Generate Project Content” to generate the source code.

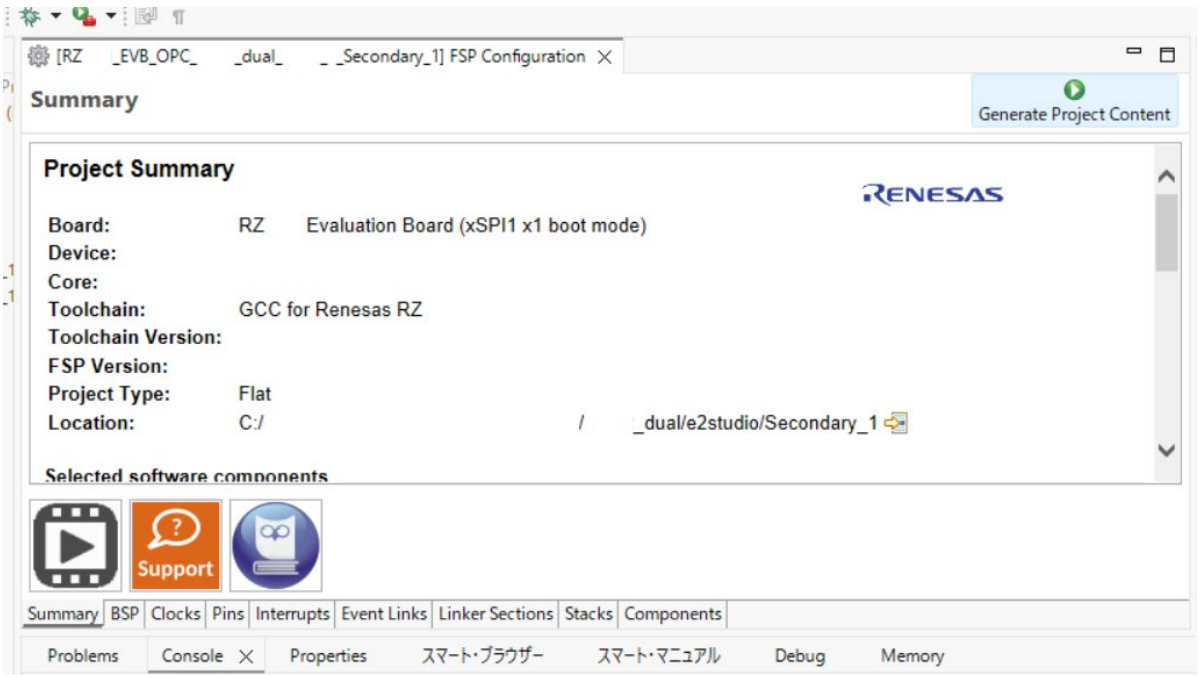


Figure 6.32. Generate Project Content

When debugging using a debugger, add loop processing appropriate for the core at the location below.

“(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
}
```

After selecting the secondary project, click Clean... in the Project menu.

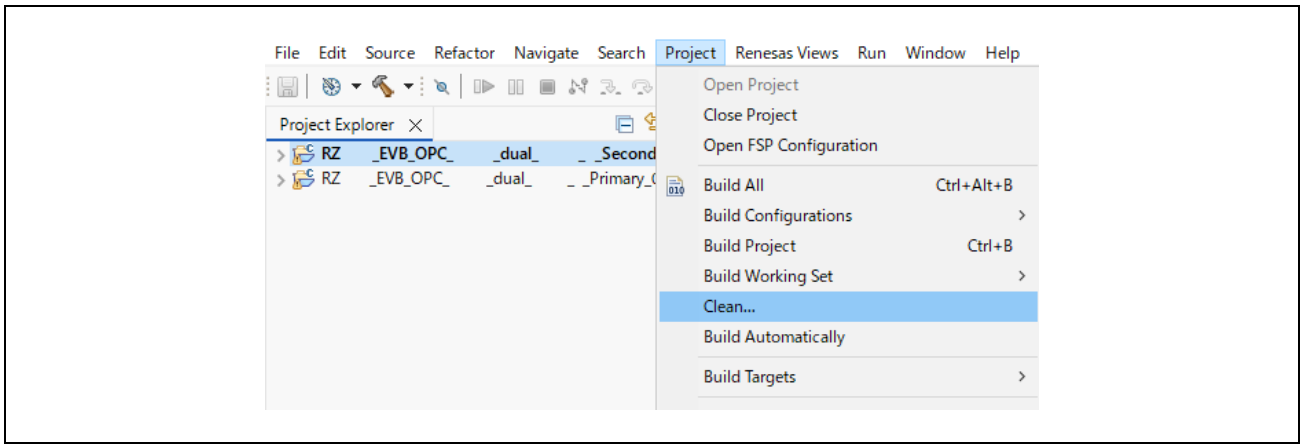


Figure 6.33. Open project Clean…

After configuring the settings as shown below and clicking Clean, the build will begin.

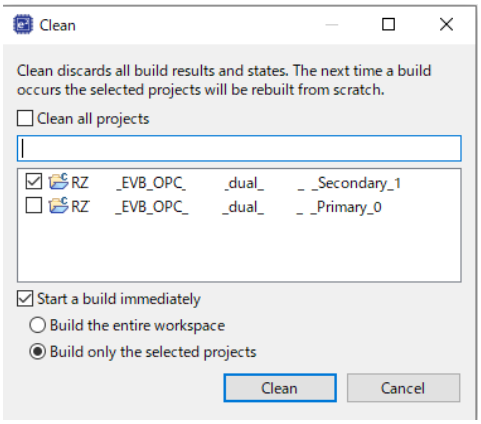


Figure 6.34. Clean and rebuild Secondary project

After confirming that the secondary project's build results show 0 errors, select the primary project in the Project Explorer window and click Clean... in the Project menu.

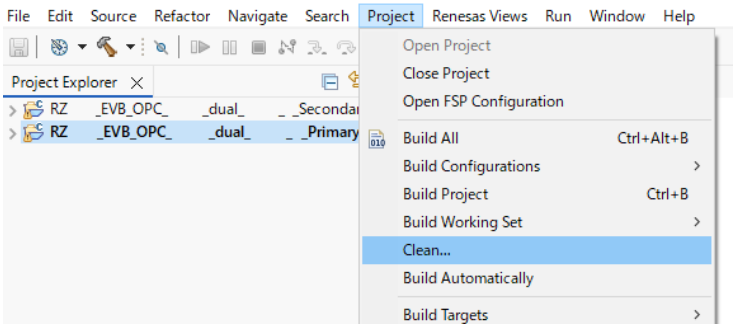


Figure 6.35. Open project Clean…

Select the following and click Clean to start the full build.

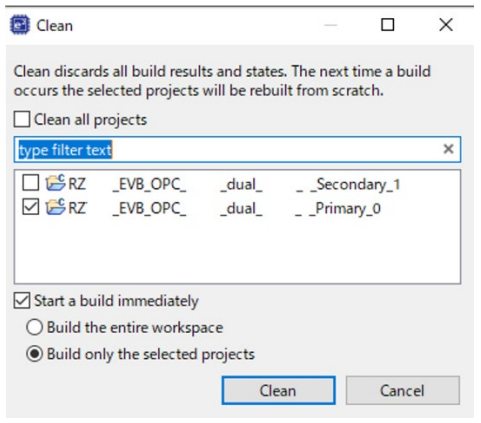


Figure 6.36. Clean and rebuild Primary project

6.2.1.2 Flash writing, debug

The download procedure after the build is completed is shown below.

Select the primary project and click the debug icon to download the program to Flash memory.

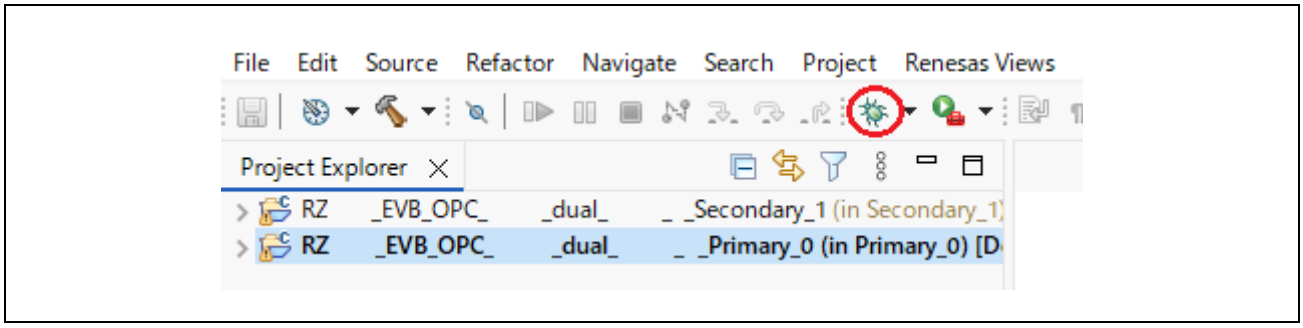


Figure 6.37. Debug Primary

Figure 6.38 window will appear, select “No”.

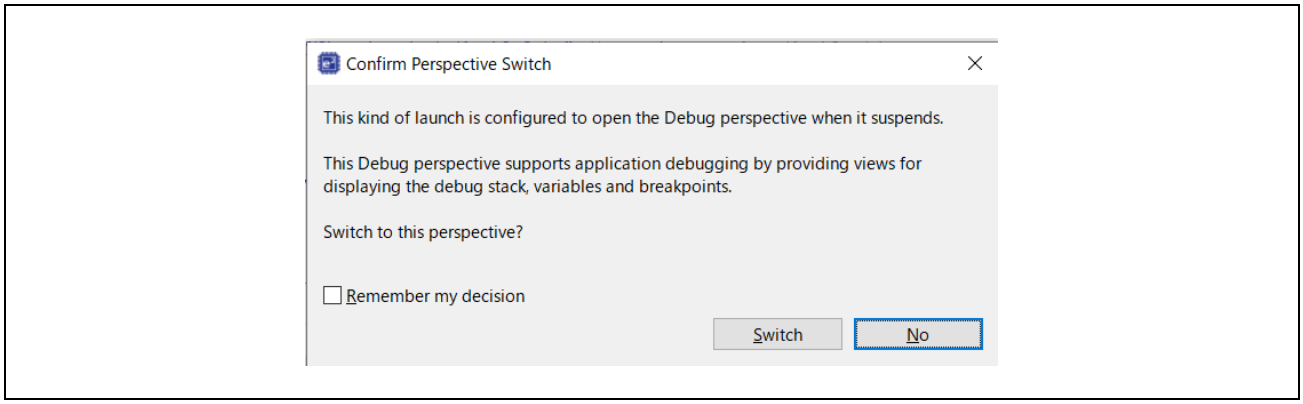


Figure 6.38. Perspective Switch

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

When using the debugger, please follow the steps below.

(1) For CR52_dual project

Select the secondary project and click the debug icon.

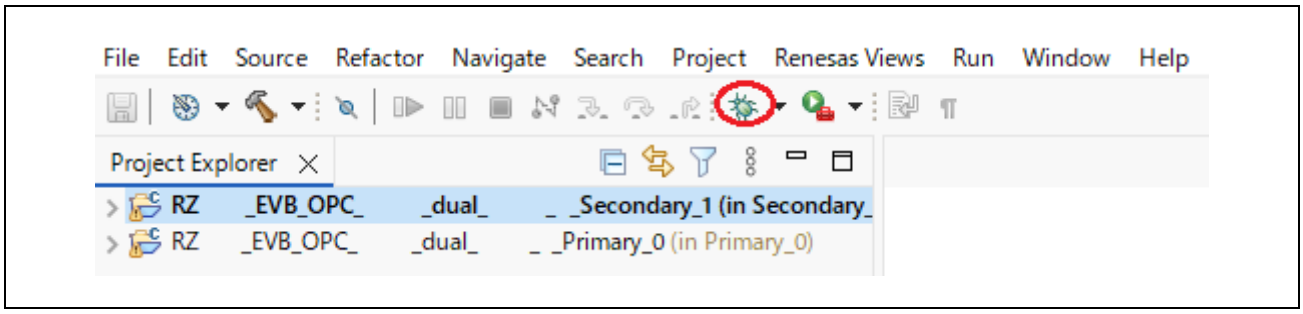


Figure 6.39. Debug Secondary

Figure 6.40 window will appear, select “No”.

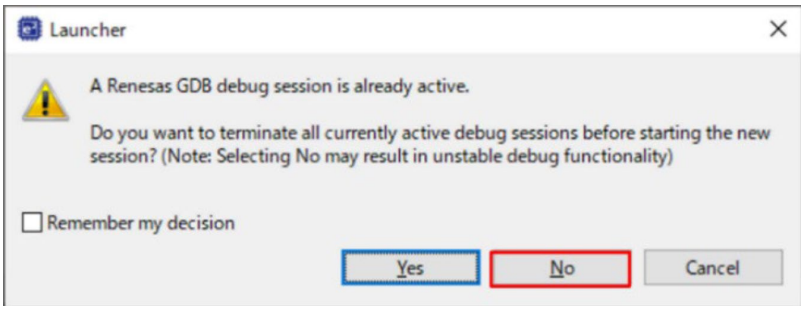


Figure 6.40. Warning Window of Starting Debug Session

Figure 6.41 window will appear, select “Yes”.

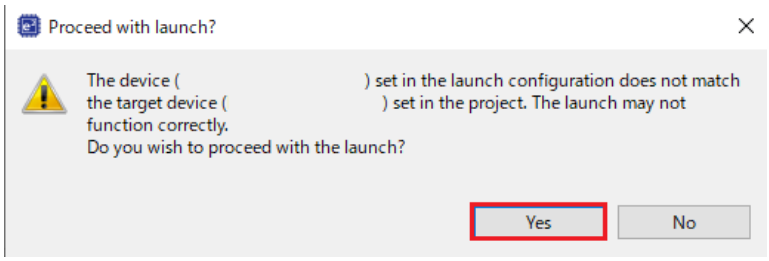


Figure 6.41. Warning Window of Device Name

Figure 6.42 window will appear, select “Switch”.

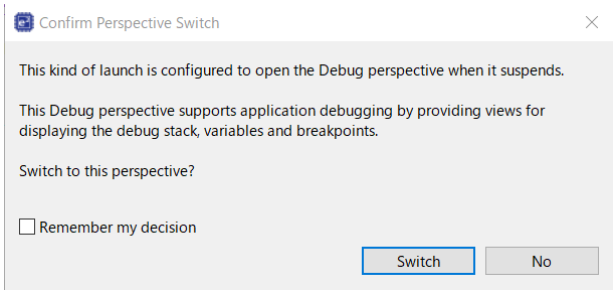


Figure 6.42. Perspective Switch

After selecting a project for each CPU from the Debug tab, click the Resume icon to run the program and break it with main(). Resume again for each project will start the program running on each CPU.

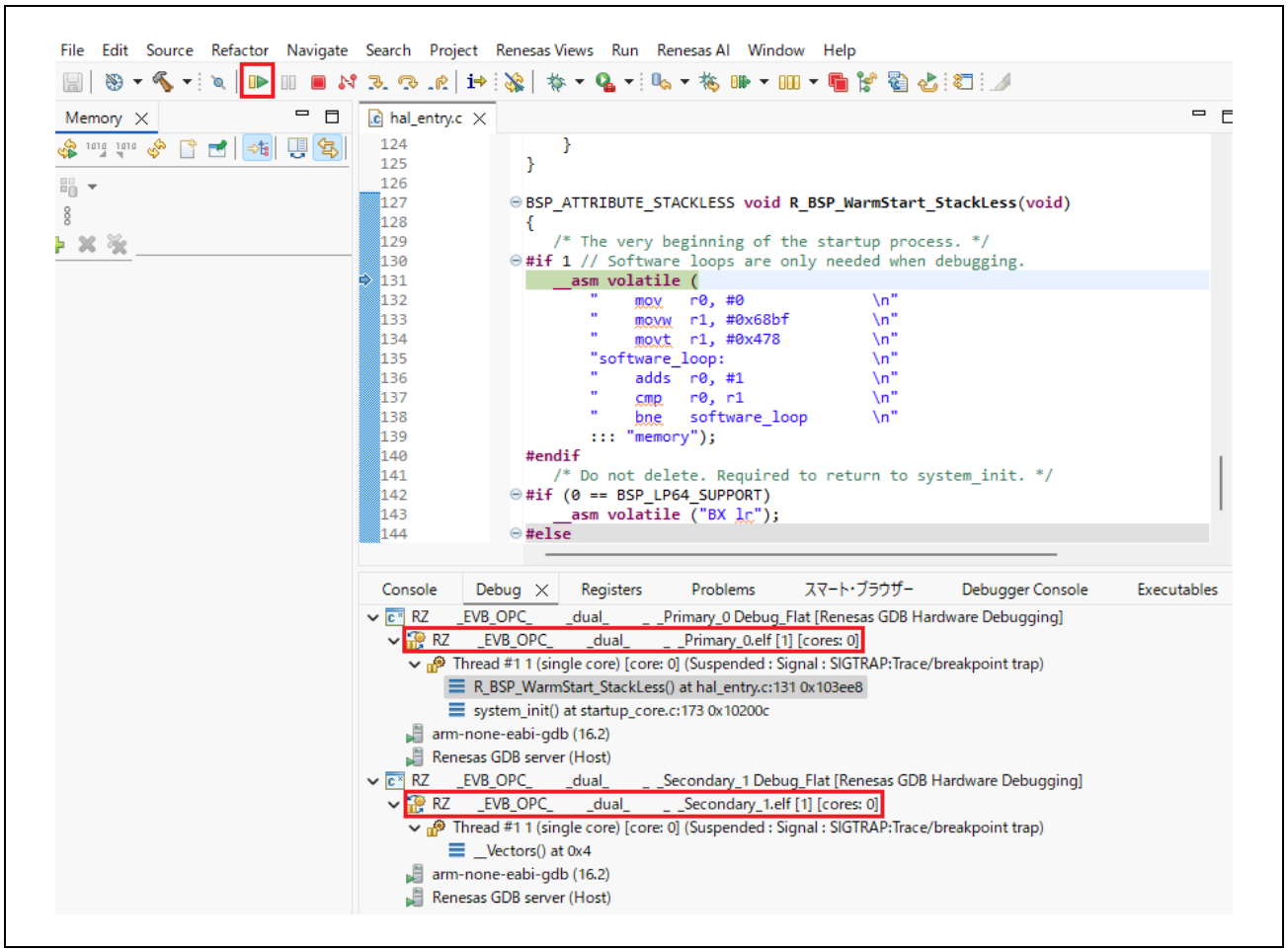


Figure 6.43. Debug

(2) For CA55_dual project

Clicking the Resume icon runs the primary project's program and breaks at main().

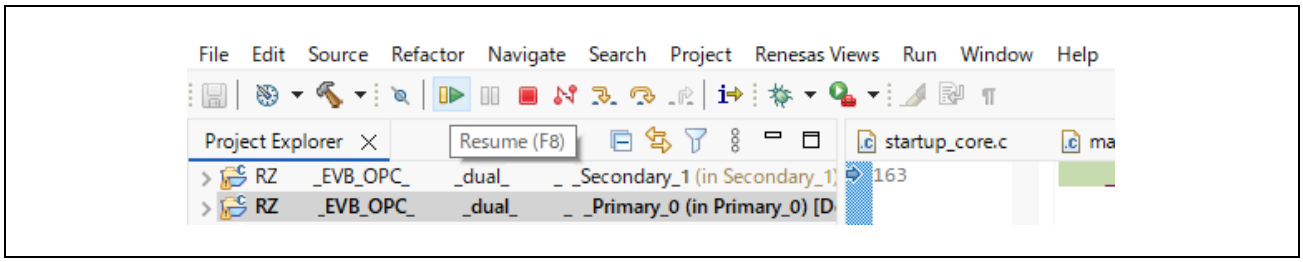


Figure 6.44. Resume Primary project

Select the secondary project and click the debug icon.

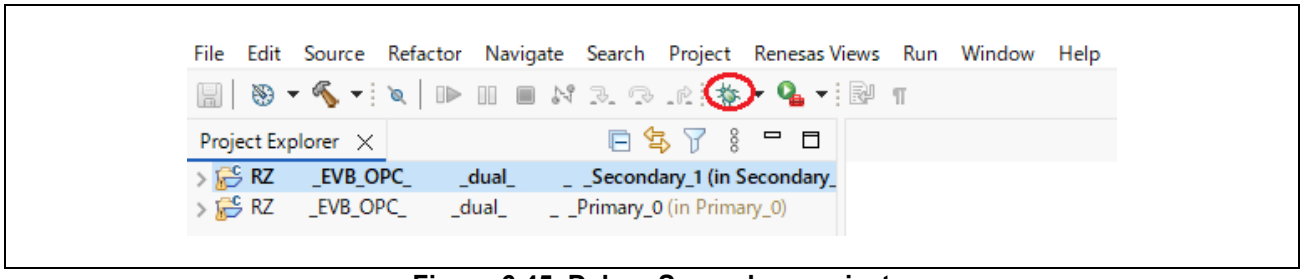


Figure 6.45. Debug Secondary project

Figure 6.46 window will appear, select “No”.

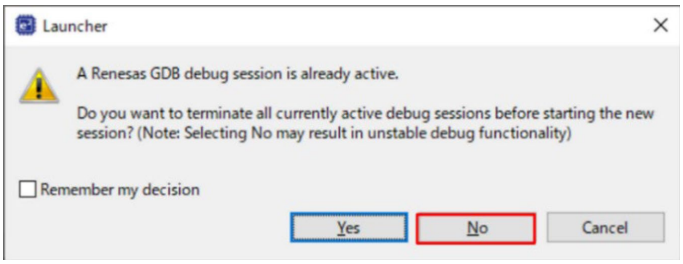


Figure 6.46. Warning Window of Starting Debug Session

Figure 6.47 window will appear, select “Yes”.

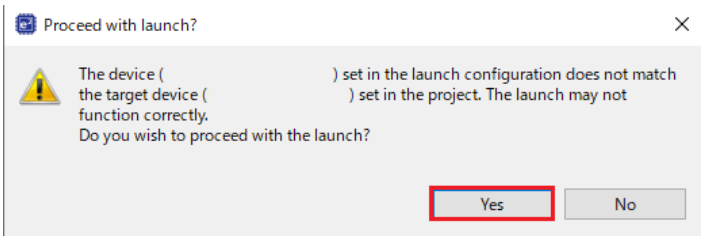


Figure 6.47. Warning Window of Device Name

Figure 6.48 window will appear, select “Switch”.

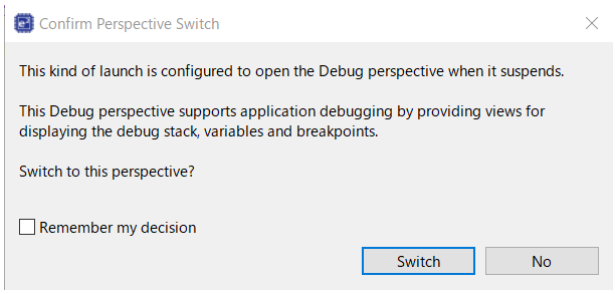


Figure 6.48. Perspective Switch

After selecting the project for each CPU from the Debug tab, click the Resume icon to run the program.

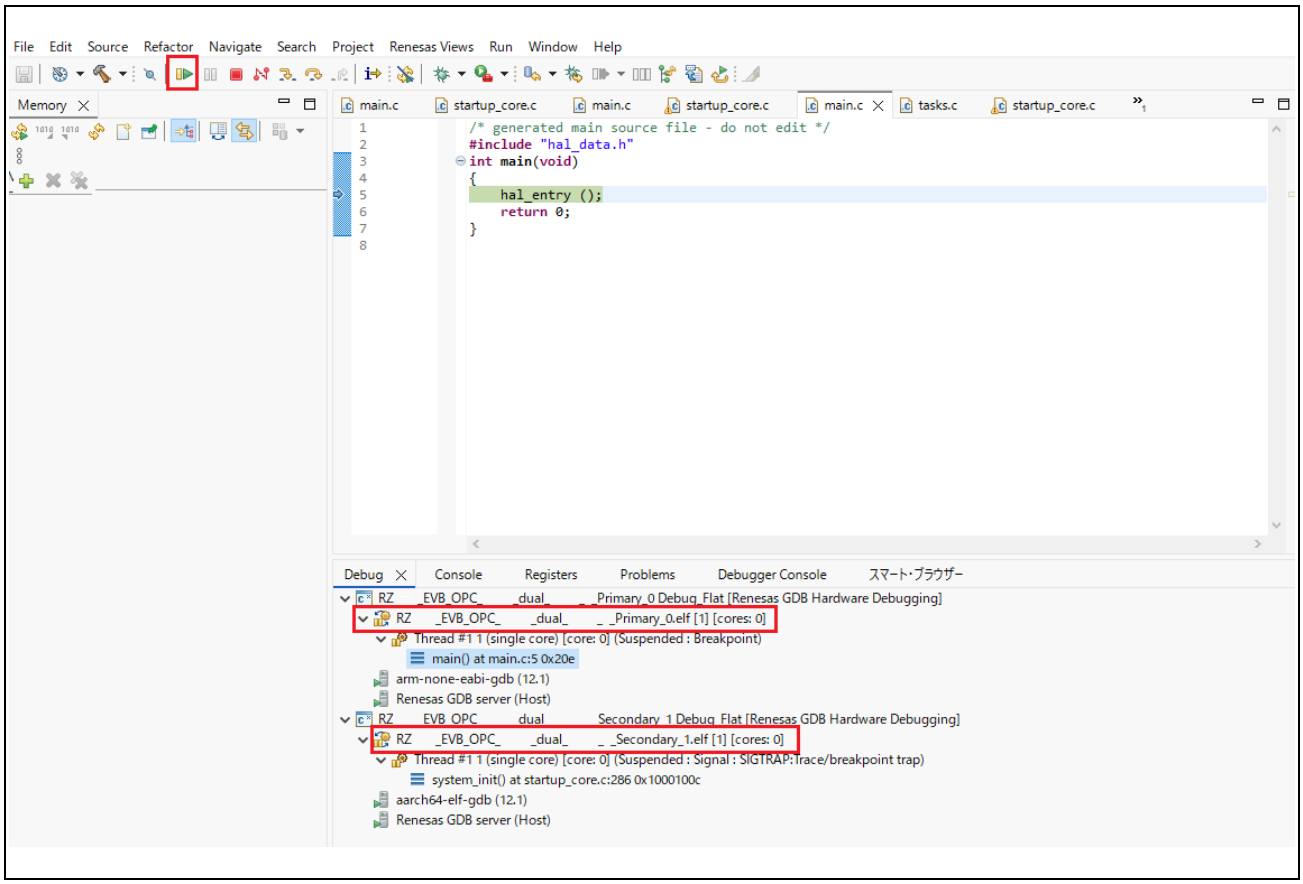


Figure 6.49. Debug

6.2.2 For Single core project

6.2.2.1 Build

* Please refer to the guide in chapter 9.7 when using FSP v4.1.0 or later for RZ/N2L.

Open configuration.xml from the Project Explorer window.

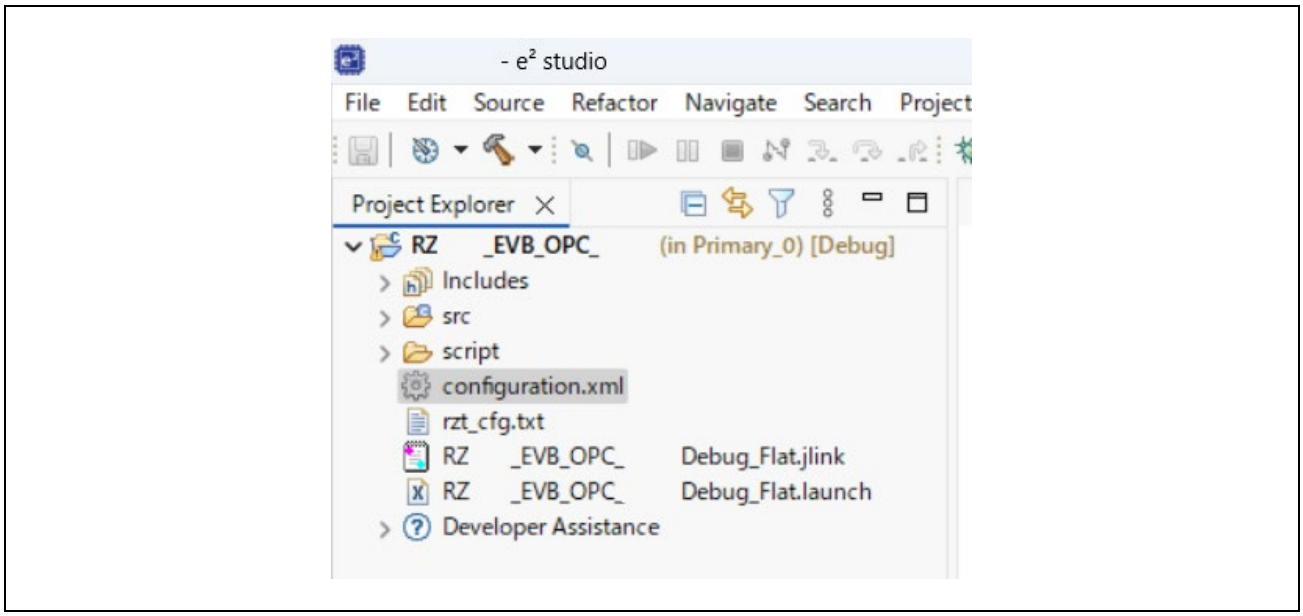


Figure 6.50. Open Smart configurator

* If you wish to change any settings in the sample software from their default values, please configure them at this time by referring to chapter 9.6.1.

Click “Generate Project Content”.

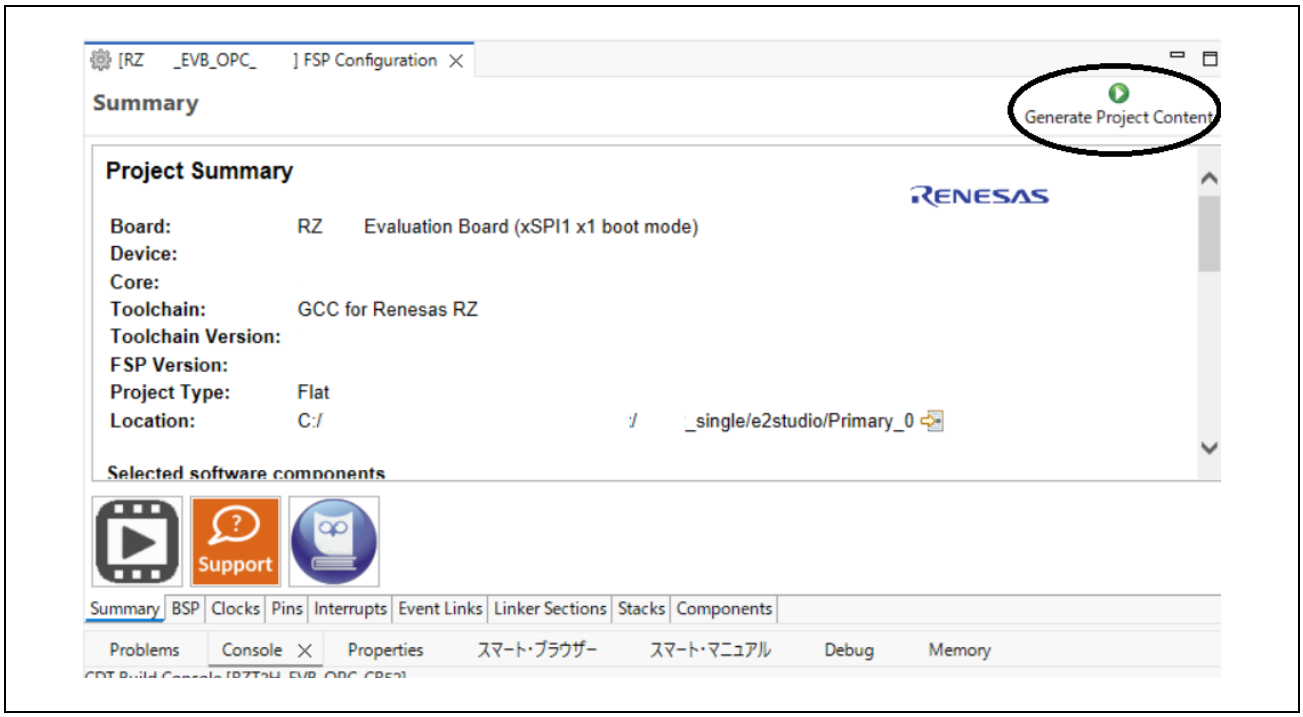


Figure 6.51. Generate Project Content

When debugging using a debugger, add loop processing appropriate for the core at the location below.

“(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
}
```

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

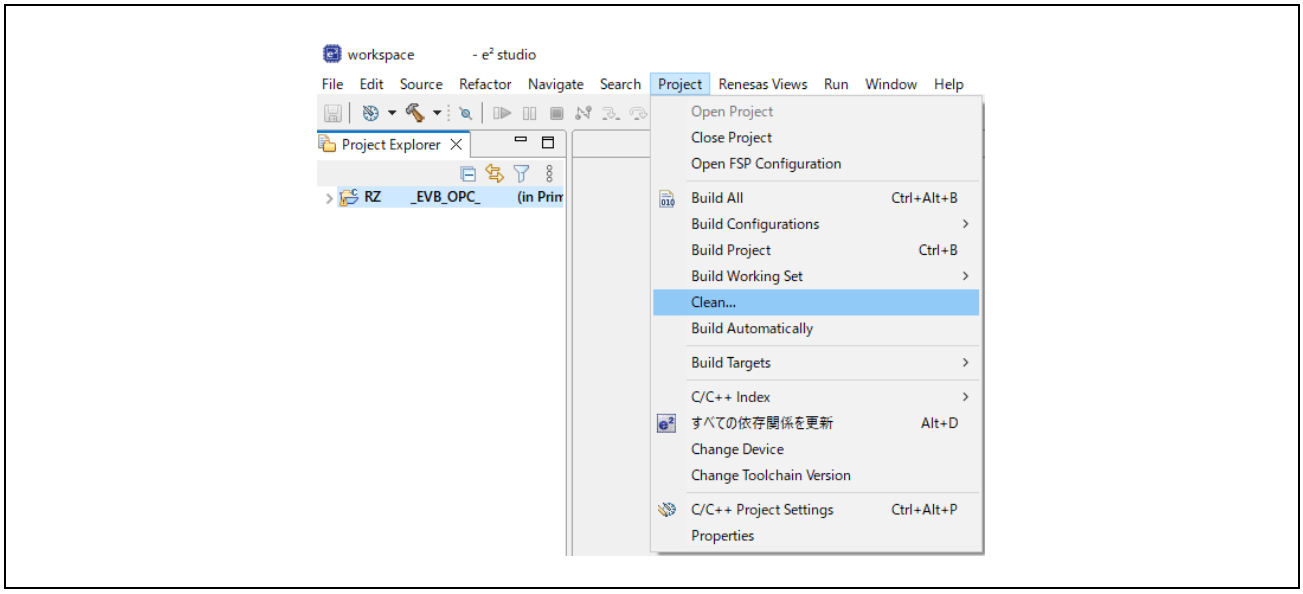


Figure 6.52. Open project Clean…

Select as shown in the figure below and click "Clean" to start build.

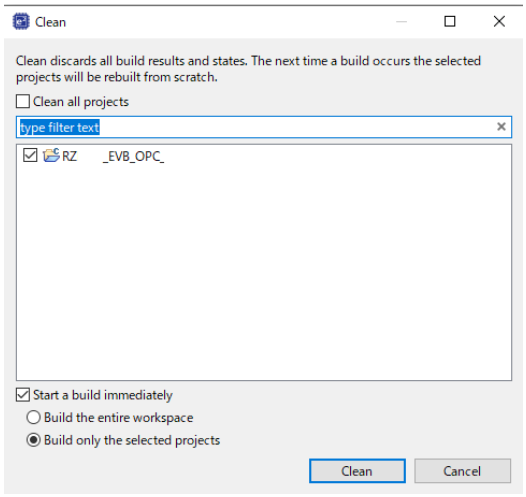


Figure 6.53. Clean and rebuild

6.2.2.2 Flash writing, debug

The download procedure after the build is shown below.

Select a project from “Project Explorer” and click the debug icon to download the program to flash memory.

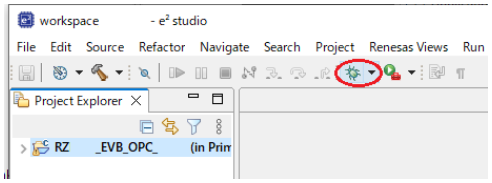


Figure 6.54. Run menu Debug As

Click Switch to change to debug view.

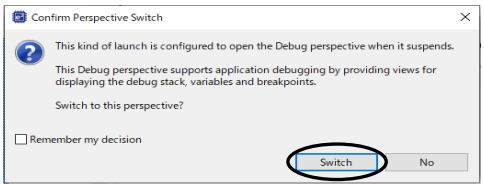


Figure 6.55. Perspective Switch

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

When using the debugger, click the "resume" after switching to the Debug screen.

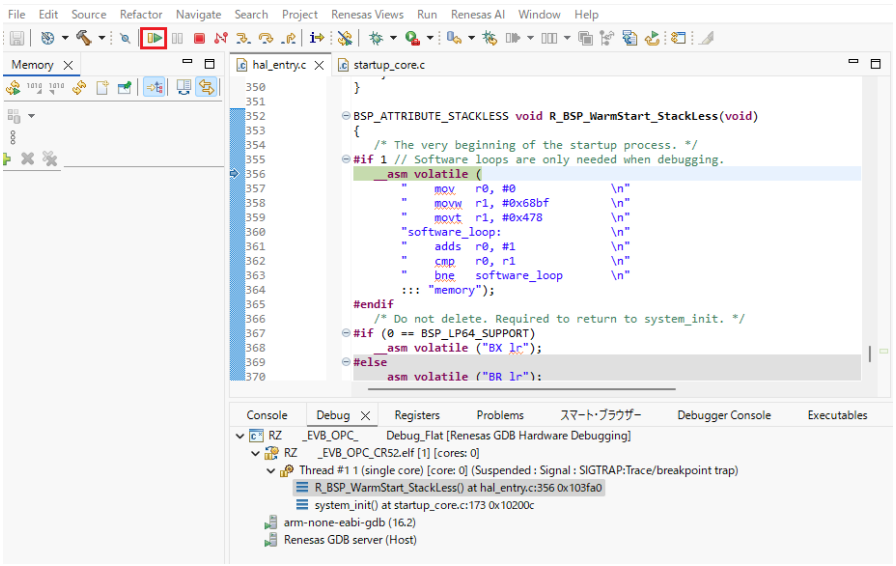


Figure 6.56. Debug view

7. Demonstration of the Sample Application

7.1 Application Behavior

This sample software consists of the OPC UA Server and the sample application that periodically reads from and writes to its Node. For detailed specifications of the software configuration, refer to chapter 8.1.

For operation verification, either connect a USB flash drive containing the files described in chapter 2.1.4 to the evaluation board, or use software built with *ENABLE_FREERTOS_PLUS_FAT* set to 0.

7.2 OPC UA Operation

This chapter explains how to verify the operation of OPC UA functions.

7.2.1 Server

The role of each device when checking OPC UA Server operation is shown in Figure 7.1.

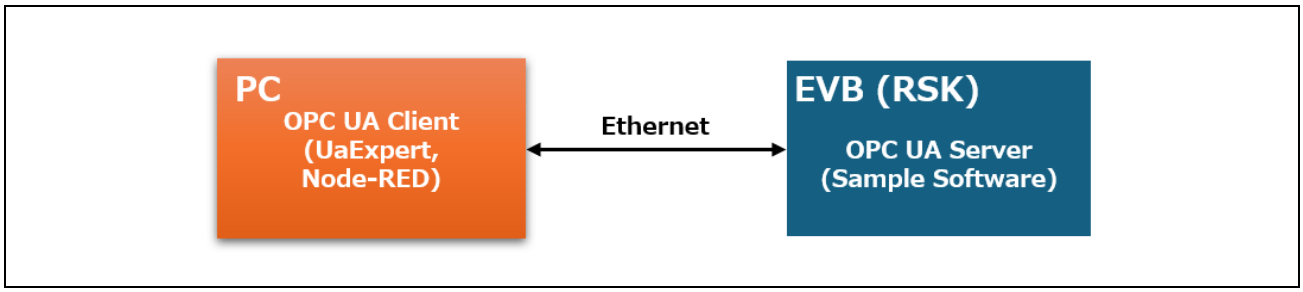


Figure 7.1. Connection Block Diagram for OPC UA Server

7.2.1.1 Without security function

(1) When using UaExpert

After turning on the power of evaluation board, wait until the time synchronization by NTP communication and the initialization are completed. It takes about 15 seconds after power-on. The Green LED (T2M, N2L, T2H:LED0, N2H:LED3) on the evaluation board lights up when the time synchronization is completed.

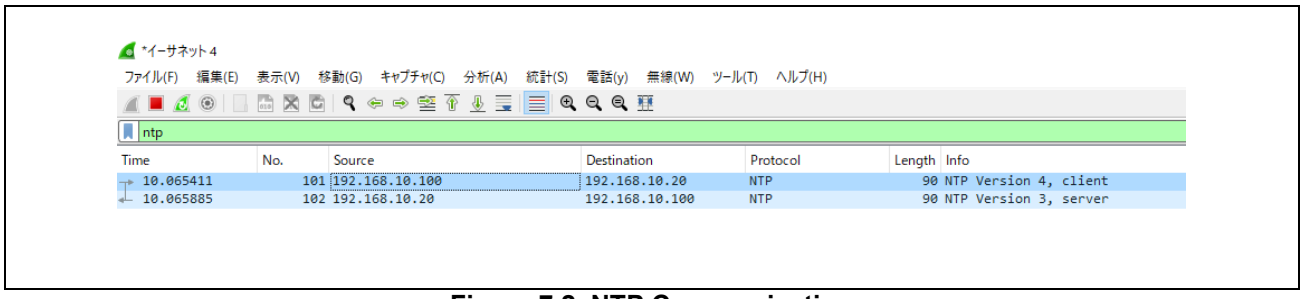


Figure 7.2. NTP Communication

Launch UaExpert and click on the tool bar in UaExpert.

Open the Advanced tab, set the “Endpoint Url” to "opc.tcp://192.168.10.100:4840", select “Anonymous”. Check "Connect Automatically" and then click OK at the end.

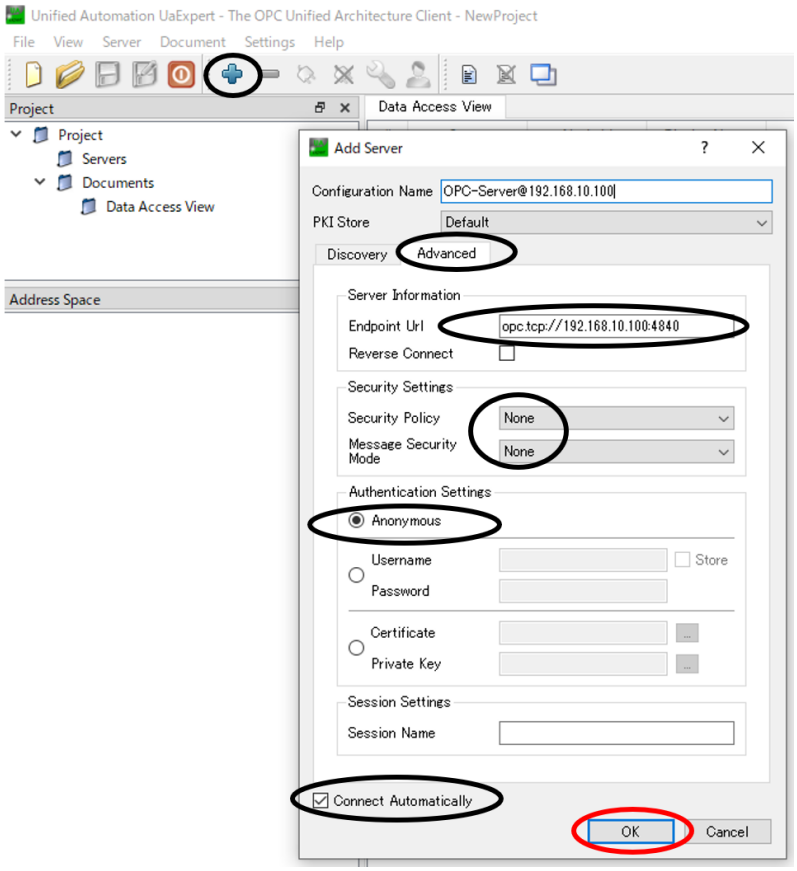


Figure 7.3. UaExpert - Add server

When the OPC UA Server is connected, an indicator icon is displayed in the “Project” window to show that the server is connected. “Renesas_RZ” displayed under the Object tree in the “Address Space” window is a Node of OPC UA server.

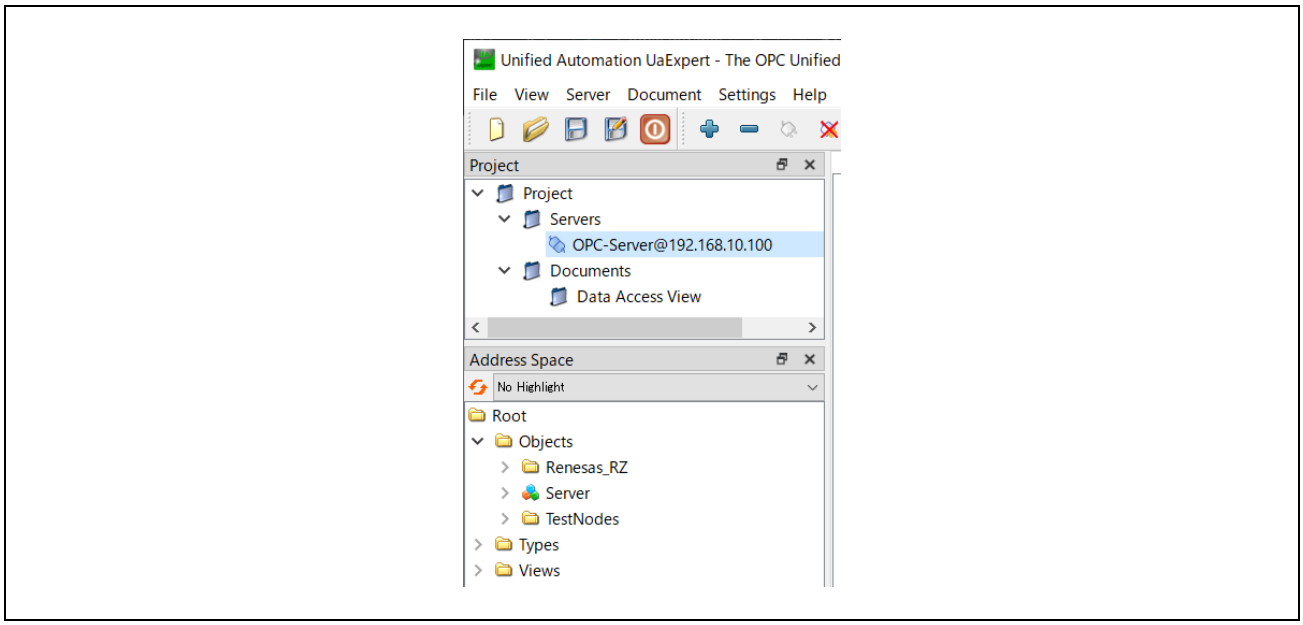


Figure 7.4. UaExpert OPC UA Server connection

As shown in Figure 7.5, drag and drop the *Root>Objects>Renesas_RZ* node in the “Address Space” window to the “Data Access View” window to read the values of each Nodes. See chapter 8.1 for the specification of each Node Value.

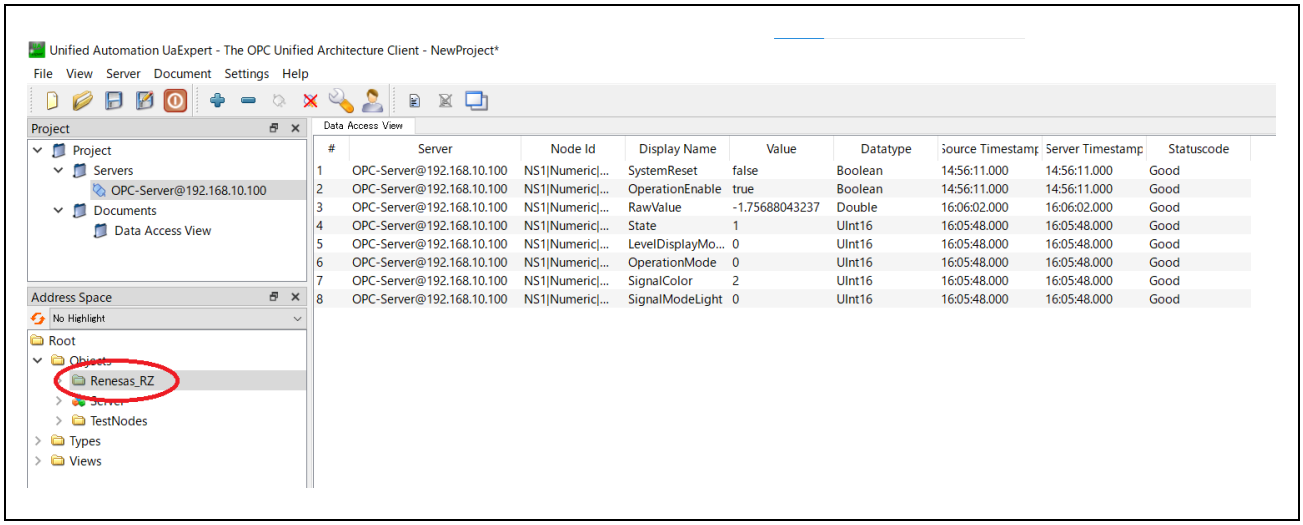


Figure 7.5. Reading Nodes

(2) When using Node-RED

Launch Node-RED following the same procedure as in section 5.8. Click Import from the menu button.

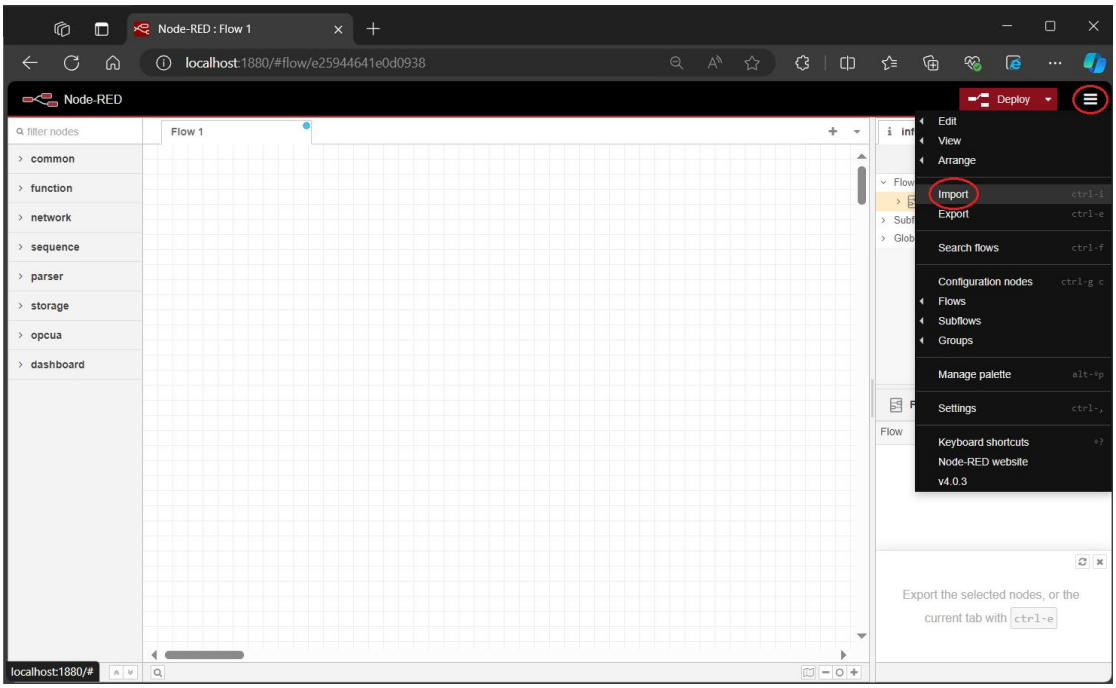


Figure 7.6. Import

Click “select a file to import” and select “node-red_opcua_v2.1.json” included in the package, then click the Import button.

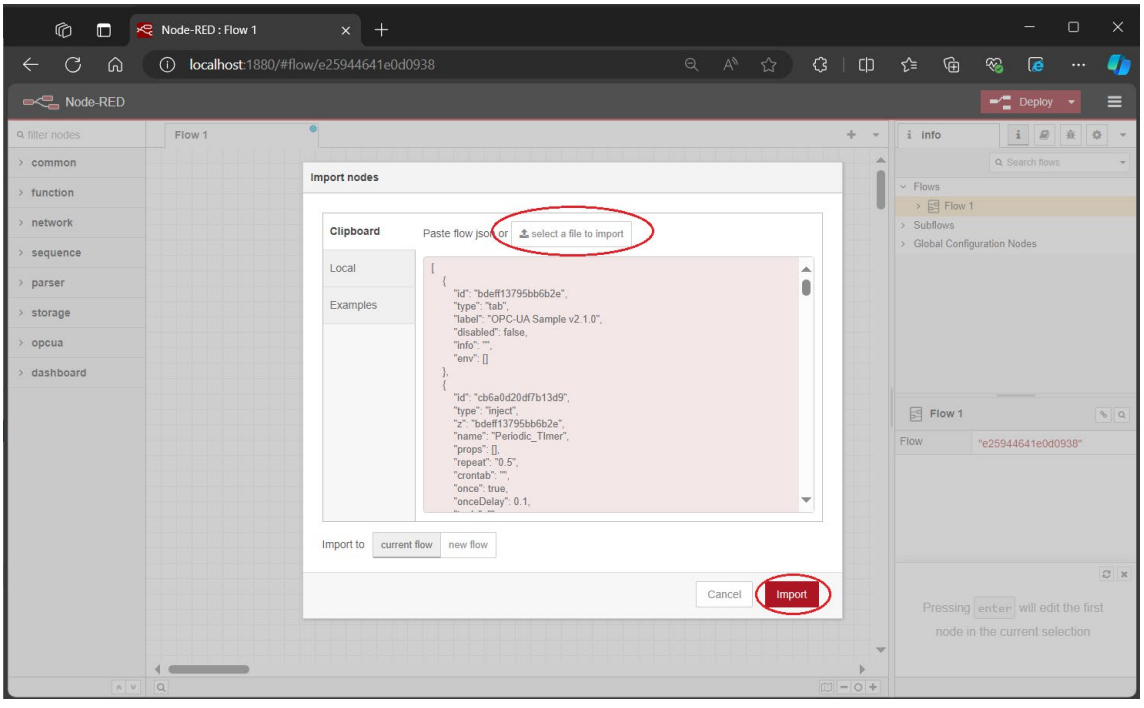


Figure 7.7. Import .json file

After importing the flow, click “Deploy” in the upper right corner of the screen.

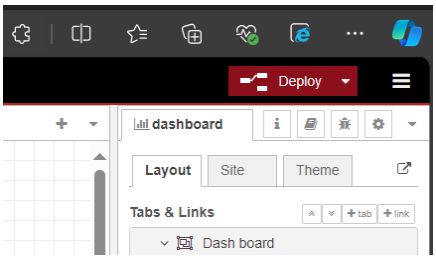


Figure 7.8. Deploy

If the connection with the OPC UA Server is successful, “active reading” is displayed under the OPC UA Client Node.

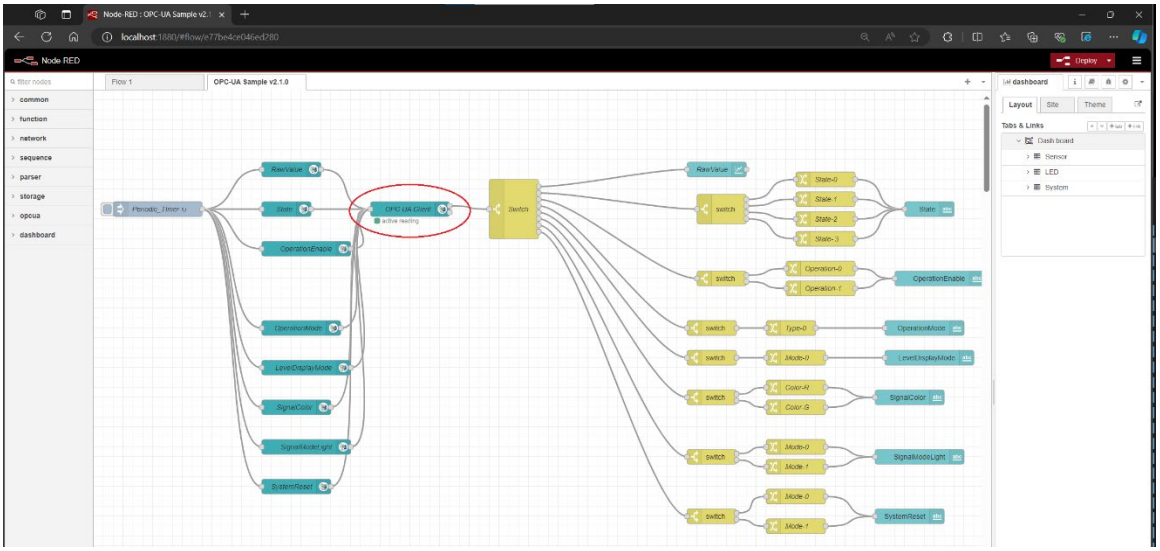


Figure 7.9. Connection Successful

Click on the Dashboard from ▼ icon in the upper right corner of the screen.

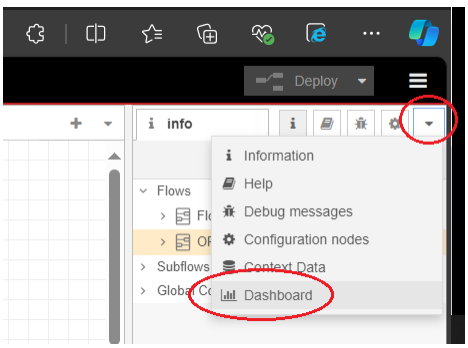


Figure 7.10. Open Dashboard (1)

Click icon on the upper right corner of the screen.

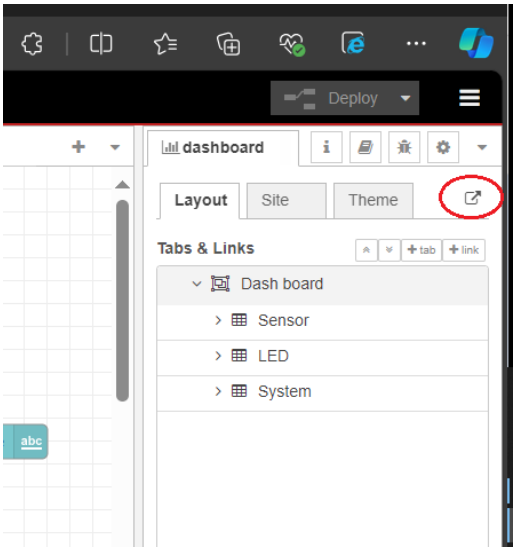


Figure 7.11. Open Dashboard (2)

The Dashboard will open in a separate window, displaying the value of each Node.

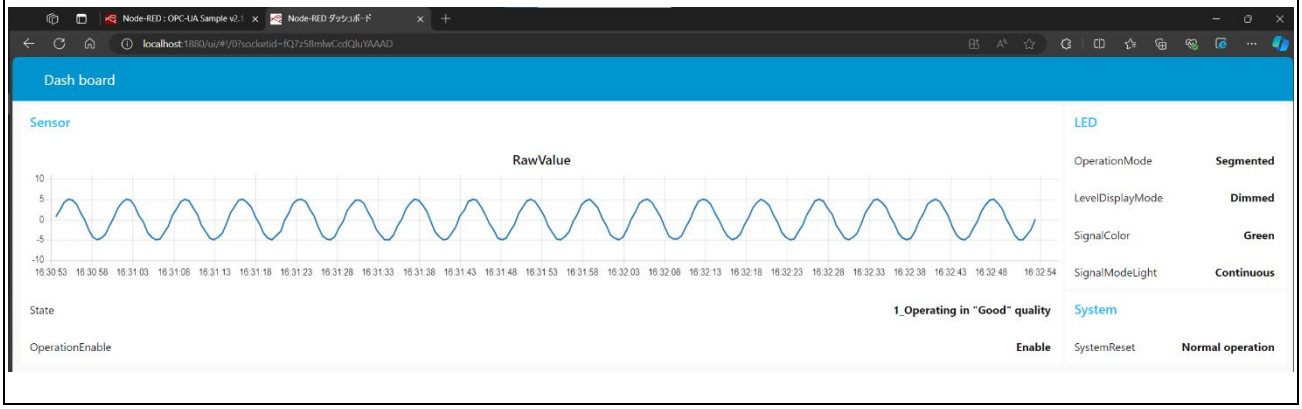


Figure 7.12. Dashboard

7.2.1.2 With security function

After turning on the power of evaluation board, wait until the time synchronization by NTP communication and the initialization are completed. It takes about 15 seconds after power-on. The green LED (T2M, N2L, T2H:LED0, N2H:LED3) on the evaluation board lights up when the time synchronization is completed.

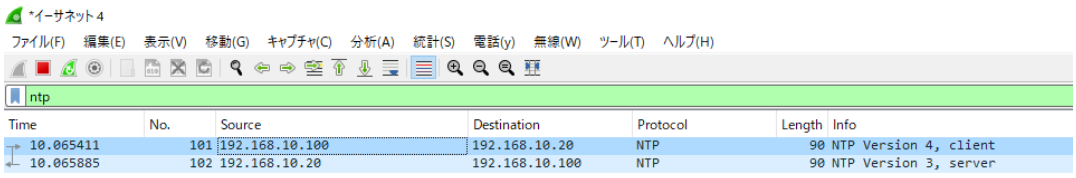


Figure 7.13. NTP Communication

Copy the “uaexpert.der” file to the \unifiedautomation\uaexpert\PKI\own\certs folder, and copy the “uaexpert_key.pem” file to the \unifiedautomation\uaexpert\PKI\own\private folder on the PC. These files are in “keyfiles.zip” attached with the package.

In UaExpert, click on the toolbar. In the Advanced tab, set the “Endpoint Url” to "opc.tcp://192.168.10.100:4840" and select the desired Security Policy and Security Mode for Security Setting. Select any desired option for the Authentication Settings. If *Username/Password* is selected, enter a username and password pair listed in Table 2.3. If *Certificate/Private Key* is selected, choose *uaexpert_usr.der* and *uaexpert_key_usr.pem* included in *keyfiles.zip*.

Check the “Connect Automatically” box, then click OK.

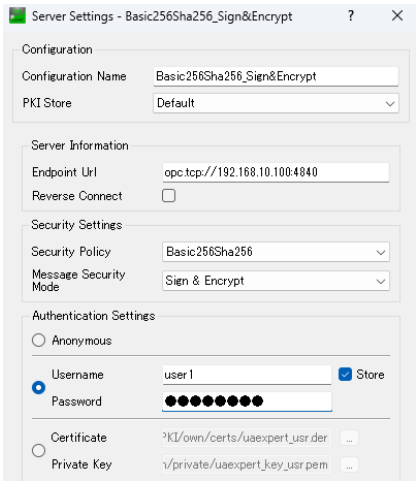


Figure 7.14. UaExpert Add server

When connecting for the first time, the window shown in Figure 7.15 will appear. Click “Trust Server Certificate” and “Continue”.

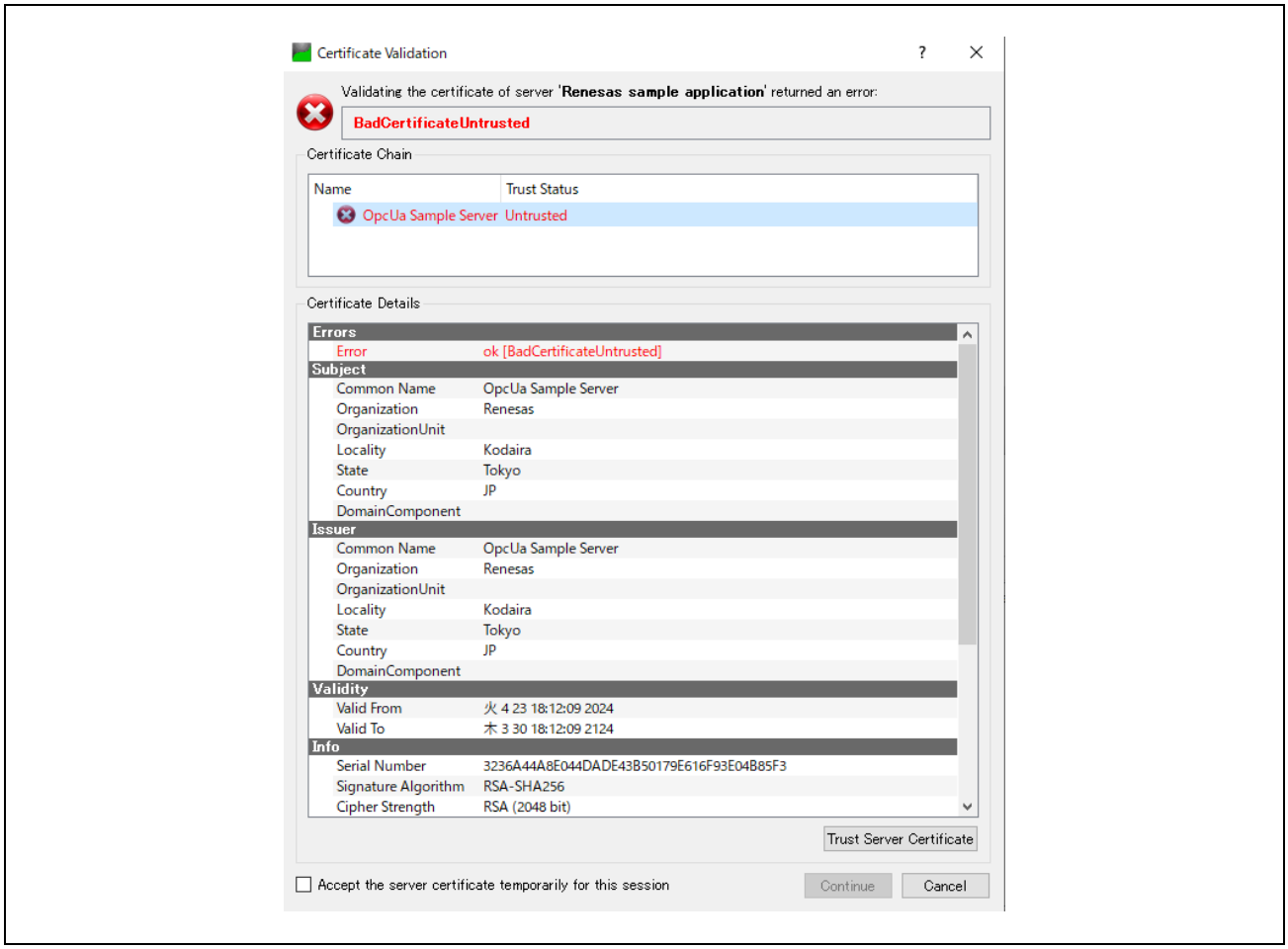


Figure 7.15. Certificate Validation

If a timeout occurs during Connect, go to "Settings" > "Configure UaExpert..." and increase the value of "General.ConnectTimeout"

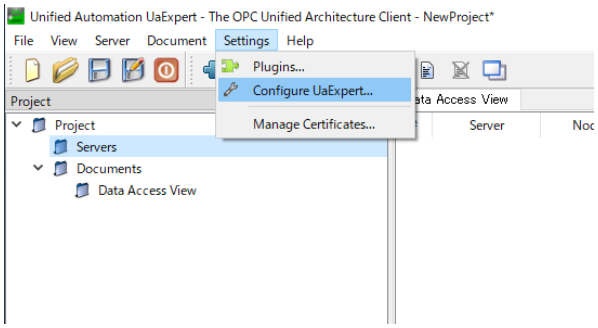


Figure 7.16. Configure UaExpert

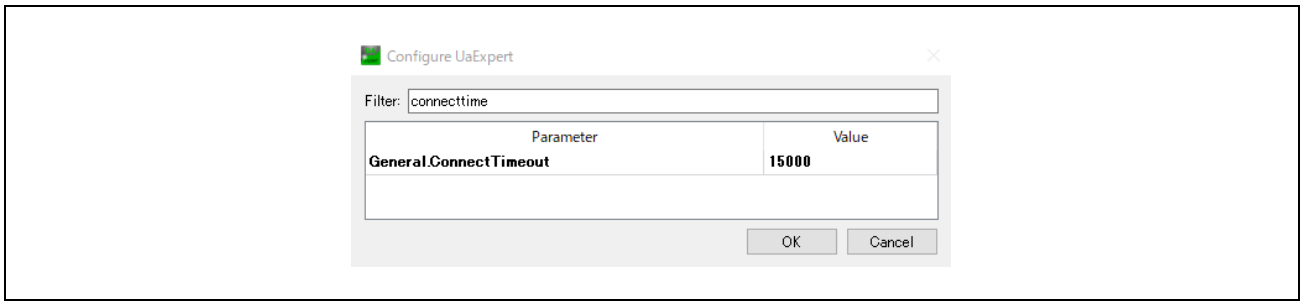


Figure 7.17. UaExpert ConnectTimeout Setting

Operation check after connection is the same as “chapter 7.2.1.1 Without security function”.

7.2.2 Publisher

This is the operation check for the given configuration.

Layout: Dynamic Layout

Security Policy: PubSub-Aes128-CTR

The role of each device when checking OPC UA Publisher operation is shown in Figure 7.18.

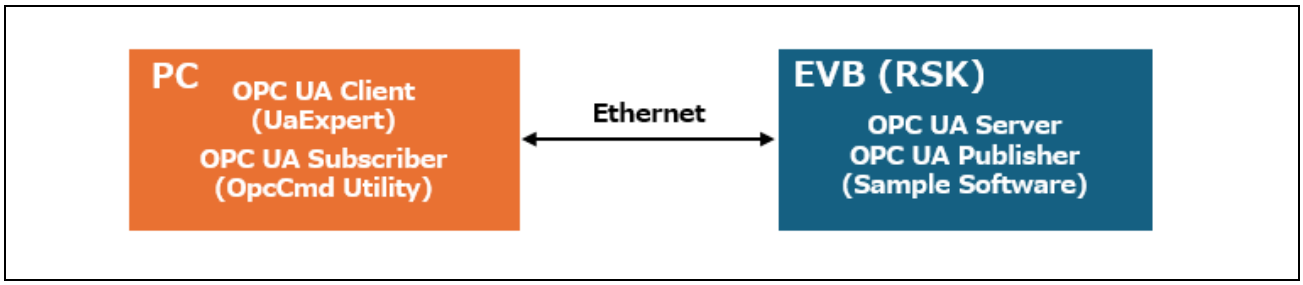


Figure 7.18. Connection Block Diagram for OPC UA Publisher

After powering on the evaluation board, launch Command prompt and change directory to the OpcCmd Utility folder. Enter the following command. This command is used to subscribe to messages encrypted with AES128 and published to “opc.udp://239.0.0.1” and decrypt it using the specified key. The key specified by “key” in this command represents the SigningKey in the upper 32 bytes, the EncryptingKey in the next 16 bytes, and the KeyNonce in the next 4 bytes.

OpcCmd uaSubscriber subscribeDataSet opc.udp://239.0.0.1:4840 --SecurityKeyServiceUri static:?key=000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223 2425262728292A2B2C2D2E2F30313233 --SecurityGroupId TestGroup --SecurityMode SecuritySignAndEncrypt -cni "(Network interface name)*"*

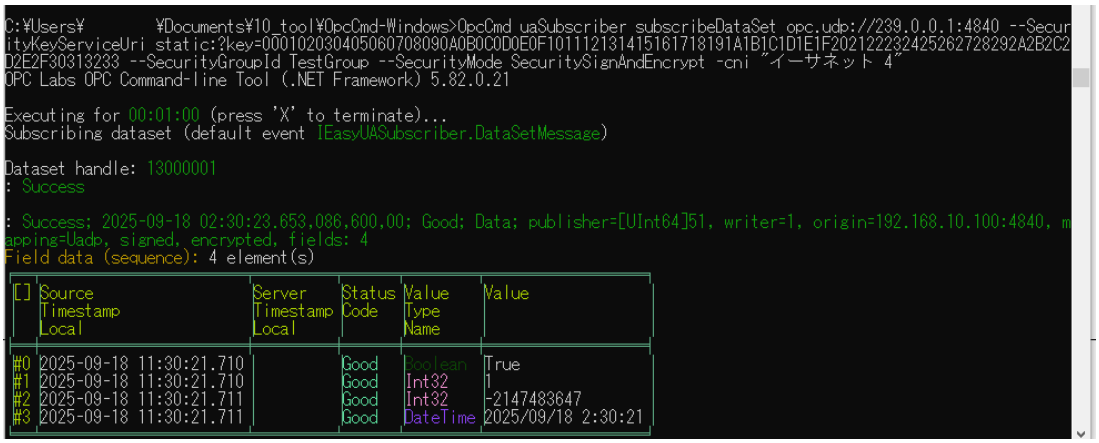


Figure 7.19. Command Execution Result

7.2.3 Subscriber

This is the operation check for the given configuration.

Layout: Dynamic Layout

Security Policy: PubSub-Aes128-CTR

The role of each device when checking OPC UA Subscriber operation is shown in Figure 7.20.

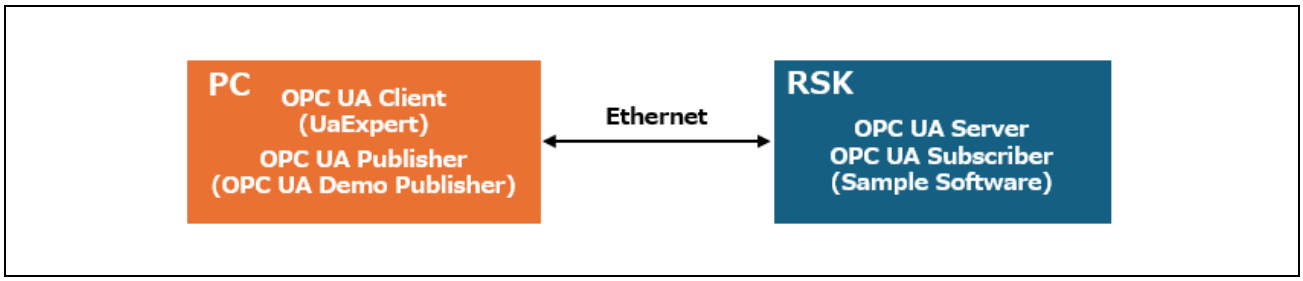


Figure 7.20. Connection Block Diagram for OPC UA Subscriber

After powering on the evaluation board, launch Command prompt and change directory to the UADemoPublisher folder. Enter the following command. This command is used to publish messages encrypted with AES128 using the specified key to “opc.udp://239.0.0.1”. The key specified by “key” in this command represents the SigningKey in the upper 32 bytes, the EncryptingKey in the next 16 bytes, and the KeyNonce in the next 4 bytes.

```
UADemoPublisher publish -cru opc.udp://239.0.0.1 -ewg DynamicLayoutGroup -edsw SimpleWriter --SecurityKeyServiceUri static:?key=000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F30313233 --SecurityGroupId TestGroup --SecurityMode SignAndEncrypt -cni "(Network interface name)"
```

Figure 7.21. Command Execution Result

After that, use UaExpert to verify that the “Subscribed_Variables_Simple” node exists.

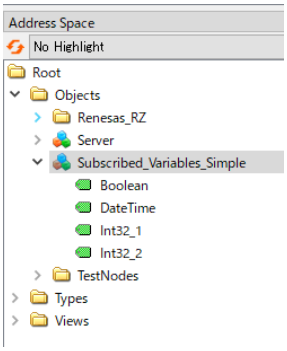


Figure 7.22. “Subscribed_Variables_Simple” Node

Checking these Nodes in the Data Access View, we can confirm that the values are updated periodically and that they are successfully subscribed.

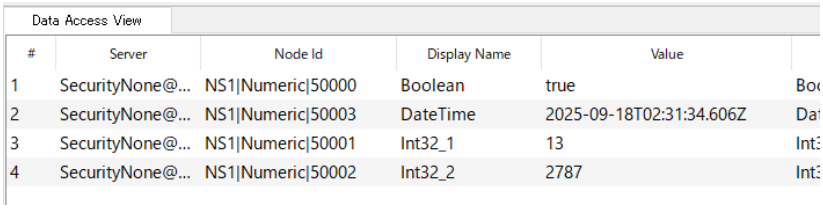


Figure 7.23. Check the Node Values

8. Software Specifications

8.1 Software Structure

8.1.1 Overview of Software Components

Figure 8.1 shows the architecture diagram of this sample software. This sample software uses FreeRTOS as its operating system.

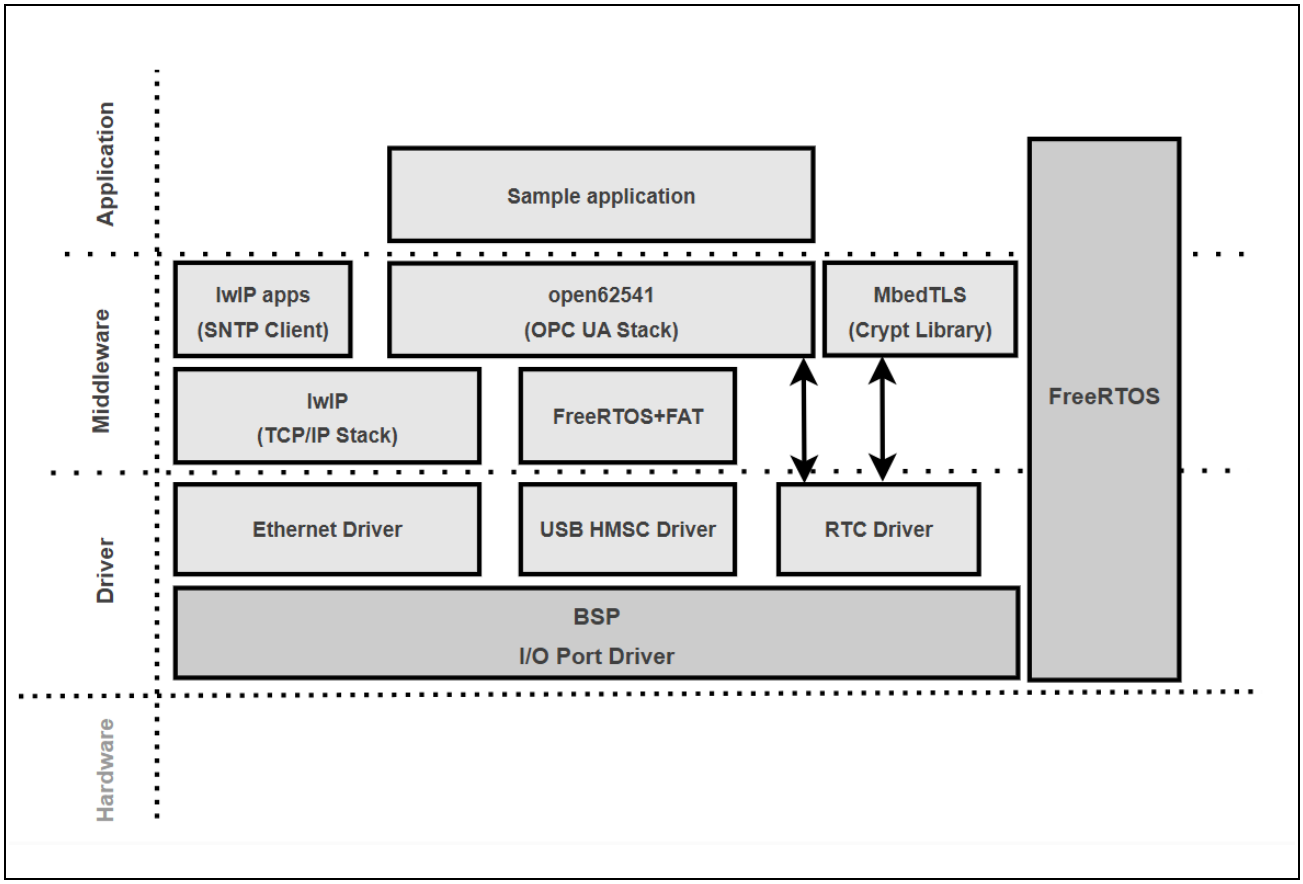


Figure 8.1. Software Architecture Diagram

8.1.2 Overview of Each Component

8.1.2.1 Driver

The drivers used in this software are generated by FSP. The FSP Stacks being used are shown in Table 8.1. Please check the Smart Configurator for the detailed settings of each.

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)

This sample software uses lwIP as TCP/IP stack. The base version used incorporates changes up to commit revision “79cd89f99d1032cc5375569e5b24c375b9d230fa” in STABLE-2_1_0. Table 8.2 shows the options changed from the default settings.

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)

This sample software uses Mbed TLS as cryptographic library. The version used is v3.6.6. For more information, please see Releases · Mbed-TLS/mbedtls.

8.1.2.4 open62541 (OPC UA Stack)

This sample software uses the open source open62541 as the protocol stack for the OPC UA Server. For more information on open62541, refer to the following link.

open62541

・Version

The base version of open62541 in this sample software is the following.

Base Version : v1.3.18

(Commit db393f2c94374ee55d89ef46e5163686932222ee)

・License

The license terms for the open62541 are MPL v2.0.

Please refer to https://www.mozilla.org/en-US/MPL/2.0/ for more information and comply with the license terms and conditions.

・open62541 files

To run open62541 in a FreeRTOS + lwIP environment, the CMake-based build flow introduced at the link below is used to configure open62541 and compile the information model NodeSet files. For more information, please refer to chapter 9.1 in the Appendix.

open62541 Documentation — open62541 1.3.0-dirty documentation

・Information models

To achieve interoperability among vendors and industries, OPC UA provides a unified data model called the “Information Model” in xml file format. It includes built-in models commonly used in OPC UA, companion models used by each industry or organization, and vendor-specific models that can be customized by each vendor. (Figure 8.2)

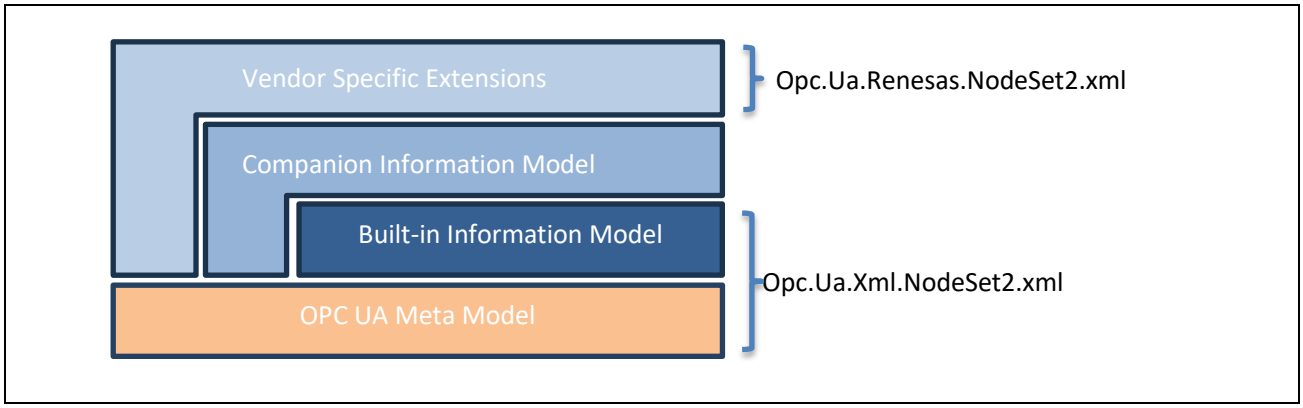


Figure 8.2. Information Model

・Opc.Ua.Xml.NodeSet2.xml

It provides the "Meta Model", which is a set of rules for describing the OPC UA information model, and the "Built-in Information Model", which is the basic information model of OPC UA described by the Meta Model.

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

・Opc.Ua.Renesas.NodeSet2.xml

This is the information model made for this sample software. The OPC UA modeler "SiOME" is used for creation. For details, please refer to the following link.

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

Renesas information model is structured as shown in Figure 8.3. Each Nodes are written and read as described in section 8.1.2.5.

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

This project operates as a mock sensor and periodically writes to and reads from the OPC UA Nodes. The output value of the mock sensor is controlled according to its state, and the lighting status of the LEDs on the evaluation board also changes.

After the evaluation board is booted and time is synchronized by NTP, it transitions from State0 to State1, and the mock sensor processing begins. The acquired value of the mock sensor changes sinusoidally in State1, and in State2, the value is a sine wave with random noise and is written to the “RawValue” Node. The state transition between State1 and State2 is made by pressing the switch on the evaluation board.

When “0: Disable” is written to the “OperationEnable” Node from the OPC UA Client, it moves to State3 and the processing of the mock sensor stops. Writing “1: Enable” again returns to the previous state and restarts the processing of the mock sensor.

The LED lighting status for each state is shown below, and the respective values are written to the “SignalColor” Node and “SignalModeLight” Node.

State0 : Red LED Lighting、State1 : Green LED Lighting
State2 : Green LED Blinking、State3 : Red LED Blinking

These state transitions are shown in Figure 8.4.

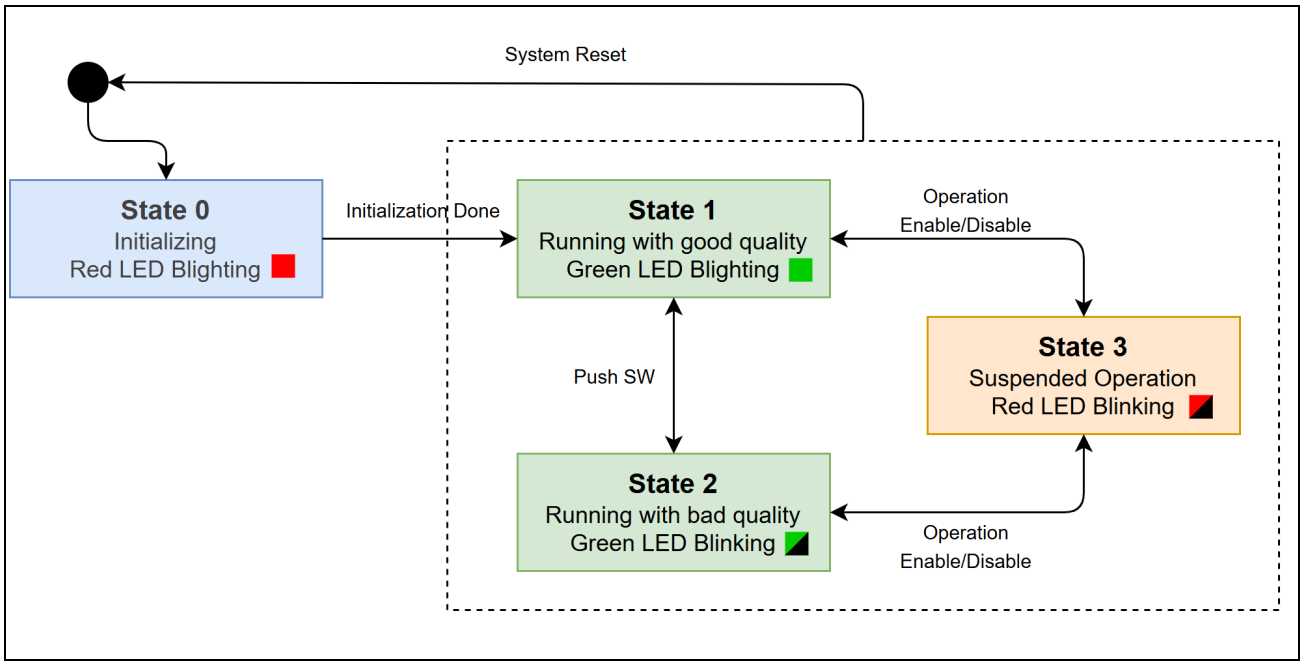


Figure 8.4. State Transition

In addition, when “1:Execute SystemReset” is written to the “SystemReset” node, the “Status” node (i=2259) changes to 4:Shutdown, and a 15-second countdown begins at the “SecondsTillShutdown” node (i=2992). After that, when the value reaches 0, a system reset is performed.

The above switch and LEDs are shown in Table 8.3.

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

This chapter explains the RTOS tasks in the sample software.

8.2.1 Task List

Table 8.4 lists the tasks used in this sample software.

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

This chapter explains the processing of each task.

8.2.2.1 Startup

Figure 8.5 shows the sequence diagram for the startup process. The “Main Thread” task is the first task created during startup, primarily responsible for module initialization and creating other tasks.

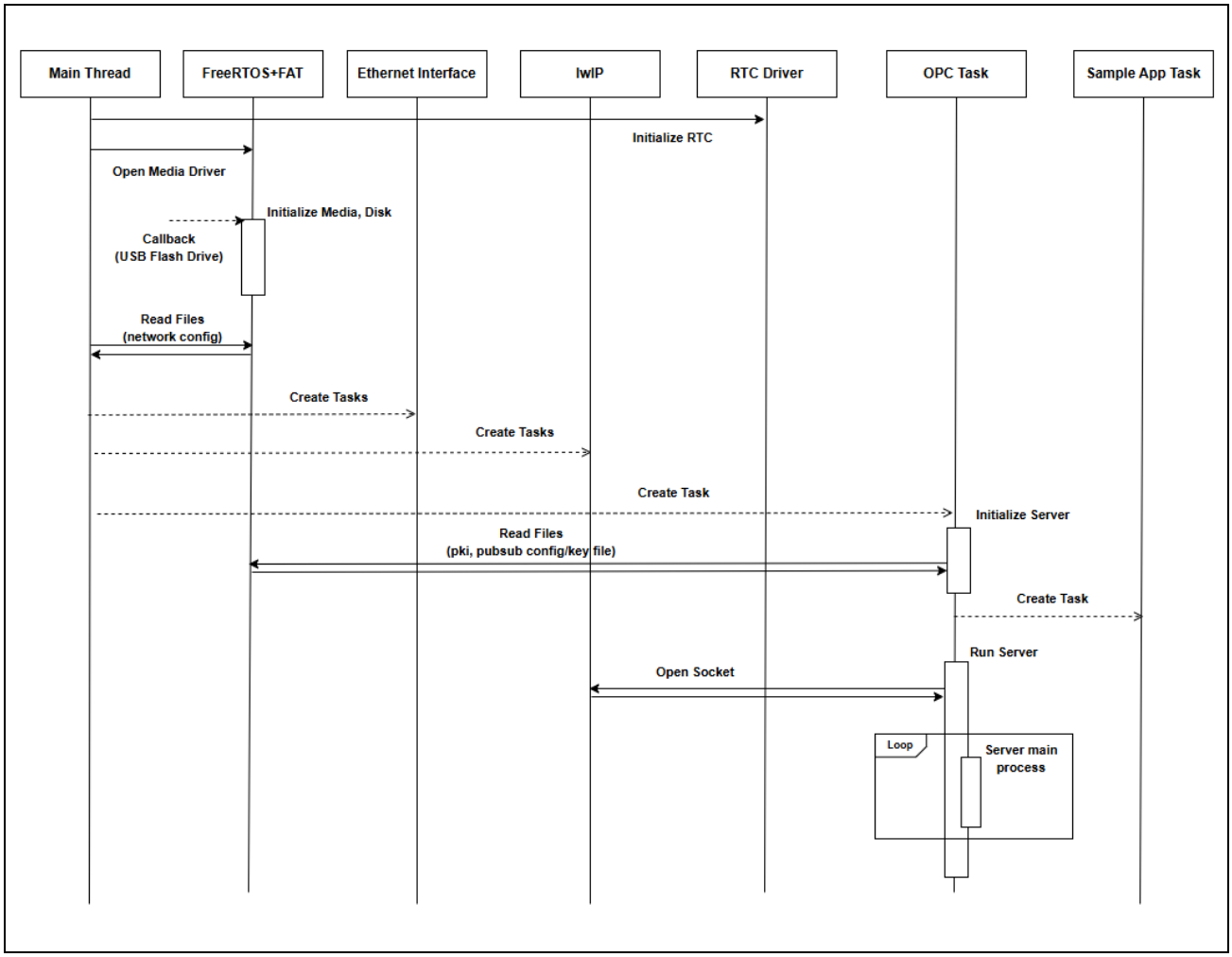


Figure 8.5. Startup

8.2.2.2 OPC UA communication

Figure 8.6 shows the sequence diagram of OPC UA transmission and reception processing. OPC UA data transmitted and received via lwIP is encrypted and decrypted using Mbed TLS.

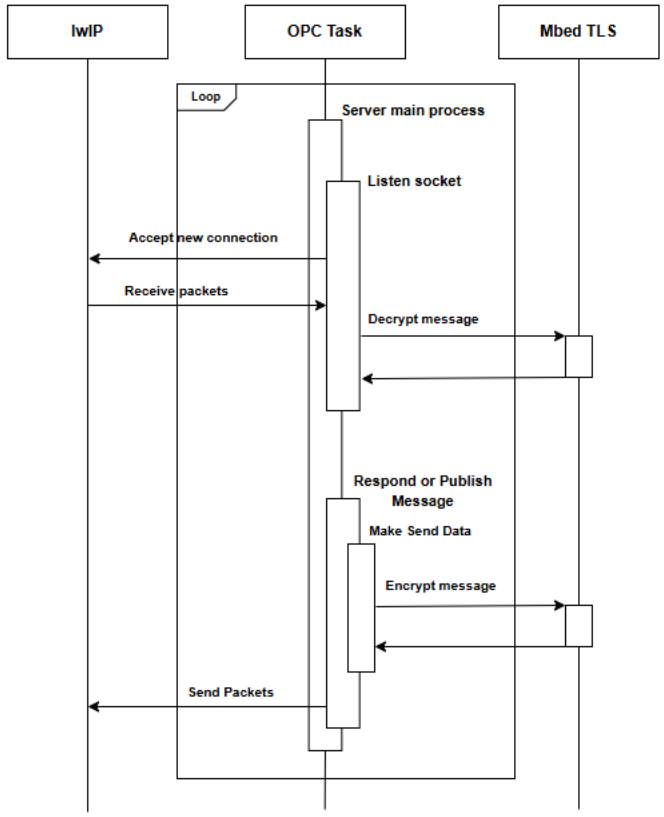


Figure 8.6. OPC UA communication

8.2.2.3 Sample application

Figure 8.7 shows the sequence diagram for the sample application. The sample application controls node values and LED outputs according to the specifications in chapter 8.1.2.5.

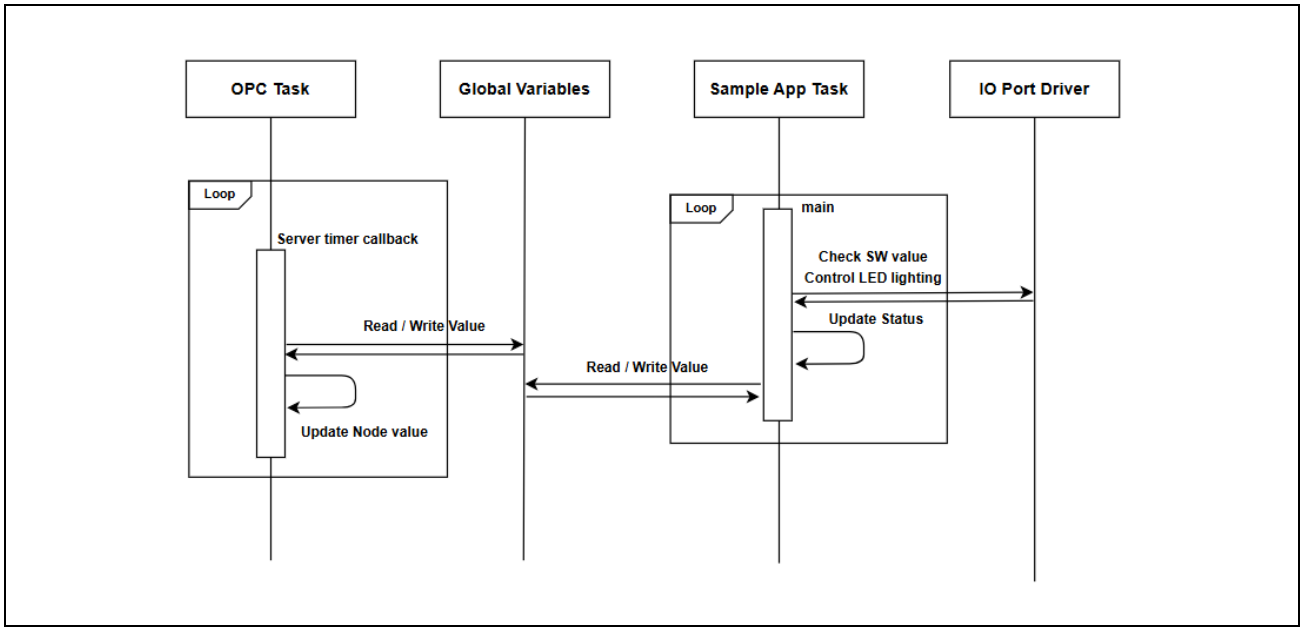


Figure 8.7. Sample application

8.2.2.4 Time synchronization

Figure 8.8 shows the sequence diagram for time synchronization processing. The time information received from the NTP server via SNTP communication is managed by the RTC, a function of the RZ device, and is used by various functions.

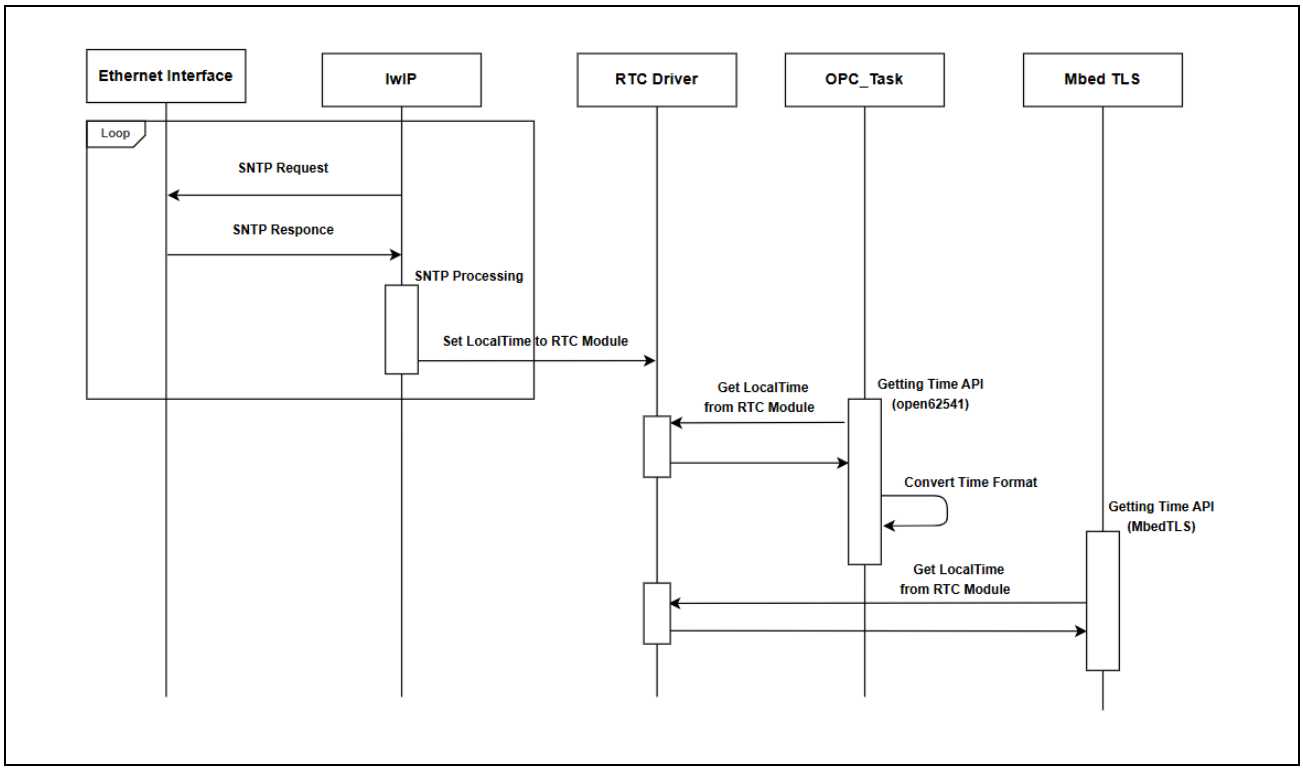


Figure 8.8. Time synchronization

8.3 Footprint (ROM and RAM Usage)

This chapter describes the memory footprint in each project.

8.3.1 RZ/T2M project

8.3.1.1 e² studio

(1) Primary project

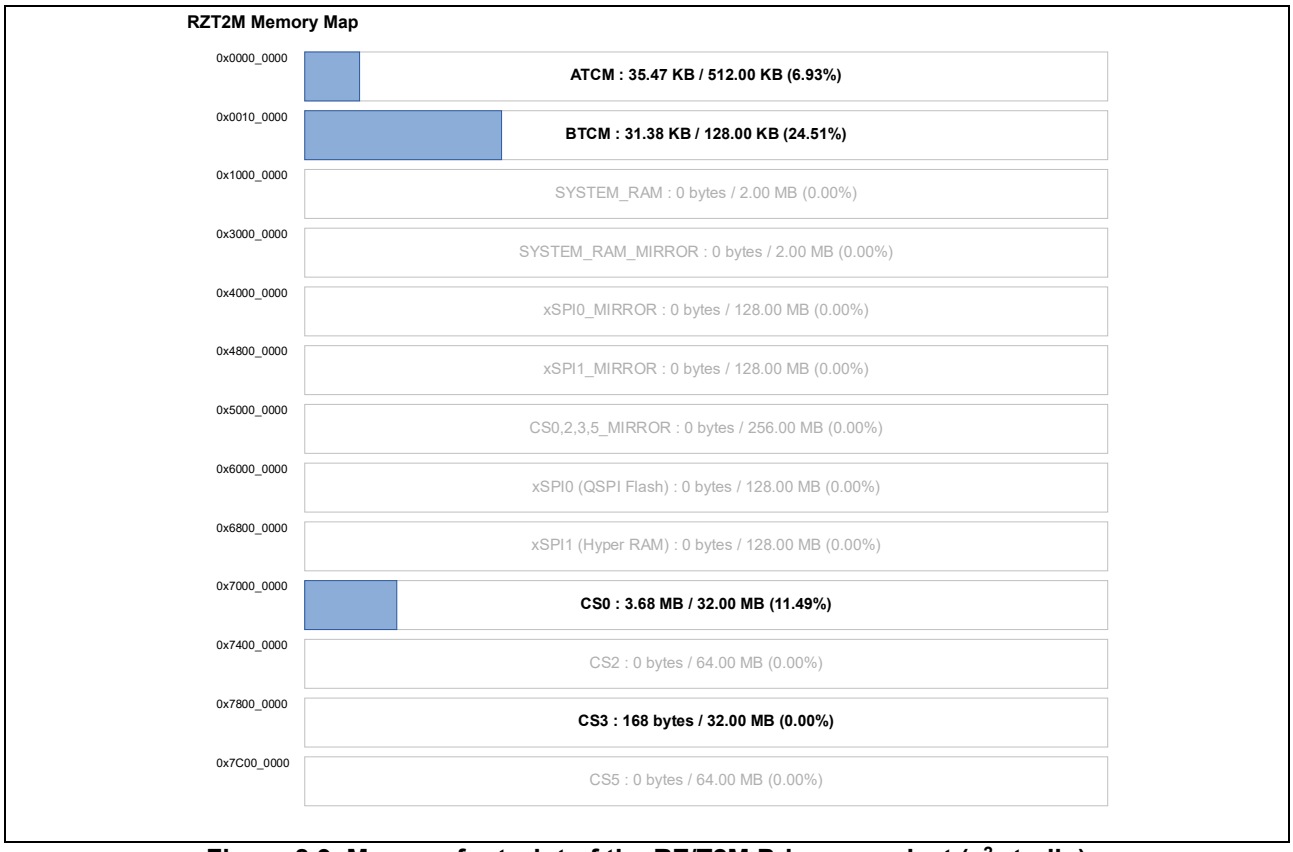
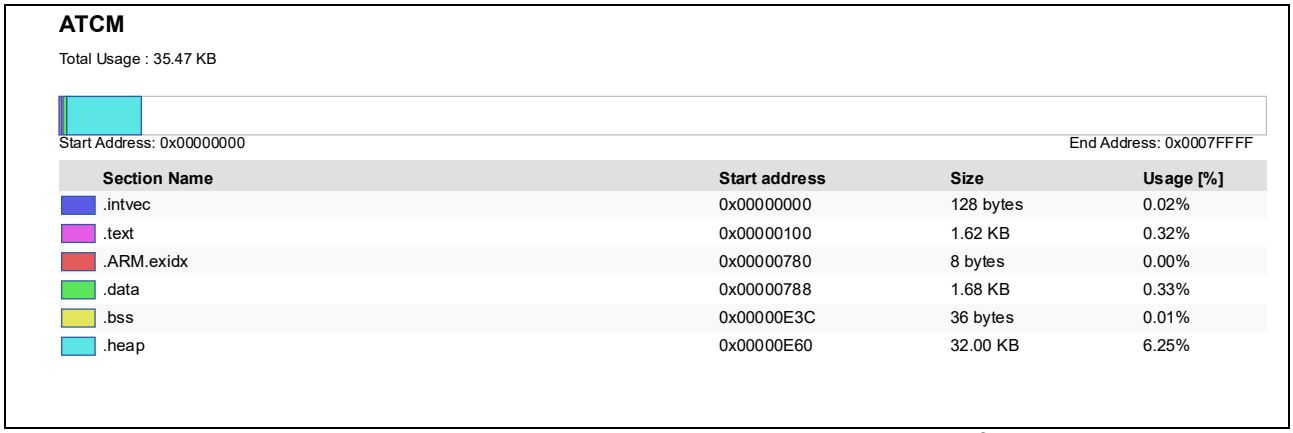


Figure 8.9. Memory footprint of the RZ/T2M Primary project (e² studio)



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)

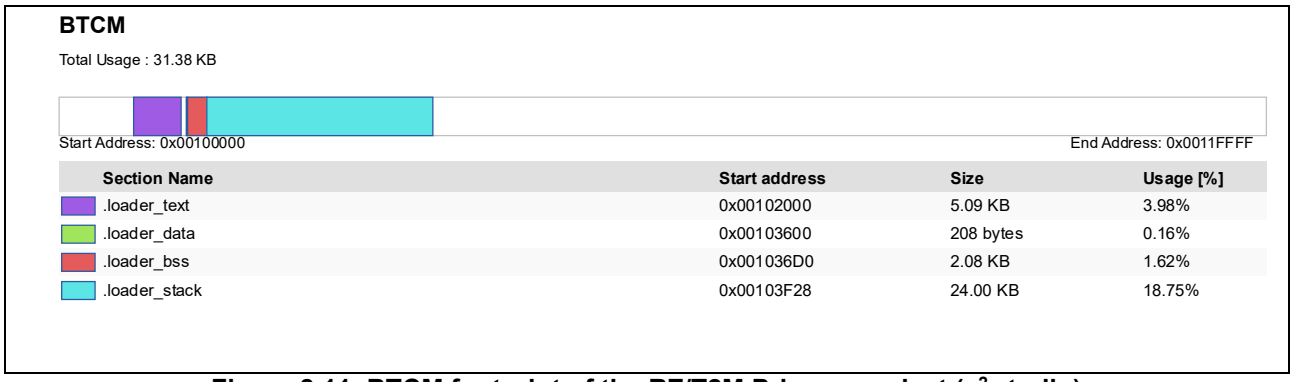


Figure 8.11. BTCM footprint of the RZ/T2M Primary project (e² studio)

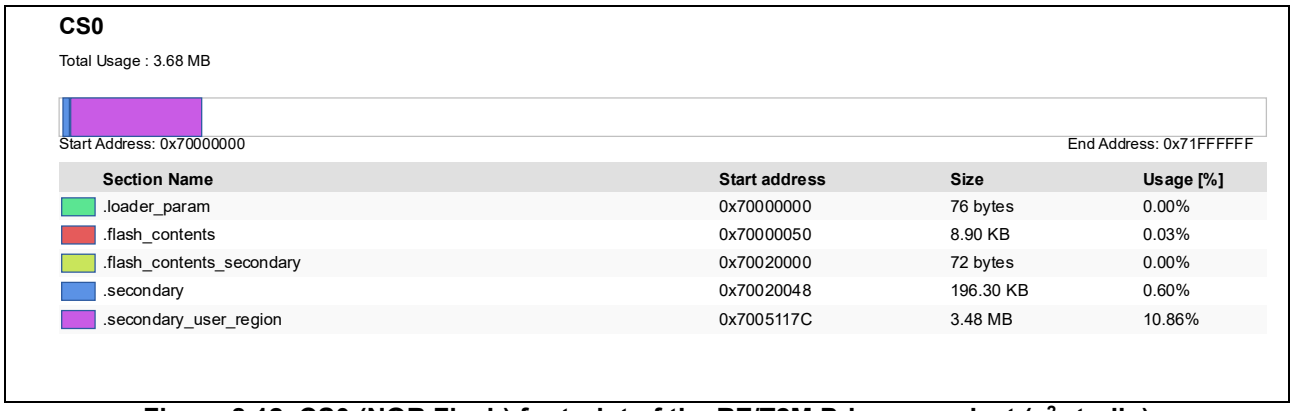


Figure 8.12. CS0 (NOR Flash) footprint of the RZ/T2M Primary project (e² studio)

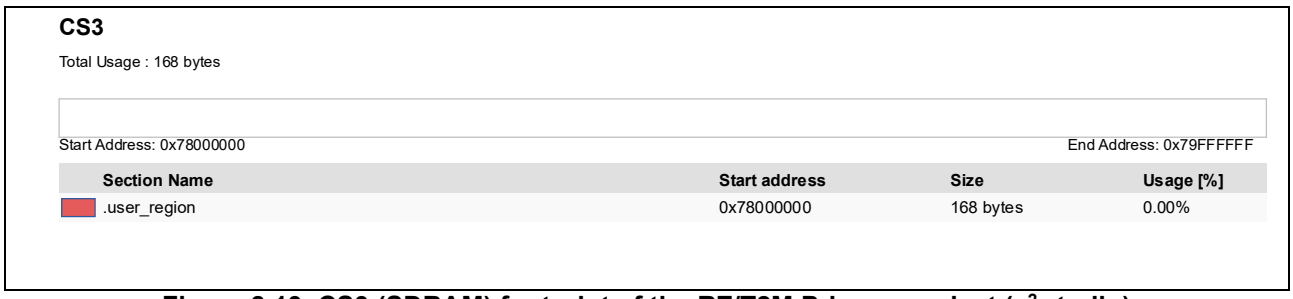


Figure 8.13. CS3 (SDRAM) footprint of the RZ/T2M Primary project (e² studio)

(2) Secondary project

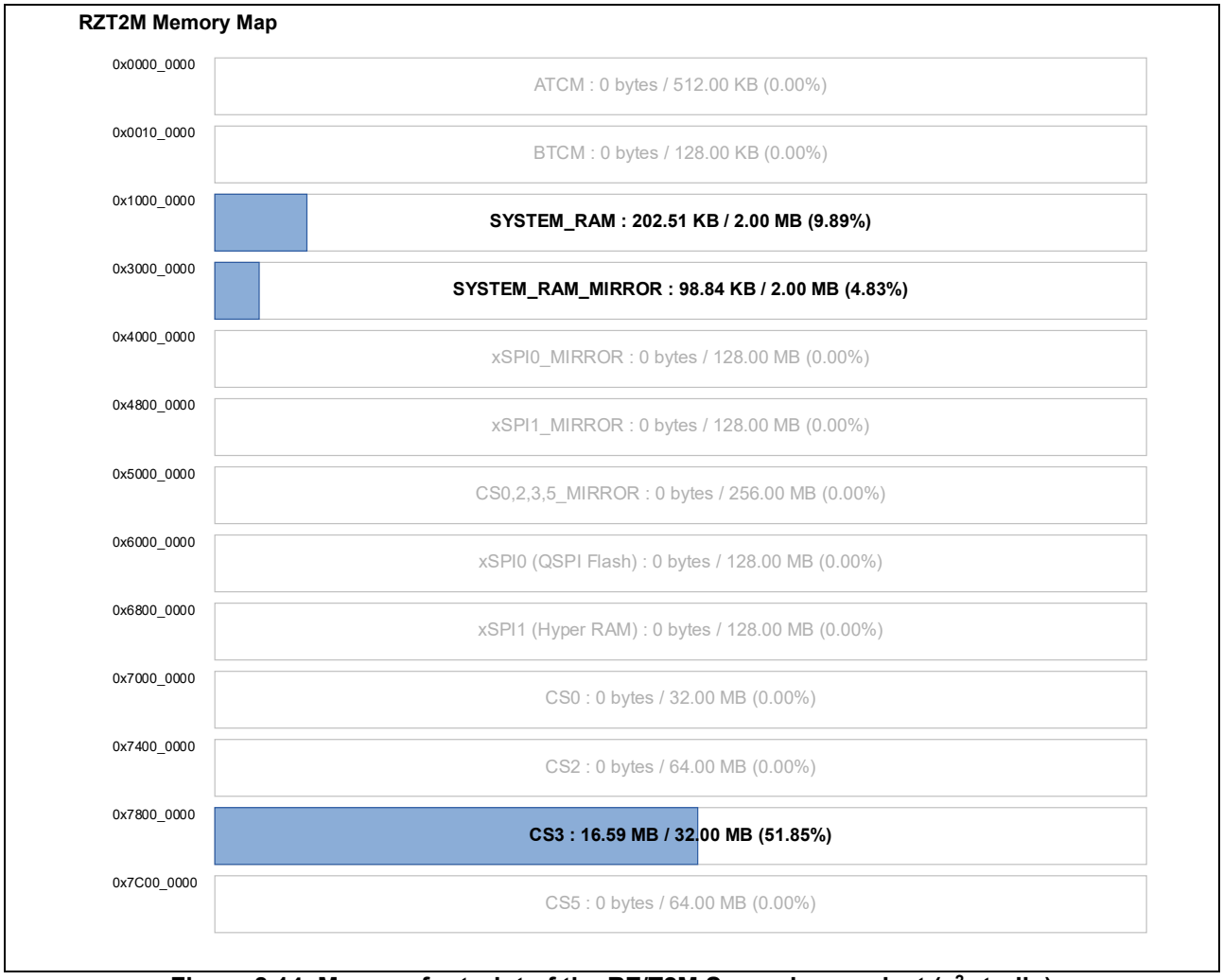
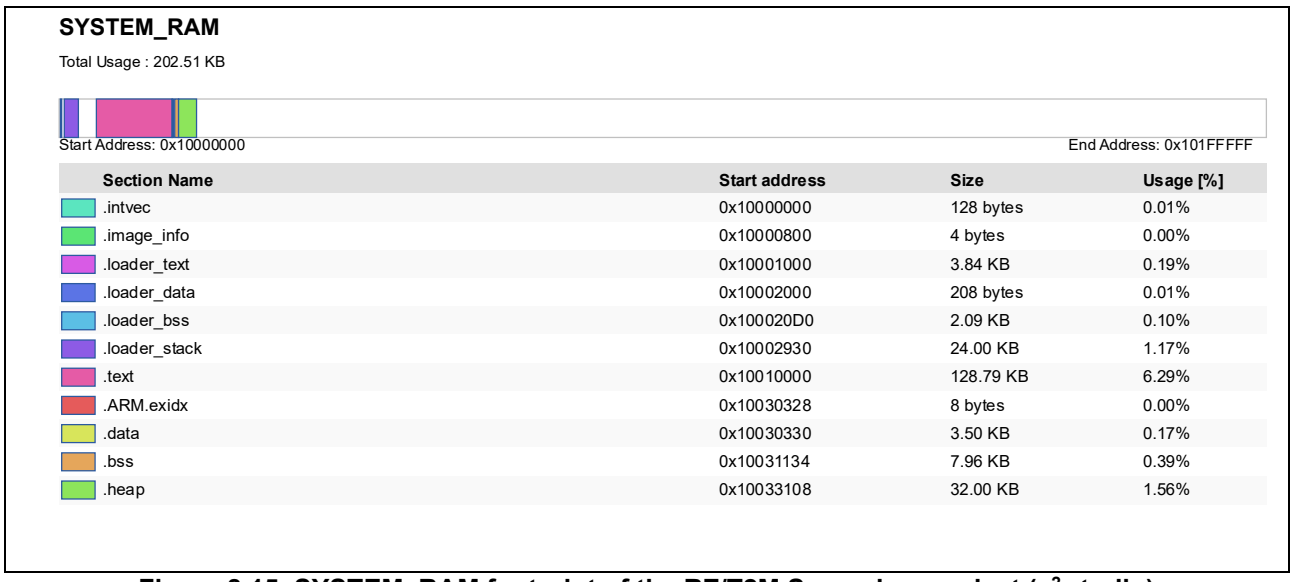


Figure 8.14. Memory footprint of the RZ/T2M Secondary project (e² studio)



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)

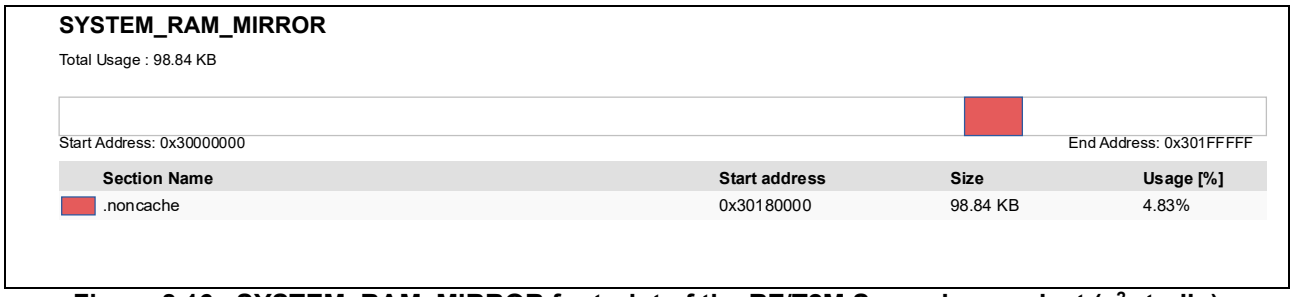


Figure 8.16. SYSTEM_RAM_MIRROR footprint of the RZ/T2M Secondary project (e² studio)

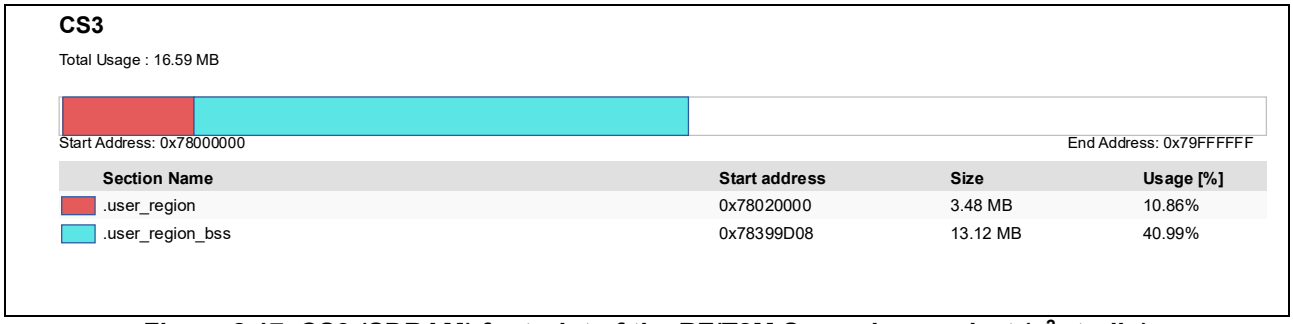


Figure 8.17. CS3 (SDRAM) footprint of the RZ/T2M Secondary project (e² studio)

8.3.1.2 EWARM

(1) Primary project

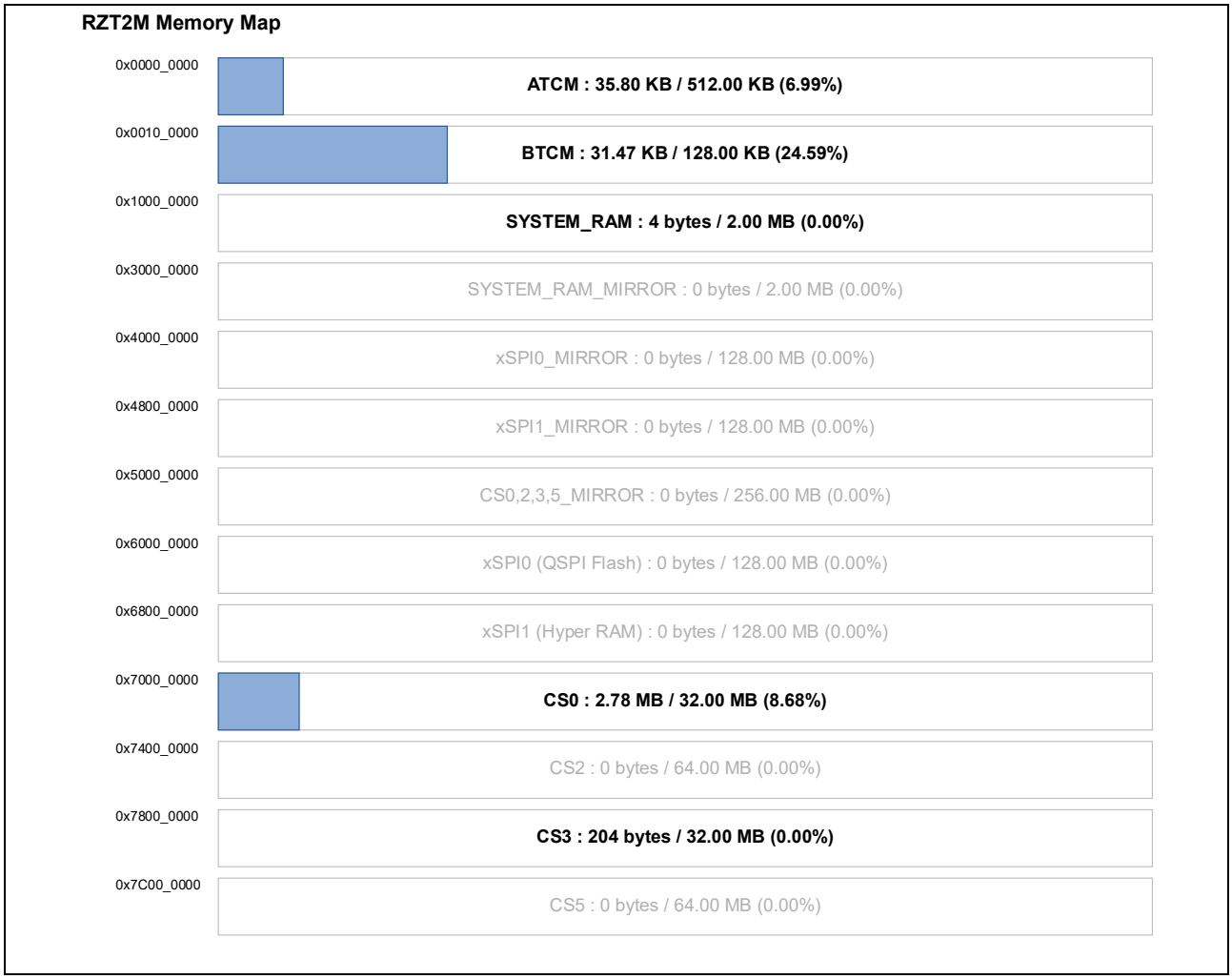
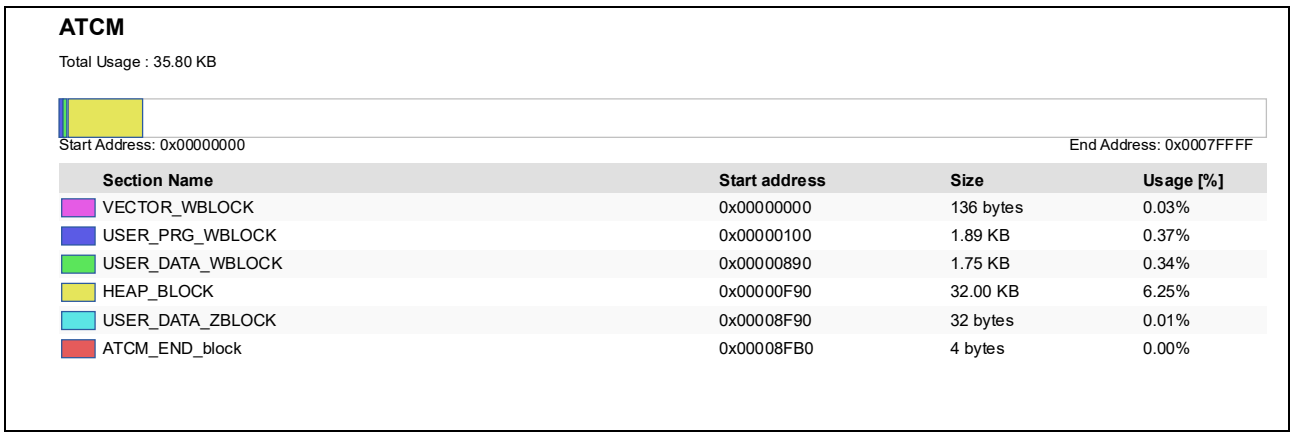
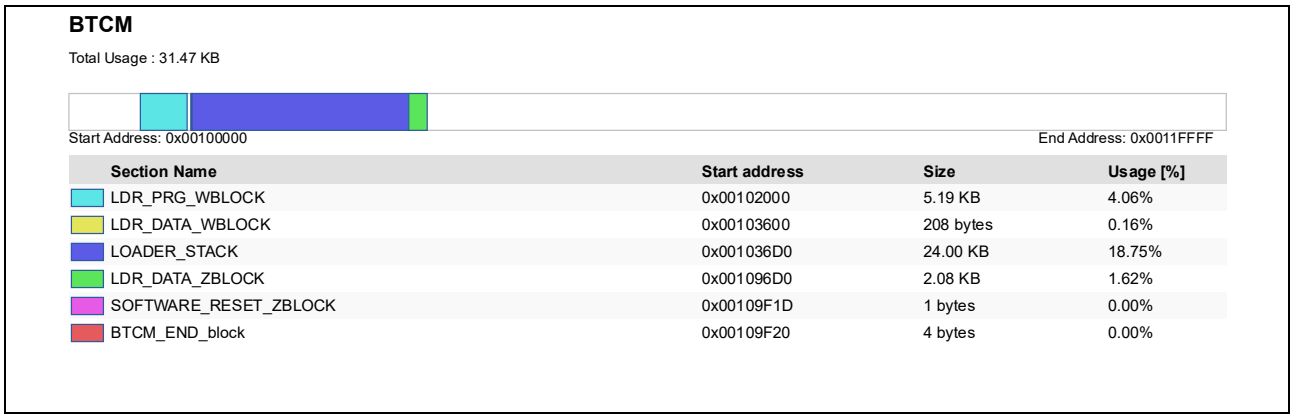


Figure 8.18. Memory footprint of the RZ/T2M Primary project (EWARM)



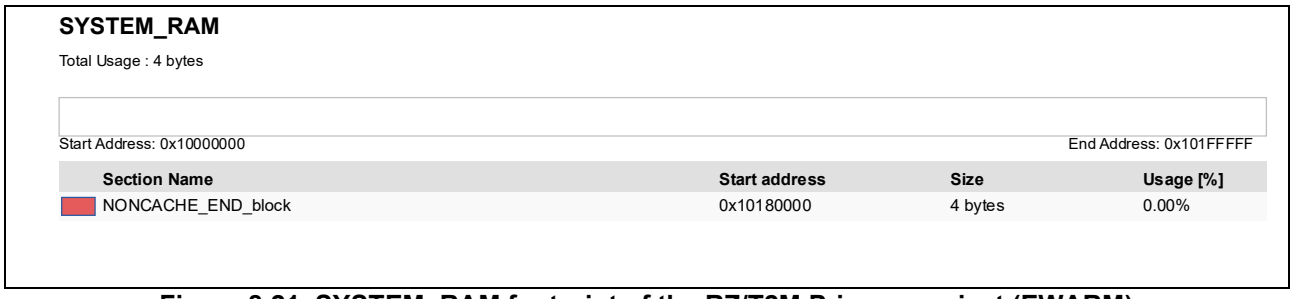
Section Name	Start address	Size	Usage [%]
VECTOR_WBLOCK	0x00000000	136 bytes	0.03%
USER_PRG_WBLOCK	0x00000100	1.89 KB	0.37%
USER_DATA_WBLOCK	0x00000890	1.75 KB	0.34%
HEAP_BLOCK	0x00000F90	32.00 KB	6.25%
USER_DATA_ZBLOCK	0x00008F90	32 bytes	0.01%
ATCM_END_block	0x00008FB0	4 bytes	0.00%

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



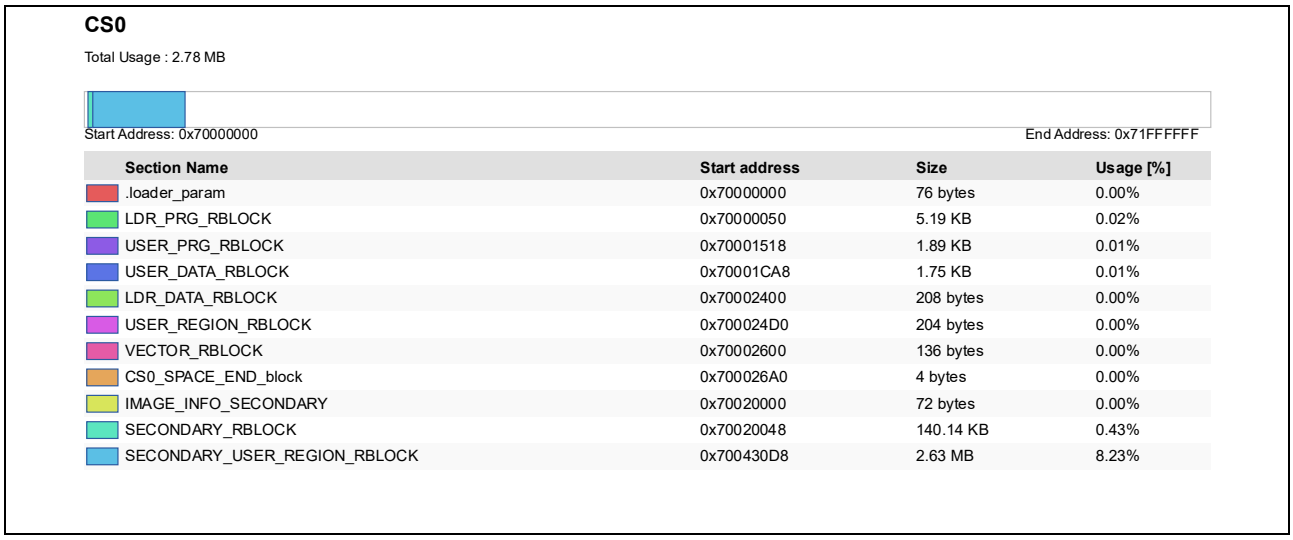
Section Name	Start address	Size	Usage [%]
LDR_PRG_WBLOCK	0x00102000	5.19 KB	4.06%
LDR_DATA_WBLOCK	0x00103600	208 bytes	0.16%
LOADER_STACK	0x001036D0	24.00 KB	18.75%
LDR_DATA_ZBLOCK	0x001096D0	2.08 KB	1.62%
SOFTWARE_RESET_ZBLOCK	0x00109F1D	1 bytes	0.00%
BTCM_END_block	0x00109F20	4 bytes	0.00%

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



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)



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)

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

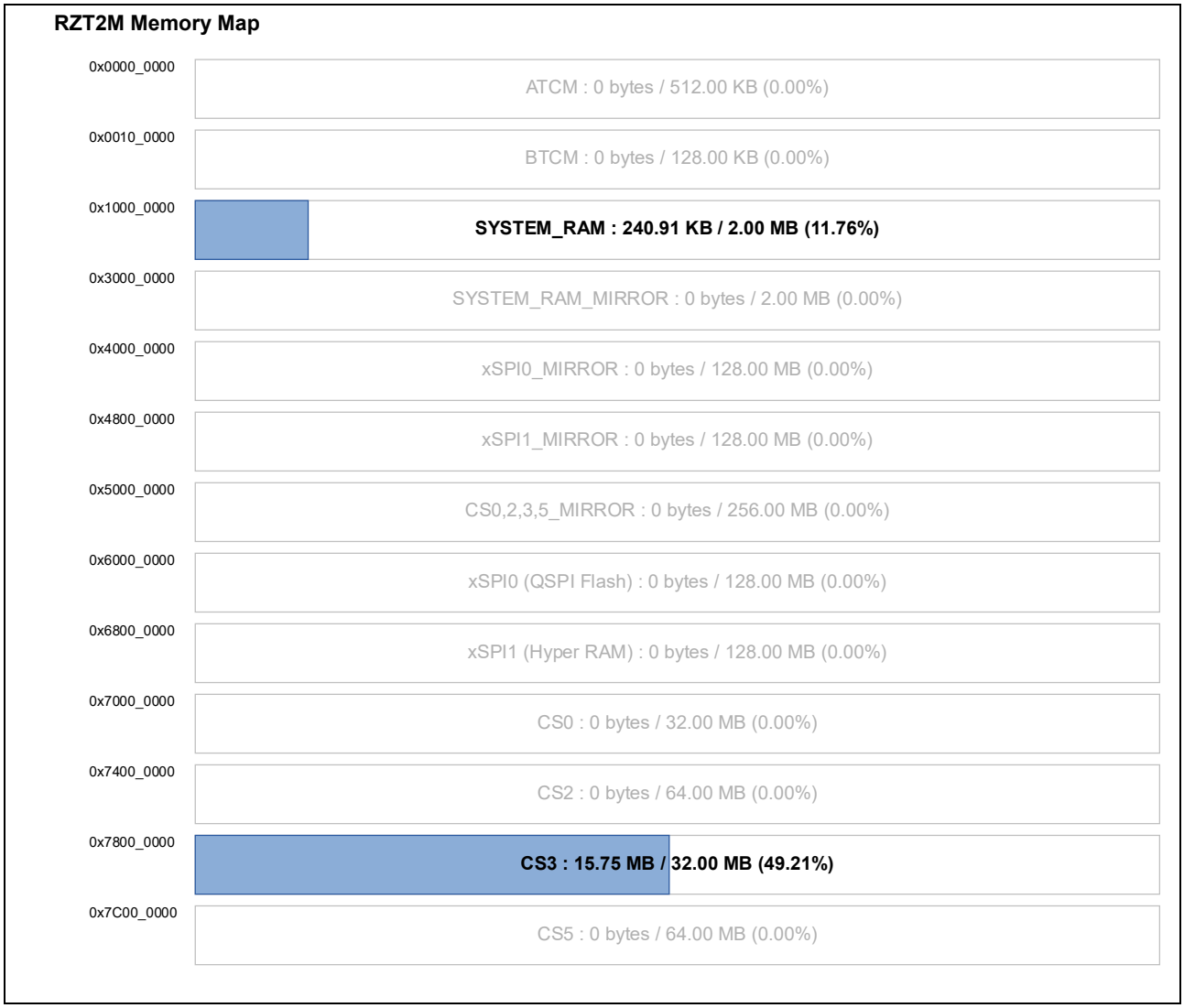
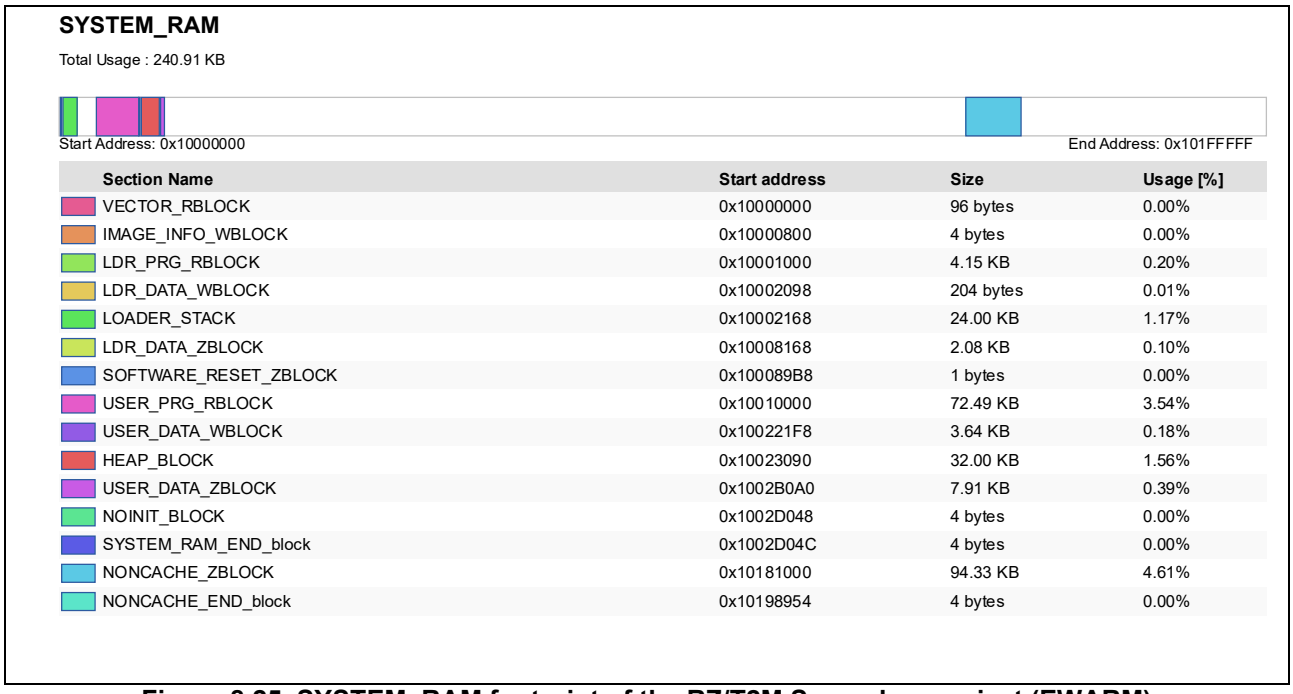


Figure 8.24. Memory footprint of the RZ/T2M Secondary project (EWARM)



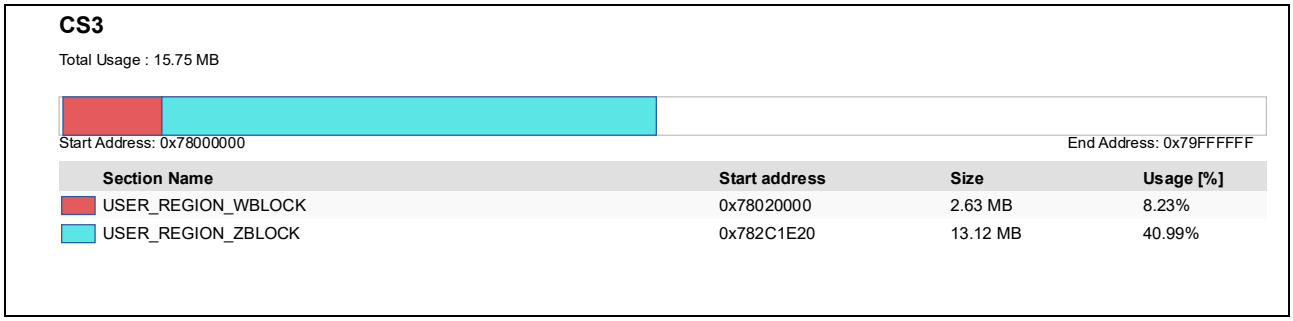
SYSTEM_RAM

Total Usage : 240.91 KB

Start Address: 0x10000000 End Address: 0x101FFFFF

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)



CS3

Total Usage : 15.75 MB

Start Address: 0x78000000 End Address: 0x79FFFFFF

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

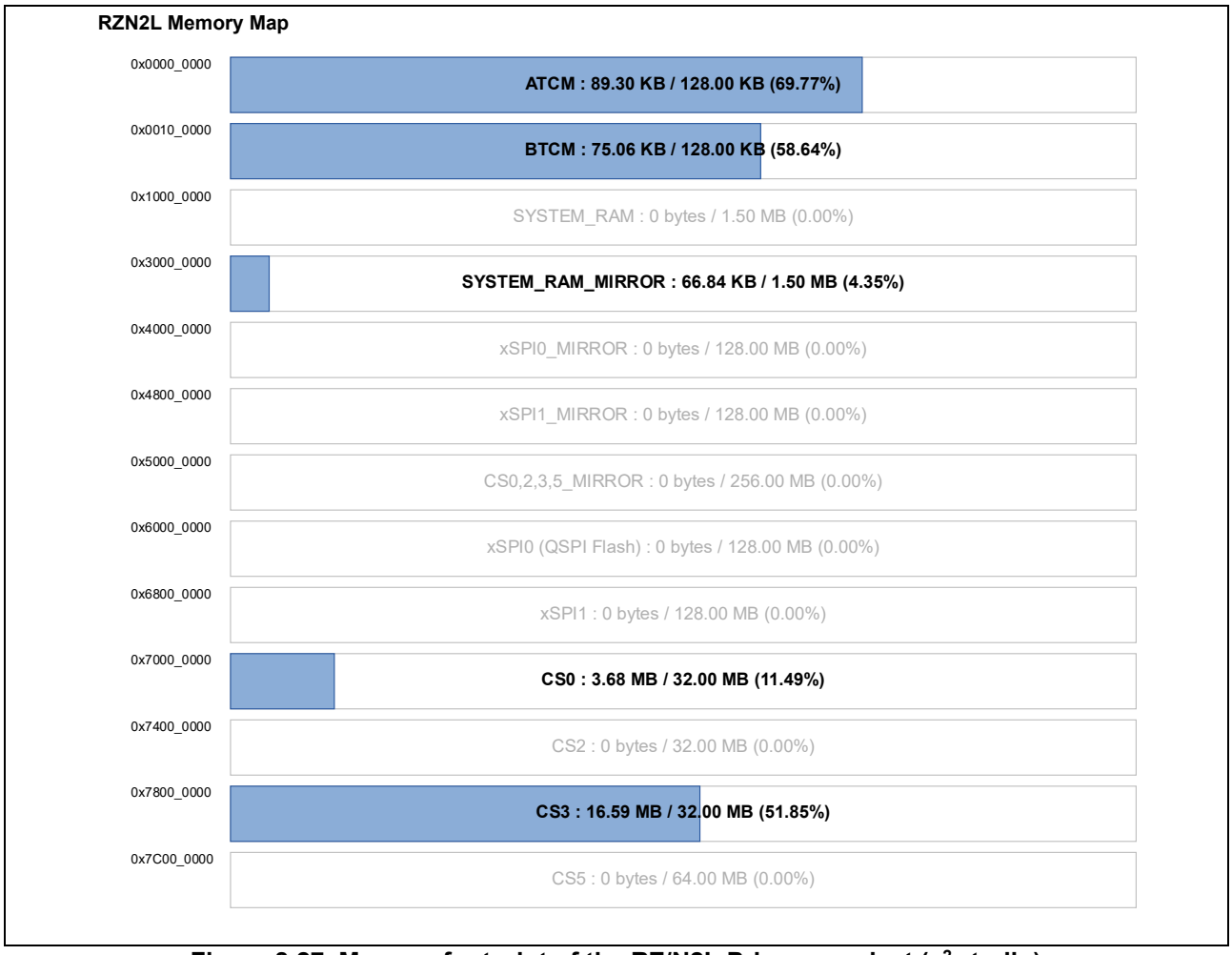
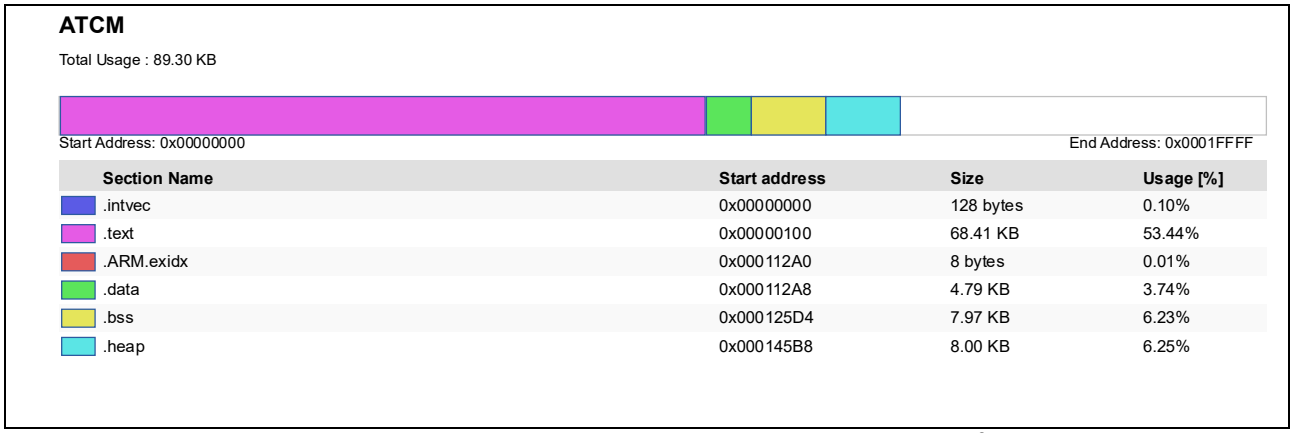


Figure 8.27. Memory footprint of the RZ/N2L Primary project (e² studio)



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)

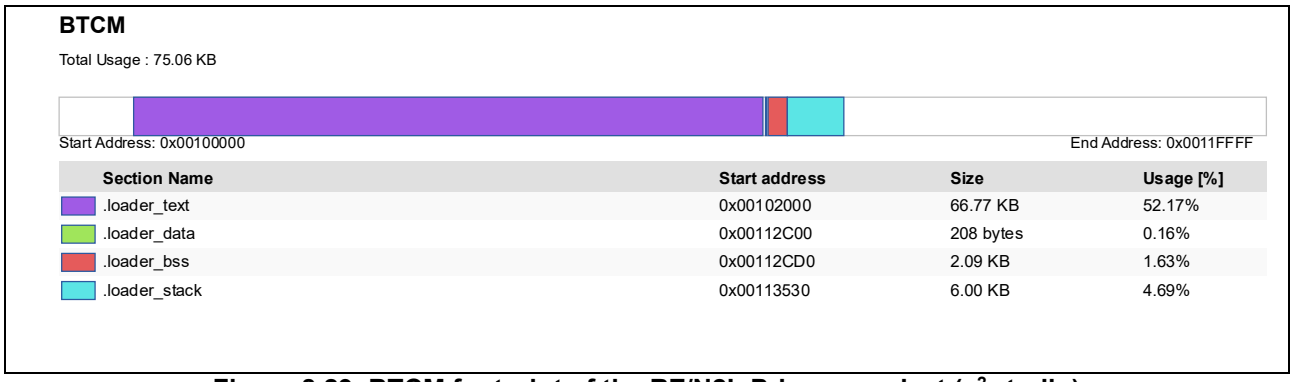


Figure 8.29. BTCM footprint of the RZ/N2L Primary project (e² studio)

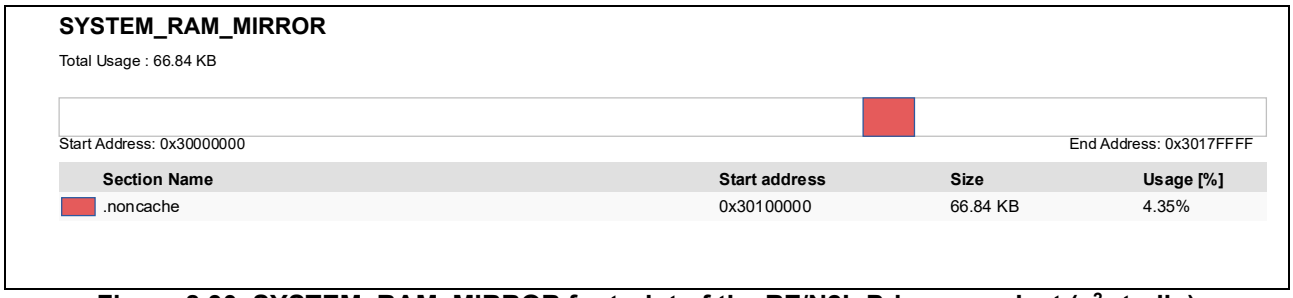


Figure 8.30. SYSTEM_RAM_MIRROR footprint of the RZ/N2L Primary project (e² studio)

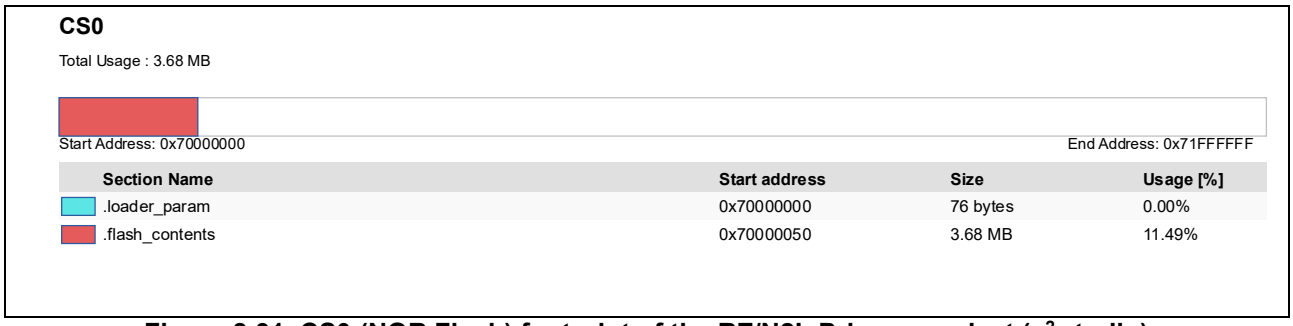


Figure 8.31. CS0 (NOR Flash) footprint of the RZ/N2L Primary project (e² studio)

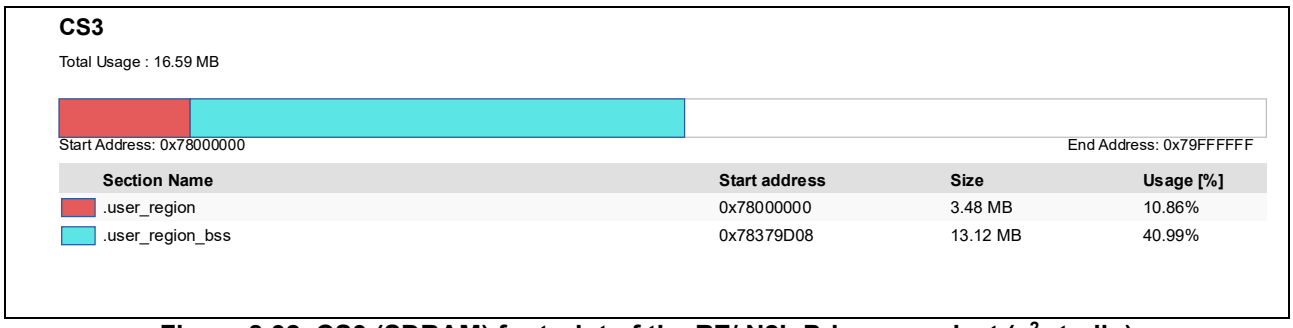


Figure 8.32. CS3 (SDRAM) footprint of the RZ/ N2L Primary project (e² studio)

8.3.2.2 EWARM

(1) Primary project

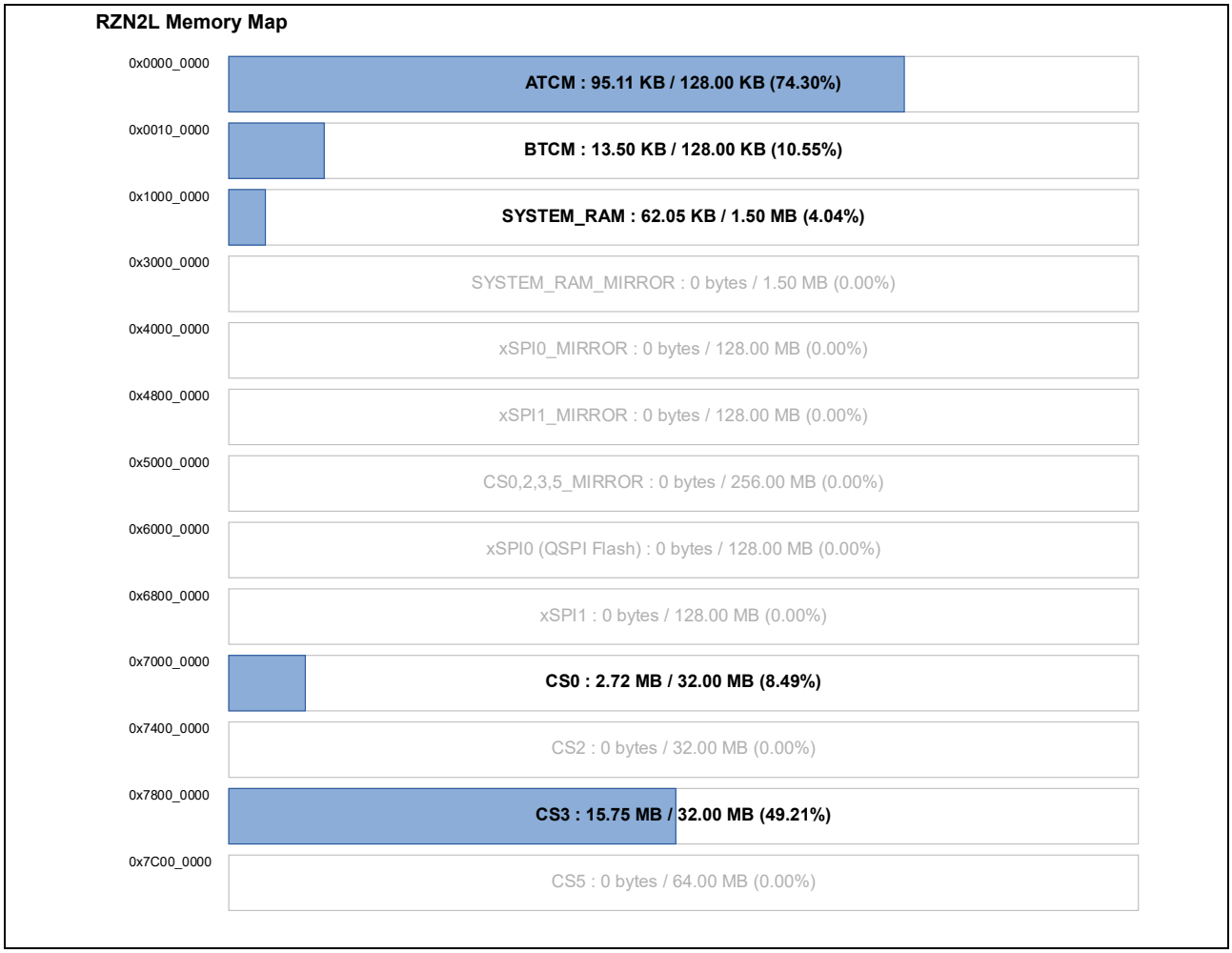
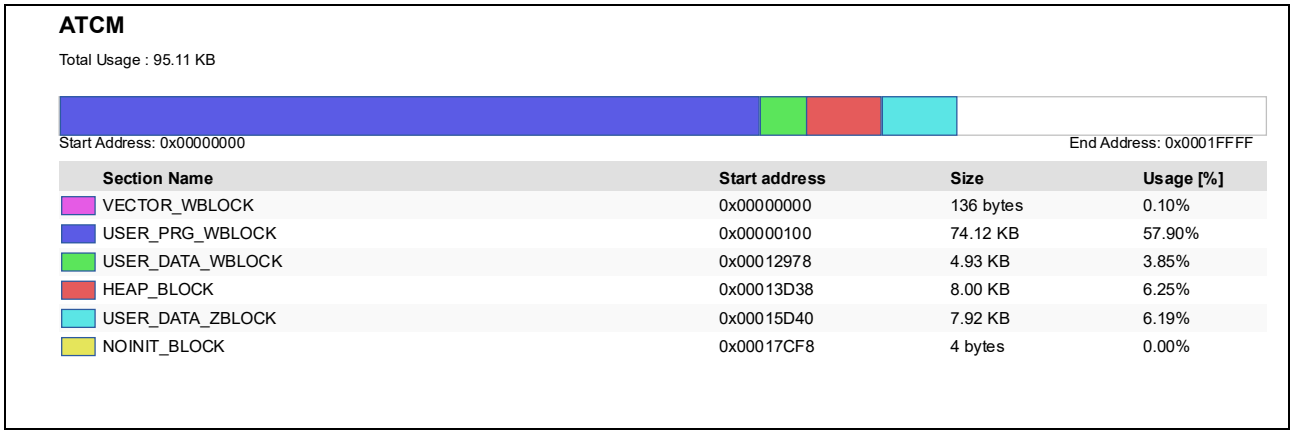
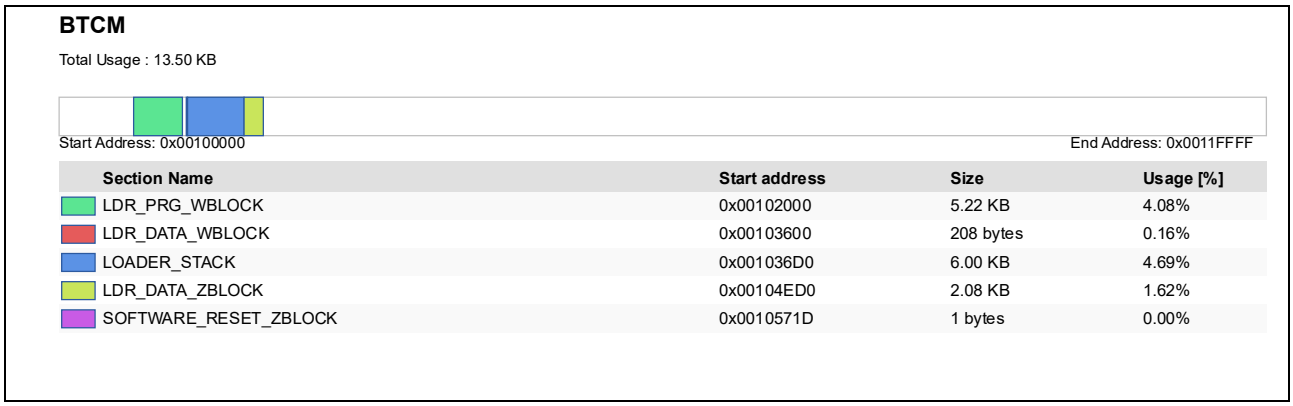


Figure 8.33. Memory footprint of the RZ/N2L Primary project (EWARM)



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)

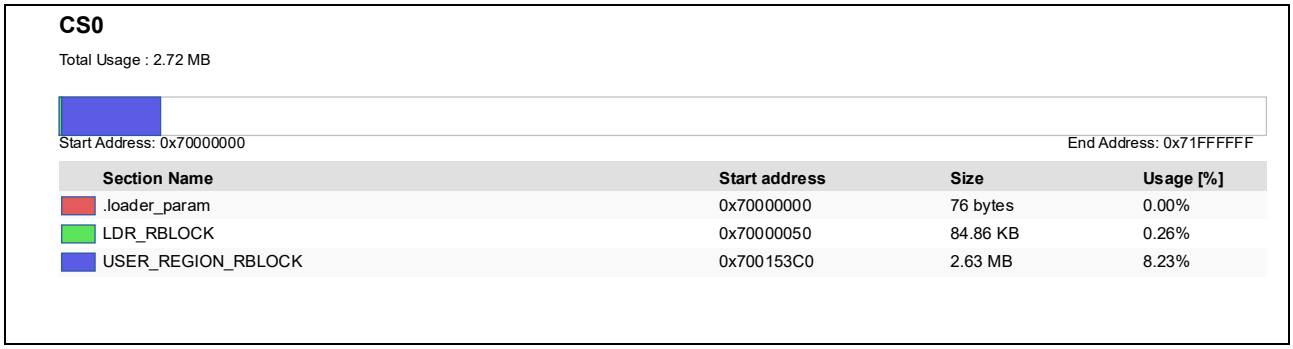


Section Name	Start address	Size	Usage [%]
LDR_PRG_WBLOCK	0x00102000	5.22 KB	4.08%
LDR_DATA_WBLOCK	0x00103600	208 bytes	0.16%
LOADER_STACK	0x001036D0	6.00 KB	4.69%
LDR_DATA_ZBLOCK	0x00104ED0	2.08 KB	1.62%
SOFTWARE_RESET_ZBLOCK	0x0010571D	1 bytes	0.00%

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

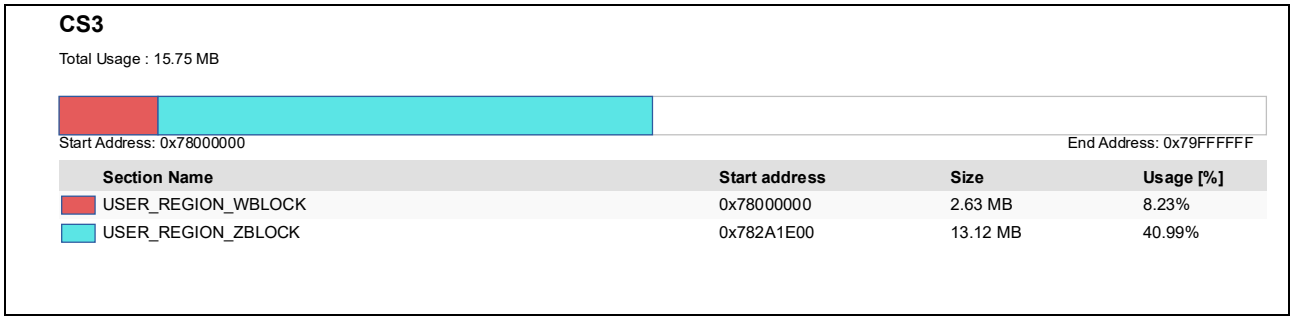
Section Name	Start address	Size	Usage [%]
NONCACHE_ZBLOCK	0x10100000	62.05 KB	4.04%

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)

* Since RZ/T2H and N2H are implemented with the same memory layout, the measurement results for RZ/T2H are presented here as representative.

8.3.3.1 e² studio

(1) Primary project

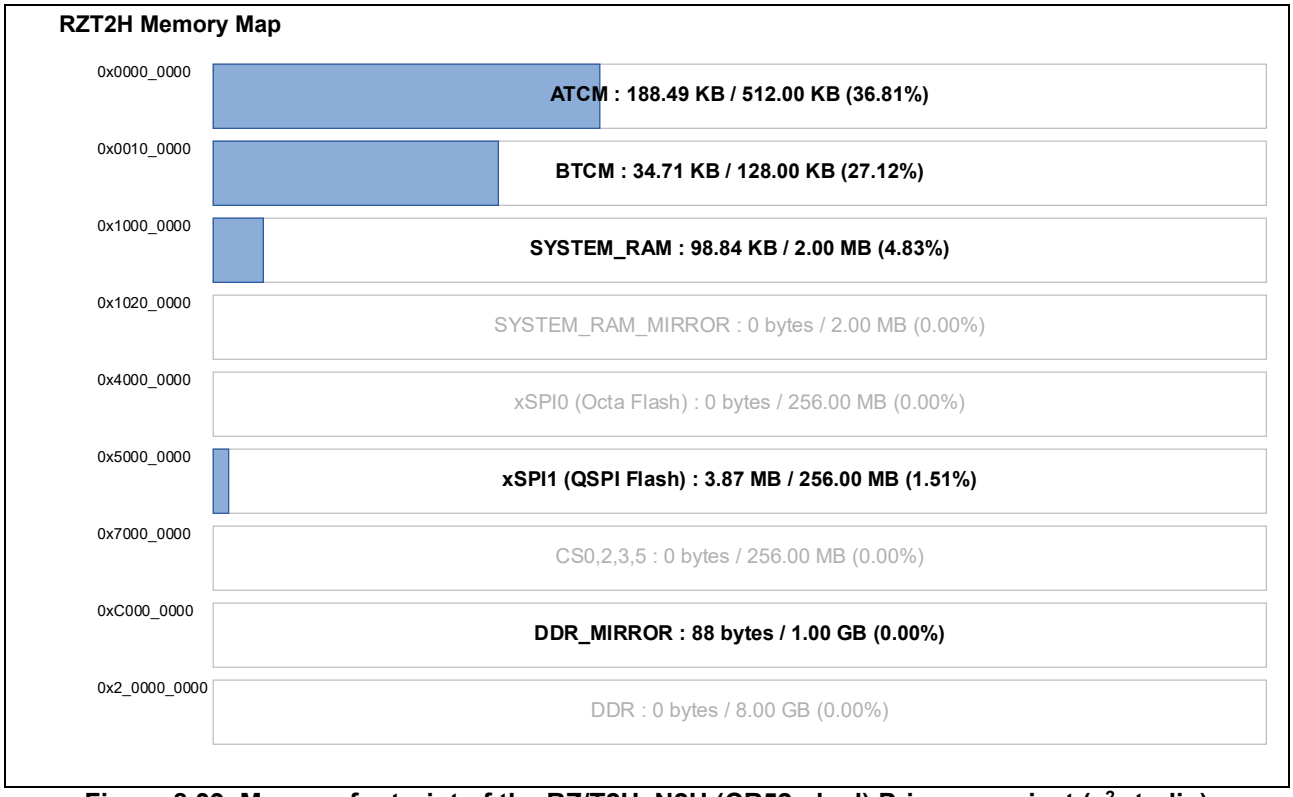
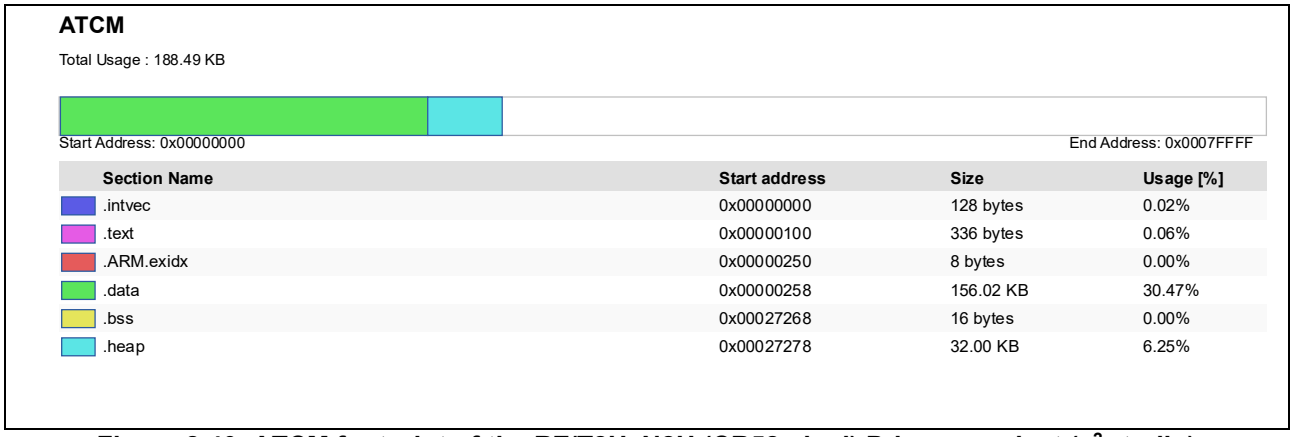
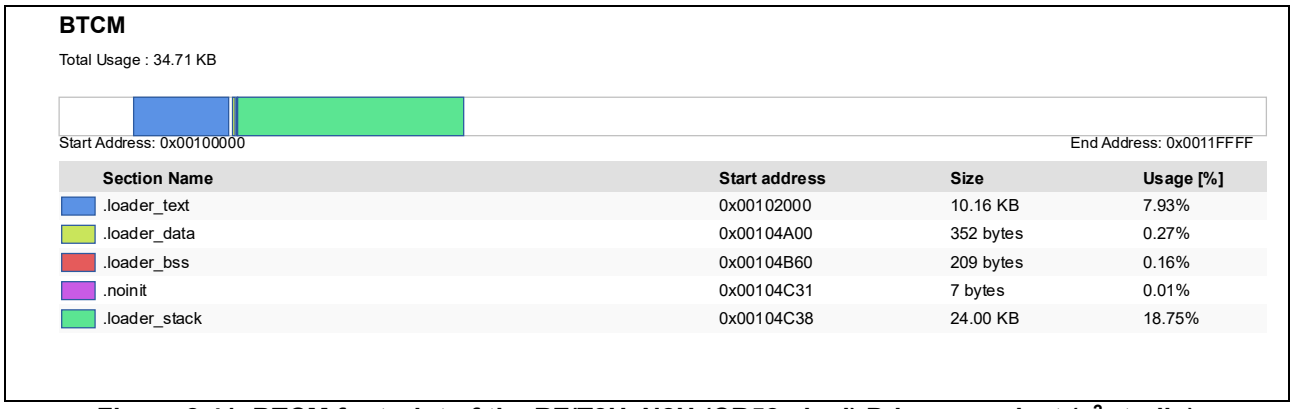


Figure 8.39. Memory footprint of the RZ/T2H, N2H (CR52_dual) Primary project (e² studio)



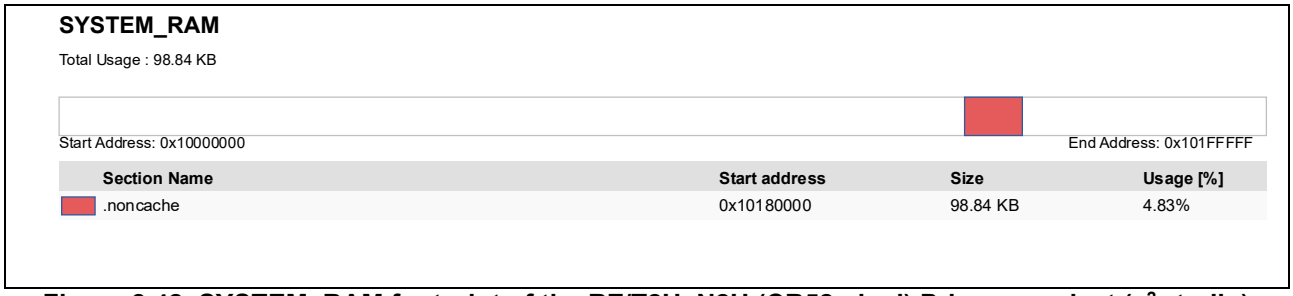
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.text	0x00000100	336 bytes	0.06%
.ARM.exidx	0x00000250	8 bytes	0.00%
.data	0x00000258	156.02 KB	30.47%
.bss	0x00027268	16 bytes	0.00%
.heap	0x00027278	32.00 KB	6.25%

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



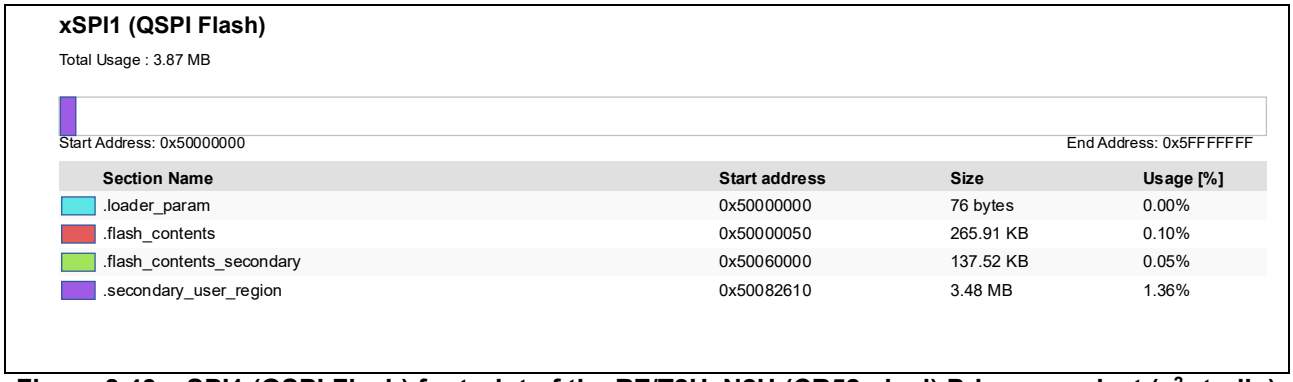
Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	10.16 KB	7.93%
.loader_data	0x00104A00	352 bytes	0.27%
.loader_bss	0x00104B60	209 bytes	0.16%
.noinit	0x00104C31	7 bytes	0.01%
.loader_stack	0x00104C38	24.00 KB	18.75%

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



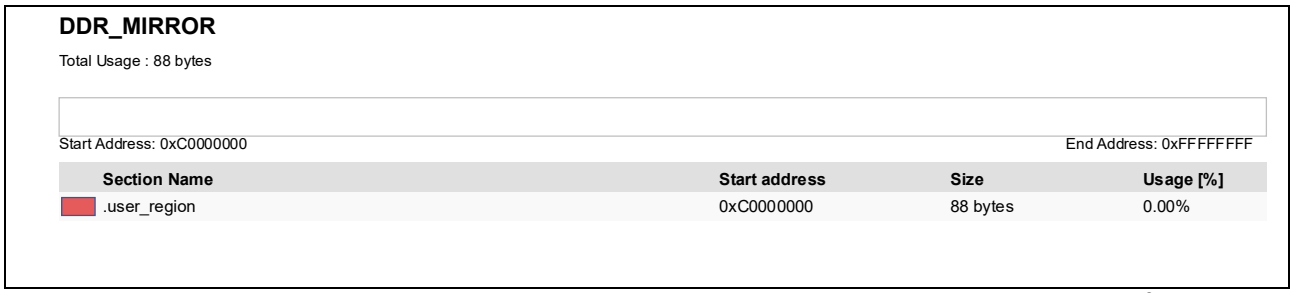
Section Name	Start address	Size	Usage [%]
.noncache	0x10180000	98.84 KB	4.83%

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

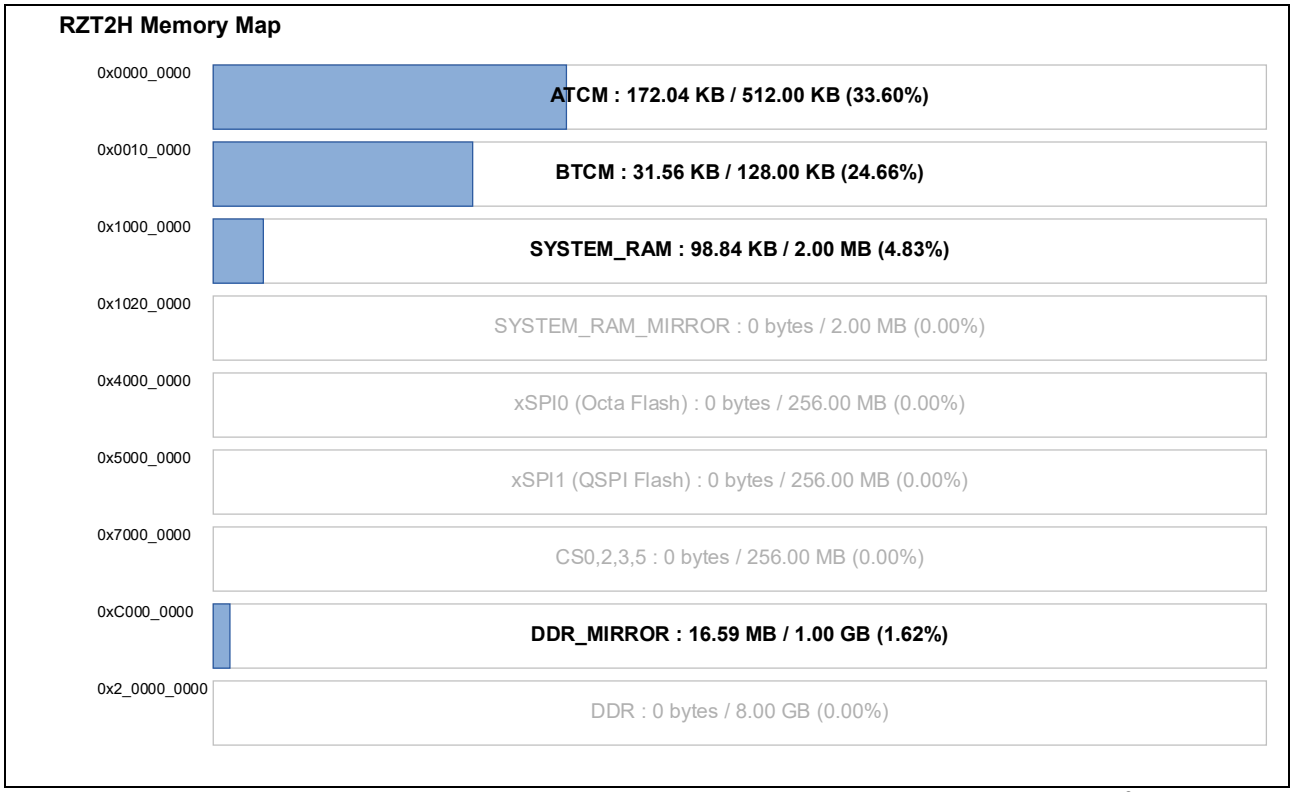
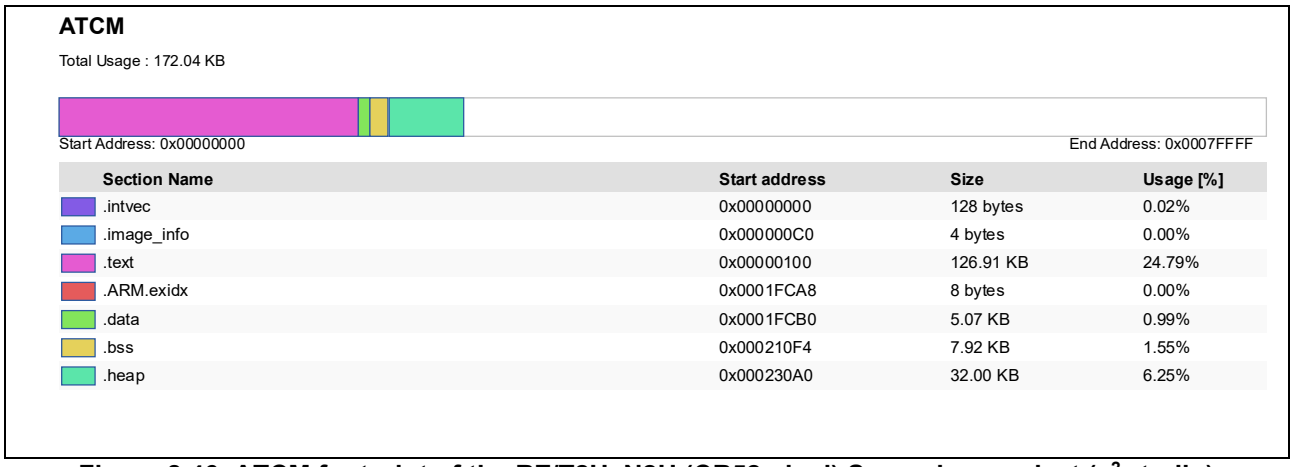
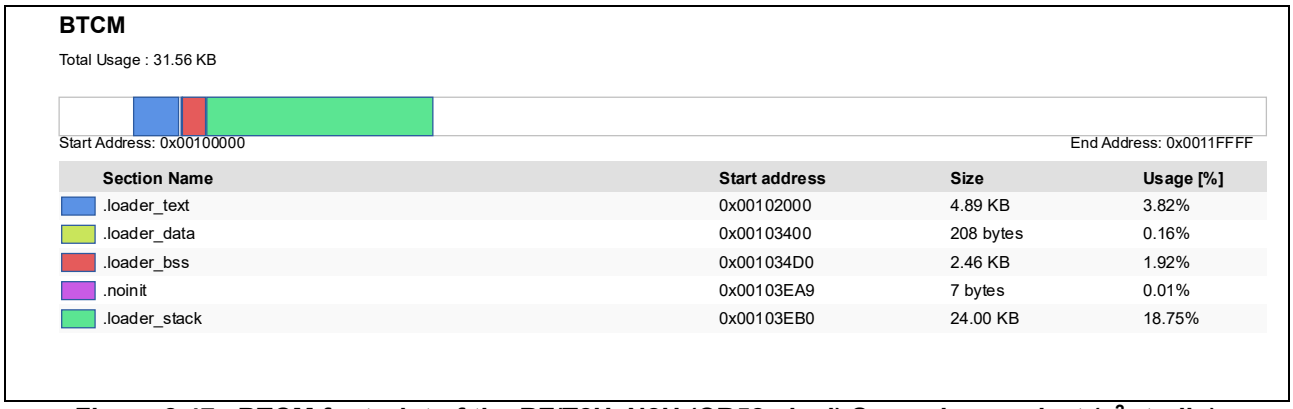


Figure 8.45. Memory footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (e² studio)



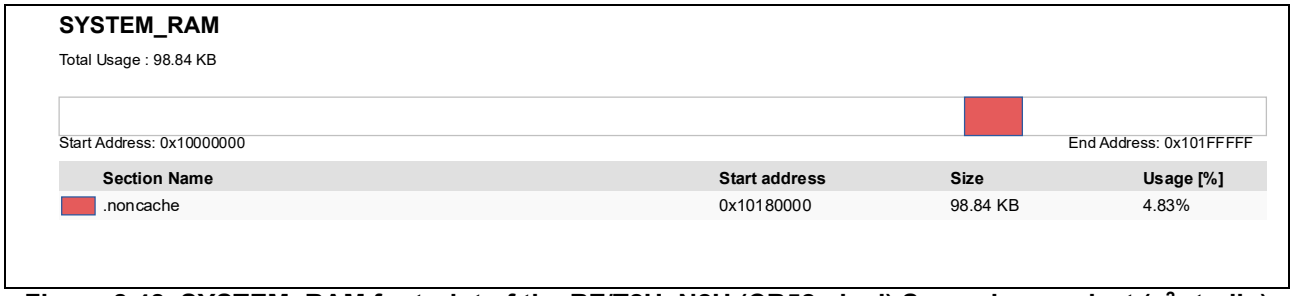
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.image_info	0x000000C0	4 bytes	0.00%
.text	0x00000100	126.91 KB	24.79%
.ARM.exidx	0x0001FCA8	8 bytes	0.00%
.data	0x0001FCB0	5.07 KB	0.99%
.bss	0x000210F4	7.92 KB	1.55%
.heap	0x000230A0	32.00 KB	6.25%

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



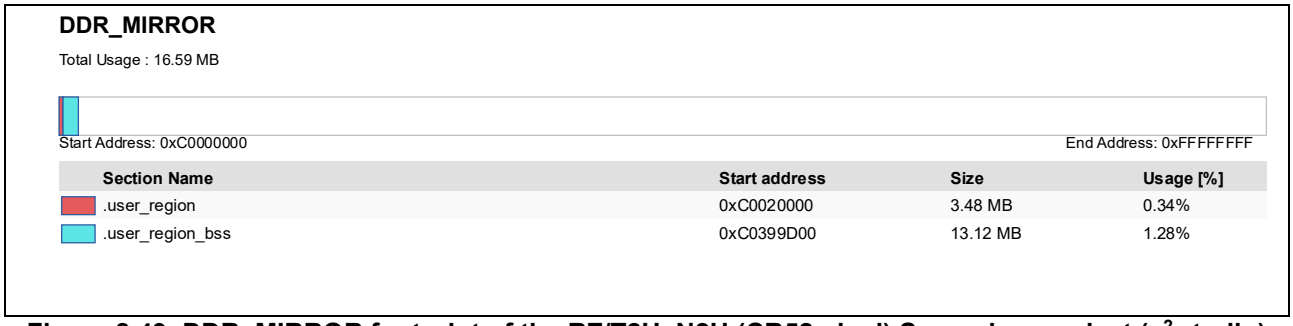
Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	4.89 KB	3.82%
.loader_data	0x00103400	208 bytes	0.16%
.loader_bss	0x001034D0	2.46 KB	1.92%
.noinit	0x00103EA9	7 bytes	0.01%
.loader_stack	0x00103EB0	24.00 KB	18.75%

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

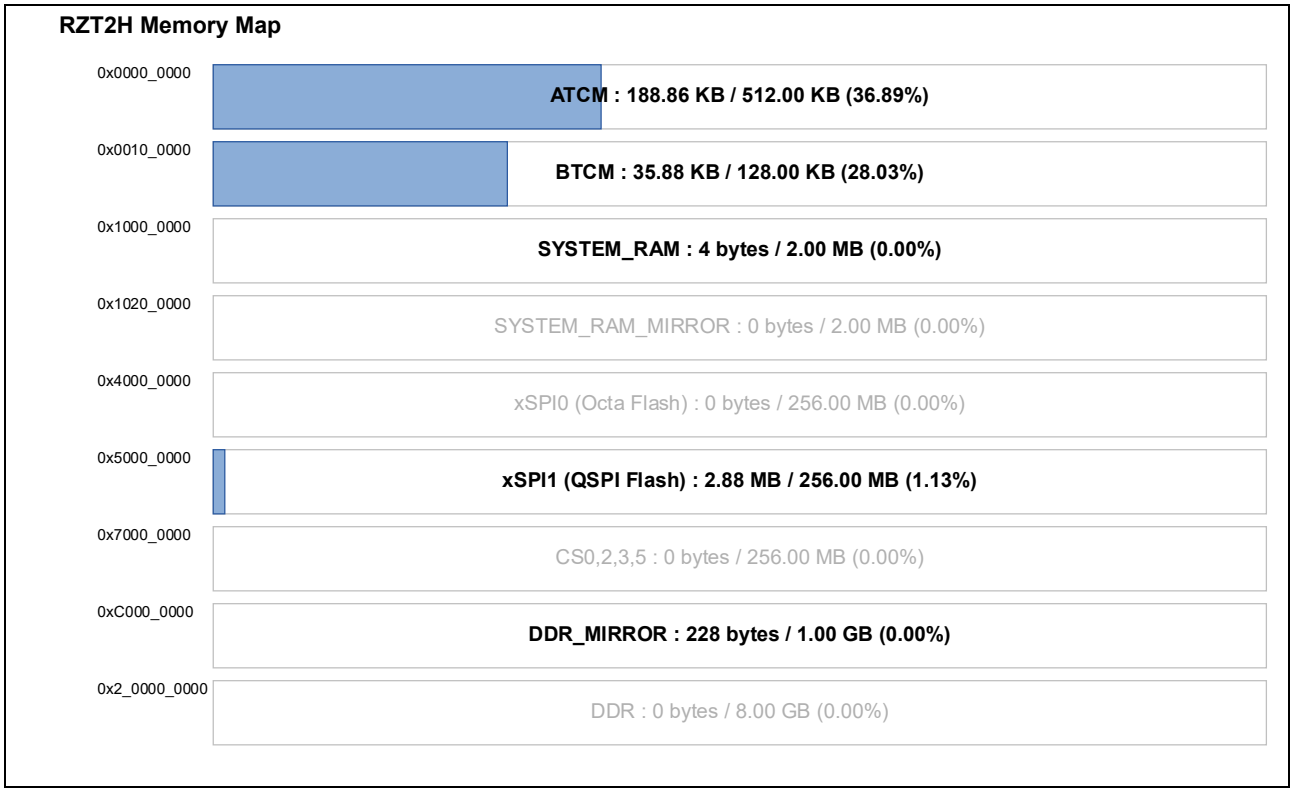
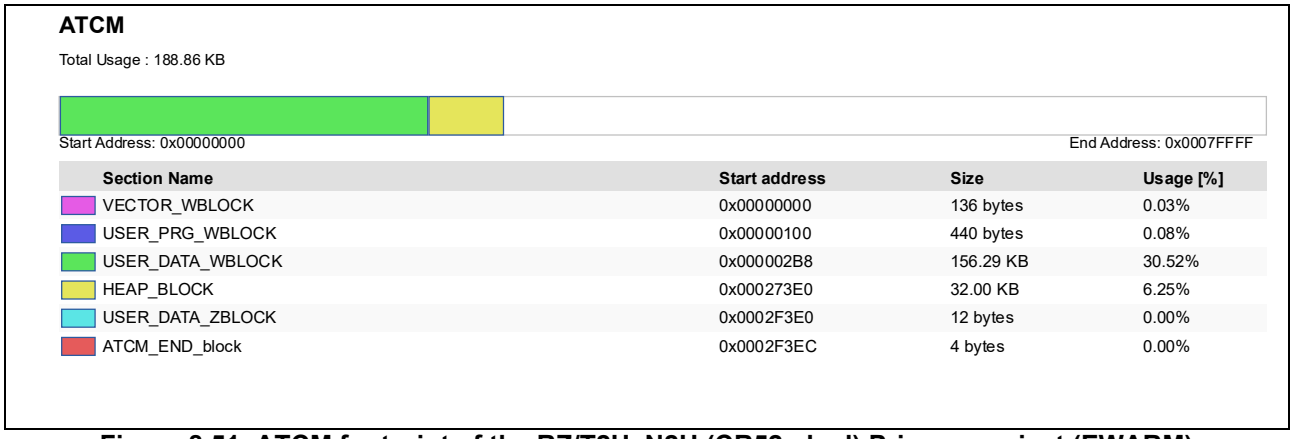


Figure 8.50. Memory footprint of the RZ/T2H, N2H (CR52_dual) Primary project (EWARM)



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)

BTCM

Total Usage : 35.88 KB

Start Address: 0x00100000 End Address: 0x0011FFFF

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)

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

xSPI1 (QSPI Flash)

Total Usage : 2.88 MB

Start Address: 0x50000000 End Address: 0x5FFFFFFF

Section Name	Start address	Size	Usage [%]
.loader_param	0x50000000	76 bytes	0.00%
LDR_PRG_RBLOCK	0x50000050	11.34 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)

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

(2) Secondary project

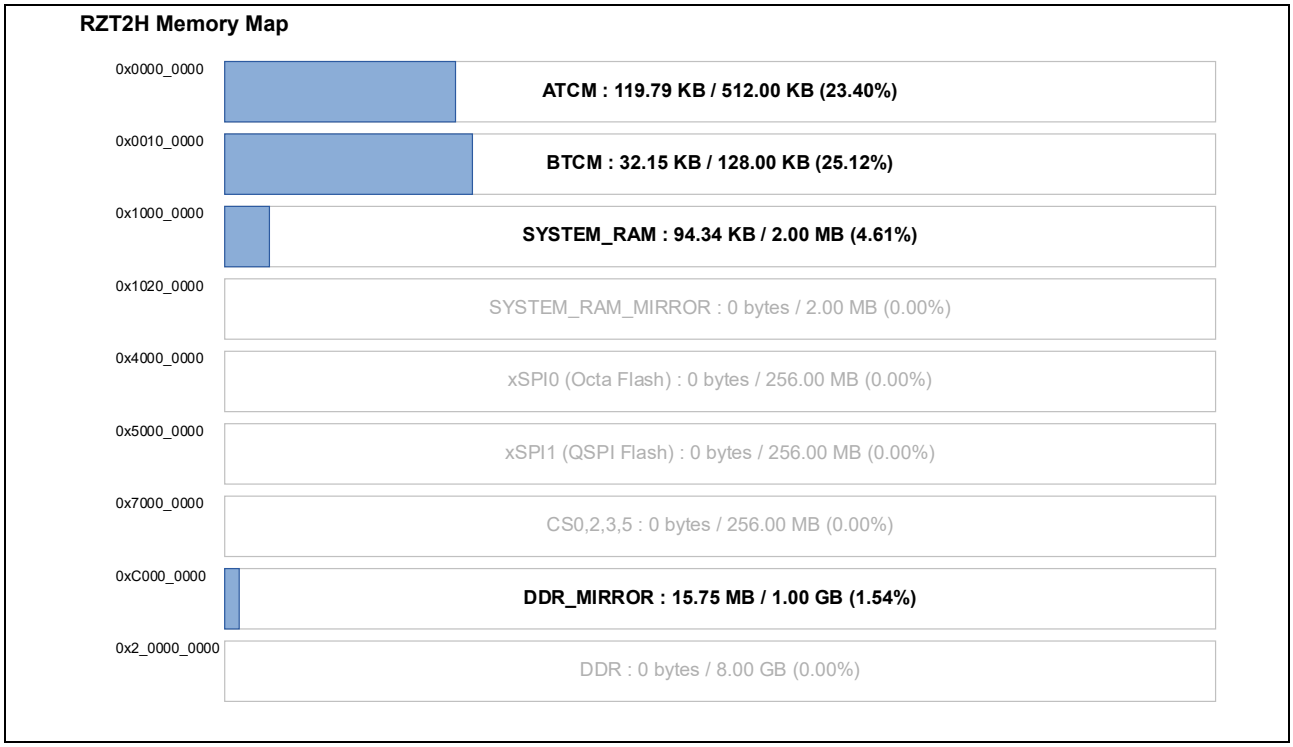
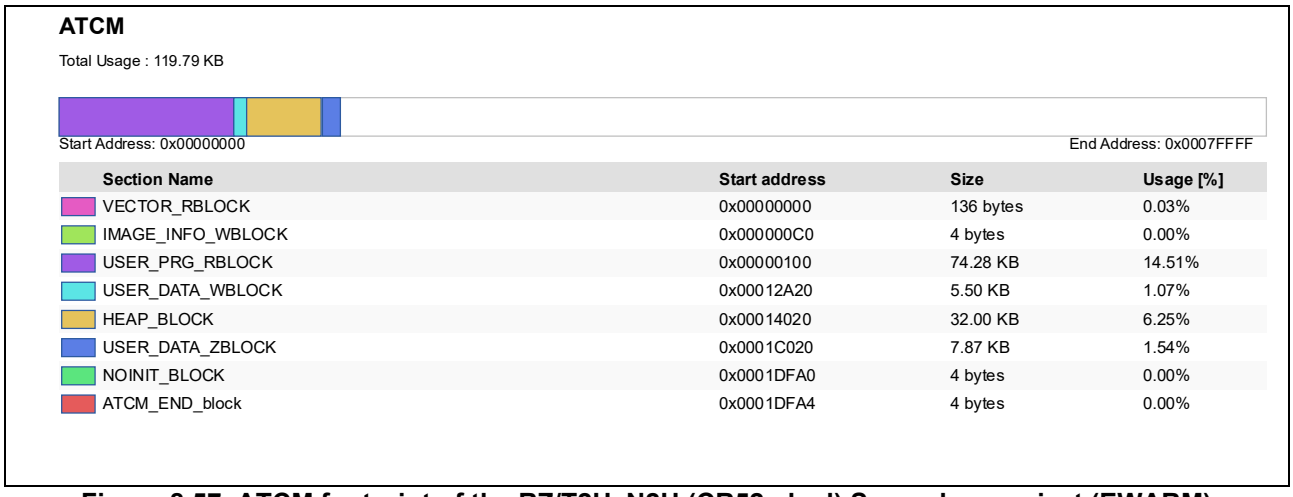
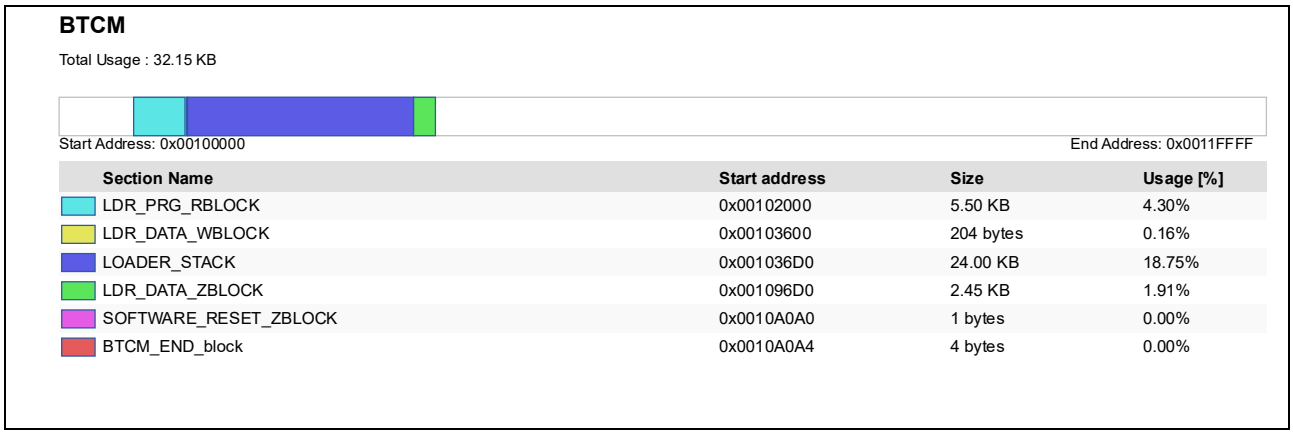


Figure 8.56. Memory footprint of the RZ/T2H, N2H (CR52_dual) Secondary project (EWARM)



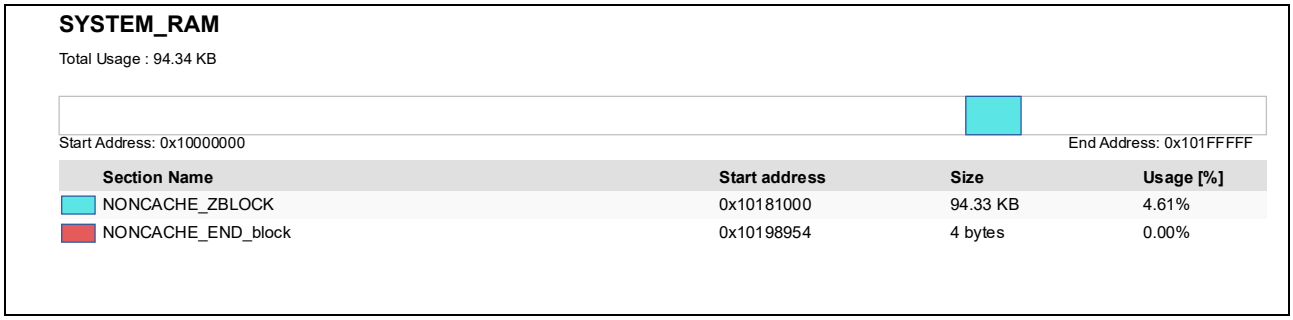
Section Name	Start address	Size	Usage [%]
VECTOR_RBLOCK	0x00000000	136 bytes	0.03%
IMAGE_INFO_WBLOCK	0x000000C0	4 bytes	0.00%
USER_PRG_RBLOCK	0x00000100	74.28 KB	14.51%
USER_DATA_WBLOCK	0x00012A20	5.50 KB	1.07%
HEAP_BLOCK	0x00014020	32.00 KB	6.25%
USER_DATA_ZBLOCK	0x0001C020	7.87 KB	1.54%
NOINIT_BLOCK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



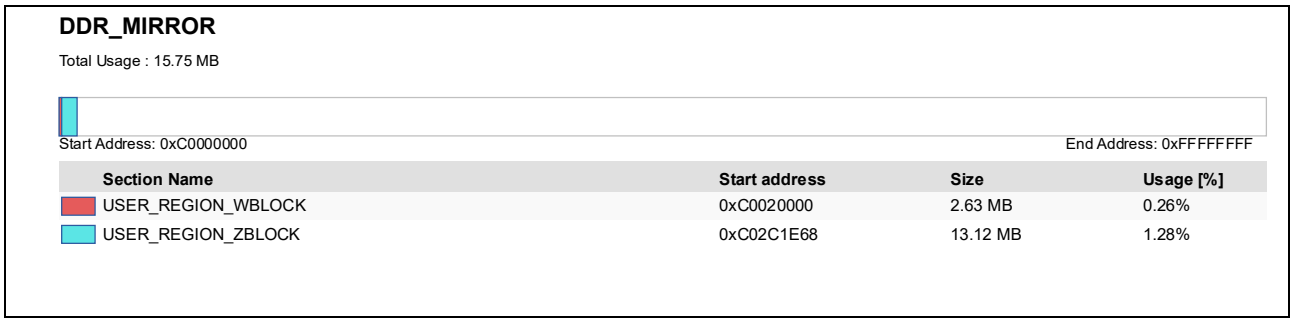
Section Name	Start address	Size	Usage [%]
LDR_PRG_RBLOCK	0x00102000	5.50 KB	4.30%
LDR_DATA_WBLOCK	0x00103600	204 bytes	0.16%
LOADER_STACK	0x001036D0	24.00 KB	18.75%
LDR_DATA_ZBLOCK	0x001096D0	2.45 KB	1.91%
SOFTWARE_RESET_ZBLOCK	0x0010A0A0	1 bytes	0.00%
BTCM_END_block	0x0010A0A4	4 bytes	0.00%

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)

* Since RZ/T2H and N2H are implemented with the same memory layout, the measurement results for RZ/T2H are presented here as representative.

8.3.4.1 e² studio

(1) Primary project

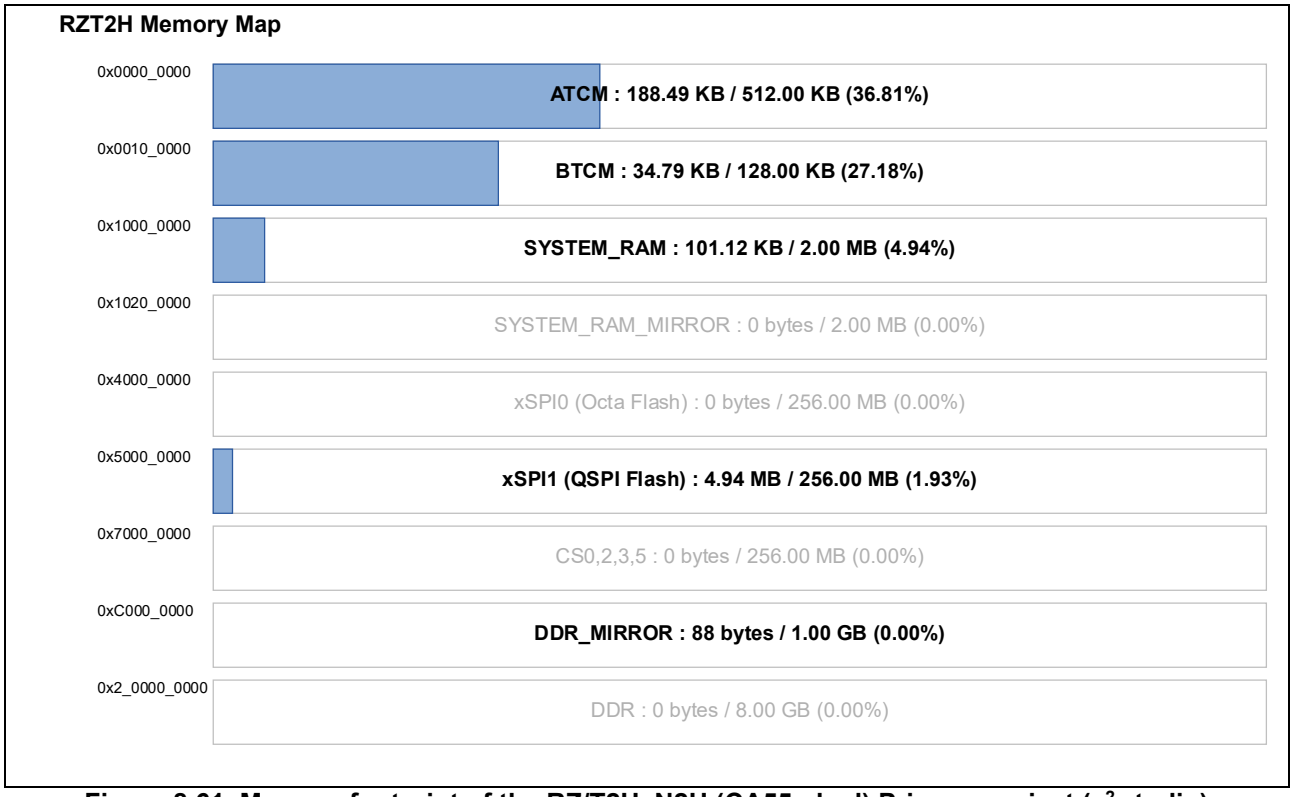
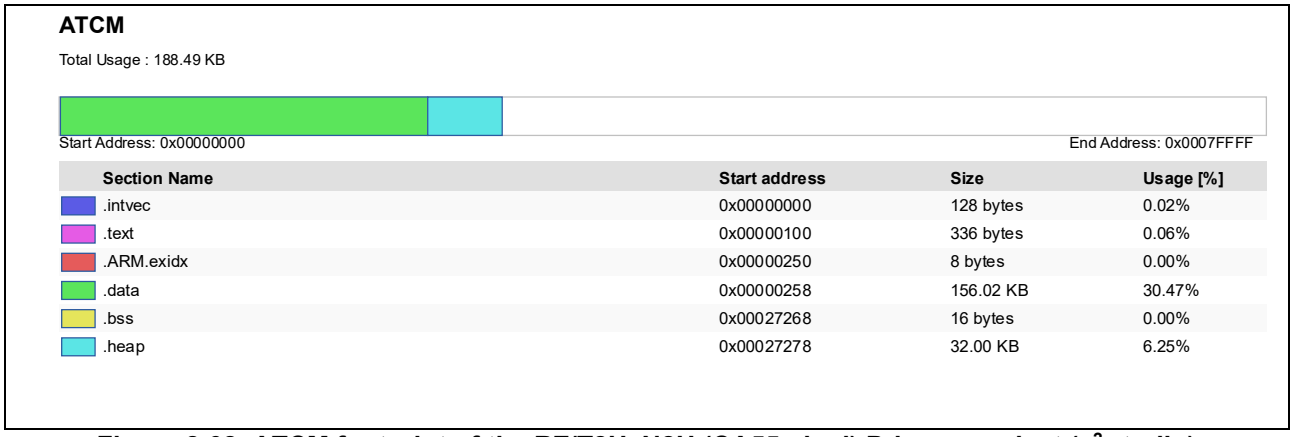
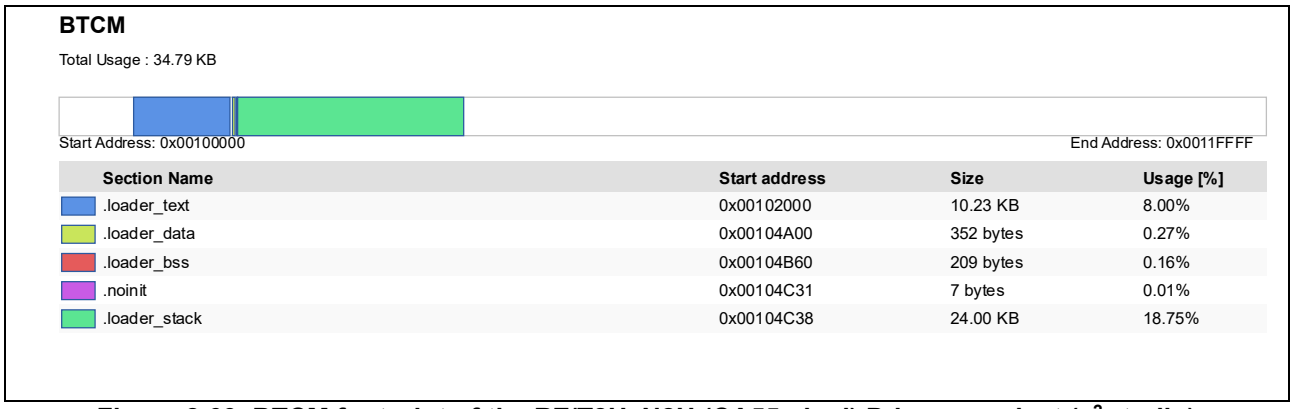


Figure 8.61. Memory footprint of the RZ/T2H, N2H (CA55_dual) Primary project (e² studio)



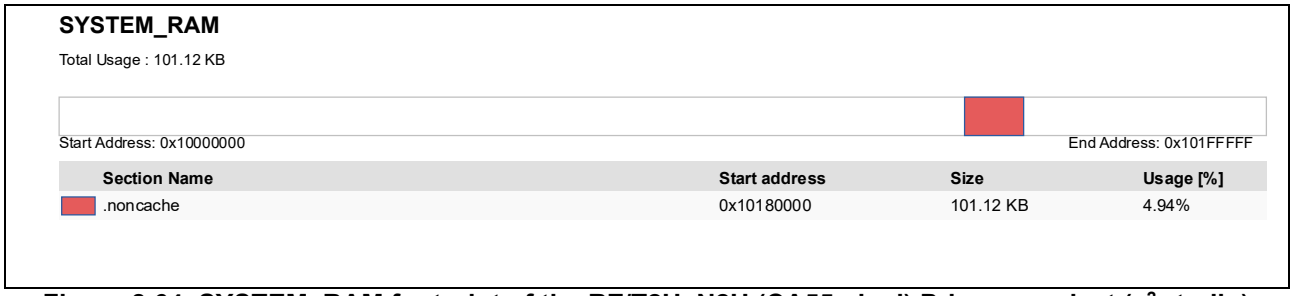
Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.02%
.text	0x00000100	336 bytes	0.06%
.ARM.exidx	0x00000250	8 bytes	0.00%
.data	0x00000258	156.02 KB	30.47%
.bss	0x00027268	16 bytes	0.00%
.heap	0x00027278	32.00 KB	6.25%

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



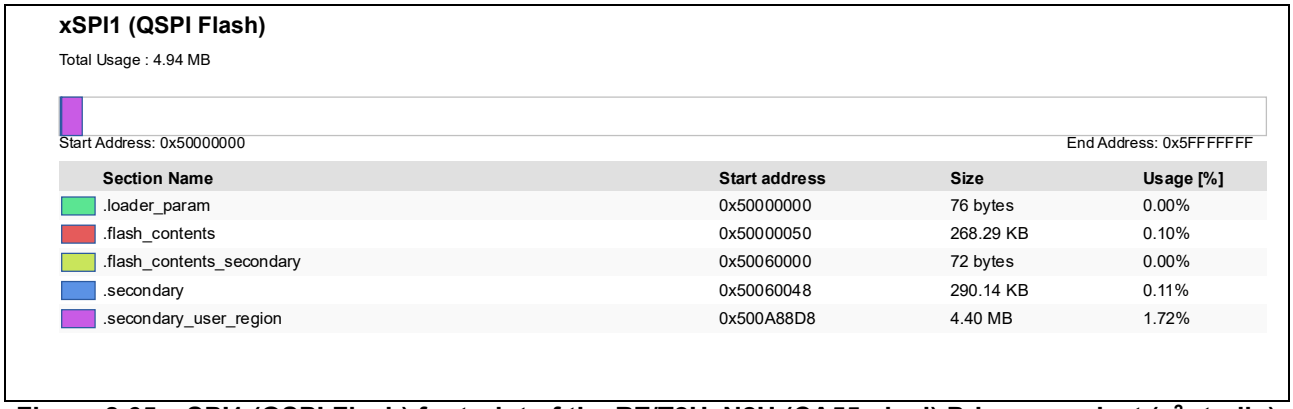
Section Name	Start address	Size	Usage [%]
.loader_text	0x00102000	10.23 KB	8.00%
.loader_data	0x00104A00	352 bytes	0.27%
.loader_bss	0x00104B60	209 bytes	0.16%
.noinit	0x00104C31	7 bytes	0.01%
.loader_stack	0x00104C38	24.00 KB	18.75%

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

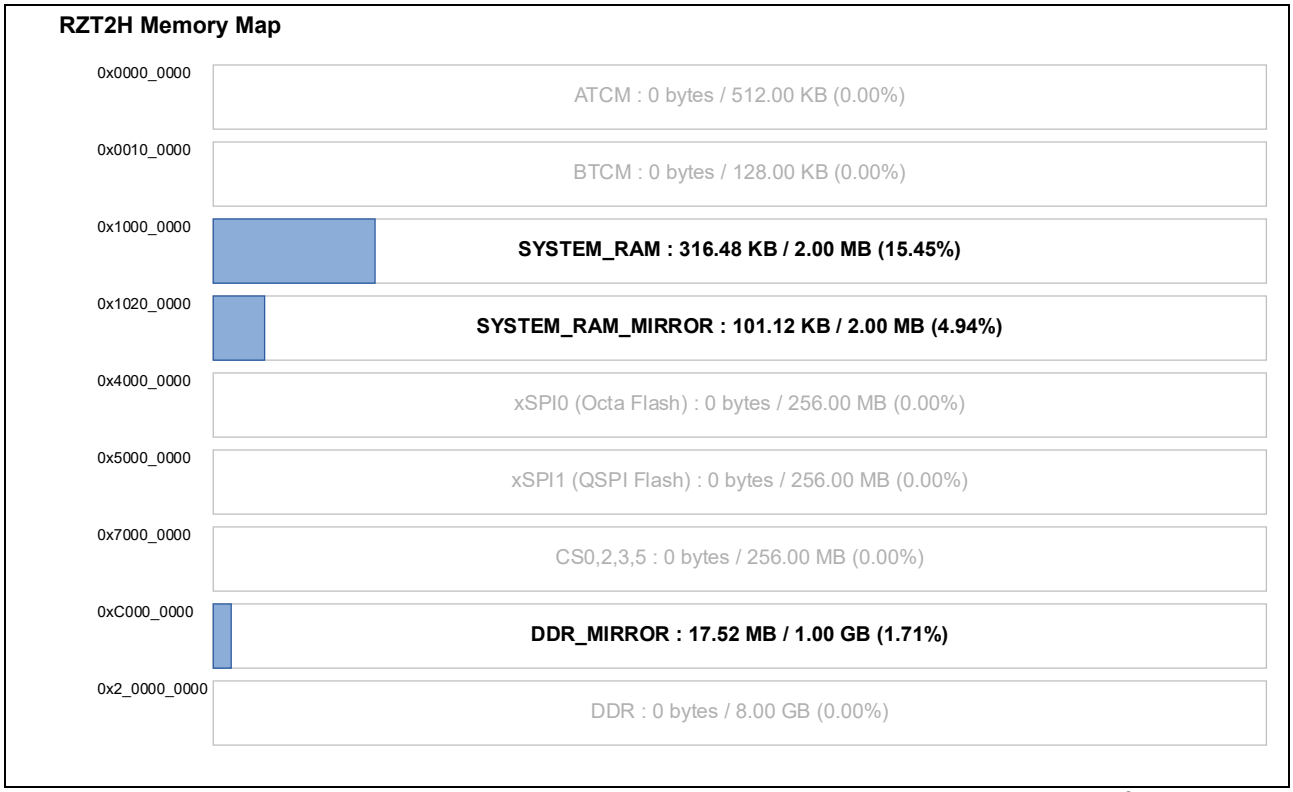
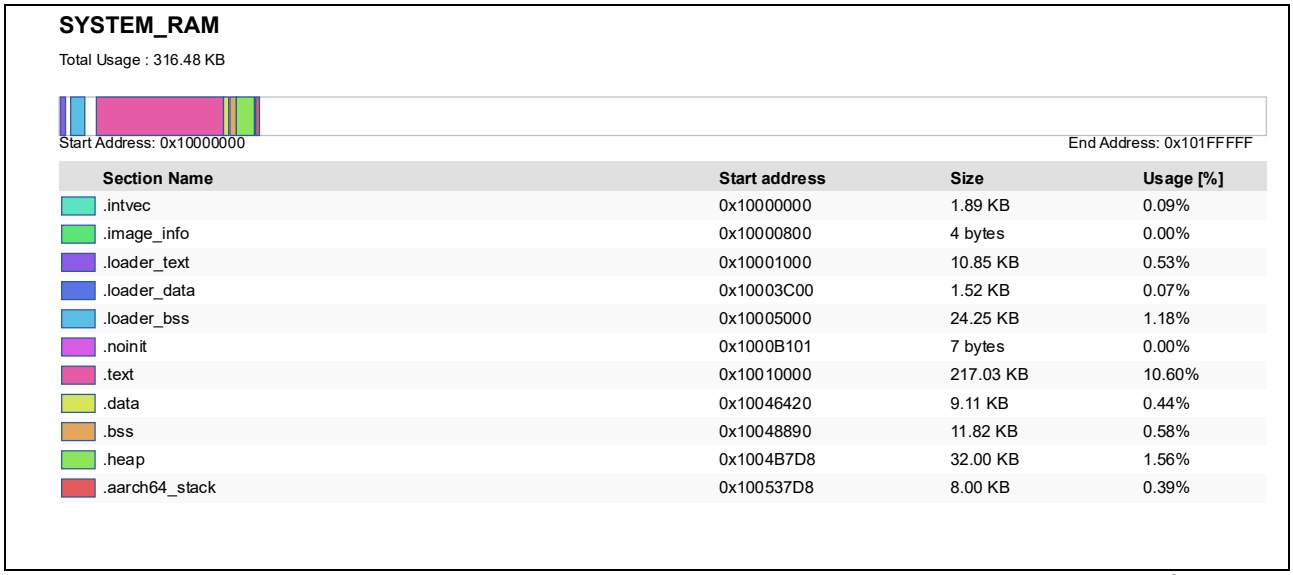


Figure 8.67. Memory footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)



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)

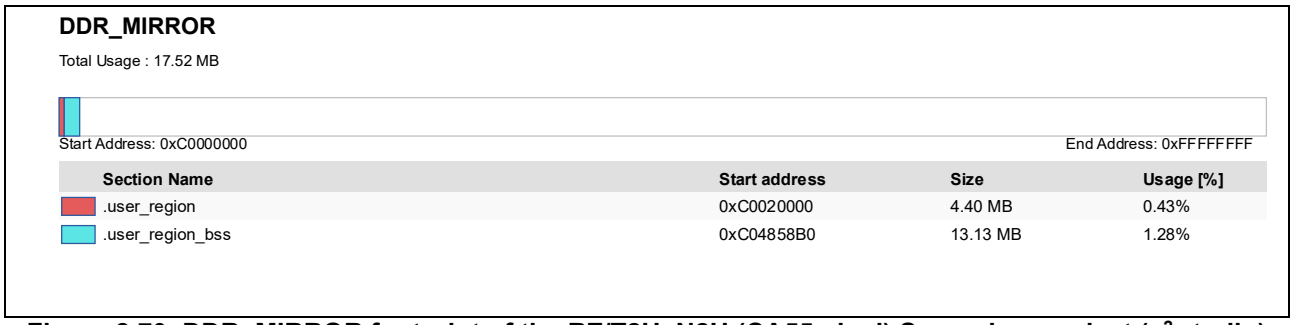


Figure 8.70. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (e² studio)

8.3.4.2 EWARM

(1) Primary project

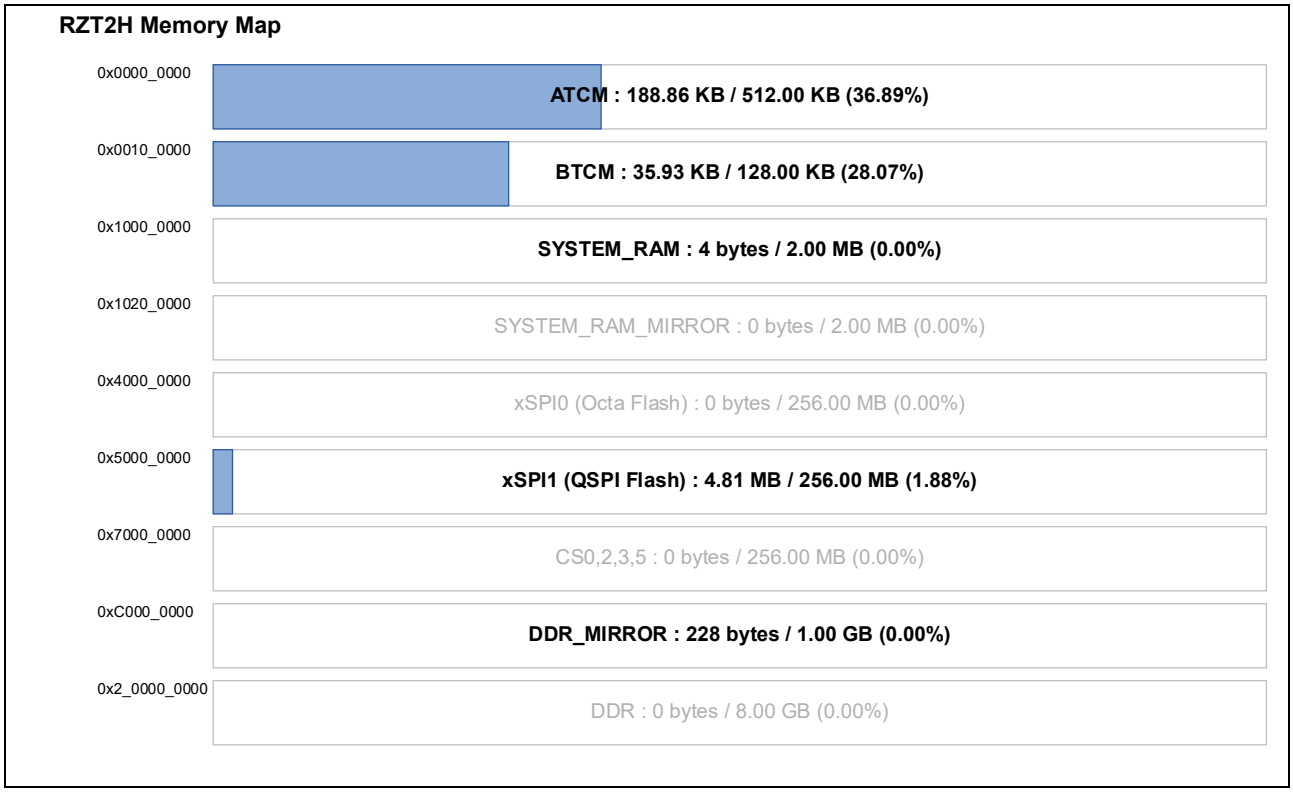
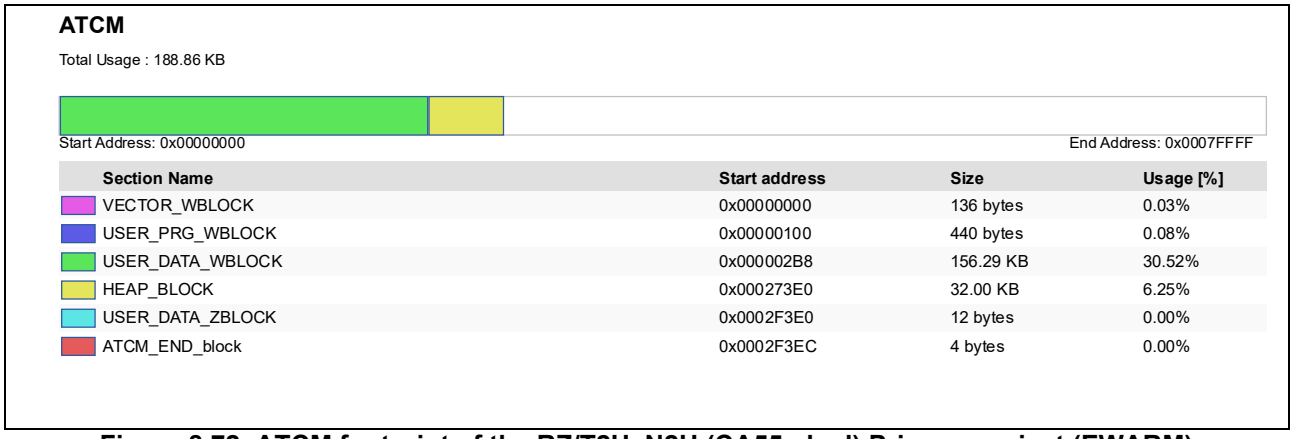


Figure 8.71. Memory footprint of the RZ/T2H, N2H (CA55_dual) Primary project (EWARM)



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

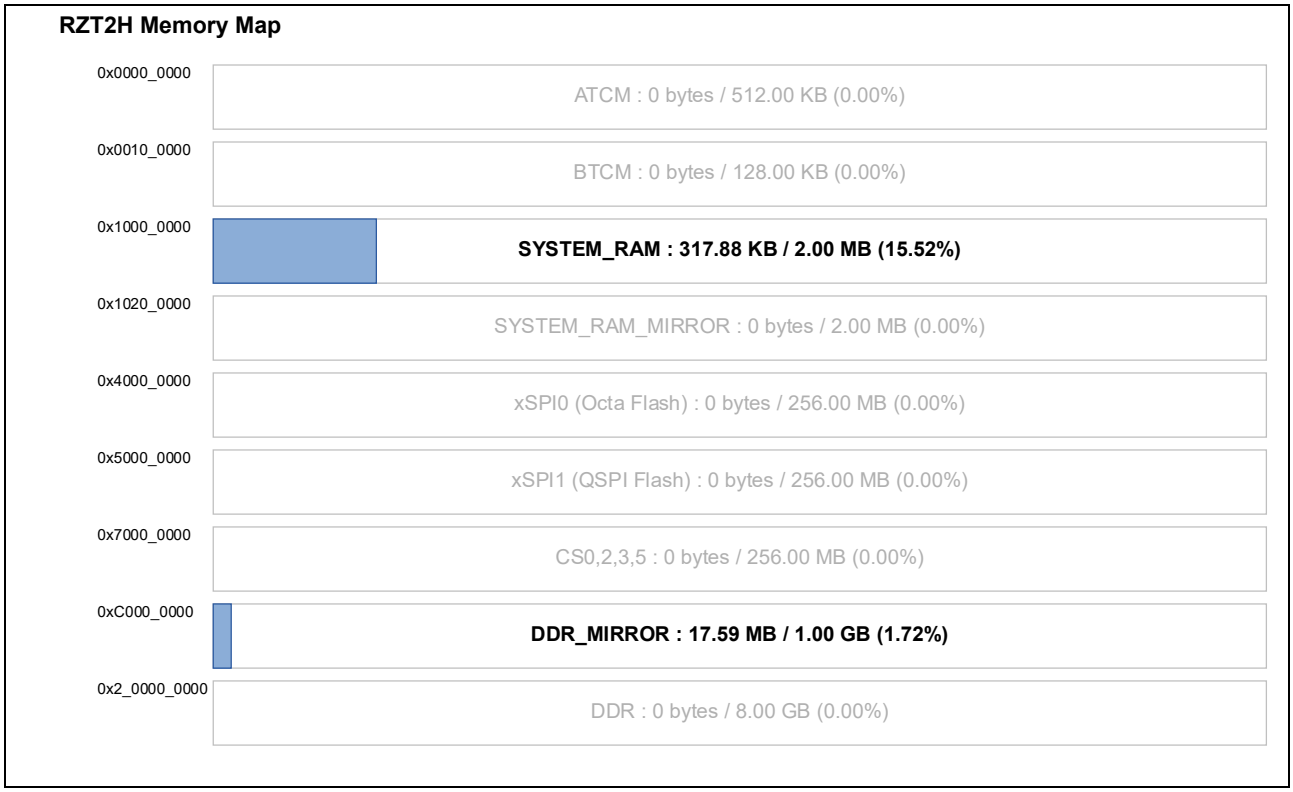
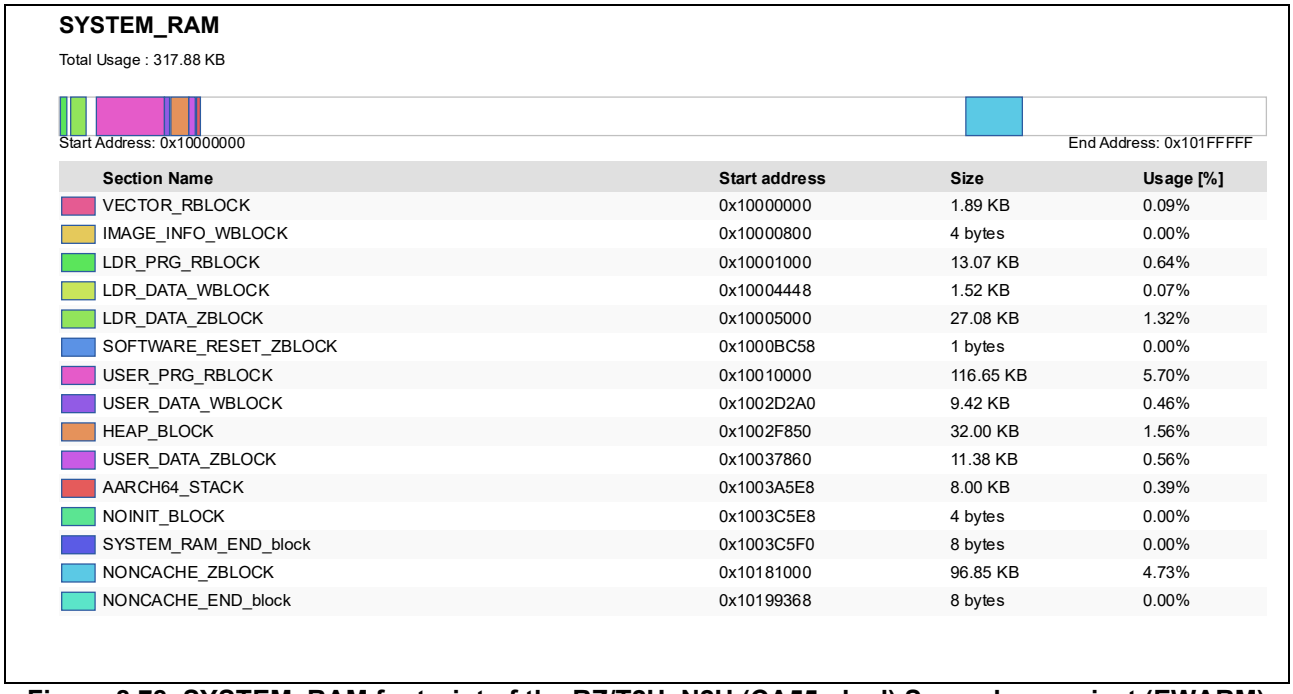


Figure 8.77. Memory footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)



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)

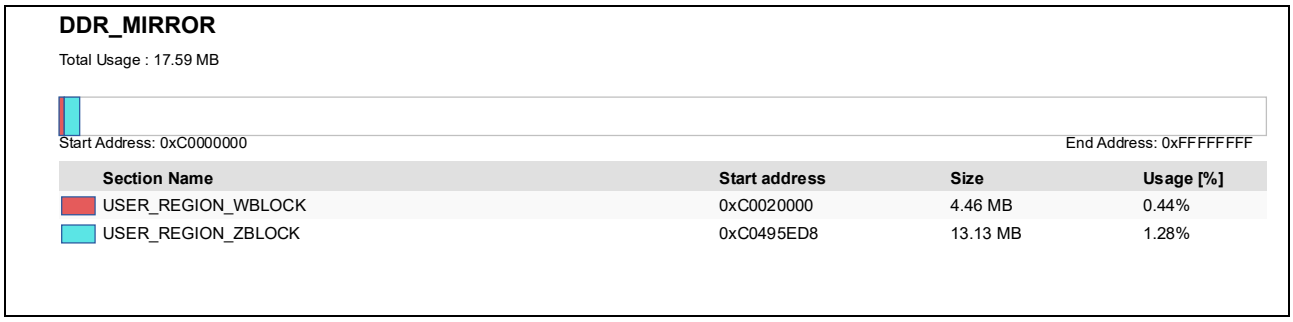


Figure 8.79. DDR_MIRROR footprint of the RZ/T2H, N2H (CA55_dual) Secondary project (EWARM)

9. Appendix

9.1 File Generation of open62541

To run open62541 in a FreeRTOS + lwIP environment, the CMake-based build flow introduced at the link below is used to configure open62541 and compile the information model NodeSet files.

open62541 Documentation — open62541 1.3.0-dirty documentation

This chapter describes the procedure for generating open62541 and Renesas sample information models as files for e² studio or EWARM execution in a Windows environment. Here Windows10 version 1903(OS Build 18362.1049) or later, or Windows11 is used, in which WSL2 is executable.

9.1.1 Linux environment Setup

Set up a Linux environment to run CMake. In this document, we will run CMake on a Linux (Ubuntu 18.04) environment installed using WSL2 with reference to the following linked pages.

(Reference) Manual installation steps for older versions of WSL | Microsoft Learn

1) Launch PowerShell as Administrator. Search PowerShell > right-click > Run as Administrator
2) Enter the following command to enable the Windows Subsystem for Linux.
   dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

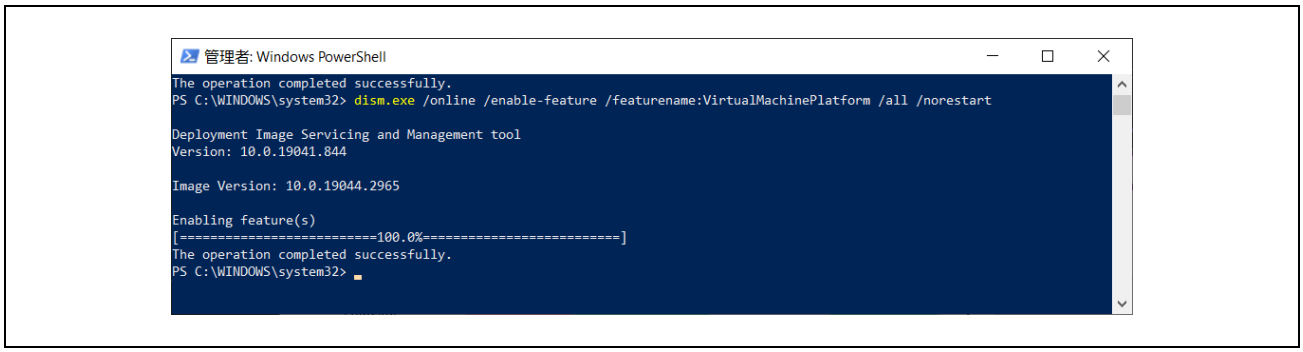


Figure 9.1. Microsoft-Windows-Subsystem-Linux

3) Enter the following command to enable the virtual machine platform feature:
   dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

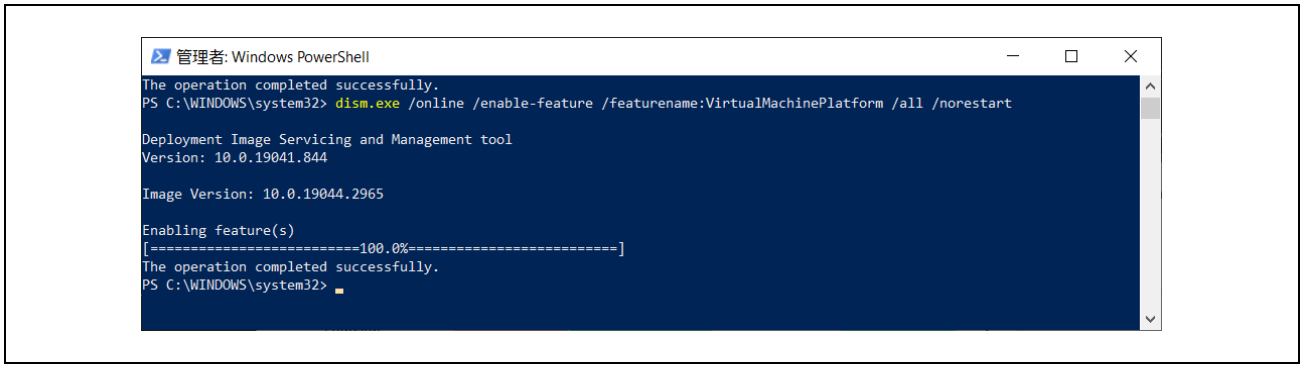


Figure 9.2. VirtualMachinePlatform

4) Restart your PC and complete the WSL installation.
5) Download and run the WSL2 Linux kernel update package for x64 machines below.

   WSL2 Linux kernel update package for x64 machines

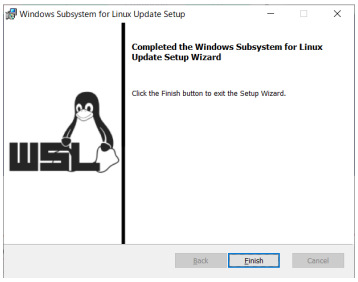


Figure 9.3. wsl_update_x64.msi

6) Run the following command to set WSL 2 as the default version.

wsl --set-default-version 2

7) Download Linux distribution. Here download Ubuntu 18.04 below.

Ubuntu 18.04

8) Go to the folder containing the downloaded file and execute the following command.

Add-AppxPackage . \Ubuntu_1804.2019.522.0_x64.appx

9) Double-click Ubuntu_1804.2019.522.0_x64.appx to install.

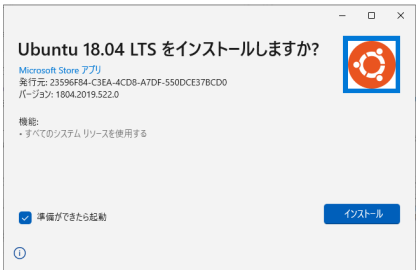


Figure 9.4. Ubuntu Install

10) Set the Linux username and password.

(Reference) Set up a WSL development environment | Microsoft Learn

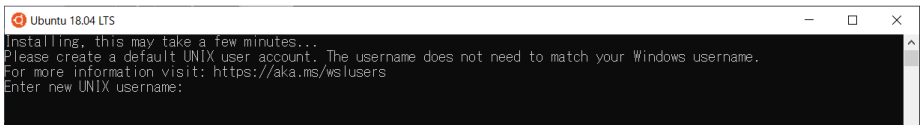


Figure 9.5. UNIX username

9.1.2 Install CMake

11) Execute the following Linux command to update apt-get

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) Execute the following Linux command

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

When the following screen appears during the process, select OK.

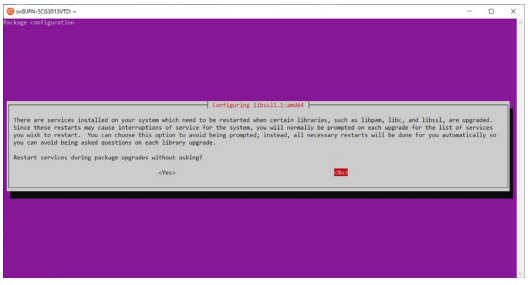


Figure 9.8. restart

13) Execute each of the following commands.

```
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
```

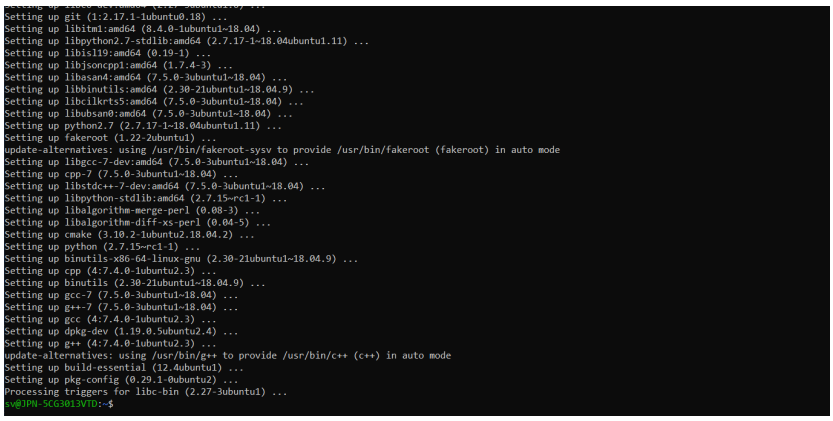


Figure 9.9. install

9.1.3 open62541 File Generation

14) Clone open62541 to any folder

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

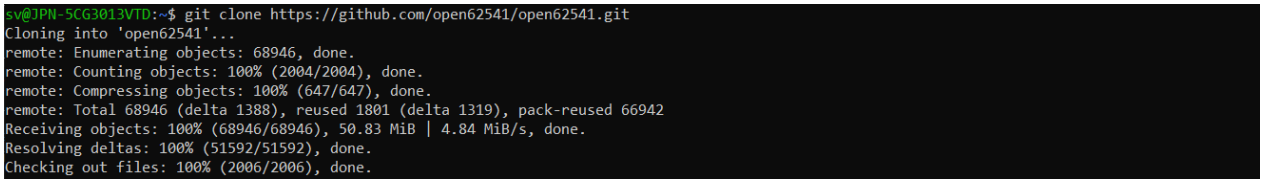


Figure 9.10. git clone

15) Go to /open62541 directory and check out the specific version (here, version v1.3.18).

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

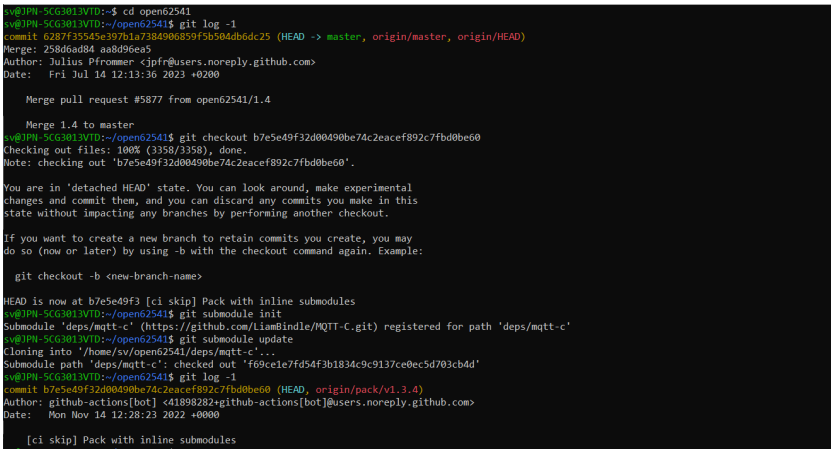


Figure 9.11. git submodule

16) Open the Linux folder from File Explorer. Confirm that CMakeLists.txt is present in */home/(username)/open62541* directory. Copy the following three patch files obtained by unzipping patch_open62541.zip attached to the sample software to this directory.

diff.patch

Opc.Ua.Renesas.NodeSet2.xml

patch.sh

17) Execute following command in /open62541 directory

bash patch.sh

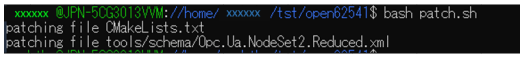


Figure 9.12. command result

18) Compile the library according to the standard procedures of the cmake project. Create /open62541/build directory and run cmake. (Some items will be Failed, but that is not a problem.)

mkdir build && cd build

cmake ..

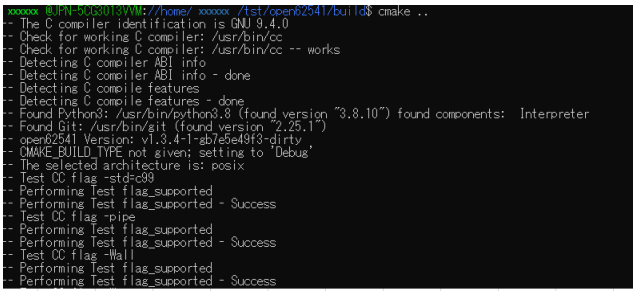


Figure 9.13. cmake

19) Execute the following command to start the ccmake setting window.

ccmake ..

20) After changing the settings as Figure 9.14, press [t] toggle advanced mode and change “UA_ENABLE_NODEMANAGEMENT” to “OFF” and “UA_ENABLE_PUBSUB_ENCRYPTION” to “ON”. After executing [c] to configure, close it by [q] to quit without generating.

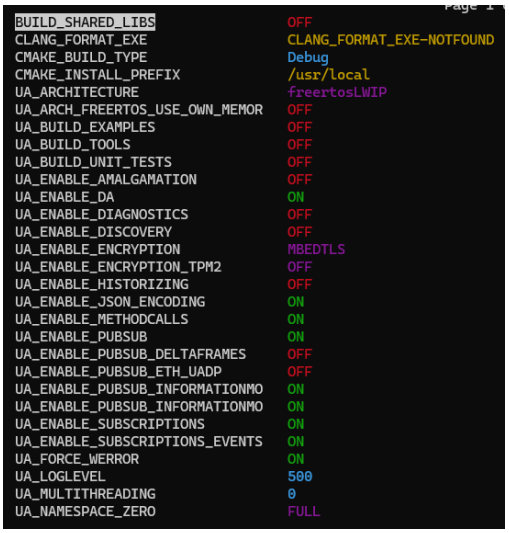


Figure 9.14. ccmake configuration

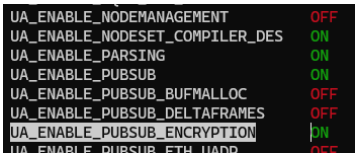


Figure 9.15. ccmake advanced configuration

21) Execute the following command to make in the */open62541/build* directory. The make process will finish with an error, but it does not matter.

make -j

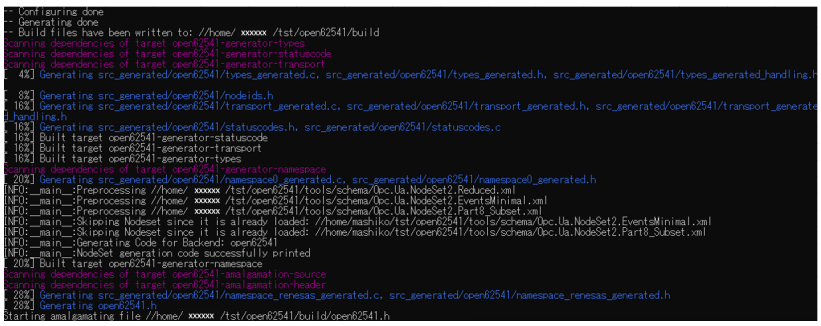


Figure 9.16. make

22) Verify that the files have been generated in the /open62541/build/src_generated/open62541 directory.

9.2 TFTP

This sample software has TFTP Server functionality. In the operation check, a Windows PC connected to the evaluation board is used as the TFTP Client. Note that the Windows TFTP Client operation is disabled by default.

9.2.1 File Writing

Launch Command Prompt and change directory to the folder where the files you wish to write reside. As an example, to write “cert_server.der”, enter the following command. “Transfer successful” is displayed, then the “cert_server.der” is successfully written to the “<USB_ROOT>/pki/application/own/certs” folder of the USB flash drive connected to the evaluation board.

tftp -i 192.168.10.100 put cert_server.der pki/application/own/certs/cert_server.der

9.2.2 File Reading

Launch Command Prompt. As an example, to read “<USB_ROOT>/pki/application/own/certs/cert_server.der” from the USB flash drive, enter the following command. If the read is successful, the message “Transfer successful” will be displayed and “cert_server.der” will be stored in the current directory.

tftp -i 192.168.10.100 get pki/application/own/certs/cert_server.der

9.3 LLDP, SNMP MIB

Table 9.1 shows the LLDP functionality and the corresponding MIB of this sample software.

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

This chapter explains how to configure OPC UA PubSub settings using Method. When configuring PubSub settings using Method, note that this software is limited in that the layout for Pub or Sub cannot be freely configured; it uses a specific custom layout. Therefore, when communicating with other devices, the communication counterpart must match this layout. Additionally, PubSub communication using this feature only supports security None.

9.4.1 Publisher

The role of each device when checking OPC UA Publisher operation is shown in Figure 9.17.

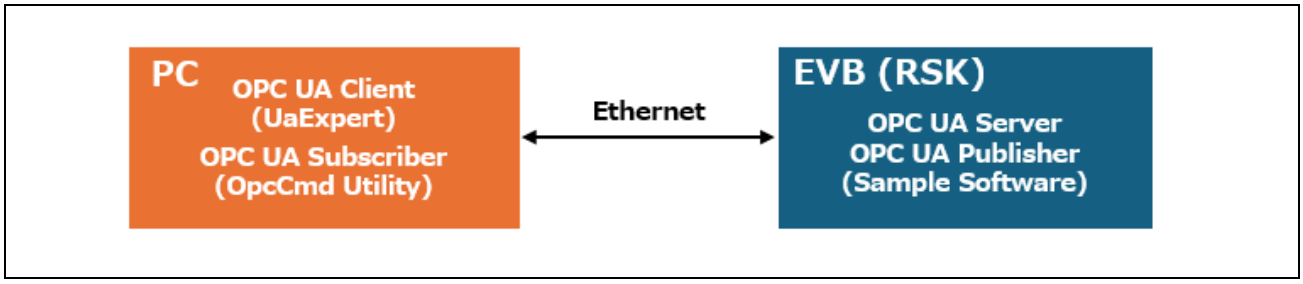


Figure 9.17. Connection Block Diagram for OPC UA Publisher

After connecting as shown in chapter 4, perform the operation check using the following procedure.

- Launch and connect to UaExpert

Connect UaExpert to the OPC UA Server using the same procedure as in chapter 7.2.1.

Right-click on the "Root>Objects>Server>PublishSubscribe>PublishedDataSets>AddPublishedDataItems" Node in the Object tree of the Address Space window and click on the “Call…” that appears.

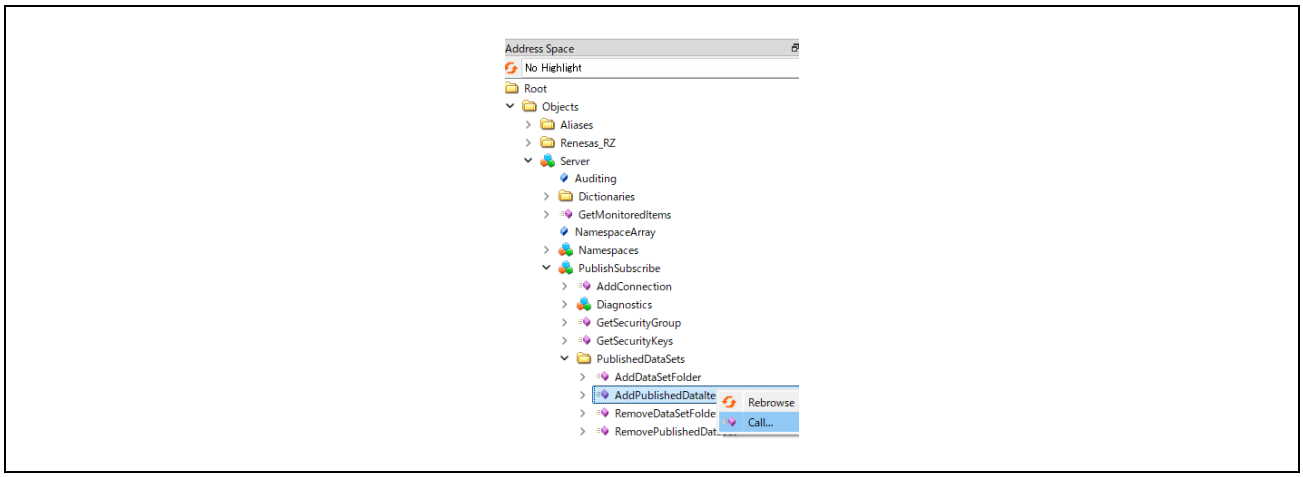


Figure 9.18. Address Space Window

In the window that appears, enter the information for the Node you wish to Publish and click Call. Figure 9.19 shows an example of two settings, Boolean(NS=2, ID=6049) and DateTime(NS=2, ID=6052).

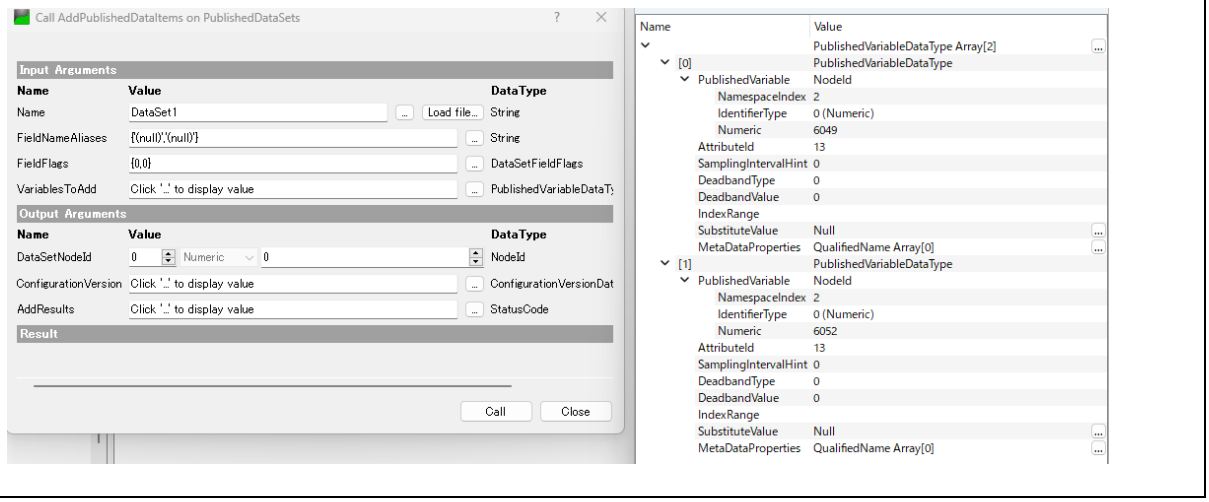


Figure 9.19. Call AddPublishedDataItems

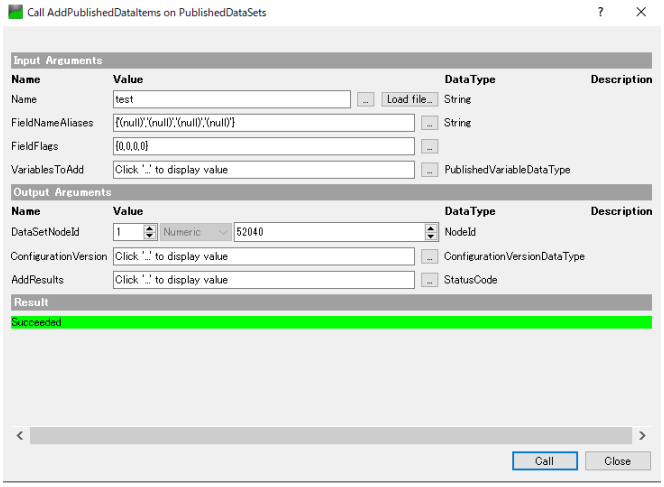


Figure 9.20. Call Result

Right-click on the "Root>Objects>Server>PublishSubscribe> AddConnection" Node and click on the “Call…” that appears.

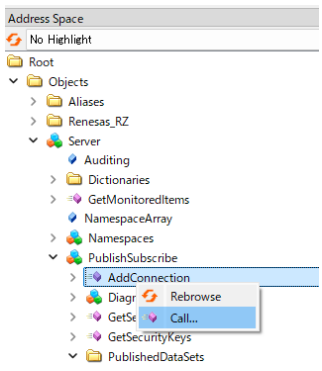


Figure 9.21 Address Space Window

In the window that appears, enter the information for each field as shown in the image below, then click “Call.”

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

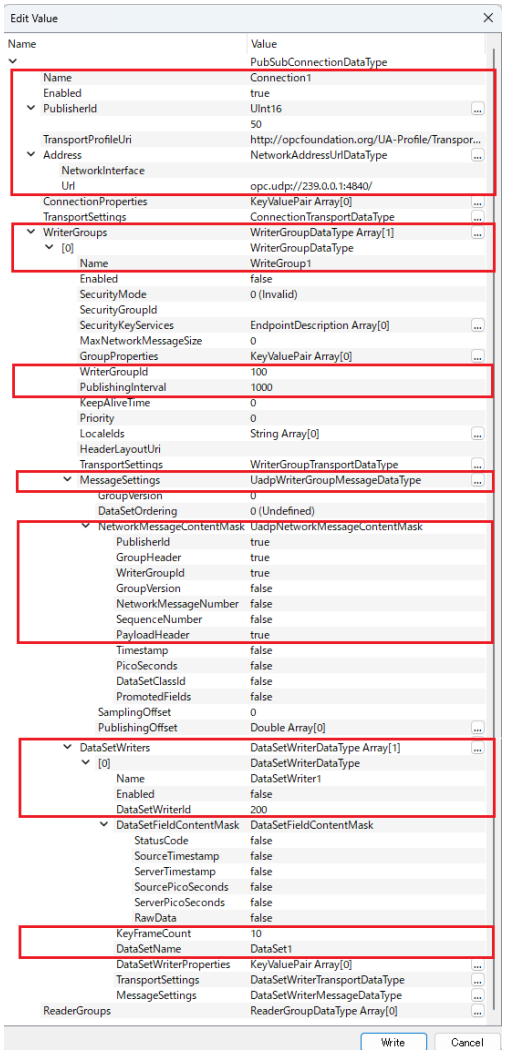


Figure 9.22. Call AddConnection

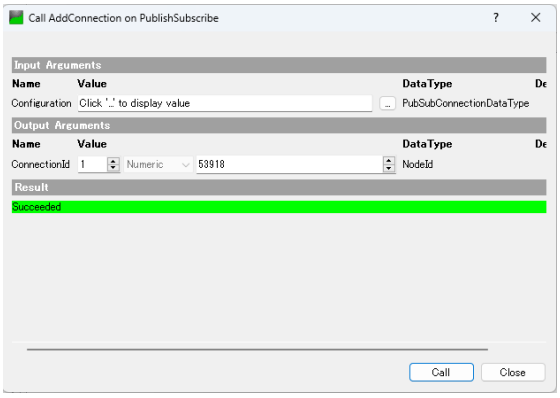


Figure 9.23. Call Result

If the method call is successful, publishing will begin.

- Launch OpcCmd Utility

  Launch Command prompt and change directory to the OpcCmd Utility folder. Enter the following command. This command is to subscribe to all published messages. Subscribing will continue for one minute.

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

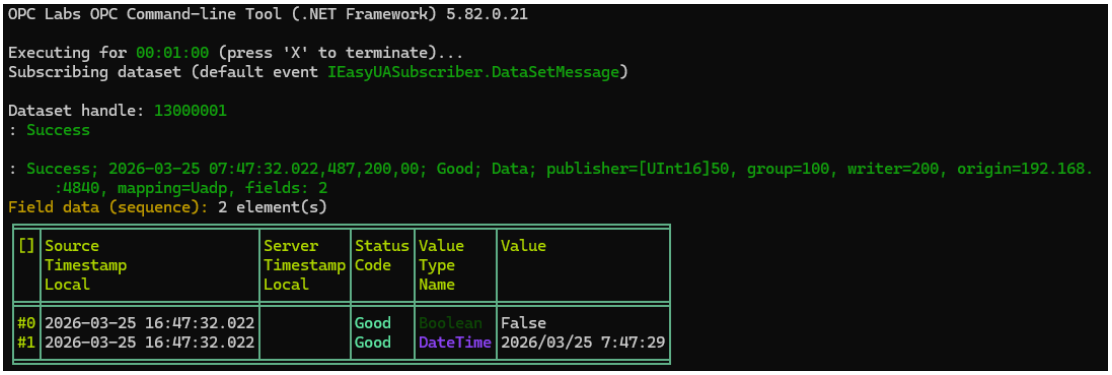


Figure 9.24. Command Execution Result

9.4.2 Subscriber

The role of each device when checking OPC UA Subscriber operation is shown in Figure 9.25. In this test, a single evaluation board serves as both the Publisher and the Subscriber.

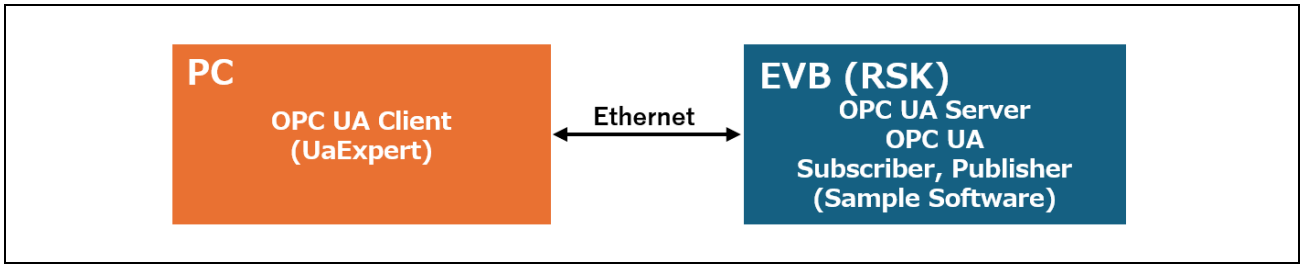


Figure 9.25. Connection Block Diagram for OPC UA Subscriber

・Operation check

Start the Publish process following the same procedure as in chapter 9.4.1.

Right-click on the "Root>Objects>Server>PublishSubscribe>AddConnection" Node in the Object tree of the Address Space window and click on the "Call…" that appears.

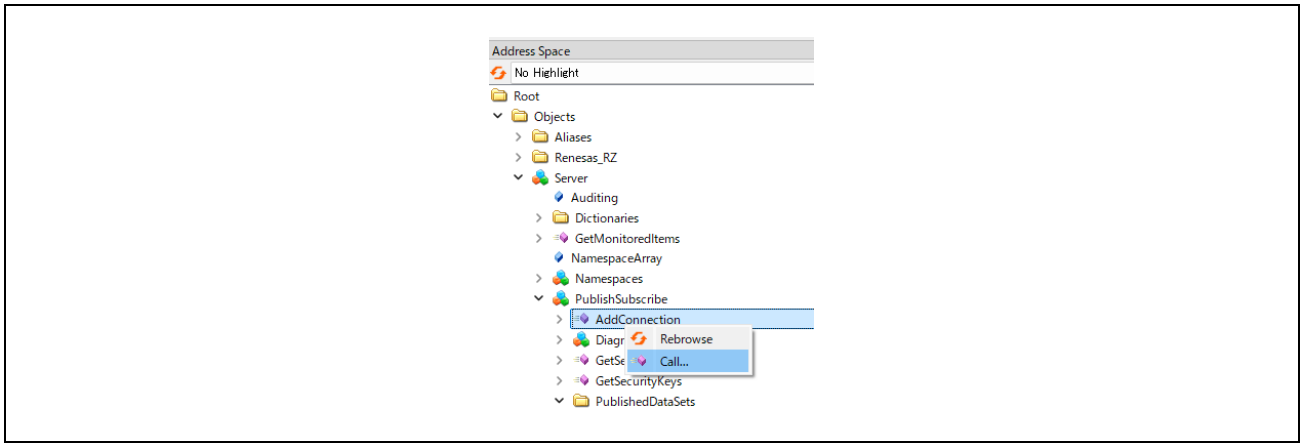


Figure 9.26. Address Space Window

In the window that appears, enter the information for each field as shown in the images below, then click “Call.”

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

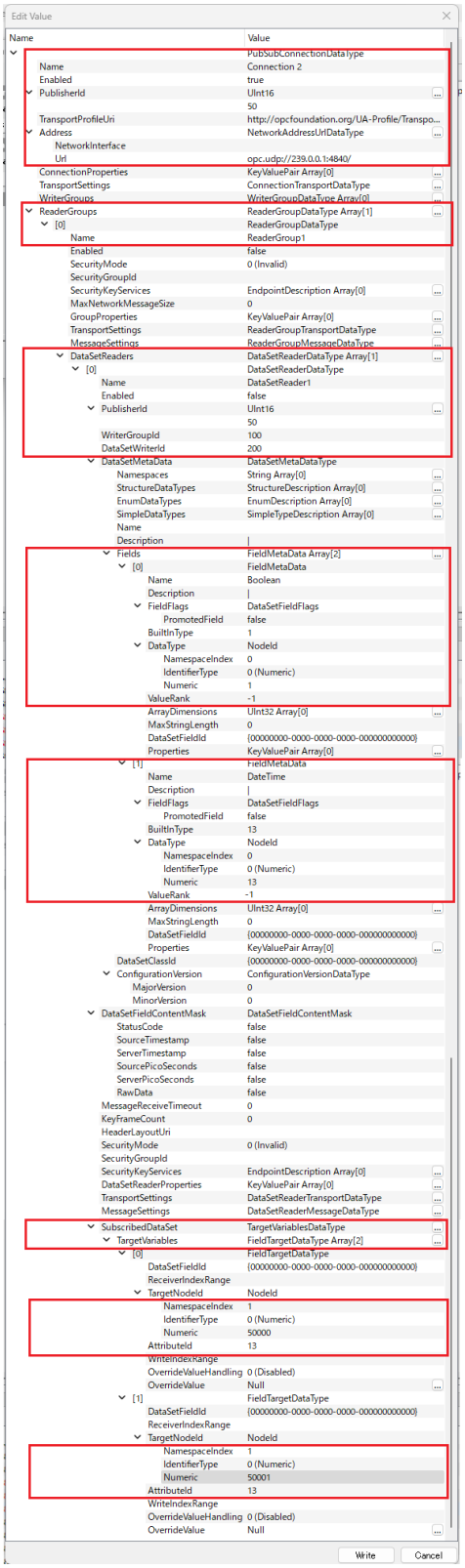


Figure 9.27. Call AddConnection

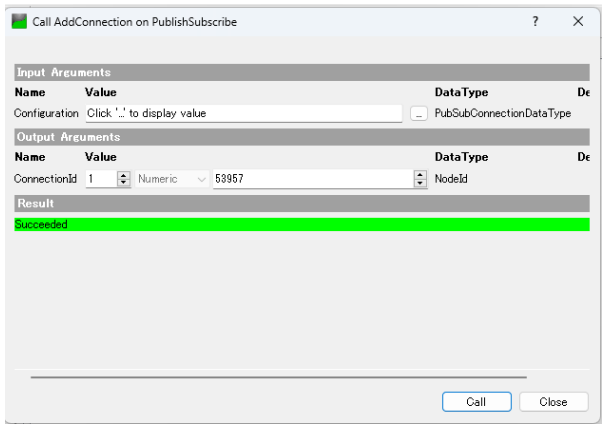


Figure 9.28. Call Result

After successful call method, right-click on "Root" in the Address Space Window of UaExpert and click “Rebrowse”. You will see that new Nodes have been added, as shown in Figure 9.30. These are the Nodes for storing the subscribed information.

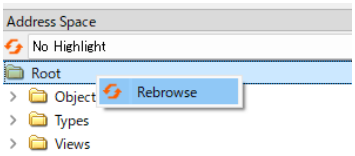


Figure 9.29. Rebrowse on UaExpert

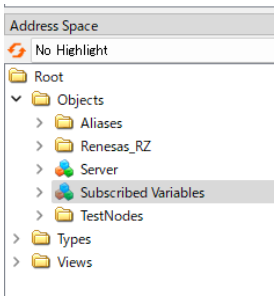
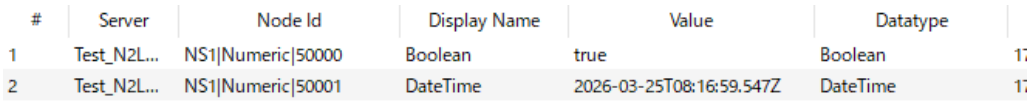


Figure 9.30. “Subscribed_Variables” Node

Checking this Node in the Data Access View, we can confirm that the values are updated periodically and that it is successfully subscribed.



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

Figure 9.31. Check the added Node Values

9.5 Configuration for launching FSP Smart Configurator from EWARM

Please follow the steps below to configure the settings.

1. Click “Tools -> Configure Tools…” in EWARM.
2. Select “New” and enter the following information for each field.
   - Menu Text: FSP Smart Configurator
   - Command: $RASC_EXE_PATH$
   - Argument: --compiler IAR configuration.xml
   - Initial Directory: $PROJ_DIR$

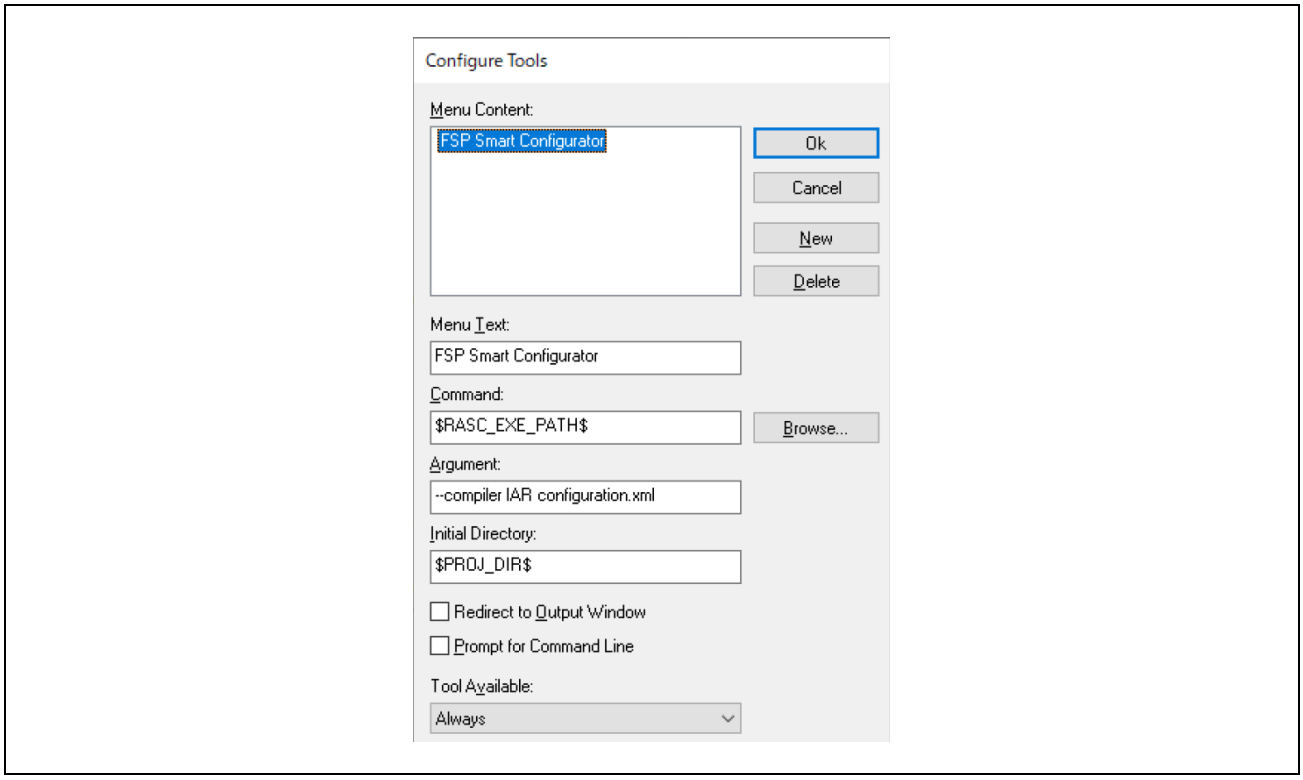


Figure 9.32. Settings “Configure Tools”

3. Open the buildinfo.ipcf file and enter the “FSP Smart Configurator” path in the “RASC_EXE_PATH” field.

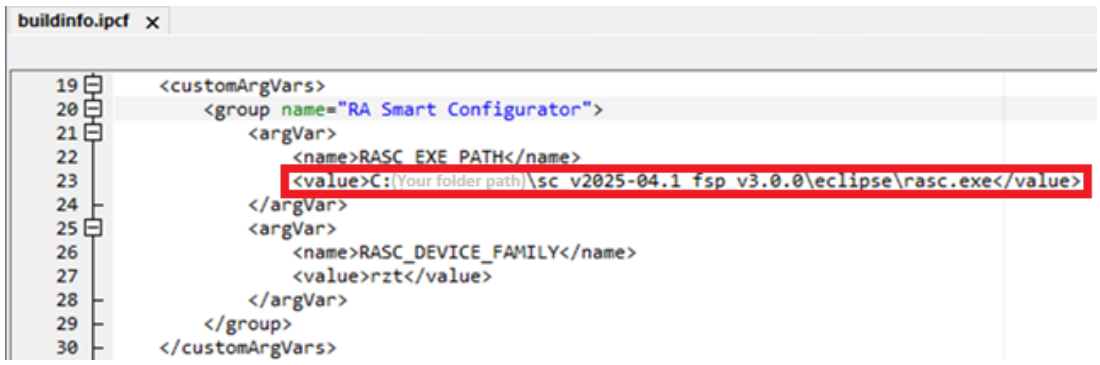


Figure 9.33. buildinfo.ipcf

9.6 How to Change Various Settings in the Sample Software

The following outlines the procedure for changing various settings in the sample software from their default values.

9.6.1 Settings before building

9.6.1.1 When using e² studio

For dual-core project, select the secondary project and configure its settings.

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

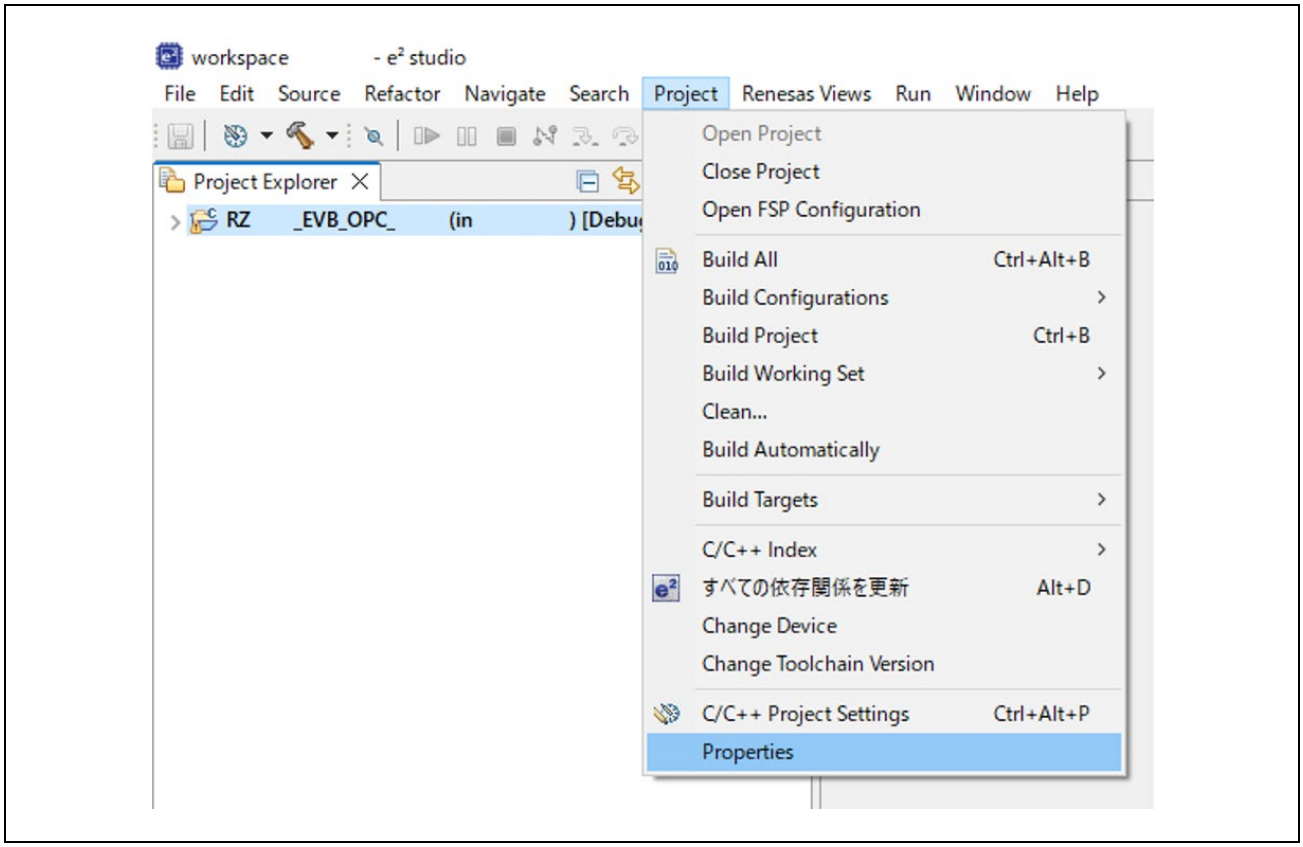


Figure 9.34. Open project properties

Select “GNU C” from the “#Symbols” tab in “C/C++General” > “Paths and Symbols”.

Here you can enable/disable various functions and configure settings. Details are shown in Table 9.2.

Table 9.2. Symbol Settings

Symbol Name	Description	Parameter	Initial Value
IPADR1, 2, 3, 4	IP Address Definition	Any IP address	192, 168, 10, 100
IPADR4_NTPServer	The lower 8 bits of the IP address of the NTP server to connect to. * The upper 24 bits are IPADR1, 2, and 3.	IP address of NTP server	20
ENABLE_PUBSUB_OVER_MQTT	Enable PubSub over MQTT functionality * Currently, only 0: Disabled is supported.	0: Disable 1: Enable	0
ENABLE_PUBSUB_OVER_UADP	Enable PubSub over UADP functionality	0: Disable 1: Enable	1
ENABLE_FREERTOS_PLUS_FAT	Enable FreeRTOS+FAT, TFTP functionality * If disabled, key files embedded in the source code (keyfiles.h) will be used for security functions.	0: Disable 1: Enable	1

To change the IP address, change the values of #IPADR1, 2, 3, and 4 here.

After changing the various settings, click “Apply and Close” to apply the settings. Click "Rebuild Index" in the pop-up dialog.

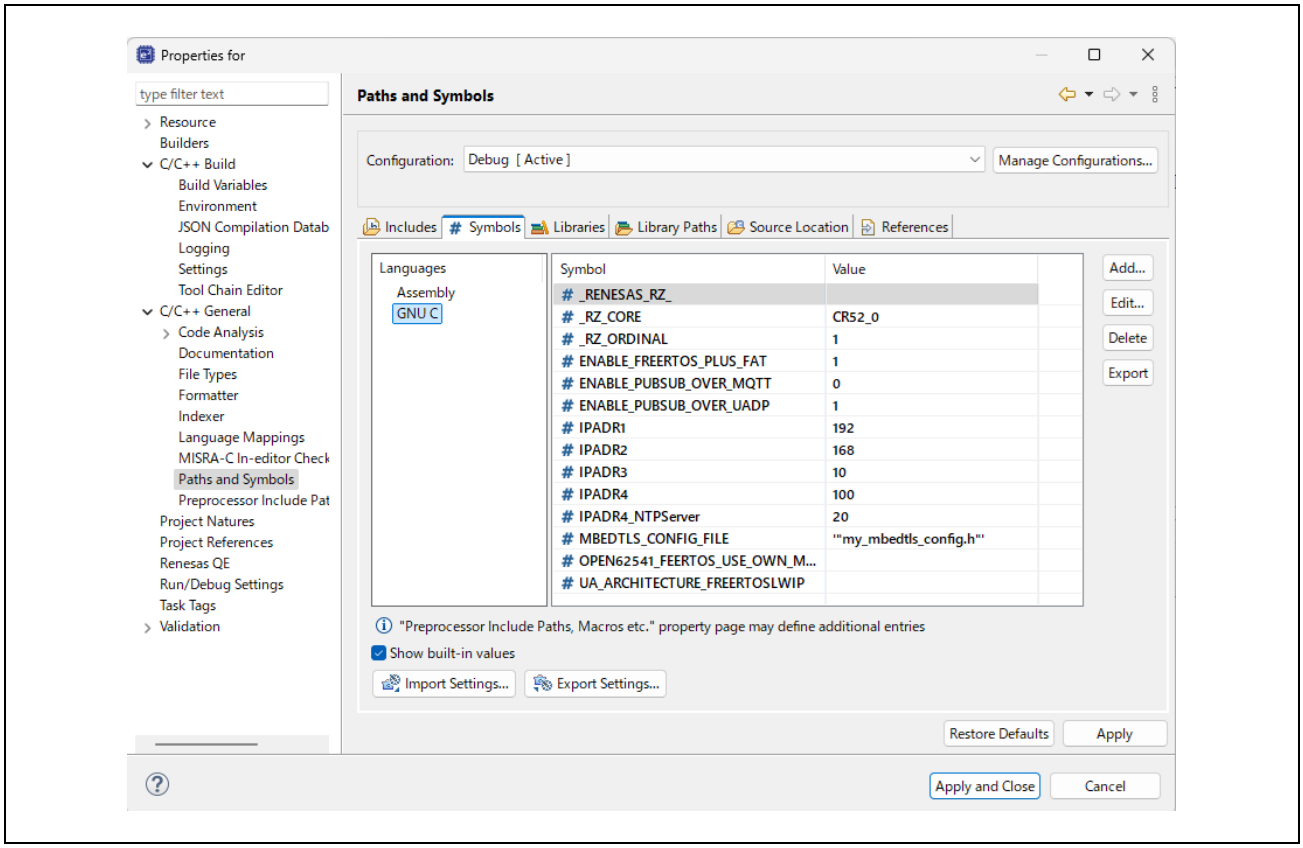


Figure 9.35. Change Build Options

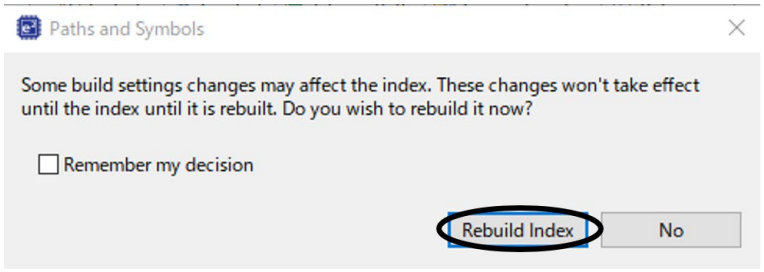


Figure 9.36. Click “Rebuild index”

To change the MAC address, follow the steps below.

Open the Smart Configurator by double-clicking configuration.xml from the tree shown in Figure 9.37. For dual-core project, select the configuration.xml for the secondary project.

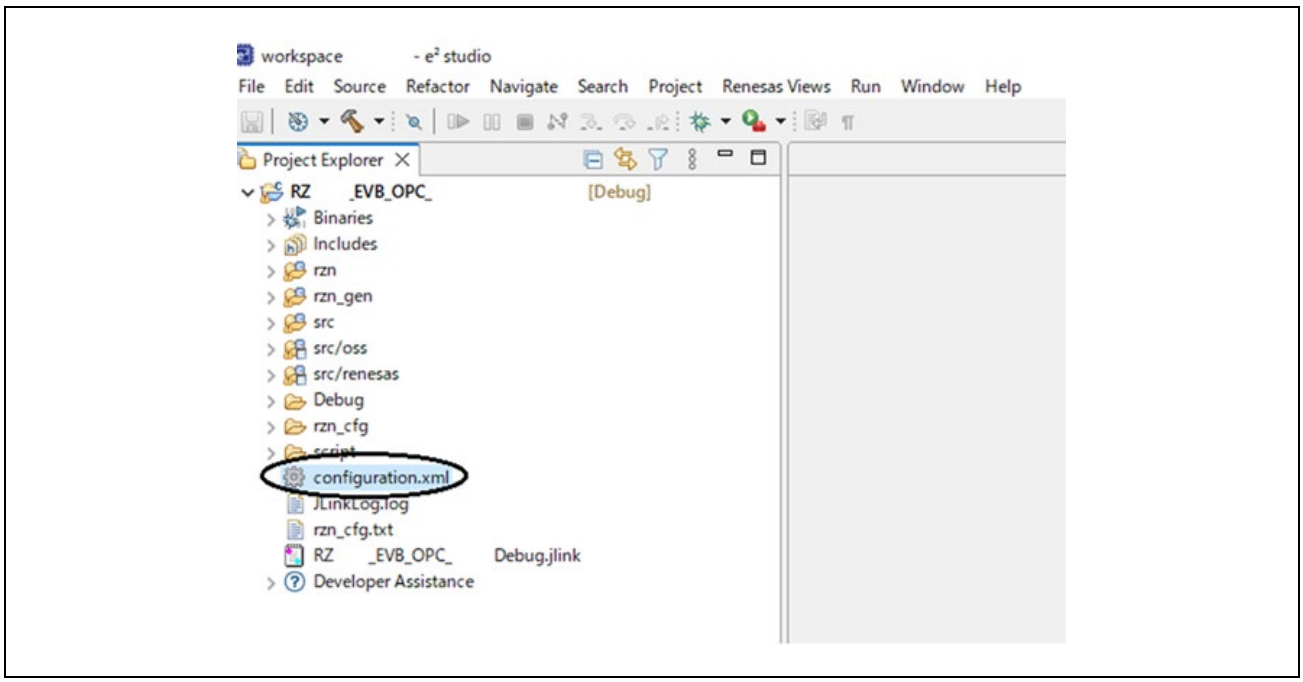


Figure 9.37. Double click configuration.xml

Open the “Stacks” tab and click on “g_ether0 Ethernet”.

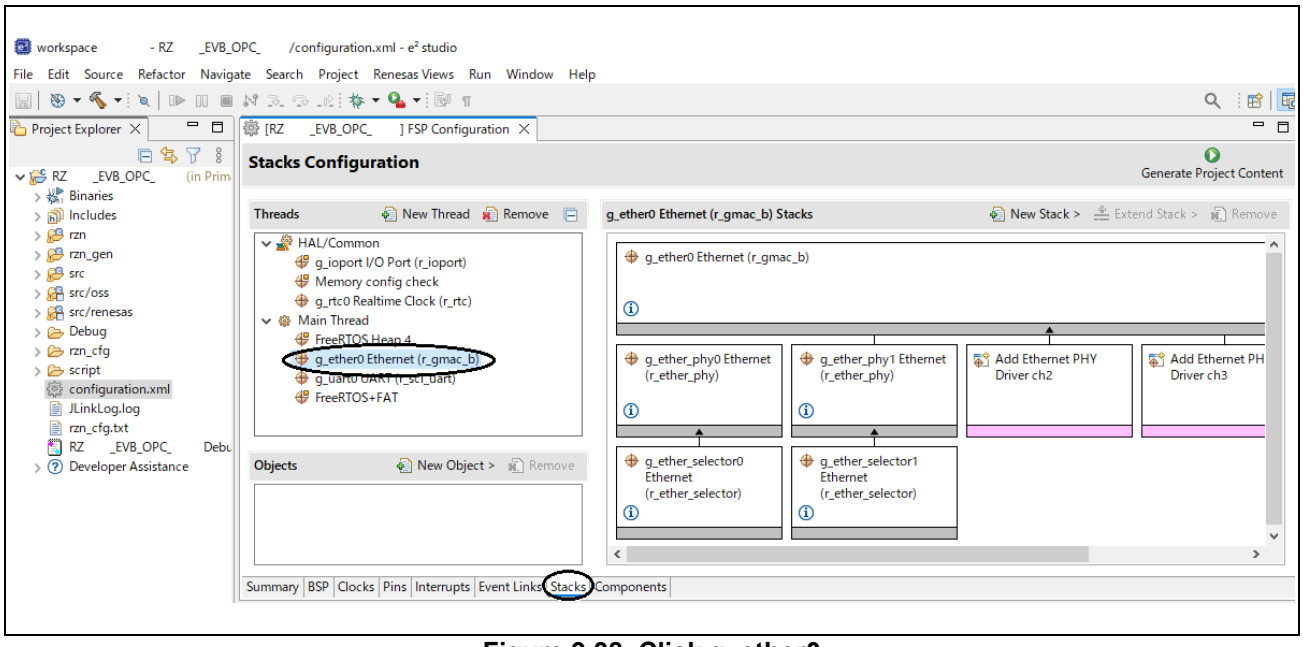


Figure 9.38. Click g_ether0

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

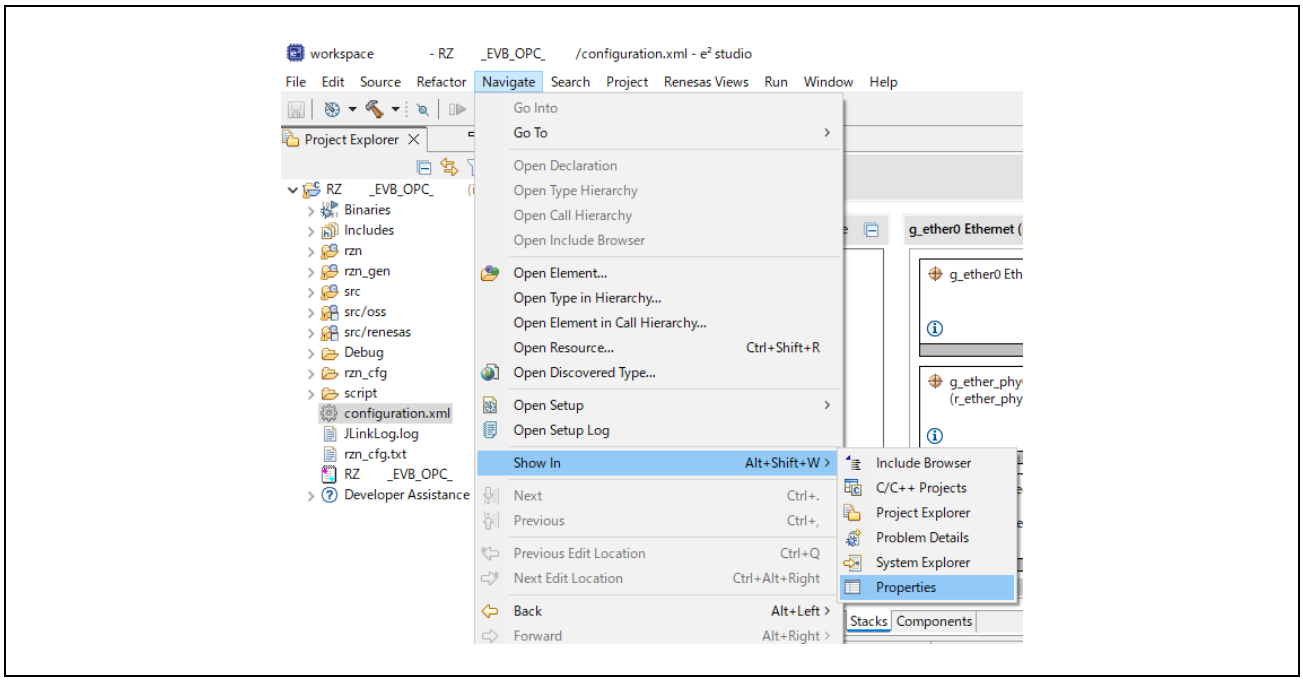


Figure 9.39. Click Properties

Open the Properties and enter General>MAC address0 (e.g. 74:90:50:10:F9:ED).

Click “Generate Project Content” after entering the address.

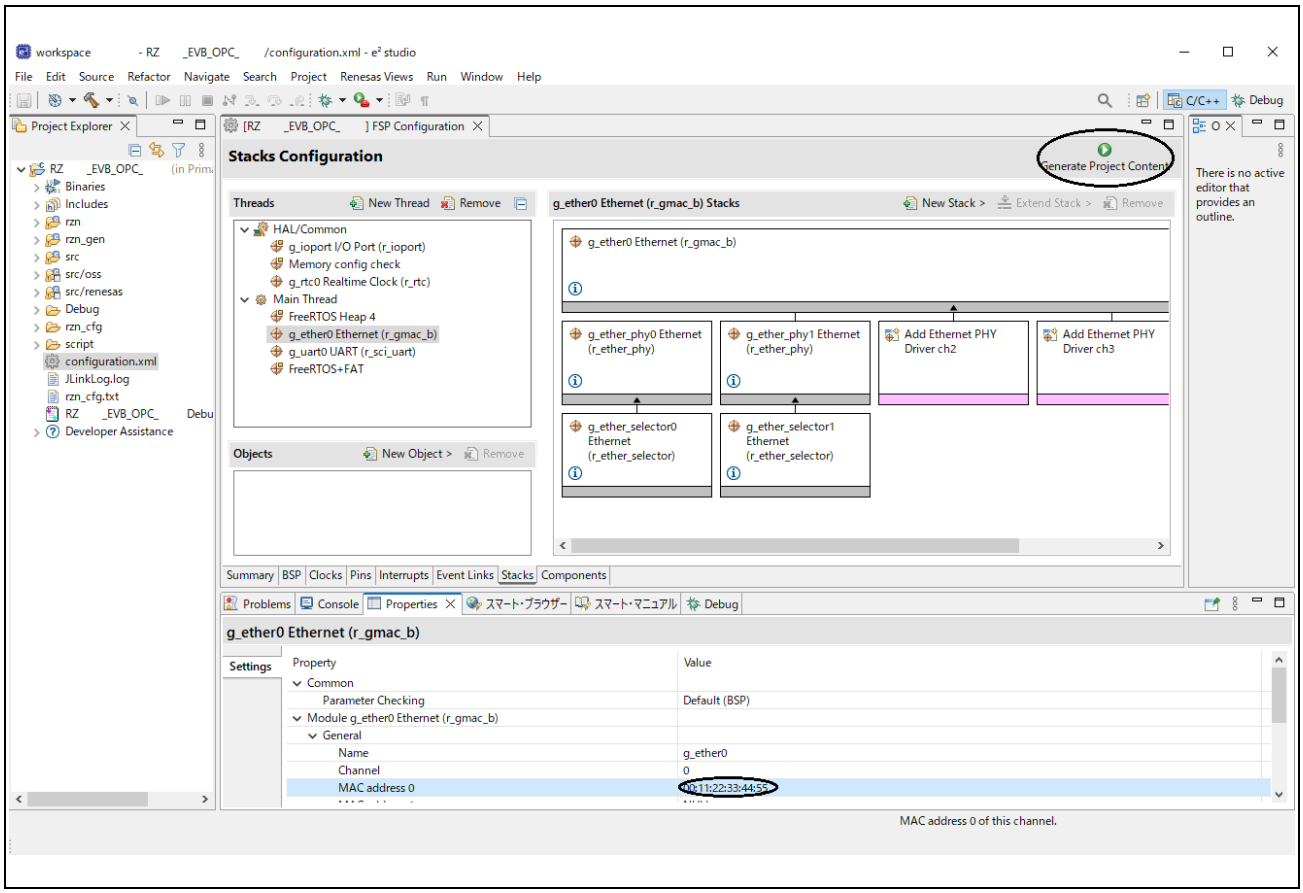


Figure 9.40. Enter MAC address

Additionally, when using the PubSub feature, please modify the definitions in *opc_pubsub_uadp.h*. Note, however, that the settings specified in the *pubsub.cfg* file, as described in chapter 9.6.2, take precedence over these definitions.

Table 9.3. opc_pubsub_uadp.h defines

Defines	Description	Parameter	Initial Value
UADP_LAYOUT	Selecting the Layout for the PubSub Example Configuration	UADP_FIXED_LAYOUT: Periodic Fixed Layout UADP_DYNAMIC_LAYOUT: Dynamic Layout	UADP_FIXED_LAYOUT
ENABLE_PUBSUB_EXAMPLE_SETTING	Enabling the PubSub Example Configuration	PUBSUB_EXAMPLE_DISABLE: Disable PUBSUB_EXAMPLE_ENABLE: Enable, No encryption PUBSUB_EXAMPLE_ENABLE_AES128: Enable, Encrypted (AES128-CTR) PUBSUB_EXAMPLE_ENABLE_AES256: Enable, Encrypted (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

*For dual-core project, select the secondary project and configure its settings.

Select the project name in the Workspace window, then open “Options...” in the Project menu.

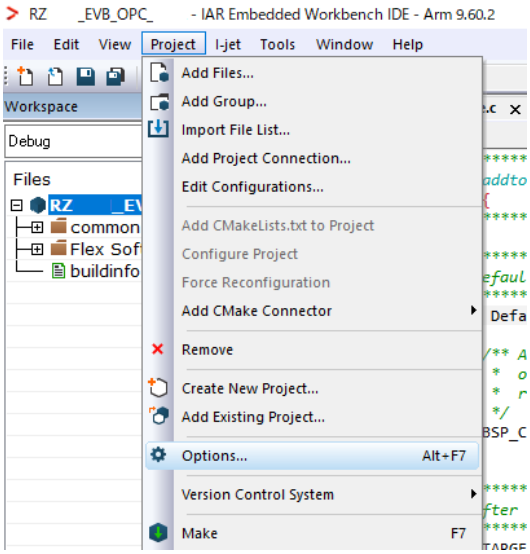


Figure 9.42. Open Options

Various functions can be enabled/disabled and configured in “Defined symbols” in “C/C++ Compiler” > “Preprocessor”. Details are shown in Table 9.4.

Table 9.4. Symbol Settings

Symbol Name	Description	Parameter	Initial Value
IPADR1, 2, 3, 4	IP Address Definition	Any IP address	192, 168, 10, 100
IPADR4_NTPServer	The lower 8 bits of the IP address of the NTP server to connect to. * The upper 24 bits are IPADR1, 2, and 3.	IP address of NTP server	20
ENABLE_PUBSUB_OVER_MQTT	Enable PubSub over MQTT functionality * Currently, only 0: Disabled is supported.	0: Disable 1: Enable	0
ENABLE_PUBSUB_OVER_UADP	Enable PubSub over UADP functionality	0: Disable 1: Enable	1
ENABLE_FREERTOS_PLUS_FAT	Enable FreeRTOS+FAT, TFTP functionality * If disabled, key files embedded in the source code (keyfiles.h) will be used for security functions.	0: Disable 1: Enable	1

To change the IP address, change the values of #IPADR1, 2, 3, and 4 here.

After changing various settings, click OK.

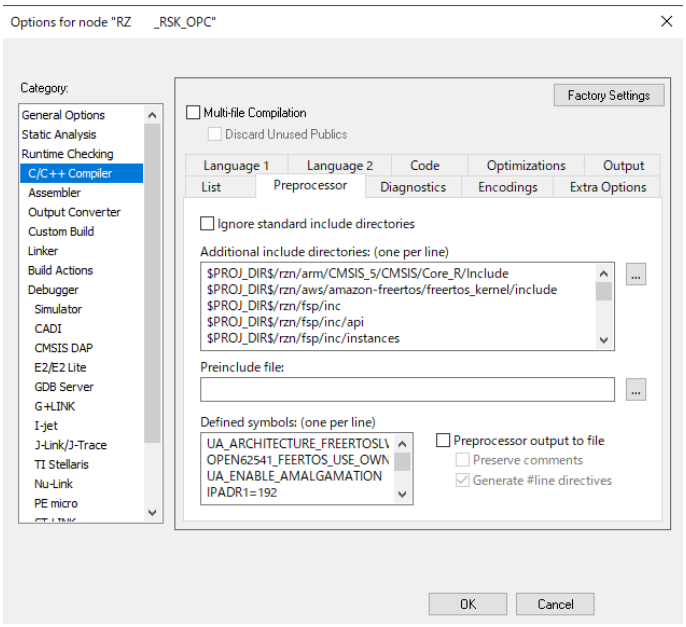


Figure 9.43. Change Build Options

If you change the MAC address, follow the steps below.

In the FSP Smart Configurator, open the Stacks tab and click on “g_ether0 Ethernet” to select it.

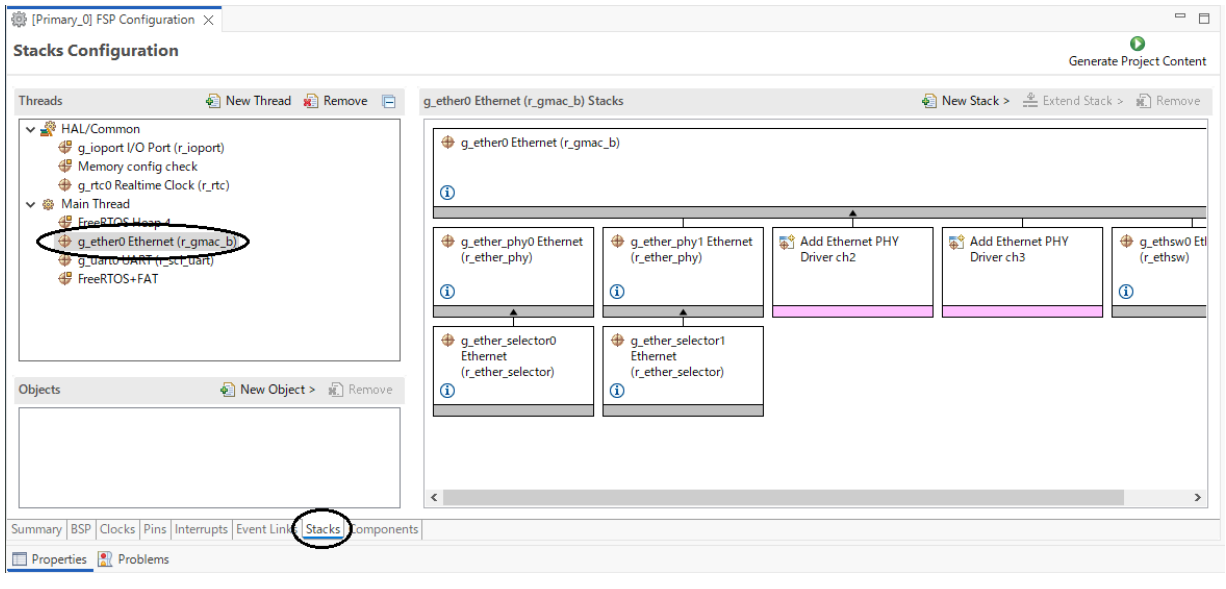


Figure 9.44. Click g_ether0

Open the Properties tab and enter “General”>”MAC address0”.

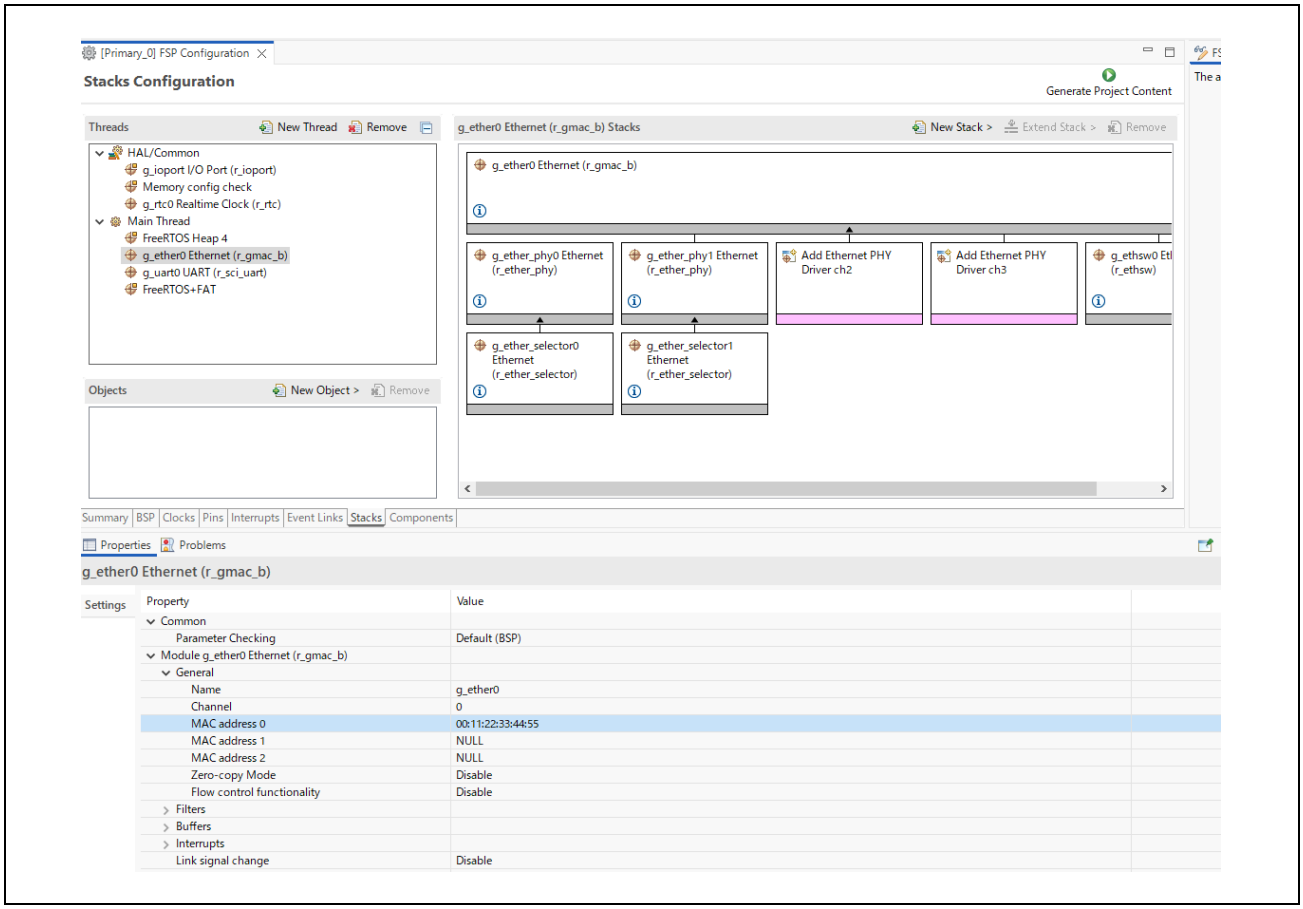


Figure 9.45. Enter MAC address

After the MAC address has been set, click the "Generate Project Content" button to generate the code.

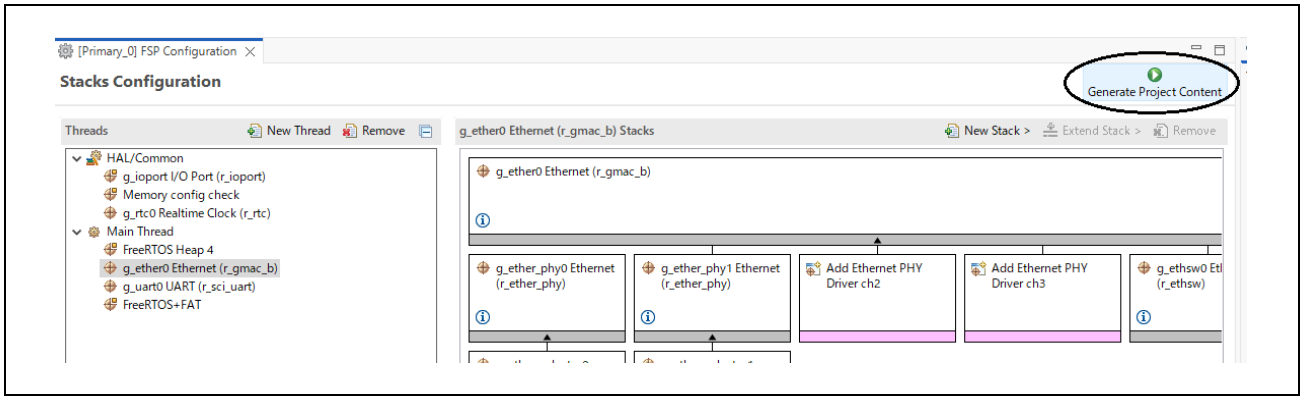


Figure 9.46. Generate Project Content

Additionally, when using the PubSub feature, please modify the definitions in *opc_pubsub_uadp.h*. Note, however, that the settings specified in the *pubsub.cfg* file, as described in chapter 9.6.2, take precedence over these definitions.

Table 9.5. opc_pubsub_uadp.h defines

Defines	Description	Parameter	Initial Value
UADP_LAYOUT	Selecting the Layout for the PubSub Example Configuration	UADP_FIXED_LAYOUT: Periodic Fixed Layout UADP_DYNAMIC_LAYOUT: Dynamic Layout	UADP_FIXED_LAYOUT
ENABLE_PUBSUB_EXAMPLE_SETTING	Enabling the PubSub Example Configuration	PUBSUB_EXAMPLE_DISABLE: Disable PUBSUB_EXAMPLE_ENABLE: Enable, No encryption PUBSUB_EXAMPLE_ENABLE_AES128: Enable, Encrypted (AES128-CTR) PUBSUB_EXAMPLE_ENABLE_AES256: Enable, Encrypted (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

If you need to dynamically switch functionality after the software has been built by using the configuration files (“pubsub.cfg” and “ip.cfg”), modify the corresponding files stored on the USB flash drive and then restart the evaluation board.

To change the IP address settings, set IP_MODE in “ip.cfg” to FILE, and specify the desired values for IP and NTP_SERVER_IP. If an invalid value is read from the file, or if the file does not exist, the default IP address configured at build time will be used.

Note that when the IP address is changed using this feature, a new server certificate is generated during startup. As a result, the startup time may be longer than usual.

```
[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
```

To dynamically change the PubSub configuration, modify the EXAMPLE and LAYOUT parameters in “pubsub.cfg” to the desired values. The settings specified in “pubsub.cfg” take precedence over the macro definitions in the source code described in chapter 9.6.1. If an invalid value is read from the file, or if the file does not exist, the default PubSub configuration specified at build time will be used.

```
[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

Please refer to the guide in this chapter when using FSP v4.1.0 or later for RZ/T2M or RZ/N2L, or when using FSP v4.2.0 or later for RZ/T2H or RZ/N2H.

1. When generating FSP source code, the following linker script files and the shell script files located in (Project Location)/script/ will be overwritten. Please use the original files prior to overwriting for the build. These original files are also provided in “AttachedFiles/original_scripts.zip” in the package.

   - 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. For e² studio projects targeting the CR52 core, please change the following settings in Properties before building after FSP code generation. However, for dual-core projects, please change these settings only for the secondary project.

   Use newlib-nano: Unchecked

   Do not use syscalls: Checked

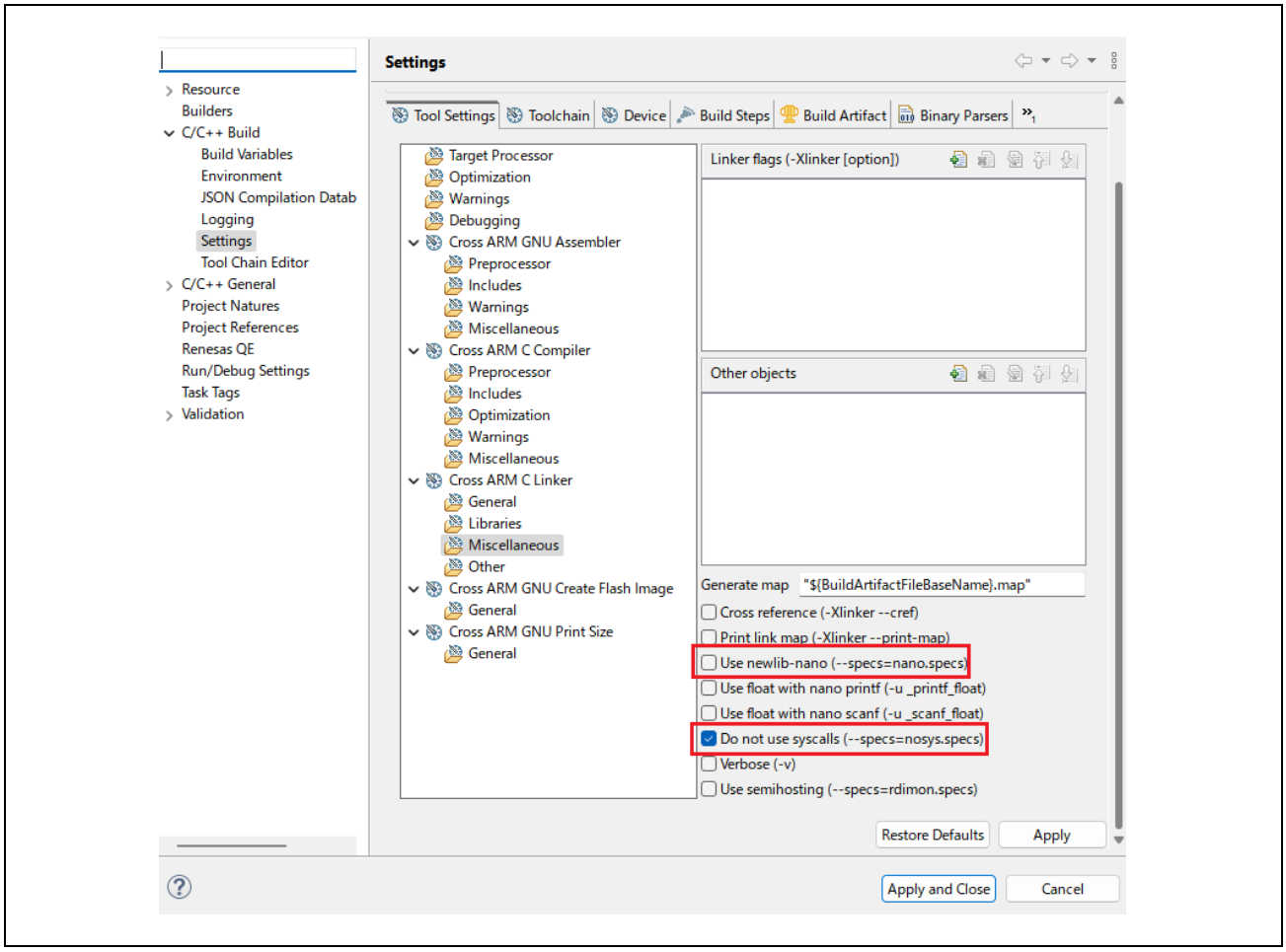


Figure 9.48 Properties

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr.15, 2026	-	Integrate the Application Note for RZ/T2M, N2L (R01AN7048xxxxxx) and the Application Note for RZ/T2H, N2H (R01AN7755xxxxxx).
		-	Supported RZ FSP v4.0.0
		p.80	Create a new chapter 8 Software Specifications
1.10	May.29, 2026	p.6	Added notes regarding OPC UA certification. Changed X.509 user authentication to unsupported.
		p.8	Removed descriptions related to files for X.509 user authentication.
		p.12	Updated FSP version for RZ/T2H and N2H. To reflect the following fix in r_gmac_b: • Fixed an issue where operation would stop after receiving or transmitting 2^{31} bytes of data via Ethernet.
		p.74	Removed descriptions related to X.509 user authentication.
		p.84	Updated specifications related to SystemReset.
		p.122	Updated configuration values for the ccmake command.
		p.126	Improved the operation verification procedure for PubSub configuration using the Method in section 9.4.
		p.145	Chapter 9.7 Added notes on using newer FSP version
1.20	Aug.28, 2026	p.6	For the supported features, the Application Profile has been updated to the “Embedded 2017 UA Server" profile”, and the “X509 Certificate Server Facet” has been added to the supported User Tokens.
		p.8	Changed the file management structure so that files on the USB flash drive are organized into separate folders according to their intended purpose.
		p.81, 82	Updated the versions of the OSS components used in this software as follows: Mbed TLS: v3.6.4 → v3.6.6 open62541: v1.3.15 → v1.3.18
		p.86	Updated the startup sequence diagram.
		p.117	Updated chapter 9.1 to disable the open62541 Amalgamation option. The related descriptions throughout the document have also been revised accordingly.
		p.144	Added a new section, chapter 9.6.2, to describe the configuration files.

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

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/.