RZ/N1D, RZ/N1S, RZ/N1L Group

Quick Start Guide: EtherCAT Slave Software

R01AN4239EJ0110
Rev.1.10
Mar 29, 2019

Introduction

This document explains test procedures for EtherCAT® slave functionalities with the adapted EtherCAT Stack Code for Renesas RZ/N1 platform. This describes steps to evaluate slave behavior and stack features using TwinCAT®Master Configuration tool.

Target Device

RZ/N1D, RZ/N1S, RZ/N1L

Contents

1. Overview

This document describes how to run EtherCAT on the RZ/N1 Series. It is possible either to run a standalone variant only using one core, the Cortex®-M3-core, or to use two separate cores in the case of the RZ/N1D communicating via Core To Core. Both cores feature the GOAL (Generic OS Abstraction Layer) which handles the communication of the cores and provides basic functionality e.g. timer handling.

The EtherCAT protocol runs on Cortex-M3-core in both the standalone and the Core To Core variant. Its task is the communication with other operators, therefore the alias name of the Cortex-M3-core is communication core (CC) in this document.

In the Core To Core variant the user application is executed on the Linux based Cortex-A7-core. This core is also named as application core (AC). In the standalone variant the user application is running on the communication core, either.

EtherCAT(Ethernet for Control Automation Technology) is an Ethernet based fieldbus system, developed by Beckhoff Automation. Development of EtherCAT was to apply Ethernet for automation applications (e.g. for motion control, I/O, sensors) requiring short data update times with low communication jitter and reduced hardware costs.

Tool to generate EtherCAT Slave Stack Code (SSC Tool) is available to the ETG members free of charge. This can be downloaded from the ETG website. SSC tool can be used to generate customized stack, device description files (ESI) and individual source code documentation to suit the developer's own needs.

Core EtherCAT stack supports basic slave functionalities and mailbox protocols such as CoE, FoE, etc.

1.1 Purpose

This document explains test procedures for EtherCAT slave functionalities with the adapted EtherCAT Stack Code for Renesas RZ/N1 platform. This describes steps to evaluate slave behavior and stack features using TwinCAT Master Configuration tool.

1.2 Scope

The document scope is limited to in explaining the usage of SSC tool for EtherCAT Slave Stack code generation, and testing of its behavior against TwinCAT master and Test Application.

1.3 Abbreviations/Definitions

Table 1. Abbreviations/Definitions

Index	Abbreviations /Definitions	Description
1	CoE	CAN application protocol over EtherCAT
2	EEPROM	Electrically Erasable Programmable Read-Only Memory
3	ESC	EtherCAT Slave Controller
4	ESI	EtherCAT Slave Information
5	FoE	File Access Over EtherCAT
6	I2C	Inter-Integrated Circuit
7	MB	Mail Box
8	PDO	Process Data Object
9	SSC	Slave Stack Code
10	EoE	Ethernet Over EtherCAT

1.4 Reference

Technical information about EtherCAT is available via ETG member site, and information about RZ/N1 is available via Renesas.

Table 2. Technical Inputs

Index	Technical Inputs
1	EtherCAT Slave Stack Code - Application Note.pdf
2	ETG2200_V3i0i1_G_R_SlaveImplementationGuide.pdf
3	r01uh0750ejxxxx-rzn1-introduction.pdf
4	r01uh0753ejxxxx-rzn1-ethernet.pdf
5	RZ_N1D_DB_Board_SCHEMATIC.Vxxx.pdf
6	RZ_N1D_DB_Board_Setup_Notes.Vxxx.pdf
7	RZ_N1S_DB_Board_SCHEMATIC.Vxxx.pdf
8	RZ_N1S_DB_Board_Setup_Notes.Vxxx.pdf
9	RZ_N1L_DB_Board_SCHEMATIC.Vxxx.pdf
10	RZ_N1L_DB_Board_Setup_Notes.Vxxx.pdf
11	RZ_N1_EB_Board_SCHEMATIC.Vxxx.pdf
12	RZ_N1_EB_Board_Setup_Notes.Vxxx.pdf

2. Features

The EtherCAT slave stack code generated by SSC Tool provides the functionality of EtherCAT slave controller.

It features:

- ESM (EtherCAT State Machine)
- mailbox protocols:
  - CoE (CAN application protocol over EtherCAT)
- synchronization Modes:
  - Free Run
  - Sync Manager Synchronization
  - DC Synchronization

3. Project Setup

3.1 Requirements

Table 3. Requirements

Item	Description
Board	Renesas Electronics • RZ/N1D-DB Development Board or • RZ/N1S-DB Development Board or • RZ/N1L-DB Development Board If 3ports of ESC is needed on RZ/N1D or RZ/N1S, requires • RZ/N1-EB Expansion Board
IDE	IAR Systems Embedded Workbench® for ARM Version 8.22.1 or later
Emulator	IAR Systems I-jet
SSC Tool	Slave Stack Code (SSC) Tool Version 5.11 or 5.12
Software PLC	Beckhoff Automation TwinCAT3

3.2 Hardware

Please take care of following the setup guidelines for the RZ/N1 Demo Board from the Linux and U-boot documentation - *RZN1x-Quick-Start-Guide.pdf*

Please follow these initial steps to setup the UART and DFU connection.

1. Connect the board to a Linux PC via the UART and the DFU interface. After the driver for the device has been installed, four additional serial ports will exist. The board uses the 3rd port for serial output. On Linux PCs, if you have no other serial-over-USB devices attached, this is /dev/ttyUSB2.
2. Open a serial terminal on the Linux PC e.g. with
   cu -e -o -115200 -l /dev/XXX
   Replace the "XXX" with the serial device where the UART of the board is connected to.

Device connection between EtherCAT master and slave is shown below.

Table 4. EtherCAT connection for RZ/N1D-DB

Function	Connector	Description
EtherCAT Port A	CN4	Connect to PC with TwinCAT3
EtherCAT Port B	CN1	No connection or connect to other EtherCAT Slave

Table 5. EtherCAT connection for RZ/N1S-DB

Function	Connector	Description
EtherCAT Port A	CN5	Connect to PC with TwinCAT3
EtherCAT Port B	CN1	No connection or connect to other EtherCAT Slave

Table 6. EtherCAT connection for RZ/N1-EB

Function	Connector	Description
EtherCAT Port C	J24	No connection or connect to other EtherCAT Slave

Table 7. EtherCAT connection for RZ/N1L-DB

Function	Connector	Description
EtherCAT Port A	CN5	Connect to PC with TwinCAT3
EtherCAT Port B	CN1	No connection or connect to other EtherCAT Slave

If RZ/N1L-DB or RZ/N1-EB is used, it's necessary to change Switch or Jumper setting on board as shown below.

Table 8. SW/Jumper Setting for RZ/N1L-DB

SW/Jumper	Default Setting	Setting for EtherCAT	Description
SW5-2	OFF	ON	I2C Cat

Table 9. SW/Jumper Setting for RZ/N1-EB

SW/Jumper	Default Setting	Setting for EtherCAT	Description
Jumper CN15 Jumper CN16	MDC1 MDIO1	MDC2 MDIO2	MDIO2 connected to PHY2 and PHY3

3.3 Sample Application

Several sample applications are provided for EtherCAT device stack They show how to set up and use the stack. The following examples can be found in the folder goal/appl/goal_ecat/.

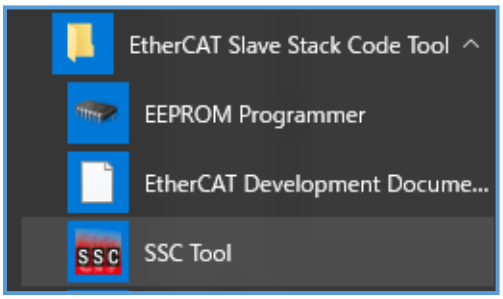
- 00_rpc_cc – communication core (Core To Core variant only)
- 01_io_data – simple IO application (Standalone variant, Core To Core variant for AC)

3.4 Configuring the sample application

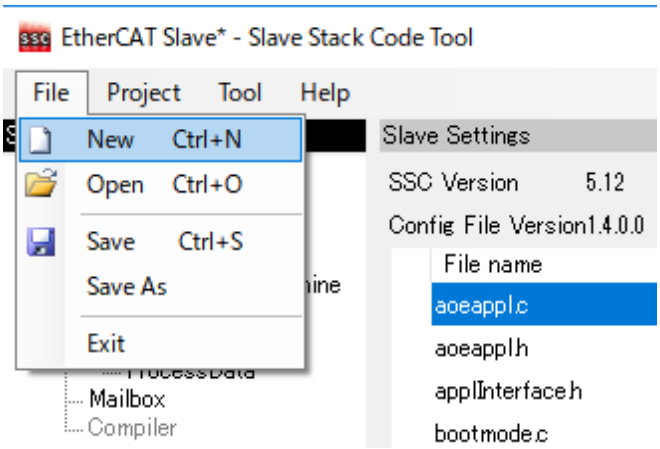
3.4.1 Generating the Slave Stack Code

SSC Tool is used for generating the slave stack code.

1. Start the SSC Tool from the Window Start menu.

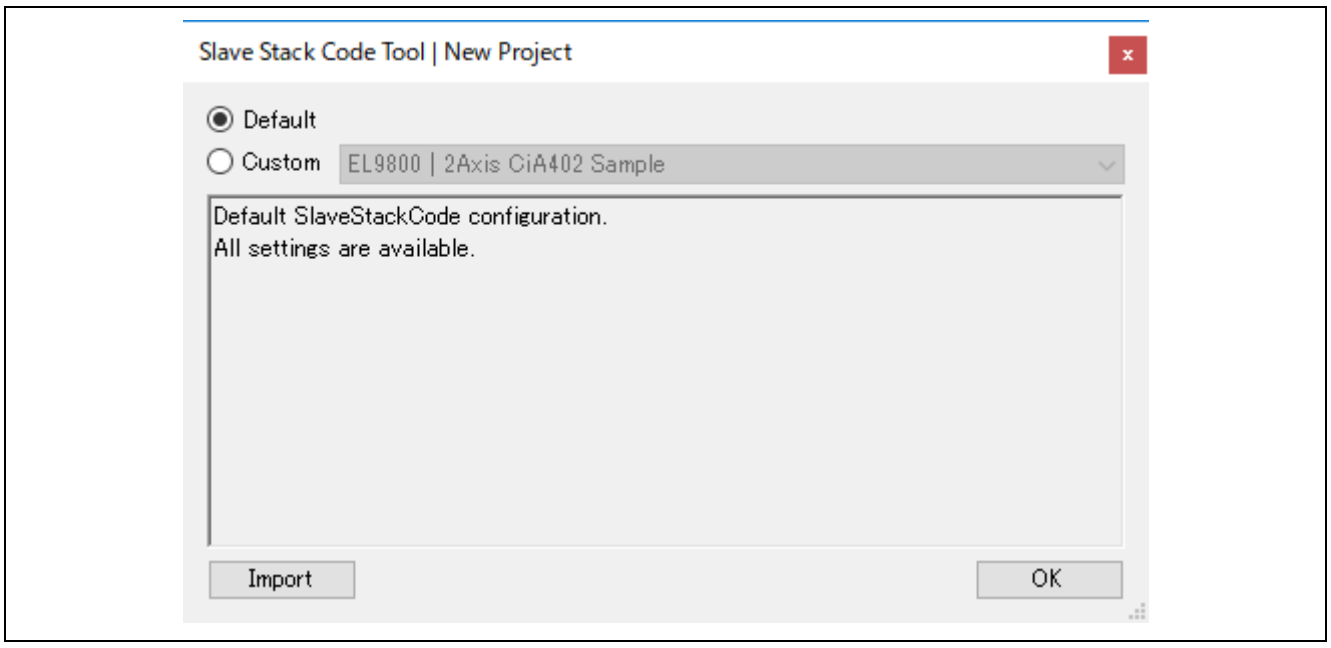


2. Select File > New.

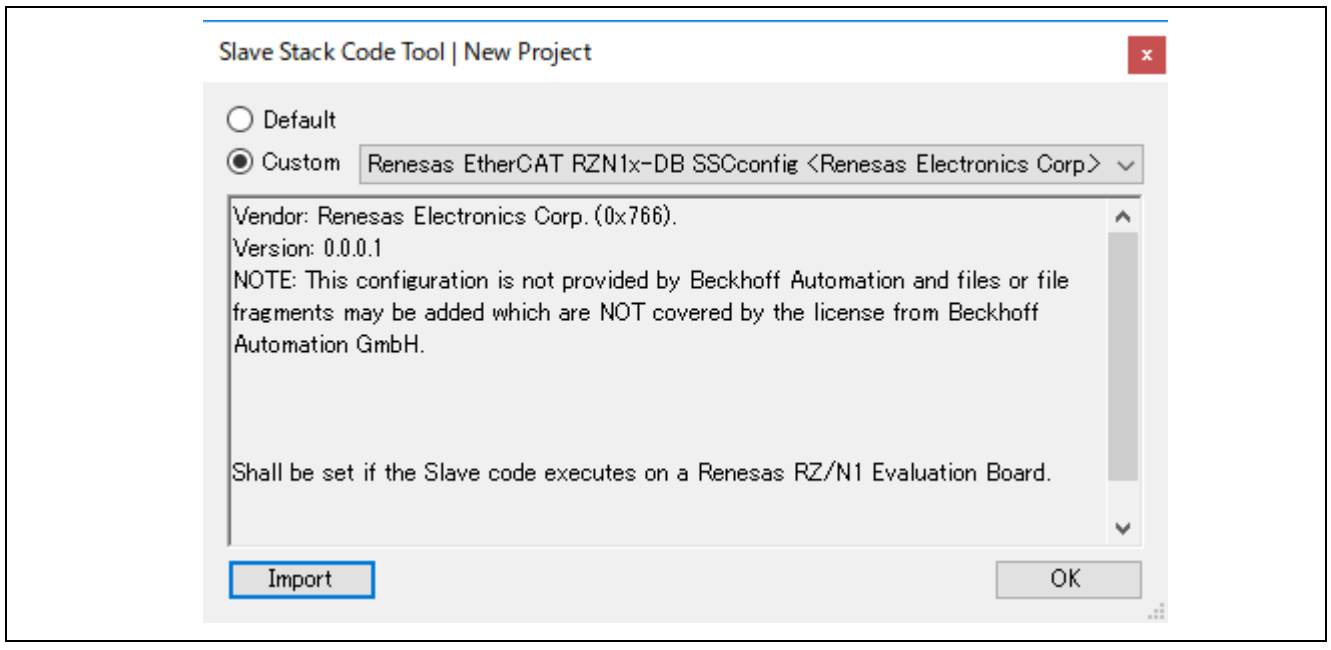


3. Click the [Import] button and select the SSC Tool configuration file at,

"goal\protos\ecat\ssc\Renesas EtherCAT RZN1x-DB SSCconfig.xml"

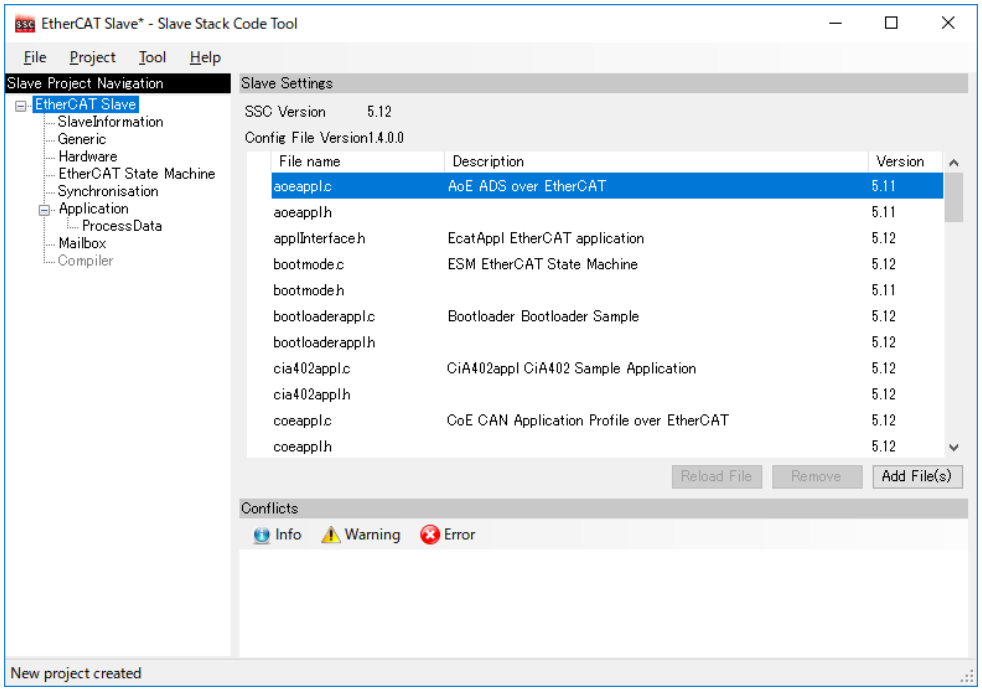


4. After the configuration file is read, the window changes as follows:

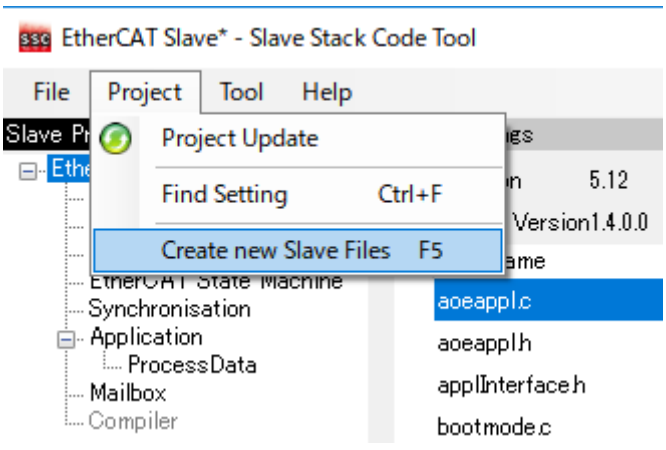


Once the configuration file is read, it is registered in Custom and is selectable from the drop-down list.

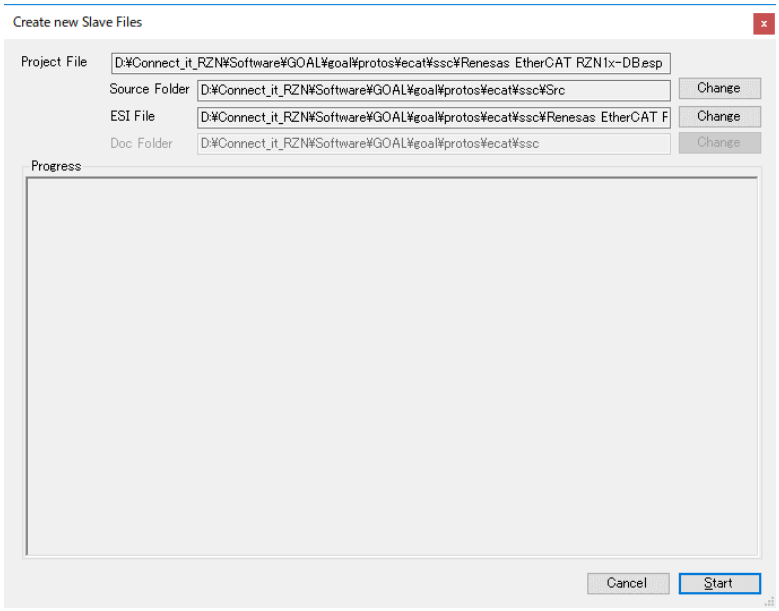
5. After clicking the [OK] button, the following window opens.



6. Select Project > Create new Slave Files.

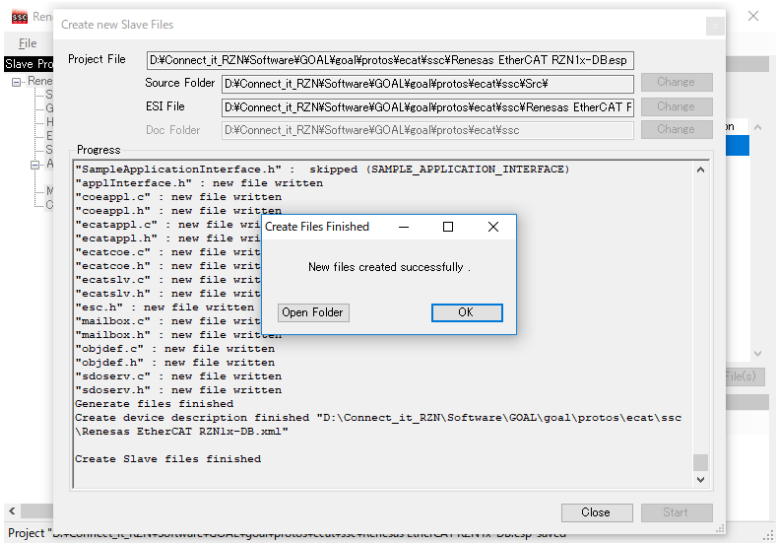


7. Click the [Start] button to start creating the EtherCAT Slave Stack Code.



8. When a message “New file created successfully” appears, the creation processing is completed and the source files are located in the following folder.

“goal\protos\ecat\ssc\Src”



3.5 Running the sample application

The RZ/N1D and RZ/N1S use the U-Boot bootloader for initial setup of the hardware and loading of the Cortex-M3 firmware. Additionally the RZ/N1D U-Boot bootloader is used for booting the Linux Kernel. The RZ/N1L is working without any bootloader. This chapter describes how to install the management software on the flash of the board. If no bootloader was yet installed on the RZ/N1D and RZ/N1S please refer to the Linux documentation - Quick Start Guide for U-Boot and Linux - *RZN1x-Quick-Start-Guide.pdf.*

There are many similarities between the derivatives of the RZ/N1 series but some minor difference, too. Therefore here is a more detailed explanation how to run a sample application on each.

All standalone projects and the CC project of the Core To Core variant contain different workspaces for each board variant. The project workspaces ending on *_eb contain the configuration for the CPU Board together with the extension board (4 switch ports). The other project workspaces contain the configuration for working with the CPU Board only.

3.5.1 Standalone Variant – RZ/N1D and RZ/N1S

It is possible to load the code via debugger into RAM, which is a very fast approach to test the user application, or to flash the Cortex-M3 core. In both cases the applications are located at goal\projects\goal_ecat and must be built using IAR Embedded Workbench.

- *N1D-DB - goal\projects\goal_ecat\01_io_data\iar\renesas\rzn1d_demo_board\rzn1d_demo_board.eww*
- *N1D-EB - goal\projects\goal_ecat\01_io_data\iar\renesas\rzn1d_demo_board_eb\rzn1d_demo_board_eb.eww*
- *N1S-DB - goal\projects\goal_ecat\01_io_data\iar\renesas\rzn1s_demo_board\rzn1s_demo_board.eww*

3.5.2 Loading application into RAM via IAR Embedded Workbench

To compile a project, follow these steps:

1. Start the IAR Embedded Workbench
2. Open a project via Open/Workspace
3. Go to the workspace folder and open it:
4. In case the CPU board is used together with the extension board, please ensure to select the correct IAR-project. Compile the project via “Project/Compile” or “Project/Rebuild all”.
5. Power up the device
6. Open a serial terminal e.g. PuTTY and connect it to the serial interface where the UART is connected to (also see step 1 in section 3.2 for selecting the correct device). The following settings must be configured for the connection:

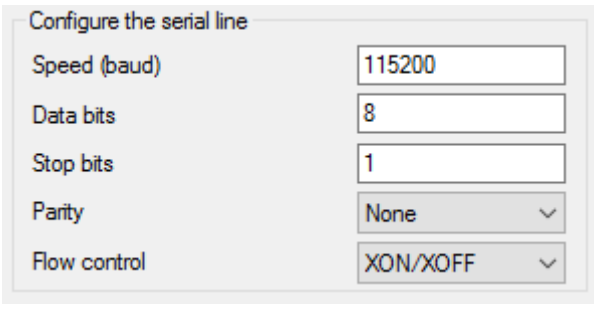


Figure 3.1: Serial Terminal settings RZ/N1D and RZ/N1S

7. Press any key on your keyboard to interrupt the bootloader.
8. Ensure to configure the U-boot boot command to release the Cortex-M3 core after reset. This is done by the command:
   setenv bootcmd "mw 0x04000004 1 && rzn1_start_cm3 && loop 0 1"
   followed by
   saveenv
   and reset the board.
9. Connect the debugger to the system via the “Download and Debug” button of the IAR Embedded Workbench.
10. After the Debug view opened, click on the “Go” button.

3.5.3 Loading application into flash via dfu-util

The board uses the u-boot bootloader for initial setup of the hardware and loading of the Cortex-M3 core firmware. This chapter describes how to install the compiled management software on the flash of the board. If no bootloader was yet installed on the board please refer to the Linux documentation – Quick Start Guide for U-Boot and Linux - *RZN1x-Quick-Start-Guide.pdf.*

The following steps describe the installation of management software:

1. Connect a Linux PC to the board according to section 3.2.
2. Power up the board.
3. Hit any key to stop the autoboot of the u-boot
4. Type “*dfu*” in the serial terminal of the board and hit enter.
5. On a Linux terminal start the command
   sudo dfu-util -a'sf_cm3' -D FIRMWARE.bin
   Replace FIRMWARE.bin with the file name of the software to install. The binary is placed at the subfolder Debug-RAM\Exe of the IAR project folder.
6. When the download process is complete, press Ctrl+C on u-boot.
7. If the autoboot command was already configured, go to step 10.
8. Set the autoboot command in the u-boot:
   setenv bootcmd "sf probe && sf read 0x4000000 d0000 80000 && rzn1_start_cm3 && loop 0 1"
9. Save the command to the flash: *saveenv*
10. Reset the device

3.5.4 Standalone Variant – RZ/N1L

The RZ/N1L does not use any bootloaders. If any application is stored in flash, it will be started automatically. Both, loading into RAM and flash can be done using IAR Embedded Workbench .

- *N1L-DB - goal\projects\goal_ecat\01_io_data\iar\ renesas\rzn1l_demo_board\rzn1l_demo_board.eww*

1. Start the IAR Embedded Workbench
2. Open a project via Open/Workspace
3. Go to the workspace folder and open it:
4. Compile the project via “Project/Compile” or “Project/Rebuild all”.
5. Power up the device
6. Open a serial terminal e.g. PuTTY and connect it to the serial interface where the UART is connected to (also see step 1 in section 3.2 for selecting the correct device). The following settings must be configured for the connection:

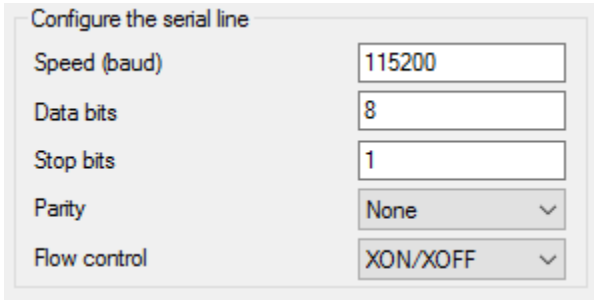


Figure 3.2: Serial Terminal settings RZ/N1L

7. Choose either the Debug-RAM or the Debug-ROM configuration. First it is used for debugging via IAR Embedded Workbench , second is loading the application into the flash.

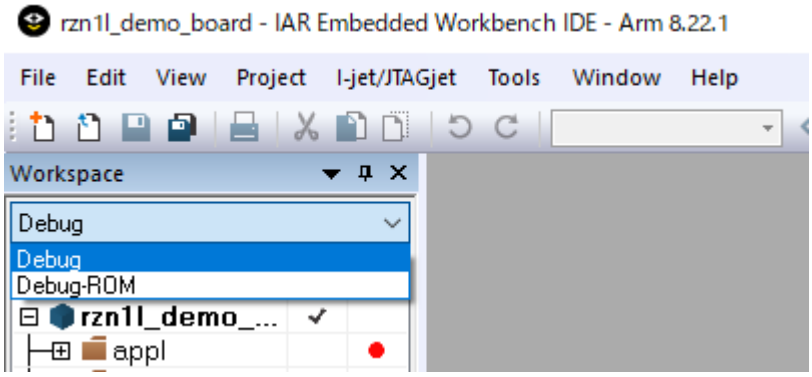


Figure 3.3 : IAR Embedded Workbench Configurations RAM and ROM for RZ/N1L

8. If you have chosen Debug-ROM click on “Download and Debug”. Change reset mode to “Hardware” and press “Stop Debugging”. This is not necessary if you choose the Debug-RAM configuration.

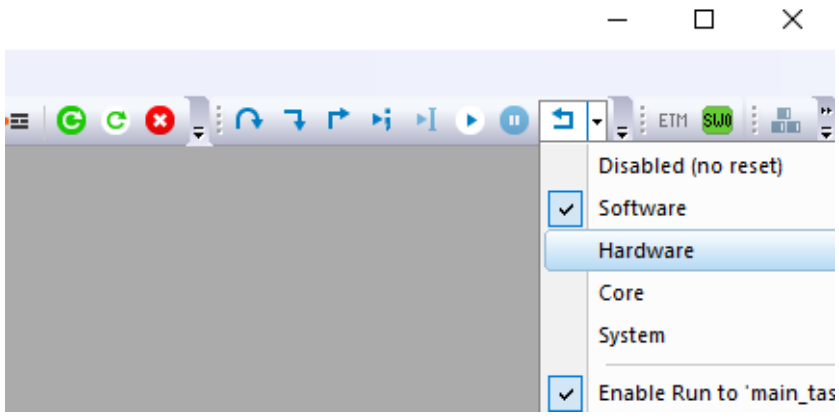


Figure 3.4: Changing Reset mode of RZ/N1L in Debug-ROM configuration

9. Connect the debugger to the system via the “Download and Debug” button of the IAR Embedded Workbench.
10. After the Debug view opened, click on the “Go” button.

3.5.5 Core To Core variant – RZ/N1D (Communication Core)

The binary file for the Cortex-M3 core is located in the board type related IAR Embedded Workbench folder

- *N1D-DB - goal\projects\goal_ecat_rpc_cc\00_goal_rpc_demo\iar\renesas\rzn1d_demo_board\rzn1d_demo_board.eww*
- *N1D-EB - goal\projects\goal_ecat_rpc_cc\00_goal_rpc_demo\iar\renesas\rzn1d_demo_board_eb\rzn1d_demo_board_eb.eww*

Load the binary file to the flash according to the following steps.

1. Connect a Linux PC to the board according to section 3.2
2. Power up the board.
3. Hit any key to stop the autoboot of the U-Boot
4. Type *dfu* in the serial terminal of the board and hit enter.
5. On a Linux terminal start the command
   sudo dfu-util -a"sf_cm3" -D FIRMWARE
   Replace FIRMWARE.bin with the file name of the software to install. The binary is placed at the subfolder Debug-RAM\Exe of the IAR project folder.
6. When the download process is complete, press Ctrl+C on u-boot.
7. If the autoboot command was already configured, go to step 10.
8. Set the autoboot command in the u-boot:
   setenv bootcmd "sf probe && sf read 0x4000000 d0000 80000 && sf read 0x80fe0000 b0000 20000 && sf read 0x80008000 1d0000 f00000 && rzn1_start_cm3 && sleep 4 && bootm 0x80008000 - 0x80fe0000"
9. Save the command to the flash: saveenv
10. Reset the device

3.5.6 Core To Core variant – RZ/N1D (Application Core)

The user application runs on the Linux system of the Cortex-A7. Its binary must be created by GCC and downloaded to the RZ/N1 board manually.

3.5.7 Building and downloading the user application

The following steps describe, how to build a binary and download it to the RZ/N1 board.

1. Navigate with the terminal of a Linux PC to the project of the application core at *goal/projects/goal_ecat_rpc_ac/01_io_data/gcc.*
2. Start the build process by executing the Makefile by typing
   make
3. Select as target platform “rzn_a7_demo_board”.
4. Power up the board and wait till Linux booted successfully.
5. Copy the binary file build/rzn_a7_demo_board/goal_rzn_a7_demo_board.bin to the RZ/N1 board by e.g. secure copy (scp).
6. Start the binary file on the target by typing the commands
   ./goal_rzn_a7_demo_board.bin -i usb0

The GOAL setups the connection to the communication core via core to core and starts the user application. The initialization is done when the log message “GOAL initialized” is printed at the terminal, if logging is activated.

4. Setting up a TwinCAT3

4.1 Copying the ESI Files

Before starting TwinCAT, copy the ESI files that are included in the release folder to TwinCAT destination "\TwinCAT\3.x\Config\IO\EtherCAT"

ESI file for current release available at,

N1D - "goal\protos\ecat\esi\N1D\Renesas EtherCAT RZN1D.xml"

N1S - "goal\protos\ecat\esi\N1S\Renesas EtherCAT RZN1S.xml"

N1L - "goal\protos\ecat\esi\N1L\Renesas EtherCAT RZN1L.xml

4.2 Connecting to TwinCAT3

Start TwinCAT3 by using the procedure described below,

From the start menu, select [Beckhoff] → [TwinCAT3] → [TwinCAT XAE (VS2013)].

After the program is started, by selecting [File] → [New] → [Project], create a new project of the TwinCAT XAE Project type. The subsequent procedure is described below.

4.3 Scanning I/O Devices

- (Scan for devices): Under solution explorer -> I/O -> Devices, select 'Scan' as in Figure below

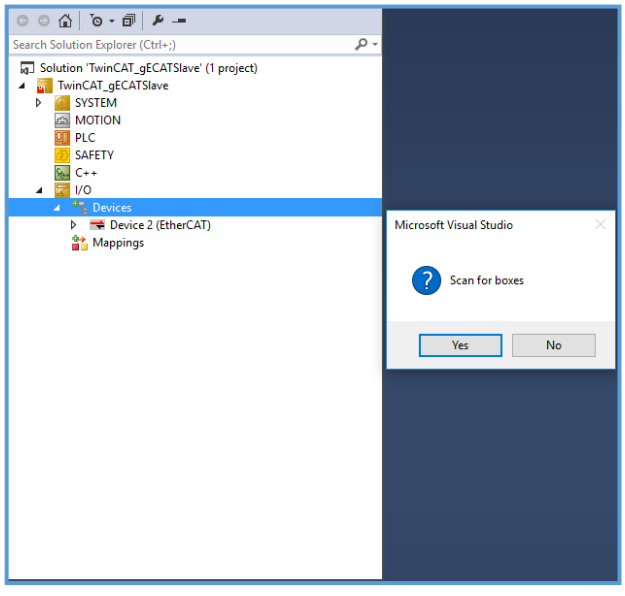


Figure 4.1: Scan Boxes

- (Selecting port): The EtherCAT port will be displayed as below. Select and press OK

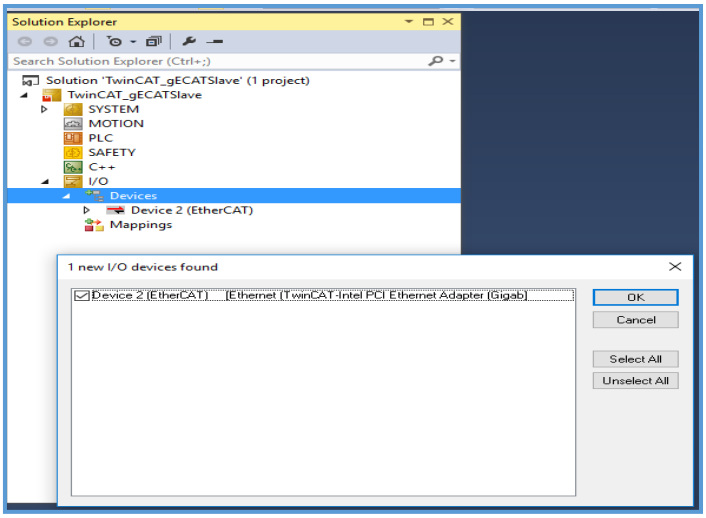


Figure 4.2: Select Port

Note: This will list EtherCAT master, if a valid slave is present in the network

- (Activate slave): The slave is listed in the boxes, in our case “Renesas EtherCAT” in box1 shown in figure below. Press activate free-run.

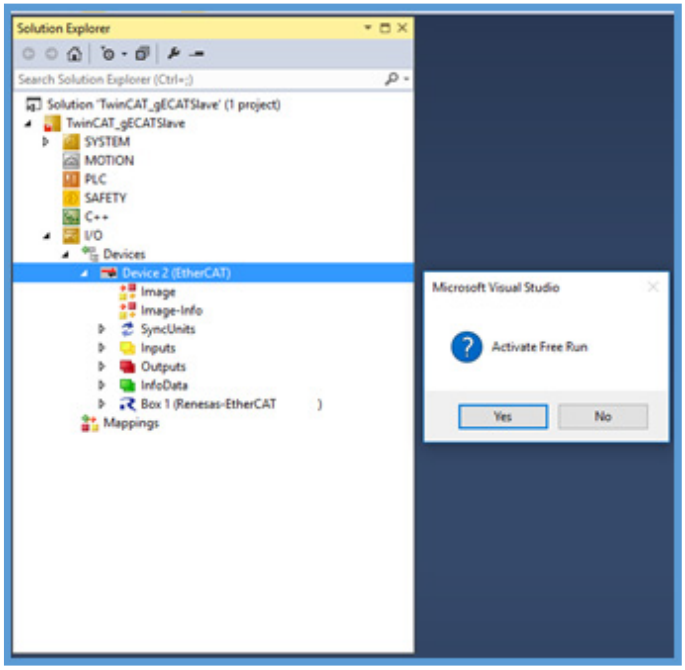


Figure 4.3: Activate Slave

4.4 Updating EEPROM Data

If the data of another application has already been written to the EEPROM, replace the data.

The following shows the procedure for replacing the data on the EEPROM:

(1) Double-click [Box 1] to display a panel on the right side of the window

(2) Select the [EtherCAT] tab.

(3) Click the [Advanced Setting] button.

(4) Select [ESC Access] → [EEPROM] → [Hex Editor].

RZ/N1D, RZ/N1S ESI file includes 2 port and 3 port description. Please select the appropriate one.

Option A - Create ESI binary file from ESI XML and download.

- SSC Tool → [Tool] → [EEPROM Programmer].
- [FILE] → [OPEN] →Browse and select the ESI file
- N1D - "goal\protos\ecat\esi\N1D\Renesas EtherCAT RZN1D.xml".
- N1S - "goal\protos\ecat\esi\N1S\Renesas EtherCAT RZN1S.xml".
- N1L - "goal\protos\ecat\esi\N1L\Renesas EtherCAT RZN1L.xml".
- [FILE] → [Save AS] → Select type as binary.
- A binary file will be generated in the specified folder.
- [Read from File] Select the ESI binary file → [Download].
- Confirm the write status using [Upload] option.

Option B – Download ESI XML.

- Select [Download from list].
- Select [Available EEPROM Description].
- Select [Renesas Electronics Corp.] → [RZ/N1- Slaves] → [RZ/N1 EtherCAT]
- Click the [OK] button.

After the data is replaced, restart the RZ/N1 (by turning it off and on, or resetting it) so that the new data is applied to the microcomputer. Execute [Restart TwinCAT System].

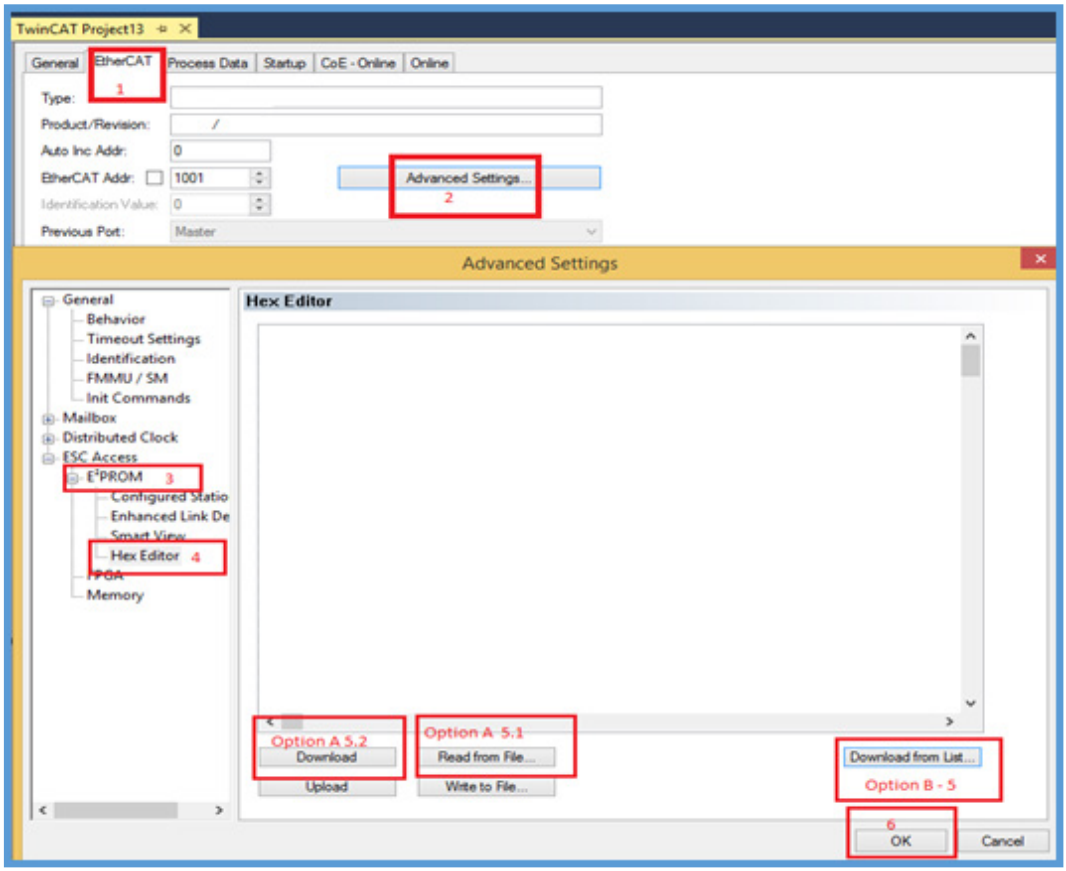


Figure 4.4: EtherCAT EEPROM Programming

4.5 Confirming the Communication Status

(Check for slave state): Check the state of EtherCAT slave in 'Online' tab. This will show "OP" state.

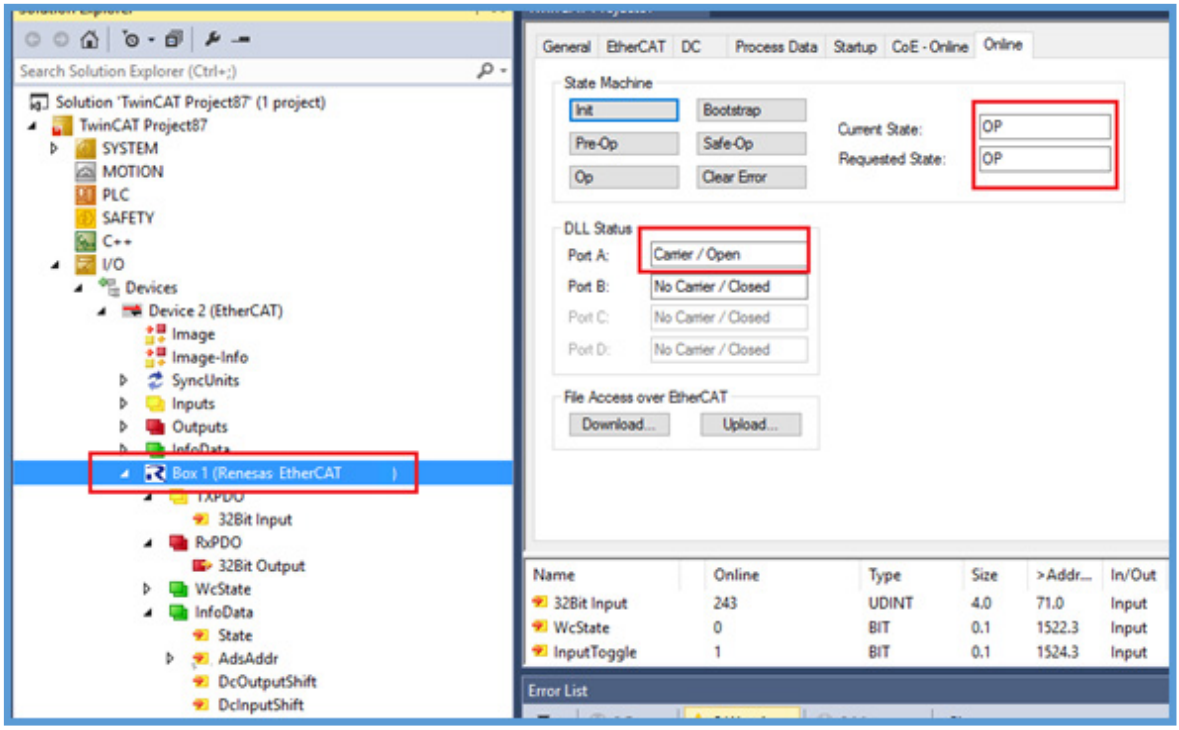


Figure 4.5: Operational Mode- Confirming communication status

5. Testing the I/O Controller Sample Program

5.1 Confirm I/O data

The sample application “01_io_data” operates as followings:

The behavior differs between Standalone Project and Core To Core CC Project.

5.1.1 Standalone Variant

- If “32Bit Output” value equals to 0, “32Bit Input” value is incremented by each communication cycle.
- If “32Bit Output” value doesn’t equal to 0, “32Bit Input” value shows the value of switch SW1 on RZ/N1D-DB board or SW4 on RZ/N1S-DB and RZ/N1L-DB board
- LEDs on RZ/N1D-DB board indicates 8-bits from the least significant bit of “32Bit Output” value.

To confirm I/O communication, select [32Bit Output] → [Online] by using TwinCAT3 and write 0xAA, and check with a 32Bit Input and status LEDs.

Based on the value the I/O expander Bank0 LEDs will glow (Example: 0xAA – 10101010, ref: figure below). Also we can confirm it with the value is updated (incremented counter in TXPDO).

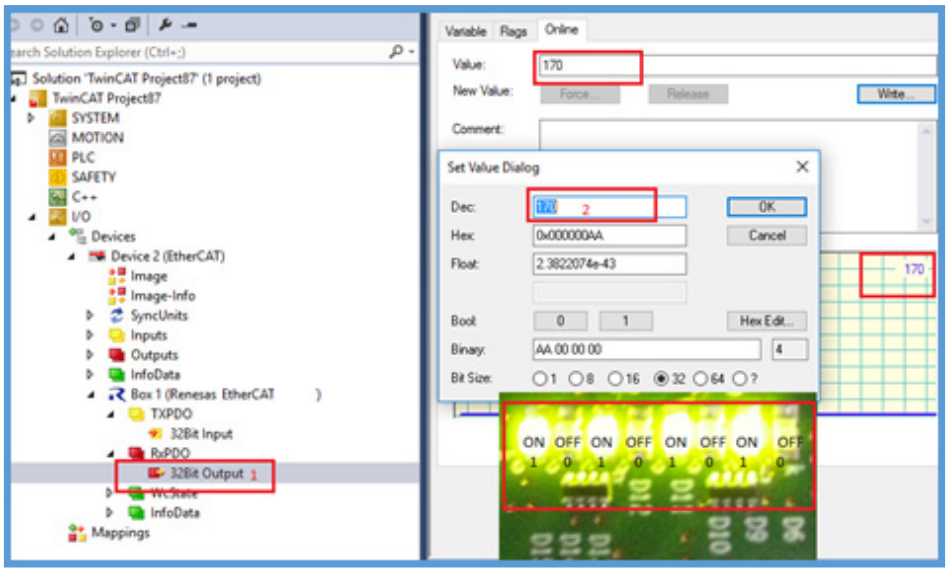


Figure 5.1: Sample IO Application Output

5.1.2 Core To Core variant

- If “32Bit Output” value equals to 0, “32Bit Input” value is incremented by each setting cycle(1sec).
- If “32Bit Output” value doesn’t equal to 0, the value of "32Bit Output" is the same as the value of "32Bit Input".

To confirm I/O communication, select [32Bit Output] → [Online] by using TwinCAT3 and write a value other than 0, and check with a 32Bit Input and Terminal software log.
Also we can confirm it with the value is updated (incremented counter in TXPDO).

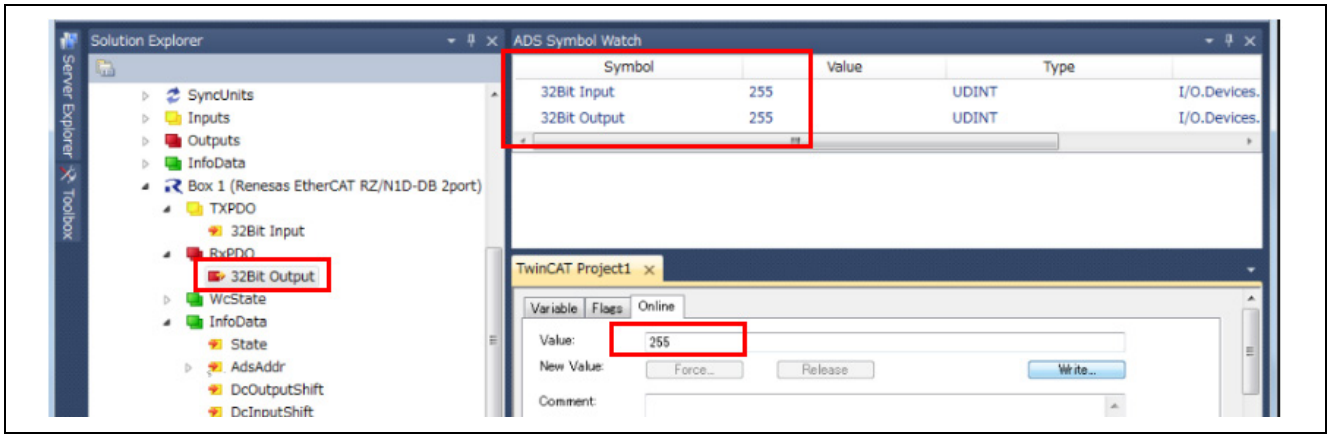


Figure 5.2: Sample IO Application Output

Figure 5.3: Log when changing Output Data

5.2 Sync Modes

The Slave Stack Code supports different modes of synchronization which are based on three physical signals: (PDI_) IRQ, Sync0 and Sync1.

After setting the synchronous mode, please reflect the setting in [TwinCAT] → [Restart TwinCAT (Config Mode)] → [Reload Devices].

5.2.1 Free Run

In this mode there is no slave application synchronization, AL_EVENT_ENABLED and DC_SUPPORTED disabled.

5.2.2 Sync Manager Synchronization

In this mode the slave application is executed as Sync Manager synchronous. AL_EVENT_ENABLED enabled and DC_SUPPORTED disabled.

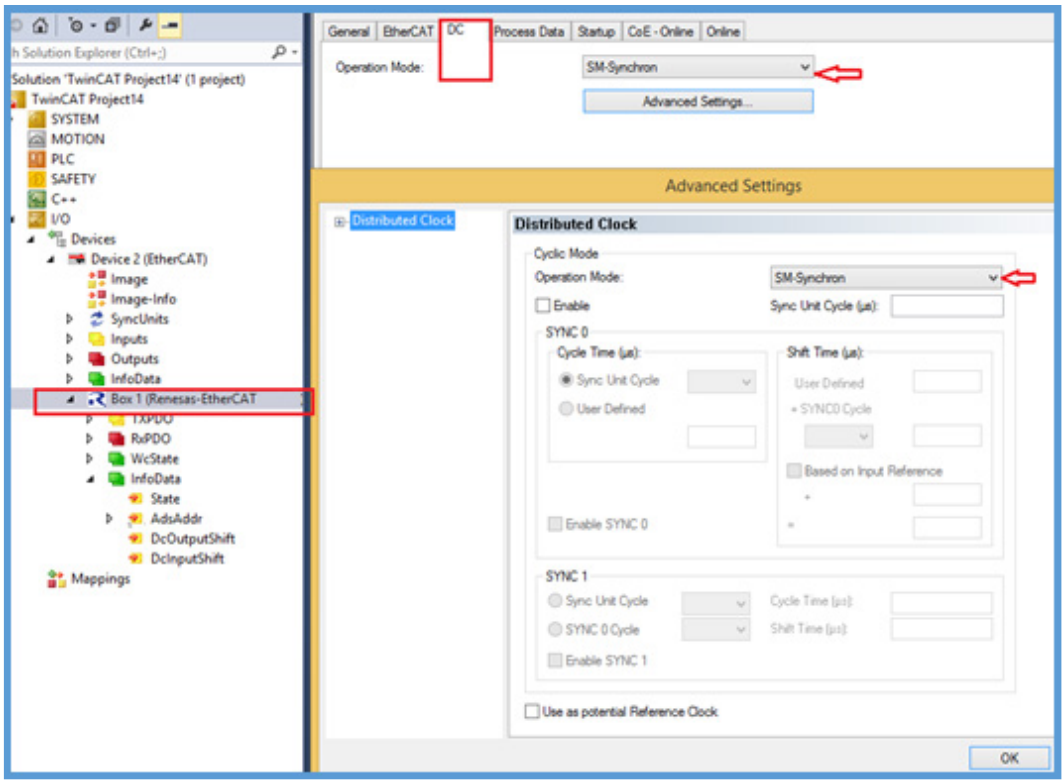


Figure 5.4: SM- Synchronization

5.2.3 DC Synchronization

SyncManager/Sync0 & SyncManager/Sync0/Sync1 synchronous, both AL_EVENT_ENABLED and DC_SUPPORTED are enabled.

1. Master setting for enabling DC synchronization.

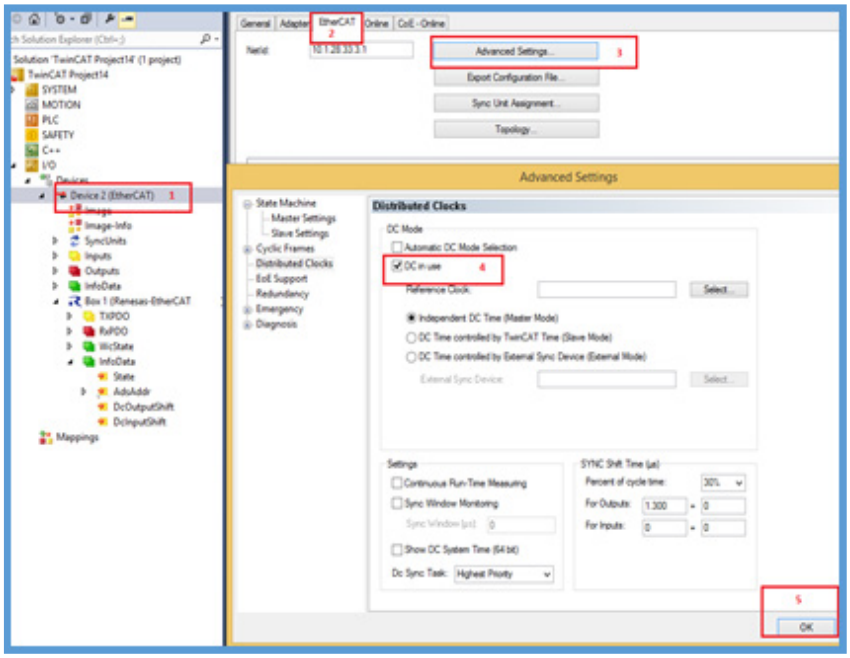


Figure 5.5: DC Setting-Master

2. Slave setting for enabling DC synchronization

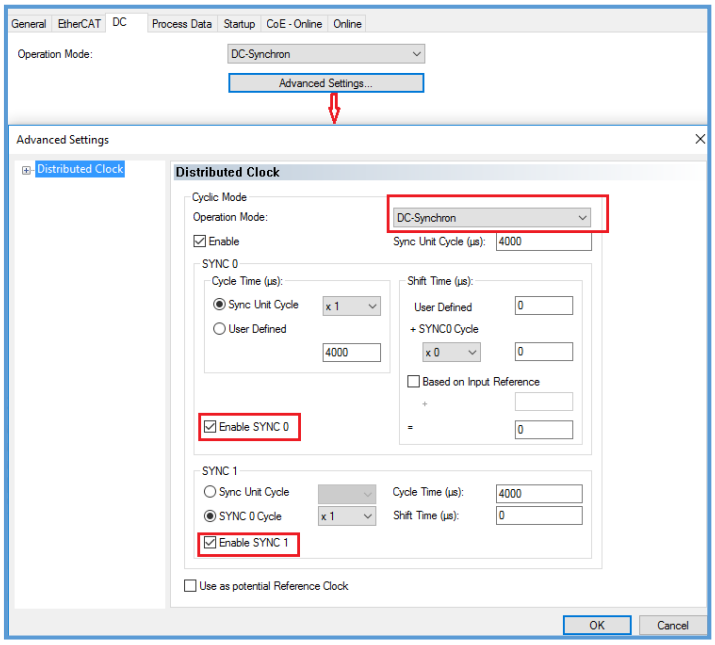


Figure 5.6: DC Setting Slave

Note: SYNC0 and SYNC1 are level triggered interrupt. It can cause synchronization issues due to multiple interrupts in a single pulse. To avoid this issue, pulse width can be reduced by changing the word 2 of EEPROM

configuration value in ESI file. It sets the register 0x982 register of EtherCAT slave during power up. Also can be solved by waiting for the line to be low in the interrupt handler by reading the status register 0x98E.

6. Configuration definition for sample program

RZ/N1x group's EtherCAT sample program can be used with the following configuration by defining EWARM Preprocessor setting, makefile, and configuration file.

■ Definition for the Board type

1. Standalone Variant, Core To Core Variant for CC

EWARM Preprocessor setting

[Project] → [Options] →C/C++ Compiler → Preprocessor → Defined symbols

Table 10. Preprocessor setting at EWARM project

Index	CPU Board	EB Board	Number of ports	Defined symbol
1	RZ/N1D-DB	no	2 port	BSP_PLAT_RZN1DDB=1 RENESAS_CFG_BOARD_RZN1EB=0
2	RZ/N1D-DB	yes	2 port or 3 port	BSP_PLAT_RZN1DDB=1 RENESAS_CFG_BOARD_RZN1EB=1
3	RZ/N1S-DB	no	2 port	BSP_PLAT_RZN1SDB=1 RENESAS_CFG_BOARD_RZN1EB=0
4	RZ/N1S-DB	yes	2 port or 3 port	BSP_PLAT_RZN1SDB=1 RENESAS_CFG_BOARD_RZN1EB=1
5	RZ/N1L-DB	no	2 port	BSP_PLAT_RZN1LDB=1 RENESAS_CFG_BOARD_RZN1EB=0

2. Core To Core Variant for AC

- "/goal/projects/goal_ecat_rpc_ac/01_io_data/Makefile"

Table 11. Makefile Settings

Index	CPU Board	EB Board	Number of ports	Defined symbol
1	RZ/N1D-DB	no	2 port	CONFIG_MAKE_FEAT_3PORTS_EN=0
2	RZ/N1D-DB	yes	2 port or 3 port	CONFIG_MAKE_FEAT_3PORTS_EN=1

■ Definition for EtherCAT and the number of ports

Standalone variant, Core To Core Variant for AC

- "/goal/appl/goal_ecat/01_io_data/goal_config.h"

Core To Core variant for CC

- "/goal/appl/goal_ecat/00_rpc_cc/goal_config.h"

3. Macro definition for EtherCAT slave and the number of port

Table 12. Macro definition for EtherCAT slave and its ports

Setting	Description
2 port	#define GOAL_CONFIG_ETHERCAT 1 #define GOAL_CONFIG_ETHERCAT_3PORTS 0
3 port	#define GOAL_CONFIG_ETHERCAT 1 #define GOAL_CONFIG_ETHERCAT_3PORTS 1

7. Limitations

1. On the RZ/N1S, L DB board, 256 byte EEPROM is mounted as standard, therefore, EtherCAT Conformance Test does not match. If want to conform to the conformance test, increase the capacity of the EEPROM.

2. In the case of the standard RZ/N1D, S DB board + EB board configuration, the EtherCAT Enhanced Link Detection function can not be used. If want to use the Enhanced Link Detection function, allocate the PHY address from ETH5 by simple increase as shown in the table below.

External Ethernet port name	Internal port of EtherCAT	PHY address (now)	PHY address for Enhanced Link Detection
ETH5	Port A	5	2
ETH4	Port B	4	3
ETH3	Port C	10	4
ETH2	(none)	1	1
ETH1	(none)	8	8

8. Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
0.10	Dec 08, 2017	—	First edition issued
0.11	Dec 18, 2017	4	Changed EWARM version 8.11 to 7.80.4
0.20	Feb 8, 2018	14--15	Add 3 port setting
0.80	Mar 5, 2018	1--16	Add RZ/N1L setting
0.90	Jun 28, 2018	3	Updated names of documents
		4	Added the EtherCAT logo
			Changed EWARM version and SSC Tool version
		6-9	Changed a usage of SSC Tool
		10	Changed names and locations of EWARM project files
1.00	Oct 22, 2018	3	Added explanation of two cores
		6	Added connection setting of UART and DFU
		7	Added folder description
		11-14	Changed execution procedure of sample application
		21	Added check of operation with Core To Core variant
		22	Added sync modes setting in TwinCAT
		24	Changed configuration definition of sample program
1.10	Mar 29, 2019	26	Added Limitation

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0