# RZ/T1 Group

## Encoder I/F A-format application package

R01AN3635EJ0231
Rev.2.31
March 31, 2025

### Summary

This document explains about RZ/T1 Encoder I/F A-format application package.

To use this application package, please obtain release package of “RZ/T1 Encoder I/F Configuration Library”.

### Device that A-format functionality is checked

RZ/T1 CPU Board（RTK7910018C00000BE）

## Version History

| Ver. | Date | Content | Note |
|---|---|---|---|
| 2.31 | March 2025 | Update description of Release Note Summary. | |
| 2.30 | October 2019 | Update the RZ/T1 A-format sample driver code.<br>(1) Fixed bug of disables input of the ELC event triggers processing.<br>(2) Fixed bug initialization of the internal variables. | |
| 2.20 | August 2018 | Update the RZ/T1 A-format sample driver code.<br>(1) Supported bypass reception for ch1.<br>(2) Fixed bug of bypass reception processing.<br>(3) Changed the operation procedure of DS-5 and e2 studio.<br>(4) Added restriction | |
| 2.10 | April 2018 | Update the RZ/T1 A-format sample driver code.<br>(1) Added ID macro definition for ch1.<br>(2) Changed register definition for ch1. | |
| | | Update the RZ/T1 Group A-format Interface User's Manual. | |
| 2.00 | April 2017 | Update the RZ/T1 A-format sample driver code.<br>(1) Supported the Configuration Data Ver.1.8. | |
| | | Updated Configuration Data to Ver.1.8.<br>(1) Added the bypass transmission and reception function using FIFO.<br>(2) Added the received data setting completion Interrupt.<br>(3) Added the ELC Event Input Trigger. | |
| | | Update the RZ/T1 Group A-format Interface User's Manual. | |
| | | Update the RZ/T1 Group A-format Sample Program Application Note. | |
| 1.01 | January 2017 | Update the RZ/T1 A-format sample driver code.<br>(1) Improved the stability of the module stop release operation. | |
| 1.0 | September 2016 | Fixed bug of interrupt routine of sample program. | |
| | | Update the RZ/T1 Group A-format Interface User's Manual. | |
| | | Update the RZ/T1 Group A-format Sample Program Application Note. | |
| 0.8 | June 2016 | Fixed bug of macro value of sample program.<br>Fixed bug of interrupt routine of sample program.<br>Fixed bug of terminal screen display of sample program.<br>Changing the input method of the console commands of the sample program. | |
| | | Update the RZ/T1 Group A-format Interface User's Manual. | |
| | | Update the RZ/T1 Group A-format Sample Program Application Note (Japanese). | |
| | | Added the RZ/T1 Group A-format Sample Program Application Note (English). | |
| 0.7 | October 2015 | Newly created | |

Table of contents

1. Contents of package ....4
1.1 Software ....4
・Source code ....4
・Configuration data ....4
1.2 Document ....4

2. File Structures ....5

3. Information about A-format sample program ....6
3.1 Software information ....6
3.1.1 Independent from Operating System ....6
3.1.2 Memory footprint ....6
3.2 Hardware information ....7
3.2.1 Device ....7
3.2.2 Target Board ....7
3.3 Procedure on Development Environments ....8
3.3.1 Preparation for the execution of the sample program ....8
3.3.2 EWARM from IAR systems ....8
3.3.3 DS-5 from ARM ....10
3.3.4 e2 studio from RENESAS ....13

4. Restriction ....15
4.1 RZ/T1 A-format sample driver ....15

5. Note ....15
5.1 Processing time ....15
5.2 About mistake of Application Note ....16

# 1. Contents of package

Contents of this package are described in this chapter.

Configuration data and sample programs in this package support only 1 channel of Encoder I/F. In order to use 2 channels of Encoder I/F, obtain the RZ/T1 group Encoder I/F 2ch Tool (R01AN4306) and change the Configuration Data and sample program.

## 1.1 Software

・Source code

| No. | Title | Version |
|---|---|---|
| 1 | A set of RZ/T1 A-format sample driver code | 2.5 |

・Configuration data

| No. | Title | Version |
|---|---|---|
| 1 | RZ/T1 Encoder I/F Configuration Data(A-format) | 1.8 |

## 1.2 Document

| No. | Document name | Ver. | File name |
|---|---|---|---|
| 1 | RZ/T1 Encoder I/F A-format application package release note | 2.31 | (English)<br>r01an3635ej0231-rzt1.pdf (this document)<br>(Japanese)<br>r01an3635jj0231-rzt1.pdf |
| 2 | RZ/T1 Group A-format Interface User's Manual | 2.10 | (English)<br>r01uh0604ej0210-rzt1-a-format.pdf<br>(Japanese)<br>r01uh0604jj0210-rzt1-a-format.pdf |
| 3 | RZ/T1 Group A-format Sample Program Application Note | 2.00 | (English)<br>r01an2948ej0200_rzt1_a-format.pdf<br>(Japanese)<br>r01an2948jj0200_rzt1_a-format.pdf |

## 2. File Structures

File structures and contents of this package are described below.

```
Top
 ├──r01an3635ej0231-rzt1.pdf
 ├──r01an3635jj0231-rzt1.pdf
 └──workspace
     ├──Software
     │   ├──armcc
     │   │   └──RZ_T1_a_as.zip : A set of RZ/T1 A-format sample driver code (DS-5)
     │   ├──iccarm
     │   │   └──RZ_T1_a_as.zip : A set of RZ/T1 A-format sample driver code (IAR)
     │   └──kpitgcc
     │       └──RZ_T1_a_as.zip : A set of RZ/T1 A-format sample driver code (e2 studio)
     └──Documentation
         ├──r01uh0604ej0210-rzt1-a-format.pdf
         ├──r01uh0604jj0210-rzt1-a-format.pdf
         ├──r01an2948ej0200_rzt1_a-format.pdf
         └──r01an2948jj0200_rzt1_a-format.pdf
```

The file structures of “RZ_T1_a_as.zip” are shown in Figure 2.1 File Structure.

```
Top folder
  lib
    ecl
      r_a_as_rzt1.dat              Multi-Protocol Encoder IF (a_as mode) Configuration data
  inc
    iodefine.h                     RZ/T1 register definition file
    iodefine_a_as.h                A-format register definition file
    r_a_as_rzt1_dat.h              r_a_as_rzt1.dat header file
    r_a_as_rzt1_if.h               A_AS common driver header file
  src
    common
      Common sources including initial settings
    drv
      scifa_uart
        SCIFA Sample program
        a_as
          r_a_as_rzt1_config.h     A_AS common driver file
          r_a_as_rzt1_private.h    A_AS common driver file
          r_a_as_rzt1.c            A_AS common driver file
          a_format
            r_a_format_rzt1_config.h   A-format driver file
            r_a_format_rzt1_private.h  A-format driver file
            r_a_format_rzt1.c          A-format driver file
    sample
      main.c                       A-format Sample program
      a_as_dat.s                   Linker setting file of the Configuration data *1
      siochar.c                    SCIFA Sample program
      siorw.c                      SCIFA Sample program
  RN_CommClass_RZ_T1.inf           USB serial port driver for PC

Note 1 : file for DS-5 and e2 studio
         DS-5 : a_as_dat.s
         e2 studio : a_as_dat_asm
```

Figure 2.1 File Structure

# 3. Information about A-format sample program

This chapter describes information to use a set of A-format sample driver code.

## 3.1 Software information

### 3.1.1 Independent from Operating System

This software is independent from operating system.

### 3.1.2 Memory footprint

| Section name | | | Memory Size | | |
|---|---|---|---|---|---|
| | | | IAR [bytes] | DS-5 [bytes] | e2 studio [bytes] |
| A-format driver | Code | | 6400 | 8440 | 16236 |
| | Data (with initial value) | | 16 | 26 | 16 |
| | Data (without initial value) | | 1160 | 1152 | 1164 |
| | Constant Data | | 132 | 132 | 140 |
| | Stack size of function | R_A_AS_Open | 60 | 68 | 144 |
| | | R_A_AS_Close | 16 | 28 | 84 |
| | | R_A_AS_Control | 64 | 88 | 192 |
| | | R_A_AS_GetVersion | 0 | 16 | 72 |
| | | a_as0_int_isr | 168+n *1 | 116+n *1 | 156+n *1 |
| | | a_as0_fss_isr | 160+n *1 | 100+n *1 | 132+n *1 |
| A-format Configuration data | Code | | 0 | 0 | 0 |
| | Data (with initial value) | | 0 | 0 | 0 |
| | Data (without initial value) | | 0 | 0 | 0 |
| | Constant Data | | 21932 | 21932 | 21932 |
| Sample program | Code | | 6008 | 8676 | 14244 |
| | Data (with initial value) | | 136 | 152 | 136 |
| | Data (without initial value) | | 1829 | 1816 | 1836 |
| | Constant Data | | 2168 | 328 | 2138 |

Note 1. “n” is the Maximum stack size of user defined callback functions that are registered to R_A_AS_Control function.

## 3.2 Hardware information

### 3.2.1 Device

RZ/T1

### 3.2.2 Target Board

- Board name

  RZ/T1 CPU Board (RTK7910018C00000BE)

(2) Settings of CPU Board

SW4-1: ON

SW4-2: ON in case of serial flash memory is used, OFF in case of NOR flash memory is used

SW4-3: ON

SW4-4: ON

SW4-5: ON

SW4-6: OFF

JP2: 2-3 Connect

JP7: 1-2 Connect

## 3.3 Procedure on Development Environments

### 3.3.1 Preparation for the execution of the sample program

This sample program communicates with the PC. And for setting the PC, please refer to 6.1.2 Preparations of "RZ/T1 Group FIFO Integrated Serial Communication Interface (SCIFA) Application Note".

### 3.3.2 EWARM from IAR systems

- Build environment

  IAR Embedded Workbench for ARM v8.40.1

- Execution environment

  I-jet

- How to build sample program

  1. Extract files from RZ_T1_a_as.zip and copy the files to arbitrary holder
  2. Copy the following files of "RZ/T1 Encoder I/F Configuration Library" (for IAR EWARM) to each folder

     lib\ecl\r_ecl_rzt1.a

     inc\r_ecl_rzt1_if.h
  3. Launch EWARM
  4. Select [File]menu -> [Open] -> [Workspace]
  5. Open RZ_T1_a_as_boot\RZ_T1_a_as_****_boot.eww

| | |
|---|---|
| NOR version | RZ_T1_a_as_nor_boot.eww |
| Serial Flash version | RZ_T1_a_as_serial_boot.eww |

  6. Select [Project]menu -> [Rebuild all]

     Following file is generated.

     RZ_T1_a_as_boot\Debug\Exe\RZ_T1_a_as_****_boot.out

| | |
|---|---|
| NOR version | RZ_T1_a_as_nor_boot.out |
| Serial Flash version | RZ_T1_a_as_serial_boot.out |

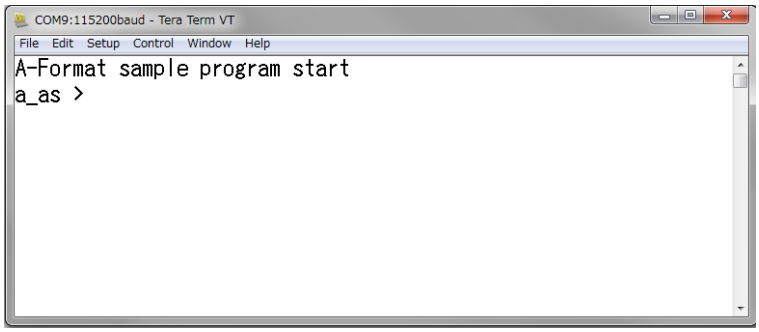
- How to execute sample program

  After executing "How to build sample program", connect the target board and the debugger properly, and execute the following operations.

  1. Select [Project] menu-> [Download and Debug]
  2. Select [Debug] menu-> [Go]

- Execution result of sample program

  After executing a sample program, input the command to " Terminal I/O" window.
  Please refer to RZ/T1 Group A-format Sample Program Application Note about the command.



### 3.3.3 DS-5 from ARM

- Build environment

  ARM Development Studio 5 (DS-5) Version 5.29.2

  ARM Compiler 5.06 update 6

- Execution environment

  ULINK2（v2.01）

- How to build sample program

  1. Startup the DS-5 environment. Go to [File] > [Import]. On the [Import] window, select [Existing Projects into Workspace] in the [General] folder and click the [Next] button.
  2. Select the [Select archive file:] radio button and click on the [Browse…] button. Select the compressed program file “RZ_T1_a_as.zip” on the list in the window and click on the [Finish] button.
  3. Copy the following files of “RZ/T1 Encoder I/F Configuration Library” (for ARM DS-5) to each folder imported and expanded.

     lib\ecl\r_ecl_rzt1.a

     inc\r_ecl_rzt1_if.h
  4. Select [Project] menu -> [Build All]

     Following file is generated.

     Debug\RZ_T_nor_sample.axf

     (In case of serial flash, use the “RZ_T_sflash_sample.axf” instead of the “RZ_T_nor_sample.axf”)

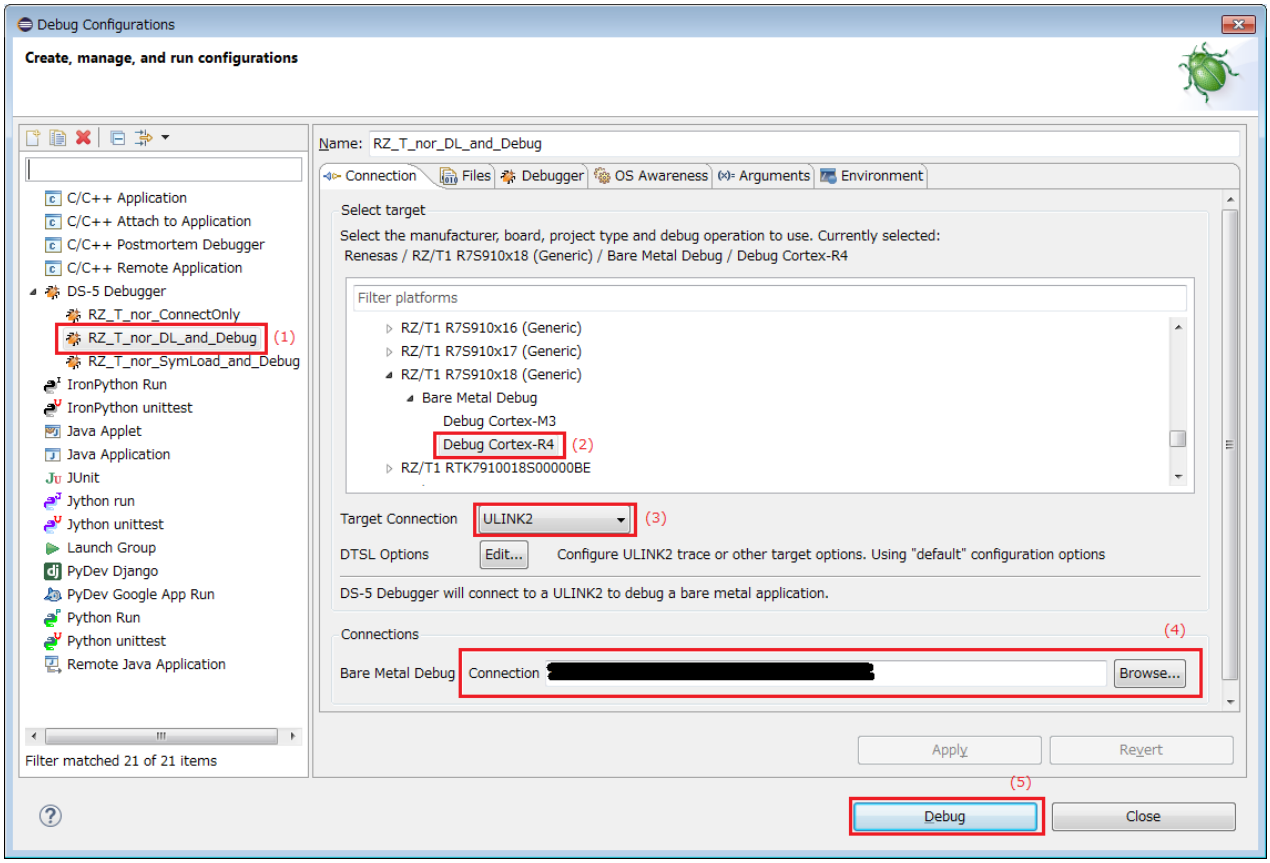
➢ How to execute sample program

After executing “How to build sample program”, connect the target board and the debugger properly, and execute the following operations.

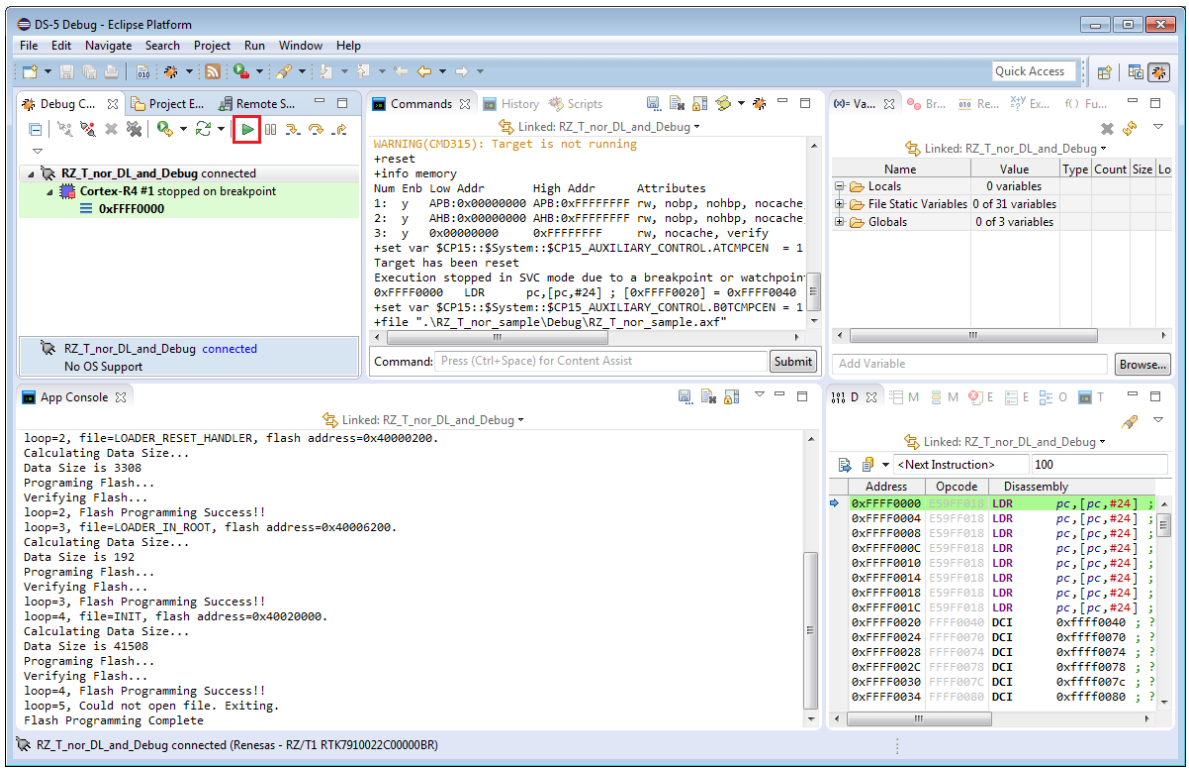
1. Open the debug configuration from the [Run] -> [Debug Configurations…], select the configuration window for “RZ_T_nor_DL_and_Debug”. (In case of serial flash, use the “RZ_T_sflash_DL_and_Debug” instead of the “RZ_T_nor_DL_and_Debug”)

   Select “Debug Cortex-R4” of “RZ/T1 R7S910x18 (Generic)” in [Select target].

   Select the ULINK2 of [Target Connection] in [Connection] tab, click on [Browse] and select the target connection from the list in the window. Click on [Debug] in the debug configurations window and start debugging.

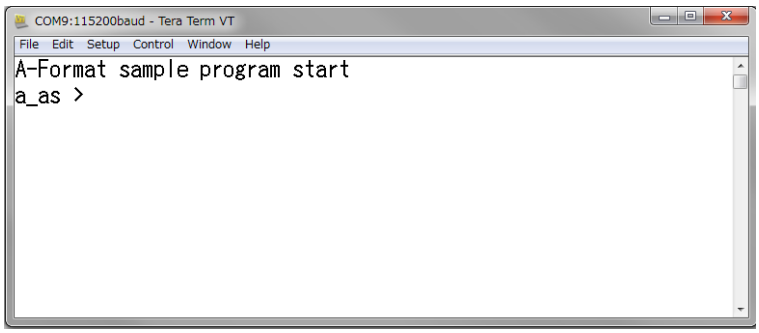


2. On completion of writing to the flash memory by the script, the message “Flash Programming Complete” appears in the application console window. Debugging can then start.




➢ Execution result of sample program

After executing a sample program, input the command to " Terminal I/O" window.
Please refer to RZ/T1 Group A-format Sample Program Application Note about the command.



### 3.3.4 e2 studio from RENESAS

- Build environment

  RENESAS e2 studio 7.5.0

  KPIT GNUARM-NONE-EABI Toolchain v16.01

- Execution environment

  J-Link BASE

- How to build sample program

  1. Start up the e2 studio environment. In the workspace, go to [File] > [Import]. On the [Import] window, select [Existing Projects into Workspace] in the [General] folder and click the [Next] button.
  2. Select the [Select archive file:] radio button and click on the [Browse..] button. Select the compressed program file “RZ_T1_a_as.zip” on the list in the window and click on the [Finish] button.
  3. Copy the following files of “RZ/T1 Encoder I/F Configuration Library” (for KPIT GCC) to each folder imported and expanded.

     lib\ecl\r_ecl_rzt1.a

     inc\r_ecl_rzt1_if.h
  4. Select [Project] menu -> [Build All]

     Following file is generated.

     HardwareDebug\ RZ_T_nor_sample.x

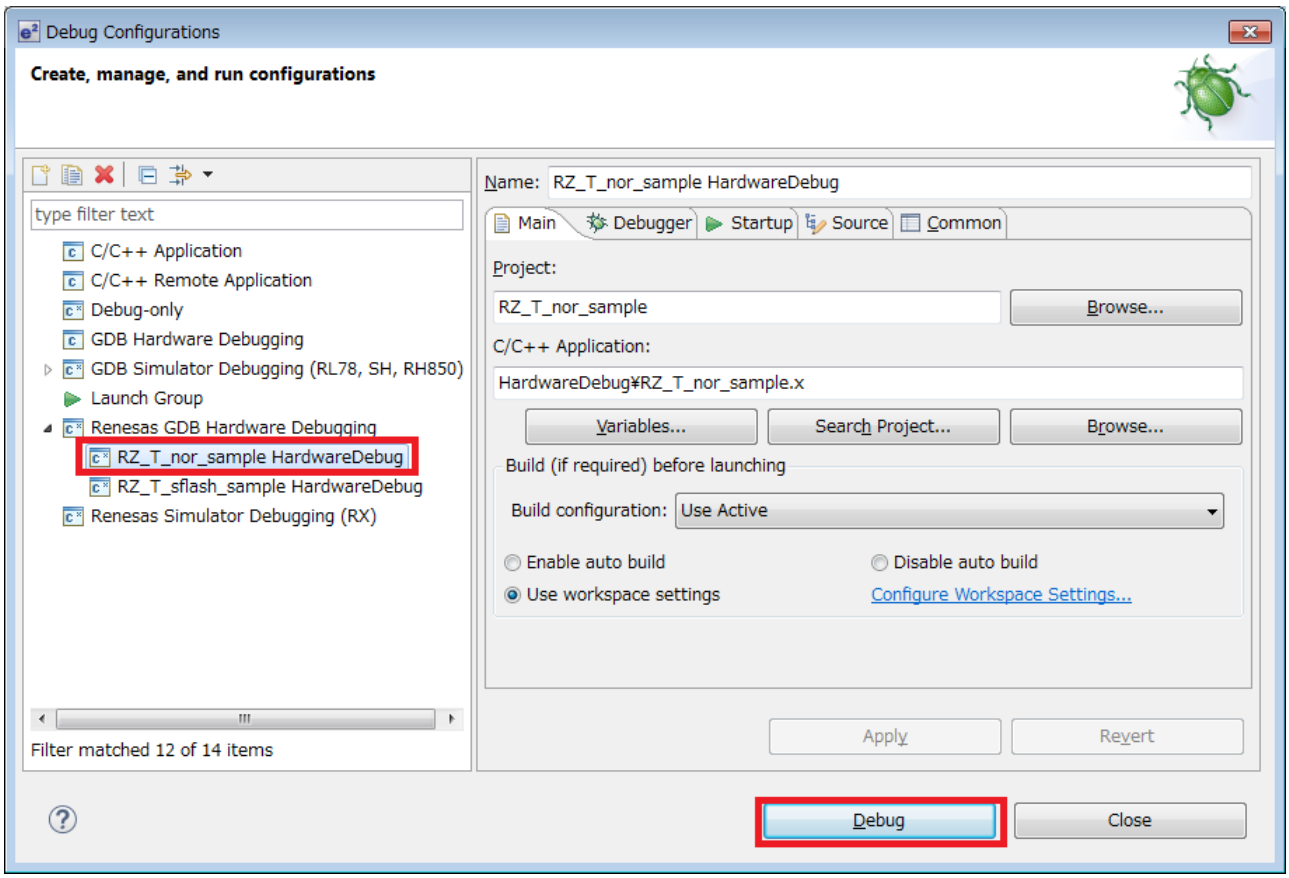
     (In case of serial flash, use the “RZ_T_sflash_sample.x” instead of the “RZ_T_nor_sample.x”)

➢ How to execute sample program

After executing “How to build sample program”, connect the target board and the debugger properly, and execute the following operations.

1. Select [Run] from the [Project] menu and then select [Debug Configurations].
2. Select the [RZ_T_nor_sample_HardwareDebug] in the following screen. Click the [Debug] and start the download to flash memory.

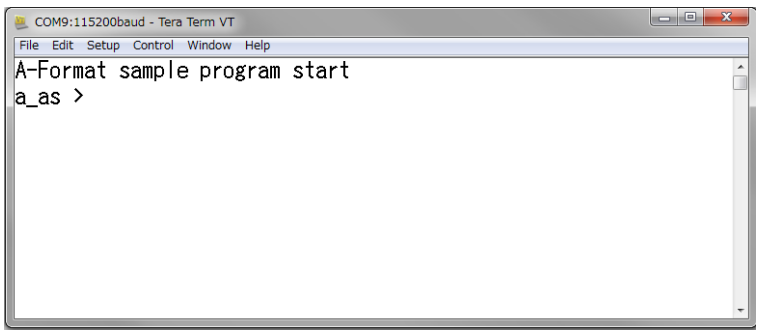
   (In case of serial flash, use the [RZ_T_sflash_sample_HardwareDebug] instead of the [RZ_T_nor_sample_HardwareDebug])




3. Click the [Resume] from the [Run] to start execution of the sample program.

➢ Execution result of sample program

After executing a sample program, input the command to " Terminal I/O" window.
Please refer to RZ/T1 Group A-format Sample Program Application Note about the command.



# 4. Restriction

## 4.1 RZ/T1 A-format sample driver

- When bypass reception is enabled, member variable "rxbpende" of structure "r_a_as_req_t" sets enable or disable of FSS_UPD interrupt request by FSS.RXEND bit = 1. If set to disable, A-format sample driver will not operate properly. Therefore, member variable "rxbpende" of structure "r_a_as_req_t" must be fixed to "true".

# 5. Note

## 5.1 Processing time

Available time for user processing of Encoder I/F A-format sample program in a control loop is as follows.

Please confirm that there are no problems in your environment.

The example of the case that the control cycle is 62.5us and the connection type is one to one is indicated below.

In the case of normal reception, the time used by the sample program is about 9 us (15%) of 62.5us, and available time for user processing is about 53.5 us (85%).

In the case of bypass reception, the time used by the sample program is about 6 us (10%) of 62.5us, and available time for user processing is about 56.5 us (90%).

| Processing | | | Time | | Occupancy rate |
|---|---|---|---|---|---|
| A-format sample processing *2 | normal reception | Time setting registers for transmission | about 4 us | about 9 us | 15% |
| | | Interrupt time | about 5 us | | |
| | | Available time for user processing | about 53.5 us *1 | | 85% |
| | bypass reception | Time setting registers for transmission | about 4 us | about 6 us | 10% |
| | | Interrupt time | about 2 us | | |
| | | Available time for user processing | about 56.5 us *1 | | 90% |

Note 1. Communication time with the encoder (when the bitrate is 8Mbps and the command to be sent is CDF0 which is individual transmission command) is 17.75us of available time for user processing. For more information, refer to the “RZ/T1 Group A-format Interface User's Manual”.

Note2. Initial setting time is not included.

## 5.2 About mistake of Application Note

The following mistake is existing in "RZ/T1 Group A-format Sample Program Application Note Rev.2.00".

➢ Table 5.4 Results of Transmission and Reception Stored in Each Array Element in Bypass Reception
Correct:

| Array Number | Content |
|---|---|
| pbp_result[0] | Result of transmission and reception for the encoder section ENC1 when the setting of received data is complete(RXSET). |
| pbp_result[1] | Result of transmission and reception for the encoder section ENC2 when the setting of received data is complete(RXSET). |
| pbp_result[2] | Result of transmission and reception for the encoder section ENC3 when the setting of received data is complete(RXSET). |
| pbp_result[3] | Result of transmission and reception for the encoder section ENC4 when the setting of received data is complete(RXSET). |
| pbp_result[4] | Result of transmission and reception for the encoder section ENC5 when the setting of received data is complete(RXSET). |
| pbp_result[5] | Result of transmission and reception for the encoder section ENC6 when the setting of received data is complete(RXSET). |
| pbp_result[6] | Result of transmission and reception for the encoder section ENC7 when the setting of received data is complete(RXSET). |
| pbp_result[7] | Result of transmission and reception for the encoder section ENC8 when the setting of received data is complete(RXSET). |
| pbp_result[8] | Result of transmission and reception when the transmission and reception of data has been completed(RXEND)*1. |

Note1. The result is invalid when the interrupt source RXEND is disabled.

Incorrect:

| Array Number | Content |
|---|---|
| pbp_result[0] | Result of transmission and reception for the encoder section ENC1 when the setting of received data is complete(RXSET). |
| pbp_result[1] | Result of transmission and reception for the encoder section ENC2 when the setting of received data is complete(RXSET). |
| pbp_result[2] | Result of transmission and reception for the encoder section ENC3 when the setting of received data is complete(RXSET). |
| pbp_result[3] | Result of transmission and reception for the encoder section ENC4 when the setting of received data is complete(RXSET). |
| pbp_result[4] | Result of transmission and reception for the encoder section ENC5 when the setting of received data is complete(RXSET). |
| pbp_result[5] | Result of transmission and reception for the encoder section ENC6 when the setting of received data is complete(RXSET). |
| pbp_result[6] | Result of transmission and reception for the encoder section ENC7 when the setting of received data is complete(RXSET). |
| pbp_result[7] | Result of transmission and reception when the transmission and reception of data has been completed(RXEND)*1. |

Note1. The result is invalid when the interrupt source RXEND is disabled.

➢ Table 5.20 Static Variables Used in the Sample Program
Correct: a_as_bp_result[A_AS_BP_RESULT_NUM]

Incorrect: a_as_bp_result[A_AS_ENC_NUM]