RZ/T1 Group

Encoder I/F EnDat2.2 application package

R01AN3580EJ0110
Rev.1.10
April 2, 2018

Summary

This document explains about RZ/T1 Encoder I/F EnDat2.2 application package.

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

If you require detailed specifications of EnDat2.2 protocol and the encoder, please contact HEIDENHAIN Co to obtain them.

Device that EnDat2.2 functionality is checked

RZ/T1 CPU Board（RTK7910022C00000BR）

Version History

Ver.	Date	Content	Note
1.10	April 2018	Update the RZ/T1 EnDat2.2 sample driver code. (1) Added ID macro definition for ch1. (2) Changed register definition for ch1. Update the RZ/T1 Group EnDat2.2 Interface (EnDat) User’s Manual.	
1.00	January 2017	Changed version of the release note.	
0.90	December 2016	Fixed bug of measurement for propagation delay. Fixed bug that misrecognized additional information. Fixed bug in Ftype1 error handling. Improved the stability of the module stop release operation. Improved the stability of the interrupt operation. Updated document “RZ/T1 Group EnDat2.2 Interface (EnDat) User’s Manual (English/ Japanese)” Updated document “RZ/T1 Group EnDat sample program Application Note (English/ Japanese)”	
0.81	January 2016	Fixed bug of compensation for propagation delay. Fixed bug of interrupt routine of sample driver. Changed the input method of the console commands of the sample program.	
0.8	July 2015	Fixed bug at the time of detecting Ftype2 error and MRSADR error. Fixed bug of interrupt routine of sample driver. Deleted unsupported mode command from sample driver. Fixed bug of “continuous reception of position values” of sample program. Added document “RZ/T1 Group EnDat2.2 Interface (EnDat) User’s Manual (English/ Japanese)”. Added document “RZ/T1 Group EnDat sample program Application Note (English/ Japanese)”.	
0.2 (Alpha2)	February 2015	Newly created	

Table of contents

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 EnDat2.2 sample driver code	1.3

・Configuration data

No.	Title	Version
1	RZ/T1 Encoder I/F Configuration Data(EnDat2.2)	1.4

1.2 Document

No.	Document name	Ver.	File name
1	RZ/T1 Encoder I/F EnDat2.2 sample program release note	1.10	(English) r01an3580ej0110-rzt1.pdf (this document) (Japanese) r01an3580jj0110-rzt1.pdf
2	RZ/T1 Group EnDat 2.2 Interface (EnDat) User's Manual	1.40	(English) r01uh0589ej0140_rzt1_endat2.2.pdf (Japanese) r01uh0589jj0140_rzt1_endat2.2.pdf
3	RZ/T1 Group EnDat Sample Program Application Note	1.20	(English) r01an2793ej0120_rzt1_endat.pdf (Japanese) r01an2793jj0120_rzt1_endat.pdf

2. File Structures

File structures and contents of this package are described below.

```
Top
 ├──r01an3580ej0110-rzt1.pdf
 ├──r01an3580jj0110-rzt1.pdf
 └──workspace
     ├──Software
     │    ├──armcc
     │    │    └──RZ_T1_endat.zip : A set of RZ/T1 EnDat2.2 sample driver code (DS-5)
     │    ├──iccarm
     │    │    └──RZ_T1_endat.zip : A set of RZ/T1 EnDat2.2 sample driver code (IAR)
     │    └──kpitgcc
     │         └──RZ_T1_endat.zip : A set of RZ/T1 EnDat2.2 sample driver code (e2 studio)
     └──Documentation
          ├──r01an2793ej0120_rzt1_endat.pdf
          ├──r01an2793jj0120_rzt1_endat.pdf
          ├──r01uh0589ej0140_rzt1_endat2.2.pdf
          └──r01uh0589jj0140_rzt1_endat2.2.pdf
```

The file structures of “RZ_T1_endat.zip” are shown in Figure 2.1 File Structure.

```
Top folder
  inc
    iodefine.h                RZ/T1 register definition file
    iodefine_fac.h            EnDat2.2register definition file
    r_endat_rzt1_dat.h        r_endat_rzt1.dat header file
    r_endat _rzt1_if.h        EnDat2.2 common driver header file
  lib
    ecl
      r_endat_rzt1.dat        Multi-Protocol Encoder IF(EnDat2.2 mode) Configuration data
  src
    common
      Common sources including initial settings
    drv
      scifa_uart
        SCIFA sample program
      fac
        r_endat_rzt1_config.h     EnDat2.2 common driver file
        r_endat_rzt1.c            EnDat2.2 common driver file
    sample
      main.c                  Sample program
      endat_dat.s             Linker setting file of the Configuration data *1
      siochar.c               SCIFA Sample program
      siorw.c                 SCIFA Sample program

Note 1 : file for DS-5 and e2 studio
         DS-5 : endat_dat.s
         e2 studio : endat_dat.asm
```

Figure 2.1 File Structure

3. Information about EnDat2.2 sample program

This chapter describes information to use a set of EnDat2.2 sample driver.

3.1 Software information

3.1.1 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]
EnDat2.2 sample driver	Code		4396	5820	8832
	Data (with initial value)		8	30	8
	Data (without initial value)		108	86	114
	Constant Data		74	76	76
	Stack size of function	R_ENDAT_Open	108	100	180
		R_ENDAT_Close	20	28	84
		R_ENDAT_Control	36	32	144
		R_ENDAT_GetVersion	0	0	0
		endat0_int_isr	112+n *1	48+n *1	90+n *1
EnDat2.2 Configuration data	Code		0	0	0
	Data (with initial value)		0	0	0
	Data (without initial value)		0	0	0
	Constant Data		21700	21700	21700
Sample program	Code		2320	2968	4620
	Data (with initial value)		20	35	20
	Data (without initial value)		2564	2552	2572
	Constant Data		1112	53	1094

*1 “n” is the Maximum stack size of user defined callback functions that are registered to R_ENDAT_Control function

3.2 Hardware information

3.2.1 Device

RZ/T1

3.2.2 Target Board

(1) Board name
RZ/T1 CPU Board (RTK7910022C00000BR)

(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 execution of 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 v7.80.2

- Execution environment

  I-jet

- How to build sample program

  1. Extract files from RZ_T1_endat.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_endat\RZ_T1_endat_****_boot.eww

NOR version	RZ_T1_endat_nor_boot.eww
Serial Flash version	RZ_T1_endat_serial_boot.eww

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

     Following file is generated.

     RZ_T1_endat\Debug\Exe\RZ_T1_endat_****_boot.out

NOR version	RZ_T1_endat_nor_boot.out
Serial Flash version	RZ_T1_endat_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]

3.3.3 DS-5 from ARM

- Build environment

  ARM Development Studio 5 (DS-5) Version 5.25.0

  ARM Compiler 5.06 update 3

- Execution environment

  ULINK2（v2.01）

- How to build sample program

  1. Extract files from RZ_T1_endat.zip and copy the files to arbitrary holder
  2. Copy the following files of “RZ/T1 Encoder I/F Configuration Library” (for ARM DS-5) to each folder

     lib\ecl\r_ecl_rzt1.a

     inc\r_ecl_rzt1_if.h
  3. Launch DS-5
  4. Select [Window]menu -> [Show View] -> [Project Explorer]
  5. Click right button on [Project Explorer]view and then select [Import] of popup menu
  6. Select [General] -> [Existing Projects into Workspace] of [Import] dialog and then click [Next] button
  7. Click [Browse…] of [Import] dialog
  8. Select holder (the arbitrary holder of procedure 1 above) in [Browse For Folder] dialog and then click [OK].
  9. Select [Copy projects into workspace] of [Import] dialog
  10. Click [Finish] of [Import] dialog
  11. 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”)

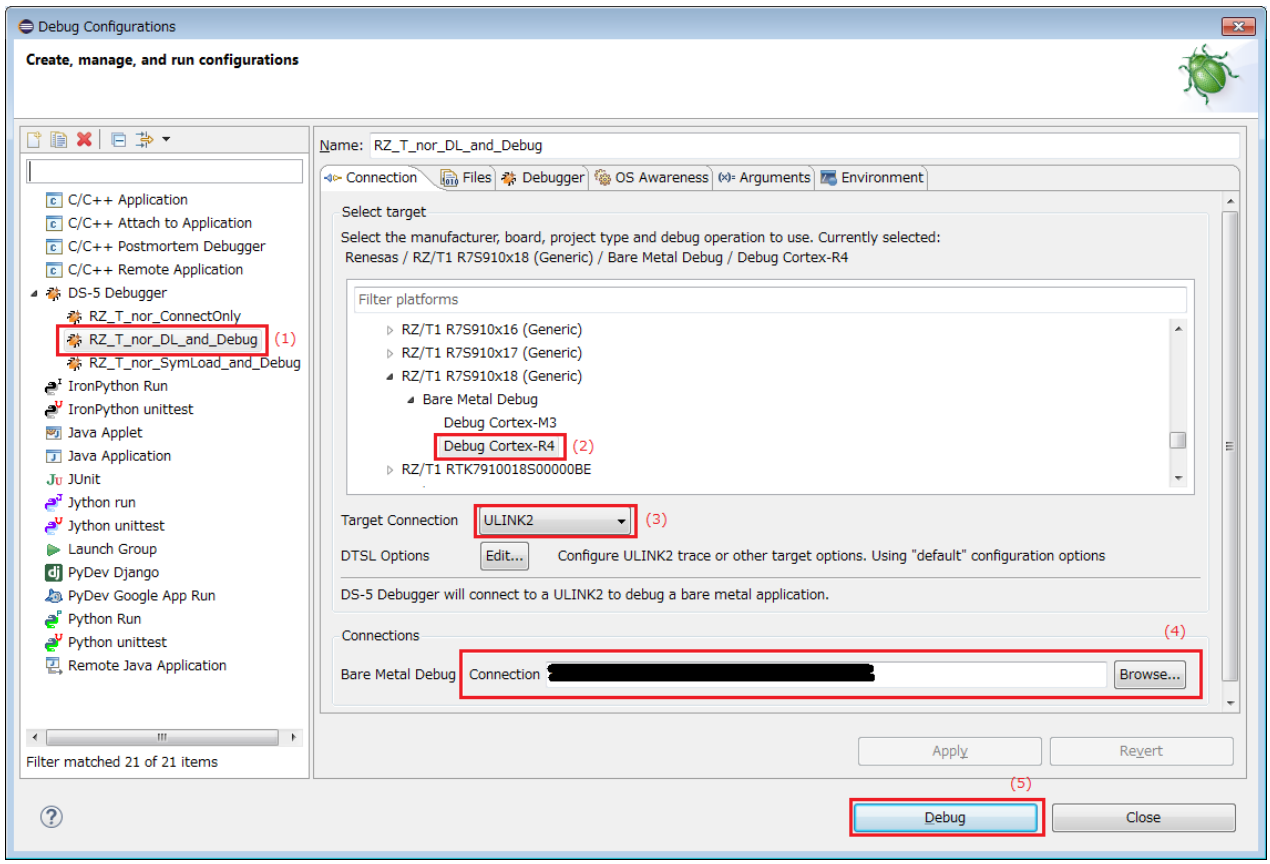
➢ How to execute sample program

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

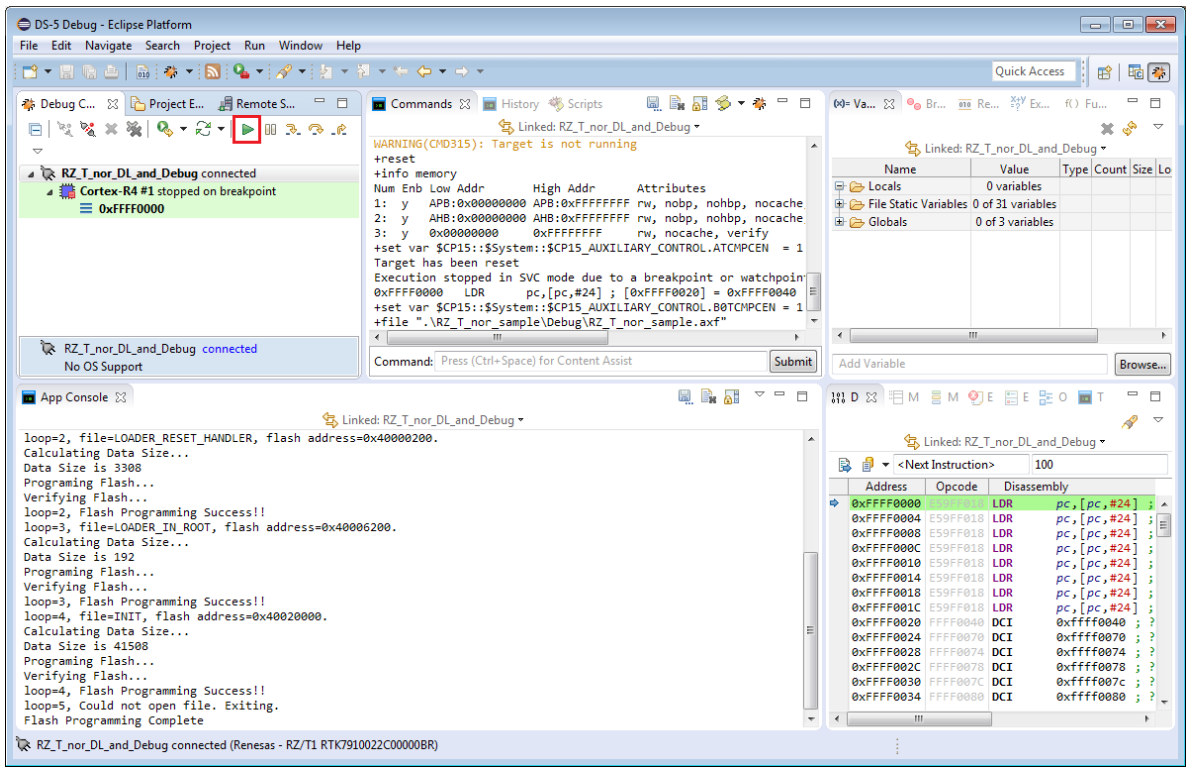
1. Open the debug configuration from the [Run] -> [Debug Configurations…], select the configuration window for “RZ_T_nor_DL_and_Debug”. (In case of serial flash, use the “RZ_T_sflash_DL_and_Debug” instead of the “RZ_T_nor_DL_and_Debug”)

   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.



3.3.4 e2 studio from RENESAS

- Build environment

  RENESAS e2 studio 5.2.0.020

  KPIT GNUARM-NONE-EABI Toolchain v16.01

- Execution environment

  J-Link BASE

- How to build sample program

  1. Extract files from RZ_T1_endat.zip and copy the files to arbitrary holder
  2. Copy the following files of “RZ/T1 Encoder I/F Configuration Library” (for KPIT GCC) to each folder

     lib\ecl\r_ecl_rzt1.a

     inc\r_ecl_rzt1_if.h
  3. Launch the e2studio
  4. Select [Window]menu -> [Show View] -> [Project Explorer]
  5. Click right button on [Project Explorer]view and then select [Import] of popup menu
  6. Select [General] -> [Existing Projects into Workspace] of [Import] dialog and then click [Next] button
  7. Click [Browse…] of [Import] dialog
  8. Select holder (the arbitrary holder of procedure 1 above) in [Browse For Folder] dialog and then click [OK].
  9. Select [Copy projects into workspace] of [Import] dialog
  10. Click [Finish] of [Import] dialog
  11. Select [Project] menu -> [Build All]

      Following file is generated.

      HardwareDebug\ RZ_T_nor_sample.x

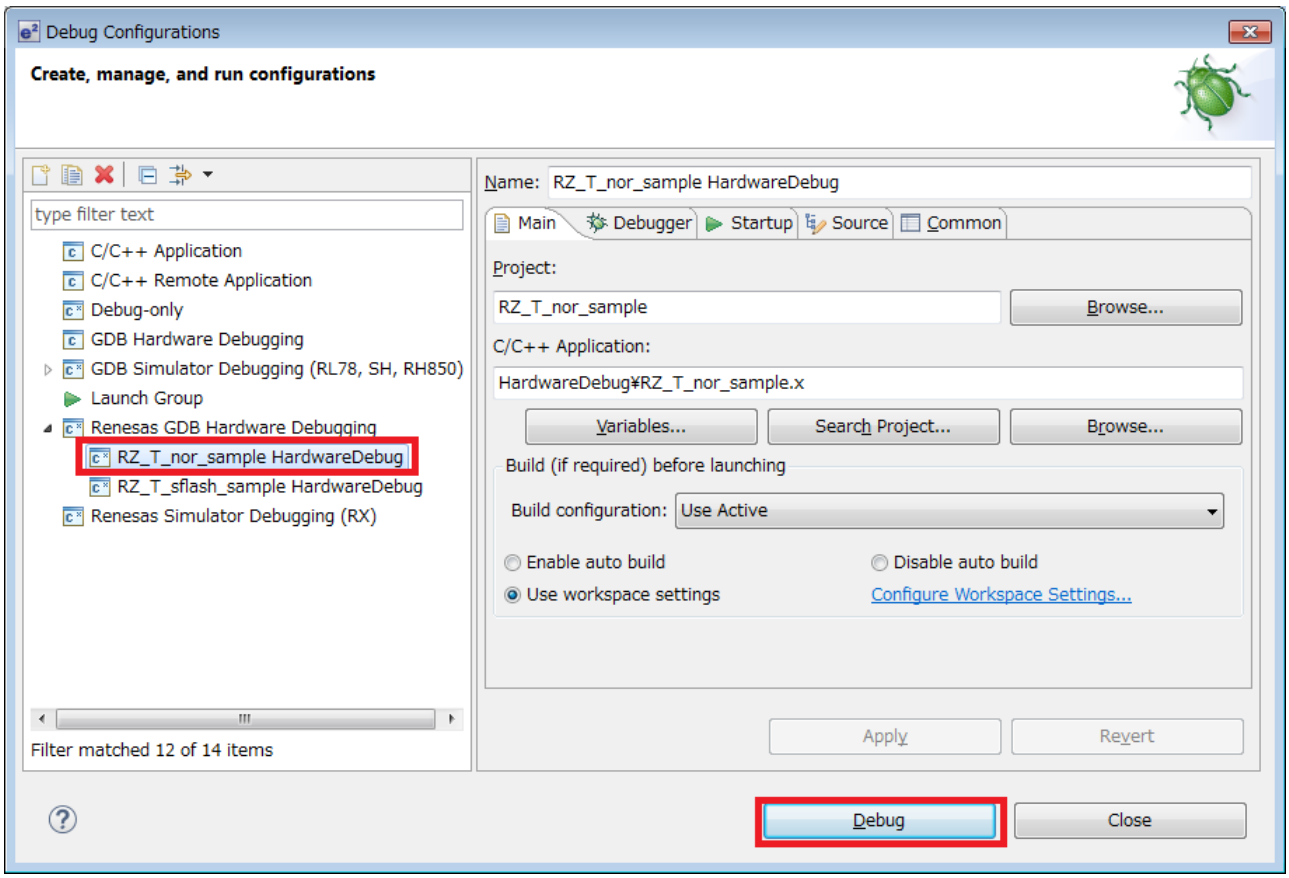
      (In case of serial flash, use the “RZ_T_sflash_sample.x” instead of the “RZ_T_nor_sample.x”)

➢ 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.

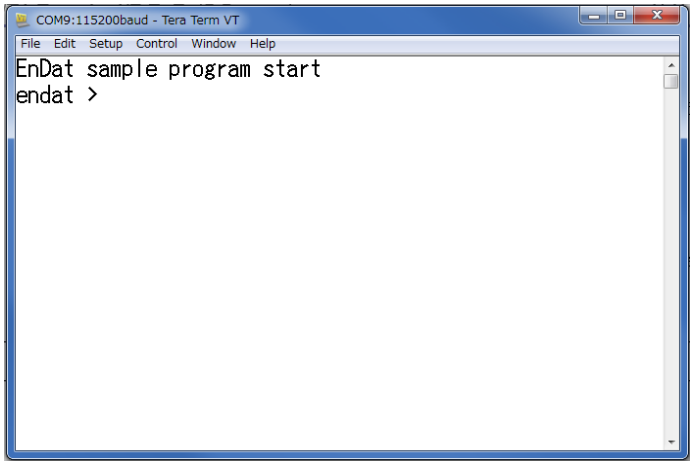
   (In case of serial flash, use the [RZ_T_sflash_sample_HardwareDebug] instead of the [RZ_T_nor_sample_HardwareDebug])



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

3.3.5 Execution result of sample program

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



4. Restriction

None.

5. Note

5.1 Processing time

Available time for user processing of Encoder I/F EnDat2.2 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 is indicated below.

The time used by the sample program is about 8.9 us (14%) of 62.5us, and available time for user processing is about 54us (86%).

Processing		Time		Occupancy rate
EnDat2.2 sample processing *2	Time setting registers for transmission	about 4.0 us	about 8.9us	14%
	Interrupt time	about 4.9 us		
Available time for user processing		about 53.6 us *1		86%

Note 1. Communication time with the encoder (Clock frequency: 8.33 MHz, tST: 2 us, mode command: 0 x 07, position value: 23 bits) is 21us of available time for user processing. For more information, refer to the “RZ/T1 Group EnDat 2.2 Interface (EnDat) User’s Manu” and the “EnDat specification” available on request from HEIDENHAIN CO.
Note 2. Initial setting time is not included.

5.2 Documentation mistake

The RZ/T1 Group EnDat Sample Program Application Note has the following mistakes.

- 5.11.1 Operation Outline
  - mistake

    (1) Send the request input from the debugger terminal I/O to the EnDat encoder (ECN1023).

    (2) Indicates the data received from the EnDat encoder (ECN1023) in the debugger terminal I/O.

  - correction

    (1) Send the request received from the SCIFA to EnDat encoder (ECN 1023).

    (2) Data received from EnDat encoder (ECN 1023) is output from the SCIFA.