RZ/T1 Group

Encoder I/F HIPERFACE DSL application package

R01AN3595EJ0110
Rev.1.10
September 27, 2018

Summary

This document explains about RZ/T1 Encoder I/F HIPERFACE DSL application package.

To use this application package, obtain release package of “RZ/T1 Encoder I/F Configuration Library” on the Renesas Electronics website.

Device that HIPERFACE DSL functionality is checked

RZ/T1 CPU Board (RTK7910018C00000BE)

Version History

Ver.	Date	Description	Note
1.10	September 2018	Updated User's Manual to Ver.1.31 Refer to the REVISION HISTORY	
		Updated sample program ● Fixed the iodefine_hfdsl.h file ● Changed the operation procedure of DS-5 and e2 studio.	
1.00	April 2018	Updated User's Manual to Ver.1.30 Refer to the REVISION HISTORY	
		Updated sample program to Ver.1.5 ● Fixed interrupt processing	
		Updated note of this document ● Fast position after protocol initialization	
0.94	February 2018	Updated User's Manual to Ver.1.30 (Preliminary version) Refer to the REVISION HISTORY	
		Updated sample program to Ver.1.4 ● Fixed the protocol initialization processing ● Fixed interrupt processing ● Supported the EXLEN, EXTRA register	
		Updated configuration data to Ver.1.3 ● Supported the EXTRA bit	
0.93	December 2017	Updated User's Manual to Ver.1.11 ● Added register that set stuffing processes ● Added descriptions of short message	
0.92	December 2017	Updated sample program to Ver.1.3 ● Fixed the protocol initialization processing ● Fixed interrupt processing	
		Updated configuration data to Ver.1.2 ● Fixed data of the quality monitor ● Fixed data of the equalization ● Fixed frame output timing in SYNC mode ● Fixed so that the stuffing value does not exceed the upper limit	
		Updated Application Note to Ver.1.20 Refer to the REVISION HISTORY	
0.9	September 2017	Updated sample program to Ver.1.20 ● Fixed the protocol initialization processing	
		Updated configuration data to Ver.1.1 ● Fixed RSSI processing ● Fixed line delay processing ● Fixed update timing of MAXDEVR register ● Fixed estimator processing ● Fixed the abnormal data output processing of RAW_FIFO function	

Ver.	Date	Description	Note
0.8	June 2017	Updated sample program to Ver.1.0 ● Be compatible with EWARM form IAR systems and DS-5 from ARM	
		Updated configuration data to Ver.1.0 ● Corresponds to all functions described in user's manual from Ver.0.1	
		Updated User’s Manual to Ver.1.10 Refer to the REVISION HISTORY	
		Newly created Application Note	
0.1	January 2017	Newly created	

Table of contents

1. Contents of package

Contents of this package are described in this chapter.

1.1 Software

・Source code

No.	Title	Version
1	A set of RZ/T1 HIPERFACE DSL sample driver code	1.5

・Configuration data

No.	Title	Version
1	RZ/T1 Encoder I/F Configuration Data (HIPERFACE DSL)	1.3

1.2 Document

No.	Document name	Ver.	File name
1	RZ/T1 Encoder I/F HIPERFACE DSL application package release note	1.10	(English) r01an3595ej0110-rzt1.pdf (this document) (Japanese) r01an3595jj0110-rzt1.pdf
2	RZ/T1 Group HIPERFACE DSL (HFDSL) Interface User's Manual	1.31	(English) r01uh0731ej0131-rzt1-hfdsl.pdf (Japanese) r01uh0731jj0131-rzt1-hfdsl.pdf (Chinese) r01uh0731cj0131-rzt1-hfdsl.pdf
3	RZ/T1 Group HIPERFACE DSL Sample Program APPLICATION NOTE	1.30	(English) r01an3869ej0130-rzt1-hfdsl.pdf (Japanese) r01an3869jj0130-rzt1-hfdsl.pdf (Chinese) r01an3869cj0130-rzt1-hfdsl.pdf

2. File Structures

File structures and contents of this package are described below.

```
Top
├──r01an3595ej0110-rzt1.pdf
├──r01an3595jj0110-rzt1.pdf
└──workspace
      ├──Software
      │    ├──iccarm
      │    │    └──RZ_T1_hfdsl.zip       :A set of RZ/T1 HIPERFACE DSL sample driver code (IAR)
      │    ├──armcc
      │    │    └──RZ_T1_hfdsl.zip       :A set of RZ/T1 HIPERFACE DSL sample driver code (DS-5)
      │    └──kpitgcc
      │         └──RZ_T1_hfdsl.zip       :A set of RZ/T1 HIPERFACE DSL sample driver code (e2 studio)
      └──Documentation
           ├──r01an3869cj0130-rzt1-hfdsl.pdf
           ├──r01an3869ej0130-rzt1-hfdsl.pdf
           ├──r01an3869jj0130-rzt1-hfdsl.pdf
           ├──r01uh0731cj0131-rzt1-hfdsl.pdf
           ├──r01uh0731ej0131-rzt1-hfdsl.pdf
           └──r01uh0731jj0131-rzt1-hfdsl.pdf
```

The file structures of "RZ_T1_hfdsl.zip" are indicated below.

```
Top folder
    inc
        iodefine.h            RZ/T1 register definition file
        iodefine_hfdsl.h      HIPERFASE DSL register definition file
        r_hfdsl_rzt1_dat.h    r_hfdsl_rzt1.dat header file
        r_hfdsl_rzt1_if.h     HFDSL sample driver header file
    lib
        ecl
            r_hfdsl_rzt1.dat        Multi-Protocol Encoder IF (HIPERFASE DSL mode) Configuration data
    src
        common
            Common sources including initial settings
        drv
            hfdsl
                r_hfdsl_rz1.c             HIPERFACE DSL sample program
                r_hfdsl_rz1_config.h      HIPERFACE DSL sample program
            scifa_uart
                scifa_uart.c              SCIFA sample driver
                scifa_uart_userdef.c      SCIFA sample driver
        sample
            main.c              Sample program
            hfdsl_dat.asm       Linker setting file of the Configuration data *1
            siorw.c             SCIFA Sample program
            siochar.c           SCIFA Sample program
```

Note 1: file for DS-5 / e2 studio
DS-5: hfdsl_dat.s
e2 studio: hfdsl_dat_asm

3. Information about HIPERFACE DSL sample program

This chapter describes information to use a set of HIPERFACE DSL 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]
HFDSL sample driver	Code		3052	3980	5752
	Data (with initial value)		8	34	8
	Data (without initial value)		87	64	96
	Constant Data		96	96	108
	Stack size of function	R_HFDSL_Open	60	60	88
		R_HFDSL _Close	16	28	56
		R_HFDSL _Control	48	120	128
		R_HFDSL _GetVersion	0	0	4
		hfdsl_int_nml_isr	128+n *1	56+n *1	80+n *1
		hfdsl_int_err_isr	112+n *1	48+n *1	56+n *1
HFDSL Configuration data	Code		0	0	0
	Data (with initial value)		0	0	0
	Data (without initial value)		0	0	0
	Constant Data		43908	43908	43908
Sample program	Code		1904	2580	3716
	Data (with initial value)		32	59	30
	Data (without initial value)		316	288	320
	Constant Data		908	40	901

Note 1. "n" is the Maximum stack size of user defined callback functions that are registered to R_HFDSL_Control function.

3.2 Hardware information

3.2.1 Device

RZ/T1

3.2.2 Target Board

(1) 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

SW4-3: 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 before 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". Download the latest version of USB serial port driver for PC from the link below.

https://www.renesas.com/jp/ja/software/D6000699.html

3.3.2 EWARM from IAR systems

- Build environment

  IAR Embedded Workbench for ARM v8.20.2

- Execution environment

  I-jet

- How to build sample program

  1. Extract files from RZ_T1_hfdsl.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_hfdsl_boot\RZ_T1_hfdsl_****_boot.eww

NOR version	RZ_T1_hfdsl_nor_boot.eww
Serial Flash version	RZ_T1_hfdsl_serial_boot.eww

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

     Following file is generated.

     RZ_T1_hfdsl_boot\Debug\Exe\RZ_T1_hfdsl_****_boot.out

NOR version	RZ_T1_hfdsl_nor_boot.out
Serial Flash version	RZ_T1_hfdsl_serial_boot.out

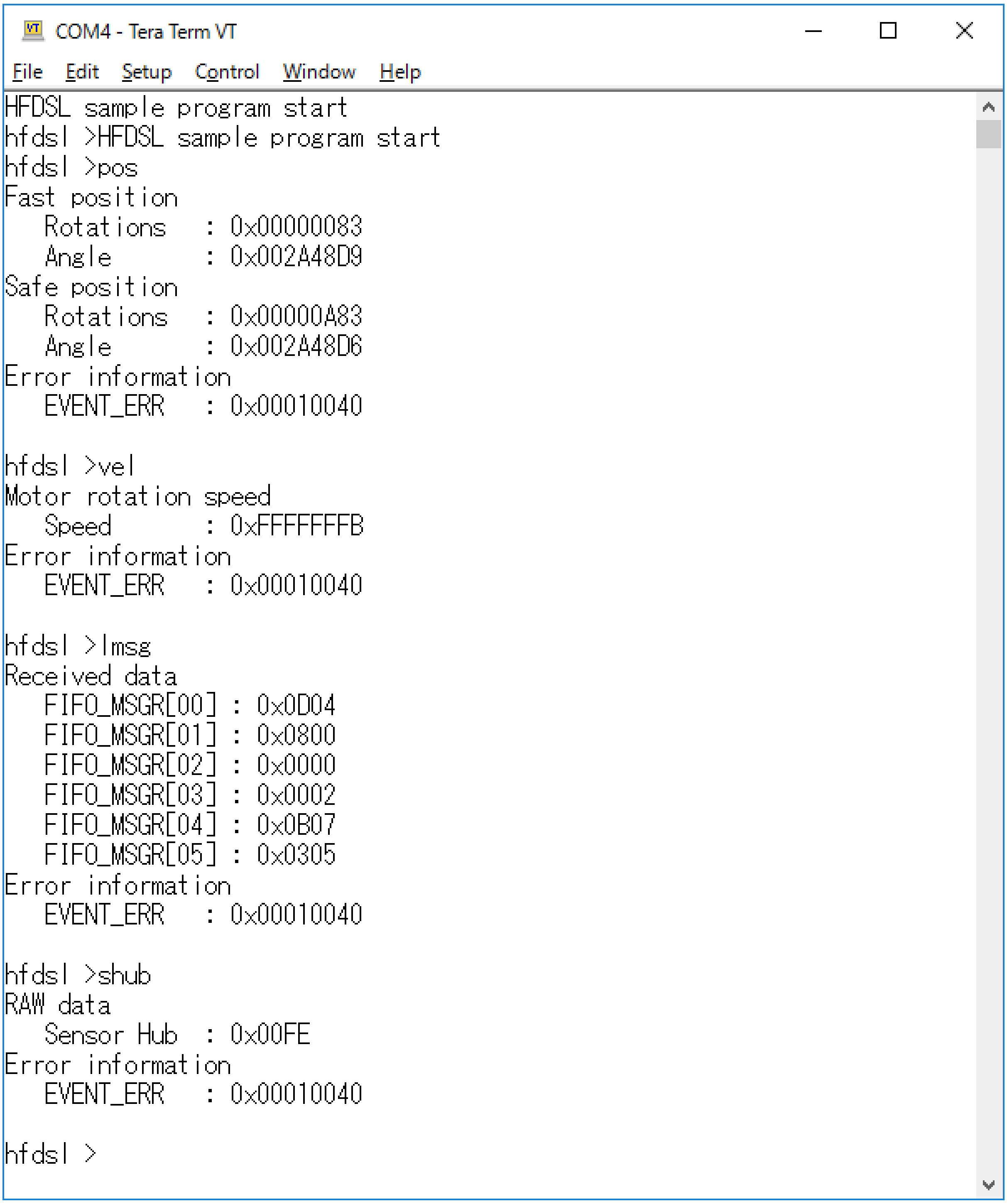
- How to execute sample program

  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 HIPERFACE DSL Sample Program Application Note” about the console command.



```
HFDSL sample program start
hfdsl >HFDSL sample program start
hfdsl >pos
Fast position
   Rotations   : 0x00000083
   Angle       : 0x002A48D9
Safe position
   Rotations   : 0x00000A83
   Angle       : 0x002A48D6
Error information
   EVENT_ERR   : 0x00010040

hfdsl >vel
Motor rotation speed
   Speed       : 0xFFFFFFFB
Error information
   EVENT_ERR   : 0x00010040

hfdsl >lmsg
Received data
   FIFO_MSGR[00] : 0x0D04
   FIFO_MSGR[01] : 0x0800
   FIFO_MSGR[02] : 0x0000
   FIFO_MSGR[03] : 0x0002
   FIFO_MSGR[04] : 0x0B07
   FIFO_MSGR[05] : 0x0305
Error information
   EVENT_ERR   : 0x00010040

hfdsl >shub
RAW data
   Sensor Hub  : 0x00FE
Error information
   EVENT_ERR   : 0x00010040

hfdsl >
```

3.3.3 DS-5 from ARM

- Build environment

  ARM Development Studio 5 (DS-5) Version 5.26.2

  ARM Compiler 5.06 update 4

- 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_hfdsl.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”)

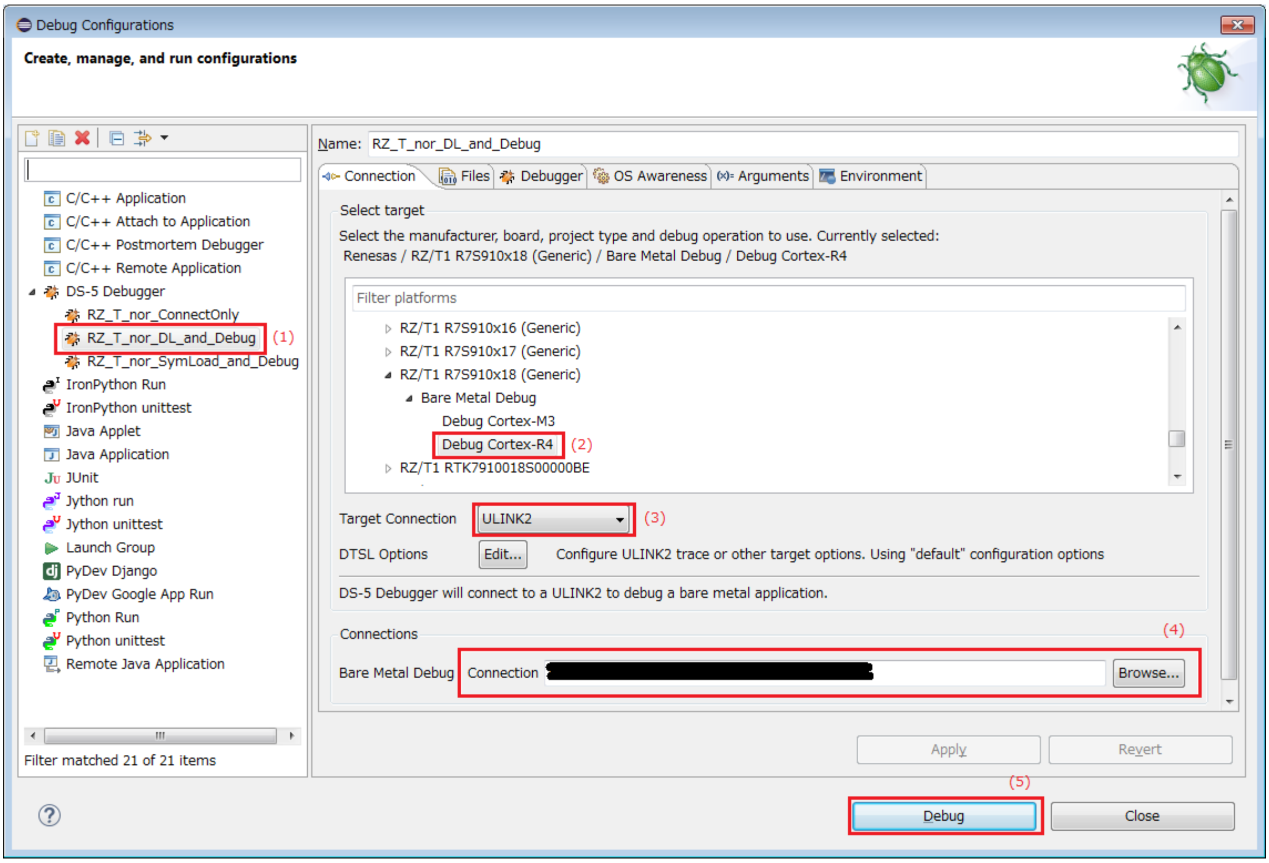
- How to execute sample program

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.



Debug Configurations

Create, manage, and run configurations

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- DS-5 Debugger
  - RZ_T_nor_ConnectOnly
  - RZ_T_nor_DL_and_Debug (1)
  - RZ_T_nor_SymLoad_and_Debug
- IronPython Run
- IronPython unittest
- Java Applet
- Java Application
- JUnit
- Jython run
- Jython unittest
- Launch Group
- PyDev Django
- PyDev Google App Run
- Python Run
- Python unittest
- Remote Java Application

Filter matched 21 of 21 items

Name: RZ_T_nor_DL_and_Debug

Select target

Select the manufacturer, board, project type and debug operation to use. Currently selected:
Renesas / RZ/T1 R7S910x18 (Generic) / Bare Metal Debug / Debug Cortex-R4

- RZ/T1 R7S910x16 (Generic)
- RZ/T1 R7S910x17 (Generic)
- RZ/T1 R7S910x18 (Generic)
  - Bare Metal Debug
    - Debug Cortex-M3
    - Debug Cortex-R4 (2)
- RZ/T1 RTK7910018S00000BE

Target Connection: ULINK2 (3)

DTSL Options: Edit... Configure ULINK2 trace or other target options. Using "default" configuration options

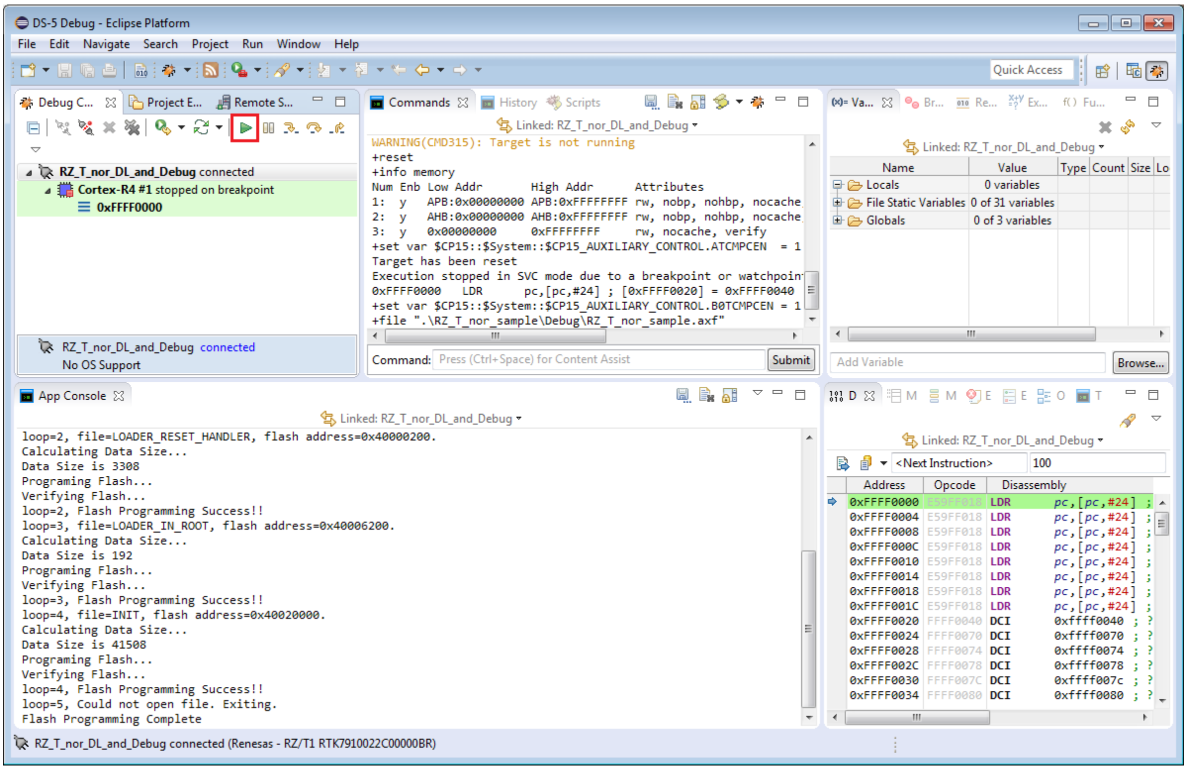
DS-5 Debugger will connect to a ULINK2 to debug a bare metal application.

Connections (4)

Bare Metal Debug Connection [redacted] Browse...

(5)

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.


DS-5 Debug - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Quick Access
Debug C... Project E... Remote S...
RZ_T_nor_DL_and_Debug connected
Cortex-R4 #1 stopped on breakpoint
0xFFFF0000
RZ_T_nor_DL_and_Debug connected
No OS Support
Commands History Scripts
Linked: RZ_T_nor_DL_and_Debug
Command: Press (Ctrl+Space) for Content Assist
Submit
Va... Br... Re... Ex... Fu...
Linked: RZ_T_nor_DL_and_Debug
Add Variable
Browse...
App Console
Linked: RZ_T_nor_DL_and_Debug
D M M E E O T
Linked: RZ_T_nor_DL_and_Debug
<Next Instruction>
100

```
WARNING(CMD315): Target is not running
+reset
+info memory
Num Enb Low Addr       High Addr      Attributes
1:  y   APB:0x00000000 APB:0xFFFFFFFF rw, nobp, nohbp, nocache
2:  y   AHB:0x00000000 AHB:0xFFFFFFFF rw, nobp, nohbp, nocache
3:  y   0x00000000     0xFFFFFFFF     rw, nocache, verify
+set var $CP15::$System::$CP15_AUXILIARY_CONTROL.ATCMPCEN  = 1
Target has been reset
Execution stopped in SVC mode due to a breakpoint or watchpoin
0xFFFF0000   LDR      pc,[pc,#24] ; [0xFFFF0020] = 0xFFFF0040
+set var $CP15::$System::$CP15_AUXILIARY_CONTROL.B0TCMPCEN = 1
+file ".\RZ_T_nor_sample\Debug\RZ_T_nor_sample.axf"
```

Name	Value	Type	Count	Size	Lo
Locals	0 variables				
File Static Variables	0 of 31 variables				
Globals	0 of 3 variables				

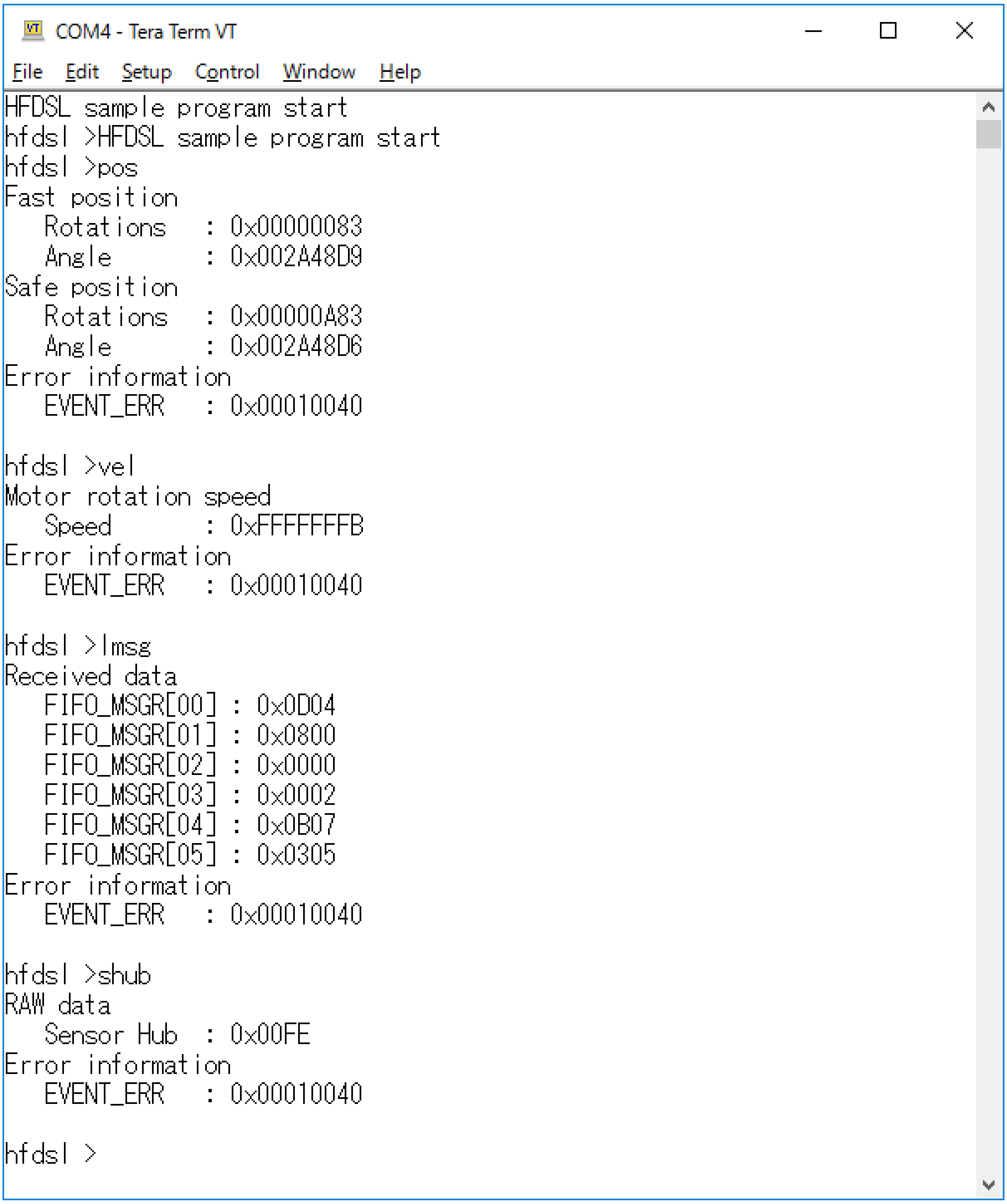
```
loop=2, file=LOADER_RESET_HANDLER, flash address=0x40000200.
Calculating Data Size...
Data Size is 3308
Programing Flash...
Verifying Flash...
loop=2, Flash Programming Success!!
loop=3, file=LOADER_IN_ROOT, flash address=0x40006200.
Calculating Data Size...
Data Size is 192
Programing Flash...
Verifying Flash...
loop=3, Flash Programming Success!!
loop=4, file=INIT, flash address=0x40020000.
Calculating Data Size...
Data Size is 41508
Programing Flash...
Verifying Flash...
loop=4, Flash Programming Success!!
loop=5, Could not open file. Exiting.
Flash Programming Complete
```

```
Address     Opcode    Disassembly
0xFFFF0000  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF0004  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF0008  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF000C  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF0010  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF0014  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF0018  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF001C  E59FF018  LDR       pc,[pc,#24]  ;
0xFFFF0020  FFFF0040  DCI       0xffff0040 ; ?
0xFFFF0024  FFFF0070  DCI       0xffff0070 ; ?
0xFFFF0028  FFFF0074  DCI       0xffff0074 ; ?
0xFFFF002C  FFFF0078  DCI       0xffff0078 ; ?
0xFFFF0030  FFFF007C  DCI       0xffff007c ; ?
0xFFFF0034  FFFF0080  DCI       0xffff0080 ; ?
```

RZ_T_nor_DL_and_Debug connected (Renesas - RZ/T1 RTK7910022C00000BR)

- Execution result of sample program

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



```
HFDSL sample program start
hfdsl >HFDSL sample program start
hfdsl >pos
Fast position
   Rotations   : 0x00000083
   Angle       : 0x002A48D9
Safe position
   Rotations   : 0x00000A83
   Angle       : 0x002A48D6
Error information
   EVENT_ERR   : 0x00010040

hfdsl >vel
Motor rotation speed
   Speed       : 0xFFFFFFFB
Error information
   EVENT_ERR   : 0x00010040

hfdsl >lmsg
Received data
   FIFO_MSGR[00] : 0x0D04
   FIFO_MSGR[01] : 0x0800
   FIFO_MSGR[02] : 0x0000
   FIFO_MSGR[03] : 0x0002
   FIFO_MSGR[04] : 0x0B07
   FIFO_MSGR[05] : 0x0305
Error information
   EVENT_ERR   : 0x00010040

hfdsl >shub
RAW data
   Sensor Hub  : 0x00FE
Error information
   EVENT_ERR   : 0x00010040

hfdsl >
```

3.3.4 e2 studio from RENESAS

- Build environment

  RENESAS e2 studio 6.1.0.020

  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_hfdsl.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

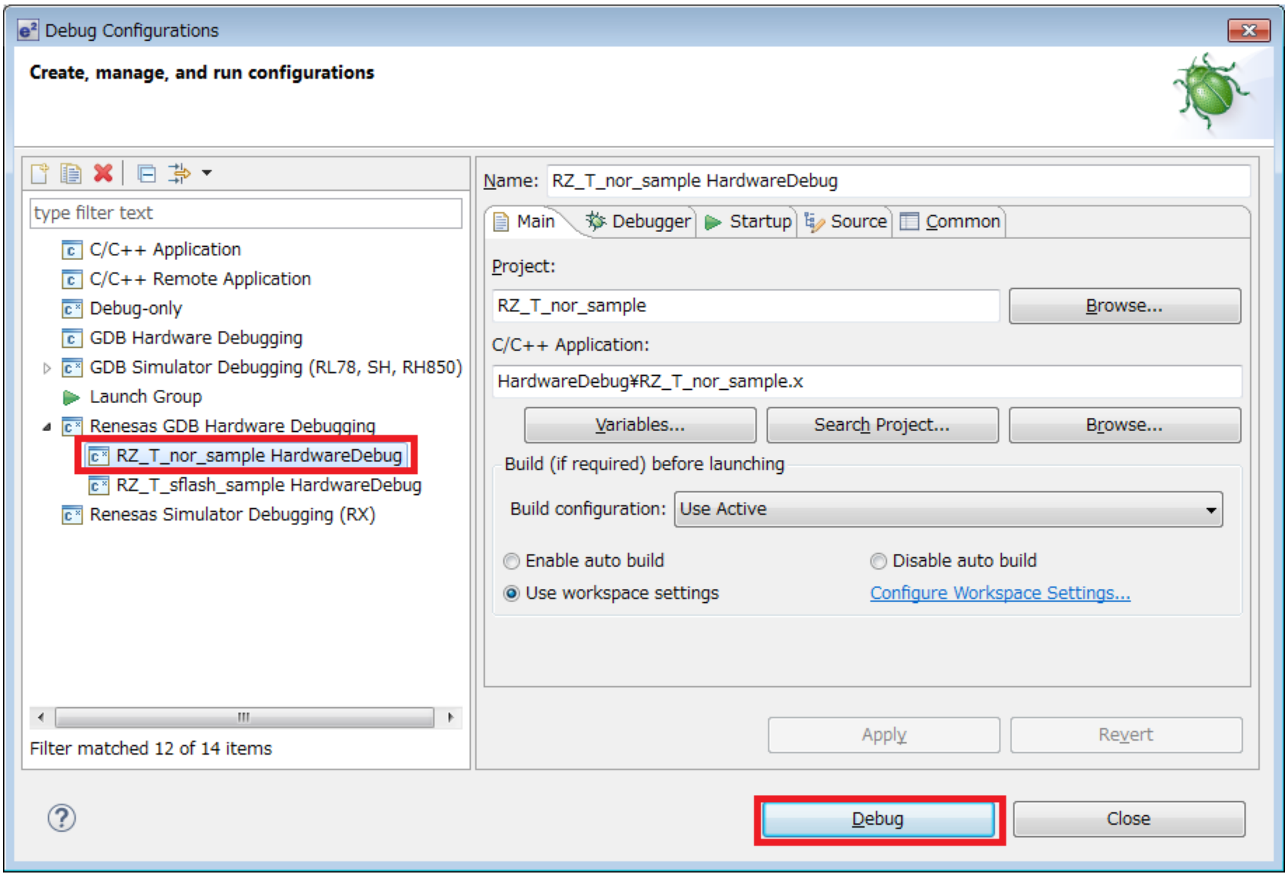
     (In case of serial flash, use the “RZ_T_sflash_sample.x” instead of the “RZ_T_nor_sample.x”)

- 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])



Debug Configurations

Create, manage, and run configurations

Name: RZ_T_nor_sample HardwareDebug

Project: RZ_T_nor_sample

C/C++ Application: HardwareDebug¥RZ_T_nor_sample.x

Build (if required) before launching

Build configuration: Use Active

○ Enable auto build ○ Disable auto build

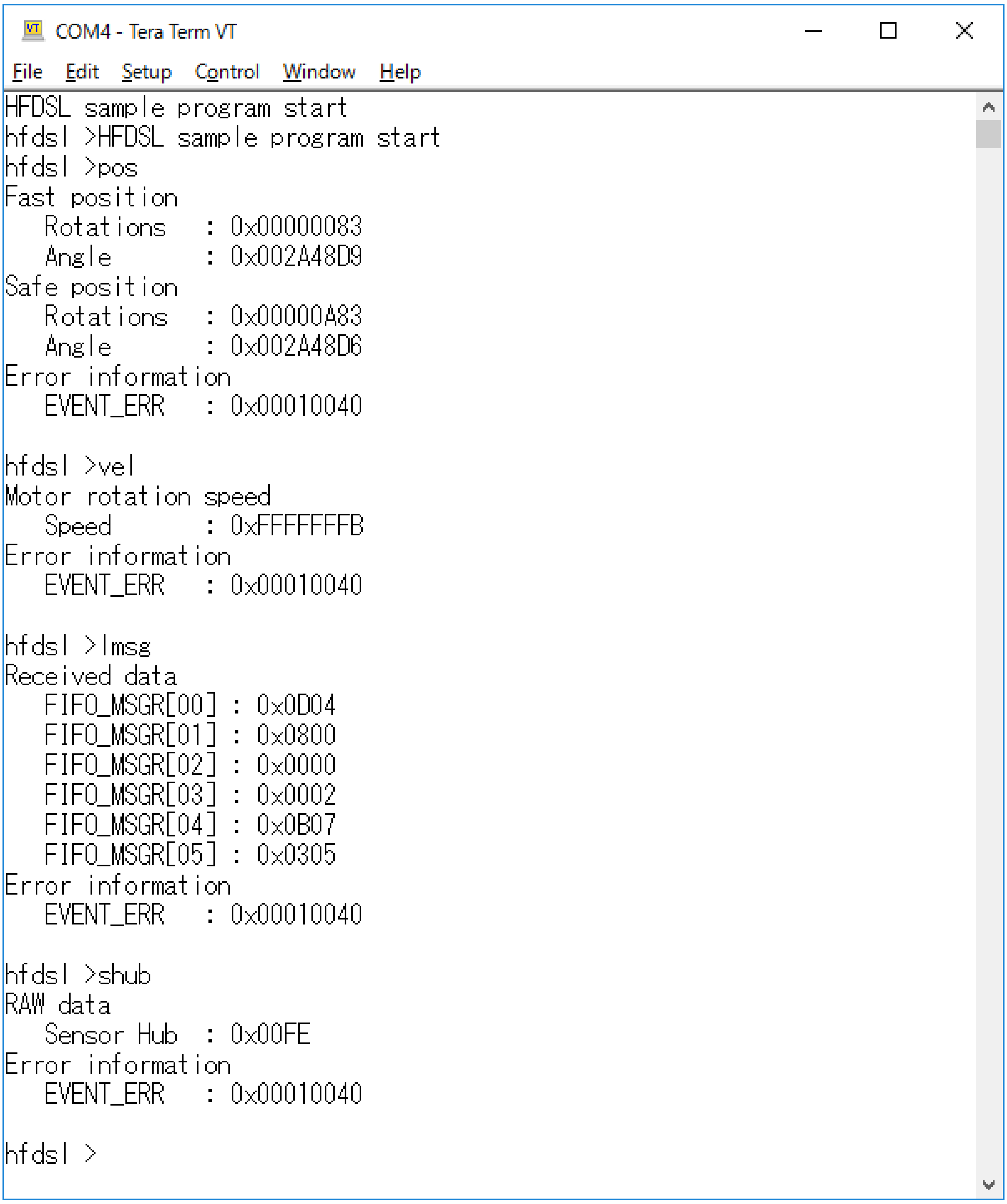
● Use workspace settings

Filter matched 12 of 14 items

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 HIPERFACE DSL Sample Program Application Note” about the console command.



```
HFDSL sample program start
hfdsl >HFDSL sample program start
hfdsl >pos
Fast position
   Rotations   : 0x00000083
   Angle       : 0x002A48D9
Safe position
   Rotations   : 0x00000A83
   Angle       : 0x002A48D6
Error information
   EVENT_ERR   : 0x00010040

hfdsl >vel
Motor rotation speed
   Speed       : 0xFFFFFFFB
Error information
   EVENT_ERR   : 0x00010040

hfdsl >lmsg
Received data
   FIFO_MSGR[00] : 0x0D04
   FIFO_MSGR[01] : 0x0800
   FIFO_MSGR[02] : 0x0000
   FIFO_MSGR[03] : 0x0002
   FIFO_MSGR[04] : 0x0B07
   FIFO_MSGR[05] : 0x0305
Error information
   EVENT_ERR   : 0x00010040

hfdsl >shub
RAW data
   Sensor Hub  : 0x00FE
Error information
   EVENT_ERR   : 0x00010040

hfdsl >
```

4. Restriction

None.

5. Note

5.1 Processing time

Available time for user processing of Encoder I/F HFDSL 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 4 us (6.1%) of 62.5 us, and available time for user processing is about 58.5 us (92.6%).

Processing		Time		Occupancy rate
HFDSL sample processing *2	Time obtaining fast position	about 2 us	about 4 us	6.1 %
	Interrupt time*3	about 2 us		
Available time for user processing		about 58.5 us *1		93.9 %

Note 1. Communication time with the encoder (when the number of data fields is three) is 11.5 us of available time for user processing. For more information, refer to the "RZ/T1 Group HFDSL Interface (HFDSL) User's Manual".

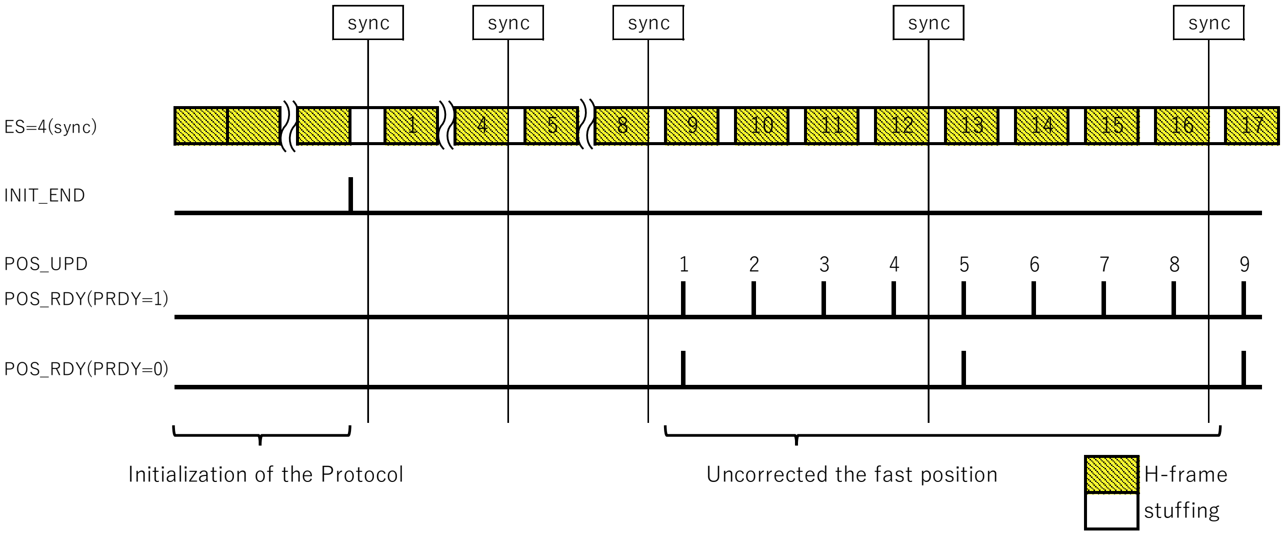
Note2. Initial setting time is not included.

Note3. Time in the case of POS_RDY bit only interrupt enabled.

5.2 Fast position after protocol initialization

Fast position is corrected by the safe position and speed data. However, the fast position of 8 times after protocol initialization is not corrected.

Example


sync
sync
sync
sync
sync
ES=4(sync)
1
4
5
8
9
10
11
12
13
14
15
16
17
INIT_END
POS_UPD
POS_RDY(PRDY=1)
1
2
3
4
5
6
7
8
9
POS_RDY(PRDY=0)
Initialization of the Protocol
Uncorrected the fast position
H-frame
stuffing