APPLICATION NOTE

RX Family

R01AN1983EJ0112
Rev.1.12
Mar 31, 2017

PTP Get Synchronous Time Using Firmware Integration Technology Modules

Introduction

This document explains one of the EPTPC FIT (Firmware Integration Technology) module [1] usage examples. This example is getting the synchronous time corrected by the time synchronization based on the PTP (Precision Time Protocol) defined by the IEEE1588-2008 specification [2].

Target Device

This API supports the following device.

- RX64M Group

- RX71M Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Overview

This document explains one of the typical usage examples of the EPTPC FIT module (hereafter PTP driver). This example gets the synchronous times from the local clock counter of the EPTPC peripheral module (EPTPC) and save them to a USB memory. The time synchronization protocol is applied to the PTP. User can verify and compare the time differences inter each clock devices from the data of USB memories.

1.1 PTP Get Synchronous Time Using FIT Modules

This module is implemented in a project and used as the application example of the PTP driver.

1.2 Related documents

[1] RX Family EPTPC Module Using Firmware Integration Technology, Rev.1.13, Document No. R01AN1943EJ0113, Mar 31, 2017

[2] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Revision of IEEE Std 1588-2008, Mar 2008

[3] RX Family Ethernet Module Using Firmware Integration Technology, Rev.1.12, Document No. R01AN2009EJ0112, Nov 11, 2016

[4] Renesas USB MCU USB Basic Host and Peripheral firmware Using Firmware Integration Technology, Rev.1.10, Document No. R01AN2025EJ0110, Dec 26, 2015

[5] Renesas USB MCU USB Host Mass Storage Class Driver (HMSC) Using Firmware Integration Technology, Rev.1.10, Document No. R01AN2026EJ0110, Dec 26, 2015

[6] RX Family Open Source FAT File System [M3S-TFAT-Tiny] Module Firmware Integration Technology, Rev.3.00, Document No. R20AN0038EJ0300, Apr 01, 2014

[7] HMI Expansion Board User’s Manual, Rev. 1.01, Document No. R0K50564MB001BR, Jan 16, 2015

[8] RX64M Group Renesas Starter Kit+ User’s Manual For e^2 studio, Rev. 1.10, Document No. R20UT2593EG0110, Jun 25, 2015

[9] RX71M Group Renesas Starter Kit+ User’s Manual, Rev. 1.00, Document No. R20UT3217EG0100, Jan 23, 2015

1.3 Terms and Abbreviations

Please refer to EPTPC FIT module application note (Sec.1.3) [1].

1.4 Hardware Structure

This example uses the Ethernet peripheral modules of the RX64M/71M. The Ethernet peripheral modules are composed of the EPTPC, the PTP Host interface peripheral module (PTPEDMAC), dual channel Ethernet MAC ones (ETHERC (CH0), ETHERC (CH1)) and dual channel Ethernet Host interface ones (EDMAC (CH0), EDMAC (CH1)).

In detail, please refer to RX64M/71M Group User’s Manual: Hardware.

1.5 Software Structure

This sample is operations example of the application layer applied to plural FIT modules. Those operations are to get a PTP configuration such as MAC address, IP address, the kind of Clock, Master or Slave and delay mechanism (P2P or E2E) from a USB memory, set the PTP configuration to the PTP driver, get synchronous times from the USB memory, save the synchronous time to the USB memory, and control the PTP protocol sequences using the PTP driver. The PTP driver always should be used with the Ether driver [3]. TCP/IP middle ware does not include in this example. Therefore, user needs to implement TCP/IP middle ware when this example applied to the TCP/IP system. The USB Host driver [4], [5], which is implemented ATA interface and USB mass storage class, does the USB memory access operations such as memory mount, data read, write, USB memory detection, etc. M3S-TFS-Tiny [6] is implemented as the FAT file system and it does the file access operations such as file open, close, read, write, etc. Figure 1.1 shows the software structure of this sample.

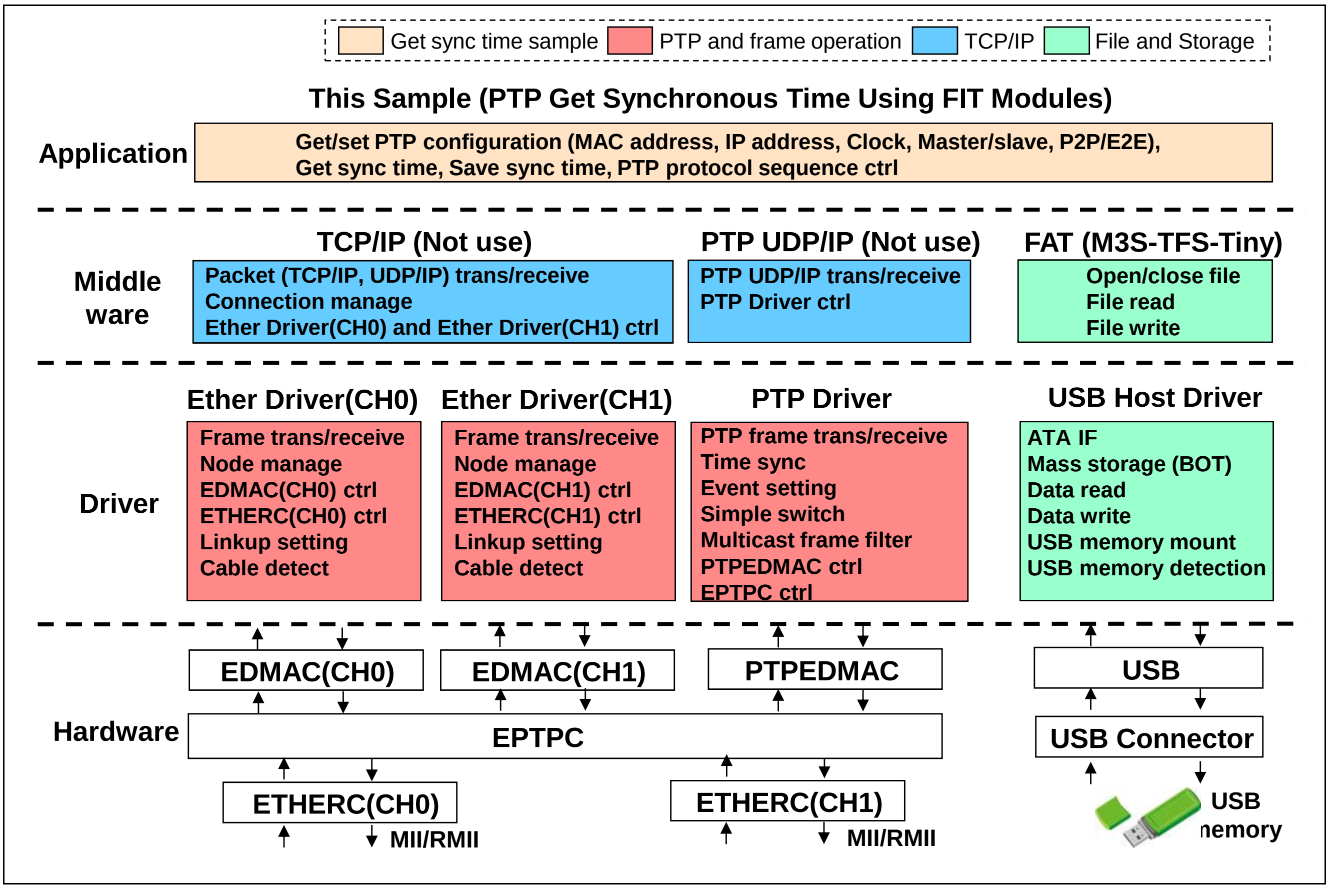


Figure 1.1 Software structure of this sample

1.6 File Structure

This sample codes are stored the “demo_src” and low hierarchical folders. Figure 1.2 shows the file structure of this sample. The execute file of the evaluate application is also equipped and stored in the “external_appli” folder. More information of the evaluate application, please refer to Sec 3.3. As for the detailed information of the FIT based modules (BSP, Ethernet Driver, PTP Driver, FAT file system and USB Drivers), please refer to the documentation of the each FIT module.

```
demo_src: Main operation and configuration
|  sample_main.c
|  sample_main.h
|
+ --- led_7seg: 7segment led control
|      led_7seg.c
|      led_7seg.h
|
+ --- sync_if: PTP synchronize operation
|      sync_if.c
|      sync_if.h
|
+ --- tfat_if: File system IF to USB driver
|      file_if.c
|      file_if.h
|      r_data_file.c
|      r_data_file.h
|      r_tfat_drv_if.c ;USB driver interface
|
+ --- usb_if: USB Host memory access control
|      r_usb_hmsc_defep.c
|      usb_memory_access.c
|
+ --- usb_memory_sample: Samples of USB memory data
|      + --- PARAM:
|             + --- ID, mode, port: ID=0 to 2
|                    + --- PARAM.txt; parameter file
|
+ -- external_appli: Evaluate application
|      SendTestPacket.exe
|
+ --- usr: LED control
|      led.c
|      led.h
```

```
r_bsp: BSP (Board Support Package) FIT module

r_cmt_rx: CMT (Compare Match Timer) FIT module

r_config: configuration setting of FIT modules
|      r_bsp_config.h
|      r_bsp_interrupt_config.h
|      r_cmt_rx_config
|      r_ether_rx_config.h
|      r_ptp_rx_config.h
|      r_usb_basic_config.h
|      r_usb_hmsc_config.h

r_ether_rx: Ethernet Driver FIT module

r_ptp_rx: PTP Driver FIT module

r_tfat_rx: FAT file system (M3S-TFS-Tiny) FIT module
|      r_tfat_lib.h ;FAT library header file
|      + --- lib:  FAT library stored folder
|                r_mw_version.h ; middleware version information
|                r_stdint.h ; integer type definition
|                tfat_rx600_big.lib ; big endian
|                tfat_rx600_little.lib ; little endian
|

r_usb_basic: USB driver (USB basic operation) FIT module

r_usb_hmsc: USB driver (Host Mass Storage Class) FIT module
```

Figure 1.2 File structure of this example

2. Functional Information

This example is developed by the following principles.

2.1 Hardware Requirements

This driver requires your MCU supports the following feature:

- EPTPC
- ETHERC
- EDMAC
- PTPEDMAC
- CMT
- USB

2.2 Hardware Resource Requirements

This section details the hardware peripherals that this example requires. Unless explicitly stated, these resources must be reserved for the following driver, and the user cannot use them.

2.2.1 EPTPC Channel

This example uses the EPTPC. This resource needs to the synchronization based on the PTP.

2.2.2 ETHERC Channel

This example uses the ETHEC (CH0), ETHEC (CH1) or both depend on the kind of Clock (Node). Those resources need to the Ethernet MAC operations.

2.2.3 EDMAC Channel

This example uses the EDMAC (CH0), EDMAC (CH1) or both depend on the kind of Clock (Node). Those resources need to the CPU Host interface of standard Ethernet frame operations.

2.2.4 PTPEDMAC Channel

This example uses the PTPEDMAC. This resource need to the CPU Host interface of PTP message frame operations.

2.2.5 CMT Channel

This example uses the CMT. This resource need to update the 7 segment LED pattern.

2.2.6 USB Channel

This example uses the USB 2.0 FS Host/Function Module to read configuration parameter from the USB memory and write evaluation result data to the USB memory.

2.3 Software Requirements

This example depends on the following packages (FIT modules):

- r_bsp
- r_cmt_rx
- r_ether_rx
- r_ptp_rx
- r_tfat_rx
- r_usb_basic
- r_usb_hmsc

2.4 Supported Toolchains

This example is tested and works with the following toolchain:

- Renesas RX Toolchain v2.06.00

2.5 Header Files

Each functions call are accessed by including a single file, *sample_main.h, led_7seg.h, sync_if.h, file_if.h, r_data_file.h or led.h* which is supplied with this driver's project code.

2.6 Integer Types

This project uses ANSI C99. These types are defined in *stdint.h*.

2.7 Configuration Overview

The configuration options in this project are specified in *sample_main.h* and *r_data_file.h.* The option names and setting values are listed in the table below.

Configuration options	
#define LINK_CH - Default value = 0	Specify the Ethernet link channel at first. - When this is set to 0, Ethernet CH0 is selected. - When this is set to 1, Ethernet CH1 is selected.
#define NUM_CH - Default value = 1	Specify the number of using Ethernet channels. - Set 1 or 2 in the RX64M. If clock is BC or TC, you need set to 2.
#define TIMES_W10_INTERVAL - Default value = 32	Interval times of getting worst10 values. Set 0 to 255. (Recommend : 32 or more)
#define NUM_EVAL (5)	Total evaluation times.
#define NUM_UPDATE (10)	Total update times.
#define USE_7SEG_LED - defined	Use 7segment LED[1] or not. - When “USE_7SEG_LED” is defined, use 7segment LED. - When “USE_7SEG_LED” is not defined, not use 7segment LED.
#define PARAM_FILE - Default value “PARAM.txt"	Specify the file name and extension of the parameter file to store the synchronous configuration.
#define RESULT_DIR - Default value "SYNC"	Specify the directory name of the evaluation result file stored.
#define RESULT_FILE - Default value "TEST"	Specify the file name of the evaluation result file saved local clocks. This string is concatenated the file number assigned evaluation order and the file extension equal to “.txt”. Ex. “TEST_1.txt”, “TEST_2.txt”, , “TEST_5.txt”
#define EXTEND_FILE - Default value "EXT"	Specify the file name of the evaluation result file saved offsetFromMaster and meanPathDelay[2]. This string is concatenated the file number assigned evaluation order and the file extension equal to “.txt”. Ex. “EXT_1.txt”, “EXT_2.txt”, , “EXT_5.txt”
#define FILESIZE - Default value = 2048	Specify the FAT file system data buffer size - Set 2048 (default value) in the RX64M/71M.

[1] This is implemented on the optional HMI Expansion Board (refer to [7]).

[2] In case of master, offsetFromMaster and meanPathDelay are not got.

2.8 Data Structures

This section details the data structures that are used with the functions of this example. Those structures are located in *sample_main.h, sync_if.h* and *sync_if.c* as the prototype declarations.

```
/* Ether & USB access state */
typedef enum
{
   APL_START = 0, /* Operation start state */
   APL_READ,      /* USB memory read state */
   APL_COM,       /* Ether communication state */
   APL_WRITE,     /* USB memory write state */
   APL_STOP,      /* Operation stop state */
} APLState;
```

```
/* Synchronous configuration structure */
typedef struct {
   uint8_t  id;
   uint8_t  mac[2][6];
   uint8_t  ip[2][4];
   uint8_t  mode;
   uint8_t  ms[2];
   uint8_t  sync[2];
} SyncConfig;
```

```
/* Synchronous result structure */
typedef struct
{
   Timestamp    clk; /* local clock */
   TimeInterval ofm; /* offsetFromMaster */
   TimeInterval mpd; /* meanPathDelay */
} SyncResult;
```

```
/* Synchronous result save flag */
bool g_save_SyncFlag;
```

```
/* Synchronization stop flag */
bool g_stop_SyncFlag;
```

```
/* Synchronization error flag */
bool g_err_SyncFlag;
```

2.9 Return Values

This section describes return values of the functions of this example. Those return values are located in *sync_if.h*, *file_if.h and led_7seg.h* as the prototype declarations.

```
/* PTP synchronous operation return value */
typedef enum
{
   SYNCIF_TOUT = -3,   /* Timeout error */
   SYNCIF_ETHERR = -2, /* Standard Ether error */
   SYNCIF_ERR = -1,    /* General error */
   SYNCIF_OK = 0,
} syncif_t;
```

```
/* File access return value */
typedef enum
{
   FLIF_ERR = -1, /* General error */
   FLIF_OK = 0,
} flif_t;
```

```
/* 7segment LED driver return value */
LED_7SEG_OK (0) /* No error */
LED_7SEG_ERR (-1) /* General error */
```

3. Specification of This Example

3.1 Outline of Functions

The function of this example shows Table 3.1.

Table 3.1 Function of This Example

Item	Contents
main()	Main operation of this sample.
led_init()	Initialize user LED.
led_ctrl()	Update user LED pattern
usb_memory_start()	USB memory task start.
Sample_Task()	Sample application task.
PTPCom ()	PTP synchronize operation.
EINT_Trig_isr()	Trigger packet received interrupt handler.
SyncErr ()	PTP or Ether error operation.
GetLcClk()	Get and save local clock counter.
ReadPTPMsg()	Read PTP messages. If announce message is received, update Master port identity.
IsTrigPacket()	Check received frame is trigger packet or not.
SaveRes()	Save synchronize operation result.
led_callback()	Compare match timer callback function for 7seg led.
file_crt_dir()	Create evaluation result data directories. (mount USB memory)
file_read()	File reading operation. (read synchronous configuration from USB memory)
file_write()	File writing operation. (write evaluation result to USB memory)
file_stop()	File finalizing operation. (un-mount USB memory)
file_err()	File error operation. (un-mount USB memory)
get_sync_config()	Get synchronous configuration.
R_LED_Open ()	Initialize and open 7segment LED.
R_LED_UpdTime()	Update data to show 7segment LED.
rx64m_led_cmt_cyclic_isr()	7segment LED interrupt handler. (output data on the 7segment LED)

3.2 Environment and Execution

Execution of this example needs RX64M/71M RSK boards[1] more than three (Master node and more than two Slave nodes), a PC, an Ethernet Hub (hereafter HUB), an Ethernet cable and a USB memory. The outline of the execution sequence is following.

- Write the sample project execution code to all RX64M/71M RSK boards (hereafter RSK boards).
- Install an evaluate application to the PC.
- Connect all RSK boards to the PC via the Hub by the Ethernet cable[2].
- Insert the USB memory to the USB port in the every RSK board. The USB memory stores the synchronous configuration. It also stores the evaluation result data after evaluation.
- Power on the all RSK boards.
- Each clock (RSK board) gets the synchronous configuration from the USB memory and start the synchronization one another.
- Execute the evaluate application in the PC. The evaluate application in the PC issues trigger broadcast packets every 1 sec interval. As for the evaluate application, please refer to Sec 3.3.
- Contents of USB are composed of the parameter file (=PARAM.txt) and the Nth synchronous evaluation result data files. The parameter file stores synchronous configuration and you have to it root directory in advance. The sample files[3,4] are prepared in this project. The Nth synchronous evaluation result data files are composed of two kinds of files and created after evaluation. One kind files (= TEST_N.txt, N = 1, 2, , ,) save the local clock value. The other kind files (= EXT_N.txt, N = 1, 2, , ,) save the offsetFromMaster and meanPathDelay to synchronization clock source.
- Each clock gets from their local counter and saves the current local clock, offsetFromMaster and meanPathDelay to the USB memory every when trigger packet receiving. The fields of saved current local clock are lower 32 bits of second order field (=LCCVRM) and nanosecond order field (=LCCVRL).
- If the optional demo board is equipped [7], the 7segment LED also shows the current local clock. The field of showed current local clock is nanosecond order field (=LCCVRL).
- Repeat 50 times (get 50 samples)[5].
- User can check and compare the time differences, offsetFromMaster and meanPathDelay inter each clock devices from the data of USB memories.

[1] Product name is a Renesas Starter Kit+ for RX64M [8] or a Renesas Starter Kit+ for RX71M [9].

[2] Connection topology depends on the clock type (refer to Figure 3.12).

[3] demo_src\usb_memory_sample\param\ID=X,,,\PARAM.txt (refer to Sec 1.6).

[4] MAC addresses of these files set Renesas vendor ID (=74-90-50) and unique ID for this sample. Please change those values when users applied to this sample their own system.

[5] Selectable times in the “NUM_EVAL” and “NUM_UPDATE” option (refer to Sec 2.7).

Figure 3.1, Figure 3.2 and Figure 3.3 show the minimum configuration of the environment, the software flow overview and the contents example of USB memories respectively. As for the evaluate application, please refer to Sec 3.3. Figure 3.4, Figure 3.5 and Figure 3.6 also show the examples of the parameter file in the 3boards, OC and port1 configuration and the 5th evaluation result saved file in that configuration respectively.

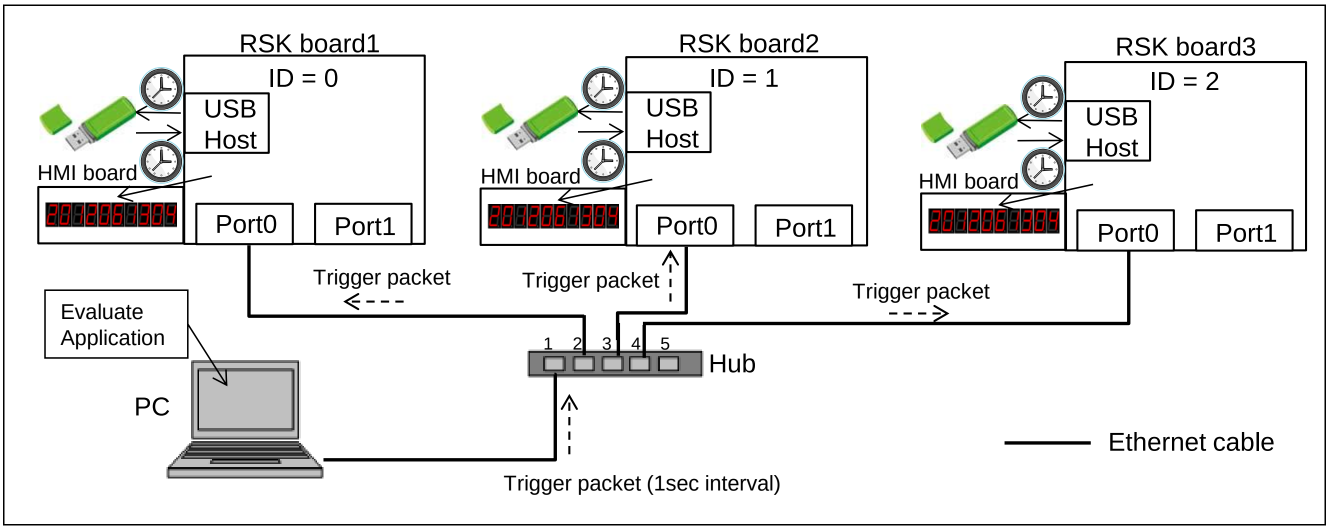


Figure 3.1 Environment (minimum configuration)

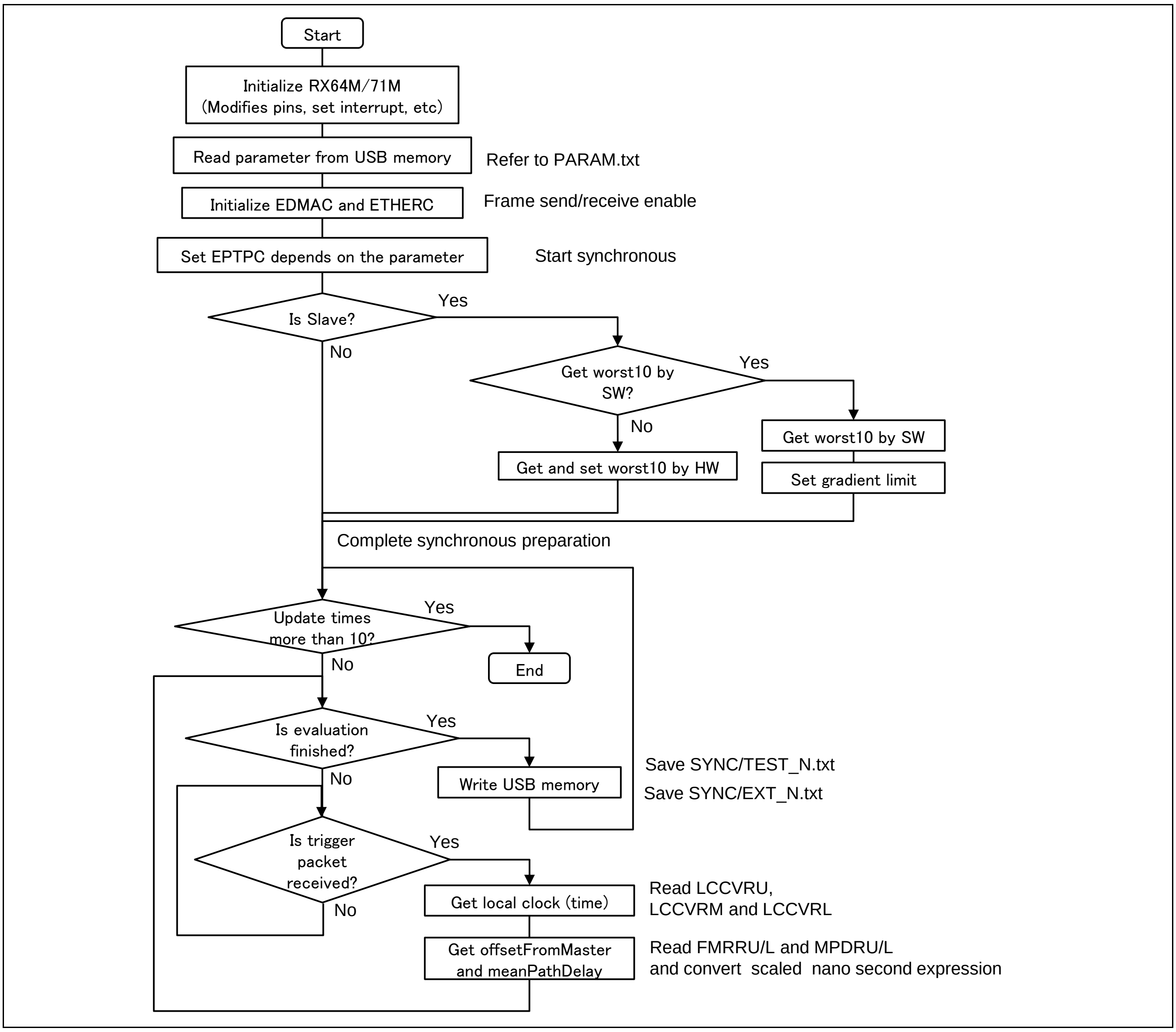
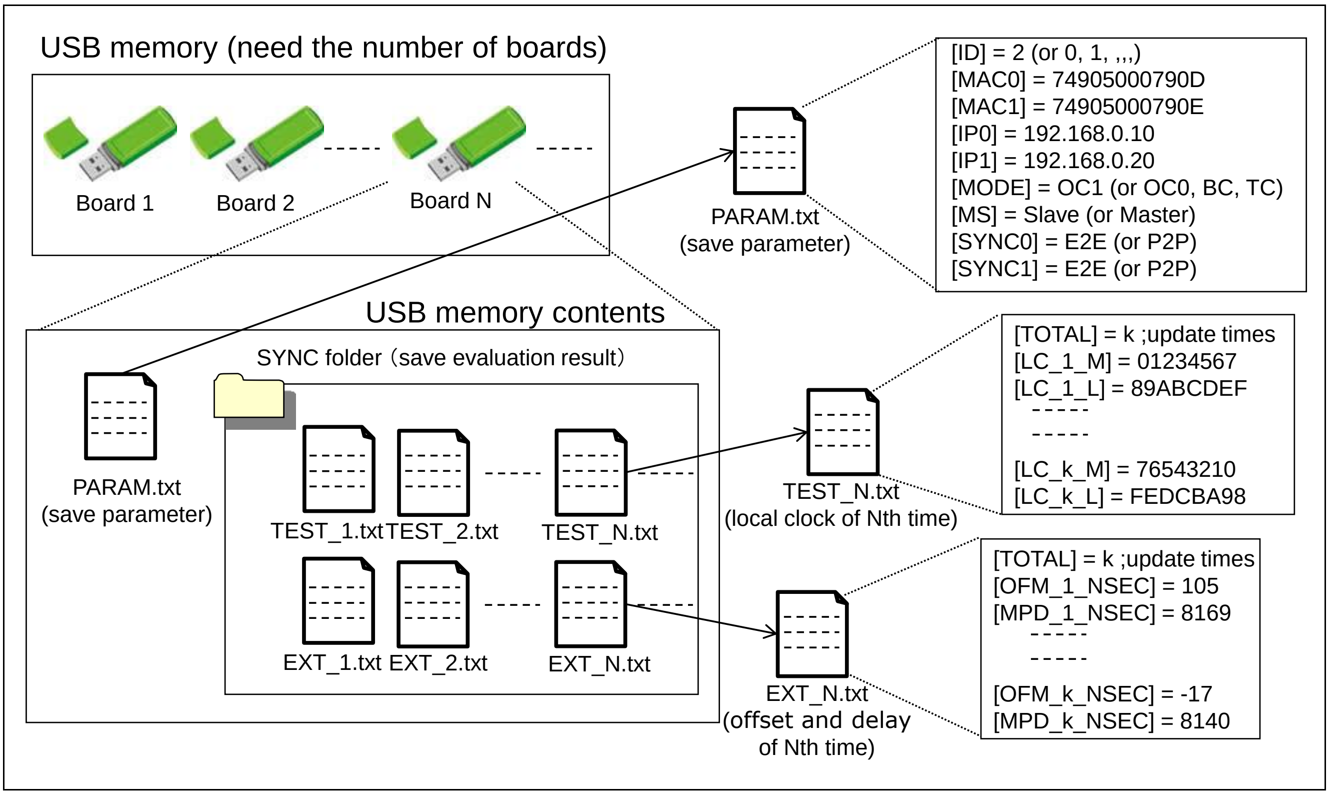

Start
Initialize RX64M/71M (Modifies pins, set interrupt, etc)
Read parameter from USB memory
Refer to PARAM.txt
Initialize EDMAC and ETHERC
Frame send/receive enable
Set EPTPC depends on the parameter
Start synchronous
Is Slave?
Yes
No
Get worst10 by SW?
Yes
No
Get worst10 by SW
Set gradient limit
Get and set worst10 by HW
Complete synchronous preparation
Update times more than 10?
Yes
No
End
Is evaluation finished?
Yes
No
Write USB memory
Save SYNC/TEST_N.txt
Save SYNC/EXT_N.txt
Is trigger packet received?
Yes
No
Get local clock (time)
Read LCCVRU, LCCVRM and LCCVRL
Get offsetFromMaster and meanPathDelay
Read FMRRU/L and MPDRU/L and convert scaled nano second expression

Figure 3.2 Software flow overview

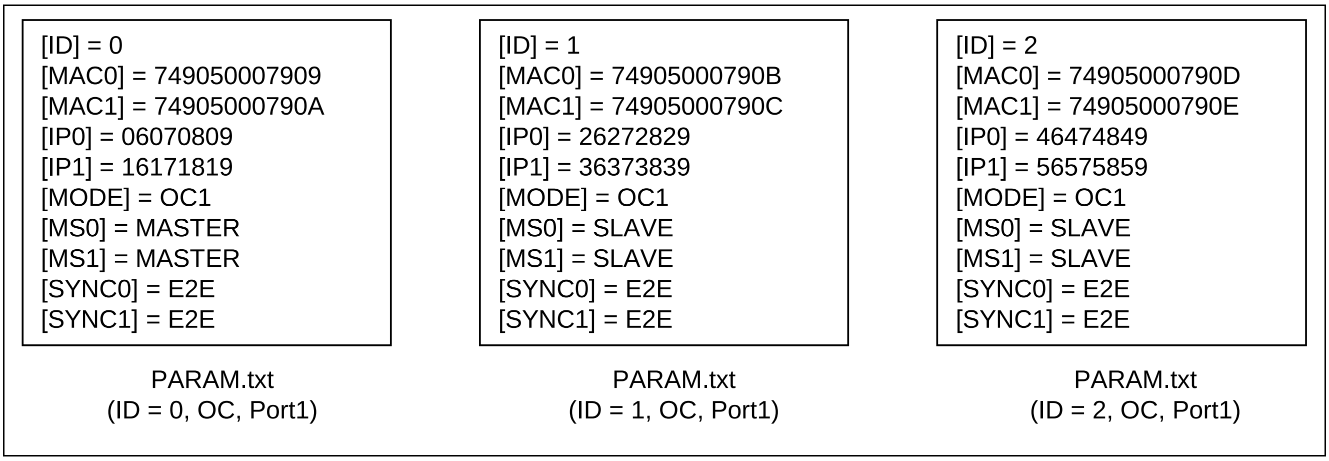


```
[ID] = 2 (or 0, 1, ,,,)
[MAC0] = 74905000790D
[MAC1] = 74905000790E
[IP0] = 192.168.0.10
[IP1] = 192.168.0.20
[MODE] = OC1 (or OC0, BC, TC)
[MS] = Slave (or Master)
[SYNC0] = E2E (or P2P)
[SYNC1] = E2E (or P2P)
```

```
[TOTAL] = k ;update times
[LC_1_M] = 01234567
[LC_1_L] = 89ABCDEF
  -----
  -----
[LC_k_M] = 76543210
[LC_k_L] = FEDCBA98
```

```
[TOTAL] = k ;update times
[OFM_1_NSEC] = 105
[MPD_1_NSEC] = 8169
  -----
  -----
[OFM_k_NSEC] = -17
[MPD_k_NSEC] = 8140
```

Figure 3.3 Contents example of USB memories



```
[ID] = 0
[MAC0] = 749050007909
[MAC1] = 74905000790A
[IP0] = 06070809
[IP1] = 16171819
[MODE] = OC1
[MS0] = MASTER
[MS1] = MASTER
[SYNC0] = E2E
[SYNC1] = E2E
```

PARAM.txt
(ID = 0, OC, Port1)

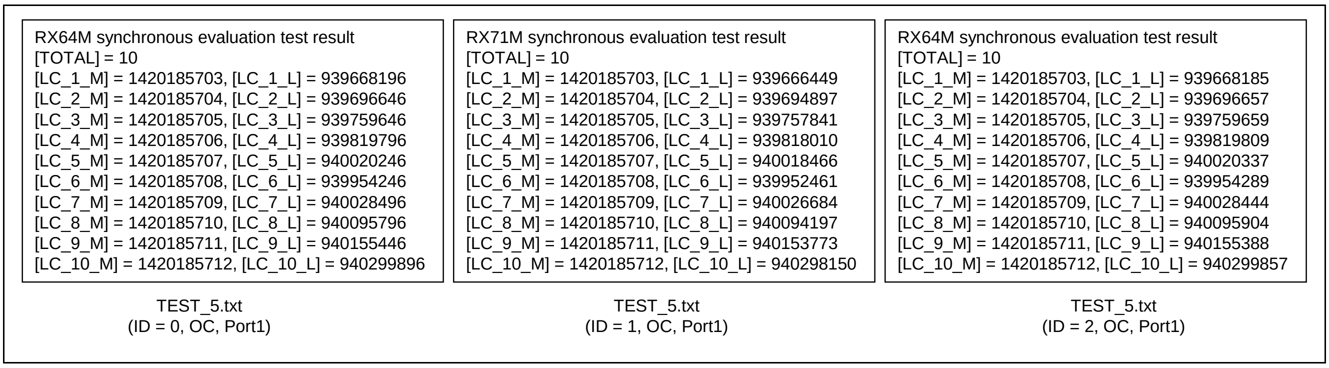
```
[ID] = 1
[MAC0] = 74905000790B
[MAC1] = 74905000790C
[IP0] = 26272829
[IP1] = 36373839
[MODE] = OC1
[MS0] = SLAVE
[MS1] = SLAVE
[SYNC0] = E2E
[SYNC1] = E2E
```

PARAM.txt
(ID = 1, OC, Port1)

```
[ID] = 2
[MAC0] = 74905000790D
[MAC1] = 74905000790E
[IP0] = 46474849
[IP1] = 56575859
[MODE] = OC1
[MS0] = SLAVE
[MS1] = SLAVE
[SYNC0] = E2E
[SYNC1] = E2E
```

PARAM.txt
(ID = 2, OC, Port1)

Figure 3.4 Example of the parameter files



```
RX64M synchronous evaluation test result
[TOTAL] = 10
[LC_1_M] = 1420185703, [LC_1_L] = 939668196
[LC_2_M] = 1420185704, [LC_2_L] = 939696646
[LC_3_M] = 1420185705, [LC_3_L] = 939759646
[LC_4_M] = 1420185706, [LC_4_L] = 939819796
[LC_5_M] = 1420185707, [LC_5_L] = 940020246
[LC_6_M] = 1420185708, [LC_6_L] = 939954246
[LC_7_M] = 1420185709, [LC_7_L] = 940028496
[LC_8_M] = 1420185710, [LC_8_L] = 940095796
[LC_9_M] = 1420185711, [LC_9_L] = 940155446
[LC_10_M] = 1420185712, [LC_10_L] = 940299896
```

TEST_5.txt
(ID = 0, OC, Port1)

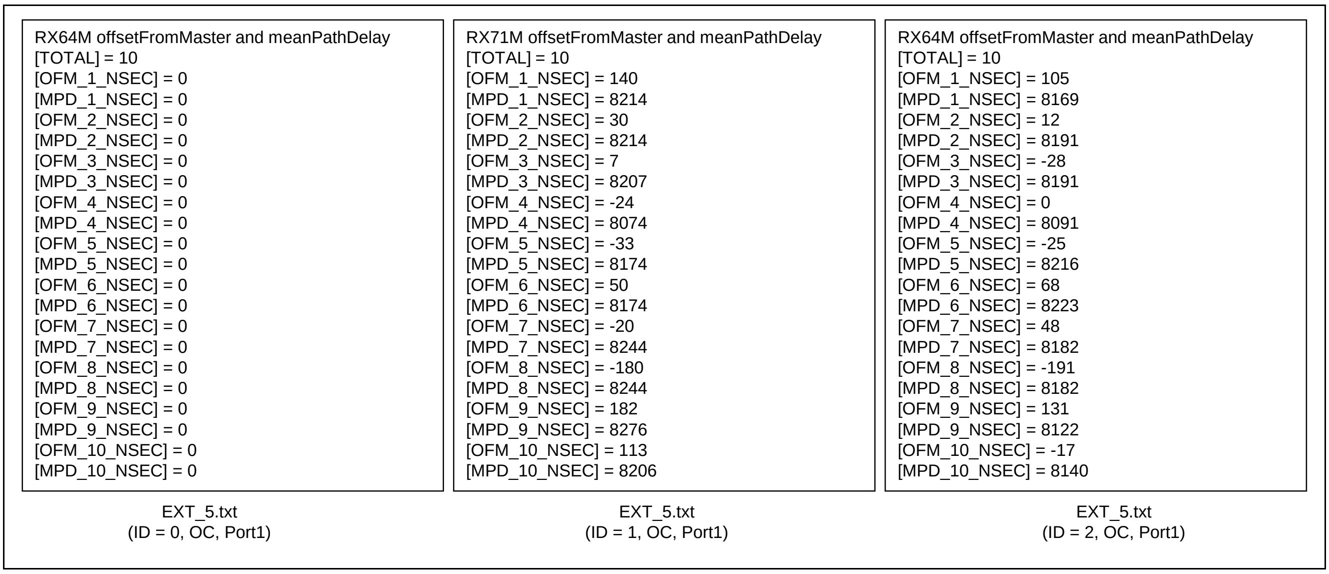
```
RX71M synchronous evaluation test result
[TOTAL] = 10
[LC_1_M] = 1420185703, [LC_1_L] = 939666449
[LC_2_M] = 1420185704, [LC_2_L] = 939694897
[LC_3_M] = 1420185705, [LC_3_L] = 939757841
[LC_4_M] = 1420185706, [LC_4_L] = 939818010
[LC_5_M] = 1420185707, [LC_5_L] = 940018466
[LC_6_M] = 1420185708, [LC_6_L] = 939952461
[LC_7_M] = 1420185709, [LC_7_L] = 940026684
[LC_8_M] = 1420185710, [LC_8_L] = 940094197
[LC_9_M] = 1420185711, [LC_9_L] = 940153773
[LC_10_M] = 1420185712, [LC_10_L] = 940298150
```

TEST_5.txt
(ID = 1, OC, Port1)

```
RX64M synchronous evaluation test result
[TOTAL] = 10
[LC_1_M] = 1420185703, [LC_1_L] = 939668185
[LC_2_M] = 1420185704, [LC_2_L] = 939696657
[LC_3_M] = 1420185705, [LC_3_L] = 939759659
[LC_4_M] = 1420185706, [LC_4_L] = 939819809
[LC_5_M] = 1420185707, [LC_5_L] = 940020337
[LC_6_M] = 1420185708, [LC_6_L] = 939954289
[LC_7_M] = 1420185709, [LC_7_L] = 940028444
[LC_8_M] = 1420185710, [LC_8_L] = 940095904
[LC_9_M] = 1420185711, [LC_9_L] = 940155388
[LC_10_M] = 1420185712, [LC_10_L] = 940299857
```

TEST_5.txt
(ID = 2, OC, Port1)

Figure 3.5 Example of the evaluation result files (current local clock)



```
RX64M offsetFromMaster and meanPathDelay
[TOTAL] = 10
[OFM_1_NSEC] = 0
[MPD_1_NSEC] = 0
[OFM_2_NSEC] = 0
[MPD_2_NSEC] = 0
[OFM_3_NSEC] = 0
[MPD_3_NSEC] = 0
[OFM_4_NSEC] = 0
[MPD_4_NSEC] = 0
[OFM_5_NSEC] = 0
[MPD_5_NSEC] = 0
[OFM_6_NSEC] = 0
[MPD_6_NSEC] = 0
[OFM_7_NSEC] = 0
[MPD_7_NSEC] = 0
[OFM_8_NSEC] = 0
[MPD_8_NSEC] = 0
[OFM_9_NSEC] = 0
[MPD_9_NSEC] = 0
[OFM_10_NSEC] = 0
[MPD_10_NSEC] = 0
```

EXT_5.txt
(ID = 0, OC, Port1)

```
RX71M offsetFromMaster and meanPathDelay
[TOTAL] = 10
[OFM_1_NSEC] = 140
[MPD_1_NSEC] = 8214
[OFM_2_NSEC] = 30
[MPD_2_NSEC] = 8214
[OFM_3_NSEC] = 7
[MPD_3_NSEC] = 8207
[OFM_4_NSEC] = -24
[MPD_4_NSEC] = 8074
[OFM_5_NSEC] = -33
[MPD_5_NSEC] = 8174
[OFM_6_NSEC] = 50
[MPD_6_NSEC] = 8174
[OFM_7_NSEC] = -20
[MPD_7_NSEC] = 8244
[OFM_8_NSEC] = -180
[MPD_8_NSEC] = 8244
[OFM_9_NSEC] = 182
[MPD_9_NSEC] = 8276
[OFM_10_NSEC] = 113
[MPD_10_NSEC] = 8206
```

EXT_5.txt
(ID = 1, OC, Port1)

```
RX64M offsetFromMaster and meanPathDelay
[TOTAL] = 10
[OFM_1_NSEC] = 105
[MPD_1_NSEC] = 8169
[OFM_2_NSEC] = 12
[MPD_2_NSEC] = 8191
[OFM_3_NSEC] = -28
[MPD_3_NSEC] = 8191
[OFM_4_NSEC] = 0
[MPD_4_NSEC] = 8091
[OFM_5_NSEC] = -25
[MPD_5_NSEC] = 8216
[OFM_6_NSEC] = 68
[MPD_6_NSEC] = 8223
[OFM_7_NSEC] = 48
[MPD_7_NSEC] = 8182
[OFM_8_NSEC] = -191
[MPD_8_NSEC] = 8182
[OFM_9_NSEC] = 131
[MPD_9_NSEC] = 8122
[OFM_10_NSEC] = -17
[MPD_10_NSEC] = 8140
```

EXT_5.txt
(ID = 2, OC, Port1)

Figure 3.6 Example of the evaluation result files (offsetFromMaster and meanPathDelay)

3.3 Evaluate Application

The evaluate application in the PC is console based application and can be operate on the .NET Framework 2.0 on the Windows ^{TM}. This application issues trigger packets with 1 sec interval to RSK boards connected to Ethernet cables such as the network of the Figure 3.1.

The trigger packet is IPv4 UDP broadcast packet and 60 byte length. The contents of data part are begun with “0x50, 0x54, 0x50, 0x54, 0x45, 0x53, 0x54” whose converted ASCII code are “PTPTEST”. The trigger packet format shows Figure 3.7.

Destination MAC address	Source MAC address	Type			
FF-FF-FF-FF-FF-FF	XX-XX-XX-XX-XX-XX	08-00	IPv4 header	UDP header	50-54-50-54-45-53-54-00-00-00-00-00-00-00-00-00-00
Header part			Data part		

Figure 3.7 Trigger packet format

When user input the command to start of issuing packets after moved to the execute file (=SendTestPacket.exe) installed folder and the start of command format is “SendTestPacket [IP address of the PC]”. Figure 3.8 shows the start command format when IP address of the PC is “192.168.0.10”.

```
SendTestPacket 192.168.0.10
```

Figure 3.8 Start of command format

When user input any keys after input the start of command, the command issue operation is stopped.

Before issuing the trigger packets, user need to wait all Slave nodes complete getting the worst 10 and setting the gradient limit values to the EPTPC.

The user LED composed of LED0, LED1, LED2 and LED3 on the RSK board indicates whether the operation is completed or not following patterns, when the RSK board is Slave node. Interval of that operation depends on the interval times of getting worst10 values set to “TIMES_W10_INTERVAL” macro. If default setting, the interval is approximately 30sec.

- Odd pattern (LED0: OFF, LED1: ON, LED2: OFF, LED3: ON)

Not completed the operation yet (During operation).

- Even pattern (LED0: ON, LED1: OFF, LED2: ON, LED3: OFF)

Complete the operation (Already finished operation).

User can start to issue the trigger packets only when the user LED signal is even pattern.

3.4 Board Setting

There are four jumpers changing from the default setting of the RX64M/71M RSK board to execute this example. The Ether PHY access channel is set consistent with the software configuration. The USB access setting has to be changed from the board default setting. When the product name of the RX64M/71M RSK board is R0K50564MC001BR or R0K5RX71MC010BR, Figure 3.9 indicates their changing. And when the product name of the RX71M RSK board is R0K50571MC000BR, Figure 3.10 indicates their changing.

- Ether PHY access setting

Jumper	LINK_CH = 1 (Default setting)	LINK_CH = 0	Functional use
J3	**2-3**	**1-2**	**ETHERC ET0MDIO or ET1MDIO**
J4	**2-3**	**1-2**	**ETHERC ET0MDC or ET1MDC**

- USB access setting

Jumper	Board default setting	This example	Functional use
J2	2-3	**1-2**	**USB Enables Host Mode**
J6	1-2	**2-3**	**USB USB0VBUSEN**

Figure 3.9 Jumper setting

- Ether PHY access setting

Jumper	LINK_CH = 1 (Default setting)	LINK_CH = 0	Functional use
J13	**2-3**	**1-2**	**ETHERC ET0MDIO or ET1MDIO**
J9	**2-3**	**1-2**	**ETHERC ET0MDC or ET1MDC**

- USB access setting

Jumper	Board default setting	This example	Functional use
J1	2-3	**1-2**	**USB Enables Host Mode**
J3	1-2	**2-3**	**USB USB0VBUSEN**

Figure 3.10 Jumper setting

3.5 HMI Expansion Board Setting

There are several resistances changing from the default setting of the HMI expansion board to use the 7segment LED. Figure 3.11 indicates their changing from default setting.

Resistance	Default setting	Demo setting	Functional use
R387	Mount	**Remove**	**7segment LED**
R433	Mount	**Remove**	**7segment LED**
R464	Mount	**Remove**	**7segment LED**
R348	Remove	**Mount**	**7segment LED**
R404	Remove	**Mount**	**7segment LED**
R429	Remove	**Mount**	**7segment LED**
R445	Remove	**Mount**	**7segment LED**

Figure 3.11 Resistance setting (7segment LED used case)

3.6 Topology

This example supports several kinds of topologies combination with the clock type and the delay mechanism even if the minimum configuration (using 3 boards). Table 3.2 lists the topology dependence with configuration and Figure 3.12 shows the each topology figures.

Table 3.2 Topology list

No.	Configuration	Node Image	Board (ID = 0)		Board (ID = 1)		Board (ID = 2)		Delay mechanism
			Port0	Port1	Port0	Port1	Port0	Port1	
1	OC	Common	Master OC	-	Slave1 OC	-	Slave2 OC	-	E2E
2	BC Slave & Master	Hub	Master1 OC	-	Slave1 BC	Master2 BC	Slave2 OC	-	E2E & P2P
3	BC Master & Master	Hub	Slave1 OC	-	Master1 BC	Master2 BC	Slave2 OC	-	E2E & P2P
4	TC	Repeater	Master OC	-	Slave1 TC	TC	Slave2 OC	-	E2E & P2P

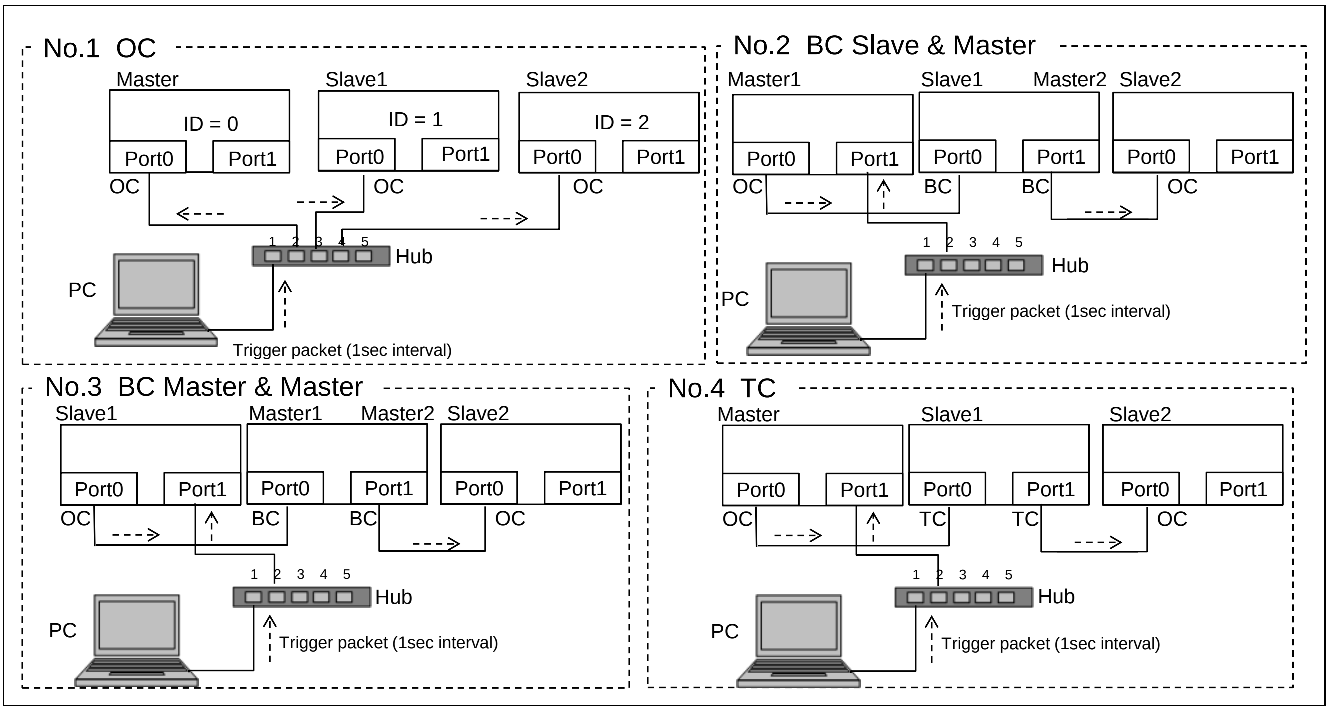


Figure 3.12 Topology figure

4. Reference Documents

User’s Manual: Hardware

RX64M Group User’s Manual: Hardware Rev.1.10 (R01UH0377EJ)
RX71M Group User’s Manual: Hardware Rev.1.00 (R01UH0493EJ)
The latest version can be downloaded from the Renesas Electronics website.

User’s Manual: Software

RX Family RXv2 Instruction Set Architecture User’s Manual: Hardware Rev.1.00 (R01US0071EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul 10, 2014	—	First edition issued.
1.01	Aug 20, 2014	—	Worst10 setting and optional 7segment LED function added.
1.02	Dec 31, 2014	—	Applied PTP driver Rev.1.02 and changed file structure.
1.10	Mar 31, 2016	—	Applied PTP driver Rev.1.10 and changed file access interface.
1.11	Nov 11, 2016	—	Applied PTP driver Rev.1.12 and Ethernet driver Rev.1.12.
1.12	Mar 31, 2017	—	Applied PTP driver Rev.1.13.
		16	Changed topology list (Table 3.2) and topology figure (Fig 3.12) for TC operation.
		—	Modified equipped USB memory sample contents.

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

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

1. 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 Microprocessing unit or Microcontroller unit products 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 disputes 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, drawing, chart, program, algorithm, 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 otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
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; and 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.

   Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate 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. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
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 majority-owned subsidiaries.

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

(Rev.3.0-1 November 2016)

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
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 II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141