RX Family

R01AN1984EJ0112
Rev.1.12
Mar 31, 2017

PTP Timer Synchronous Start Using Firmware Integration Technology Modules

Introduction

This document explains one of the EPTPC FIT (Firmware Integration Technology) module [1] usage examples. This example is starting the motor control timer at a synchronized time without CPU operation. The time synchronization is 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 starts the Multi-Function Timer Pulse Unit (MTU3) or the General PWM Timer (GPT) at a specific synchronous time via Event Link Controller (ELC) without CPU operation. The time synchronization protocol is applied to the PTP. Users can observe the phase differences of the output PWM waves and can use the software sample to implement their own system such as the motor control one.

1.1 PTP Timer Synchronous Start 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

The Ethernet peripheral modules of the RX64M/71M group 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)). The EPTPC is divided to PTP Frame Operation (CH0) part, PTP Frame Operation (CH1) part, Packet Relation Control part and Statistical Time Correction Algorithm part from their functionality. The EPTPC is also connected to the motor control timers (MTU3 and GPT peripheral modules) and the general ports (I/O ports) via ELC peripheral module to synchronous activation of multiple motors.

Followings are the summary of the Ethernet peripheral modules and Figure 1.1 shows the related hardware’s block diagram.

1. Synchronous function (EPTPC and PTPEDMAC)

- Based on the IEEE1588-2008 Version2
- Time synchronous function issuing PTP messages (Ethernet frame[1] and UDP IPv4 format[2])
- Master and Slave, OC, BC, TC functionality
- Time deviation is corrected by the statistical correction method (Gradient prediction time correction algorithm)
- Timer event output (6CH, rise/fall edges, event flag auto clear)

- Motor control timer (MTU3, GPT) is started synchronously with the timer event via ELC

- Synchronous pulses are outputted connecting EPTPC to I/O ports via ELC

- Selectable PTP message operation (PTP module internal operation, CPU via PTPEDMAC, to other port)

2. Enhanced standard Ethernet function

- Possible to use the independent dual channels Ethernet

- HW switch (selectable Cut Through or Store & Forward internal frame propagation)

- HW multicast frame filter (all receive, all cancel, receive specific two frames)

[1] In case of RX64M Group, supports only Ethernet II frame format (not support IEEE802.3 frame format)

[2] Not support UDP IPv6

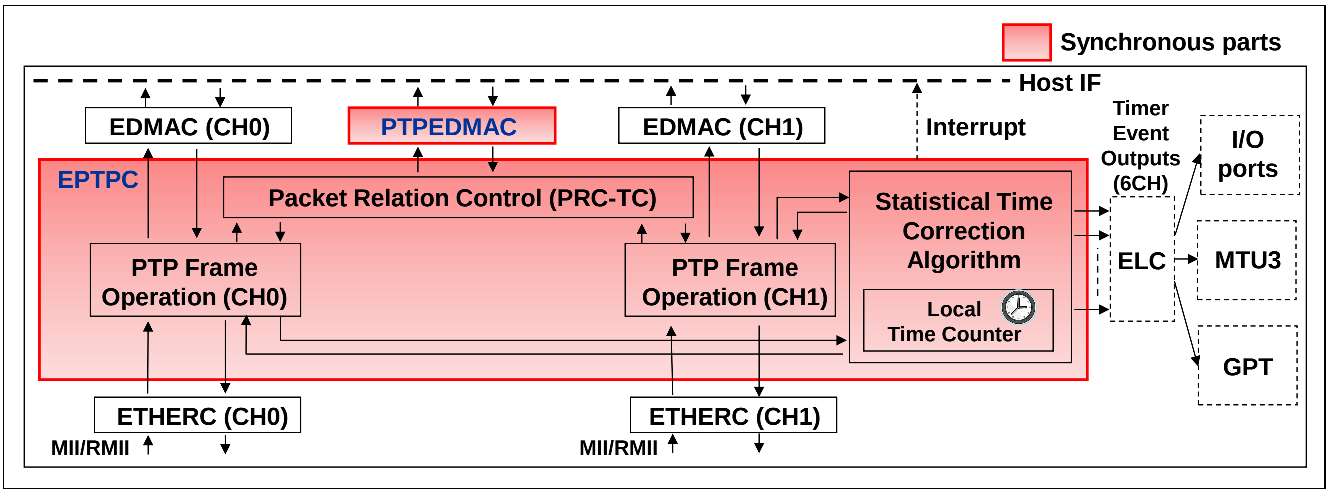

Figure 1.1 Hardware block diagram

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 a specific timer start time from the USB memory, set the specific timer start time to the PTP driver, set an event link connection between EPTPC and MTU3/GPT, set PMM output setting to MTU3/GPT driver 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 1.2 shows the software structure of this sample.

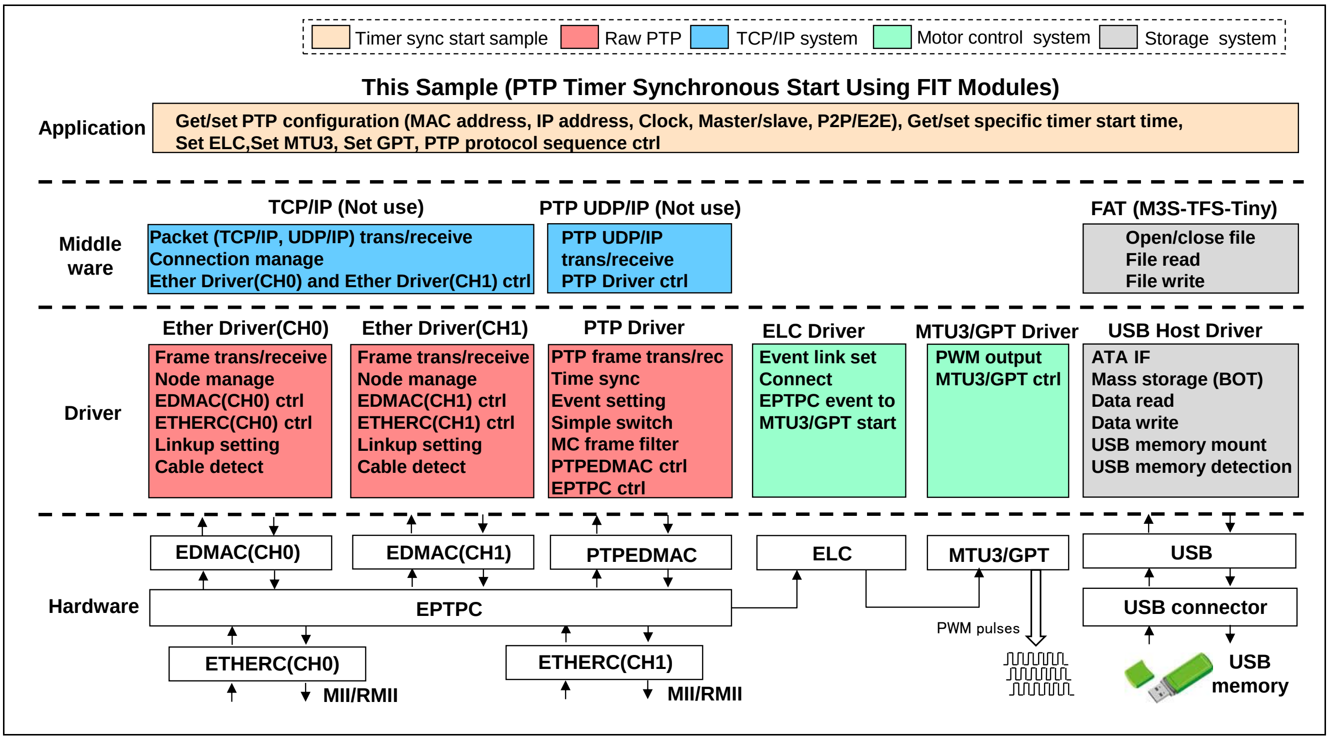

Figure 1.2 Software structure of this sample

1.6 File Structure

This sample codes are stored the "demo_src" and low hierarchical folders. ELC, MTU3 and GPT drivers are stored each driver folders respectively. Figure 1.3 shows the file structure of this sample. 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: Synchronous configuration sample files
|      + --- PULSE:  Output timer parameter sample file
|            + --- MTU3:  Case of MTU3
|            + --- GPT:    Case of GPT
|
+ --- 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_elc_rx: ELC driver folder
|  r_elc_rx_if.h ;ELC driver header file
|
+ --- src:
|      r_elc.c ; ELC driver source file

r_ether_rx: Ethernet Driver FIT module

r_gpt_rx: GPT driver folder
|  r_gpt_rx_if.h ;GPT driver header file
|
+ --- src:
|      r_gpt.c ; GPT driver source file

r_mtu3_rx: MTU3 driver folder
|  r_mtu3_rx_if.h ;MTU3 driver header file
|
+ --- src:
|      r_mtu3.c ; MTU3 driver source file

r_ptp_rx: PTP Driver FIT module

r_tfat_rx: FAT file system (M3S-TFS-Tiny) FIT module

r_usb_basic: USB driver (USB basic operation) FIT module

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

Figure 1.3 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
- ELC
- MTU3
- GPT
- 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 and start MTU3/GPT timers via ELC.

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 ELC

This example uses the ELC to connect events between EPTPC and MTU3/GPT. Those resources need to start the MTU3/GPT synchronously.

2.2.5 MTU3 Channel

This example uses an MTU3 channel for the synchronous PWM output. Please do not modify the settings or try to use the peripheral during this module's operations.

2.2.6 GPT Channel

This example uses a GPT channel for the synchronous PWM output. Please do not modify the settings or try to use the peripheral during this module's operations.

2.2.7 CMT Channel

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

2.2.8 USB Channel

This example uses the USB 2.0 FS Host/Function Module to read synchronous configuration parameter and pulse output time from 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, r_elc_rx_if.h, r_mtu3_rx_if.h, r_gpt_rx_if.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 = 1**	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 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 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]).

2.8 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_READ = 0,   /* USB memory read state */
    APL_COM,        /* Ether communication 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;
```

```
/* Timer configuration structure */
typedef struct {
    uint8_t  tmr;
    uint8_t  ch;
    uint8_t  edge;
    uint32_t time_u;
    uint32_t time_l;
} PulseConfig;
```

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, r_elc_rx_if.h, r_mtu3_rx_if.h, r_gpt_rx_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;
```

```
/* ELC driver return value */
ELC_OK (0) /* No error */
ELC_ERROR (-1) /* General error */
```

```
/* MTU3 driver return value */
MTU3_OK (0) /* No error */
MTU3_ERROR (-1) /* General error */
```

```
/* GPT driver return value */
GPT_OK (0) /* No error */
GPT_ERROR (-1) /* General error */
```

```
/* 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.
SyncErr ()	PTP or Ether error operation.
ReadPTPMsg()	Read PTP messages. If announce message is received, update Master port identity.
led_callback()	Compare match timer callback function for 7seg led.
file_read()	File reading operation. (read synchronous configuration and output pulse parameter from 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.
get_pulse_time()	Get output pulse 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 two (Master node and more than one Slave node), an Ethernet Hub (hereafter HUB), an Ethernet cable, a USB memoryand an Oscilloscope. The PWM output pin of the each RX64M/71M RSK boards connects the input of the oscilloscope. The outline of the execution sequence is following.

- Write the sample project execution code to all RX64M/71M RSK boards (hereafter RSK boards).
- Connect all RSK boards one another via the Hub by the Ethernet cable.
- Connect the PWM output pins of each RSK board to oscilloscope.
- Insert the USB memory to the USB port in the every RSK board. The USB memory stores the synchronous configuration and output pulse parameter.
- Power on the all RSK boards.
- Each clock (RSK board) gets the synchronous configuration and the output pulse parameter include a specific timer start time from the USB memory and start the synchronization one another.
- Contents of USB memories are composed of the synchronous configuration file (=PARAM.txt) and the output pulse parameter file (= TEST_1.txt). The sample files[2,3] are prepared in this project. The fields of saved timer start time are higher and lower 32 bits of nanosecond order field (=TMSTTRUm and TMSTTRLm)[4].
- If the optional demo board is equipped [7], the 7segment LED also shows the timer start time. The field of showed the timer start time is lower 32 bits of nanosecond order field (=TMSTTRLm).
- Each clock starts the MTU3/GPT timer via ELC when their local clock counter synchronized of the PTP compares match the specific time.
- User can observe the phase differences of the output PWM waves.

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

[2] demo_src\usb_memory_sample\PARAM\ID=X,,,\PARAM.txt, demo_src\usb_memory_sample\PULSE\GPT or MTU3\PULSE\TEST_1.txt, (refer to Sec 1.6).

[3] 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.

[4] The index “m” indicates timer channel from 0 to 5.

Figure 3.1 shows the environment using three board’s configuration.

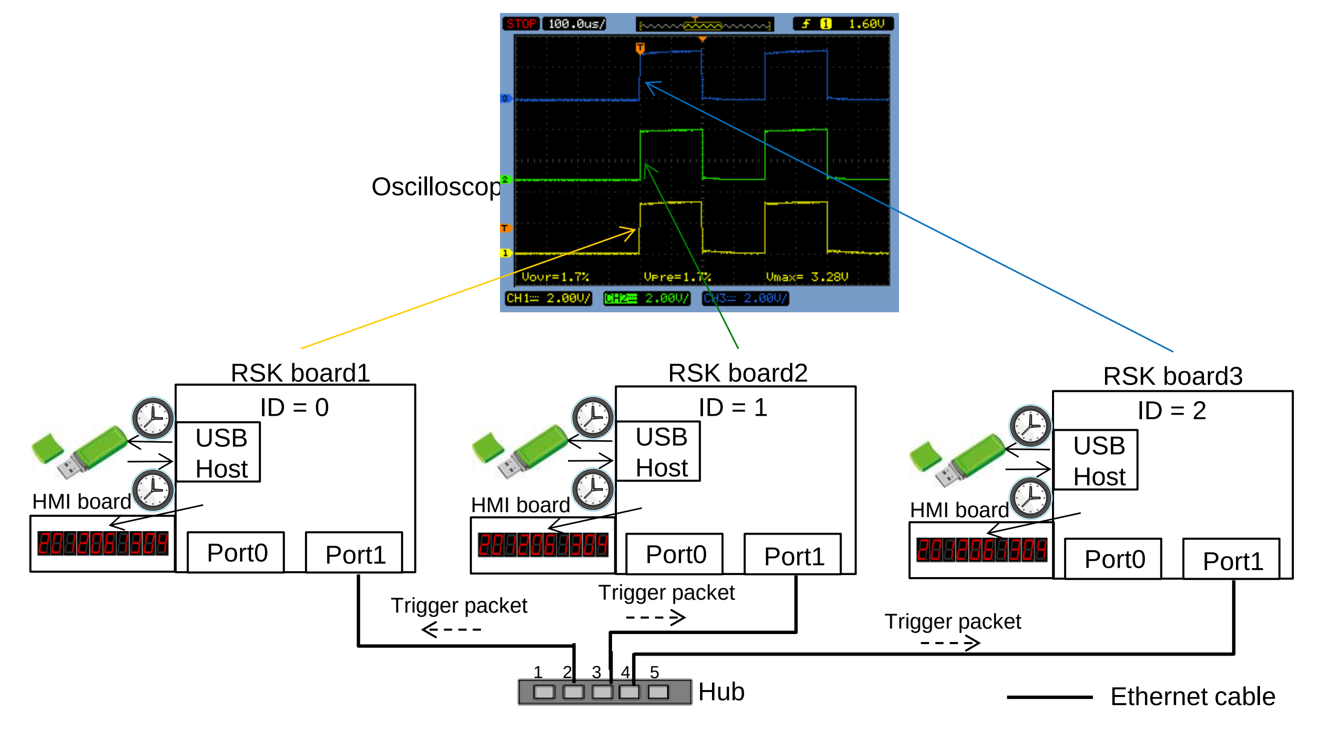

Figure 3.1 Environment (three board configuration)

Figure 3.2 and Figure 3.3 show the software flow overview and the contents example of USB memories respectively. Figure 3.4 and Figure 3.5 also show the examples of the synchronous configuration file in the 3boards, OC and port1 configuration and the output pulse parameter file respectively.

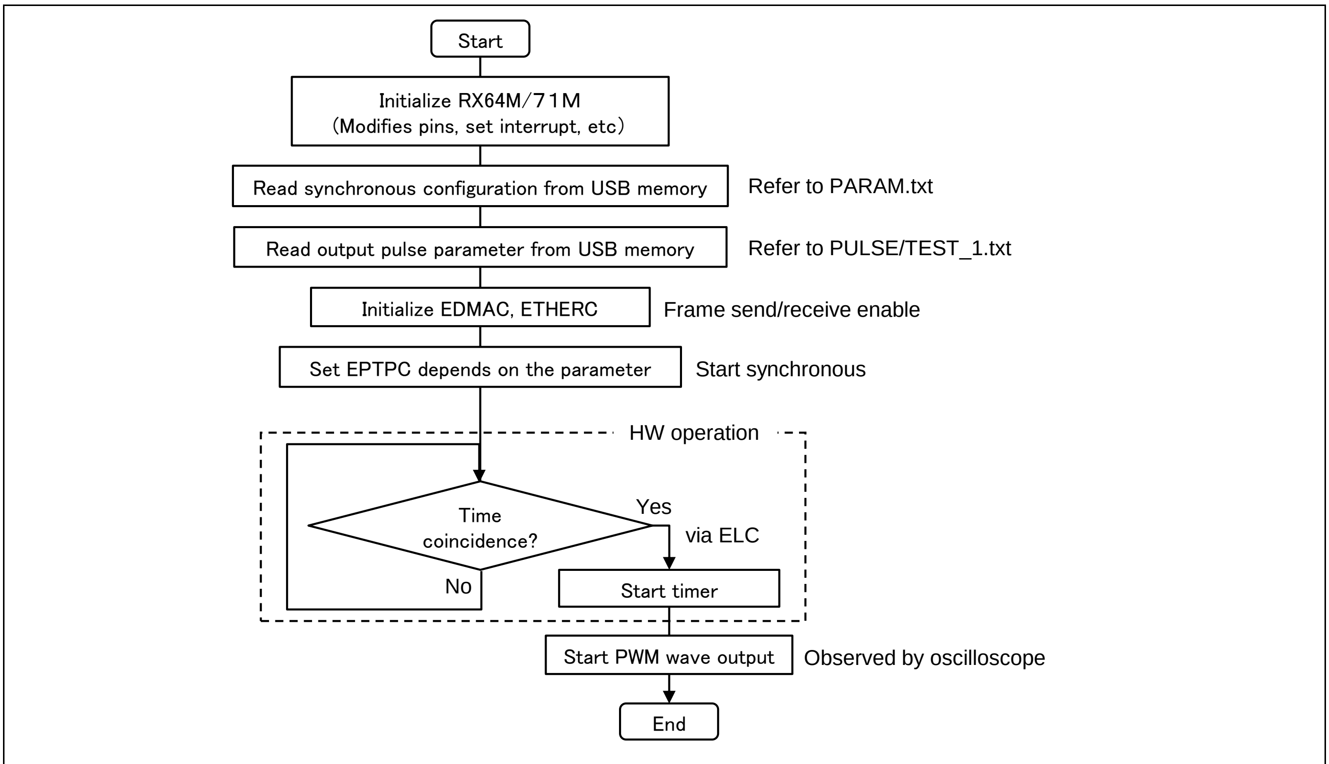

Figure 3.2 Software flow overview

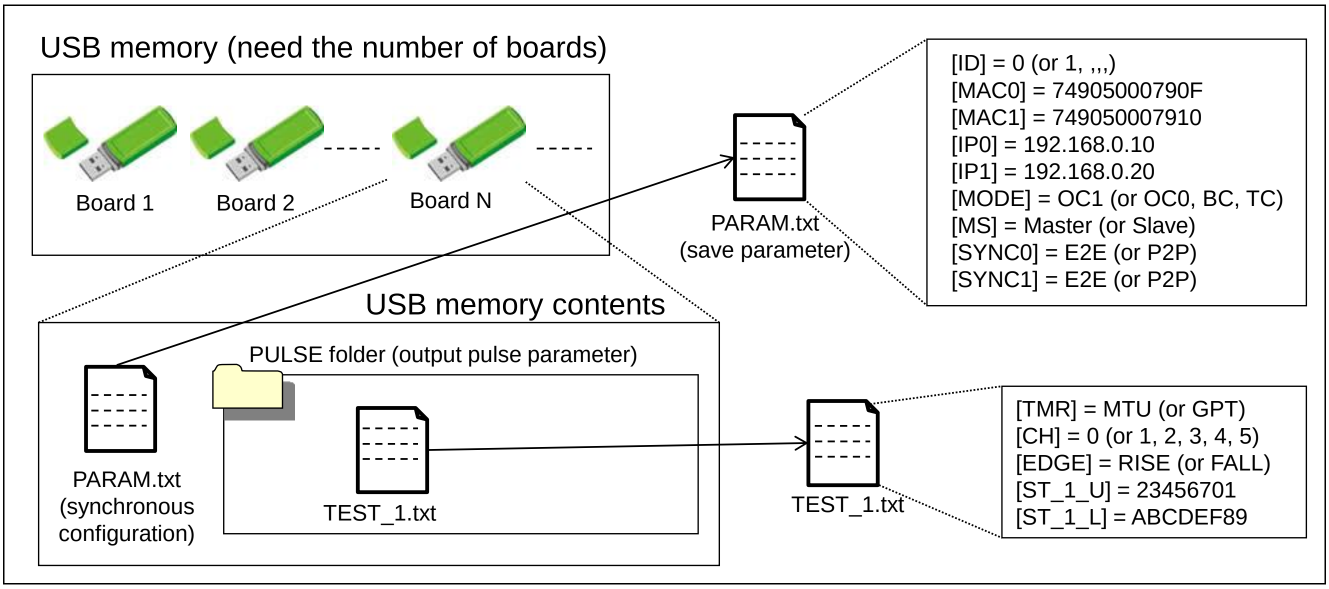

Figure 3.3 Contents example of USB memories

PARAM.txt (ID = 0, OC, Port1)	PARAM.txt (ID = 1, OC, Port1)	PARAM.txt (ID = 2, OC, Port1)
[ID] = 0	[ID] = 1	[ID] = 2
[MAC0] = 74905000790F	[MAC0] = 749050007911	[MAC0] = 749050007913
[MAC1] = 749050007910	[MAC1] = 749050007912	[MAC1] = 749050007914
[IP0] = 06070809	[IP0] = 26272829	[IP0] = 46474849
[IP1] = 16171819	[IP1] = 36373839	[IP1] = 56575859
[MODE] = OC1	[MODE] = OC1	[MODE] = OC1
[MS0] = MASTER	[MS0] = SLAVE	[MS0] = SLAVE
[MS1] = MASTER	[MS1] = SLAVE	[MS1] = SLAVE
[SYNC0] = E2E	[SYNC0] = E2E	[SYNC0] = E2E
[SYNC1] = E2E	[SYNC1] = E2E	[SYNC1] = E2E

Figure 3.4 Example of the synchronous configuration files

TEST_1.txt (MTU3, CH = 3, RISE Edge)	TEST_1.txt (GPT, CH = 4, FALL Edge)
[TMR] = MTU	[TMR] = GPT
[CH] = 3	[CH] = 4
[EDGE] = RISE	[EDGE] = FALL
[ST_1_U] = 14417BF1	[ST_1_U] = 14417BF3
[ST_1_L] = EEFBC200	[ST_1_L] = 90374400

Figure 3.5 Example of the output pulse parameter files

3.3 Board Setting

There are five 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 and the PWM output settings have to be changed from the board default setting. When the product name of the RX64M/71M RSK board is R0K50564MC001BR or R0K5RX71MC010BR, Figure 3.6 indicates their changing. And when the product name of the RX71M RSK board is R0K50571MC000BR, Figure 3.7 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

- PWM output setting

Jumper	Board default setting	This example	Functional use
J15	1-2	2-3	MTU3 MTIOC0C (CH0)

Figure 3.6 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

- PWM output setting

Jumper	Board default setting	This example	Functional use
J15	1-2	2-3	MTU3 MTIOC0C (CH0)

Figure 3.7 Jumper setting

User need to connect the PWM output pin of the RX64M/71M RSK board to the oscilloscope pin. Figure 3.8 indicates the board pins outputted PWM wave.

Application header	Pin	Header name	MCU pin	PWM output timer
JA2	23	IRQ2/M1_EncZ/M1_HSIN2	29	MTU3 MTIOC0C (CH0)
JA3	18	D1	156	GPT GTIOC1A (CH1)

Figure 3.8 PWM output pins

3.4 HMI Expansion Board Setting

There are several resistances changing from the default setting of the HMI expansion board to use the 7segment LED. Figure 3.9 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 3.9 Resistance setting (7segment LED used case)

3.5 Measurement Examples

The measurement results of the three board configuration show followings as the reference data.

Please keep your mind those result are depend on the measurement condition and environment.

3.5.1 Condition

- Topology

Using three RSK boards, synchronization mode is OC E2E.

- Synchronous mode

The gradient correction, which is the functionality of STCA unit, was applied (=mode2).

- PTP commands interval

PTP commands interval was 1sec[1].

[1] The intervals of Sync and Delay_Req message were 1sec.

3.5.2 Method

We measured the PWM output waves using oscilloscope (Identical configuration of the Figure 3.1) when MTU3 and GPT applied to PWM output timer.

3.5.3 Result

Figure 3.10 and Figure 3.11 show the PWM output waves using MTU3 channel3 and GPT channel4 and those time scale are 100µsec and 100nsec per unit coordinate respectively. The yellow line, green line and blue line indicate the Master, Slave1 and Slave2 respectively. The phase differences of the output PWM waves were approximately 100nsec.

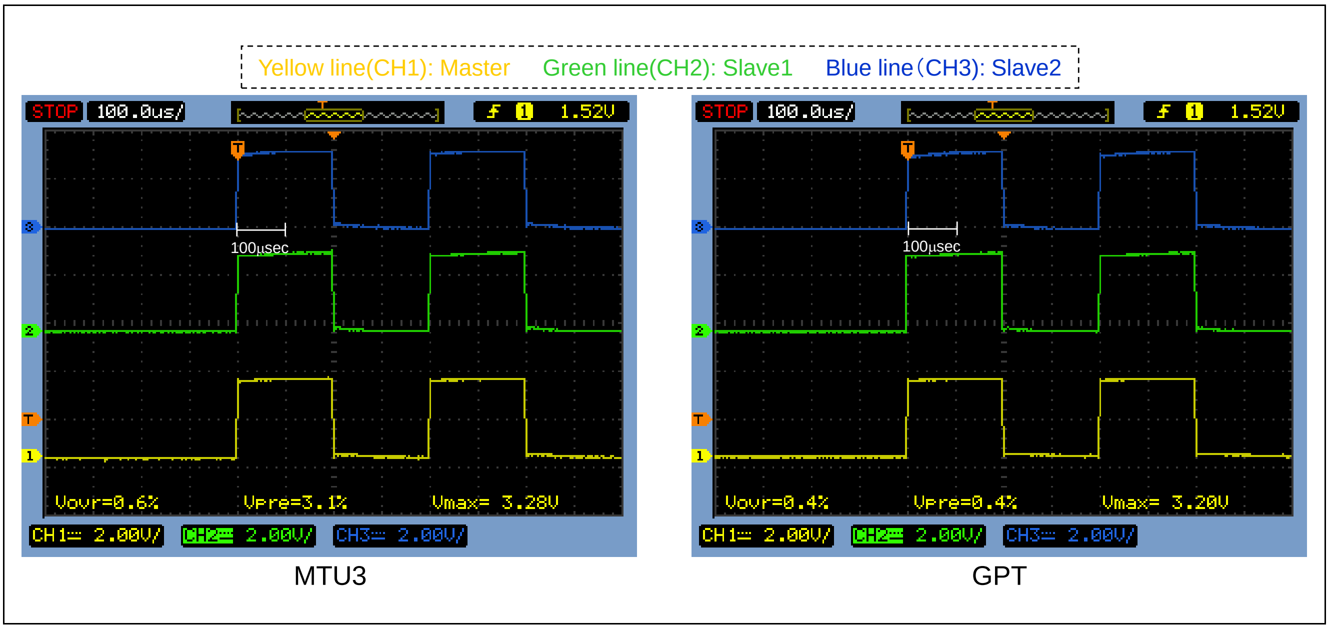

Figure 3.10 PWM wave example (100µsec scale unit)

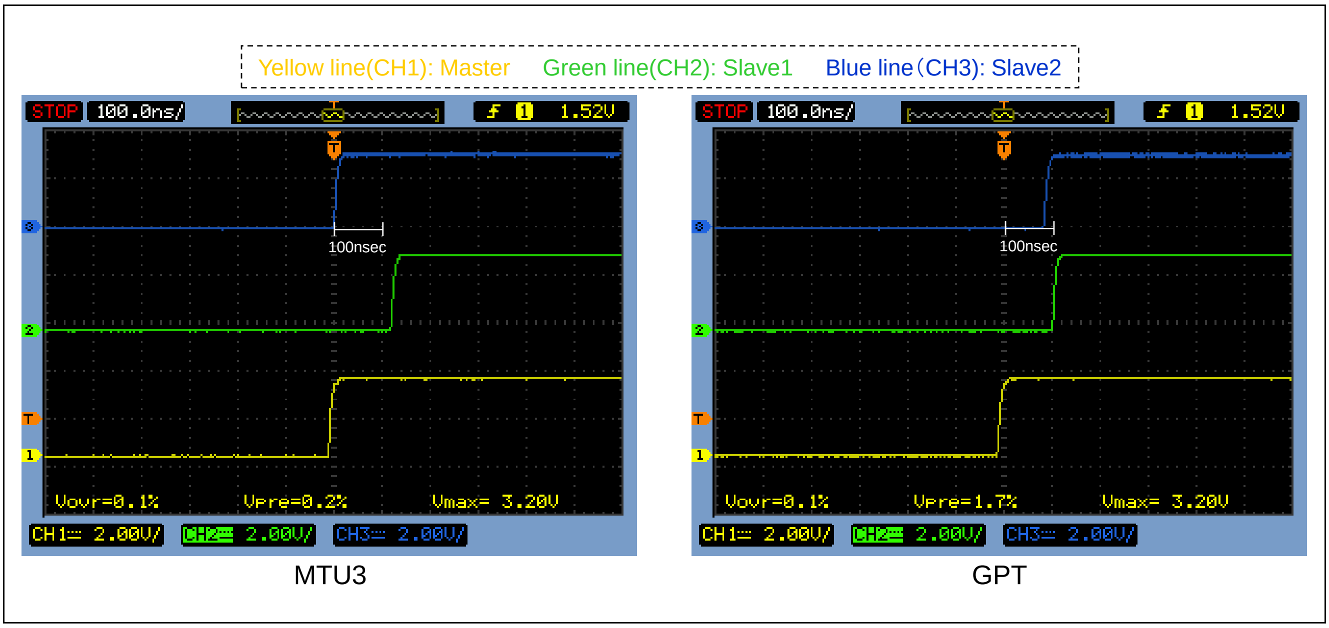

Figure 3.11 PWM wave example (100nsec scale unit)

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	Description: Summary
1.00	Jul 10, 2014	—	First edition issued.
1.01	Aug 20, 2014	—	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.
		14	Added and corrected PWM output pin setting.
1.12	Mar 31, 2017	—	Applied PTP driver Rev.1.13.
		—	Corrected timer start connection setting of ELC.

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.

Renesas Electronics Corporation

SALES OFFICES 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

Colophon 6.0