R01AN3475EJ0100
Rev.1.00
Jul 31, 2017

RX Family

Sample Program for Multi-Function Timer Fault Diagnosis

Introduction

This document describes a fault diagnosis example of a multi-function timer peripheral circuit using an EPTPC FIT (Firmware Integration Technology) module [1].

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

A multi-function timer peripheral circuit such as the Multi-Function Timer Pulse Unit (MTU3) or the General PWM Timer (GPT) is primary module to control the industrial machines. A fault diagnosis of that module also has become an important operation from Functional Safety [2] aspect.

This document explains the fault diagnosis example of the multi-function timer peripheral circuit (hereafter, timer module) using an EPTPC FIT module (hereafter, PTP (Precision Time Protocol) driver). This example compares a timer counter of the timer module to a local clock counter of the EPTPC and detects fault of timer module if those difference exceeds a threshold defined by previous fault diagnosis result adding correction value defined in advance. It is possible to implement the robust and accurate fault diagnosis of the timer module because the local clock counter of EPTPC is used as reference clock synchronizes an external clock in the IEEE1588 synchronous Ethernet network [3] and timer module relation control via Event Link Controller (ELC) of PTP driver [4].

1.1 Sample Program for Multi-Function Timer Fault Diagnosis

This example is implemented in a project and used as the application example of timer module fault diagnosis using plural FIT modules.

1.2 Related documents

[1] RX Family EPTPC Module Using Firmware Integration Technology, Rev.1.14, Document No. R01AN1943EJ0114, Apr 30, 2017

[2] Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC61508, Edition 2.0, Apr, 2010

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

[4] RX Family PTP Timer Synchronous Start Using Firmware Integration Technology Modules, Rev.1.12, Document No. R01AN1984EJ0112, Mar 31, 2017

[5] RX Family Board Support Package Module Using Firmware Integration Technology, Rev.3.50, Document No. R01AN1685EJ0350, May 15, 2017

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

[7] RX Family Flash Module Using Firmware Integration Technology, Rev.2.10, Document No. R01AN2184EU0210, Dec 20, 2016

[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 Hardware Structure

This example uses MTU3, GPT, EPTPC, ETHERC, EDMAC, ELC and Data Flash. The MTU3 or the GPT is used as the fault diagnosis target module. 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 MTU3, the GPT and I/O Ports via ELC to synchronous activation of any event. The Data Flash stores the Ethernet MAC address and previous fault diagnosis results applying the next fault diagnosis.
In detail, please refer to RX64M/71M Group User's Manual: Hardware.

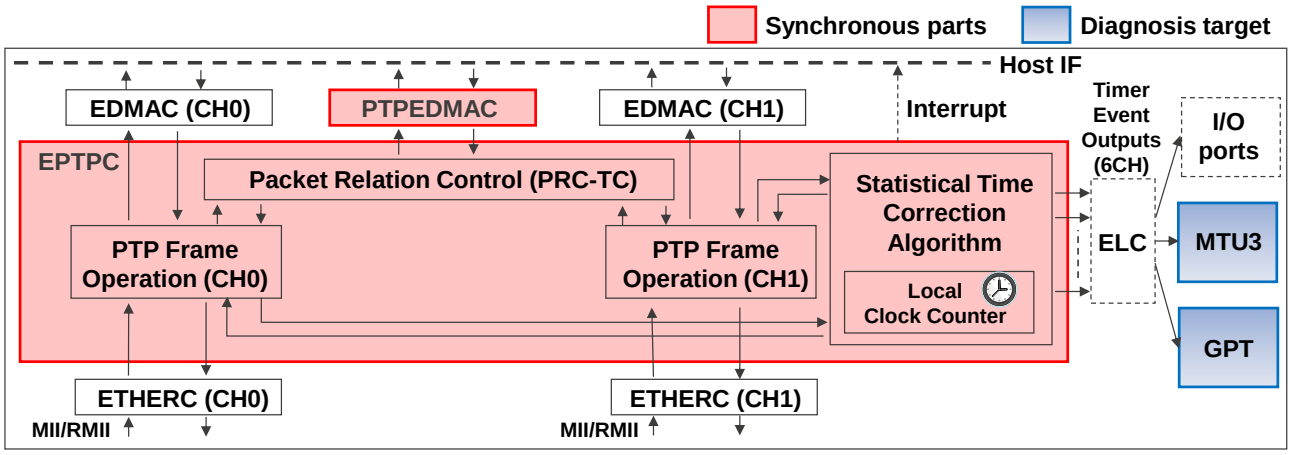

Figure 1.1 Hardware structure of this sample

1.4 Software Structure

This sample is operations example of the application layer applied to plural FIT modules whose common initial setting are supplied by a Board Support Package [5].

Figure 1.2 shows the typical structure and functional overview of the software. The application manages the operation sequence composed of the timer module fault diagnosis, the external clock synchronization, the diagnosis parameter setting, etc. The timer module fault diagnosis starts timer counter of MTU3/GPT and local clock counter of EPTPC synchronized each other, compares each counter value with margin due to several errors, and detects fault of timer module. The PTP driver does the synchronization of external clock based on the PTP protocol. The PTP driver always should be used with the Ethernet driver [6] doing standard Ethernet frame operation. 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. ELC driver connects EPTPC's event to MTU3/GPT's event, and makes enable to start timer counter of MTU3/GPT and local clock counter of EPTPC synchronized each other. MTU3/GPT driver set count mode, stops count operation, and reads current counter value. The flash driver (Flash API) [7] erases/writes the diagnosis parameter and MAC address from/to the Data Flash.

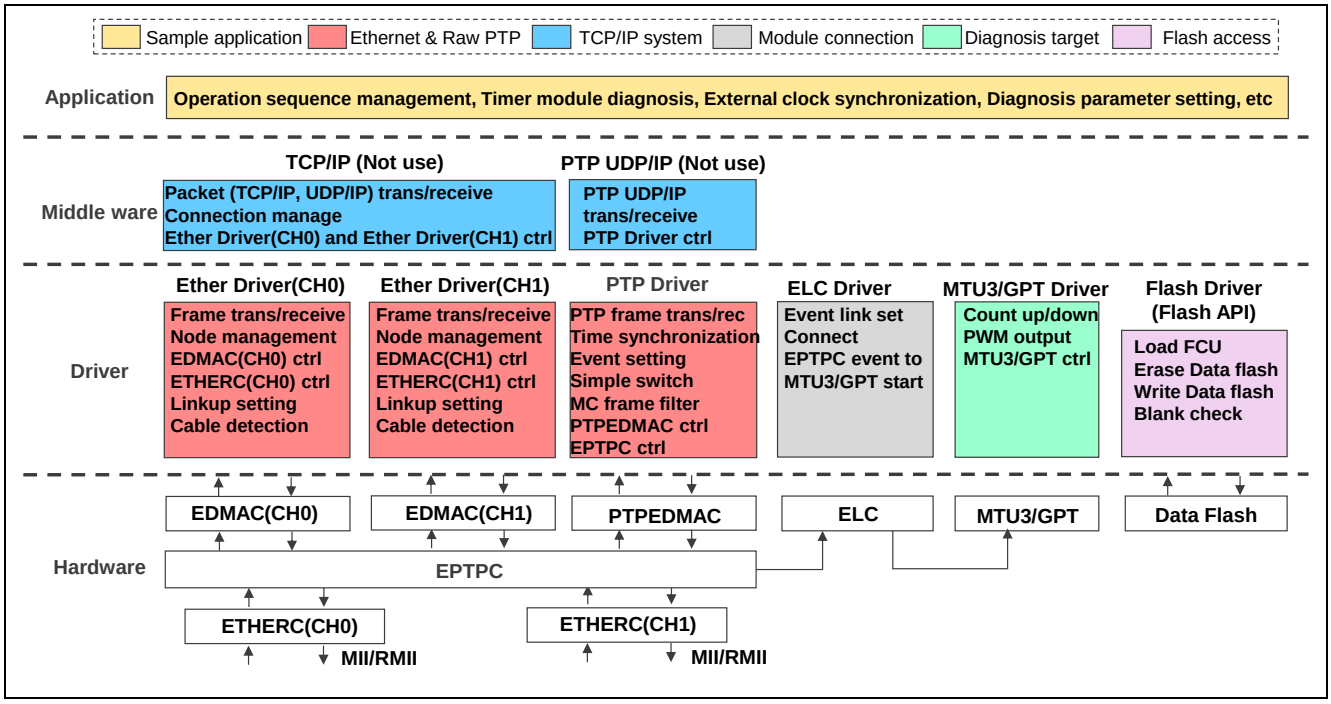

Figure 1.2 Software structure of this sample

1.5 File Structure

This sample codes are stored the “demo_src” and lower hierarchical folders. ELC, GPT and MTU3 drivers are stored the low hierarchical folders of “private_drivers” folder respectively as the sample customized driver. Figure 1.3 shows the file structure of this sample.

The “r_pincfg” folder includes pin setting files of Ethernet communication. As for the detailed information of the FIT based modules (BSP driver, Ethernet driver, Flash driver and PTP driver), please refer to the documentation of the each FIT module.

```
demo_src: main operation and configuration
|   sample_main.c
|   sample_main.h
|
+ --- flash_if: Data flash access operation
|       flash_if.c
|       flash_if.h
|
+ --- sync: PTP synchronize operation
|       sync.c
|       sync.h
|
+ --- usr: LED control
|       led.c
|       led.h

private_drivers: sample customized drivers (not FIT module)
|
+ --- r_elc_rx: ELC driver folder
|   r_elc_rx_if.h ;ELC driver header file
|
+ --- src:
|       r_elc.c ; ELC driver source file
|
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_config: configuration setting of FIT modules
|       r_bsp_config.h
|       r_bsp_interrupt_config.h
|       r_ether_rx_config.h
|       r_flash_rx_config.h
|       r_mcu_config.h
|       r_ptp_rx_config.h

r_bsp: BSP (Board Support Package) FIT module

r_ether_rx: Ethernet Driver FIT module

r_flash_rx: Flash Driver (Flash API) FIT module

r_pincfg: pin setting files folder

r_ptp_rx: PTP Driver 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 (PTPEDMAC)
- ELC
- MTU3
- GPT
- Data Flash

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 synchronizes with external devices based on the PTP and starts 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 do the Ethernet MAC operations.

2.2.3 EDMAC Channel (PTPEDMAC)

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

2.2.4 ELC

This example uses the ELC to connect events between EPTPC and MTU3/GPT. This resource starts the MTU3/GPT synchronously.

2.2.5 MTU3 Channel

This example uses the MTU3 channel as the fault diagnosis target device. This resource does count up operations.

2.2.6 GPT Channel

This example uses the GPT channel as the fault diagnosis target device. This resource does count up operations.

2.2.7 Data Flash

This example uses the data flash to store diagnosis parameter.

2.3 Software Requirements

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

- r_bsp
- r_ether_rx
- r_ptp_rx
- r_flash_rx

2.4 Supported Toolchains

This example is tested and works with the following tool chain:

- Renesas RX Toolchain v2.06.00

2.5 Header Files

Each functions call are accessed by including a single file, *sample_main.h, flash_if.h, sync.h, led.h, r_elc_rx_if.h, r_gpt_rx_if.h or r_mtu3_rx_if.h* which is supplied with this sample 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, flash_if.h, r_gpt_rx_if.h* and *r_mtu3_rx_if.h.* The option names and setting values are listed in the table below.

Configuration options	
#define DEVICE_ID **- Default value = 0**	Set the device id number of this sample. - Set 0 to 2. Other values are not supported in this sample.
#define MODE_PORT **- Default value = 1**	Specify the kind of clock. - When this is set to 0, clock is OC port0. - When this is set to 1, clock is OC port1. BC and TC are not supported in this sample. If select OC port0, please set "ETHER_CFG_CH0/1_PHY_ACCESS = 0" in the r_ether_rx_config.h. If select OC port1, please set "ETHER_CFG_CH0/1_PHY_ACCESS = 1" in the r_ether_rx_config.h.
#define MS_PORT0/1 **- Default value = 0**	Select Master or Slave for port0/port1 - When this is set to 0, clock is Master. - When this is set to 1, clock is Slave.
#define SYNC_PORT0/1 **- Default value = 1**	Select the delay mechanism (P2P or E2E) for port0/port1 - When this is set to 0, the delay mechanism is P2P. - When this is set to 1, the delay mechanism is E2E.
#define SAMPLE_VID **- Default value = 0x00749050**	Set the vendor ID of the Ethernet MAC address. The default value is set the Renesas vendor ID (=74-90-50). Please change this value when users applied to this sample on their own system.
#define MAC_ADDR_1H/2H **- Default value = 0x00749050 (port0)** **- Default value = 0x00749050 (port1)**	Set the Ethernet MAC address upper 24 bits for port0/port1. The lower 24 bits are vendor ID field and default values are set the Renesas vendor ID (=74-90-50). The upper 8 bits of default value are reserved field and should be set 00. Please change this value when users applied to this sample on their own system.
#define MAC_ADDR_1L/2L **Case of device id = 0,** **- Default value = 0x0000795F (port0)** **- Default value = 0x00007960 (port1)** **Case of device id = 1,** **- Default value = 0x00007961 (port0)** **- Default value = 0x00007962 (port1)** **Case of device id = 2,** **- Default value = 0x00007963 (port0)** **- Default value = 0x00007964 (port1)**	Set the Ethernet MAC address lower 24 bits for port0/port1. The lower 24 bits are product ID field and default values are set the unique value for this sample. The upper 8 bits are reserved field and should be set 00. Please change these values when users applied to this sample on their own system.
#define IP_ADDR_1/2 **Case of device id = 0,** **- Default value = 0x0A0B0C0D (port0)** **- Default value = 0x1A1B1C1D (port1)**	Set the IP (IPv4) address for port0/port1. Please change this value when users applied to this sample on their own system.

Configuration options	
Case of device id = 1, - Default value = 0x2A2B2C2D (port0) - Default value = 0x3A3B3C3D (port1) Case of device id = 2, - Default value = 0x4A4B4C4D (port0) - Default value = 0x5A5B5C5D (port1)	
#define DIAG_START_SEC - Default value = 15	Set second unit of diagnosis start time. - Set 0 to 0xFFFFFFFF(= 4,294,967,295)
#define DIAG_START_NSEC - Default value = 0	Set nano second unit of diagnosis start time. - Set 0 to 999,999,999
#define STCA_TMR_CH - Default value = 0	Select the channel of STCA pulse output timer. - Set 0 to 5
#define STCA_TMR_EDGE - Default value = 0	Select the rise or fall edge of ELC event signal trigger. - When this is set to 0, rising edge is selected. - When this is set to 1, falling edge is selected. Set 0 (= rising edge) in this sample.
#define DIAG_TMR_KND - Default value = 0	Select the diagnosis target multi-function timer. - When this is set to 0, the diagnosis target timer is MTU3. - When this is set to 1, the diagnosis target timer is GPT.
#define DIAG_TMR_CH - Default value = 0	Select the diagnosis target multi-function timer's channel. Case of MTU3, 0:CH0, 3:CH3 or 4:CH4 is selectable. Case of GPT, 0:CH0, 1:CH1, 2:CH2 or 3:CH3 is selectable. Other values are not supported in this sample.
#define DIAG_TMR_FREQ - Default value = 1	Select the diagnosis target multi-function timer's count up frequency. - When this is set to 0, frequency is 120MHz (=PCLKA/1). - When this is set to 1, frequency is 60MHz (=PCLKA/2). - When this is set to 2, frequency is 30MHz (=PCLKA/4). - When this is set to 3, frequency is 15MHz (=PCLKA/8). Other values are not supported in this sample.
#define INIT_TMR_THRESH - Default value = 1000000	Set timer comparison initial threshold value. Setting value is ns unit.
#define COR_TMR_THRESH - Default value = 1000	Set timer comparison threshold correction value (add to mean absolute difference of the previous diagnosis). Setting value is ns unit.
#define DIAG_PERIOD - Default value = 300000000	Set timer diagnosis period. Setting value is ns unit.
#define FORCE_ERASE_BPAT - undefined	Erase diagnosis parameter from the data flash or not? - If defined, forcing erase the parameter from the data flash after current operation completed.
#define USE_HYST_THRESH - undefined	Set threshold value to hysteresis data on the data flash or not? - If defined, threshold is set from hysteresis data. - If undefined, initial threshold value (=INIT_TMR_THRESH) is set.
#define NUM_DEV - Default value = 3	Set the total number of device id of this sample. Other values are not supported in this sample.
#define GPT_CFG_INTERRUPT_LEVEL - Default value = 3	Specifies interrupt priority levels of GPT interrupts. Specify the level between 1 and 15.
#define MTU3_CFG_INTERRUPT_LEVEL - Default value = 3	Specifies interrupt priority levels of MTU3 interrupts. Specify the level between 1 and 15.

2.8 Data Structures

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

```
/* Communication parameter */
typedef struct
{
   uint32_t vd; /* MAC address vendor ID field */
   uint32_t pd; /* MAC address product ID field */
} CommParm;
```

```
/* Timer diagnosis result hysteresis */
typedef struct
{
   uint32_t num;  /* Number of sampling */
   uint32_t max;  /* Maximum absolute difference */
   uint32_t min;  /* Minimum absolute difference */
   uint32 t mean; /* Mean absolute difference */
} DiagHyst;
```

```
/* Communication parameter segment address */
const flash_block_address_t comm_addr[NUM_DEV] =
{ /* Refer to "r_flash_rx/src/targets/rx64m/r_flash_rx64m.h or
rx71m/r_flash_rx71m.h". */
   FLASH_DF_BLOCK_256, /* ID = 0: 0x00104000 to 0x0010403F (64B) */
   FLASH_DF_BLOCK_257, /* ID = 1: 0x00104040 to 0x0010407F (64B) */
   FLASH_DF_BLOCK_258, /* ID = 2: 0x00104080 to 0x001040BF (64B) */
};
```

```
/* Timer diagnosis result hysteresis address */
const flash_block_address_t diag_addr[NUM_DEV] =
{ /* Refer to "r_flash_rx/src/targets/rx64m/r_flash_rx64m.h or
rx71m/r_flash_rx71m.h". */
   FLASH_DF_BLOCK_320, /* 0x00105000 to 0x0010503F (64B) */
   FLASH_DF_BLOCK_321, /* 0x00105040 to 0x0010507F (64B) */
   FLASH_DF_BLOCK_322, /* 0x00105080 to 0x001050BF (64B) */
};
```

2.9 Return Values

This section describes return values of the functions of this example. Those return values are located in *flash_if.h, r_elc_rx_if.h, r_mtu3_rx_if.h and r_gpt_rx_if.h* as the prototype declarations.

```
/* Data flash access return value */
typedef enum
{
    FLSIF_ERR_ERASE = -5,  /* Flash erase error */
    FLSIF_ERR_WRITE = -4,  /* Flash write error */
    FLSIF_ERR_VERIFY = -3, /* Flash verify error */
    FLSIF_ERR_PARAM = -2,  /* Parameter error */
    FLSIF_ERR = -1,        /* General error */
    FLSIF_OK = 0,
    FLSIF_BLANK = 1,       /* Flash was blank */
} flshif_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 */
```

3. Specification of This Example

3.1 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), and Ethernet cables.
The outline of the execution sequence is following.

- Write the project execution code to a code Flash of in the RX64M/71M RSK boards (hereafter RSK boards).
- Connect all RSK boards one another via the Hub by the Ethernet cable.
- Power on the all RSK boards.
- When a RSK board finishes EPTPC, Ethernet, Flash, ELC, MTU3 and GPT driver initialization process, a user LED composed of LED3, LED2, LED1 and LED0 shows the all-on pattern (LED3: ON, LED2: ON, LED1: ON, LED0: ON). The MAC addresses[2] assigned “74-90-50-00-79-5F” for CH0 and “74-90-50-00-79-60” for CH1 in the source file in advance.
- Push the SW1 switch, the RSK board opens EPTPC and Ethernet drivers. The user LED shows the “1” pattern (LED3: OFF, LED2: OFF, LED1: OFF, LED0: ON) after synchronization started.
- The RSK board set the diagnosis starting time that is 30 second after synchronization started.
- After the diagnosis start time, the RSK board starts the multi-function timer diagnosis.
- Push the SW2 switch, the RSK board finishes the multi-function timer diagnosis and the user LED shows the even pattern (LED3: OFF, LED2: ON, LED1: OFF, LED0: ON) pattern.
- If a fault or any other error detected during the operation, the user LED shows the odd pattern (LED3: ON, LED2: OFF, LED1: ON, LED0: OFF) pattern.

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

[2] Please change this value when users applied to this sample their own system.

3.2 Operation Sequence

In this section, explain an operation sequence in this example. Figure 3.1 explains the operation overview in the two boards configuration when MTU3 is applied as the multi-function timer and a fault detected at t(N) (i = 1, 2, 3,,, N). t(N) indicates the time of nth PTP timer start event of EPTPC and δ is infinitesimal delay from t(N) (δ << t(N) – t(N-1)).

(1) Before t (0): Start synchronization between a master and a slave by PTP protocol.
(2) At t(0): Set PTP timer start event interrupt.
(3) At t(1): Connect ELC event to start MTU3 counter's count-up operation by the PTP timer start event.
(4) At t(2): Start MTU3 counter's count-up operation by the PTP timer start event via ELC.
(5) At t(3): Stop MTU3 counter's count-up operation.
(6) At t(3) + δ: Compare the absolute difference between MTU3 counter and EPTPC advanced local clock counter value from the previous interrupt at t(2). If the difference is within threshold value, no fault was detected.
(7) At t(4): Start MTU3 counter's count-up operation by the PTP timer start event via ELC.
(8) At t(5): Stop MTU3 counter's count-up operation.
(9) At t(5) + δ: Compare the absolute difference between MTU3 counter and EPTPC advanced local clock counter value from the previous interrupt at t(4). If the difference is within threshold value, no fault was detected.
(10) At t(N-1): Start MTU3 counter's count-up operation by the PTP timer start event via ELC.
(11) At t(N): Stop MTU3 counter's count-up operation.
(12) At t(N) + δ: Compare the absolute difference between MTU3 counter and EPTPC advanced local clock counter value from the previous interrupt at t(N-1). If the difference deviates from threshold value, any fault was detected.

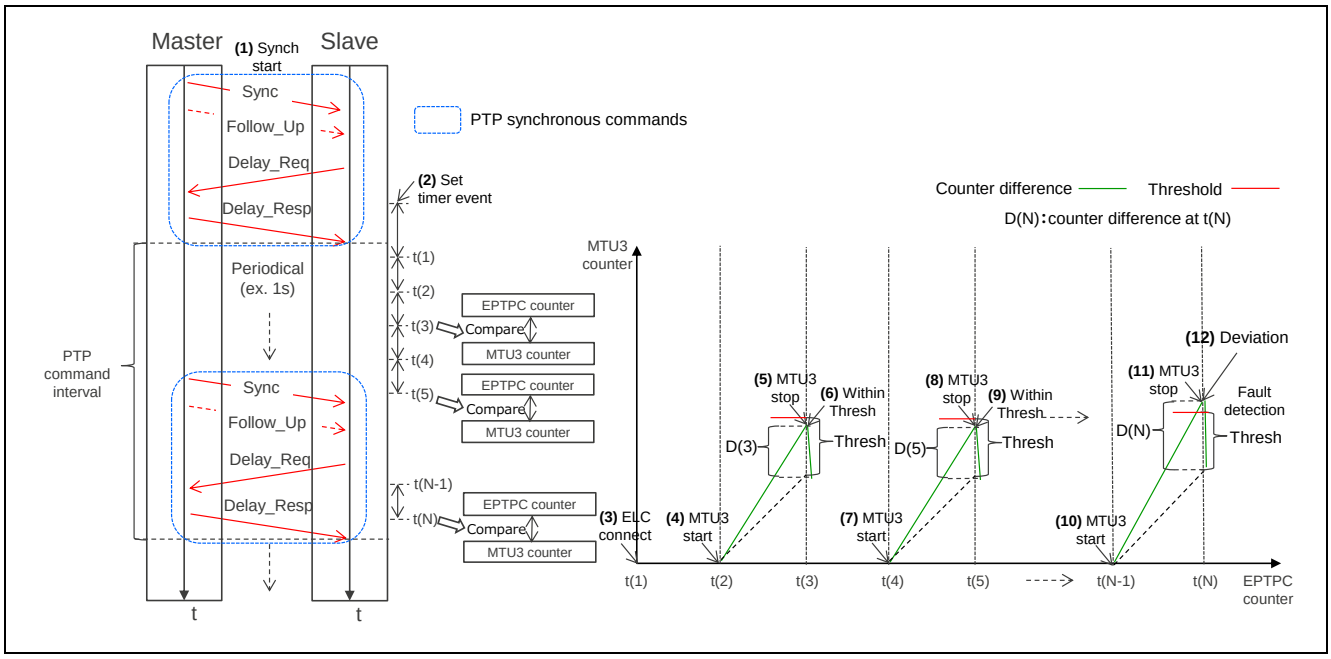

Figure 3.1 Operation overview

3.3 Software Operation Flow

In this section, describes the software operation flow of this sample.
Figure 3.2 and Figure 3.3 show the initial setting drivers and time synchronization respectively. Figure 3.4 shows the static routine whose main operation is diagnosis of multi-functional timer. Figure 3.5 shows the operation to compare multi-functional timer counter to local clock counter of EPTPC and to judge whether fault or not by threshold value. Figure 3.6 shows the calculating operation to update diagnosis hysteresis. Figure 3.7 and Figure 3.8 show the parameter loading operation from data flash and the parameter updating operation to data flash respectively.

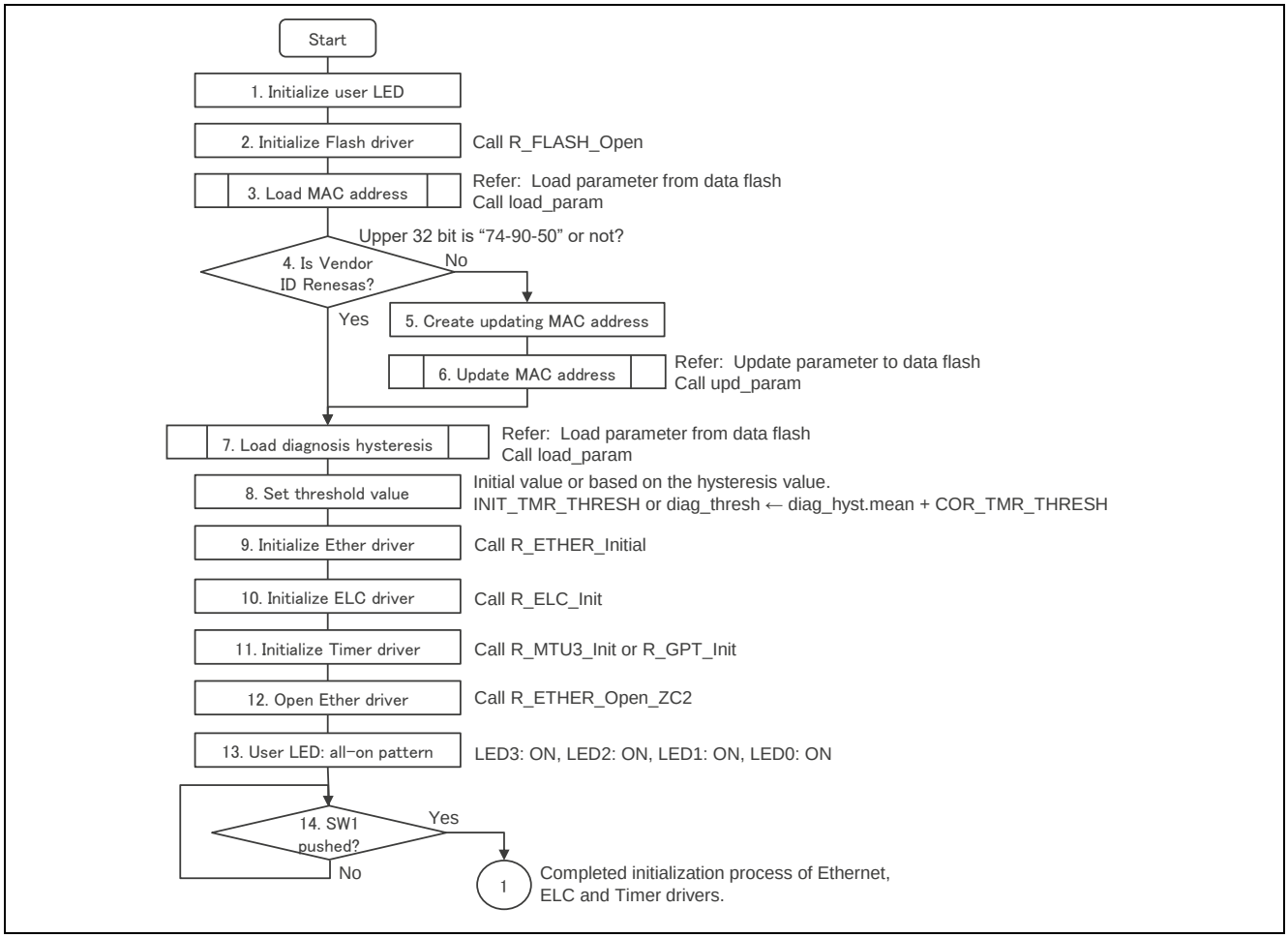

Figure 3.2 Drivers initial setting

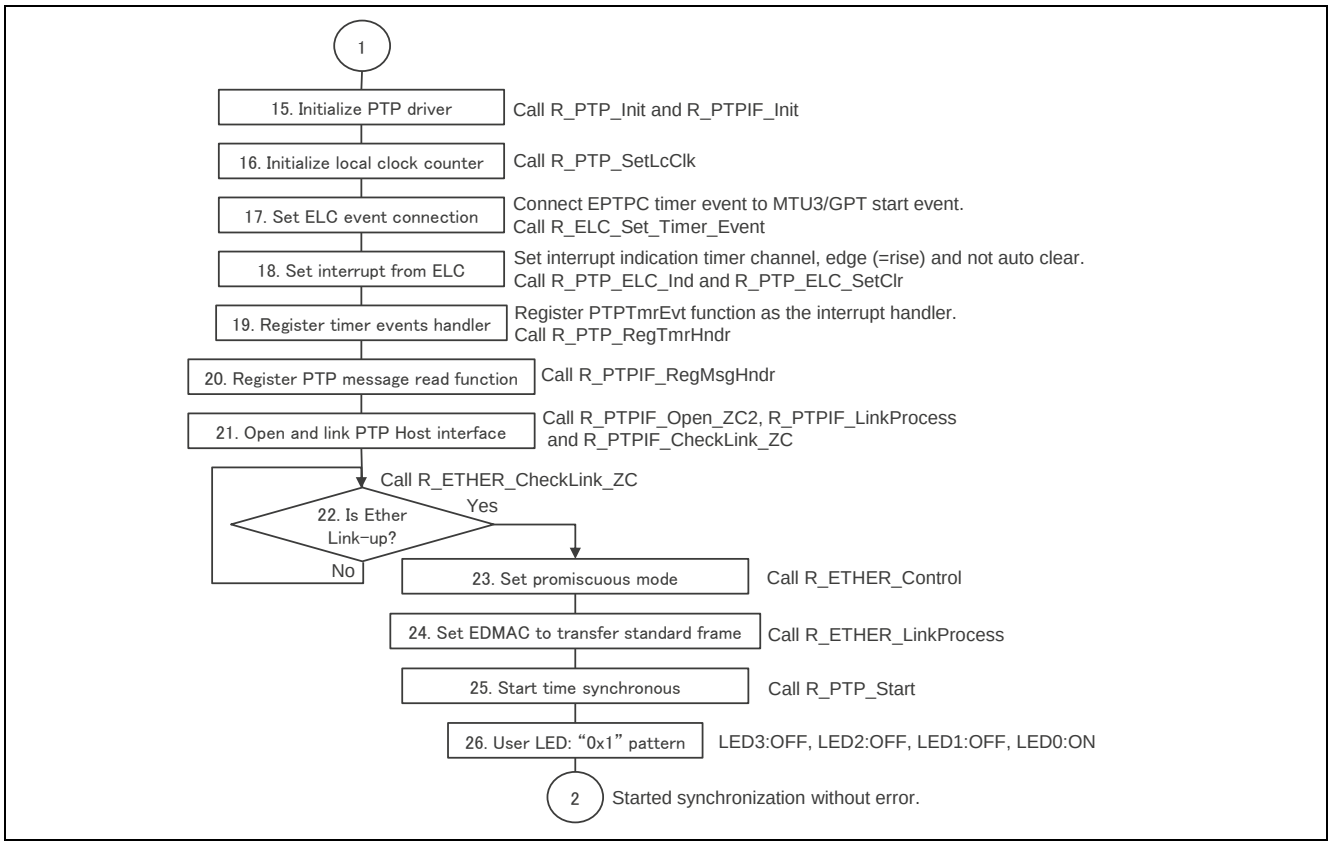

Figure 3.3 Synchronization initial setting

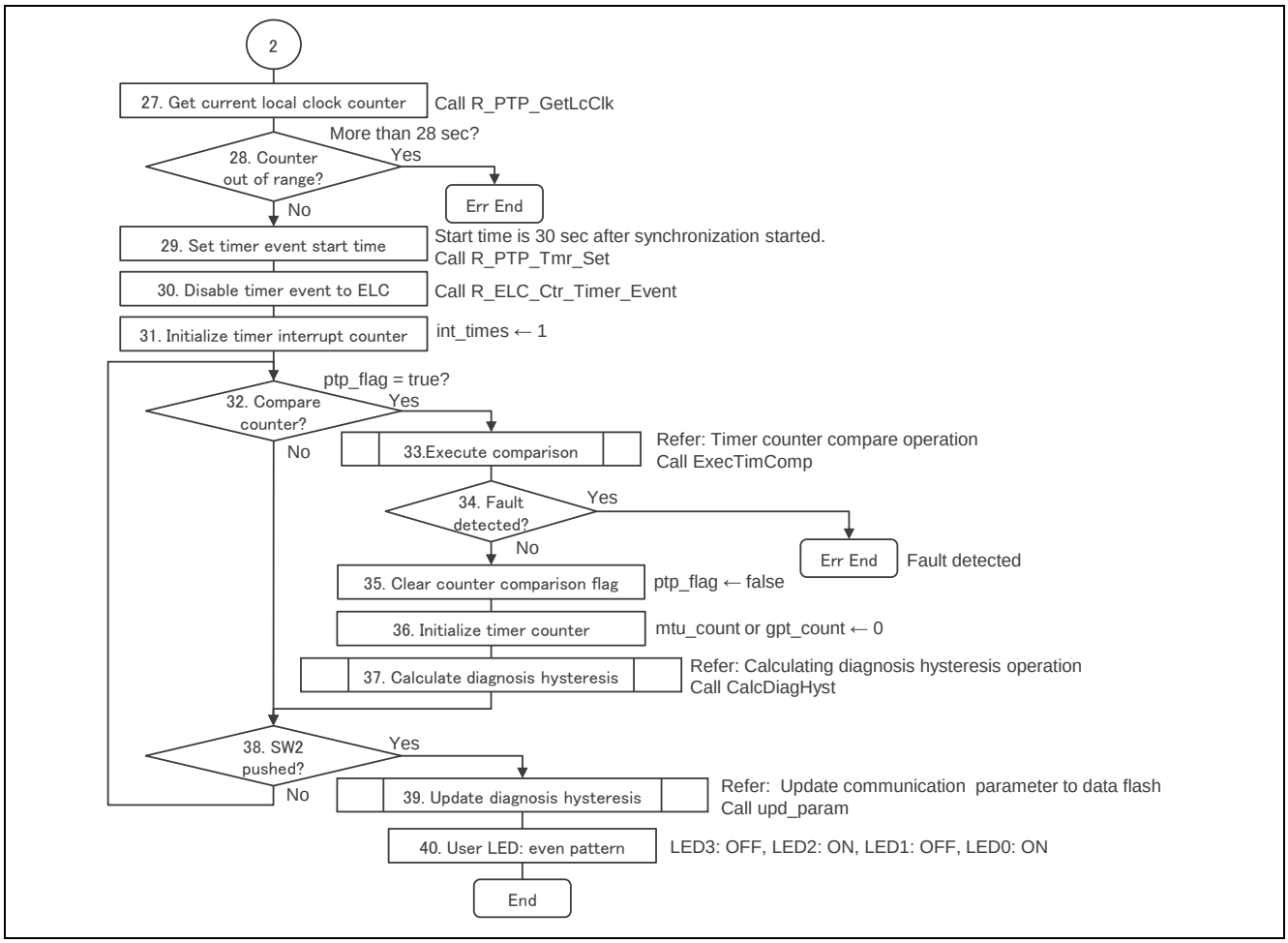

Figure 3.4 Multi-functional timer diagnosis operation

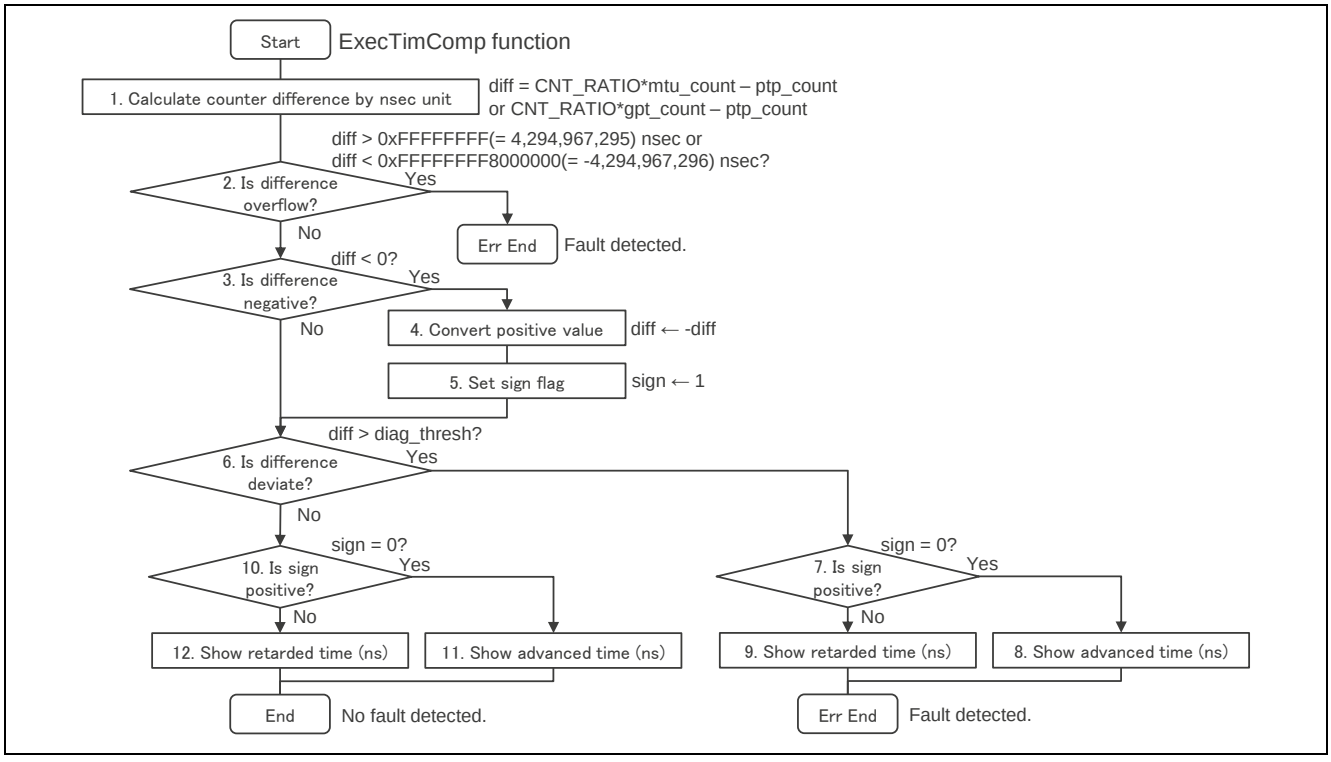

Figure 3.5 Multi-functional timer comparison's operation

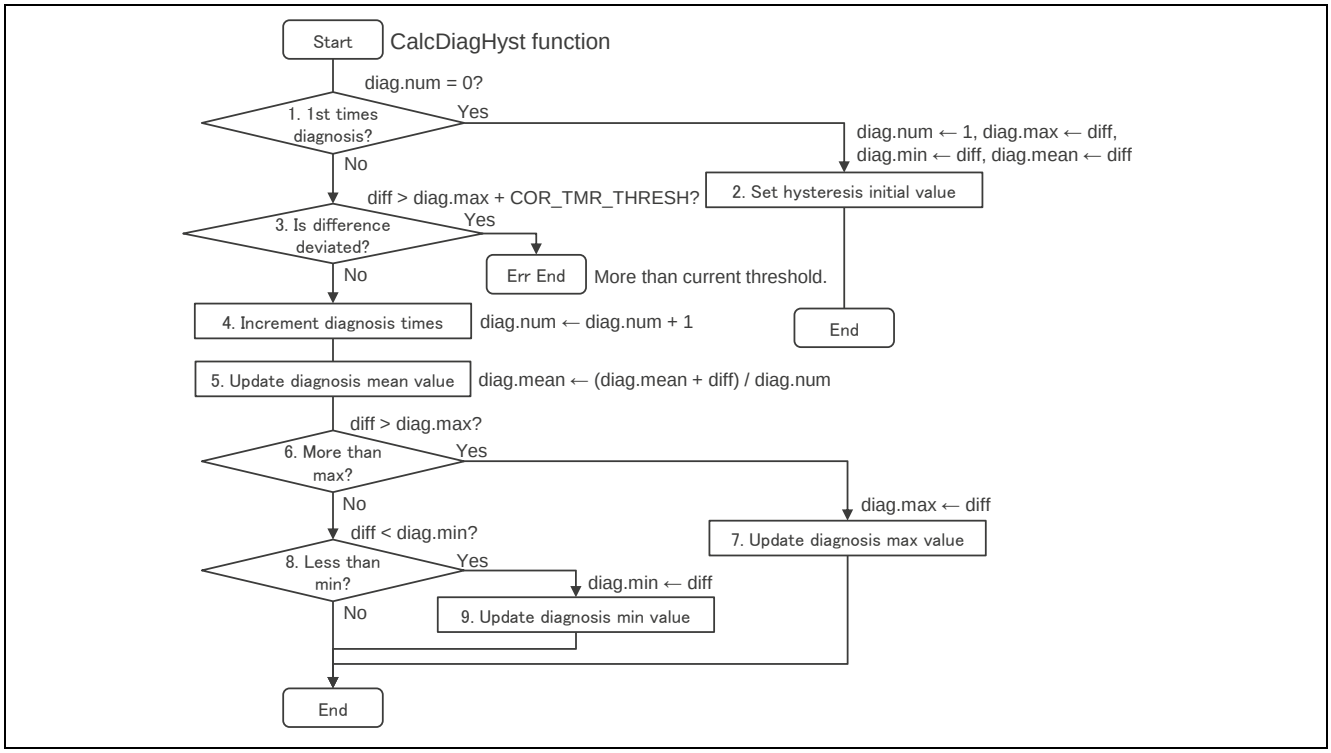

Figure 3.6 Calculating diagnosis hysteresis

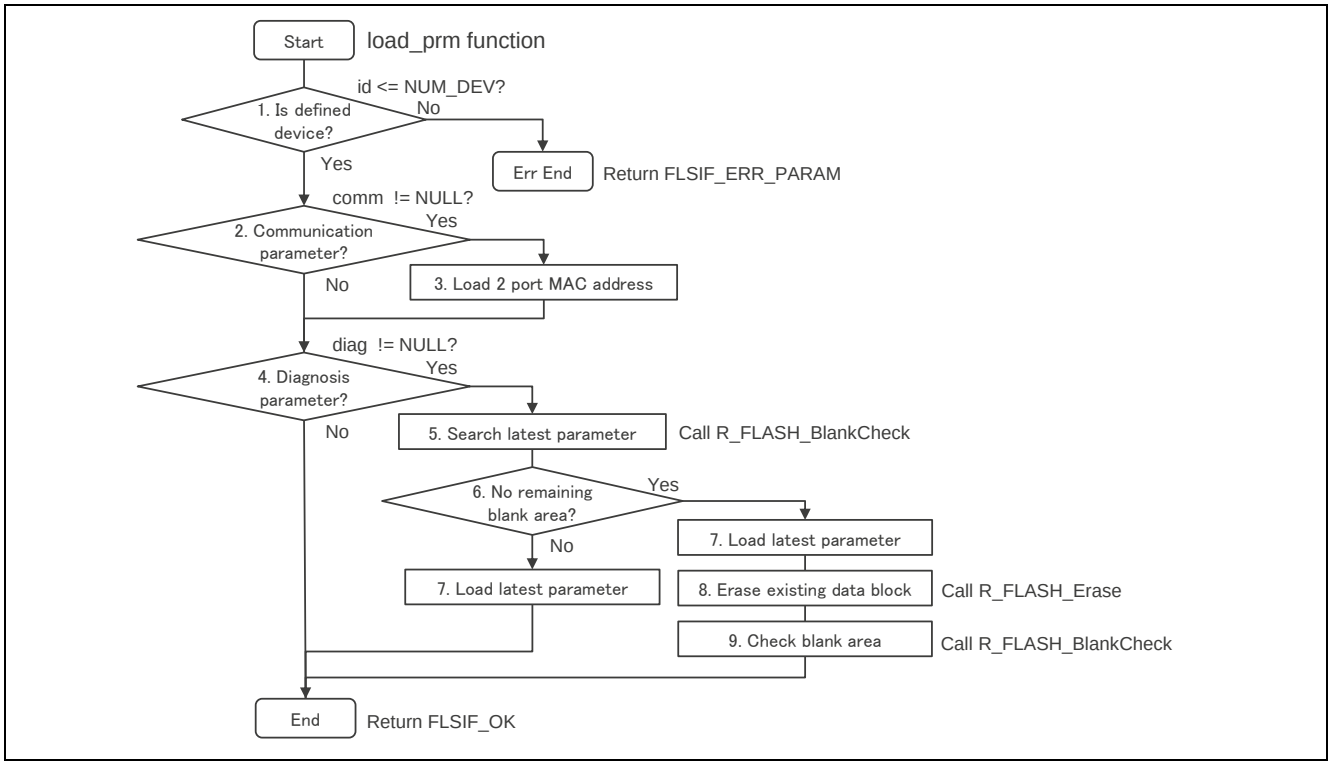

Figure 3.7 Parameter loading operation

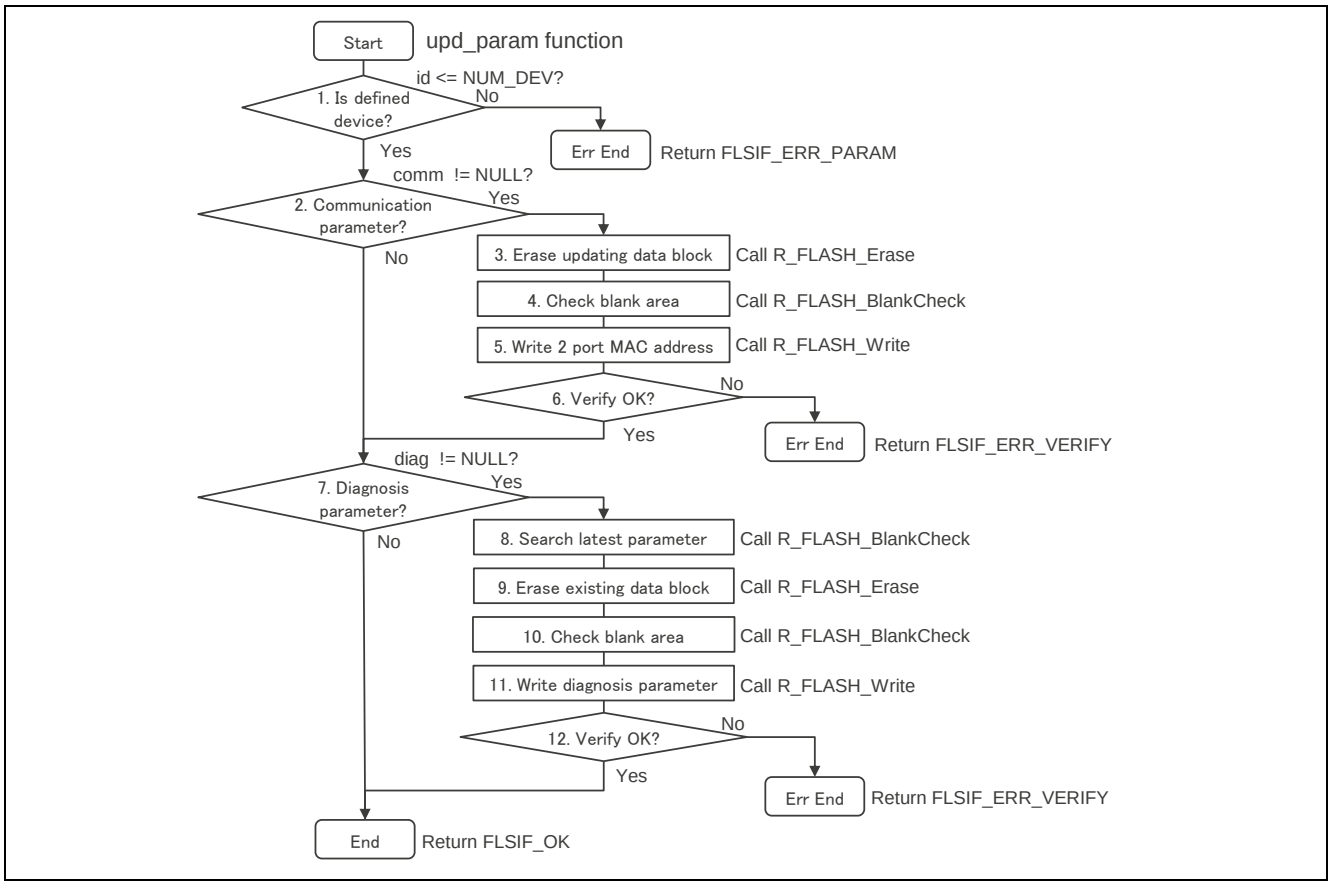

Figure 3.8 Parameter updating operation

3.4 Example of Operation Results

Figure 3.9 and Figure 3.10 show a diagnosis result message when no fault was detected and any fault detected respectively. Those messages are captured in the “Renesas Debug Console” equipped with the e2 studio.

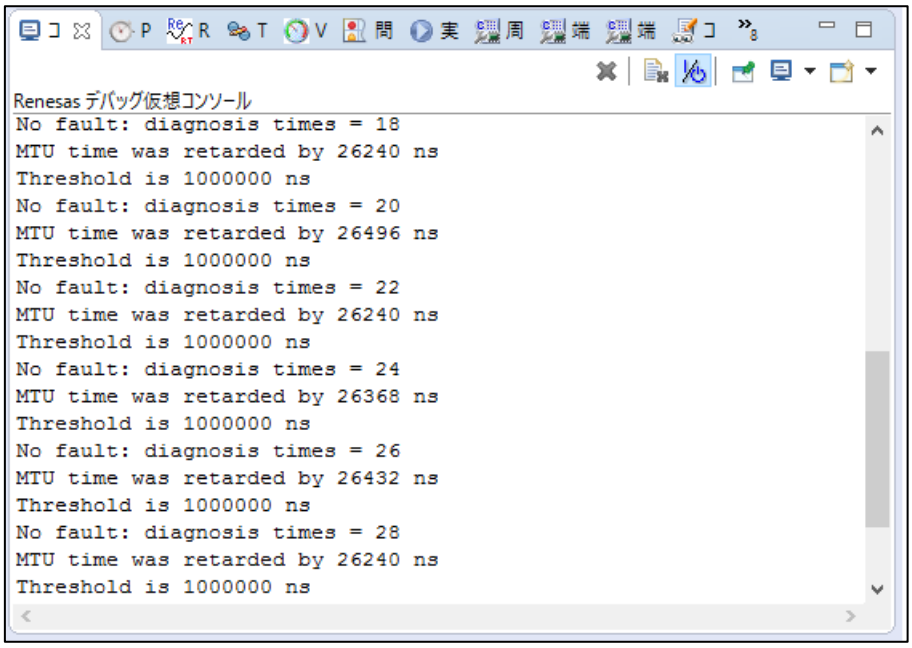

Figure 3.9 Result Message (No fault detection)

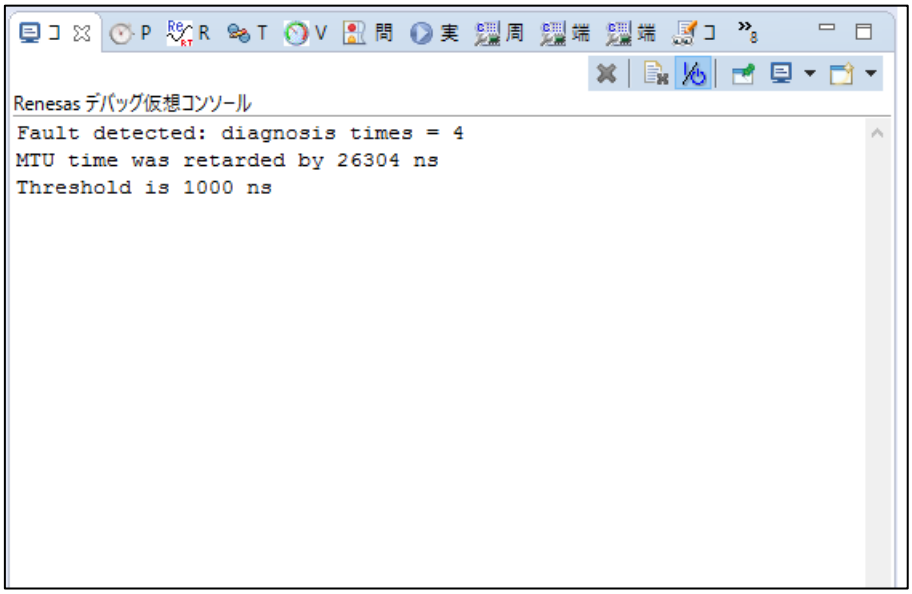

Figure 3.10 Result Message (Fault detection)

3.5 Board Setting

There are two jumpers setting of the RX64M/71M RSK board depending on the PHY access channel of the configuration option. When the product name of the RX64M/71M RSK board is R0K50564MC001BR or R0K5RX71MC010BR, Figure 3.11 indicates their changing. And when the product name of the RX71M RSK board is R0K50571MC000BR, Figure 3.12 indicates their changing.

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

[1] In case of MODE_PORT = 1, LINK_CH = 1 is selected.
[2] In case of MODE_PORT = 0, LINK_CH = 0 is selected.

Figure 3.11 Jumper setting

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

[1] In case of MODE_PORT = 1, LINK_CH = 1 is selected.
[2] In case of MODE_PORT = 0, LINK_CH = 0 is selected.

Figure 3.12 Jumper setting

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/

All trademarks and registered trademarks are the property of their respective owners.

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jul 31, 2017	—	First edition issued.

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)

RENESAS

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

© 2017 Renesas Electronics Corporation. All rights reserved.

Colophon 6.0