RL78/G13

Data Flash Library Type04 CC-RL

R01AN2827EJ0100
Rev. 1.00
May 28, 2015

Introduction

This application note explains how to writes and reads data to and from data flash memory using the Data Flash Library Type04 (Data Flash Library).

Target Device

RL78/G13

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications

This application note explains how to use the data flash library.

The sample program covered in this document displays the target write address, the write value, and the read value on the LCD. The sample program can be manipulated using three switches which are used to set the write value, set the target write address, and program into the data flash memory, respectively. Data in the data flash memory is erased by pressing the two switches for setting the write value and target write address at the same time for one second. The display on the LCD is updated every time the write value and the target write address are changed. The read value is updated after power is turned on, data is programmed, or it is erased.

Table 1.1 lists the peripheral functions to be used and their uses.

Table 1.1 Peripheral Functions to be Used and their Uses

Peripheral Function	Use
Port I/O	Displays text on the LCD. Turns on and off LED0.
Interval timer	Generates the wait time for avoiding chatters.
External interrupt input (INTP1)	Sets the write data. Erases the data flash memory (pressed long together with INTP2).
External interrupt input (INTP2)	Sets the write data. Erases the data flash memory (pressed long together with INTP1).
External interrupt input (INTP4)	Executes programming.

Figure 1.1 shows the outline of the operation to display the value on the LCD.

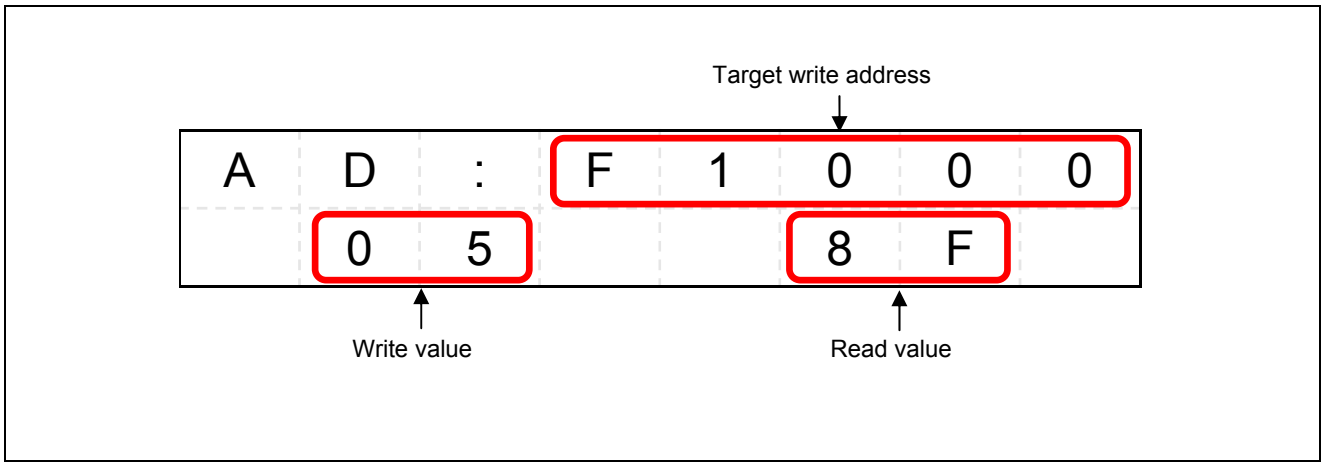

Figure 1.1 Operation to Display Value on the LCD

1.1 Outline of the Data Flash Library

The data flash library is a software library that is used to manipulate the data flash memory using the firmware installed on the RL78 microcontroller.

The data flash library carries out the reprogramming and reading of the data flash memory by being called by the user program. For notes and cautions with respect to the hardware and software environments of the data flash library, refer to RL78 Family Data Flash Library Type04 User's Manual (R01US0049E).

1.2 Hardware Environment of the Data Flash Library

The Data Flash Library Type04 for the RL78 microcontrollers controls the reprogramming of the data flash memory using a sequencer. Since the control of data flash memory is carried out by the sequencer, it is possible to run user programs while the data flash memory is being controlled. This is referred to as BGO (background operation).

Although the data flash memory cannot be referenced while it is being subjected to reprogramming, the code flash memory can be referenced during that period. Consequently, interrupt processing routines, user programs, and the Data Flash Library Type04 can be allocated to ROM as usual for execution.

Figure 1.2 shows a reprogramming state of data flash memory. Figure 1.3 shows a reprogramming control example of data flash memory.

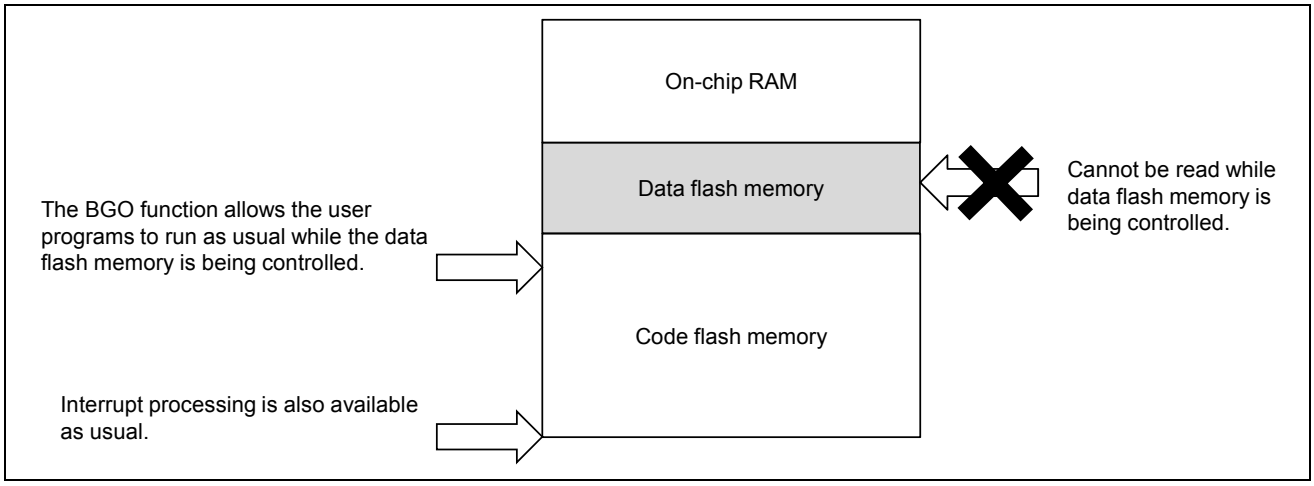

Figure 1.2 Reprogramming State of Data Flash Memory

Control is returned to the user program immediately after a call for executing the required processing is made to the sequencer of the RL78 microcontroller. For the result of controlling the data flash memory, the user program needs to check the data flash memory control state by calling a status check function (PFDL_Handler function).

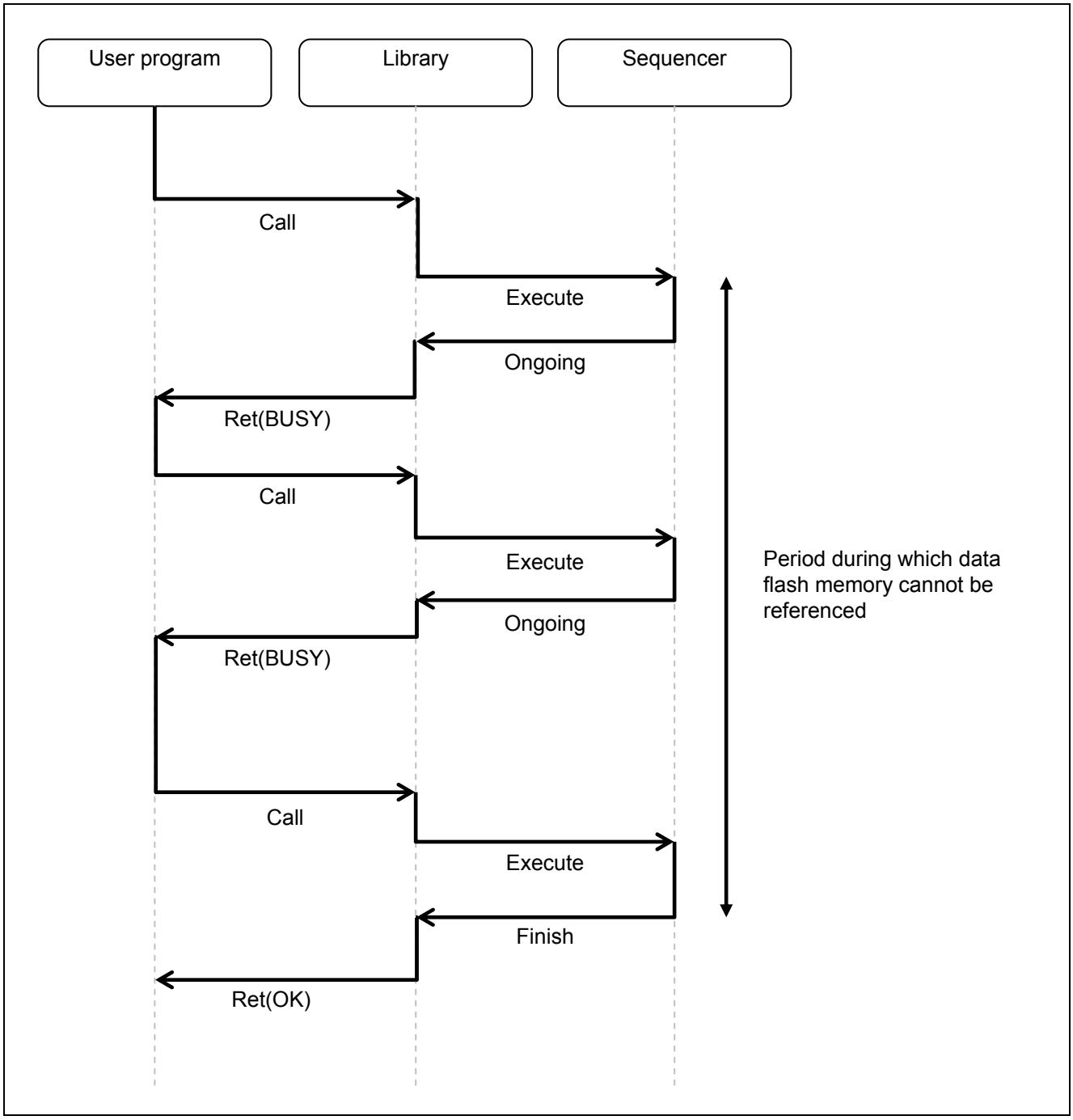

Figure 1.3 Reprogramming Control Example of Data Flash Memory

1.2.1 Data Flash Memory

The configuration of the data flash memory for the RL78/G13 (R5F100LE) is shown below.

The flash memory of the RL78 microcontrollers is divided into 1-Kbyte blocks. The data flash library performs erase processing on the data flash memory on a block basis. It specifies the start address and the execution size when performing read, write, blank check, and internal verify processing.

Figure 1.4 shows the placement of blocks in data flash memory and their block numbers.

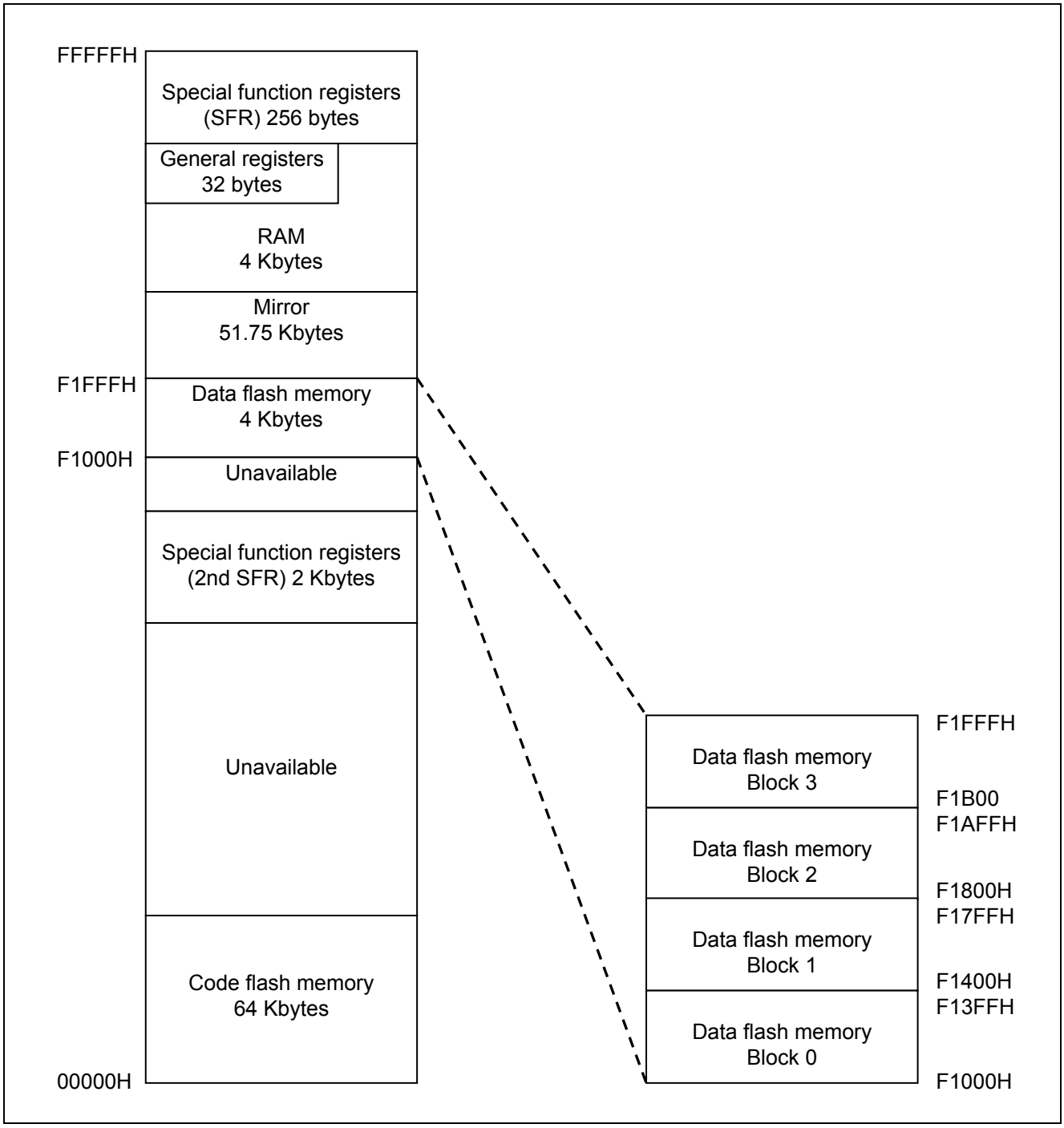

Figure 1.4 Placement of Blocks in Data Flash Memory and their Block Numbers

1.3 Software Environment of the Data Flash Library

The Data Flash Library Type04 consumes the volume of program area equal to the size of the library to be used to allocate the correspondent program to the user area. The Data Flash Library Type04 uses the CPU, stack, and data buffer.

1.3.1 Self-RAM

The Data Flash Library Type04 sometimes uses 1 Kbyte of RAM as a work area. This area is called the self-RAM when it is used as a work area. Since it is defined within the library, the user needs to make no setting for this area.

Data in the self-RAM area is rewritten by calling a data flash library function.

1.3.2 Register Bank

The Data Flash Library Type04 uses the general registers, ES/CS registers, SP, and PSW on the register bank that is selected by the user.

1.3.3 Stack Data Buffer

The Data Flash Library Type04 uses a sequencer to perform programming into the data flash memory. It uses the CPU for preliminary configuration and control. Accordingly, the stack that is designated by the user program is required to use the Data Flash Library Type04.

Caution: Link directives are used to allocate the stack and data buffer to the user-specified addresses.

- Stack
  It is necessary to reserve in advance the size of stack area necessary for the data flash library functions in addition to the size of the stack to be used by the user programs and allocate it in such a manner that the RAM that is being used by the user is not destroyed during the stack processing that is executed for the Data Flash Library Type04. The areas for the stack that can be specified are the self-RAM and the internal RAM area excluding addresses FFE20H to FFEFFH.
- Data buffer
  The uses of the data buffer are listed below.
  — As the work area for internal processing of the Data Flash Library Type04
  — As the area for storing the programming data in write mode
  As the area for storing the read data in read mode
  The start address of the data buffer must fall within the self-RAM or the internal RAM area excluding addresses FFE20H to FFEFFH as for the stack.

1.4 How to Get the Data Flash Library

Before compiling the sample program, please download the latest data flash library and copy the library files to the following folders below “r01an2827_fld”.

incrl78 folder : pfdl.h, pfdl.inc, pfdl_types.h

librl78 folder : pfdl.lib

The Data Flash library can be obtained from the following URL:

http://www.renesas.com/products/tools/flash_prom_programming/flash_libraries/index.jsp

2. Operation Check Conditions

The sample code described in this application note has been checked under the conditions listed in the table below.

Table 2.1 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G13 (R5F100LEA)
Operating frequency	• High-speed on-chip oscillator (HOCO) clock: 32 MHz • CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (Operation is possible over a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVD}): Reset mode which uses 2.81V (2.76 V to 2.87 V)
Integrated development environment (CS+)	CS+ V3.01.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.01.00 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio V4.0.0.26 from Renesas Electronics Corp.
C compiler (e^2 studio)	CC-RL V1.01.00 from Renesas Electronics Corp.
Board to be used	Renesas Starter Kit for RL78/G13 (R0K50100LS000BE)
Data Flash Library (Type, Ver)	FDLRL78 Type04, Ver 1.05 Note

Note: Use and evaluate the latest version.

3. Related Application Notes

The application notes that are related to this application note are listed below for reference.

- RL78/G13 Initialization (R01AN2575E) Application Note

4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows hardware configuration that is used for this application note.

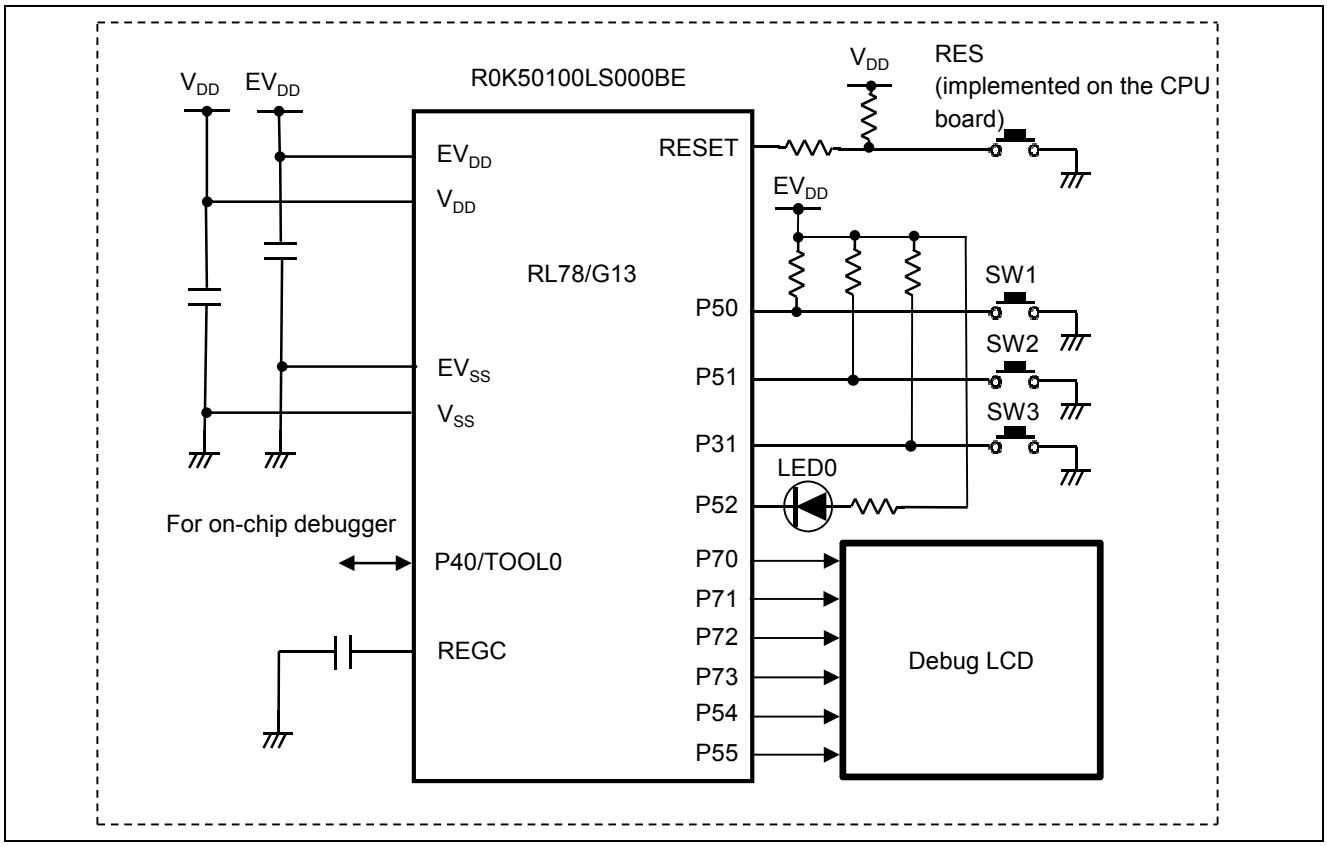

Figure 4.1 Hardware Configuration

Cautions:
1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).
2. V_{DD} must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

4.2 List of Pins to be Used

Table 4.1 lists pins to be used and their functions.

Table 4.1 Pins to be Used and their Functions

Pin Name	I/O	Description
P31/TI03/TO03/INTP4	Input	Executes writing.
P50/INTP1/SI11/SDA11	Input	Sets the target write address. Erases data flash memory (pressed long together with INTP2).
P51/INTP2/SO11	Input	Increments "write value." Erases data flash memory (pressed long together with INTP1).
P52	Output	LED (indicating flash memory access status) on/off control
P54	Output	Debug LCD control
P55	Output	Debug LCD control
P70/KR0/SCK21/SCL21	Output	Debug LCD control
P71/KR1/SI21/SDA21	Output	Debug LCD control
P72/KR2/SO21	Output	Debug LCD control
P73/KR3/SO01	Output	Debug LCD control

5. Description of Software

5.1 Operation Outline

This application note explains how to use the data flash library.

The sample program covered in this document displays the target write address, the write value, and the read value on the LCD. The sample program can be manipulated using three switches which are used to set the write value, set the target write address, and program into the data flash memory, respectively. Data in the data flash memory is erased by pressing the two switches for setting the write value and target write address at the same time for one second. The display on the LCD is updated every time the write value and the target write address are changed. The read value is updated after power is turned on, data is programmed, or it is erased.

(1) Setup up the TAU0 channel 0.

<Setting conditions>

- Uses the TAU0 channel 0.
- Uses the 500-kHz operation clock.
- Enables only the software start trigger as the start trigger.
- Sets the enable edge to the falling edge
- Sets the operation mode to interval timer mode.
- Sets the count start and interrupt settings to “Generates no timer interrupt at the beginning of counting.”
- Uses the timer interrupt (INTTM00).
- Sets the interrupt timing to 100 ms.

(2) Sets up the 12-bit interval timer.

< Setting conditions >

- Uses the 15-kHz operation clock.
- Sets the interrupt priority level to 2.
- Sets the interval time to 10 ms.

(3) Sets up the external interrupt input.

- Sets the interrupt priority level to 1.
- Sets the enable edge to the falling edge.

(4) Initializes the LCD.

(5) Starts the INTP.

- Enables edge detection interrupt processing of the INTP1, INTP2, and INTP4 pins.
- Enables interrupts of the INTP1, INTP2, and INTP4 pins.

(6) Initializes the data flash library.

- Initializes the RAM that the data flash library is to use.
- If an error is caused during initialization, the sample program displays "ERROR!" on the LCD and suppresses the execution of the subsequent operations.

(7) Reads the contents of the data flash memory.

- Reads data from the start address of block 0.

(8) Stops the data flash library and sets it up so that it is ready to switch into HALT mode.

(9) Clears the pressed status of the switch.

(10) Updates the data to be sent to the upper and lower columns of the LCD.

(11) Displays the target write address on the upper column of the LCD and the write and read values of on the lower column of the LCD.

(12) If none of the switches are pressed, switches into HALT mode and waits for a press of the switch.

(13) When a switch-triggered external interrupt occurs, exits the HALT mode and takes the following actions to avoid chatters:

- Starts the interval timer for counting on INTP1, INTP2, or INTP4 interrupt.
- Waits until an interval timer interrupt occurs.
- Has the interval timer interrupt handler test the switch status. More specifically, the interrupt handler checks the level of the input at P31, P50, and P51.
- If the level of P31, P50, or P51 is 0, sets the switch-pressed flag determining that a switch has been pressed.
- If the level of P31, P50, and P51 is 1, clears the switch-pressed flag and returns to step (9), determining that none of the switches have been pressed.

(14) Determines which switch has been pressed.

(15) Takes actions according to the pressed status of the switches.

- Increments the write value if only SW1 is pressed.
- Updates the target write address if only SW2 is pressed.
- If only SW3 is pressed, turns on LED0 to indicate that flash memory is being accessed and program the write value into the selected address.
  - After programming the write value, the sample program reads it and compares the read value with the write value. If they match, the sample program turns off LED0.
  - If the write value and read value do not match, the sample program displays "ERROR" on the LCD and suppresses the execution of the subsequent operations.
- If SW1 and SW2 are pressed at the same time for one second, turns on LED0 to indicate that flash memory is being accessed and initialize the data flash memory.
  - If the initialization fails, the sample program displays "ERROR" on the LCD and suppresses the execution of the subsequent operations.
  - After the initialization is completed, the sample program turns off LED0 and reads data from the selected address.

(16) Returns to step (9).

5.2 File Configuration

Table 5.1 lists the additional functions for files that are automatically generated in the integrated development environment and other additional files.

Table 5.1 List of Additional Functions and Files

File Name	Outline	Remarks
r_main.c	Main module	Additional functions: R_MAIN_INTCStart R_MAIN_ClearSwitchFlag R_MAIN_GetSwitchStatus R_MAIN_DetectLongPush R_MAIN_IncrementValue R_MAIN_SwitchProcess R_MAIN_UpdateStringUpper R_MAIN_UpdateStringDowner
r_pfdl.c	Data Flash Library execution	R_FDL_Init R_FDL_BlankCheck R_FDL_Erase R_FDL_Verify R_FDL_Read R_FDL_Write R_FDL_ChangeAddress R_FDL_ExecuteWrite R_FDL_ClearDataFlash

5.3 List of Option Byte Settings

Table 5.2 summarizes the settings of the option bytes.

Table 5.2 Option Byte Settings

Address	Setting	Description
000C0H/010C0H	11101111B	Disables the watchdog timer. (Stops counting after the release from the reset status.)
000C1H/010C1H	01111111B	LVD reset mode 2.81 V (2.76 V to 2.87 V)
000C2H/010C2H	11101000B	HS mode, HOCO: 32 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger Erases the data in the flash memory when on-chip debug security ID authentication fails.

The option bytes of the RL78/G13 comprise the user option bytes (000C0H to 000C2H) and on-chip debug option byte (000C3H).

The option bytes are automatically referenced and the specified settings are configured at power-on time or the reset is released.

The option bytes must be specified from "User Option Byte Values" on the "Device" panel in the "Link Options" of CS+. Set "Set up user option bytes" to "Yes (-gb)."

5.4 Link Option

The –start option, which is one of the link options, is provided for allocating the Data Flash Library Type04 to a ROM area.

Use the –start option to specify all sections for which setting are required by the Data Flash Library Type04.

Caution: For details on the link option procedures, refer to RL78 Compiler CC-RL User's Manual (R20UT3123E).

5.5 List of Constants

Table 5.3 lists the constants for the sample program.

Table 5.3 Constants for the Sample Program

Constant	Setting	Description
SW_ON	1	Confirmation that a switch is pressed
SW_OFF	0	Switch-pressed status cleared.
ON_SW_1	0x01	Switch 1 pressed.
ON_SW_2	0x02	Switch 2 pressed.
ON_SW_3	0x04	Switch 3 pressed.
OFF_SW_ALL	0x00	All switch-pressed statuses cleared.
FLASH_START_ADDRESS	0xF1000	Start address of data flash memory
TARGET_BLOCK	0	Target write block Note
BLOCK_SIZE	0x400	Size of one block (bytes)
WRITE_SIZE	1	Size of write data (bytes)
MAX_VALUE	0xFF	Maximum write value of data flash memory
MAX_ADDRESS	(TARGET_BLOCK+1) * BLOCK_SIZE -1	Maximum write address of data flash memory
PFDL_NG	1	Abnormal termination of data flash library
FDL_FRQ	32	Frequency setting [MHz]
FDL_VOL	0x00	Voltage mode (full-speed mode)
LCD_SIZE	8	Maximum number of characters to be dieplayed on LCD

Note: The valid values of TARGET_BLOCK are 0 to 3. An error will be caused at build if any other value is specified. The relationships between the TARGET_BLOCK values and the target write blocks are given below.
0: The write target block is block 0 of the data flash memory (addresses 0xF1000 to 0xF13FF).
1: The write target block is block 1 of the data flash memory (addresses 0xF1400 to 0xF17FF).
2: The write target block is block 2 of the data flash memory (addresses 0xF1800 to 0xF1BFF).
3: The write target block is block 3 of the data flash memory (addresses 0xF1C00 to 0xF1FFF).

5.6 List of Variables

Table 5.4 lists the global variables that are used in this sample program.

Table 5.4 Global Variables for the Sample Program

Type	Variable Name	Contents	Function Used
uint8_t	g_sw_push	Confirmation that a switch is pressed	r_main r_it_interrupt R_MAIN_ClearSwitchFlag
uint8_t	g_it_flag	Interval timer interrupt flag	r_main r_it_interrupt
uint8_t	g_read_value	Read value	R_FDL_Read R_MAIN_UpdateStringDowner
uint8_t	g_write_value	Write value	R_FDL_Read R_FDL_Write R_MAIN_Increment R_MAIN_UpdateStringDowner
uint16_t	g_write_address	Target write address	R_FDL_BlankCheck R_FDL_Read R_FDL_Write R_FDL_ChangeAddress R_MAIN_UpdateStringUpper

5.7 List of Functions

Table 5.5 lists the functions that are used in this sample program.

Table 5.5 List of Functions

Function Name	Outline
R_MAIN_INTCStart	Starts INTP.
R_INTC1_Start	Starts INTP1.
r_intc1_interrupt	INTP1 external interrupt
R_INTC2_Start	Starts INTP2.
r_intc2_interrupt	INTP2 external interrupt
R_INTC4_Start	Starts INTP4.
r_intc4_interrupt	INTP4 external interrupt
R_IT_Start	Starts interval timer.
r_it_interrupt	Interval timer interrupt
R_IT_Stop	Stops interval timer.
R_FDL_Init	Starts data flash library.
R_FDL_Read	Processes data read command.
R_MAIN_ClearSwitchFlag	Clears switch-pressed status.
R_MAIN_UpdateStringUpper	Updates string displayed on upper column of LCD.
R_MAIN_UpdateStringDowner	Updates string displayed on lower column of LCD.
R_MAIN_GetSwitchStatus	Gets switch status.
R_MAIN_SwitchProcess	Processes switch-pressed status.
R_MAIN_IncrementValue	Increments write value.
R_FDL_ChangeAddress	Changes target write address.
R_FDL_ExecuteWrite	Executes writing.
R_FDL_BlankCheck	Processes blank check command.
R_FDL_Erase	Processes block erase command.
R_FDL_Write	Processes data write command.
R_FDL_Verify	Processes verify command.
R_MAIN_DetectLongPush	Detects long press.
R_MAIN_INTCStop	Stops INTP.
R_INTC1_Stop	Stops INTP1.
R_INTC2_Stop	Stops INTP2.
R_INTC4_Stop	Stops INTP4.
R_TAU0_Channel0_Start	Starts TAU0 channel 0.
R_TAU0_Channel0_Stop	Stops TAU0 channel 0.
R_FDL_ClearDataFlash	Initializes data flash memory.

5.8 Function Specifications

This section describes the specifications for the functions that are used in the sample program.

[Function Name] R_ MAIN_INTCStart

Synopsis	Start INTP.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h
Declaration	void R_MAIN_INTCStart(void)
Explanation	This function starts the INTP.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC1_Start

Synopsis	Start INTP1.
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
Declaration	void R_INTC1_Start(void)
Explanation	This function starts the INTP1.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intc1_interrupt

Synopsis	INTP1 external interrupt
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_it.h r_cg_userdefine.h
Declaration	static void __near r_intc1_interrupt(void)
Explanation	This function starts the interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC2_Start

Synopsis	Start INTP.
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
Declaration	void R_INTC2_Start(void)
Explanation	This function starts the INTP2.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intc2_interrupt

Synopsis	INTP2 external interrupt
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_it.h r_cg_userdefine.h
Declaration	static void __near r_intc2_interrupt(void)
Explanation	This function starts the interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC4_Start

Synopsis	Start INTP4.
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
Declaration	void R_INTC4_Start(void)
Explanation	This function starts the INTP4.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intc4_interrupt

Synopsis	INTP4 external interrupt
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_it.h r_cg_userdefine.h
Declaration	static void __near r_intc4_interrupt(void)
Explanation	This function starts the interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IT_Start

Synopsis	Starts interval timer.
Header	r_cg_macrodriver.h r_cg_it.h r_cg_userdefine.h
Declaration	void R_IT_Start(void)
Explanation	This function starts the interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_it_interrupt

Synopsis	Interval timer interrupt
Header	r_cg_macrodriver.h r_cg_it.h r_cg_userdefine.h
Declaration	static void __near r_it_interrupt(void)
Explanation	This function stops the interval timer. If any of switches SW1 to SW3 is pressed, the function sets the switch-pressed status flag (g_sw_push) to 1.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IT_Stop

Synopsis	Stop interval timer.
Header	r_cg_macrodriver.h r_cg_it.h r_cg_userdefine.h
Declaration	void R_IT_Stop(void)
Explanation	This function stops the interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_FDL_Init

Synopsis	Start data flash library.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	void R_FDL_Init(void)
Explanation	This function initializes and starts the RAM to be used by the Data Flash Library Type04.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_FDL_Read

Synopsis	Process data read command.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	void R_FDL_Read(void)
Explanation	This function executes the data read command and stores the read data in the read data storage variable g_read_value.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_MAIN_ClearSwitchFlag

Synopsis	Clear switch-pressed status.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h
Declaration	void R_MAIN_ClearSwitchFlag(void)
Explanation	This function clears the switch-pressed status flag g_sw_push.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_MAIN_UpdateStringUpper

Synopsis	Update string displayed on upper column of LCD.	
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h	
Declaration	void R_MAIN_UpdateStringUpper(int8_t upper_string[LCD_SIZE + 1])	
Explanation	This function updates the string displayed on the upper column of the LCD to the value of g_write_address.	
Arguments	upper_string[LCD_SIZE + 1]	String displayed on upper column of LCD
Return value	None	
Remarks	None	

[Function Name] R_MAIN_UpdateStringDowner

Synopsis	Update string displayed on lower column of LCD.	
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h	
Declaration	void R_MAIN_UpdateStringDowner(int8_t downer_string[LCD_SIZE + 1])	
Explanation	This function updates the string displayed on the lower column of the LCD to the values of g_write_address and g_read_value.	
Arguments	downer_string[LCD_SIZE + 1]	String to be displayed on lower column of LCD
Return value	None	
Remarks	None	

[Function Name] R_MAIN_GetSwitchStatus

Synopsis	Get switch status.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h
Declaration	uint8_t R_MAIN_GetSwitchStatus(void)
Explanation	This function gets the status of SW1, SW2, and SW3.
Arguments	None
Return value	Switch-pressed status: sw_status (Initial value = 0) • No SW is pressed: sw_status • SW1 is pressed: sw_status + ON_SW_1 • SW2 is pressed: sw_status + ON_SW_2 • SW3 is pressed: sw_status + ON_SW_3
Remarks	None

[Function Name] R_MAIN_SwitchProcess

Synopsis	Process switch-pressed status.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h
Declaration	uint8_t R_MAIN_SwitchProcess(uint8_t sw_status)
Explanation	This function causes a branch according to the switch-pressed status.
Arguments	sw_status Switch-pressed status of SW1, SW2, and SW3
Return value	• Normal termination: PFDL_OK • Abnormal termination: PFDL_NG
Remarks	None

[Function Name] R_MAIN_IncrementValue

Synopsis	Increment write value.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h
Declaration	void R_MAIN_IncrementValue(void)
Explanation	This function increments the value to be written to the data flash memory, g_write_value.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_FDL_ChangeAddress

Synopsis	Change target write address.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	void R_FDL_ChangeAddress(void)
Explanation	This function changes the target write address and reads the data from the new target address into the read data storage variable g_read_value.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_FDL_ExecuteWrite

Synopsis	Execute writing.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	uint8_t R_FDL_ExecuteWrite(void)
Explanation	This function writes the write value to the target write address in the data flash memory.
Arguments	None
Return value	• Normal termination: PFDL_OK • Abnormal termination: PFDL_NG
Remarks	None

[Function Name] R_FDL_BlankCheck

Synopsis	Process blank check command.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	uint8_t R_FDL_BlankCheck(void)
Explanation	This function checks the target address to determine if it is blank.
Arguments	None
Return value	• Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Blank check error: PFDL_ERR_MARGIN
Remarks	None

[Function Name] R_FDL_Erase

Synopsis	Process block erase command.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	uint8_t R_FDL_Erase(void)
Explanation	This function erases the entire block.
Arguments	None
Return value	• Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Erase error: PFDL_ERR_ERASE
Remarks	None

[Function Name] R_FDL_Write

Synopsis	Process data write command.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	uint8_t R_FDL_Write(void)
Explanation	This function writes data into the data flash memory.
Arguments	None
Return value	• Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Write error: PFDL_ERR_WRITE
Remarks	None

[Function Name] R_FDL_Verify

Synopsis	Process verify command.
Header	r_cg_macrodriver.h pfdl.h pfdl_types.h r_cg_userdefine.h
Declaration	uint8_t R_FDL_Verify(void)
Explanation	This function checks whether the written data is correct.
Arguments	None
Return value	• Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Internal verify error: PFDL_ERR_MARGIN
Remarks	None

[Function Name] R_MAIN_DetectLongPush

Synopsis	Detects long press.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd_h rskrl78g13def.h pfdl.h pfdl_types.h string.h r_cg_userdefine.h
Declaration	uint8_t R_MAIN_DetectLongPush(void)
Explanation	This function checks whether a switch is pressed long.
Arguments	None
Return value	• Long press is detected: SW_ON • Long press is not detected: SW_OFF
Remarks	None

[Function Name] R_ MAIN_INTCStop

Synopsis	Stop INTP.
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_intc.h r_cg_timer.h r_cg_it.h lcd.h string.h r_cg_userdefine.h
Declaration	void R_MAIN_INTCStop(void)
Explanation	This function stops the INTP1, INTP2, and INTP4.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC1_Stop

Synopsis	Stop INTP1.
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
Declaration	void R_INTC1_Stop(void)
Explanation	This function stops the INTP1.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC2_Stop

Synopsis	Stop INTP2.
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
Declaration	void R_INTC2_Stop(void)
Explanation	This function stops the INTP2.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC4_Stop

Synopsis	Stop INTP4.
Header	r_cg_macrodriver.h r_cg_intc.h r_cg_userdefine.h
Declaration	void R_INTC4_Stop(void)
Explanation	This function stops the INTP4.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_TAU0_Channel0_Start

Synopsis	Start TAU0 channel 0.
Header	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h
Declaration	void R_TAU0_Channel0_Start(void)
Explanation	This function starts channel 0 of the timer array unit 0.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_TAU0_Channel0_Stop

Synopsis	Stop TAU0 channel 0.
Header	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h
Declaration	void R_TAU0_Channel0_Stop(void)
Explanation	This function stops channel 0 of the timer array unit 0.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_FDL_ClearDataFlash

Synopsis	Initialize data flash memory.
Header	r_cg_macrodriver.h r_cg_userdefine.h pfdl.h pfdl_types.h
Declaration	uint8_t R_FDL_ClearDataFlash(void)
Explanation	This function initializes the data flash memory.
Arguments	None
Return value	• Normal termination: PFDL_OK • Abnormal termination: PFDL_NG
Remarks	None

5.9 Flowcharts

Figure 5.1 shows the overall flow of the sample program described in this application note.

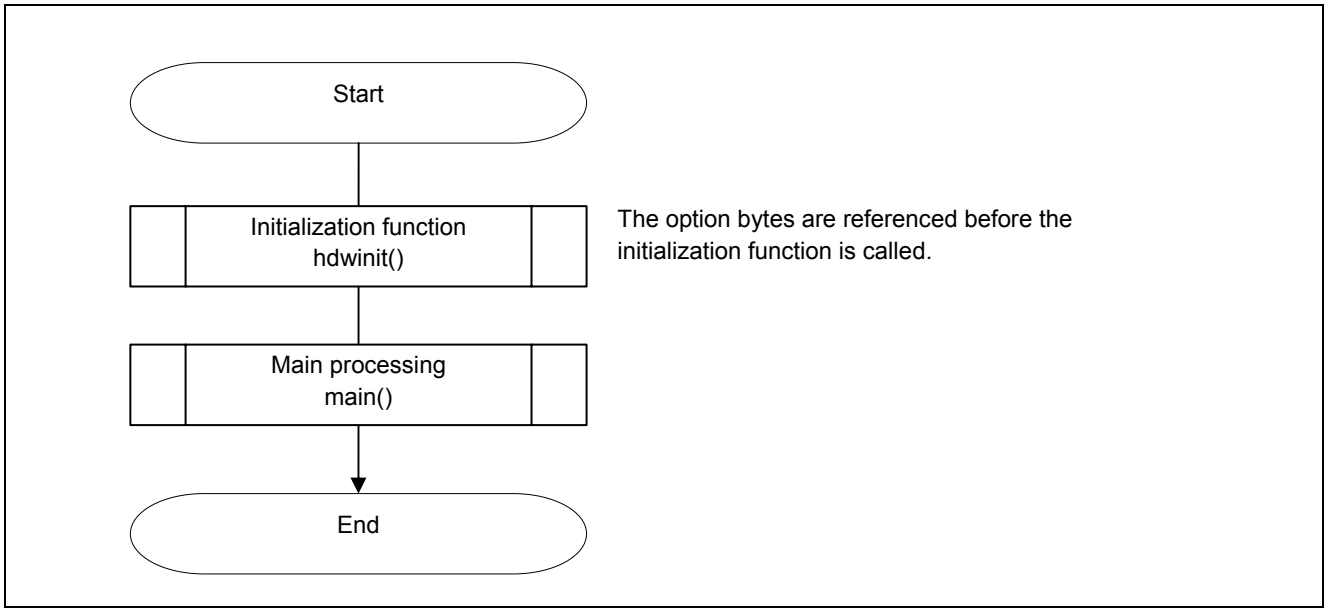

Figure 5.1 Overall Flow

5.9.1 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.

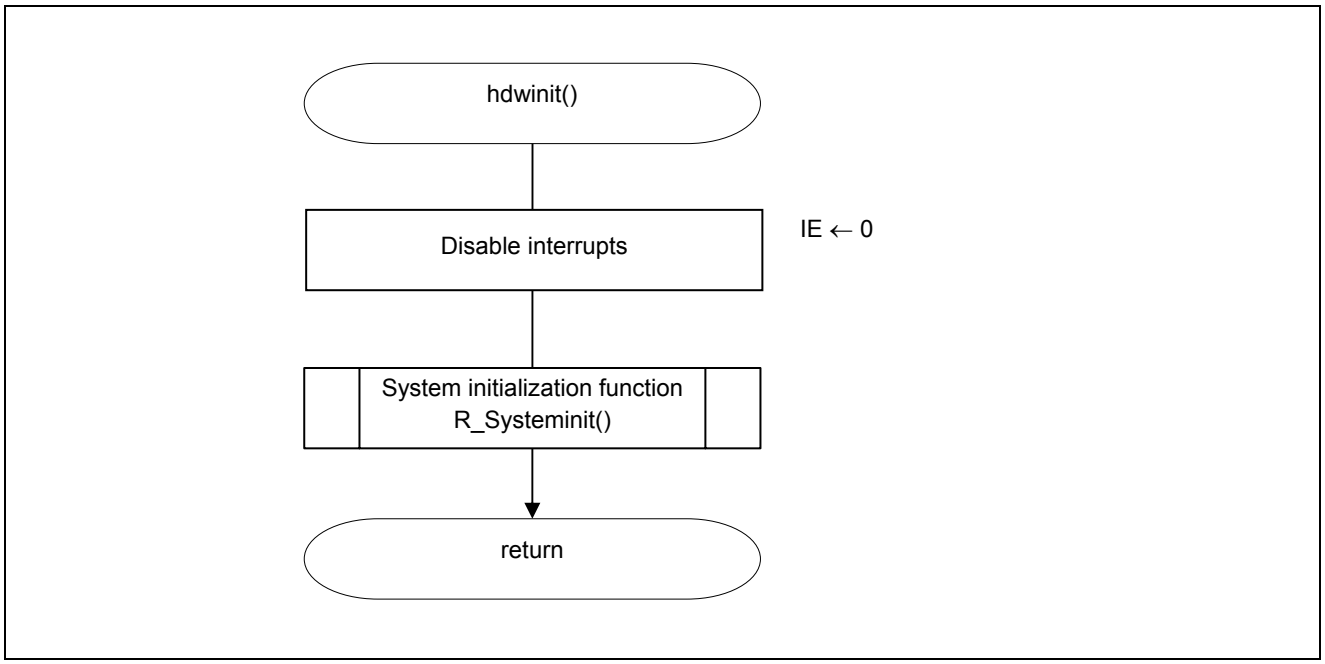

Figure 5.2 Initialization Function

5.9.2 System Initialization Function

Figure 5.3 shows the flowchart for the system initialization function.

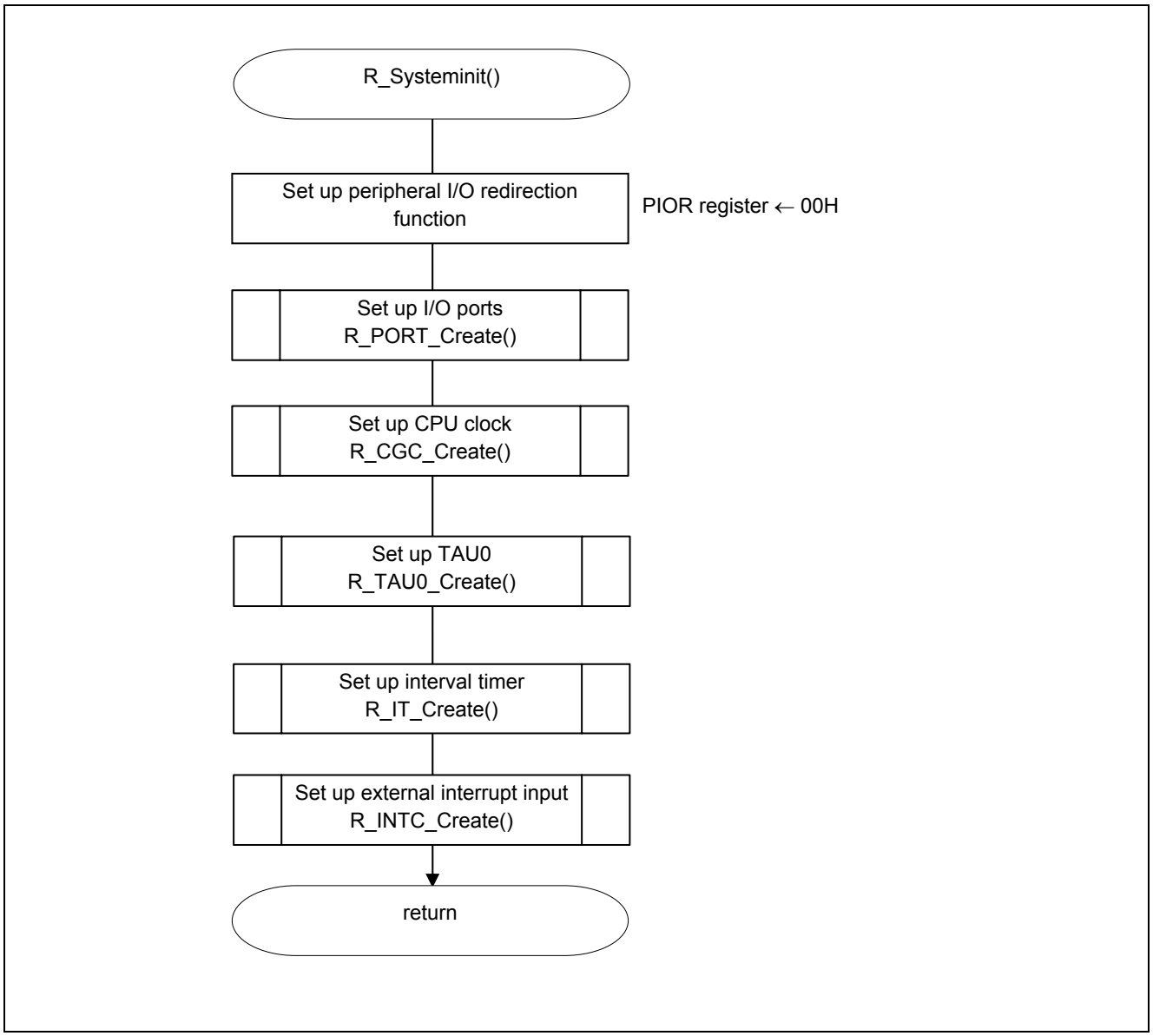

Figure 5.3 System Initialization Function

5.9.3 I/O Port Setup

Figure 5.4 shows the flowchart for I/O port setup.

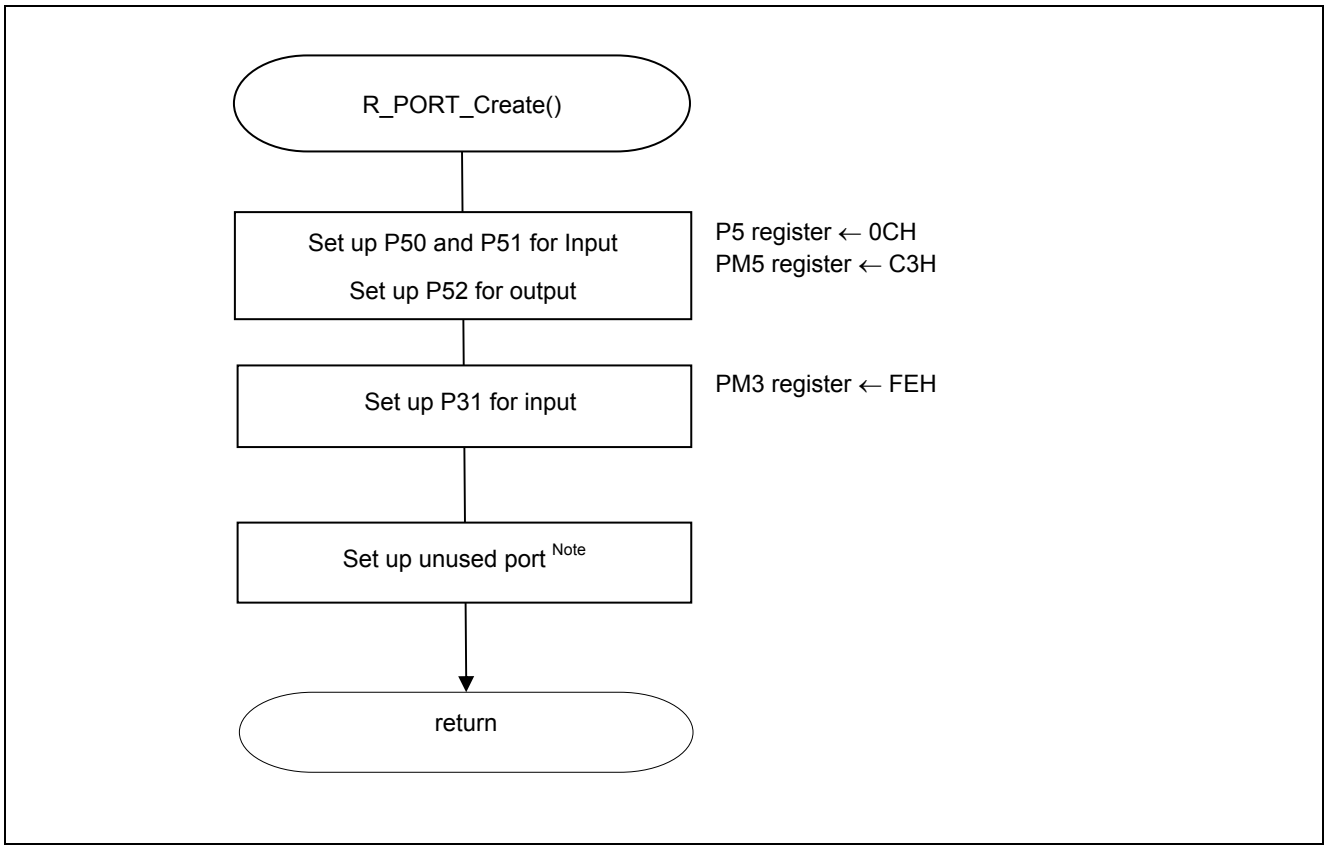

Figure 5.4 I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G13 Initialization (R01AN2575E) Application Note for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to V_{DD} or V_{SS} via a separate resistor.

5.9.4 CPU Clock Setup

Figure 5.5 shows the flowchart for CPU clock setup.

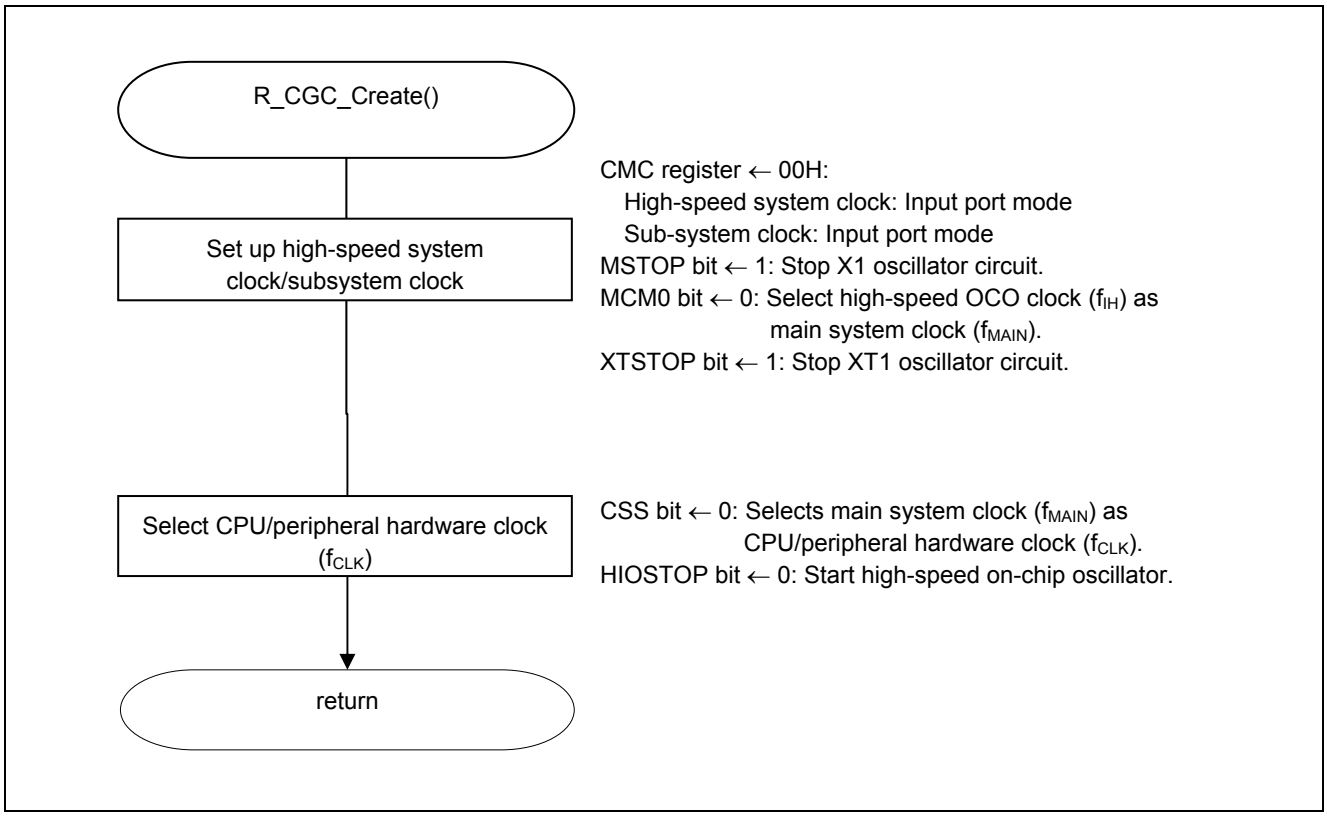

Figure 5.5 CPU Clock Setup

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization (R01AN2575E) Application Note.

5.9.5 TAU0 Setup

Figure 5.6 shows the flowchart for TAU0 setup.

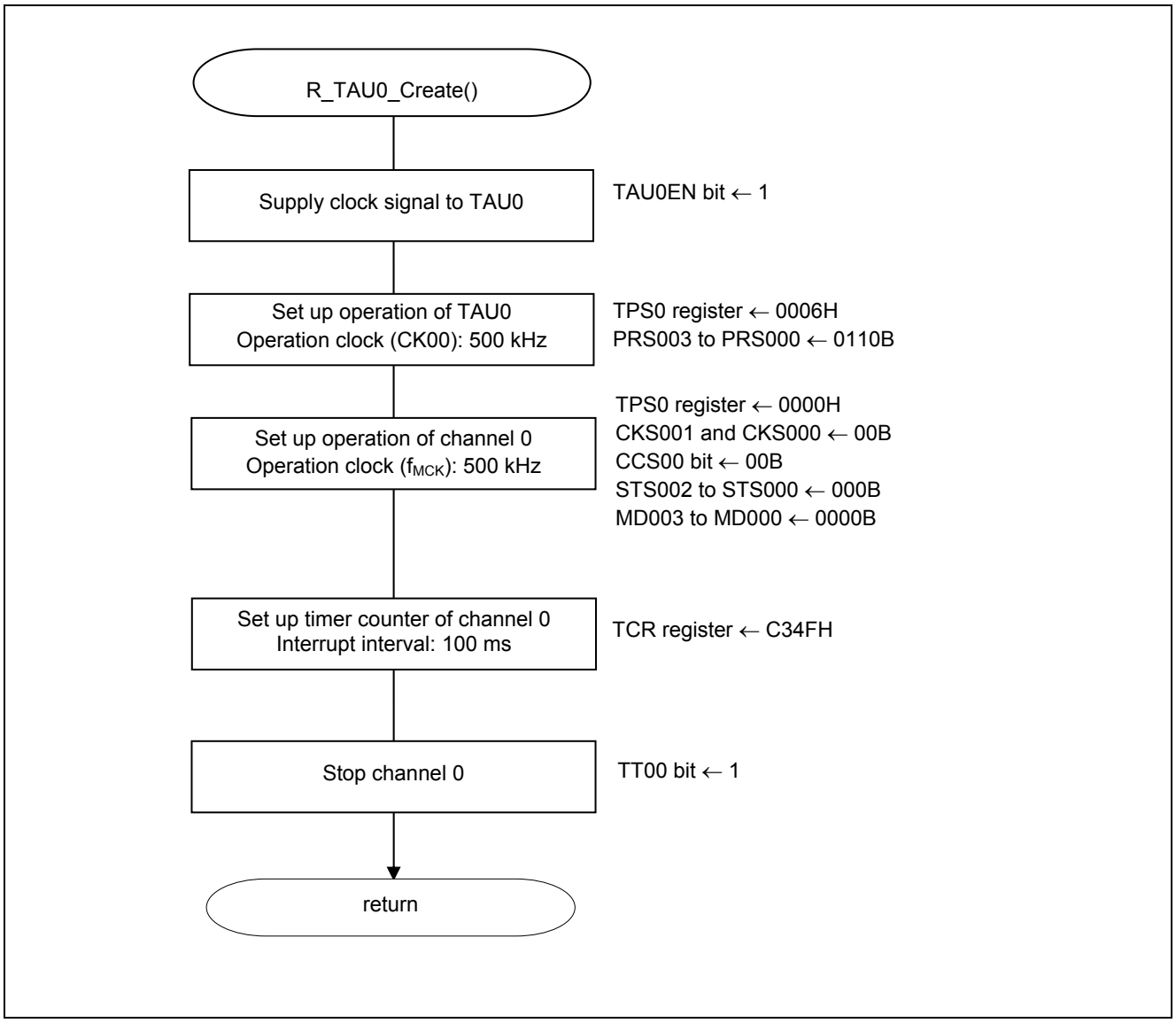

Figure 5.6 TAU0 Setup

5.9.6 Interval Timer Setup

Figure 5.7 shows the flowchart for interval timer setup.

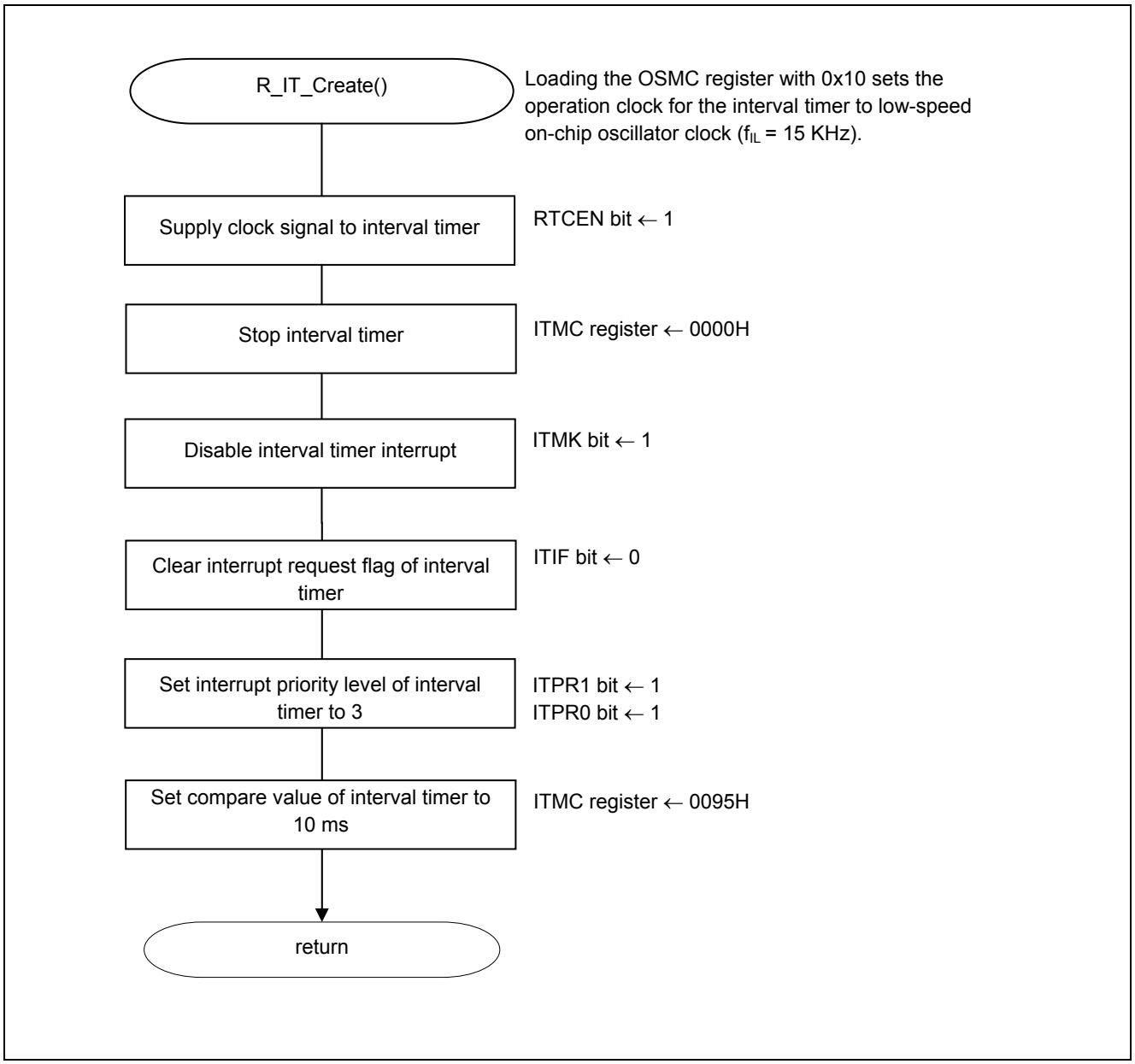

Figure 5.7 Interval Timer Setup

5.9.7 External Interrupt Input Setup

Figure 5.8 shows the flowchart for external interrupt input setup.

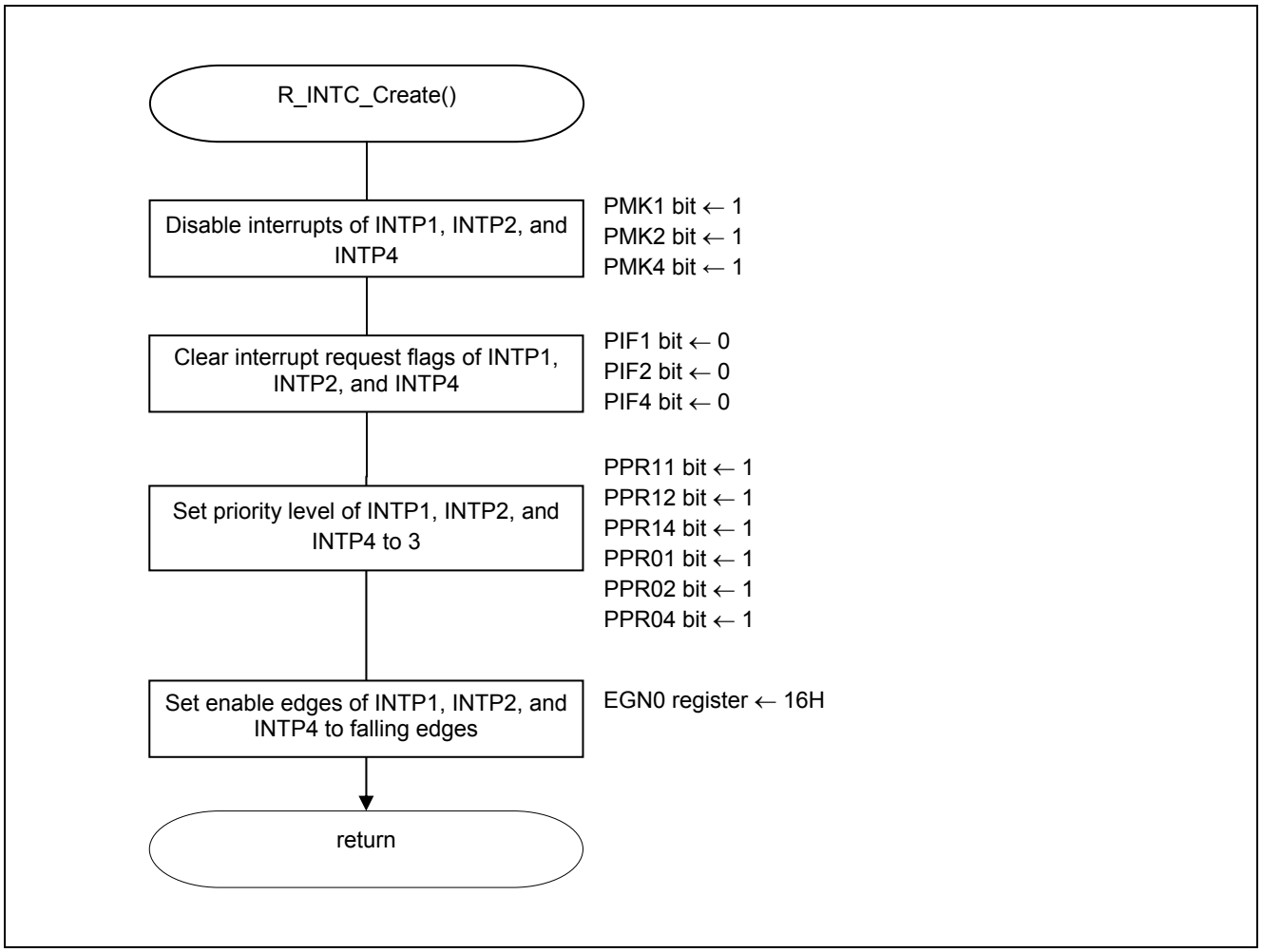

Figure 5.8 External Interrupt Input Setup

5.9.8 Main Processing

Figures 5.9 and 5.10 show the flowcharts for main processing.

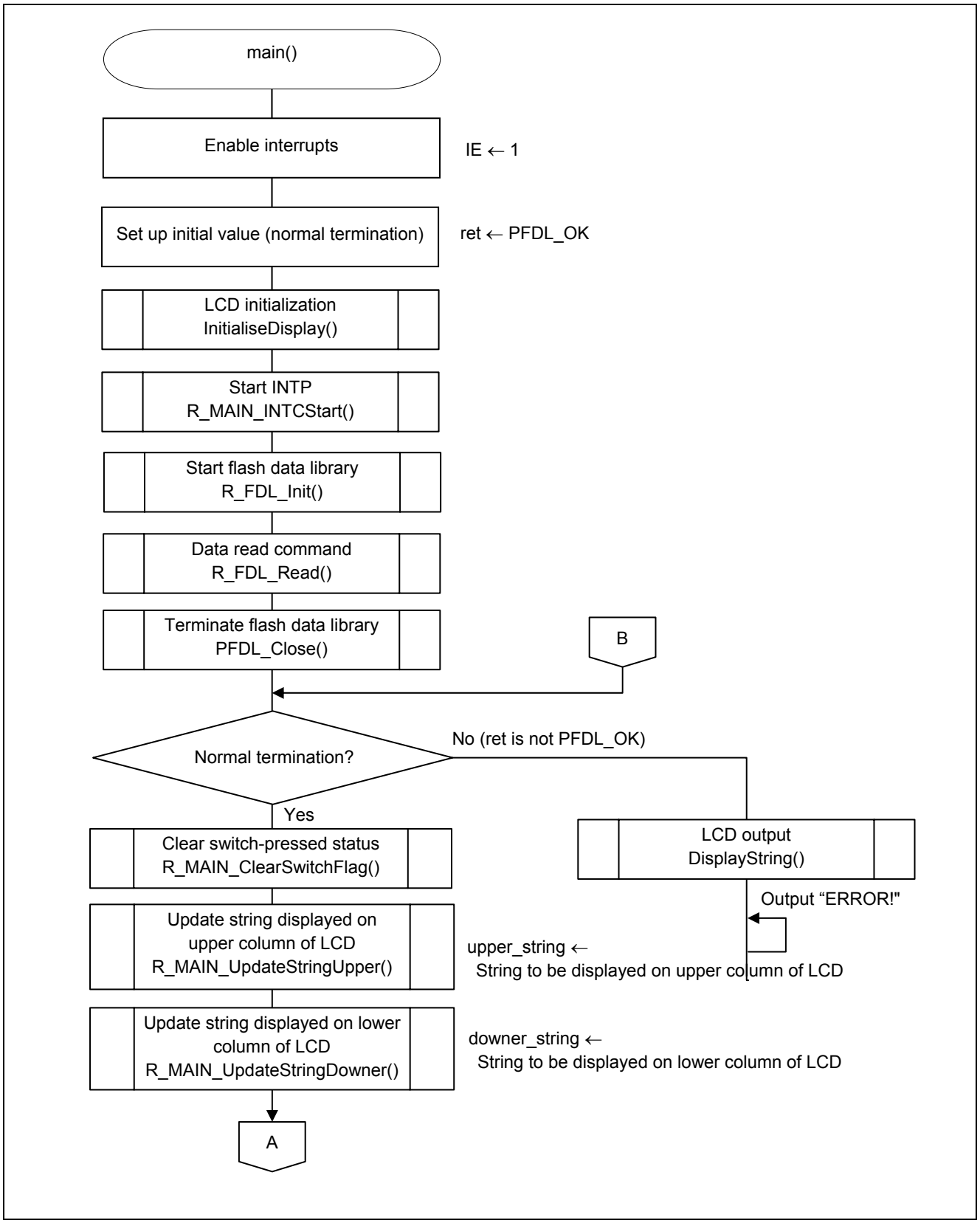

Figure 5.9 Main Processing (1/2)

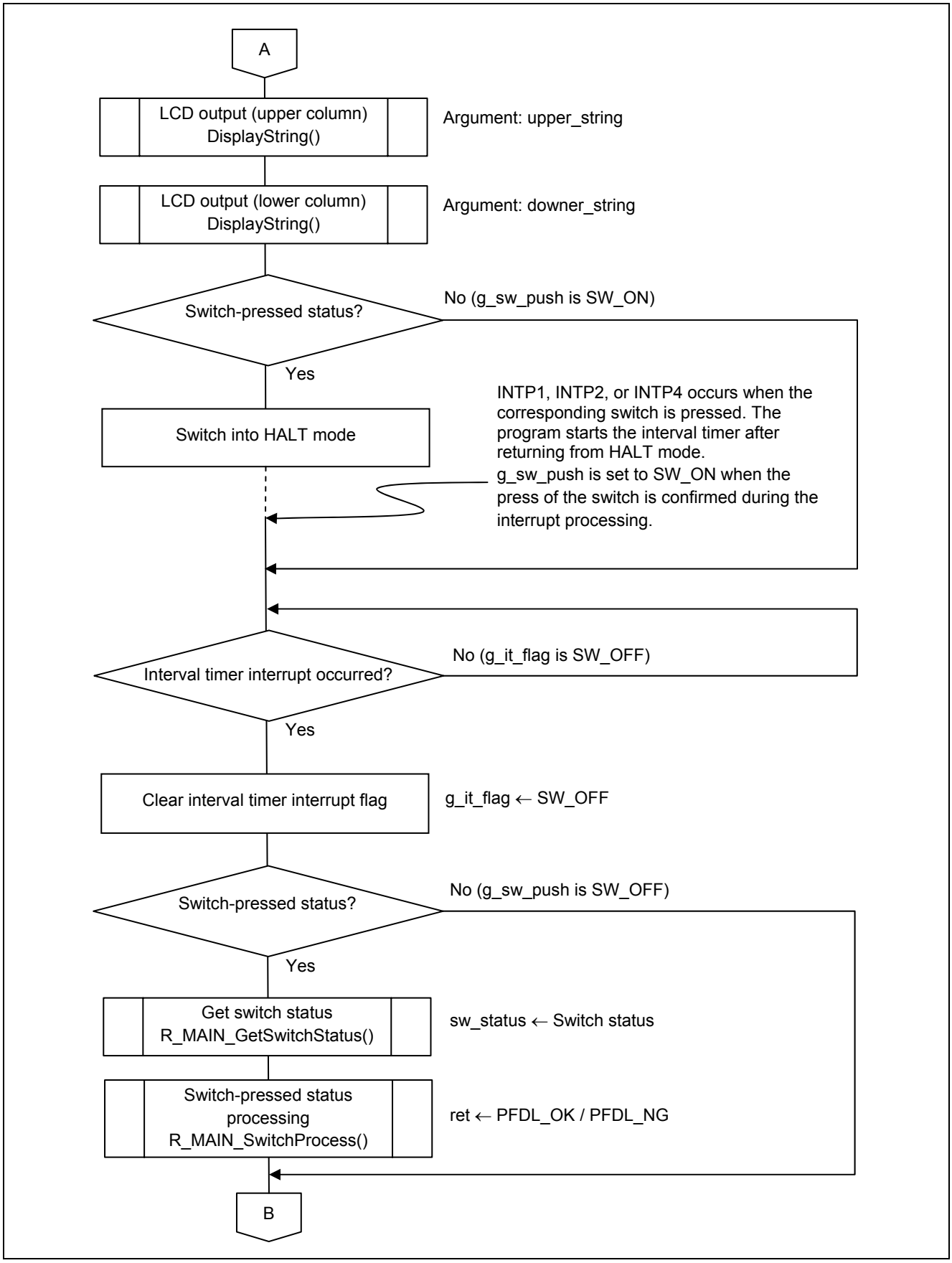

Figure 5.10 Main Processing (2/2)

5.9.9 Starting the INTP

Figure 5.11 shows the flowchart for starting the INTP.

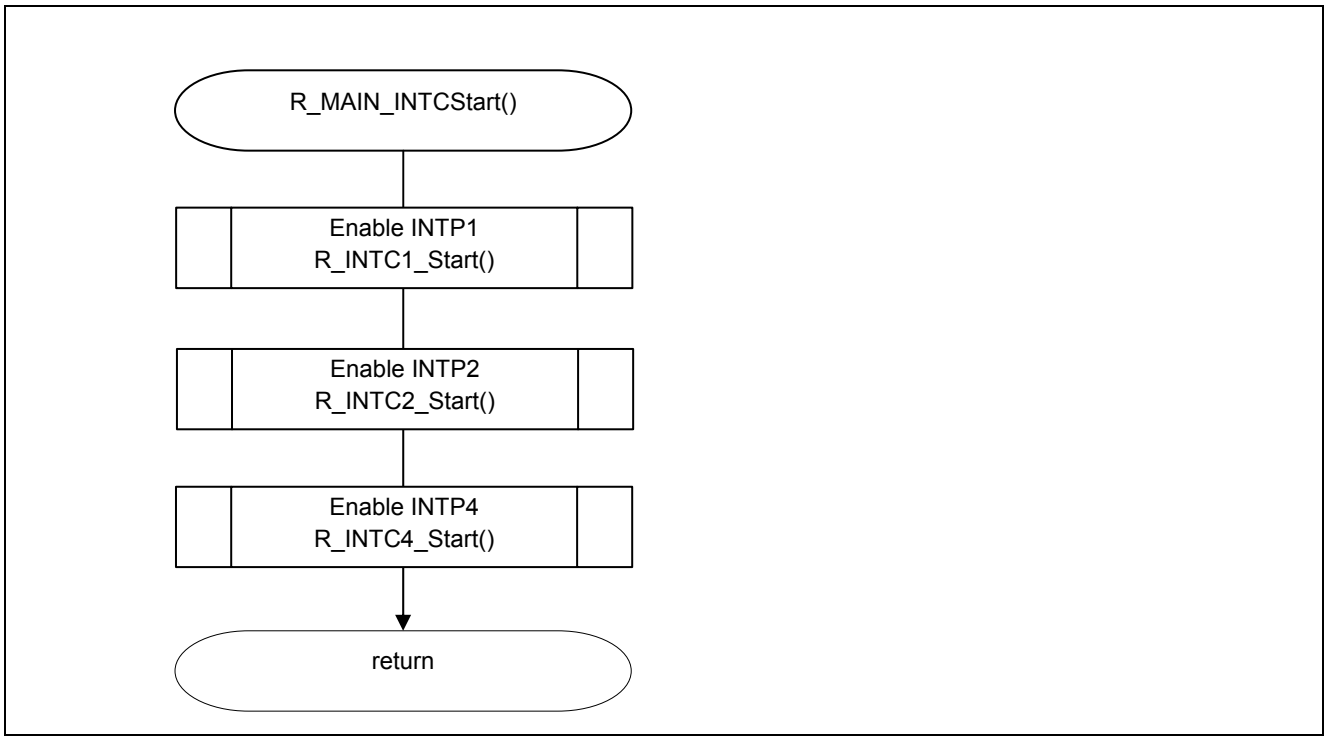

Figure 5.11 Starting the INTP

5.9.10 Starting the INTP1

Figure 5.12 shows the flowchart for starting the INTP1.

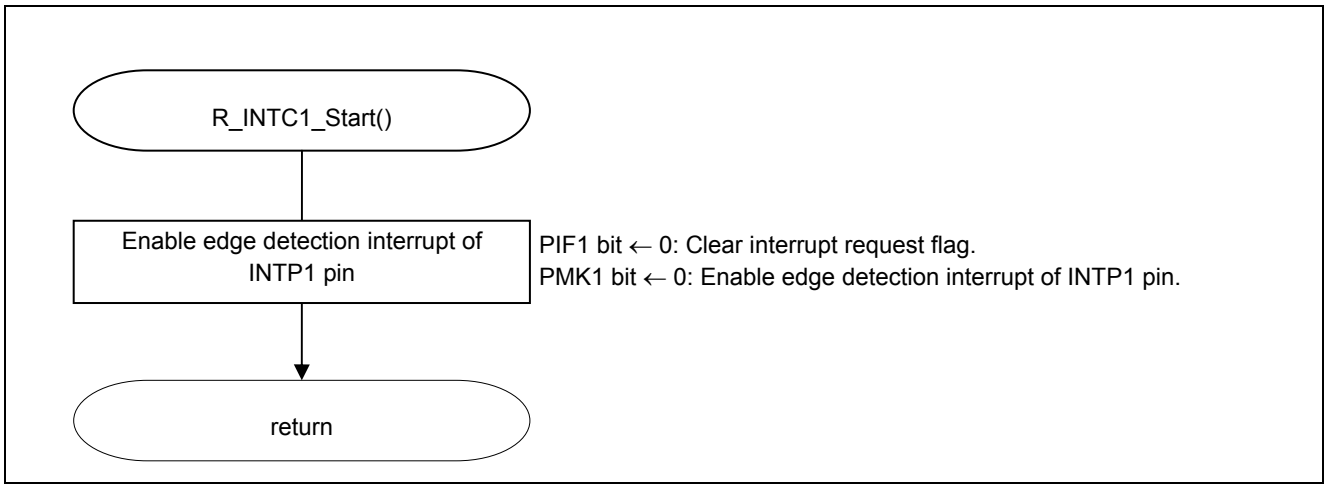

Figure 5.12 Starting the INTP1

5.9.11 INTP1 External Interrupt

Figure 5.13 shows the flowchart for INTP1 external interrupt.

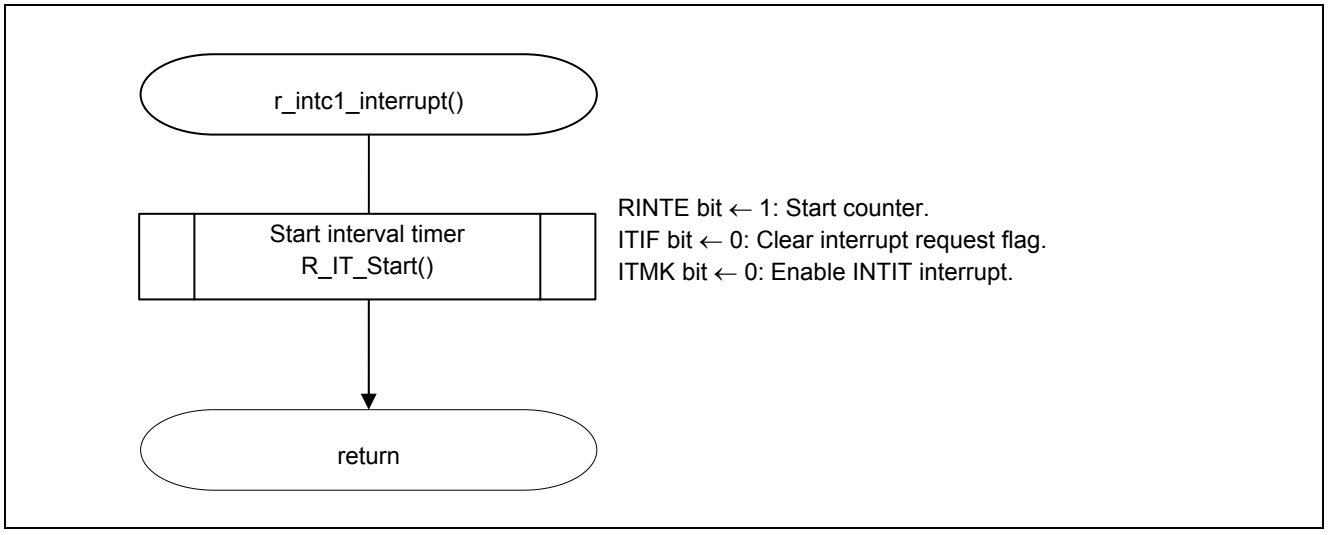

Figure 5.13 INTP1 External Interrupt

5.9.12 Starting the INTP2

Figure 5.14 shows the flowchart for starting the INTP2.

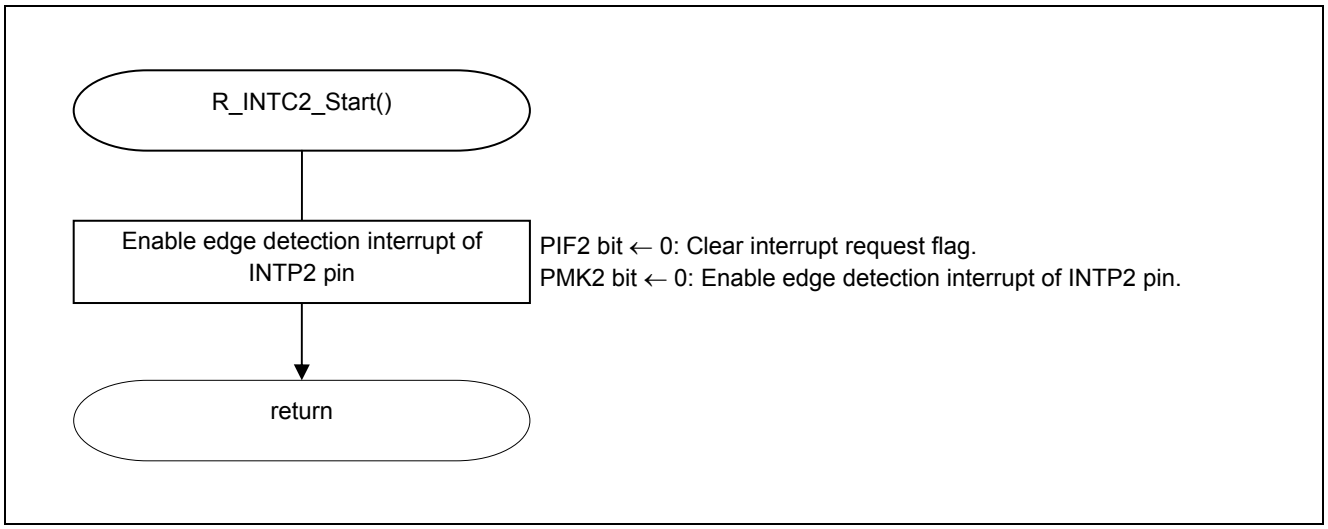

Figure 5.14 Starting the INTP2

5.9.13 INTP2 External Interrupt

Figure 5.15 shows the flowchart for INTP2 external interrupt.

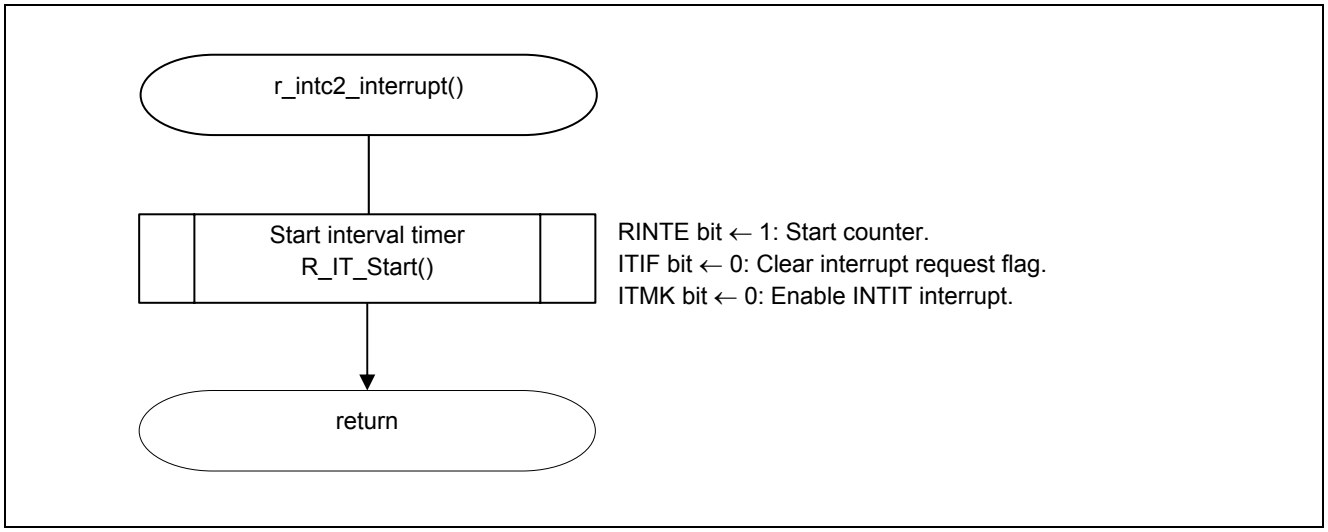

Figure 5.15 INTP2 External Interrupt

5.9.14 Starting the INTP4

Figure 5.16 shows the flowchart for starting the INTP4.

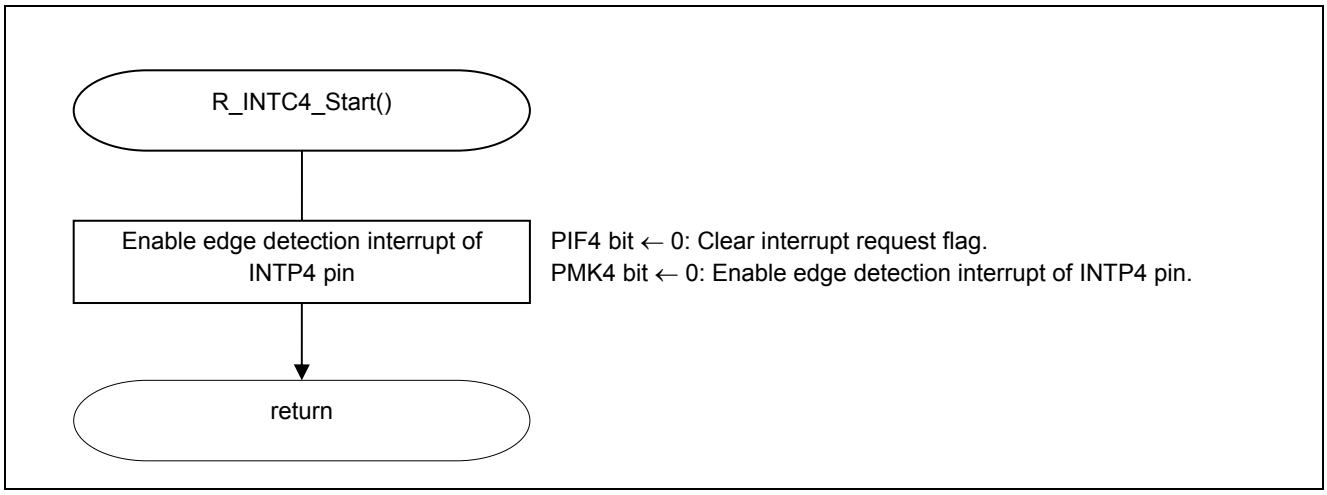

Figure 5.16 Starting the INTP4

5.9.15 INTP4 External Interrupt

Figure 5.17 shows the flowchart for INTP4 external interrupt.

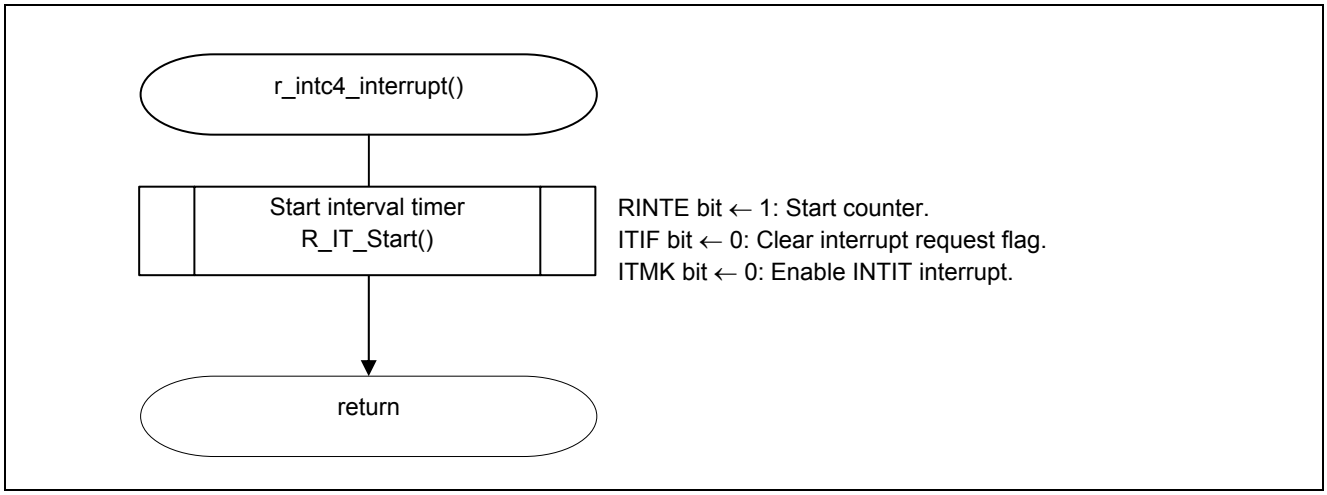

Figure 5.17 INTP4 External Interrupt

5.9.16 Starting the Interval Timer

Figure 5.18 shows the flowchart for starting the interval timer.

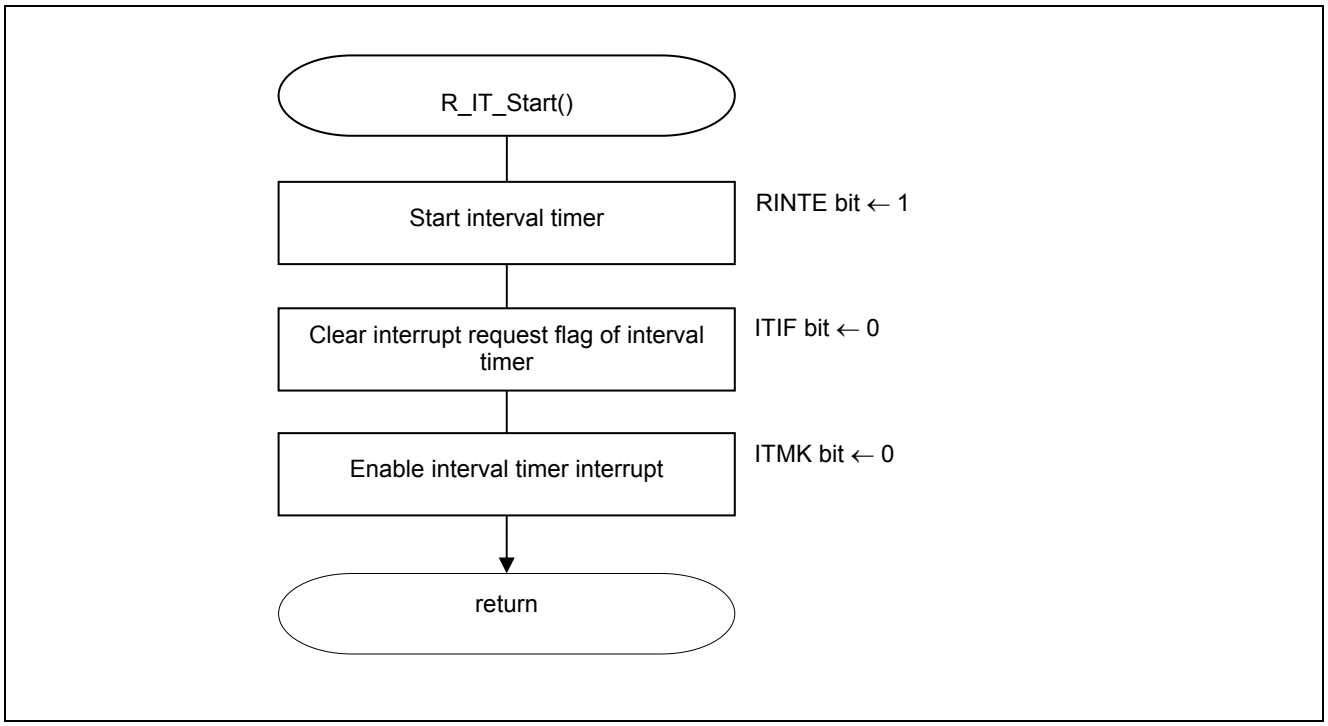

Figure 5.18 Starting the Interval Timer

5.9.17 Interval Timer Interrupt

Figure 5.19 shows the flowchart for interval timer interrupt.

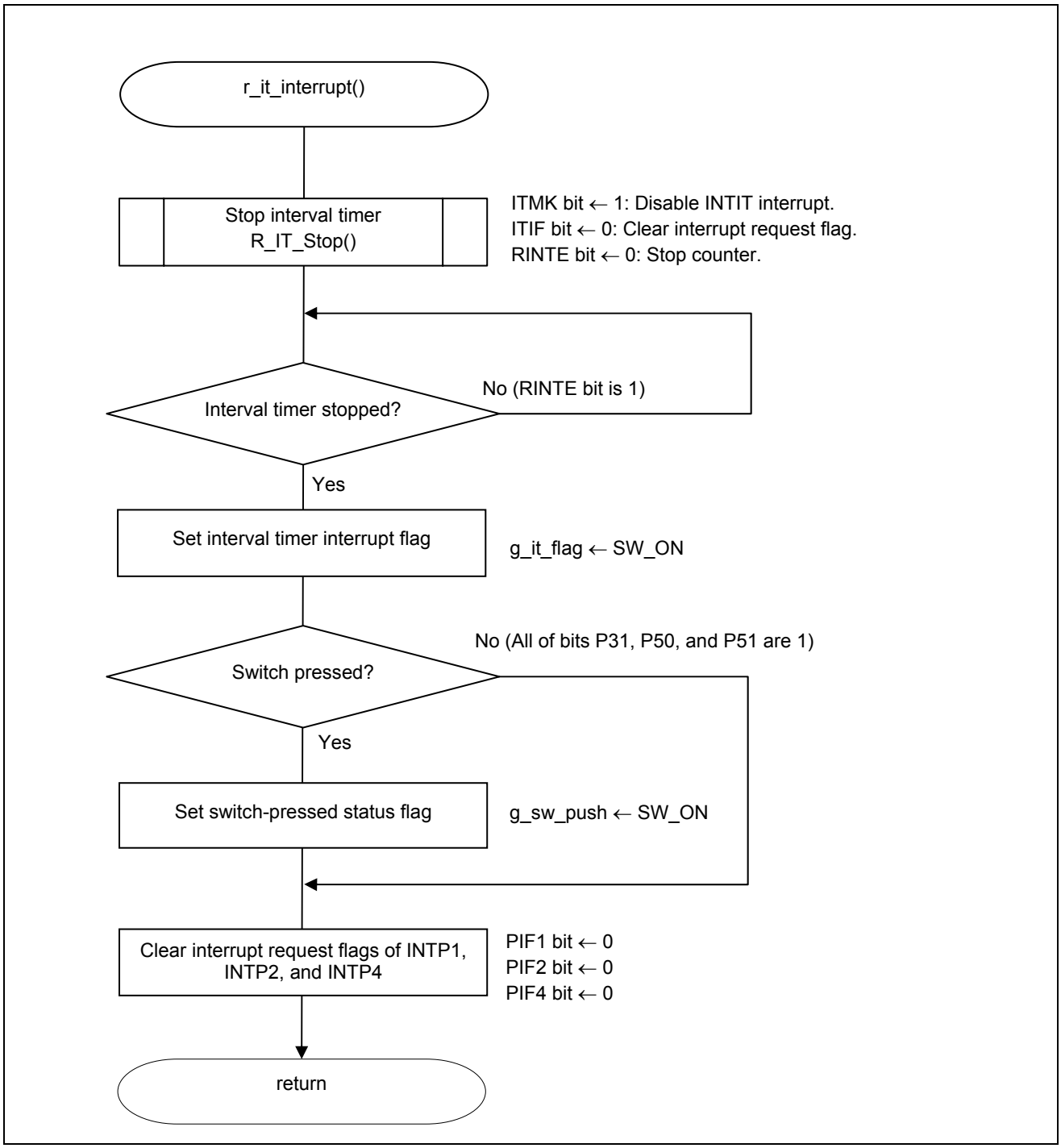

Figure 5.19 Interval Timer Interrupt

5.9.18 Stopping the Interval Timer

Figure 5.20 shows the flowchart for stopping the interval timer.

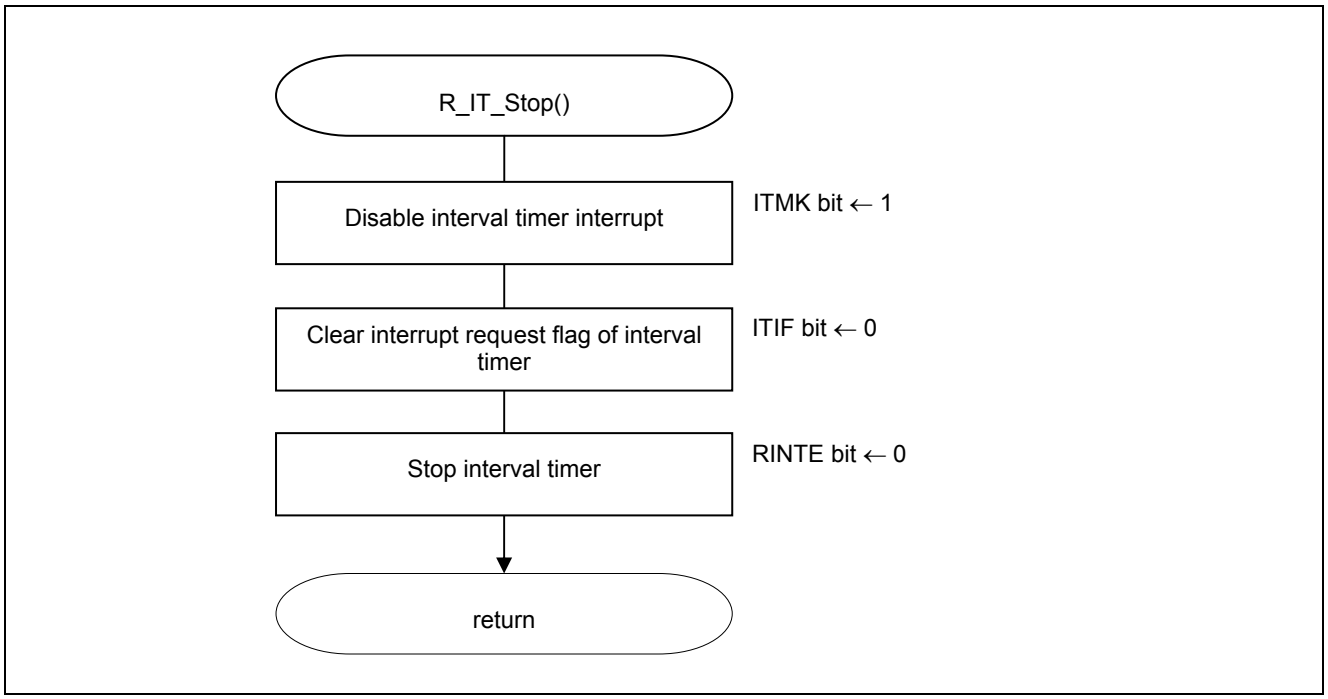

Figure 5.20 Stopping the Interval Timer

5.9.19 Starting the Data Flash Library

Figure 5.21 shows the flowchart for starting the data flash library.

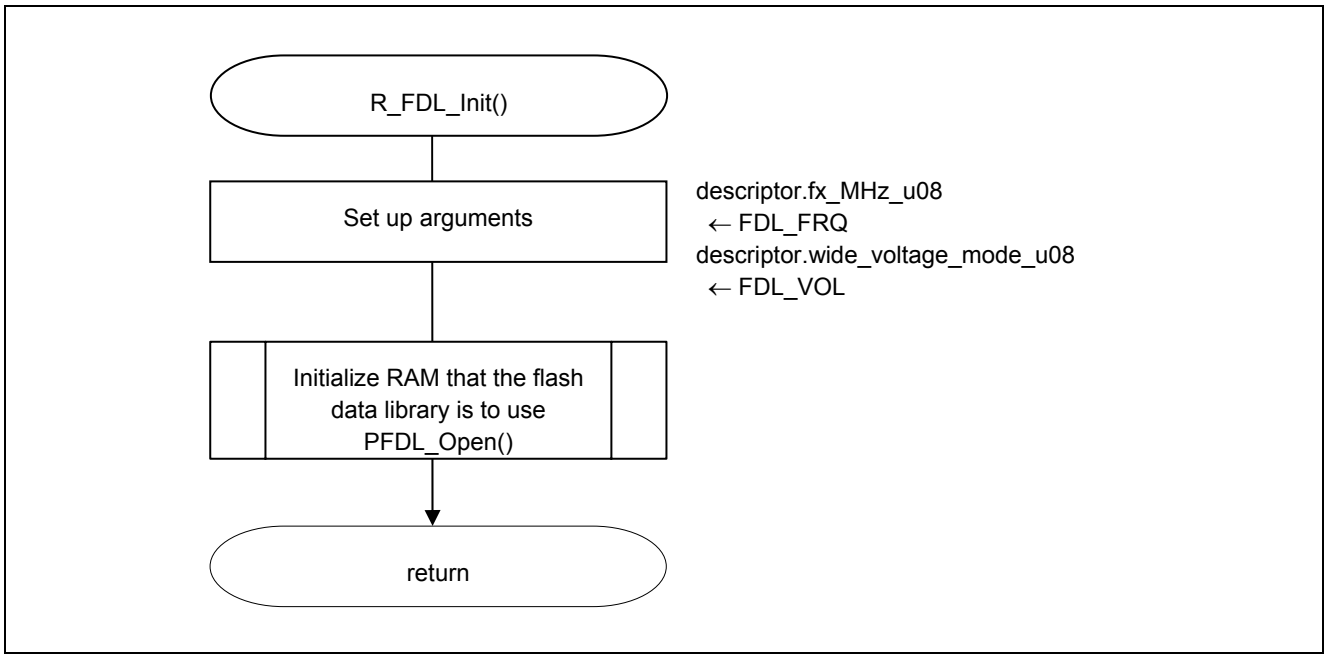

Figure 5.21 Starting the Data Flash Library

5.9.20 Processing the Data Read Command

Figure 5.22 shows the flowchart for processing the data read command.

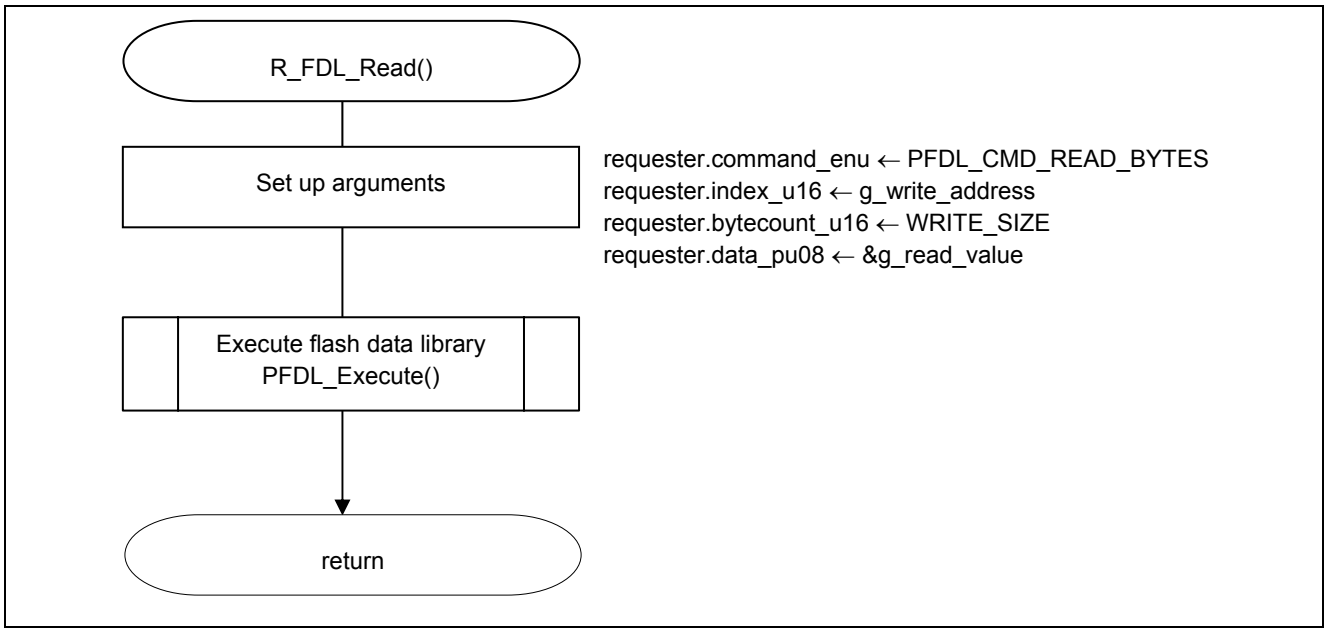

Figure 5.22 Processing the Data Read Command

5.9.21 Clearing a Switch-Pressed Status

Figure 5.23 shows the flowchart for clearing a switch-pressed status.

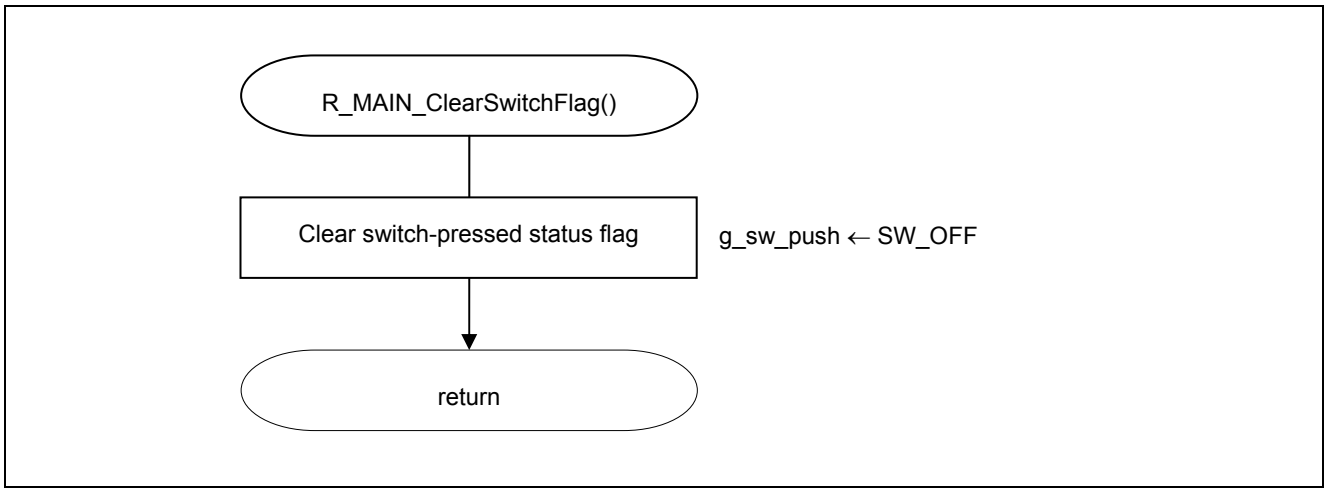

Figure 5.23 Clearing a Switch-Pressed Status

5.9.22 Update of String Displayed on Upper Column of LCD

Figure 5.24 shows the flowchart for updating a string displayed on the upper column of the LCD.

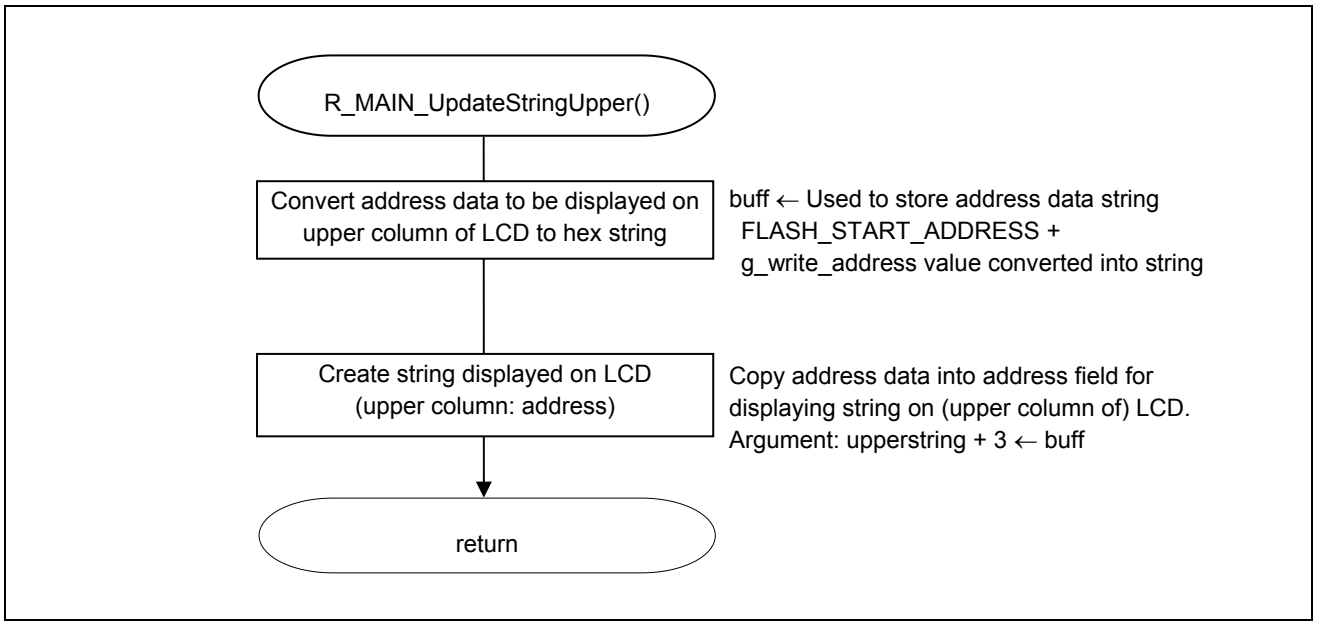

Figure 5.24 Update of String Displayed on Upper Column of LCD

5.9.23 Update of String Displayed on Lower Column of LCD

Figure 5.25 shows the flowchart for updating a string displayed on the lower column of the LCD.

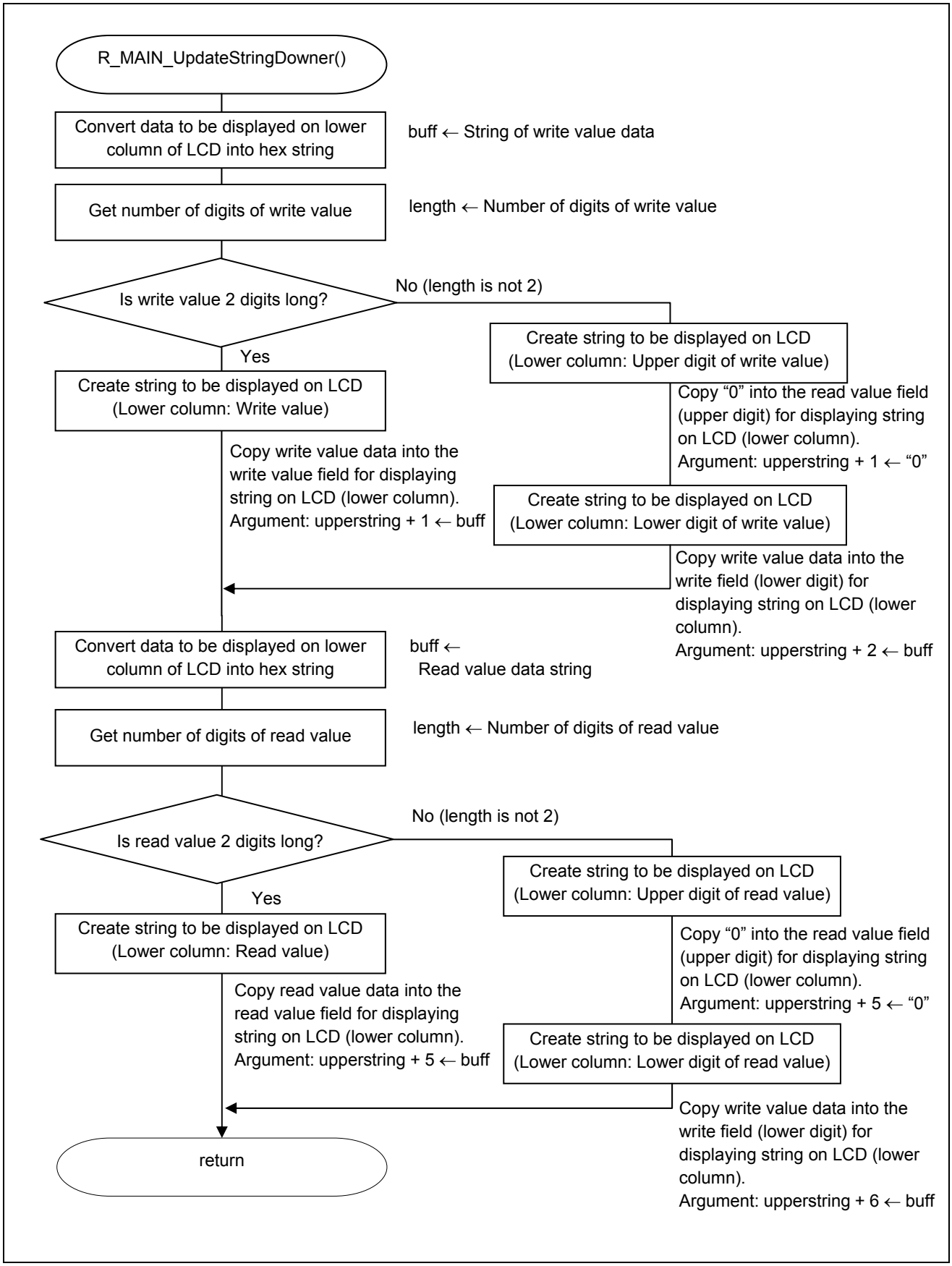

Figure 5.25 Update of String Displayed on Lower Column of LCD

5.9.24 Getting a Switch Status

Figure 5.26 shows the flowchart for getting a switch status.

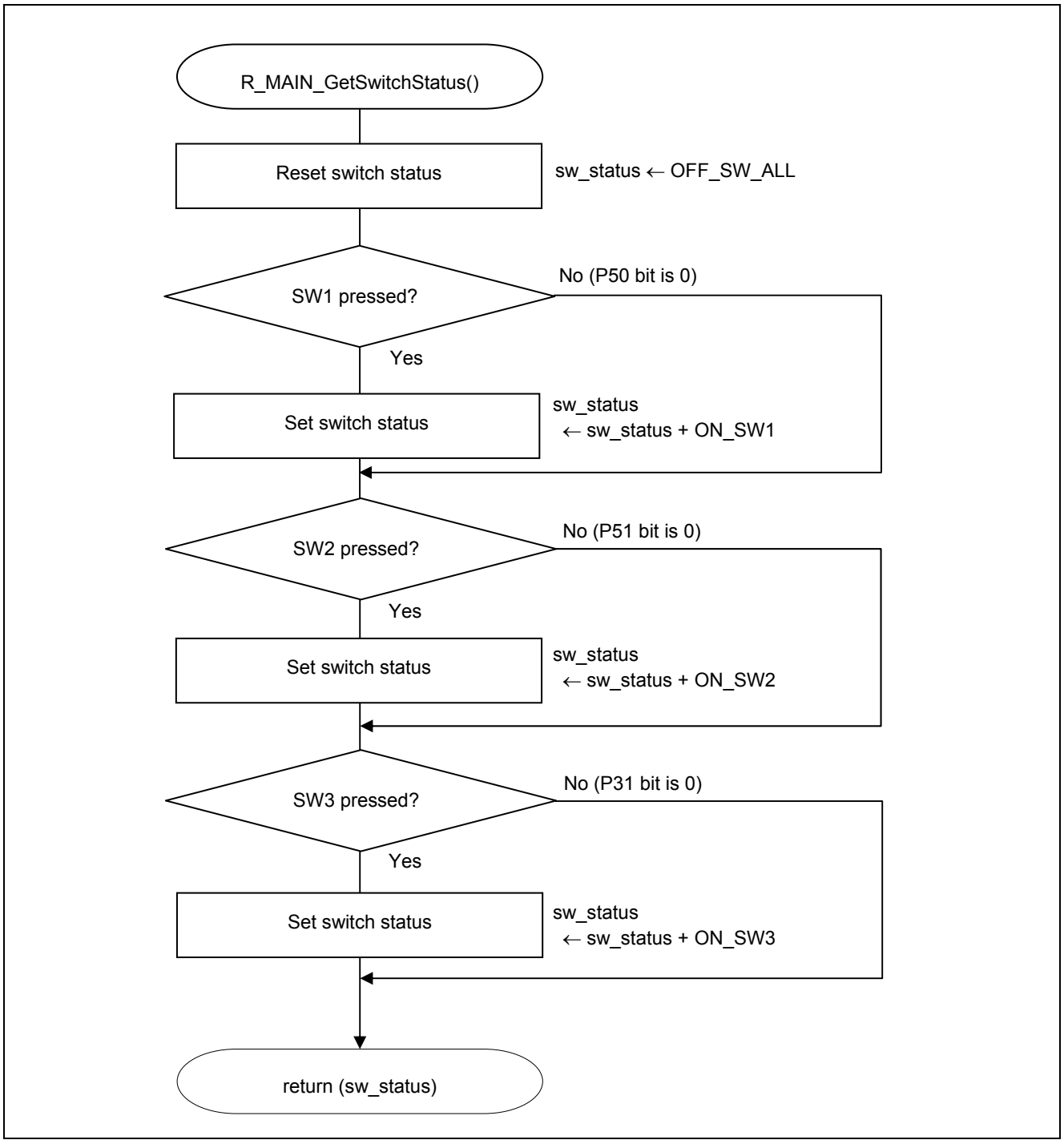

Figure 5.26 Getting a Switch Status

5.9.25 Processing of Switch-Pressed Status

Figures 5.27 and 5.28 show the flowcharts for processing a switch-pressed status.

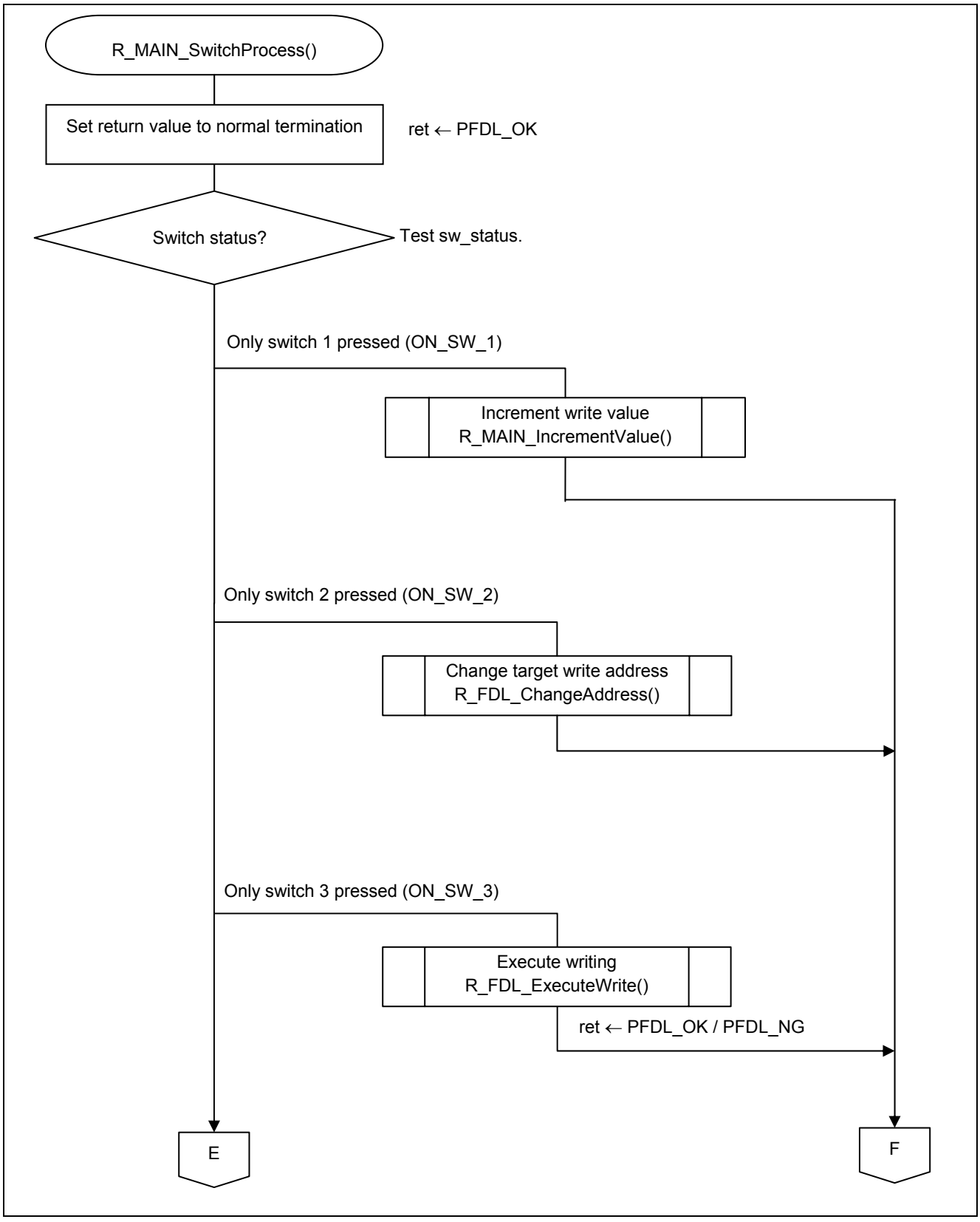

Figure 5.27 Processing of Switch-Pressed Status (1/2)

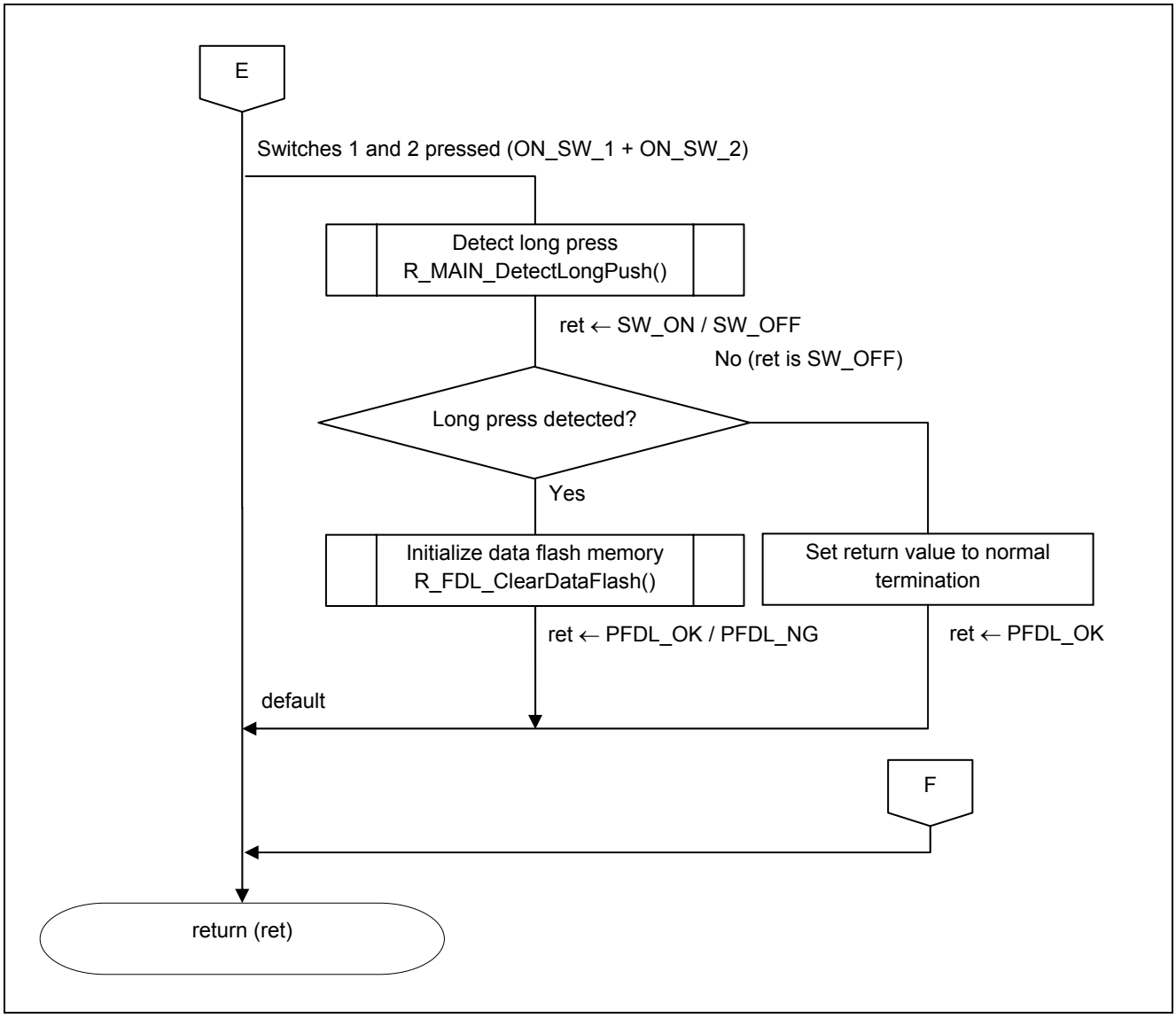

Figure 5.28 Processing of Switch-Pressed Status (2/2)

5.9.26 Increment of Write Value

Figure 5.29 shows the flowchart for incrementing a write value.

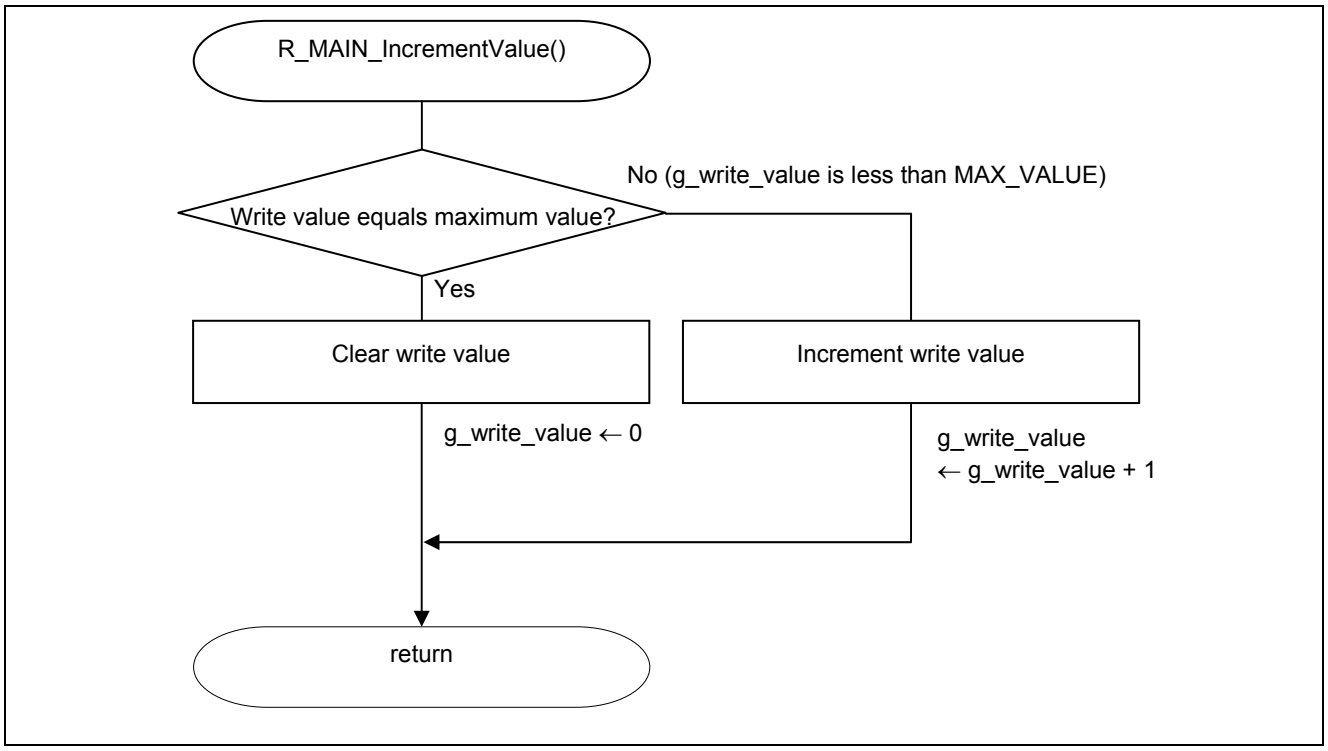

Figure 5.29 Increment of Write Value

5.9.27 Change of Target Write Address

Figure 5.30 shows the flowchart for changing a target write address.

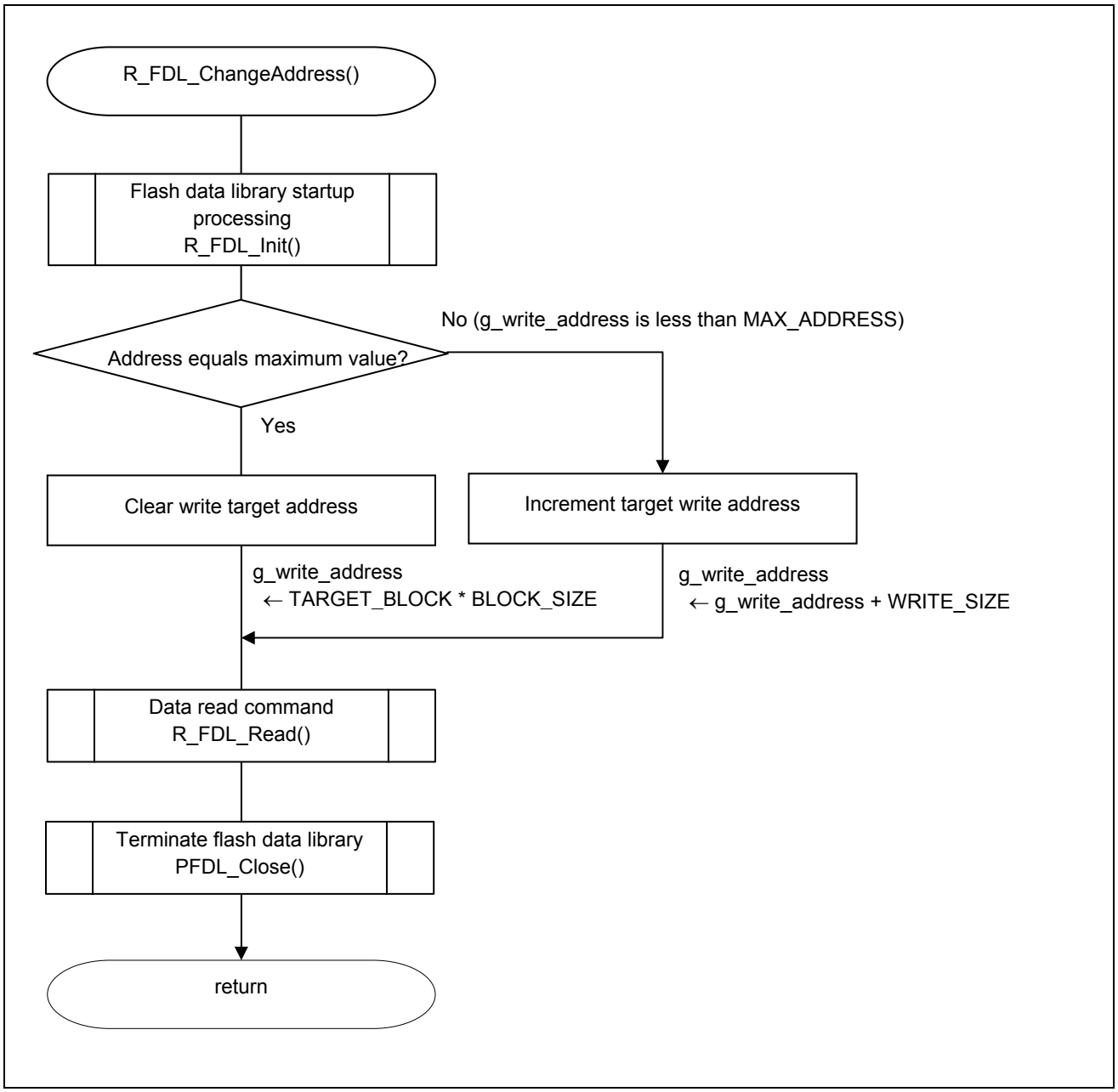

Figure 5.30 Change of Target Write Address

5.9.28 Writing Execution

Figures 5.31 and 5.32 show the flowcharts for writing execution.

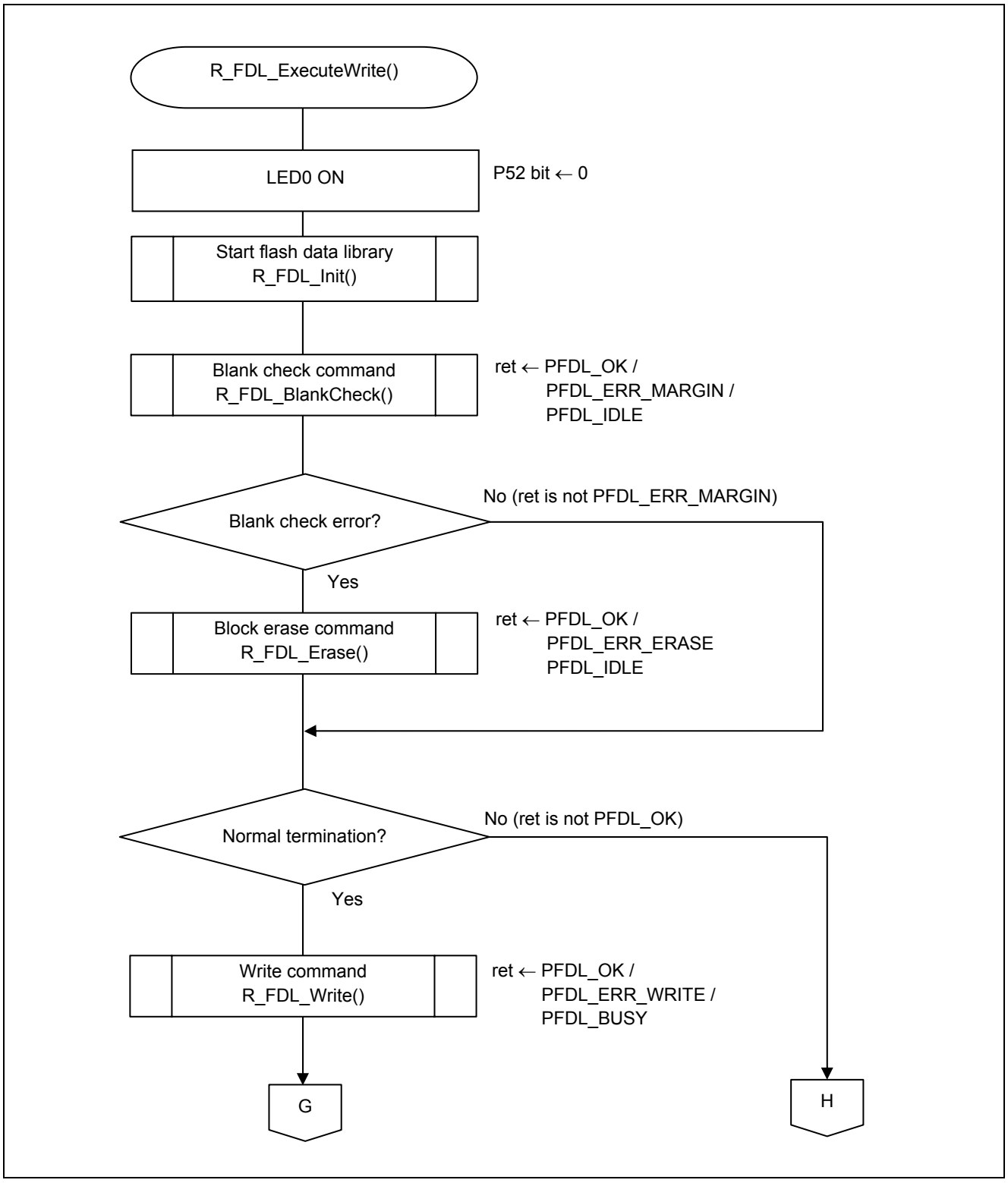

Figure 5.31 Writing Execution (1/2)

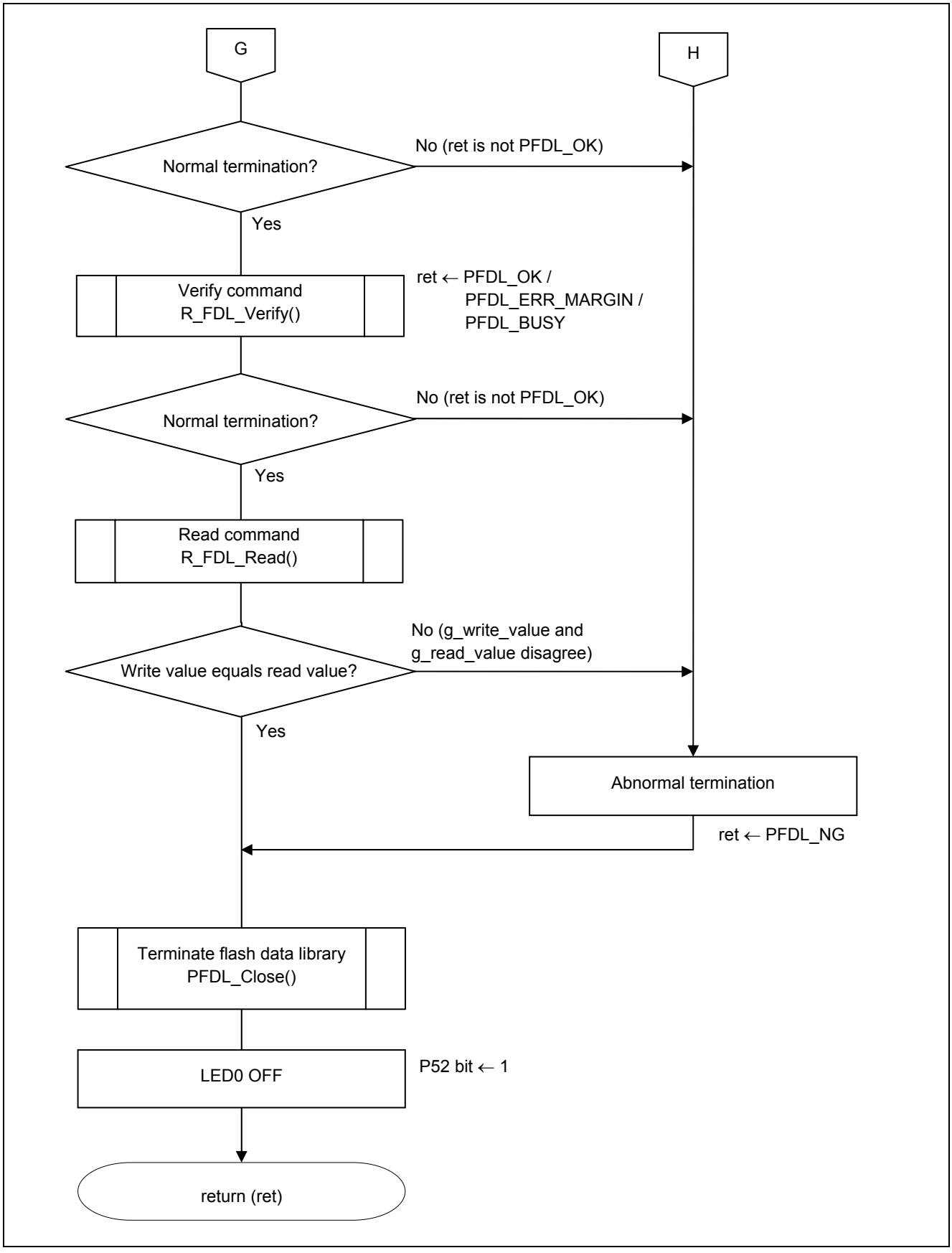

Figure 5.32 Writing Execution (2/2)

5.9.29 Processing the Blank Check Command

Figure 5.33 shows the flowchart for processing the blank check command.

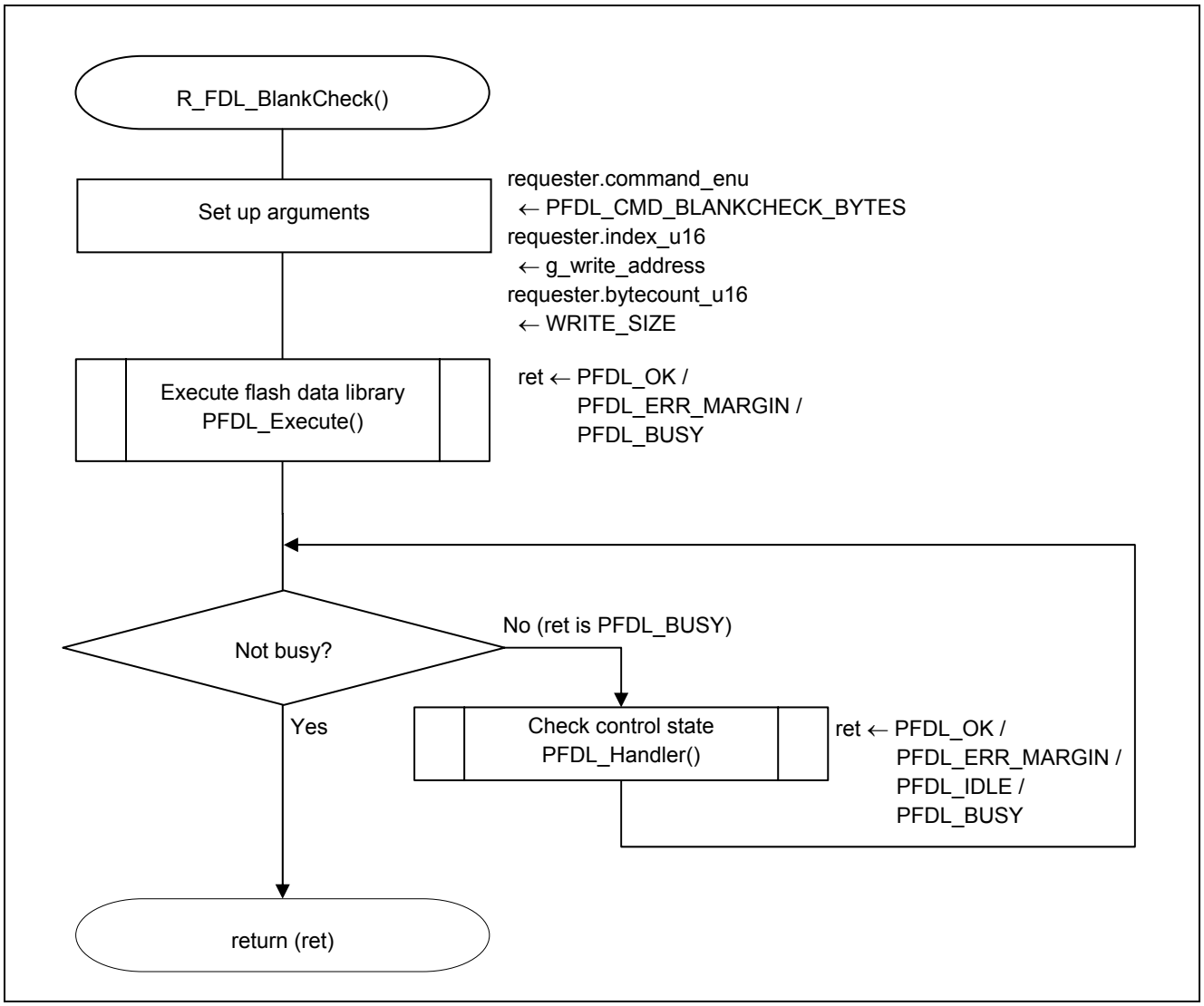

Figure 5.33 Processing the Blank Check Command

5.9.30 Processing the Block Erase Command

Figure 5.34 shows the flowchart for processing the block erase command.

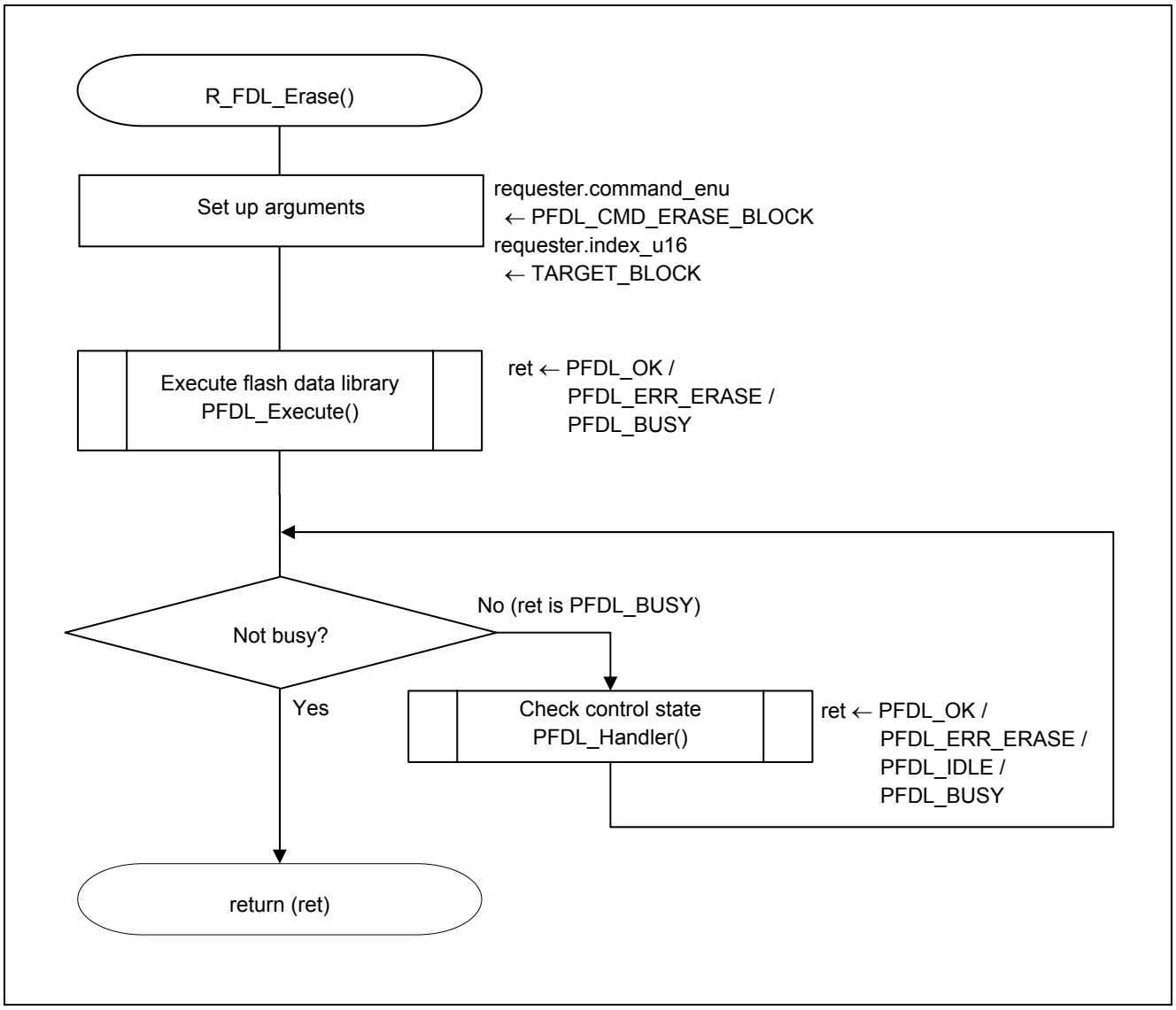

Figure 5.34 Processing the Block Erase Command

5.9.31 Processing the Data Write Command

Figure 5.35 shows the flowchart for processing the data write command.

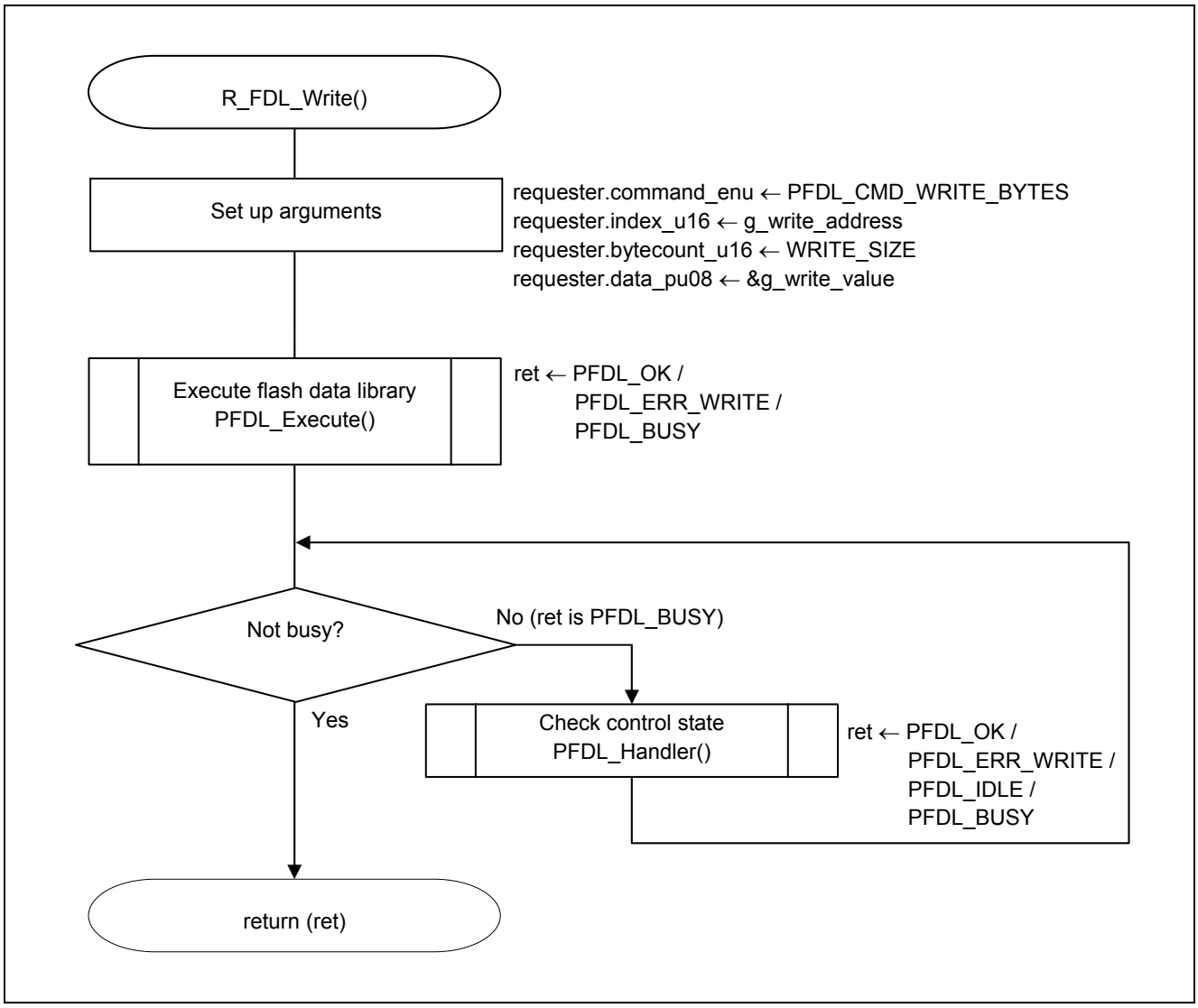

Figure 5.35 Processing the Data Write Command

5.9.32 Processing the Verify Command

Figure 5.36 shows the flowchart for processing the verify command.

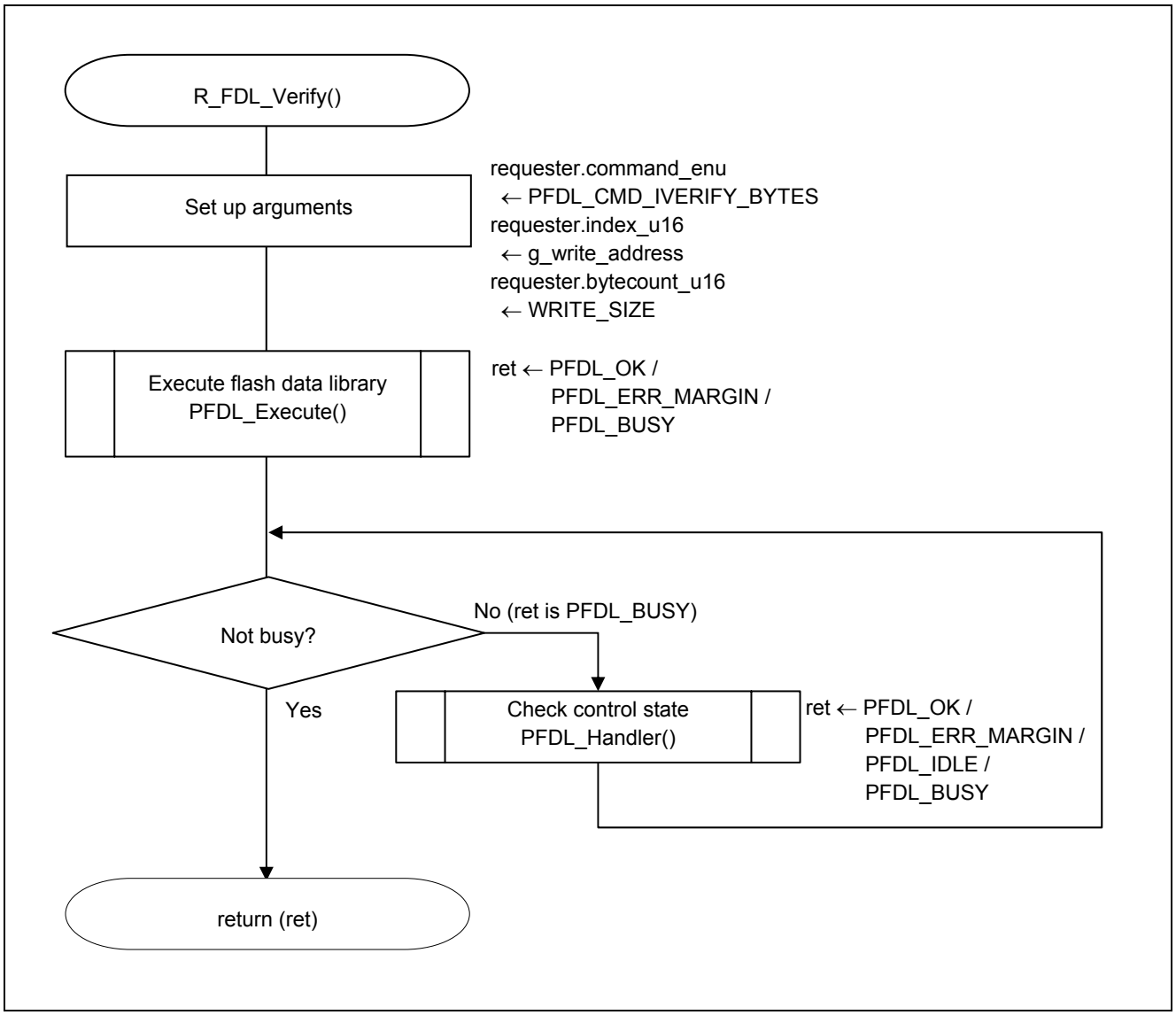

Figure 5.36 Processing the Verify Command

5.9.33 Detection of Long Press

Figures 5.37 and 5.38 show the flowcharts for detection of long press.

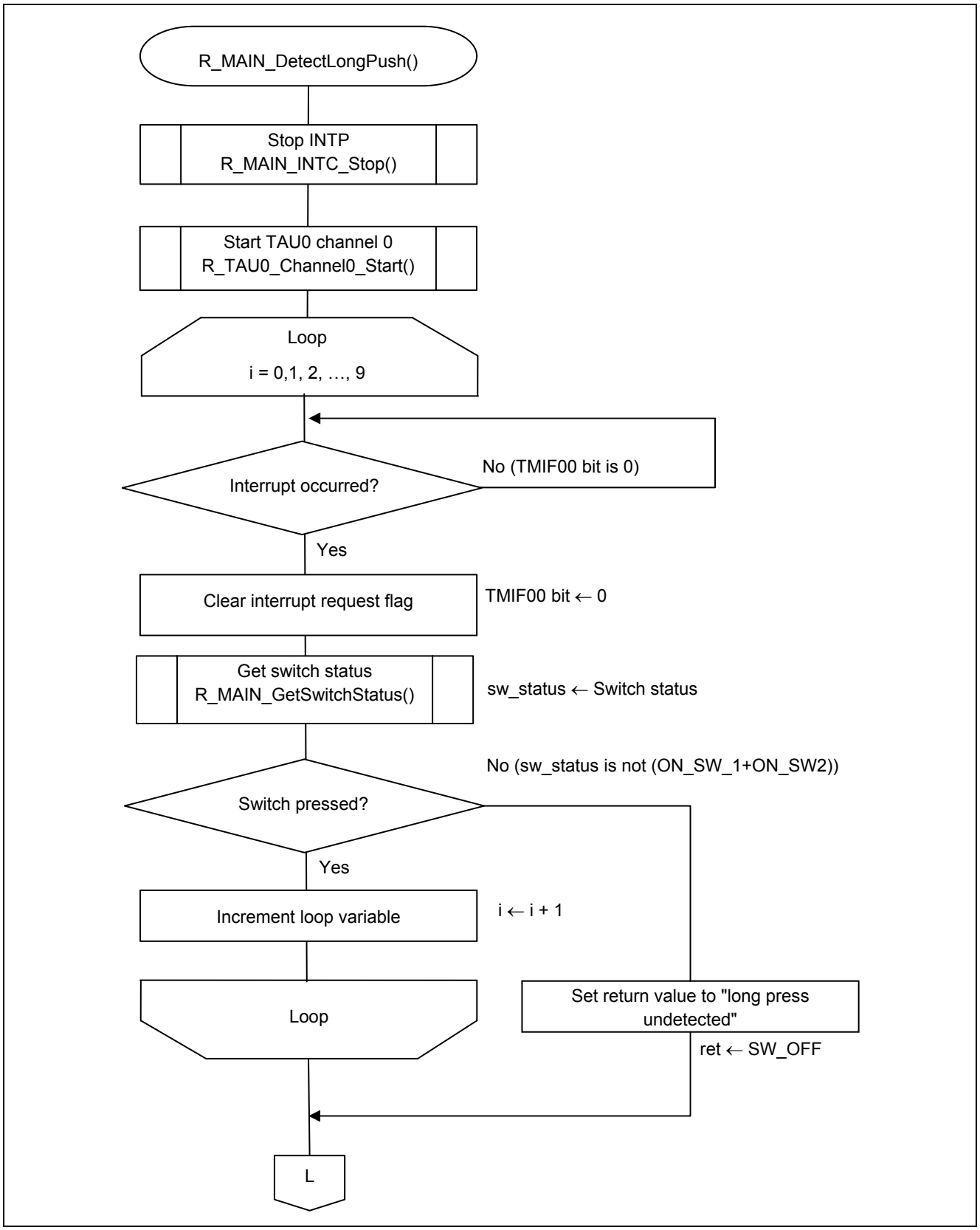

Figure 5.37 Detection of Long Press (1/2)

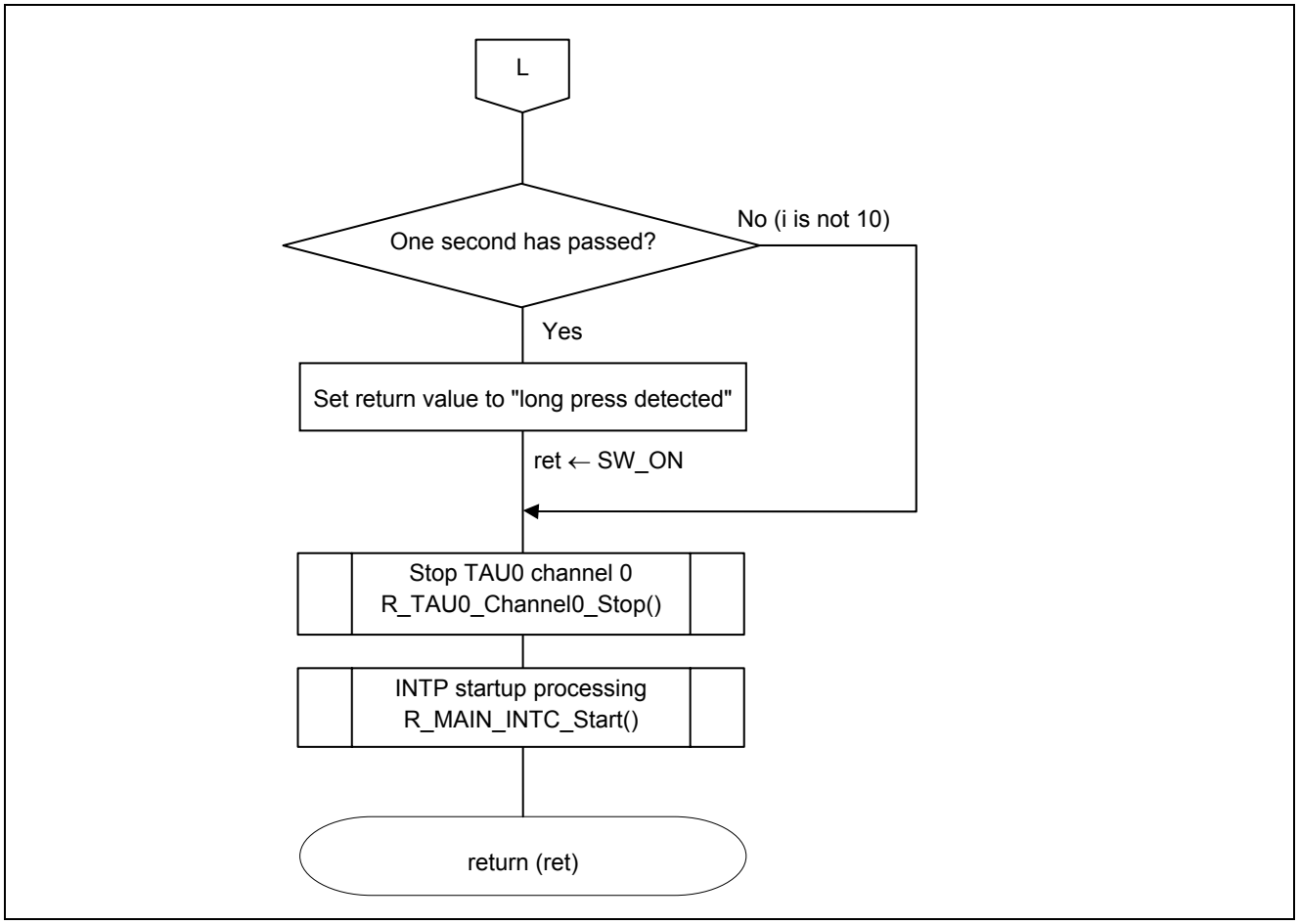

Figure 5.38 Detection of Long Press (2/2)

5.9.34 Stopping the INTP

Figure 5.39 shows the flowchart for stopping the INTP.

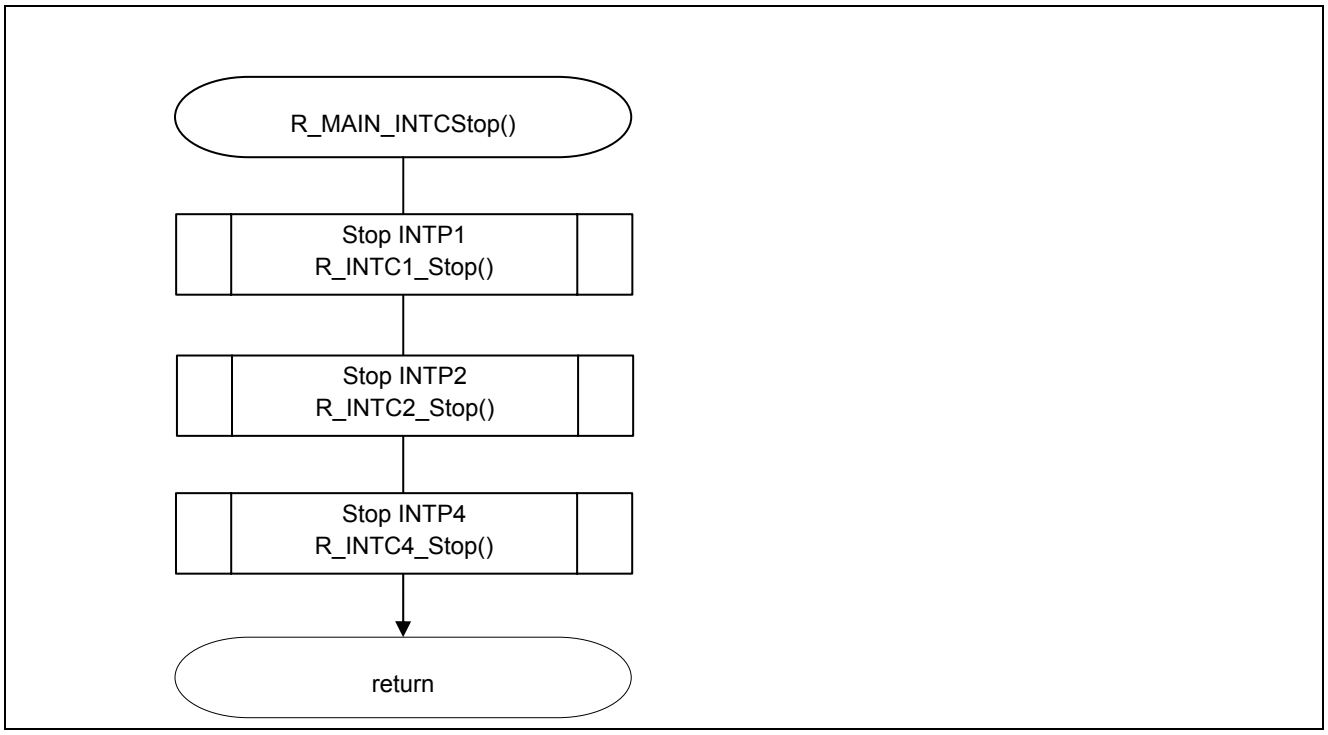

Figure 5.39 Stopping the INTP

5.9.35 Stopping the INTP1

Figure 5.40 shows the flowchart for stopping the INTP1.

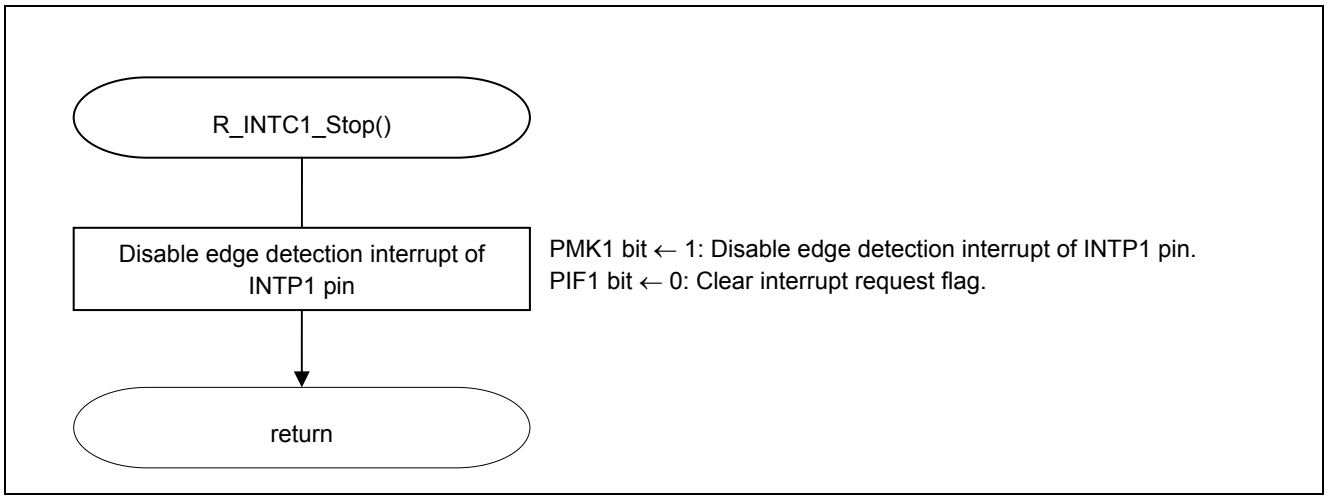

Figure 5.40 Stopping the INTP1

5.9.36 Stopping the INTP2

Figure 5.41 shows the flowchart for stopping the INTP2.

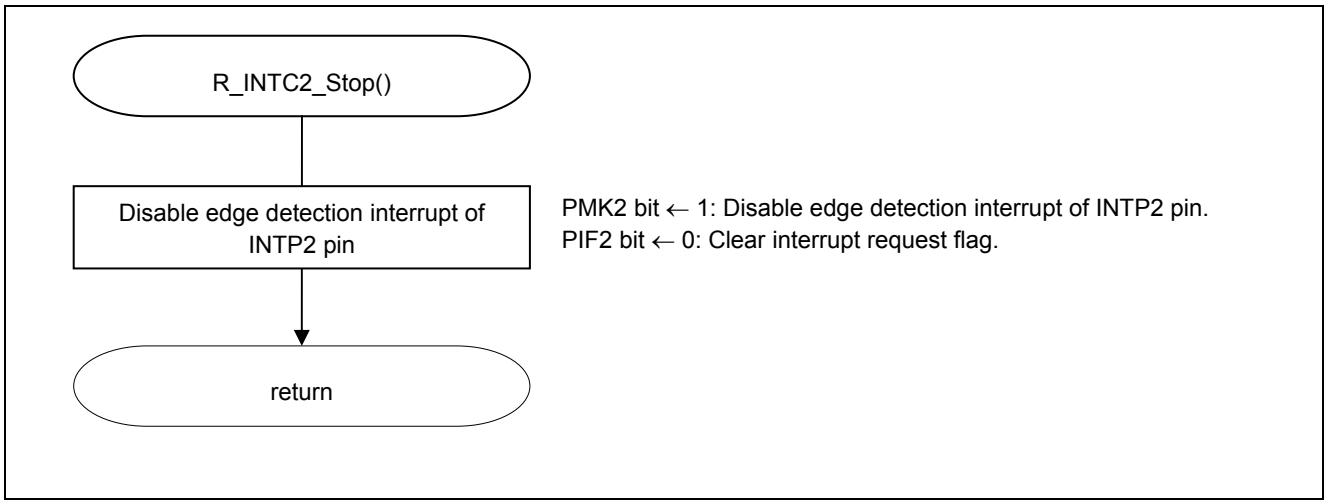

Figure 5.41 Stopping the INTP2

5.9.37 Stopping the INTP4

Figure 5.42 shows the flowchart for stopping the INTP4.

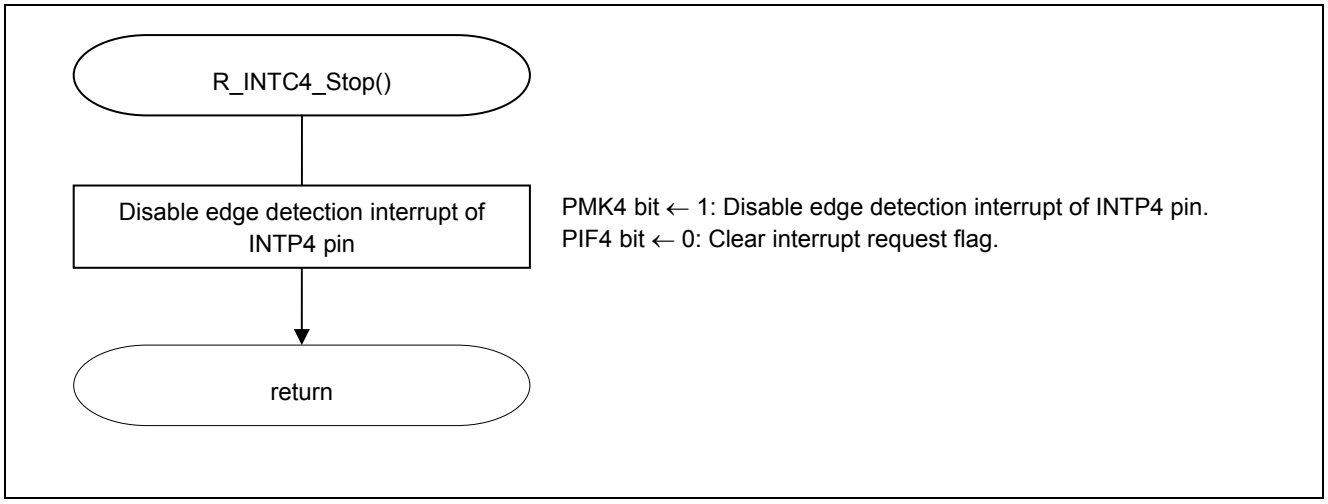

Figure 5.42 Stopping the INTP4

5.9.38 Starting the TAU0 Channel 0

Figure 5.42 shows the flowchart for starting the TAU0 channel 0.

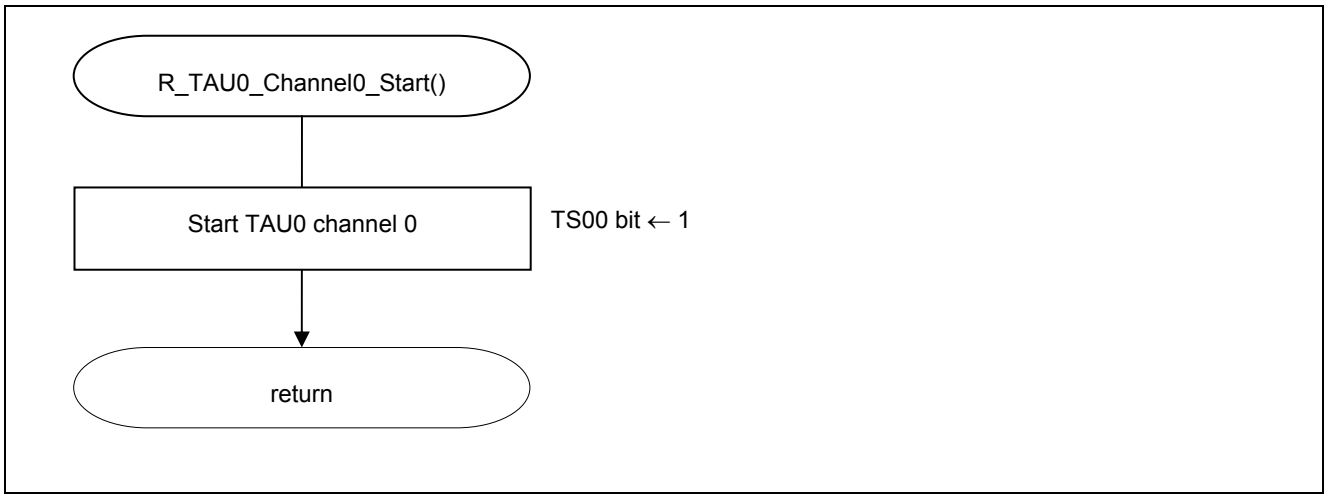

Figure 5.42 Starting the TAU0 Channel 0

5.9.39 Stopping the TAU0 Channel 0

Figure 5.43 shows the flowchart for stopping the TAU0 channel 0.

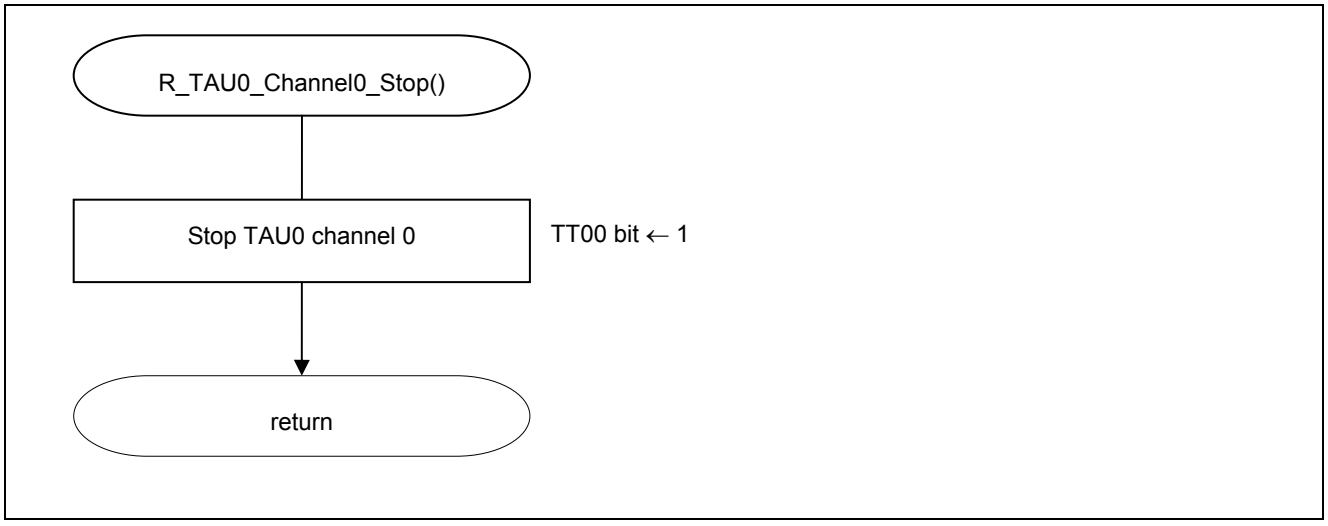

Figure 5.43 Stopping the TAU0 Channel 0

5.9.40 Data Flash Memory Initialization

Figure 5.44 shows the flowchart for data flash memory initialization.

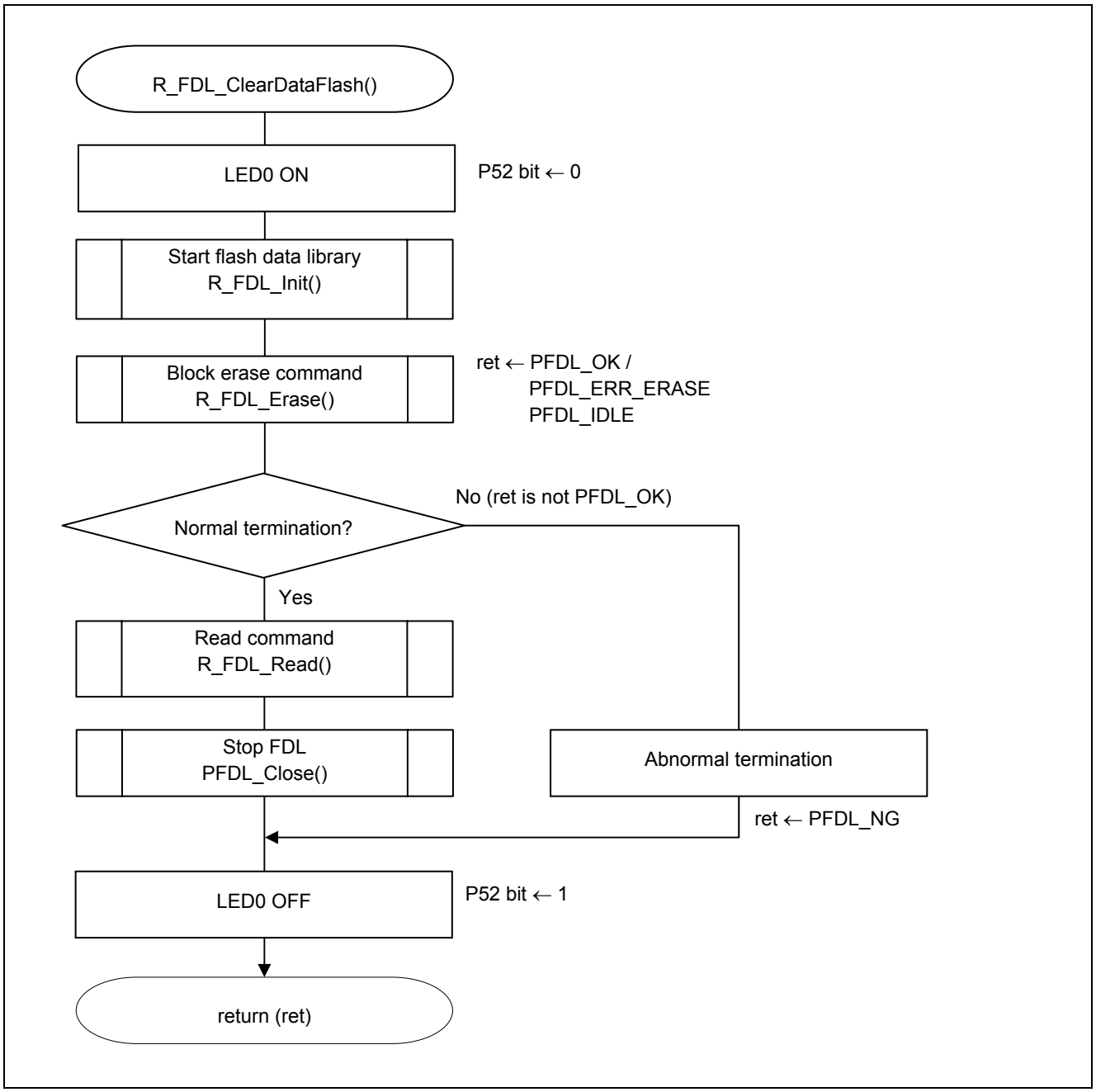

Figure 5.44 Data Flash Memory Initialization

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/G13 User's Manual: Hardware (R01UH0146E)

RL78 Family User's Manual: Software (R01US0015E)

RL78 Family Data Flash Library Type04 User's Manual (R01US0049E)

(The latest versions of the documents are available on the Renesas Electronics Website.)

Technical Updates/Technical Brochures

(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website

- http://www.renesas.com/index.jsp

Inquiries

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

Revision Record	RL78/G13 Data Flash Library Type04

Rev.	Date	Description	
		Page	Summary
1.00	May 28, 2015	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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 an MPU or MCU 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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 systems; anti-disaster systems; anti-crime systems; and safety 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 (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable 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. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages 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 its 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 be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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. Because the evaluation of microcomputer software alone is very difficult, 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 use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be 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, or if you have any other inquiries.

(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.

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, HALII 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 5.0