RL78/G11

Digital Voltmeter

R01AN4063EJ0100
Rev. 1.00
Jan. 19, 2018

Introduction

With the application described in this application note, a voltmeter is implemented by using the A/D converter incorporated in the RL78/G11.

Target Device

RL78/G11

When applying this application note to another microcontroller, careful evaluation is recommended after making modifications to comply with the microcontroller used.

Contents

1. Specifications

1.1 Voltage Measurement Method

With this application, the analog input voltage is measured by the A/D converter incorporated in the RL78/G11. Three ranges of voltages are measured: 0 to ±1 V (±1 V range), 0 to ±10 V (±10 V range), and 0 to ±100 V (±100 V range).

First, the analog voltage input circuit (Figure 1.1) is considered. The ANI0 pin is connected to GND via resistor R0. A switch is inserted between the ANI0 pin and the voltage to be measured, and either 0 Ω, R1, or R2 is used to connect the pin and the voltage.

When 0 Ω is used, the voltage is measured in the ±1 V range. When resistor R1, whose resistance is nine times the R0 resistance, is used, the voltage is measured in the ±10 V range, and when resistor R2, whose resistance is 99 times the R0 resistance, is used, the voltage is measured in the ±100 V range.

By using a resistor divider, the voltage applied to the ANI0 pin is controlled to 1 V or smaller. Note that it is not allowed to apply voltage equal to or larger than VDD to any pin of the RL78/G11.

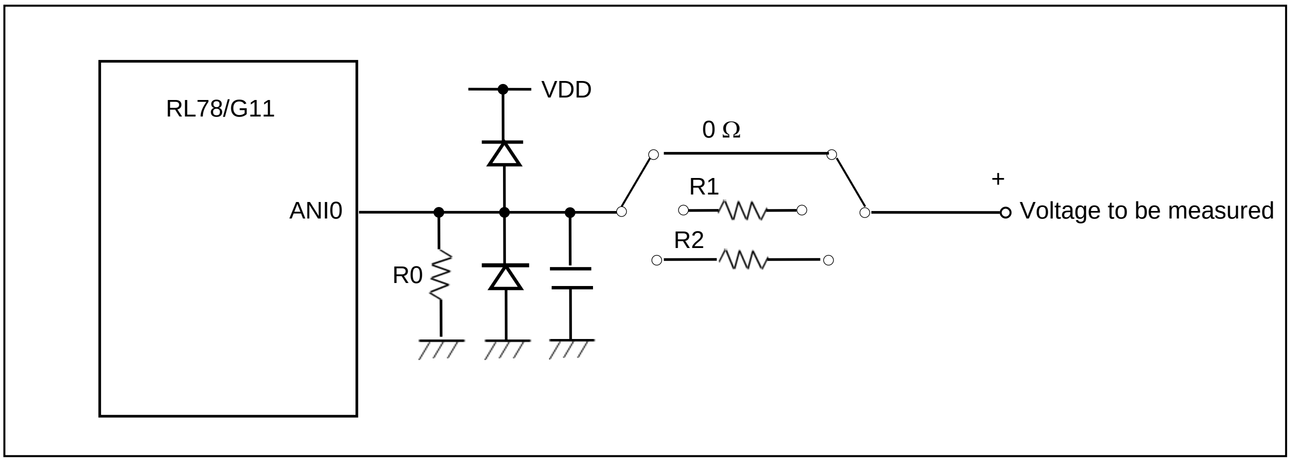


Figure 1.1 Analog Voltage Input Circuit

Next, the case is considered in which a negative voltage as well as a positive voltage is accepted as the voltage to be measured. As shown in Figure 1.2, two ports are added; the ports are connected to each other via two resistors; and the midpoint is connected both to the analog input pin and the minus (–) side voltage to be measured. By setting these ports to the high level output and low level output, the minus (–) side voltage to be measured is controlled to the intermediate potential (approximately 1/2 VDD based on the RL78/G11). With these arrangements, a voltage applied to the ANI0 pin is always positive. By measuring the plus (+) side and minus (–) side voltages and calculating the difference between the two voltages, the voltage of the target can be obtained.

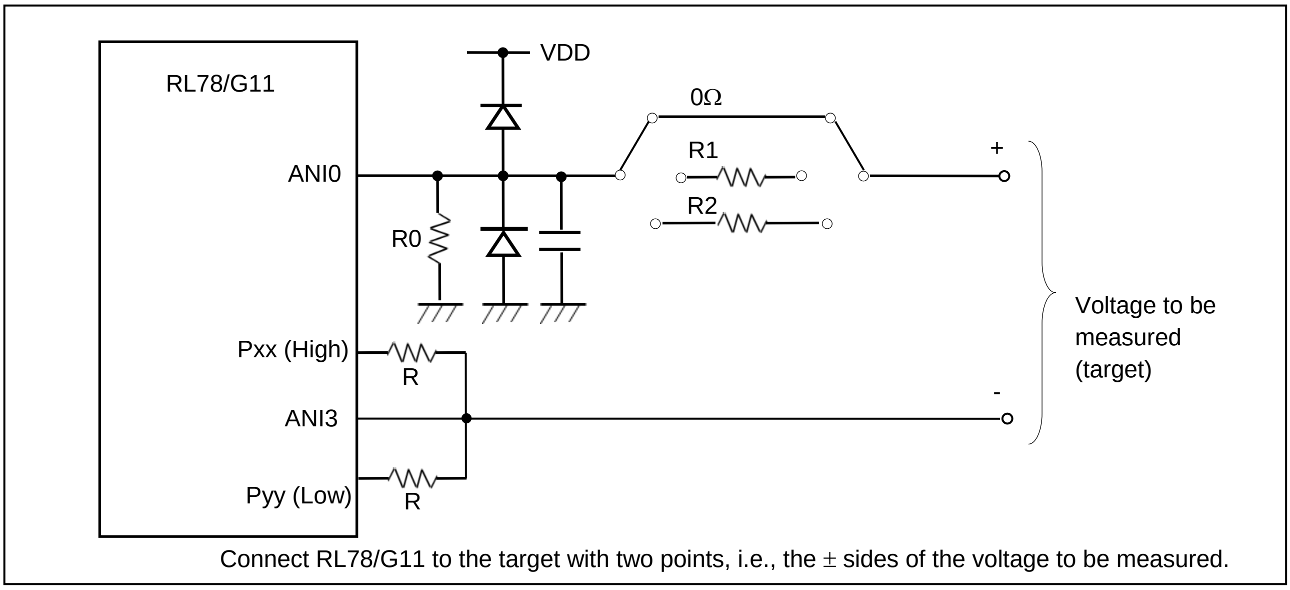


Figure 1.2 Handling Negative Voltage

1.2 Automatic Measurement Range Switching

In manual range switching, the resistors between the analog input pin and voltage to be measured are switched over, whereas in automatic range switching, the resistors that are connected to the GND side in manual range switching are switched over.

The R1 resistance is 1/9 the R0 resistance and the R2 resistance is 1/99 the R0 resistance. When port 1 is set to the high-impedance state and port 2 to low level output, R2 is selected and the voltage applied to the ANI0 pin is 1/100 the voltage to be measured. Measure the voltage applied to the ANI0 pin in this condition. If the measurement result falls between VDD/10 and VDD/100, R1 is selected; if the measurement result falls between VDD/100 and VDD/1000, neither R1 nor R2 is selected (both port 1 and port 2 are set to the high-impedance state). Measuring again the voltage applied to the ANI0 pin enables more accurate measurement of the voltage to be measured.

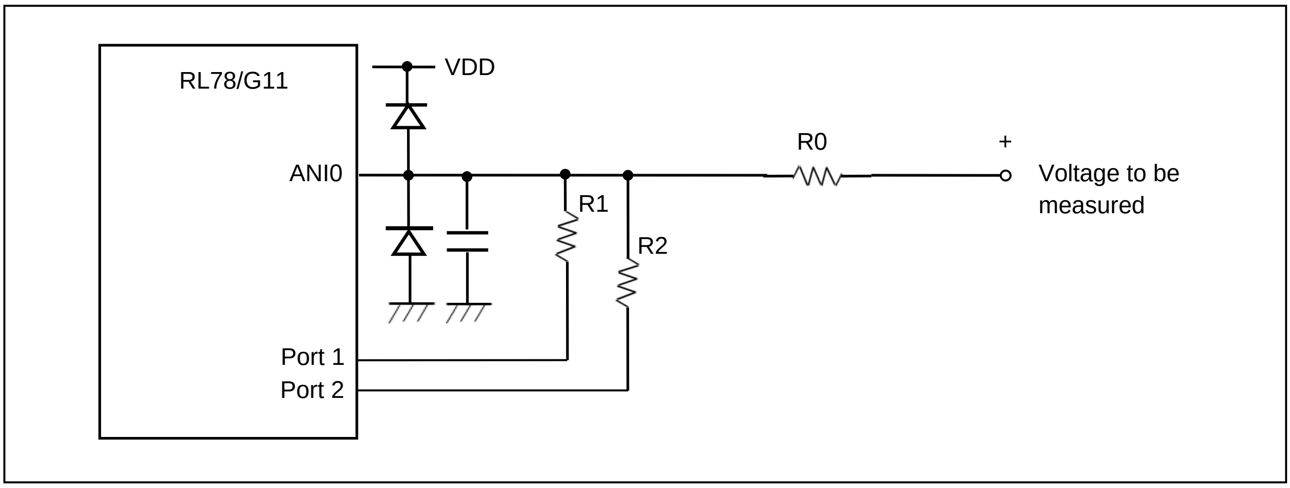


Figure 1.3 Automatic Measurement Range Switching

1.3 Determining Resistances and Measurement Ranges

The GND level for the resistor divider is generated for each range. The GND level is set to the intermediate potential (approximately 1/2 VDD based on the RL78/G11) by dividing the high-level output and low-level output of the ports by the resistors.

The GND level for the resistor divider is actually measured and the actually measured value is used for calculating the voltage to be measured.

The maximum voltage measurement range is ±100 V (the peak value of the alternating current is approximately ±150 V). The input impedance is approximately 1 MΩ, and R0, R1, and R2 are 910 kΩ, 82 kΩ, and 8.2 kΩ, respectively.

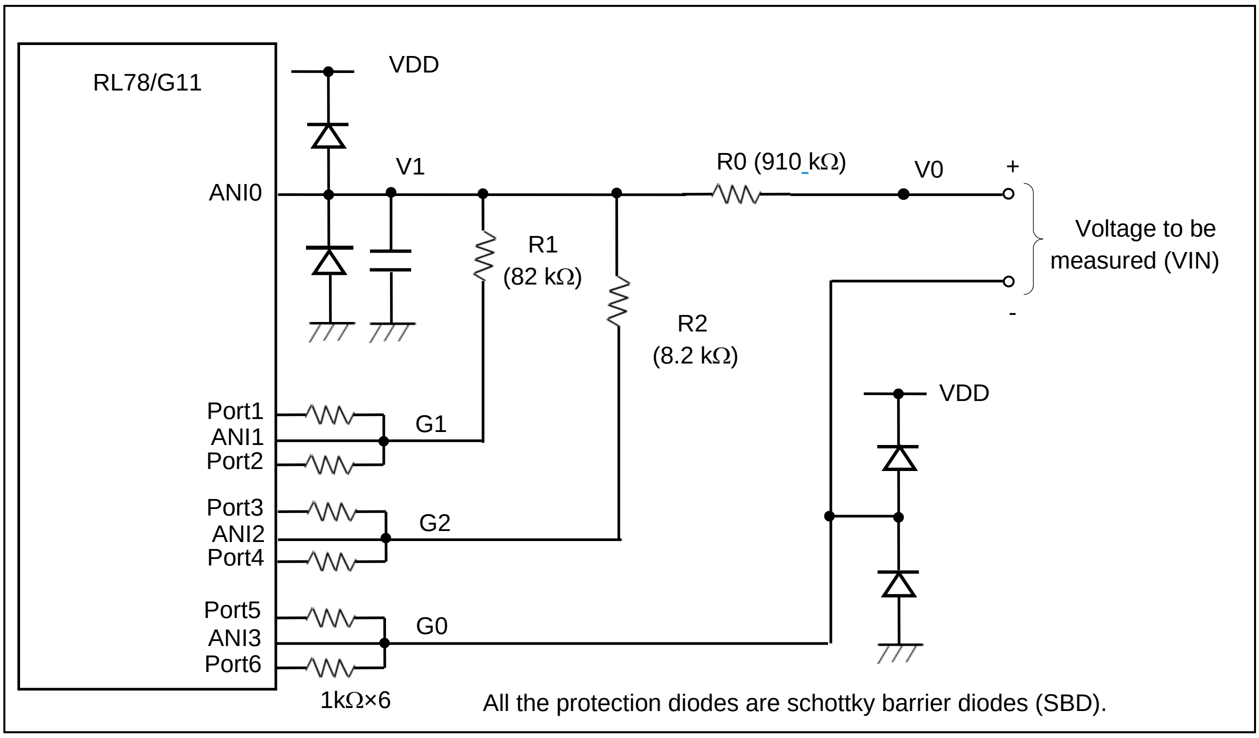


Figure 1.4 Measurement Circuit

1.4 Operation Processing of Measurement Results

Although using the floating-point operation for operation processing of voltage measurement enables easy programming, it takes longer operation time and also increases the program size because it uses the floating-point library. In this application, therefore, the floating-point operation is not used but the number of data bits is considered instead. The multiply, divide, and multiply-accumulate instructions incorporated in the RL78/G11 are used for multiplication and division according to the length of the bits to be operated.

When voltage-dividing resistor Rn (R1 or R2) is active, the following relationship holds between voltage V0 and the actually measured voltage (V1).

$$
\begin{aligned}
V1 &= (V0 - Gn) \times Rn/(R0 + Rn) + Gn \\
&= (V0 \times Rn + Gn \times R0)/(R0 + Rn)
\end{aligned}
$$

When this equation is rearranged,

$$
V0 = \{V1 \times (R0 + Rn) - Gn \times R0\}/Rn
$$

Therefore, voltage VIN to be measured is calculated as follows since it can be obtained by subtracting G0 from V0.

$$
\begin{aligned}
VIN &= V0 - G0 \\
&= \{V1 \times (R0 + Rn) - Gn \times R0\}/Rn - G0 \\
&= \{V1 \times (R0 + Rn) - (Gn \times R0 + G0 \times Rn)\}/Rn
\end{aligned}
$$

By comparing the first and second terms in the curly braces in the above equation, the voltage to be measured can be determined whether it is positive or negative. If the first term is larger, the voltage is positive, and if smaller, the voltage is negative.

1.5 Correction based on Internal Reference Voltage

VDD is used as the reference voltage of the A/D converter. When three alkaline cells are used for a VDD power source, the output voltage from the battery fluctuates between approximately 4.8 V (1.6 V × 3) to approximately 3 V (1.0 V × 3). This means that the fluctuation rate of the reference voltage of the A/D converter is more than 30%, resulting in poor A/D conversion accuracy. To avoid this, the measurement result is corrected by A/D-converting the internal reference voltage incorporated in the RL78/G11 (TYP. 1.45 V). However, the internal reference voltage varies by 5%.

When the internal reference voltage is A/D-converted, the result (SAR) is expressed as follows.

$$SAR = (1.45/VDD) \times 1024$$

When this equation is rearranged,

$$VDD = (1.45/SAR) \times 1024$$

From a slightly different point of view, the voltage of a single bit can be expressed as 1.45/SAR.

Therefore, the measurement result voltage is expressed as follows in this application note.

$$VIN = [\{V1 \times (R0 + Rn) - (Gn \times R0 + G0 \times Rn)\}/Rn] \times 1.45/SAR$$

Since the above equation is solved by integer operation not by floating-point operation, the following corrections are made.

(1) Resistance is calculated in 0.1 kΩ units.

(2) 1.45 is assumed as “145/100”; and the result is not divided by 100 but obtained in 10 mV units.

Although the voltage to be measured is calculated in 10 mV units, it can be handled with 16 bits as long as the voltage to be measured is alternating 100 V.

Note that, in this application, division is performed to enable 32-bit operation on the voltage to be measured (effective value). Since division is performed using a shift instruction, divisors should be 16 and 4 for the ±100 V range and ±10 V range, respectively. This controls the number of bits of the calculation result (g_vin_data[32]) of the voltage to be measured to 12 bits or fewer, which enables 32-bit operation to handle addition (calculation of the effective value) of the 32 squares of the calculation result (g_vin_data[32]).

1.6 Measurement Range Switching

Be sure to start measuring the voltage to be measured when it is connected to the intermediate potential (G2) via R2 (8.2 kΩ) (±100 V range). Also be sure to measure the voltage to be measured 32 times with the above settings. Determine the appropriate measurement range according to the maximum absolute value of the measurement result.

If the calculation result (g_vin_data[32]) of the voltage to be measured is 9 or smaller (equivalent to ±1.5 V) when R2 is selected, select the ±1 V range. If the calculation result is 10 to 93 (equivalent to ±15 V), select the ±10 V range. If the calculation result (g_vin_data[32]) exceeds 937 (equivalent to ±150 V), over range is indicated. After measurement is completed, select R2 (8.2 kΩ) to assure safety.

The absolute value of the voltage to be measured can be obtained by multiplying the calculation result (g_vin_data[32]) by the constant specific to the measurement range (16 for the ±100 V range and 4 for the ±10 V range) and changing the 10 mV units to 1 V units.

1.7 AC Source Support

This application supports 50- to 60-Hz sine waves.

A sine wave is sampled 32 times in a 1-ms period to measure the interval between the points (zero crosses) at which the polarity is reversed. In calculation of the effective value, only the voltage to be measured during the interval is included. If there are no zero crosses, the last 20 samples are used to calculate the effective value.

Note that this application does not support measurement of the pulsating current. To measure the pulsating current, the increased number of samples are required to calculate the effective values.

1.8 Display Period

The display switching period is 1 second. After the internal reference voltage is measured, the voltage to be measured is calculated each time an interrupt of the 1-ms period interval timer occurs to determine the voltage to be measured (effective value). Then, the 1-ms period interval timer is stopped; the voltage to be measured (effective value) is displayed; and the standby state (STOP mode) is set. After this, the above operations are repeated every second.

If the voltage to be measured (effective value) is equal to or smaller than ±10 mV for 20 consecutive times, the display is turned off; the 1-s period interval timer is stopped; and the standby state (STOP mode) is set. When the switch is pressed (INTP0 occurs), the voltage measurement and display are started again.

2. Operation Confirmation Conditions

The sample code covered in this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Description
Microcontroller used	RL78/G11 (R5F1056)
Operating frequency	● High-speed on-chip oscillator (HOCO) clock: 16 MHz ● CPU/peripheral hardware clock: 16 MHz
Operating voltage	3.0 V to 5.5 V LVD operation mode: reset mode; voltage: VLVD2 = Min. 3.00 V
Integrated development environment(CS+)	CS+ V5.00.00 from Renesas Electronics
Integrated development environment(e2studio)	e2 studio 5.3.0.023 from Renesas Electronics
Compiler	CC-RL V1.04.00 from Renesas Electronics
Board used	RL78/G11 target board + LCD

3. Related Application Notes

Refer to the following related application notes for additional information.

RL78/G13 Initialization CC-RL(R01AN2575E) Application Note
RL78/G10 Square Root Program CC-RL (R01AN3079E) Application Note

4. Hardware Descriptions

4.1 Hardware Configuration Example

Figure 4.1 shows the hardware configuration example covered in this application note.

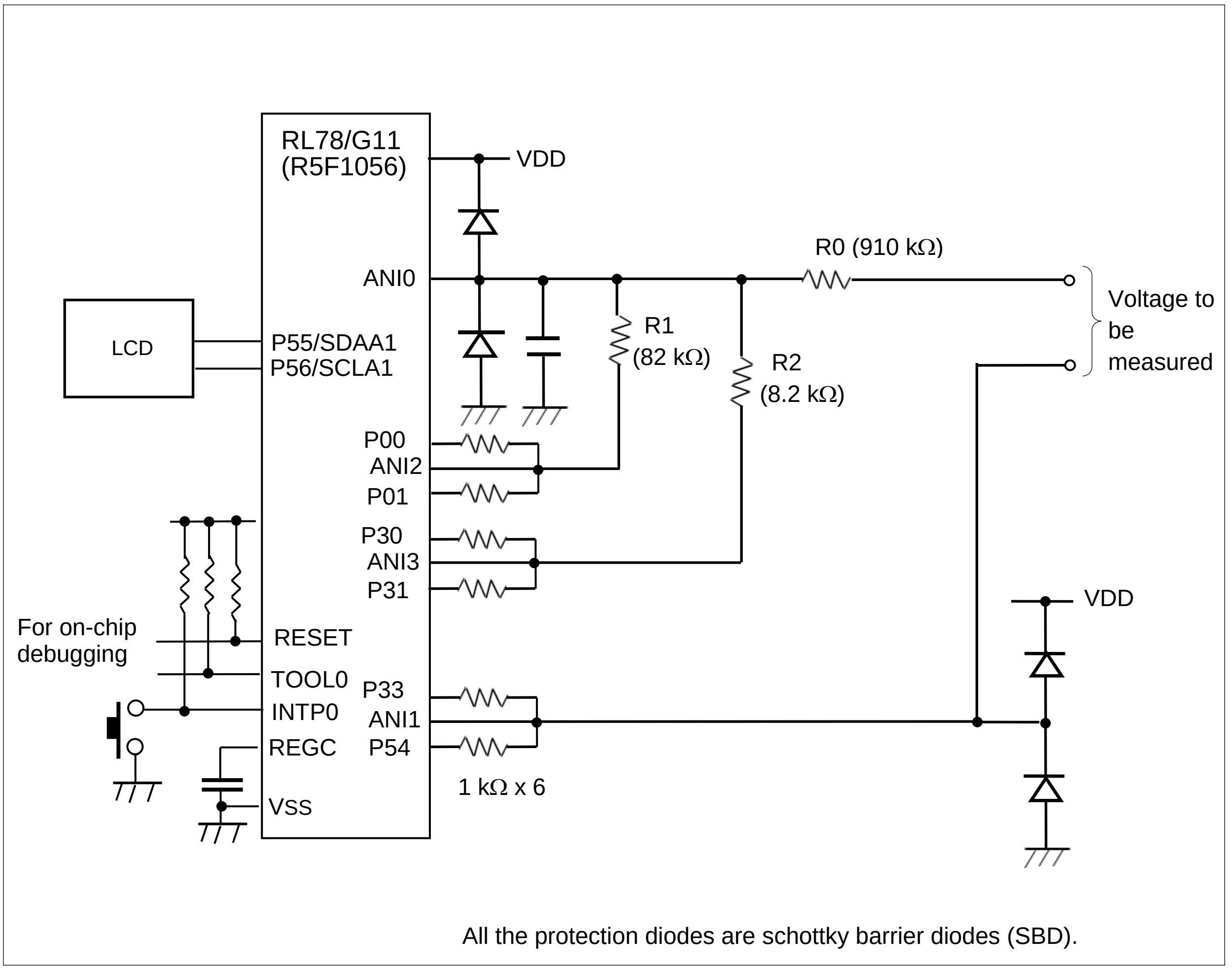


Figure 4.1 Hardware Configuration

Notes: 1. The above figure is a simplified circuit image for showing the outline of the connections. The actual circuit should be designed so that the pins are handled appropriately and that the electrical characteristics are satisfied (input-only ports should be each connected to V_{DD} or V_{SS} via a resistor).

2. VDD must be equal to or greater than the reset release voltage (V_{LVD}) specified with LVD.

4.2 List of Pins Used

Table 4.1 lists the pins used and their functions.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Description
ANI0 to ANI3	Input	Analog signal input
P00, P01, P30, P31, P33, P54	Output	Intermediate voltage generation for each measurement range
P55/SDAA1, P56/SCLA1	I/O	I2C bus control for transferring display data to LCD
INTP0/P137	Input	SW input

5. Software Descriptions

5.1 Operation Summary

With this application, a voltmeter is implemented by using the A/D converter incorporated in the RL78/G11.

(1) After the initial settings, “Renesas RL78/G11” is displayed on the LCD and the standby state (STOP mode) is set.

(2) When the switch is pressed, INTP0 is generated and the standby state is released.

(3) The 1-s timer is activated.

(4) The internal reference voltage is measured. The voltage to be measured is measured each time an interrupt of the 1-ms period interval timer occurs. The analog input voltages of ANI0 to ANI3 are measured in this order. The A/D conversion results are saved in the internal RAM by using the DTC.

(5) The voltage to be measured is sampled 32 times in a 1-ms period to measure the maximum absolute value of the voltage to be measured and the interval between the points (zero crosses) at which the polarity of the voltage to be measured is reversed.

(6) The maximum absolute value of the voltage to be measured is judged, the measurement ranges are switched when necessary, and the voltage is measured every 1 ms 32 times again. Also, the interval is measured between the points (zero crosses) at which the polarity is reversed.

(7) After 32nd voltage measurement, the voltage to be measured (effective value) between the points at which the polarity is reversed is calculated. When there are no zero crosses, the last 20 samples are used to calculate the effective value.

(8) The 1-ms period interval timer is stopped; the voltage to be measured (effective value) that has been calculated is displayed on the LCD; and the standby state (STOP mode) is set. After this, (4) to (8) above are repeated every second.
If the voltage to be measured (effective value) is equal to or smaller than ±10 mV for 20 consecutive times, the display is turned off; the 1-s period interval timer is stopped; and the standby state (STOP mode) is set. When the switch is pressed (INTP0 occurs), the operation is restarted at (2).

5.2 List of Option Byte Settings

Table 5.1 shows an example of the option byte settings.

Table 5.1 Option Byte Setting Example

Address	Setting	Description
0x000C0	0b11101110	Disables the watchdog timer. (Stops counting after the release from the reset state.)
0x000C1	0b01111111	LVD reset mode: 2.45 V (2.40 V to 2.50 V)
0x000C2	0b11101001	HS mode; HOCO: 16 MHz
0x000C3	0b10000100	Enables the on-chip debugger.

5.3 List of Constants

Table 5.2 lists the constants used in the sample code.

Table 5.2 Constants used in Sample Code

Constant Name	Setting	Description
TRUTH	1	True
FALSE	0	False
COMBYTE	0x00	Command write mode to LCD
DATABYTE	0x80	Data write mode to LCD
CLRDISP	0x01	LCD clear instruction
HOMEPOSI	0x02	LCD home position instruction
LCD_Mode	0b00111000	LCD display mode (2-line display)
DISPON	0b00001111	LCD display on
ENTRY_Mode	0b00000110	Right-shift of LCD display data
SLAVEADDR	0xA0	LCD slave address on I2C bus
R0_OHM	10000	R0 resistance (1 MΩ)
R1_OHM	820	R1 resistance (82 kΩ)
R2_OHM	82	R2 resistance (8.2 kΩ)
R0_R2	R0_OHM + R2_OHM	Combined resistance of R0 and R2
R0_R1	R0_OHM + R1_OHM	Combined resistance of R0 and R1
LIMIT100	937	Maximum value for 100-V range
LIMIT10	93	Maximum value for 10-V range
LIMIT1	8	Maximum value for 1-V range
SLEEPTIME	20	Number of no-inputs before stop

5.4 List of Variables

Table 5.3 lists the variables used in the sample code.

Table 5.3 Global Variables used in Sample Code

Type	Variable Name	Contents	Function Used
uint16_t	ad_buff[4]	A/D conversion result buffer	R_ADC_Init(), r_adc_interrupt()
uint16_t	g_vin_data[32]	Measurement result buffer	R_ADC_Init(), r_adc_interrupt()
uint16_t	g_vref	Internal reference voltage measurement result	R_ADC_Init(), r_adc_interrupt(), main()
uint16_t	g_vmax	Maximum voltage measured	R_ADC_Init (), r_adc_interrupt(), main()
uint8_t	g_range	Measurement range	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_times	Number of measurements	main(),R_MAIN_UserInit(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_vin_sign	Sign of measurement result	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_sign_chng1	Zero cross point 1	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_sign_chng2	Zero cross point 2	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_sign_chng3	Zero cross point 3	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint16_t	g_voltage	Measurement result (10-mV units)	main()
uint8_t	g_period	Number of valid measured data	main()
uint8_t	g_start	Start position of valid data	main()
uint8_t	g_stpcnt	Voltage not-detect counter	main(),R_MAIN_UserInit()
int8_t	g_2ndline	Display data buffer	main()
st_dtc_data_t	dtc_controldata_0	DTC parameter	R_DTC_Create(), R_ADC_DTC_Init(), r_adc_interrupt(), main()
uint8_t	g_iica_status	IICA1 status	R_IICA_bus_check, r_iica_interrupt(), R_IICA_wait_comend
uint8_t*	gp_iica_tx_address	IICA1 transmission pointer	R_IICA_Master_Send(), r_iica_interrupt()
uint8_t	g_iica_tx_cnt	IICA1 transmission counter	R_IICA_Master_Send(), r_iica_interrupt()
uint8_t	g_write_commmand[2]	LCD command buffer	set_command()
uint8_t	g_write_data[2]	LCD data buffer	set_data()

5.5 List of Functions

Table 5.4 lists the functions used in this application.

Table 5.4 List of Functions

Function Name	Outline
R_MAIN_UserInit()	Activates the various features etc.
InitialiseDisplay	Initializes LCD display.
wait_time	Waits for time in 60us units.
wait_60us	Waits for 60 us.
set_command	Sets the LCD command.
set_data	Sets LCD display data.
DisplayString	Displays ASCII character strings on LCD.
R_IICA_Master_Send	Transmits data to LCD with IICA.
R_IICA_wait_comend	Waits for completion of IICA transfer.
R_IICA_StopCondition	Issues the stop condition from IICA.
R_IICA_bus_check	Checks the IIC bus state and issues the start condition.
r_iica_interrupt	Processes the IICA interrupt.
r_intp0_interrupt	Processes the INTP0 interrupt.
r_it_interrupt	Processes the INTIT00 interrupt.
R_ADC_inter_ref	Measures the internal reference voltage.
R_ADC_DTC_Init	Sets A/D and DTC.
r_tm01_start	Activates TM01 (1-ms interval timer).
delay_us	Waits for 1 us.
r_adc_interrupt	Completes DTC transfer of A/D conversion results.
__ssqrt	Square-root operation of 32-bit data (For details, refer to R01AN3079.)
__r_mul32	16 bits × 16 bits
__r_div32	32 bits/16 bits
__r_div16	16 bits/8 bits
__r_mod16	Modulo operation of 16 bits/8 bits
__r_rms16	Root-mean-square operation of 16-bit data

5.6 Function Specifications

The following gives the specifications of the functions used in the sample code.

[Function Name] R_MAIN_UserInit

Outline	Initializes the various features used.
Header	r_cg_macrodriver.h, r_cg_userdefine.h
Declaration	void R_ MAIN_UserInit (void);
Explanation	Initializes the variables and others used.
Arguments	None
Return value	None
Remarks	None

[Function Name] InitialiseDisplay

Outline	Initializes LCD display.
Header	lcd.h
Declaration	void InitialiseDisplay (void);
Explanation	Initializes the display controller and LCD display .
Arguments	None
Return value	None
Remarks	None

[Function Name] wait_time

Outline	Waits for time in 60 us units.
Header	r_cg_macrodriver.h
Declaration	void wait_time(uint16_t time);
Explanation	Waits for a time represented by 60 us × the number specified with the argument.
Arguments	Wait time in 60 us units
Return value	None
Remarks	None

[Function Name] wait_60us

Outline	Waits for 60 us using TM03.
Header	r_cg_macrodriver.h
Declaration	void wait_60us(void);
Explanation	Waits for 60 us using TM03 (delay count).
Arguments	None
Return value	None
Remarks	None

[Function Name] set_command

Outline	Transmits the command to LCD via I2C.
Header	r_cg_macrodriver.h, lcd.h, r_iica_user.h
Declaration	void set_command(uint8_t comcode);
Explanation	Transmits the command specified with the argument to the LCD controller via I2C.
Arguments	Command to LCD
Return value	None
Remarks	None

[Function Name] set_data

Outline	Transmits data to LCD via I2C.	
Header	r_cg_macrodriver.h, lcd.h, r_iica_user.h	
Declaration	void set_data(uint8_t datacode);	
Explanation	Transmits the display data specified with the argument to the LCD controller via I2C.	
Arguments	Display data to LCD	
Return value	None	
Remarks	None	

[Function Name] DisplayString

Outline	Displays ASCII character strings on LCD.	
Header	r_cg_macrodriver.h, lcd.h, r_iica_user.h	
Declaration	void DisplayString(uint8_t position, __far int8_t * string);	
Explanation	Displays the ASCII character strings specified with the second argument on the LCD position specified with the first argument via I2C.	
Arguments	First argument	Display start position on LCD
	Second argument	Pointer to the character strings to be displayed
Return value	None	
Remarks	None	

[Function Name] R_IICA_Master_Send

Outline	Starts data transmission to the I2C slave.	
Header	r_cg_macrodriver.h, r_iica_user.h	
Declaration	MD_STATUS R_IICA_Master_Send(uint8_t adr, uint8_t * const tx_buf, uint16_t tx_num);	
Explanation	Starts transmitting the specified data to the slave specified with the argument by using IICA1. Returns an error as the transmission status if the I2C bus is currently used, and starts transmitting the slave address when the I2C bus is available.	
Arguments	First argument	Slave address
	Second argument	Pointer to data to be transmitted
	Third argument	Number of data to be transmitted
Return value	Transmission status:	
	MD_OK	Successfully started
	MD_ERROR1	I2C bus is currently used.
Remarks	None	

[Function Name] R_IICA_wait_comend

Outline	Waits for completion of IICA transfer.	
Header	r_cg_macrodriver.h, r_iica_user.h	
Declaration	MD_STATUS R_IICA_wait_comend(void);	
Explanation	Checks the IICA1 status and waits for completion of transfer (communication).	
Arguments	None	
Return value	Communication result status:	
	MD_OK	Communication successfully completed
	MD_ERROR3	No ACK response from the slave
Remarks	None	

[Function Name] R_IICA_StopCondition

Outline	Issues the stop condition from IICA to the I2C bus.
Header	r_cg_macrodriver.h, r_iica_user.h
Declaration	void R_IICA_StopCondition(void);
Explanation	Issues the stop condition from IICA to the I2C bus and waits for the stop condition to be detected.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IICA_bus_check

Outline	Checks the I2C bus state.	
Header	r_cg_macrodriver.h, r_iica_user.h	
Declaration	MD_STATUS R_IICA_bus_check(void);	
Explanation	Checks the I2C bus state and issues the start condition if the bus is available. Successfully ends if the start condition is detected by the 100th checking.	
Arguments	None	
Return value	Transmission status:	
	MD_OK	Successfully started.
	MD_ERROR1	I2C bus is currently used.
Remarks	None	

[Function Name] r_iica_interrupt

Outline	Processes the IICA1 transfer end interrupt.
Header	r_cg_macrodriver.h, r_iica_user.h
Declaration	#pragma interrupt r_iica_interrupt(vect=INTIICA1,BANK=rb1) static void r_iica_interrupt(void)
Explanation	Processes the IICA1 transfer end interrupt. When transfer of the specified data is completed, sets the pertinent status to “communication successfully completed”.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intp0_interrupt

Outline	Processes the INTP0 interrupt.
Header	r_cg_macrodriver.h, r_timer_user.h
Declaration	#pragma interrupt r_intp0_interrupt(vect=INTP0) static void __near r_intp0_interrupt(void)
Explanation	Initiated when SW is pressed; activates the 8-bit interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_it_interrupt

Outline	Processes the interval timer interrupt.
Header	r_cg_macrodriver.h, r_timer_user.h
Declaration	#pragma interrupt r_it_interrupt(vect=INTIT00) static void __near r_it_interrupt(void)
Explanation	Initiated by the INTIT00 every second; measures the internal reference voltage by using the A/D, sets DTC and TM01, and activates the A/D hardware trigger every millisecond.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_ADC_inter_ref

Outline	Measures the internal reference voltage.
Header	r_cg_macrodriver.h, r_timer_user.h, r_cg_adc.h
Declaration	void R_ADC_inter_ref(void);
Explanation	Measures the internal reference voltage, stores it to variable g_vref, sets the measurement range to 2 (100 V), and initializes the parameters for measurement.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_ADC_DTC_Init

Outline	Makes the initial settings for A/D and DTC.
Header	r_cg_macrodriver.h, r_timer_user.h, r_cg_adc.h, r_cg_dtc.h
Declaration	void R_ADC_DTC_Init(void);
Explanation	Sets ADC to hardware trigger and single scan mode of ANI3-ANI0, and sets the parameters for DTC transfer again.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_tm01_start

Outline	Activates TM01.
Header	r_cg_macrodriver.h, r_timer_user.h
Declaration	void r_tm01_start(void);
Explanation	Activates TM01 for A/D conversion trigger with interrupt disabled.
Arguments	None
Return value	None
Remarks	None

[Function Name] delay_1u

Outline	Waits for 1 us.
Header	r_cg_macrodriver.h, r_timer_user.h
Declaration	void delay_1us(void);
Explanation	Waits for 1 us after ADCE setting until ADCS setting.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_adc_interrupt

Outline	Processes the INTAD interrupt.
Header	r_cg_macrodriver.h, r_timer_user.h
Declaration	#pragma interrupt r_adc_interrupt(vect=INTAD) static void __near r_adc_interrupt(void)
Explanation	Initiated by the DTC transfer end interrupt of 4-channel A/D conversion data. Performs the appropriate operation on the A/D converted data according to the measurement range, and records the maximum value and sign change points of the result. After the 32nd measurement, stops TM01 and completes A/D conversion. Before the 32nd measurement, sets the parameters for DTC transfer again.
Arguments	None
Return value	None
Remarks	None

[Function Name] __ssqrt

Outline	Performs square-root operation of 32-bit data.	
Header	arith_lib.h	
Declaration	uint16_t _sqrt(uint32_t);	
Explanation	Calculates the square root of the 32-bit data passed with the argument.	
Arguments	32-bit data	BC: Upper 16 bits, AX: lower 16 bits
Return value	Square root	AX (= BC)
Remarks	For details, refer to R01AN3079.	

[Function Name] __r_mul32

Outline	16 bits × 16 bits	
Header	r_cg_macrodriver.h, arith_lib.h	
Declaration	uint32_t _r_mul32(uint16_t multiplicand, uint16_t multiplier);	
Explanation	Performs multiplication on two 16-bit arguments.	
Arguments	First argument	Multiplicand (AX)
	Second argument	Multiplier (BC)
Return value	Product	（BC-AX）
Remarks	None	

[Function Name] __r_div32

Outline	32 bits/16 bits	
Header	r_cg_macrodriver.h, arith_lib.h	
Declaration	uint32_t _r_div32(uint32_t dividend, uint16_t divisor);	
Explanation	Divides the first argument (32 bits) by the second argument (16 bits).	
Arguments	First argument	Dividend (BC-AX)
	Second argument	Divisor (DE)
Return value	Quotient	(BC-AX)
Remarks	None	

[Function Name] __r_div16

Outline	16 bits/8 bits	
Header	r_cg_macrodriver.h, arith_lib.h	
Declaration	uint16_t _r_div16(uint16_t dividend, uint8_t divisor);	
Explanation	Divides the first argument (16 bits) by the second argument (8 bits).	
Arguments	First argument	Dividend (AX)
	Second argument	Divisor (C)
Return value	Quotient	AX
Remarks	None	

[Function Name] __r_mod16

Outline	Modulo operation of 16 bits/8 bits	
Header	r_cg_macrodriver.h, arith_lib.h	
Declaration	uint8_t _r_mod16(uint16_t dividend, uint8_t divisor);	
Explanation	Finds the modulo after dividing the first argument (16 bits) by the second argument (8 bits).	
Arguments	First argument	Dividend (AX)
	Second argument	Divisor (C)
Return value	Modulo	A
Remarks	None	

[Function Name] __r_rms16

Outline	Performs root-mean-square operation of 16-bit data.	
Header	r_cg_macrodriver.h, arith_lib.h	
Declaration	uint16_t _r_rms16(uint8_t start, uint18_t data_no);	
Explanation	Calculates the root-mean-square of as many data as the number specified with the second argument, starting at the data specified with the first argument of the expected result (_g_vin_data[]).	
Arguments	First argument	Starting data (A)
	Second argument	Number of data (X)
Return value	Root-mean-square	AX
Remarks	None	

5.7 Flowchart

Figure 5.1 shows the overall flow of the process described in this application note.Figure 1.1 Analog Voltage Input Circuit

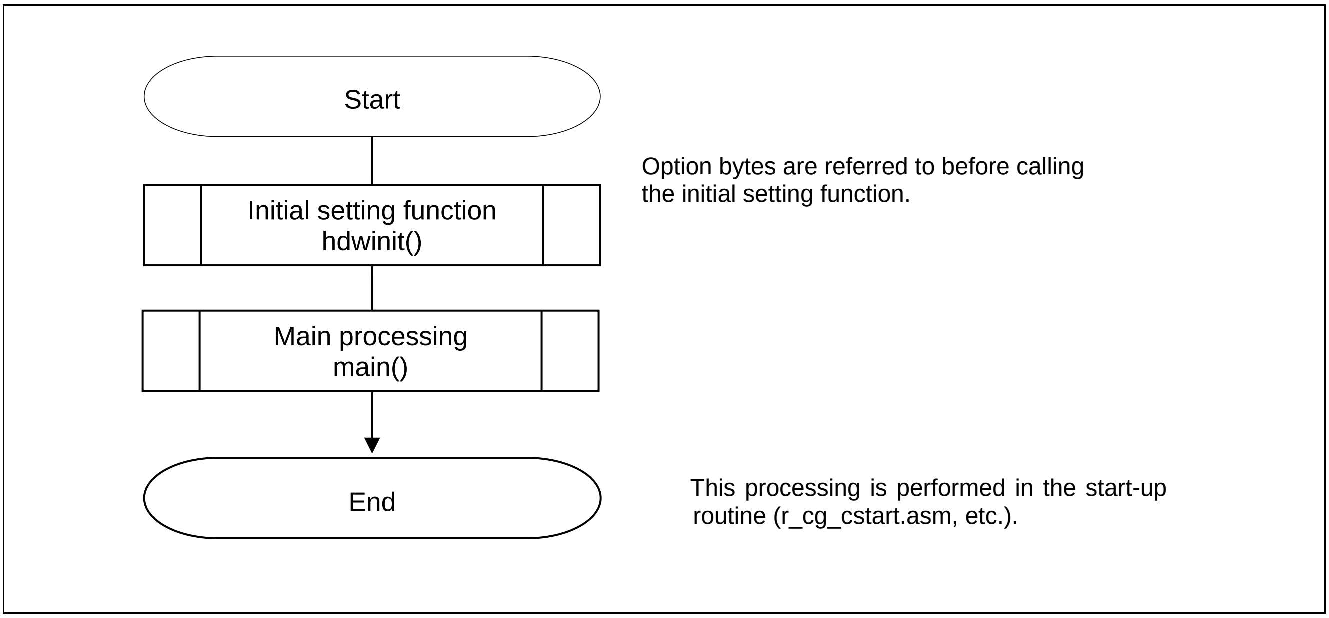


Figure 5.1 Overall Flow

Note: This processing is performed in the start-up routines (r_cg_cstart.asm, etc.). The setting for the memory is performed between calling the initial setting function and the main processing function.

5.7.1 Initial Setting Function

Figure 5.2 shows the flowchart of the initial setting function.

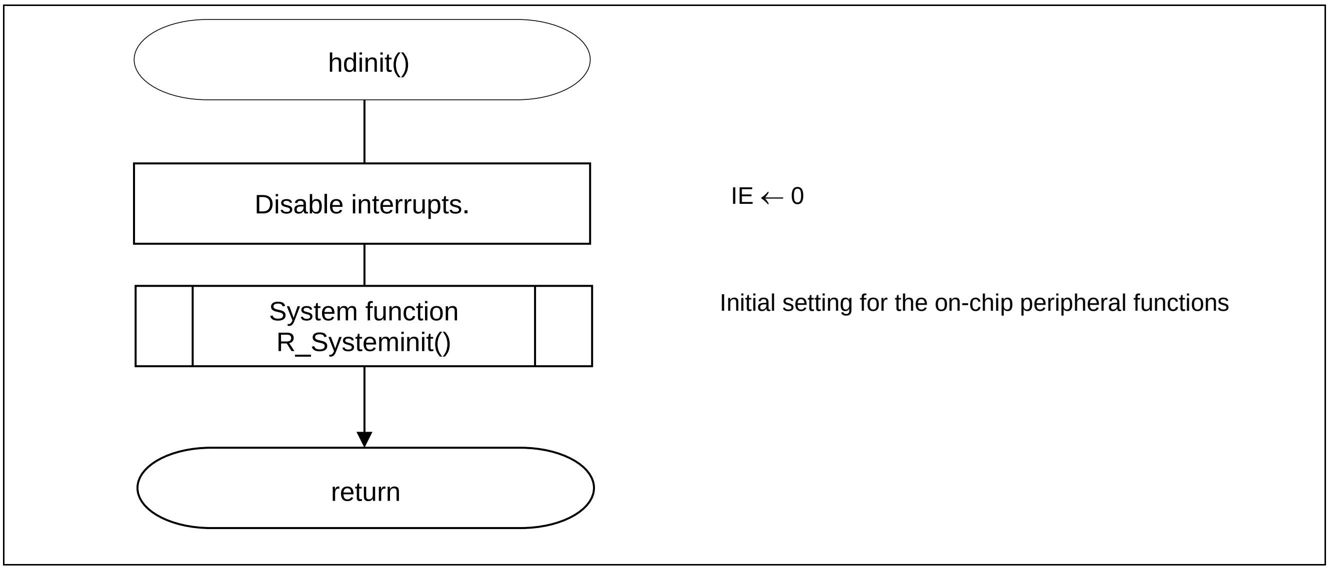


Figure 5.2 Initial Setting Function

5.7.2 System Function

Figure 5.3 shows the flowchart of the system function.

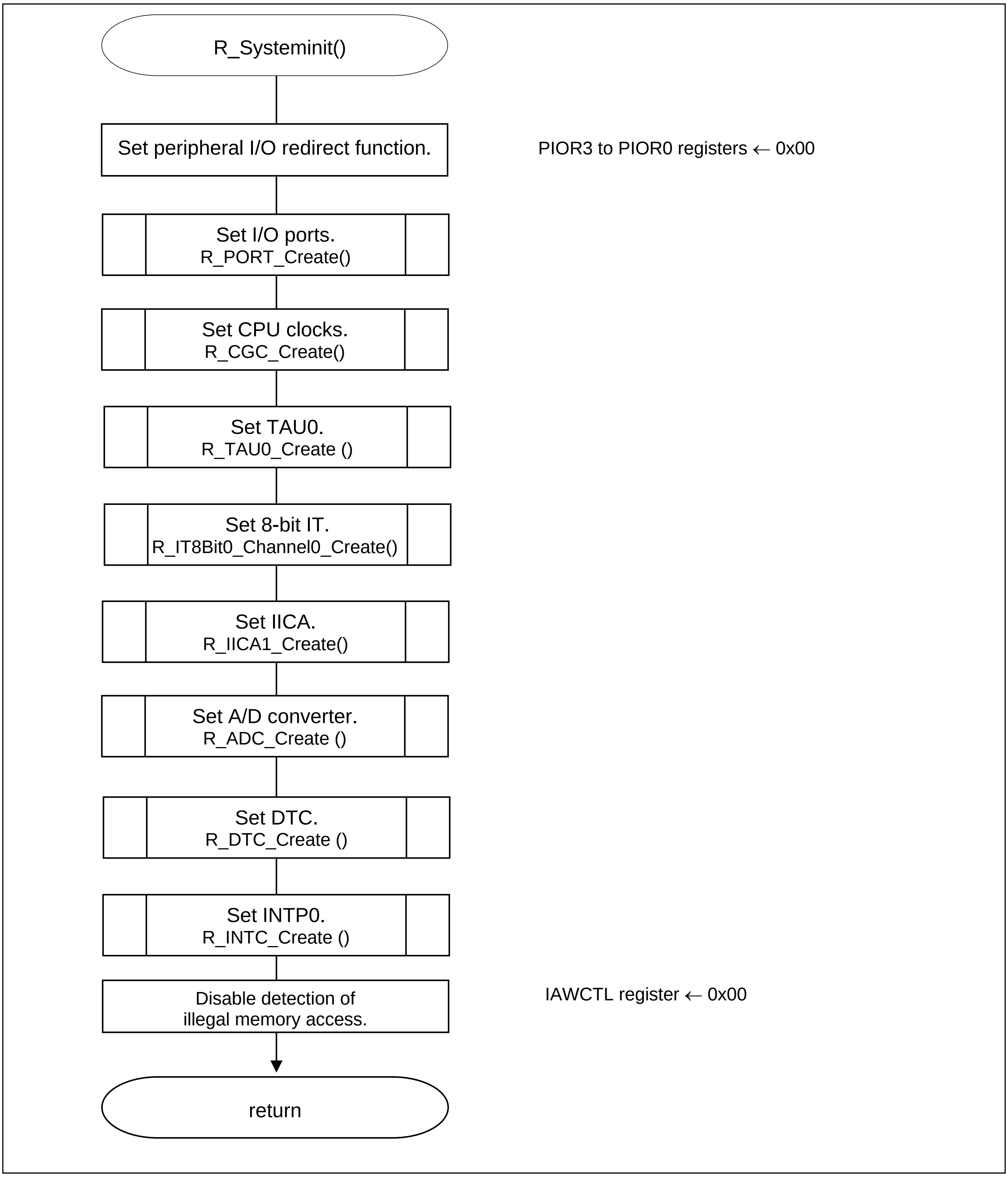


Figure 5.3 System Function

5.7.3 Setting I/O Ports

Figure 5.4 shows the flowchart for setting the I/O ports.

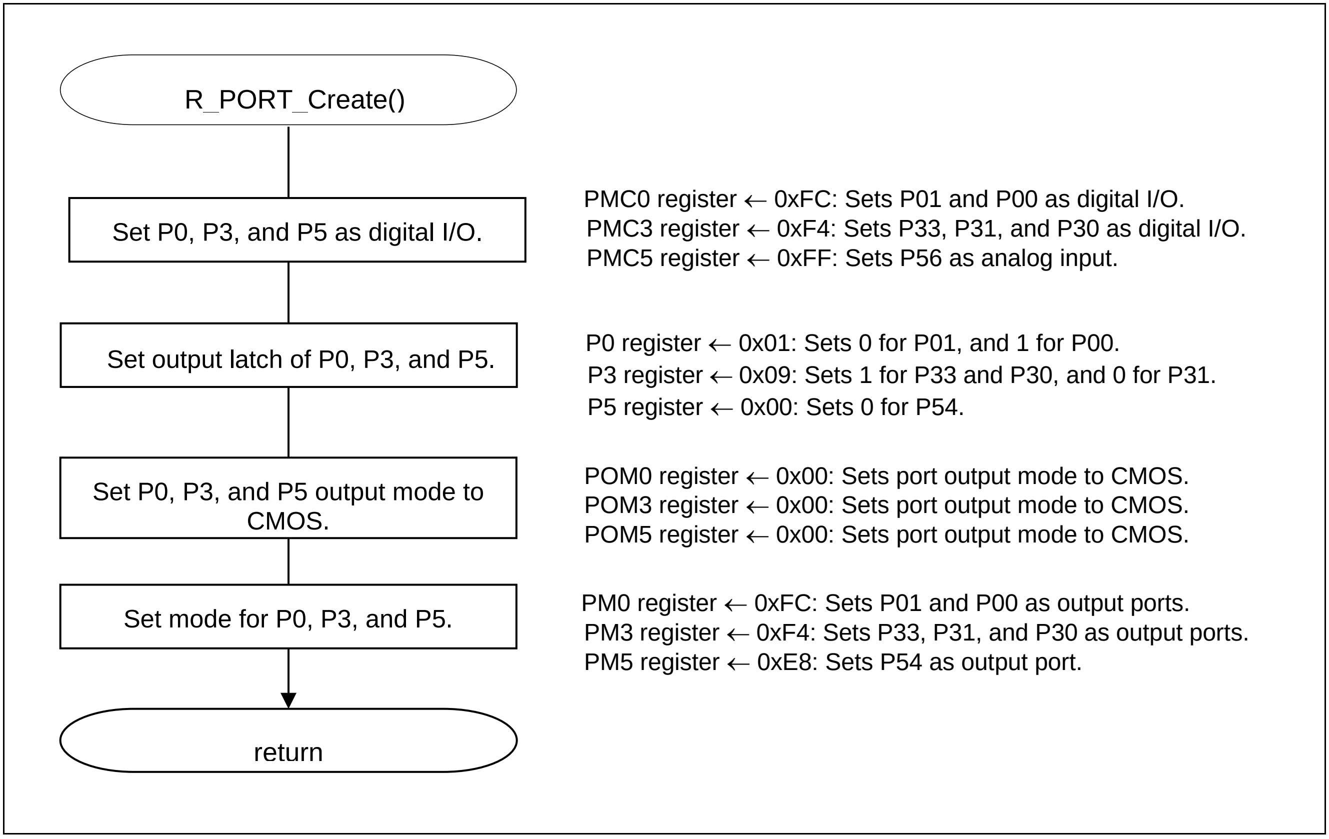


Figure 5.4 Setting I/O Ports

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 separate resistors.

5.7.4 Setting CPU Clocks

Figure 5.5 shows the flowchart for setting the CPU clocks.

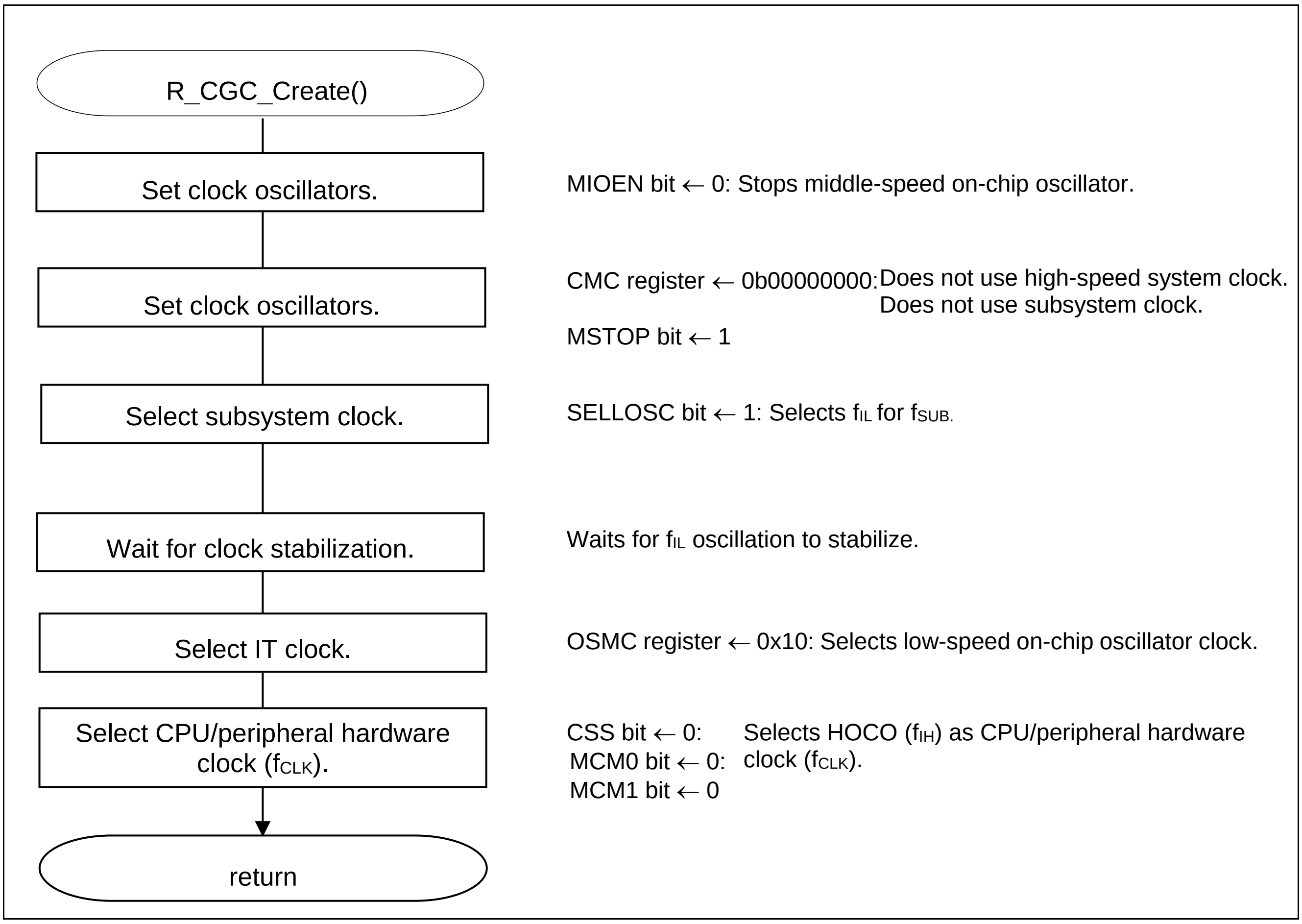


Figure 5.5 Setting CPU Clocks

5.7.5 Setting Timer Array Unit

Figure 5.6 shows the flowchart for setting the timer array unit.

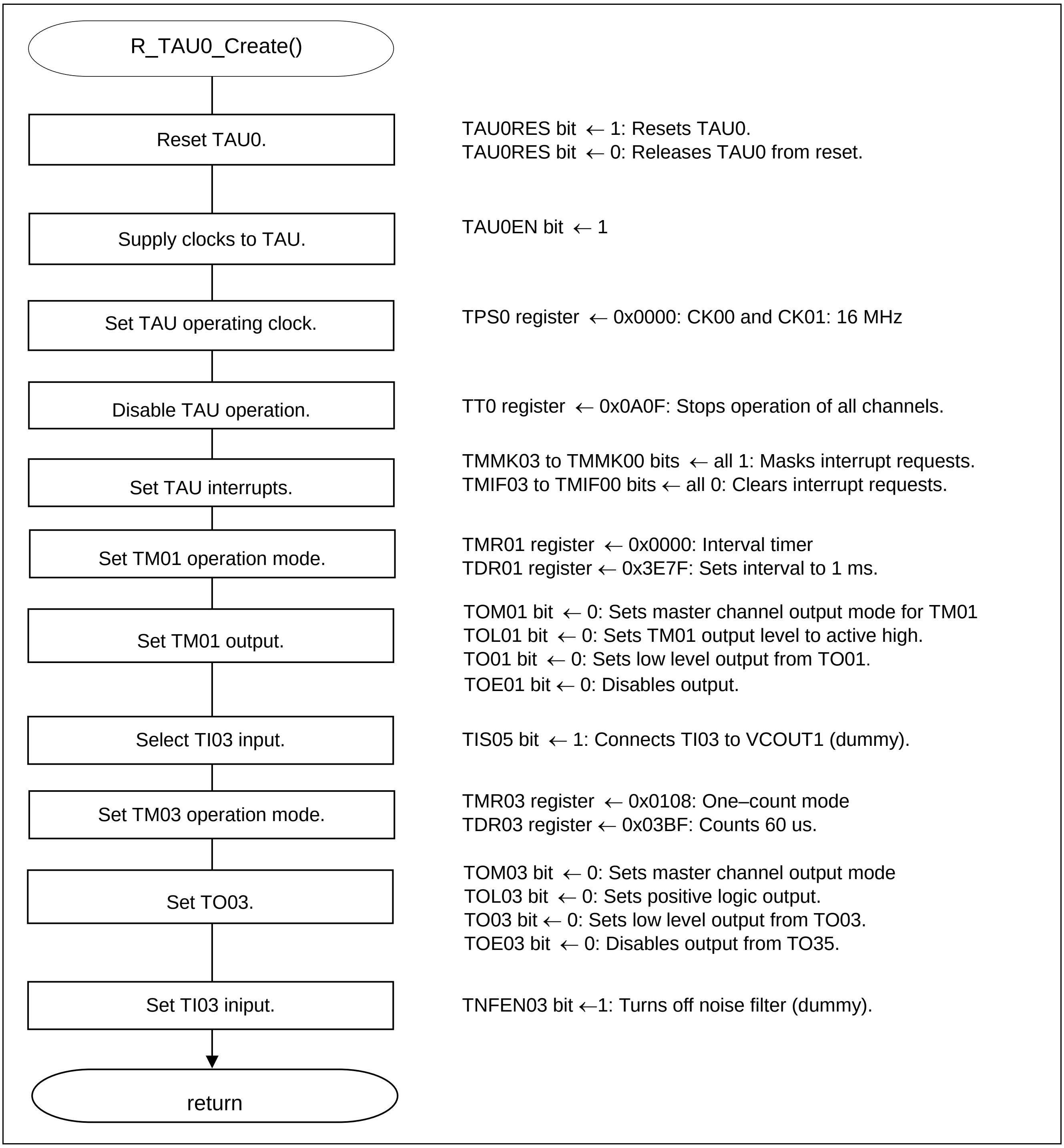


Figure 5.6 Setting Timer Array Unit

5.7.6 Setting 8-Bit Interval Timer

Figure 5.7 shows the flowchart for setting the 8-bit interval timer.

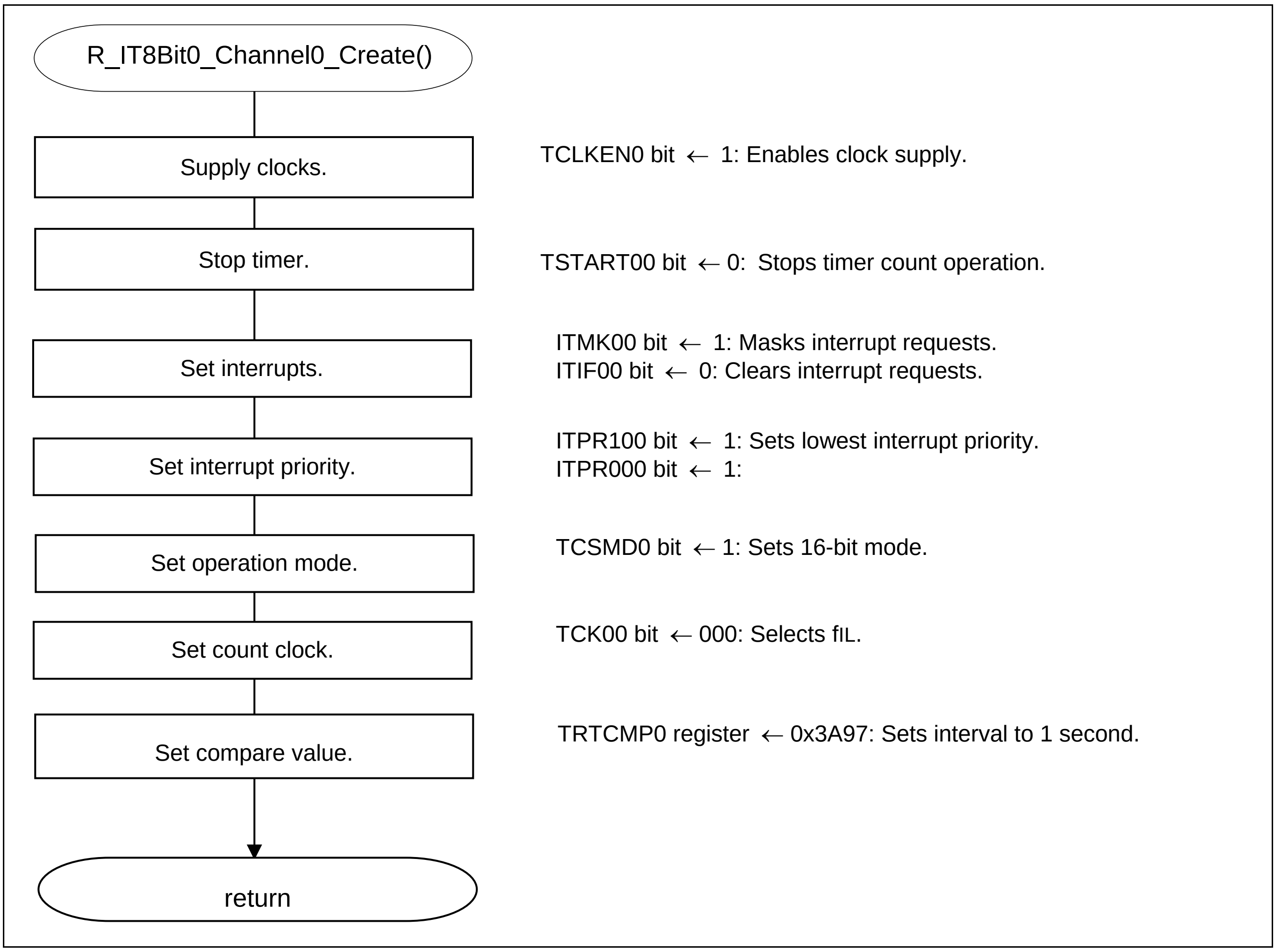


Figure 5.7 Setting 8-Bit Interval Timer

5.7.7 Setting IICA

Figure 5.8 shows the flowchart for setting IICA.

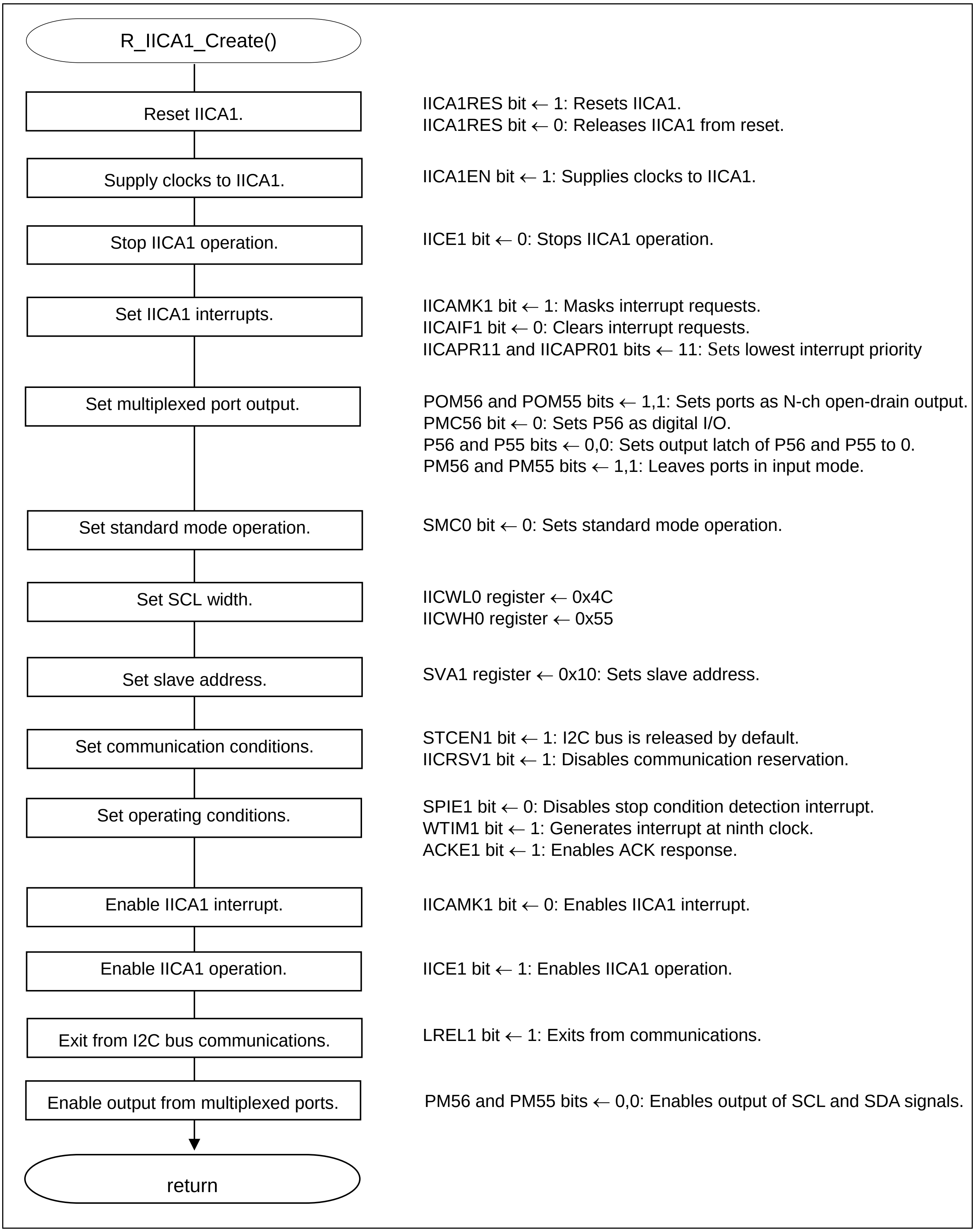


Figure 5.8 Setting IICA

5.7.8 Setting A/D Converter

Figure 5.9 shows the flowchart for setting the A/D converter.

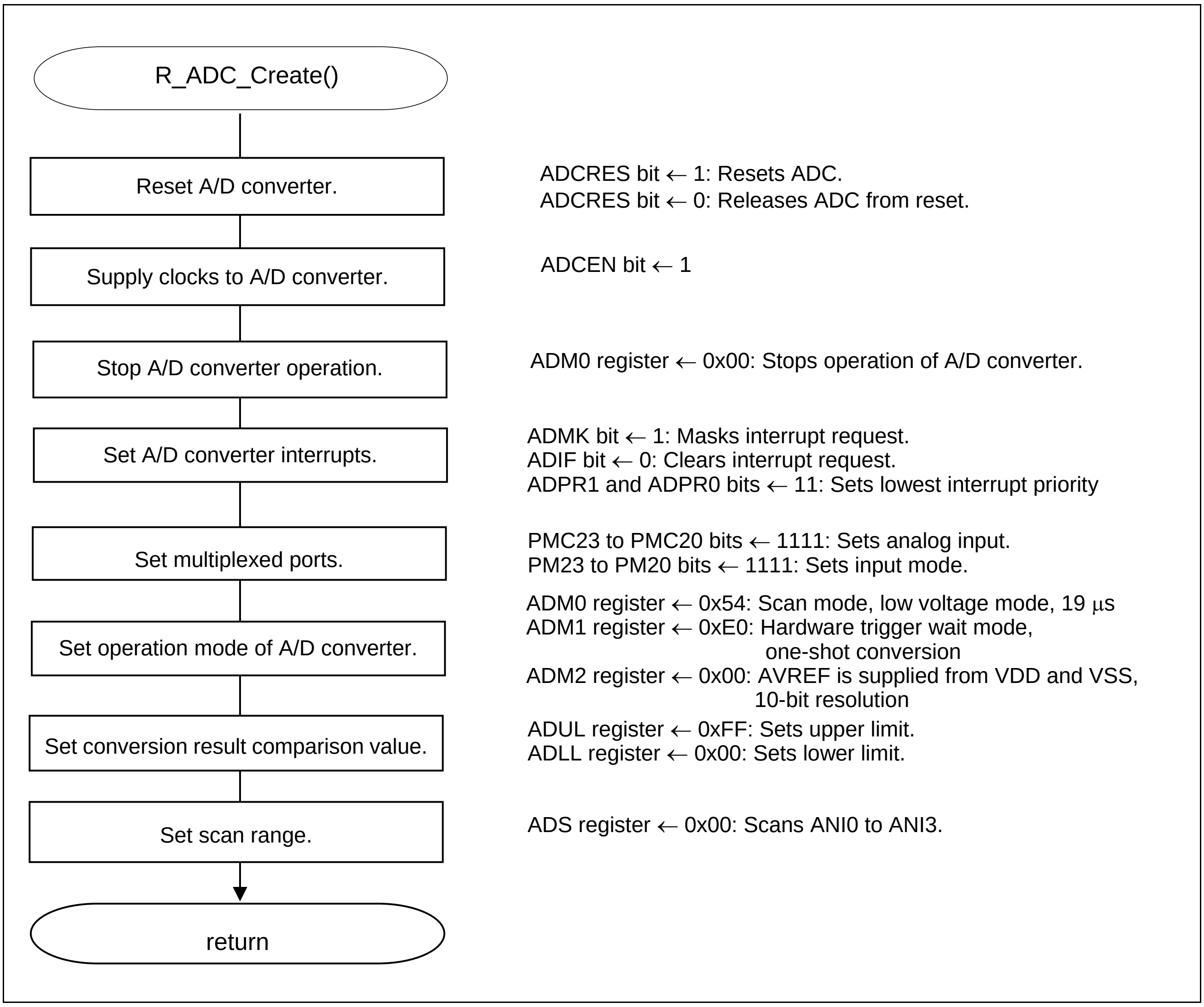


Figure 5.9 Setting A/D Converter

5.7.9 Setting DTC

Figure 5.10 shows the flowchart for setting the DTC.

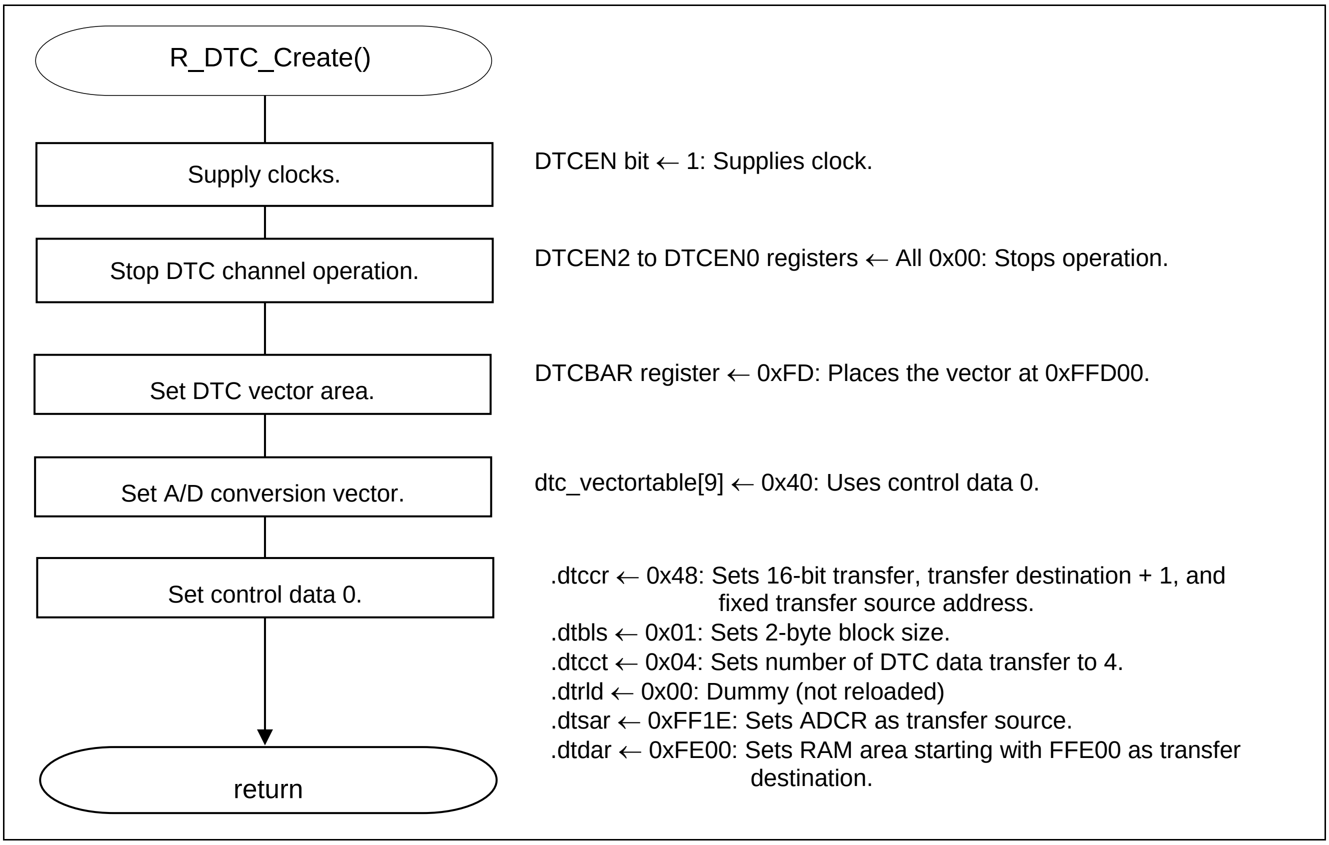


Figure 5.10 Setting DTC

5.7.10 Setting INTP0

Figure 5.11 shows the flowchart for setting INTP0.

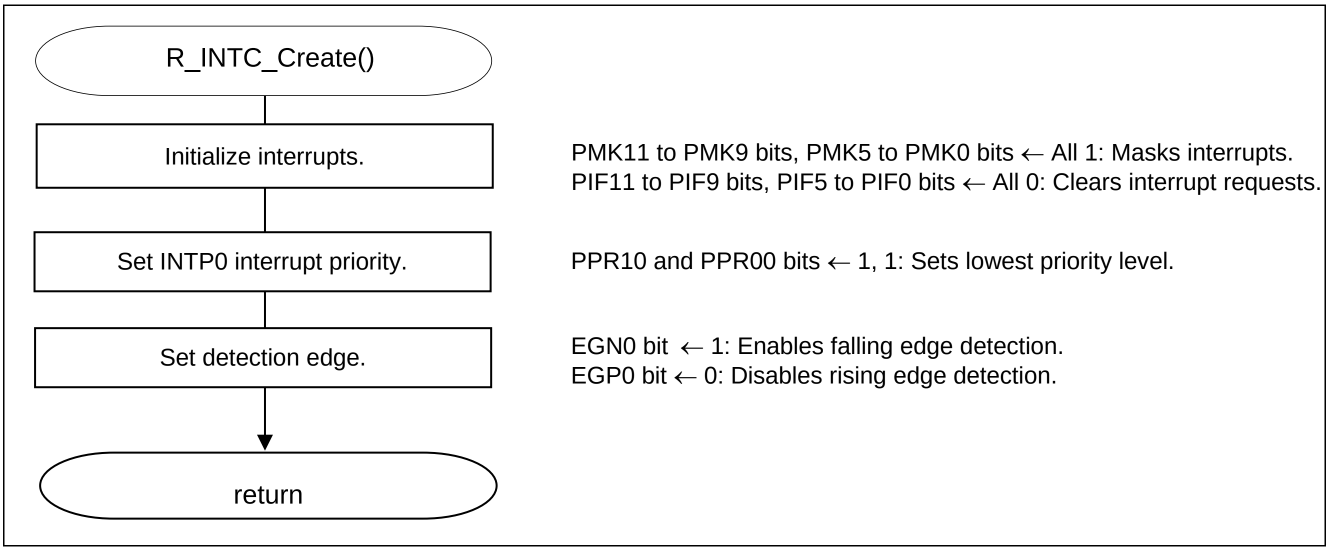


Figure 5.11 Setting INTP0

5.7.11 Main Processing

Figure 5.12 to Figure 5.14 show the flowcharts of the main processing.

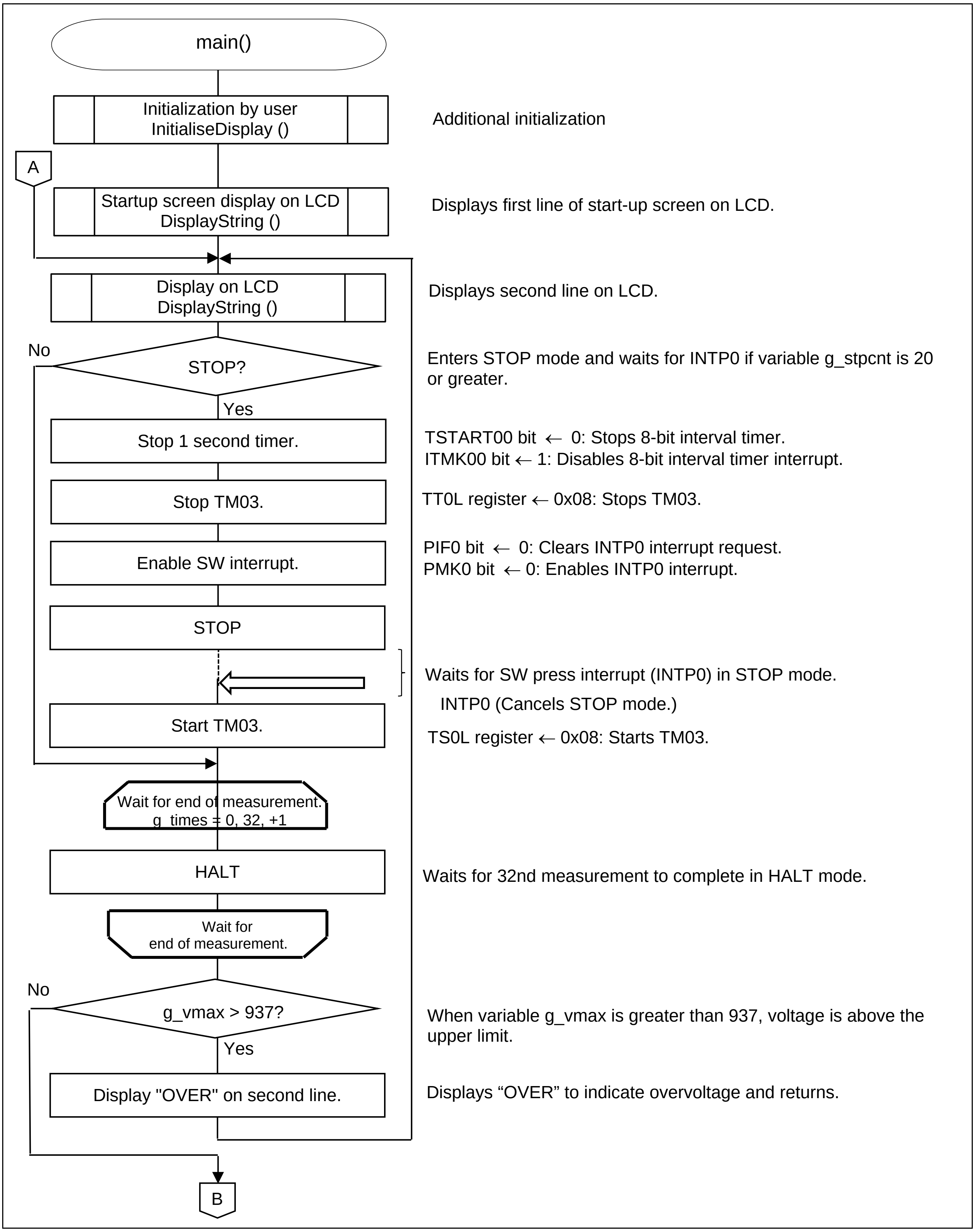
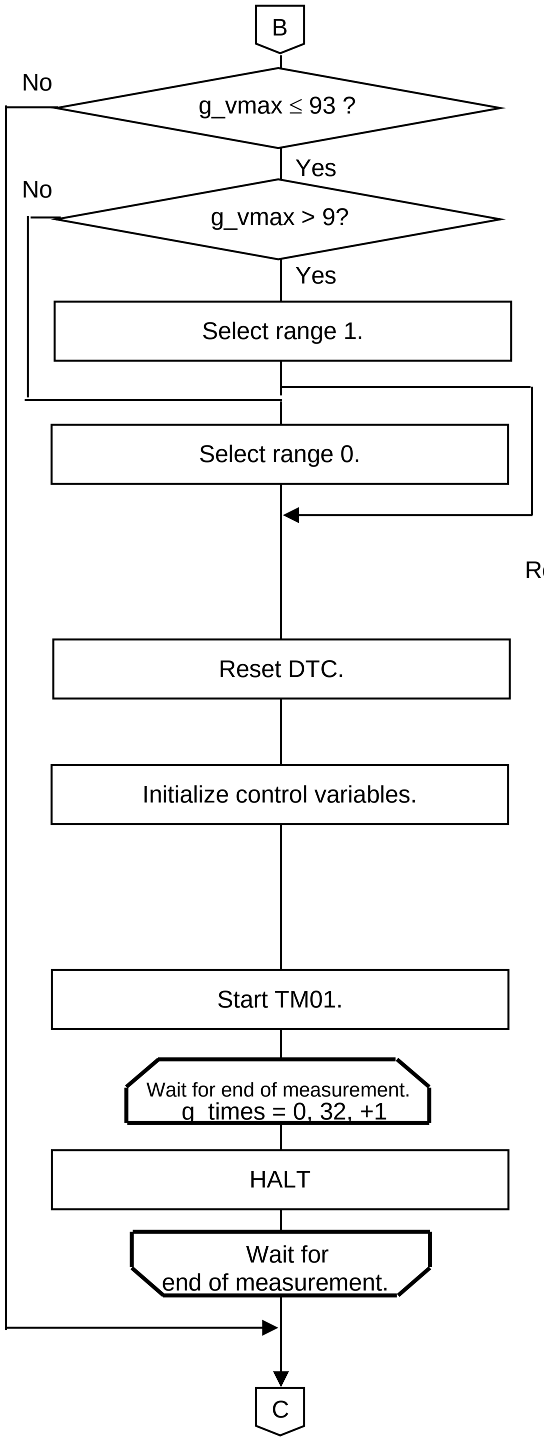


Figure 5.12 Main Processing (1/3)



When variable g_vmax is greater than 93, voltage is measured in 100-V range (range 2).

When variable g_vmax is greater than 9, voltage is measured in 10-V range (range 1). When variable g_vmax is 9 or smaller, it is measured in 1-V range (range 0).

Variable g_range ← 1: Sets range 1 for measurement.
PM01 and PM00 bits ← 0,0: Turns on 1/10 voltage-dividing resistor.
PM31 and PM30 bits ← 1,1: Turns off 1/100 voltage-dividing resistor.

Variable g_range ← 0: Sets range 0 for measurement.

PM31 and PM30 bits ← 1,1: Turns off 1/100 voltage-dividing resistor.

Repeats measurement for range 1 and range 0.

Variable dtc_controldata_0.dtdar ← 0xFE00:
Sets 0xFFE00 as transfer destination address.
Variable dtc_controldata_0.dtcct ← 0x04:
Sets number of transfers to 4.
DTCEN16 bit ← 1: Enables DTC activation on completion of A/D conversion.
Variable g_times ← 0: Sets number of measurements to 0.
Variable g_vin_sign ← 0x00: Initializes signs.
Variable g_sign_chng1 ← 0x00: Clears zero cross point.
Variable g_sign_chng2 ← 0x00: Clears zero cross point.
Variable g_sign_chng3 ← 0x00: Clears zero cross point.
Variable g_vmax ← 0x00: Clears maximum value.

Starts TM01 (A/D trigger) to perform A/D conversion in target range.

Waits for 32nd measurement to complete in HALT mode.

End of 32nd measurement in target range

Figure 5.13 Main Processing (2/3)

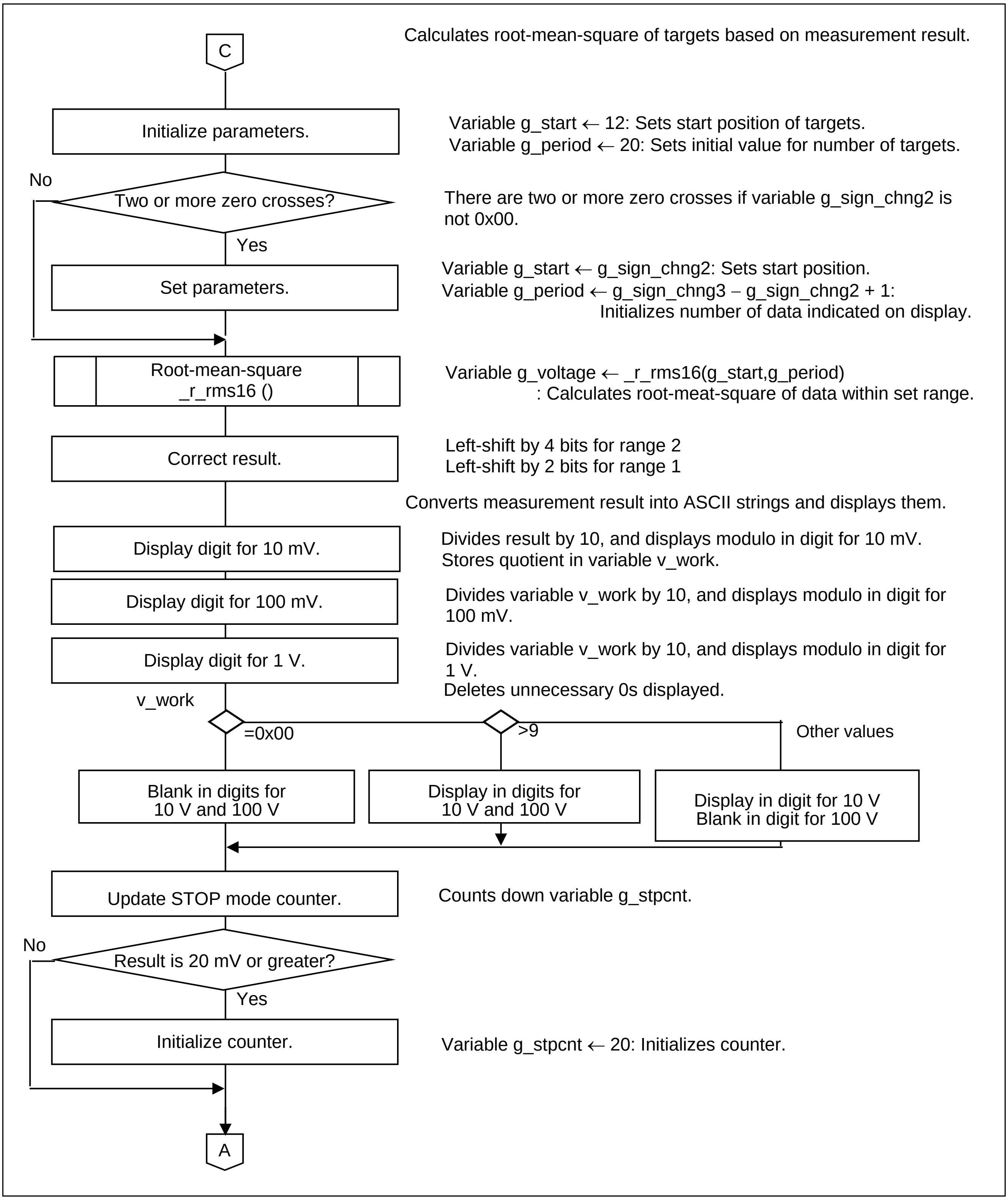


Figure 5.14 Main Processing (3/3)

5.7.12 R_MAIN_UserInit Processing

Figure 5.15 shows the flowchart of the R_MAIN_UserInit processing.

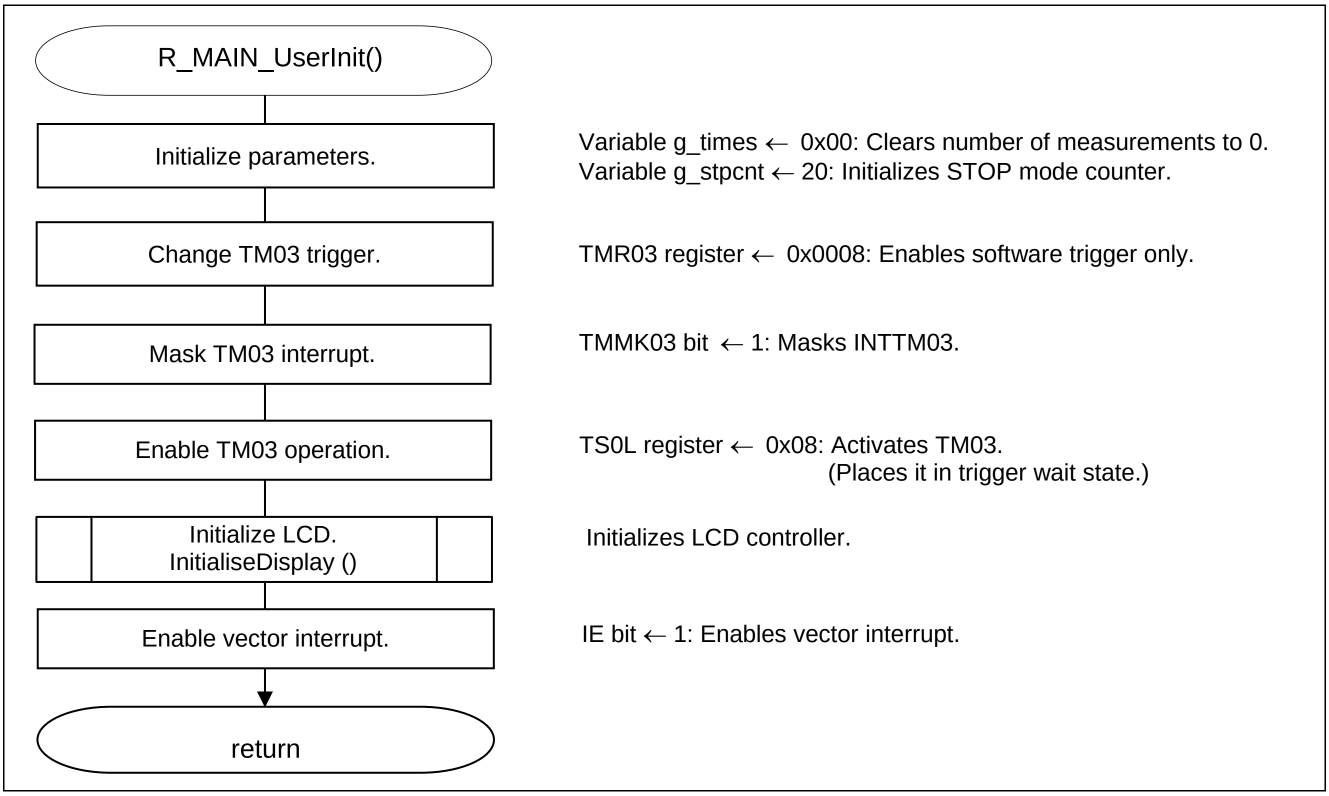


Figure 5.15 R_MAIN_UserInit Processing

5.7.13 Initializing LCD

Figure 5.16 shows the flowchart for initializing the LCD.

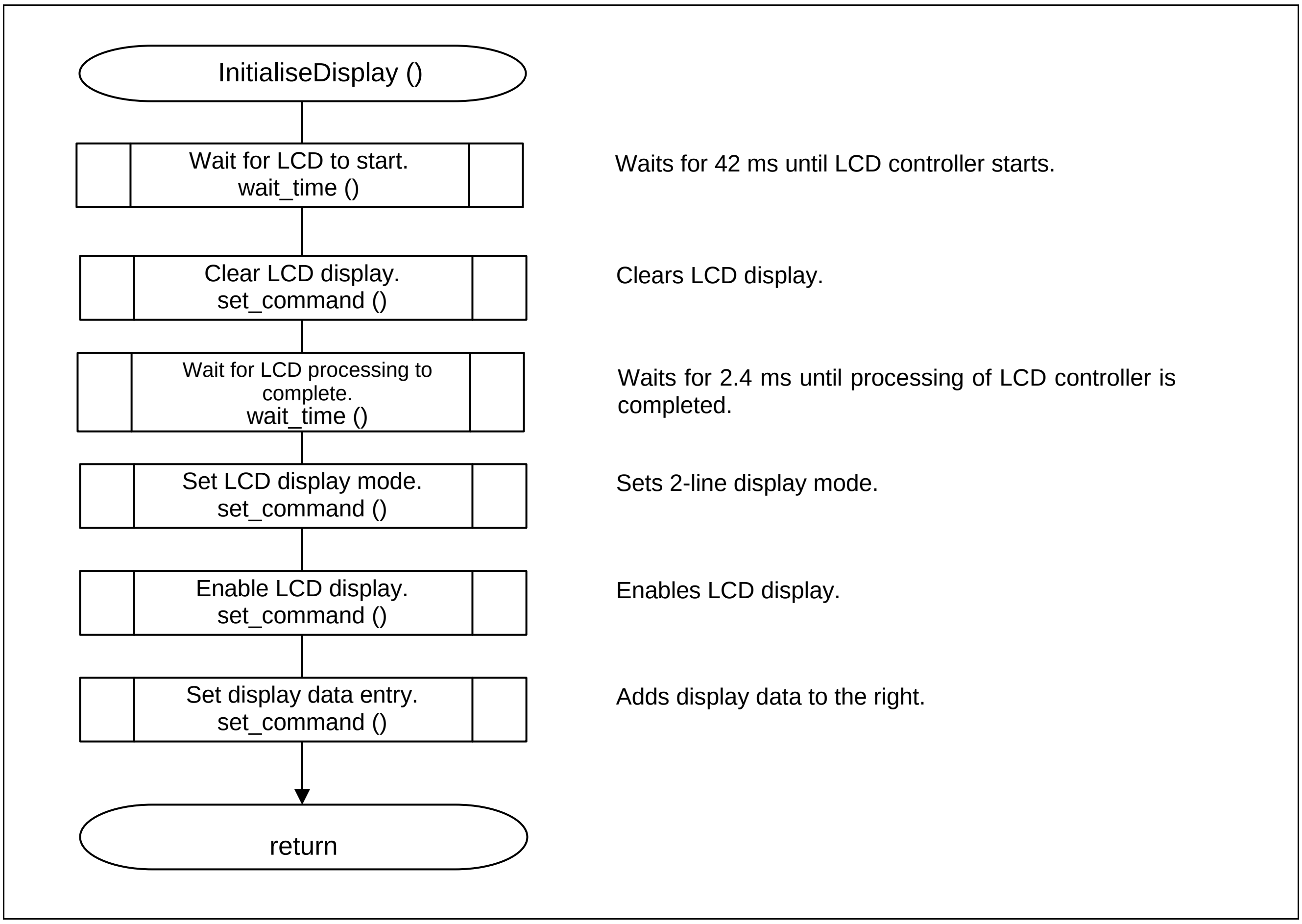


Figure 5.16 Initializing LCD

5.7.14 Setting LCD Commands

Figure 5.17 shows the flowchart for setting the LCD commands.

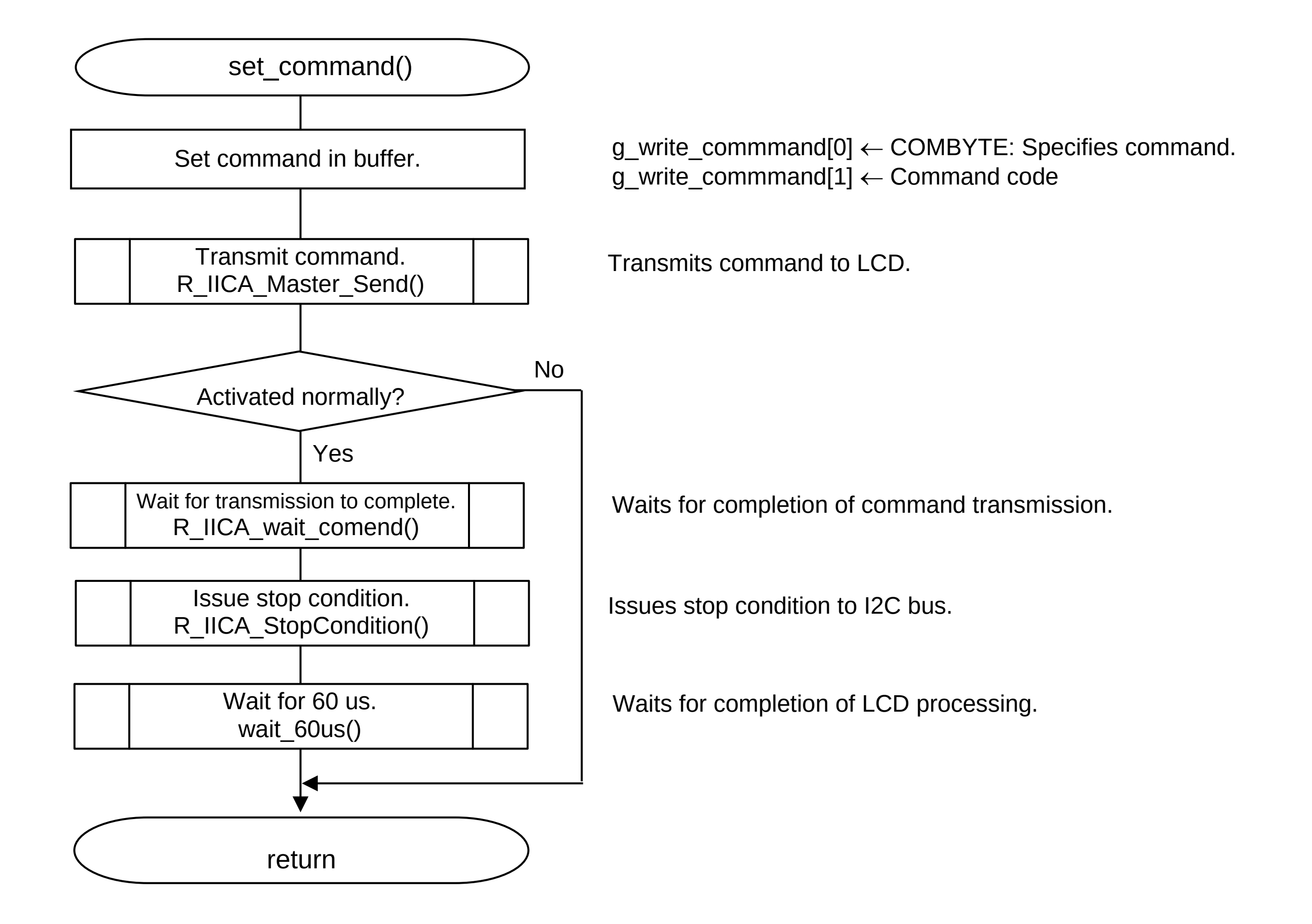


Figure 5.17 Setting LCD Commands

5.7.15 Wait Time Processing

Figure 5.18 shows the flowchart of the wait time processing.

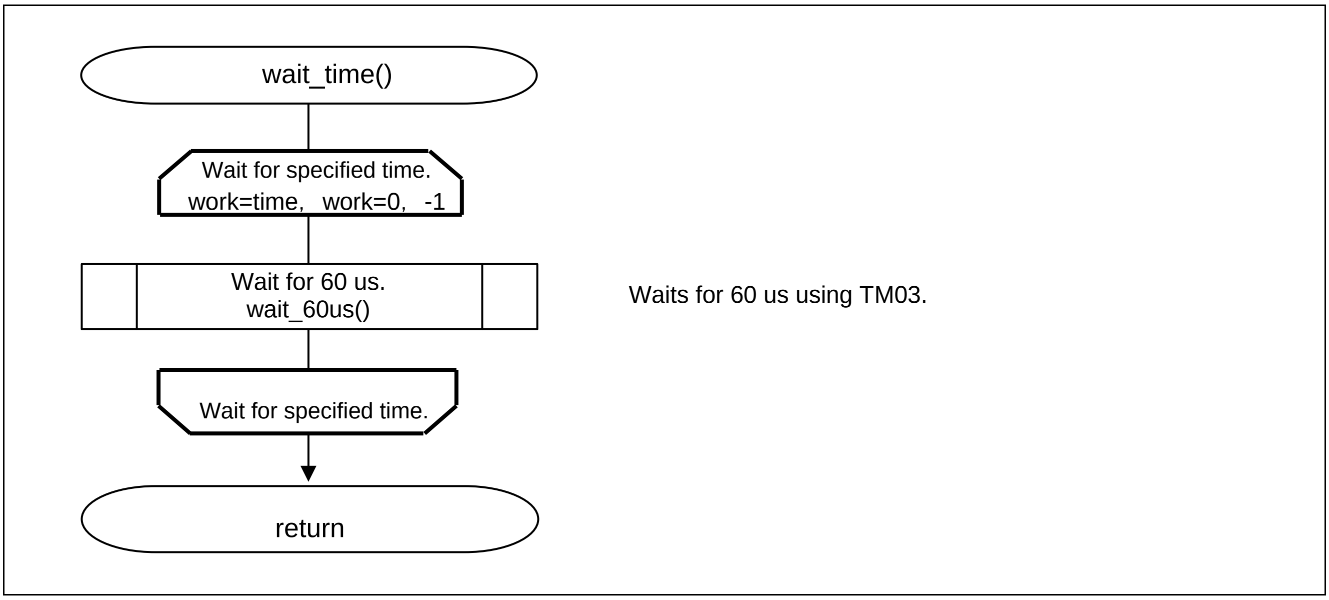


Figure 5.18 Wait Time Processing

5.7.16 60-us Wait Processing

Figure 5.19 shows the flowchart of the 60-us wait processing.

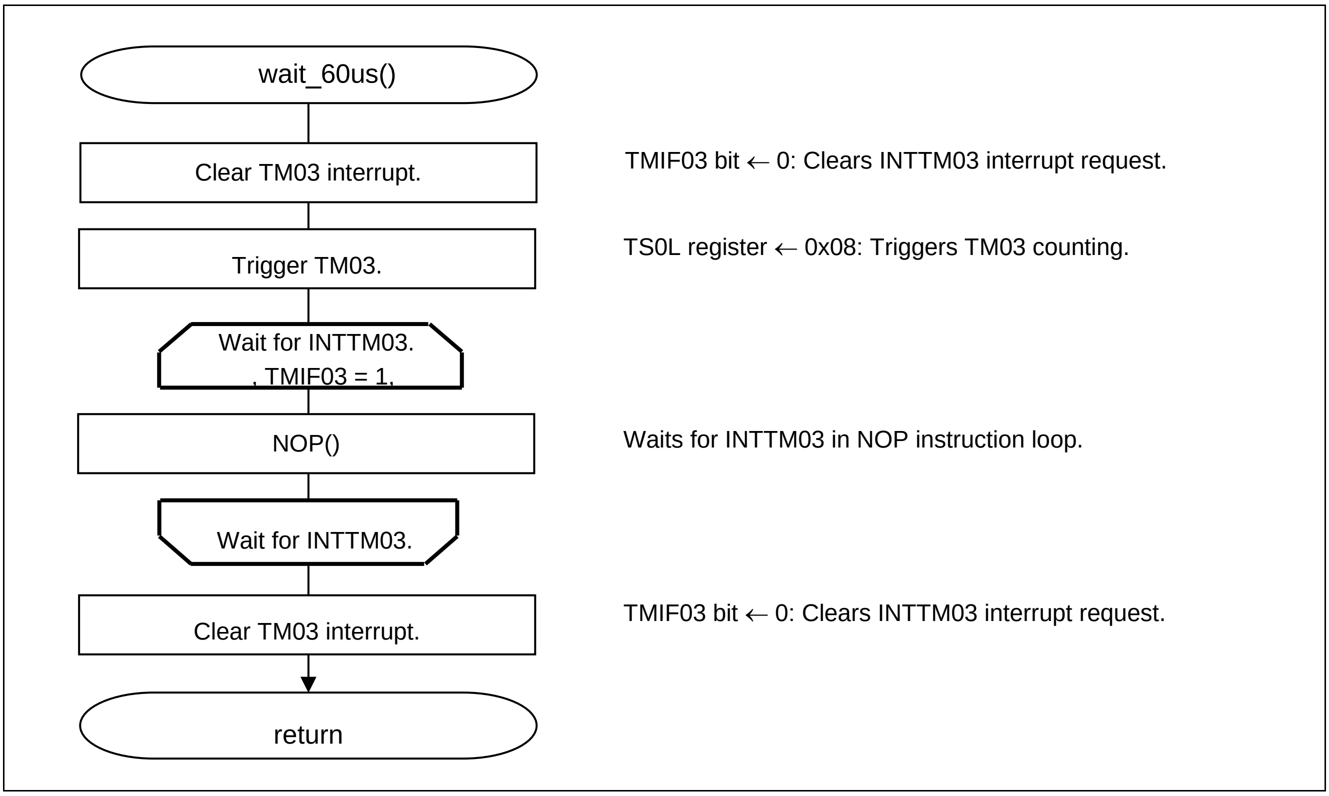


Figure 5.19 60-us Wait Processing

5.7.17 Displaying Character Strings on LCD

Figure 5.20 shows the flowchart for displaying character strings on the LCD.

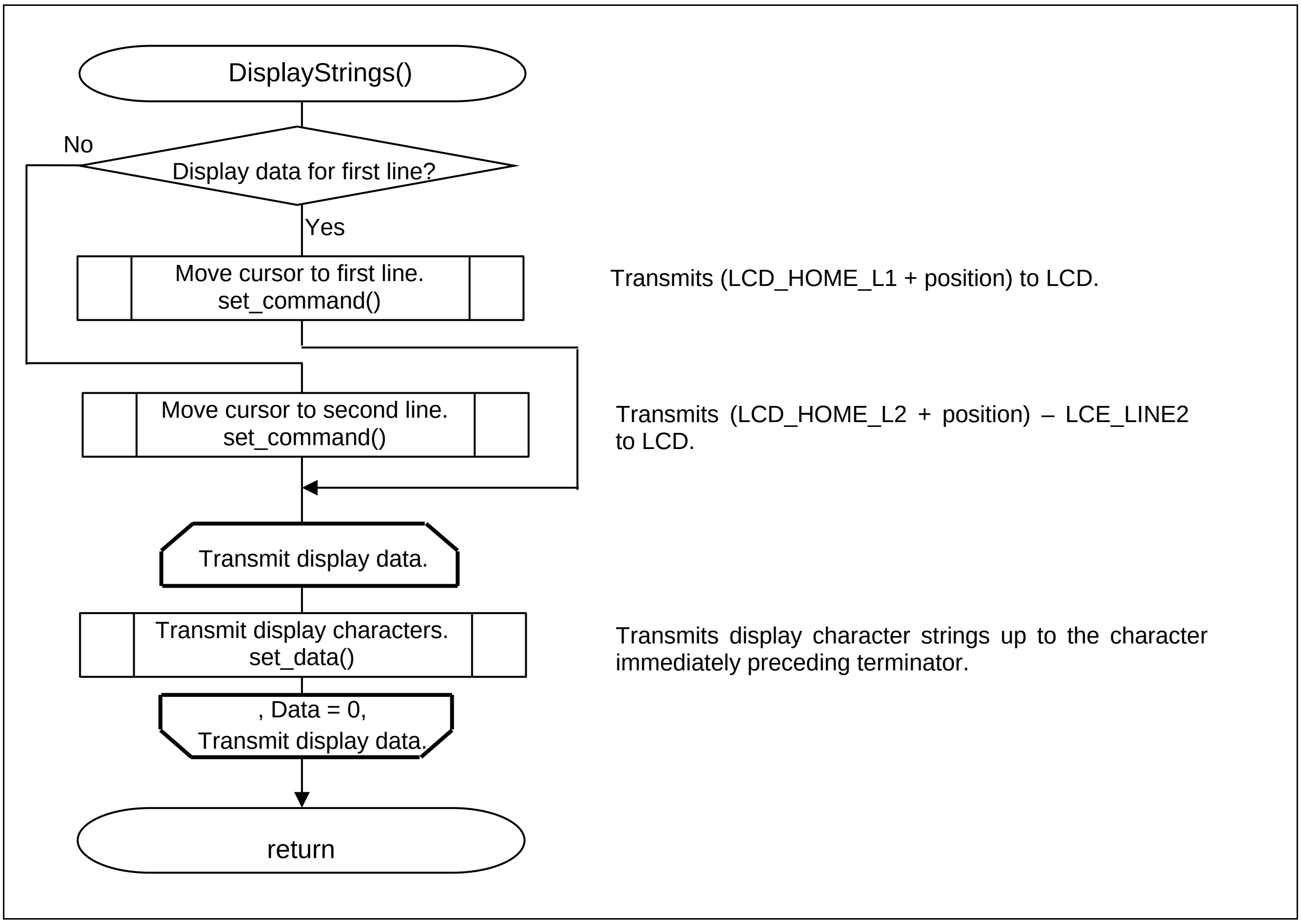


Figure 5.20 Displaying Character Strings on LCD

5.7.18 Setting Display Data

Figure 5.21 shows the flowchart for setting display data.

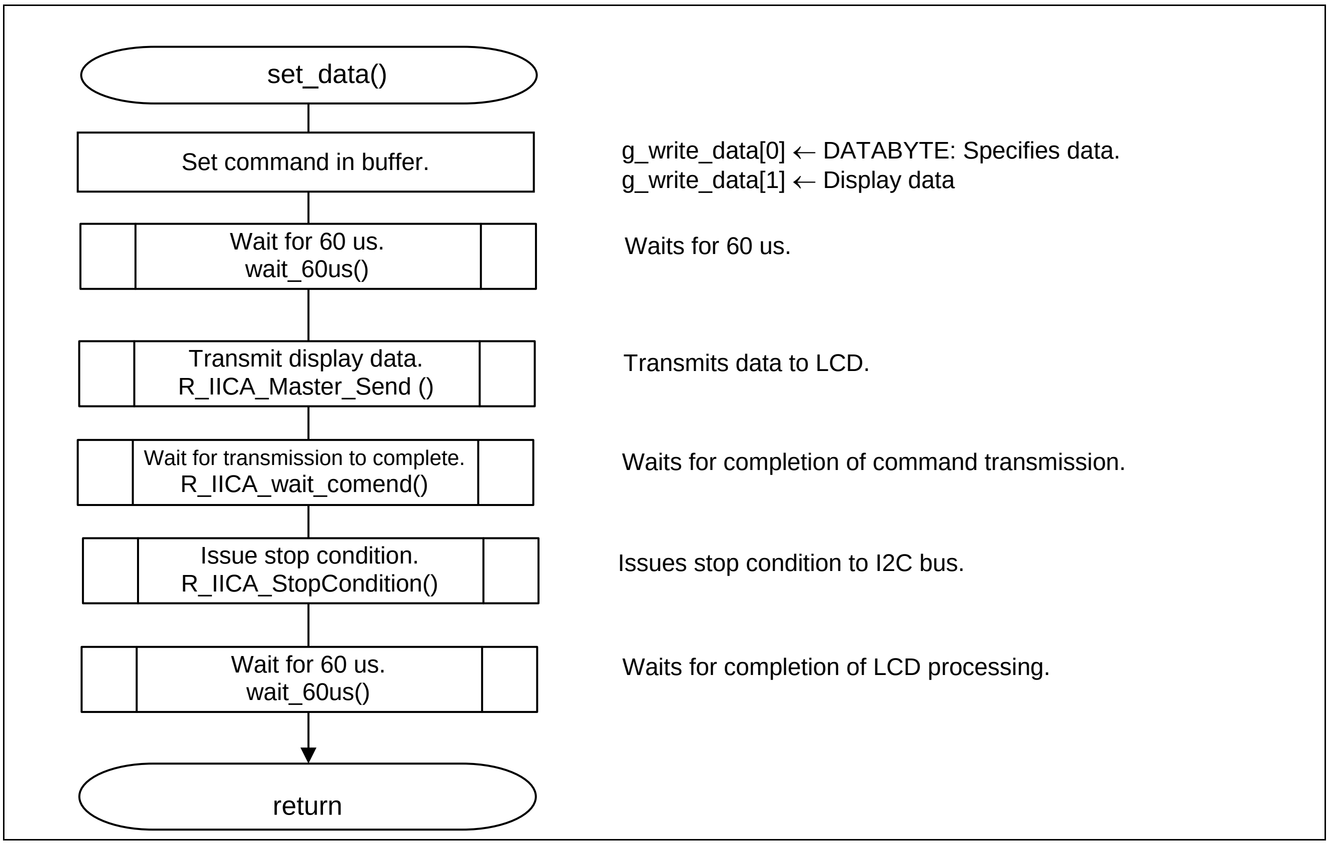


Figure 5.21 Setting Display Data

5.7.19 Starting I2C Transmission

Figure 5.22 shows the flowchart for starting I2C transmission.

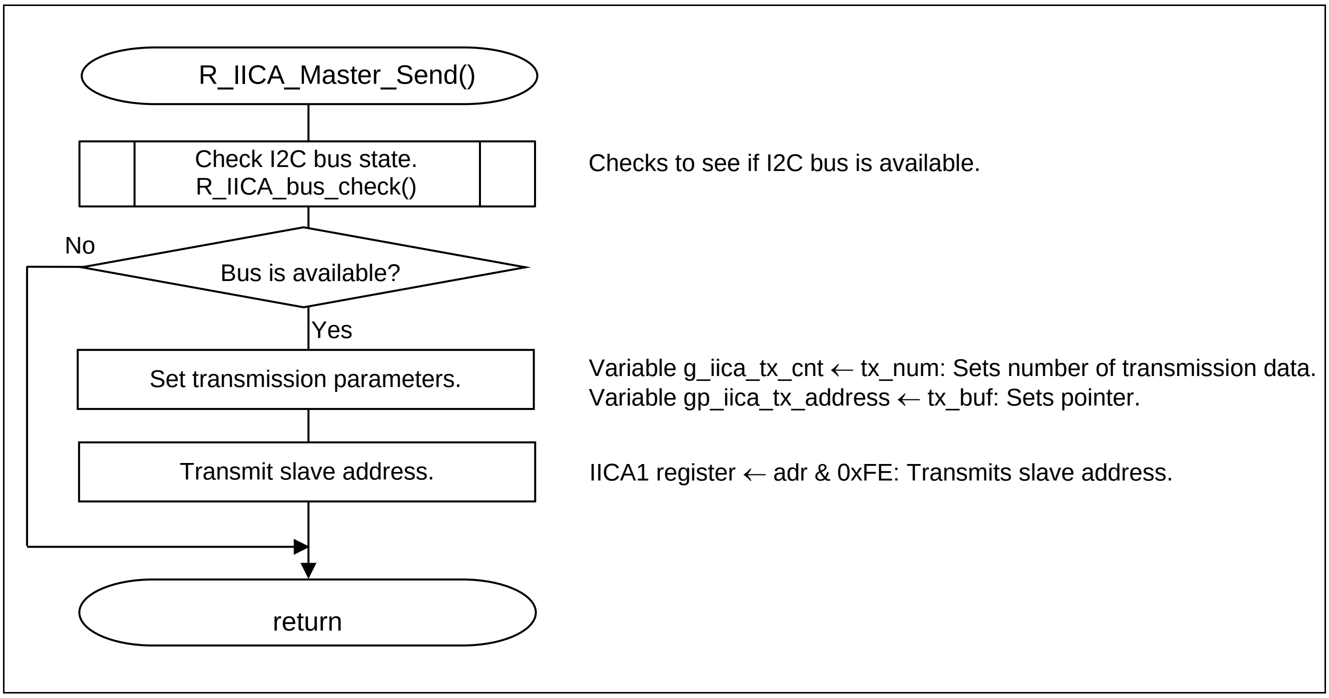


Figure 5.22 Starting I2C Transmission

5.7.20 Waiting for Completion of I2C Communication

Figure 5.23 shows the flowchart for waiting for completion of I2C communication.

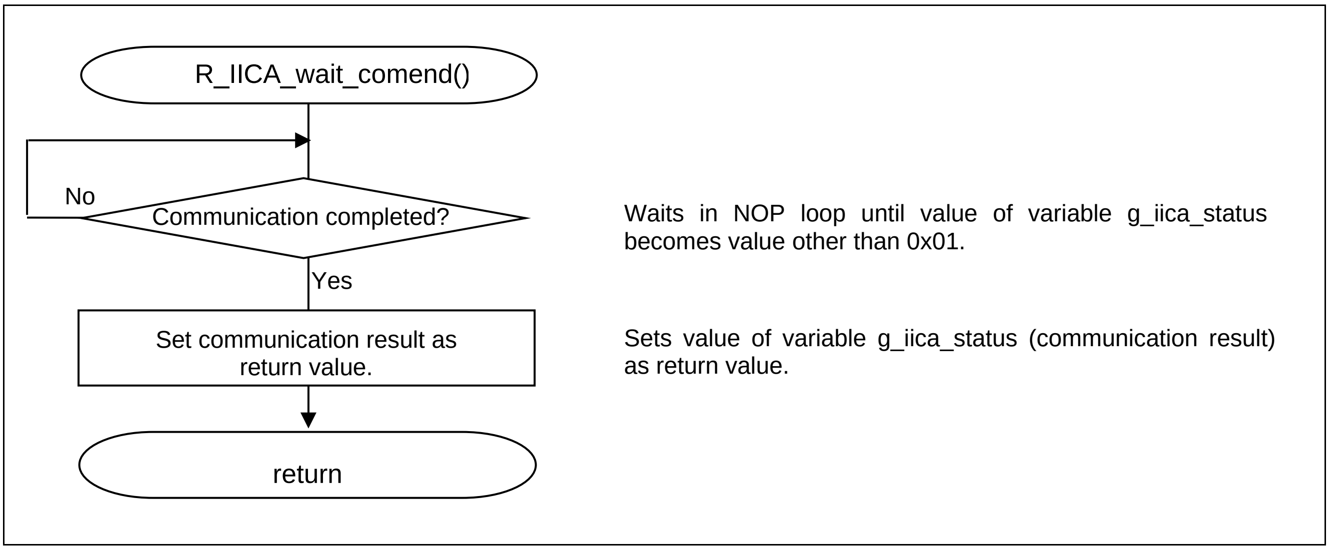


Figure 5.23 Waiting for Completion of I2C Communication

5.7.21 Issuing Stop Condition

Figure 5.24 shows the flowchart for issuing the stop condition.

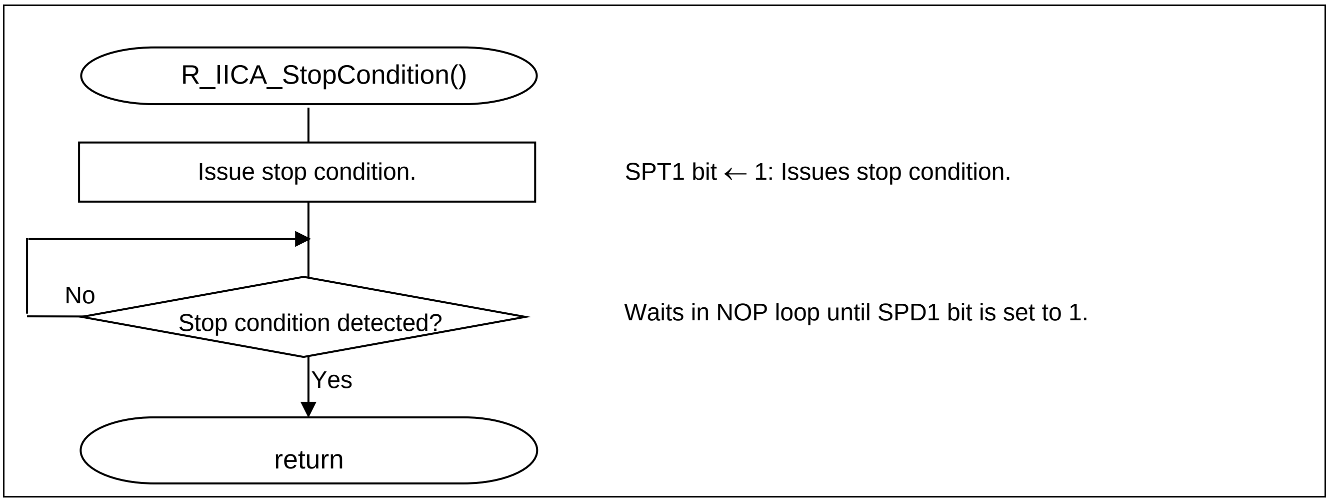


Figure 5.24 Issuing I2C Stop Condition

5.7.22 Checking I2C Bus Status

Figure 5.25 shows the flowchart for checking the I2C bus status.

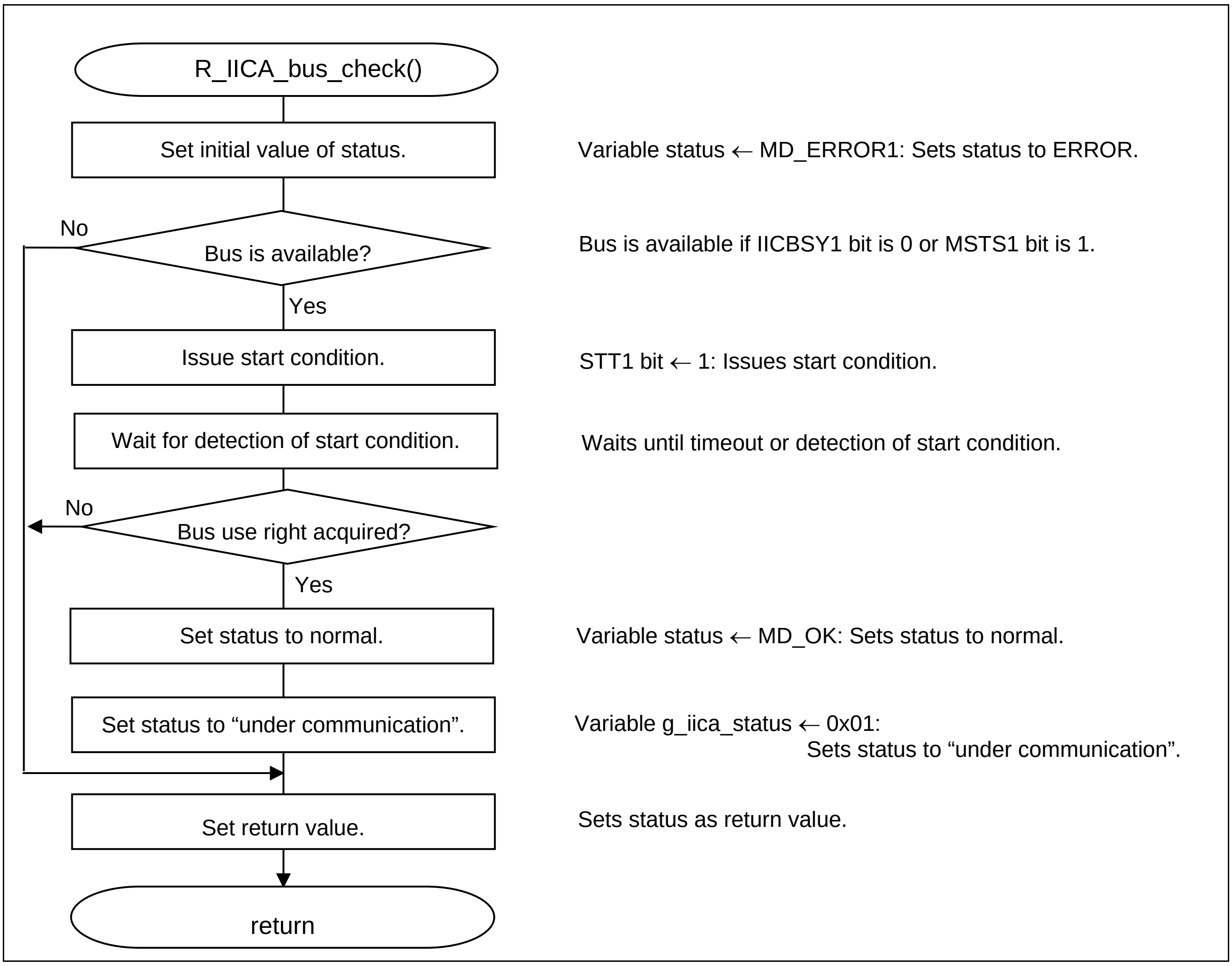


Figure 5.25 Checking I2C Bus Status

5.7.23 INTP0 Interrupt Processing

Figure 5.26 shows the flowchart of INTP0 interrupt processing.

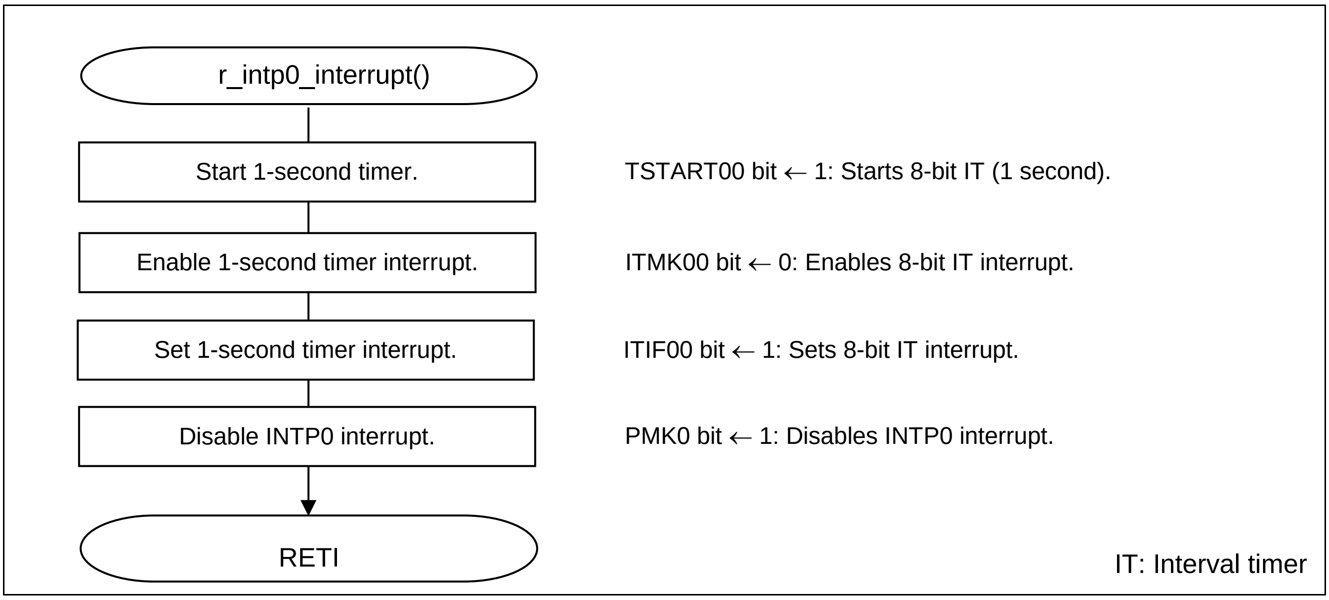


Figure 5.26 INTP0 Interrupt Processing

5.7.24 INTIT00 Interrupt Processing

Figure 5.27 shows the flowchart of INTIT00 interrupt processing.

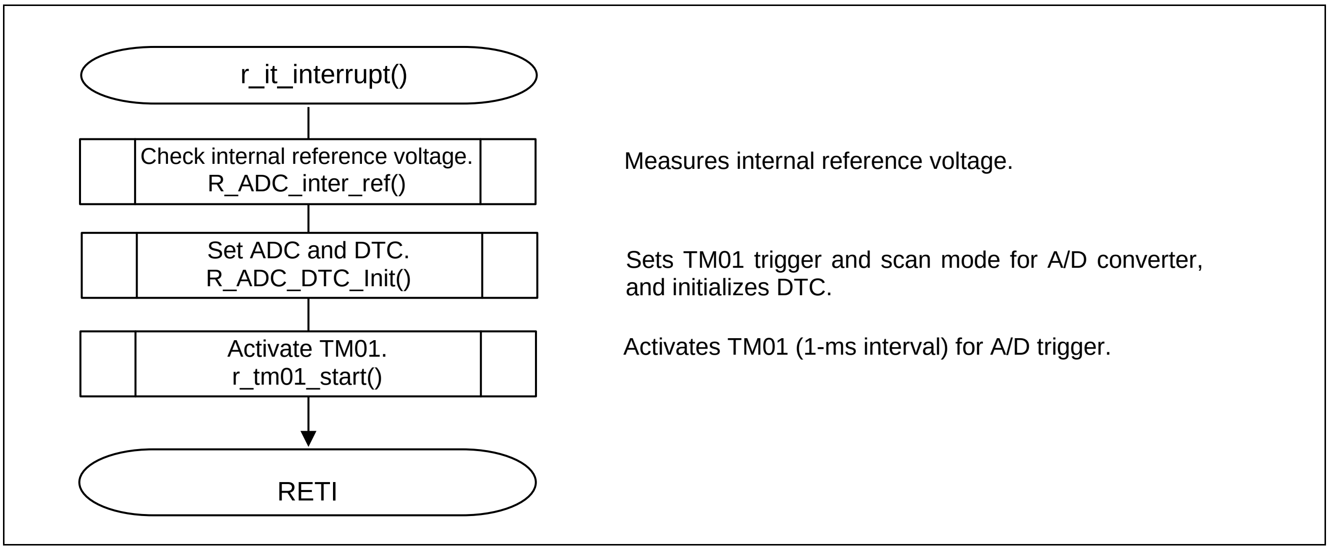


Figure 5.27 INTIT00 Interrupt Processing

5.7.25 Checking Internal Reference Voltage

Figure 5.28 and Figure 5.29 show the flowcharts for checking internal reference voltage.

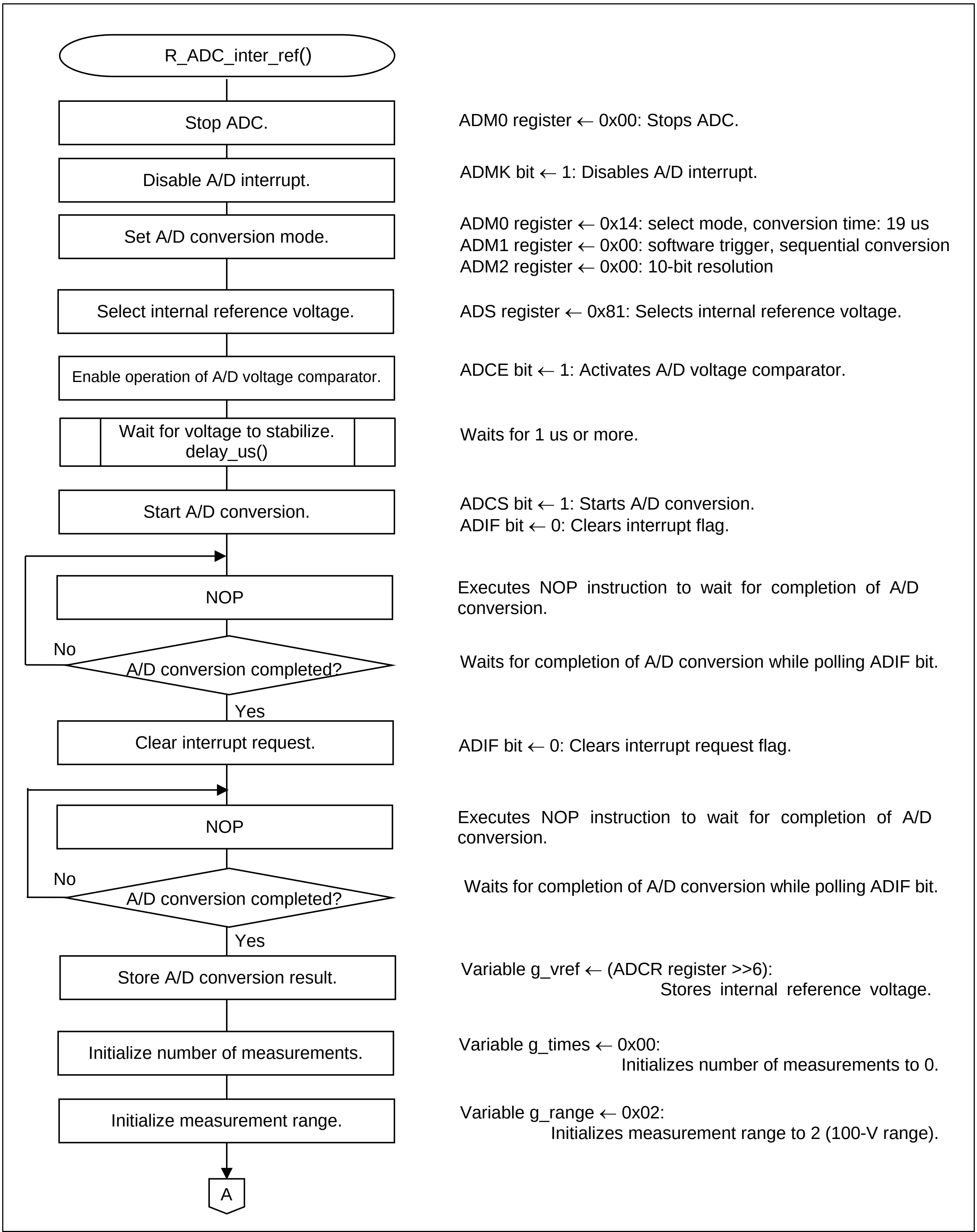


Figure 5.28 Checking Internal Reference Voltage (1/2)

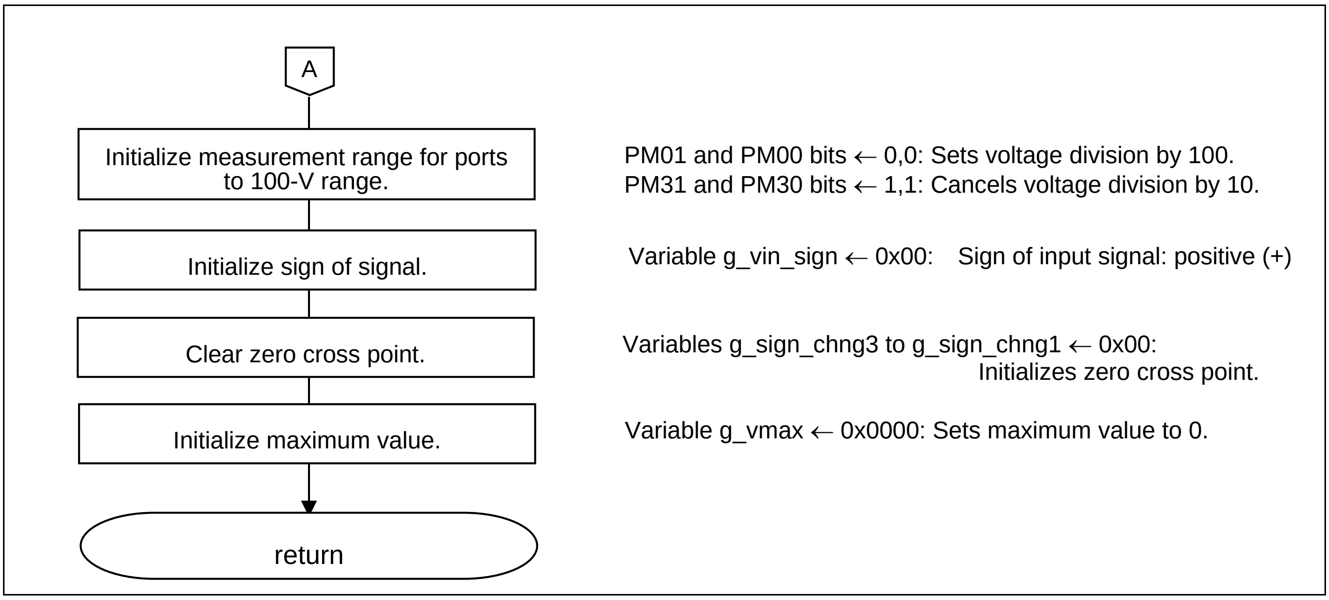


Figure 5.29 Checking Internal Reference Voltage (2/2)

5.7.26 1-us Wait Processing

Figure 5.30 shows the flowchart of the 1-us wait processing.

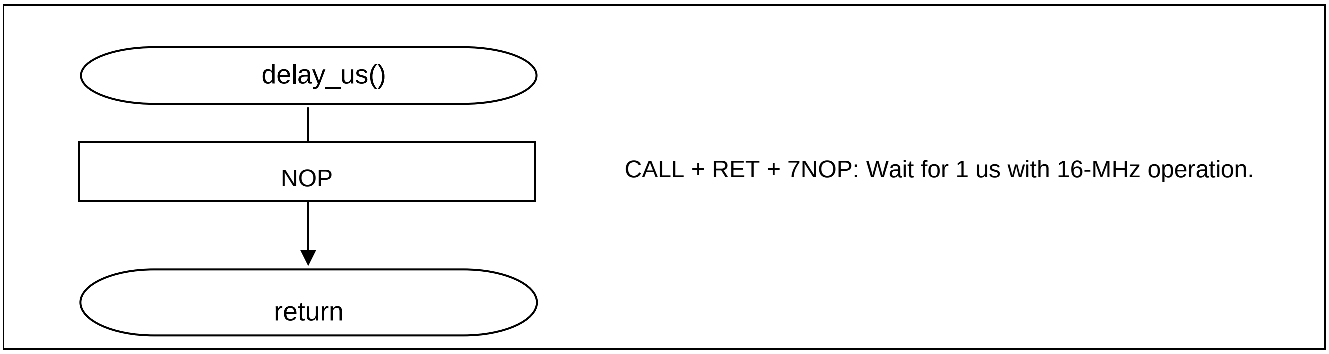


Figure 5.30 1-us Wait Processing

5.7.27 Setting ADC and DTC

Figure 5.31 shows the flowchart for setting the ADC and the DTC.

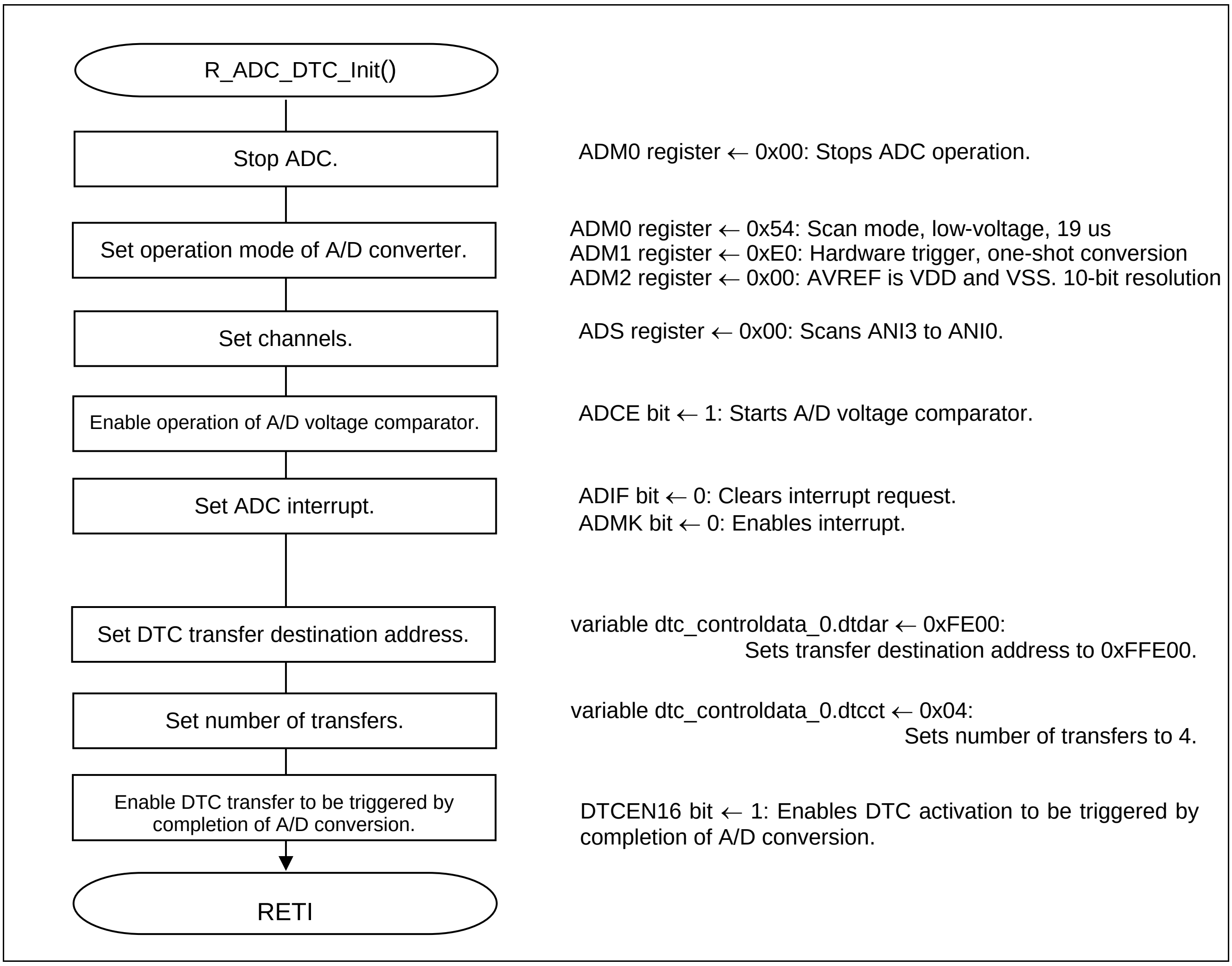


Figure 5.31 Setting ADC and DTC

5.7.28 Starting TM01

Figure 5.32 shows the flowchart for starting TM01 (1-ms interval) that is used for triggering ADC.

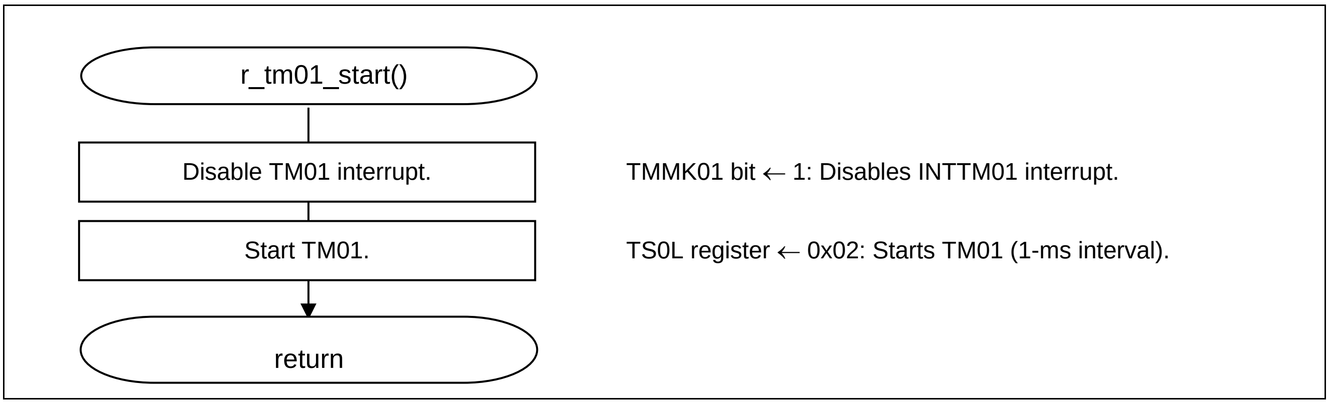


Figure 5.32 Starting TM01

5.7.29 A/D Conversion End Interrupt Processing

Figure 5.33 and Figure 5.34 show the flowcharts of A/D conversion end interrupt processing.

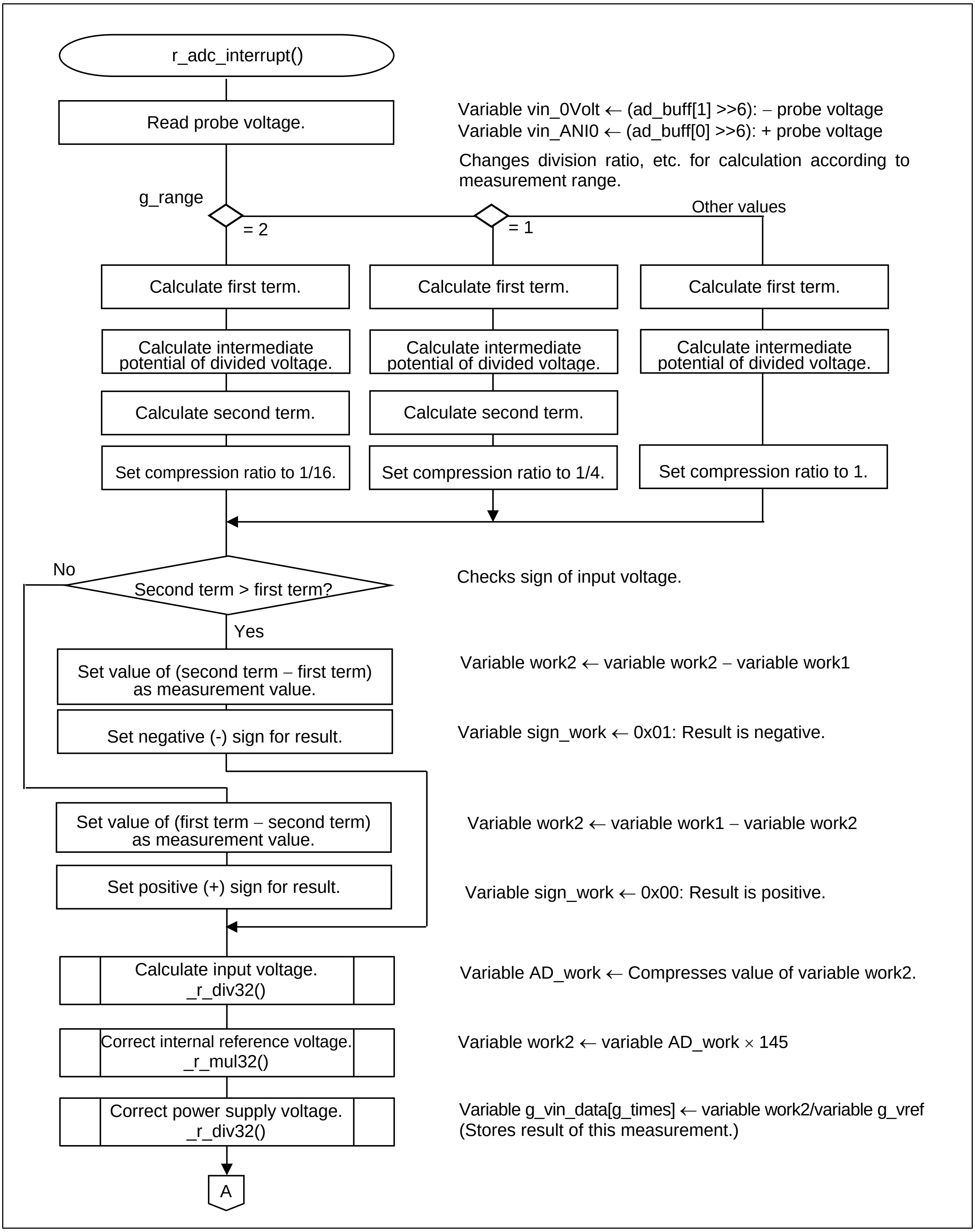


Figure 5.33 A/D Conversion End Interrupt Processing (1/2)

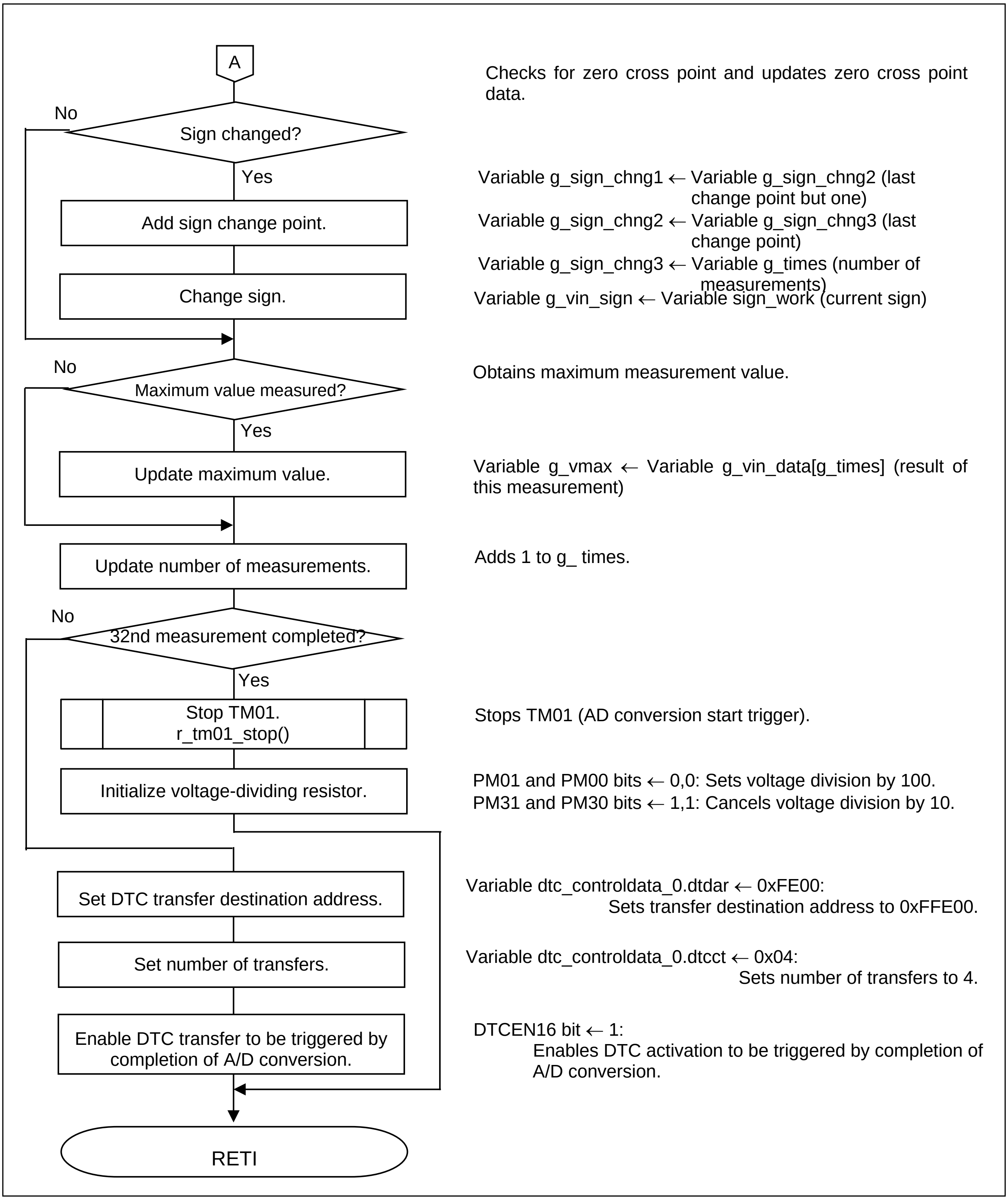


Figure 5.34 A/D Conversion End Interrupt Processing (2/2)

5.7.30 I2C Communication End Interrupt Processing

Figure 5.35 and Figure 5.36 show the flowcharts of I2C communication end interrupt processing.

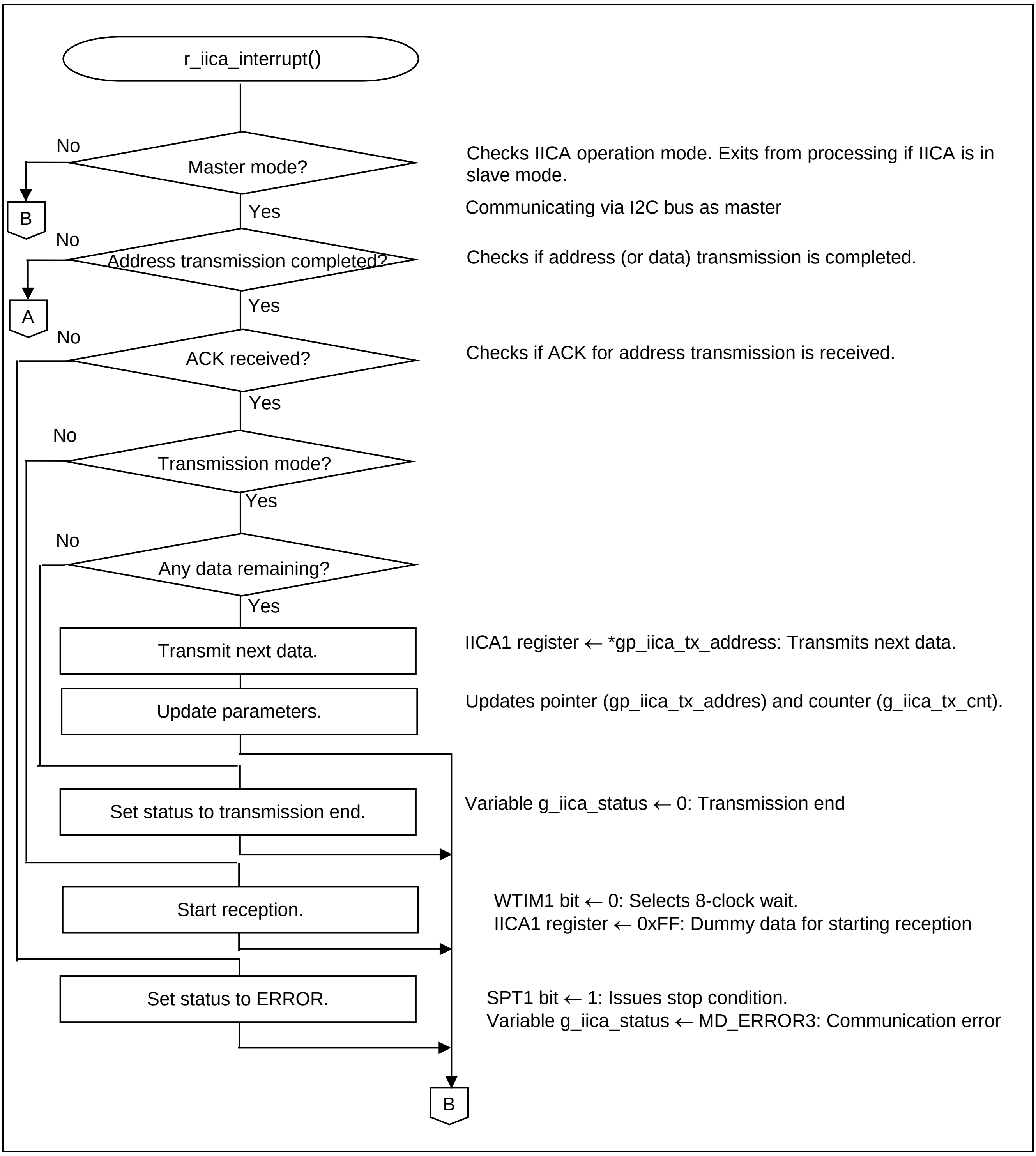


Figure 5.35 I2C Communication End Interrupt Processing (1/2)

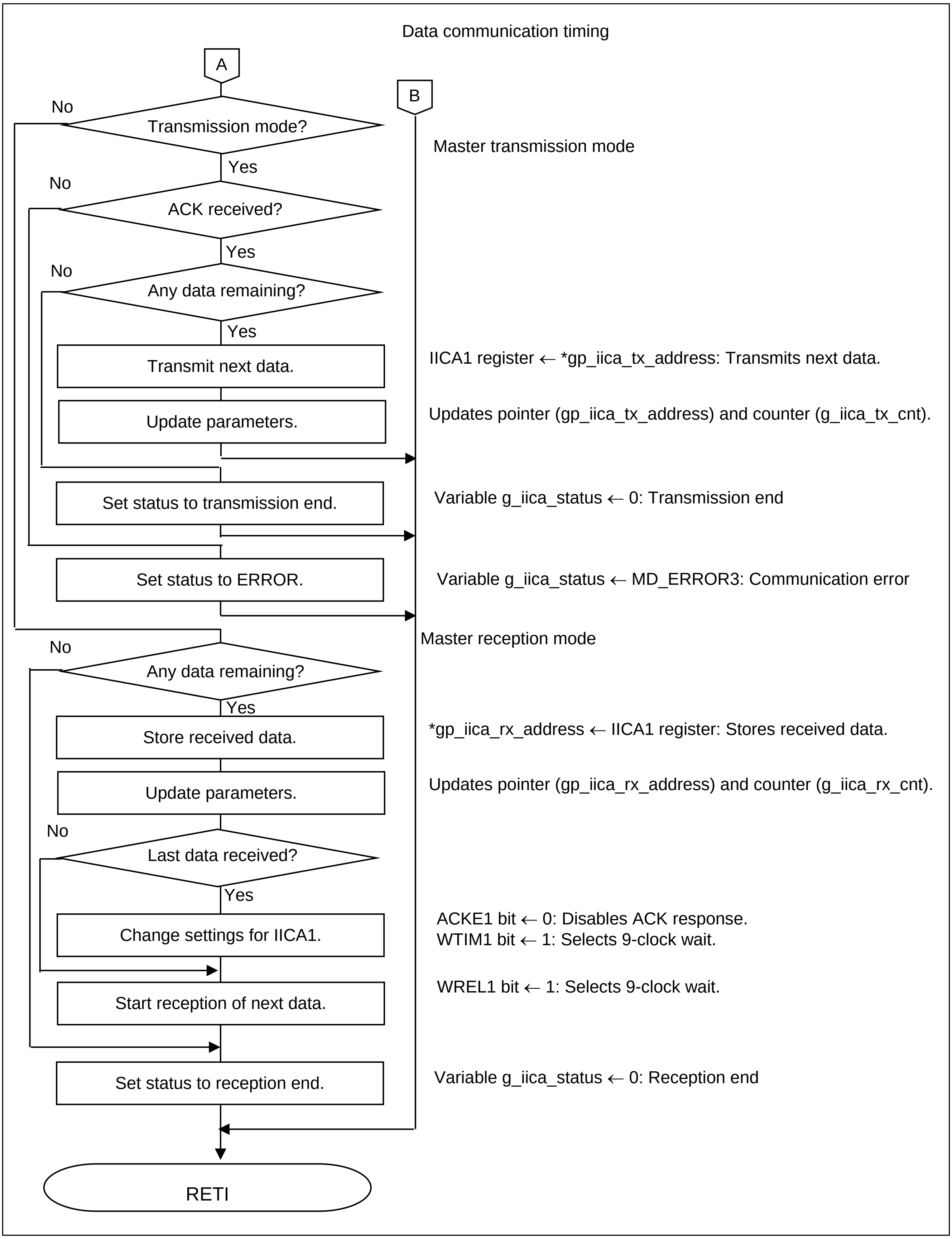


Figure 5.36 I2C Communication End Interrupt Processing (2/2)

5.7.31 16-Bit Multiplication

Figure 5.37 shows the flowchart of 16-bit multiplication.

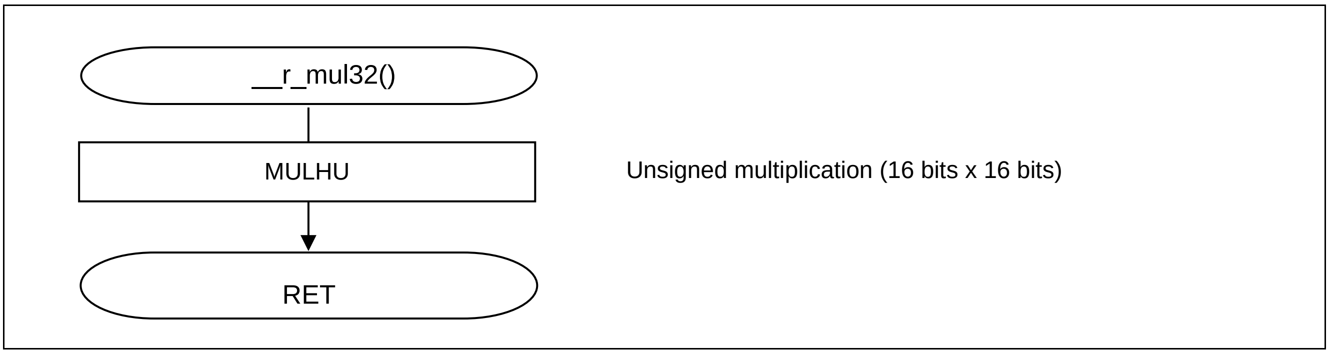


Figure 5.37 16-Bit Multiplication

5.7.32 32-Bit Division

Figure 5.38 shows the flowchart of 32-bit division.

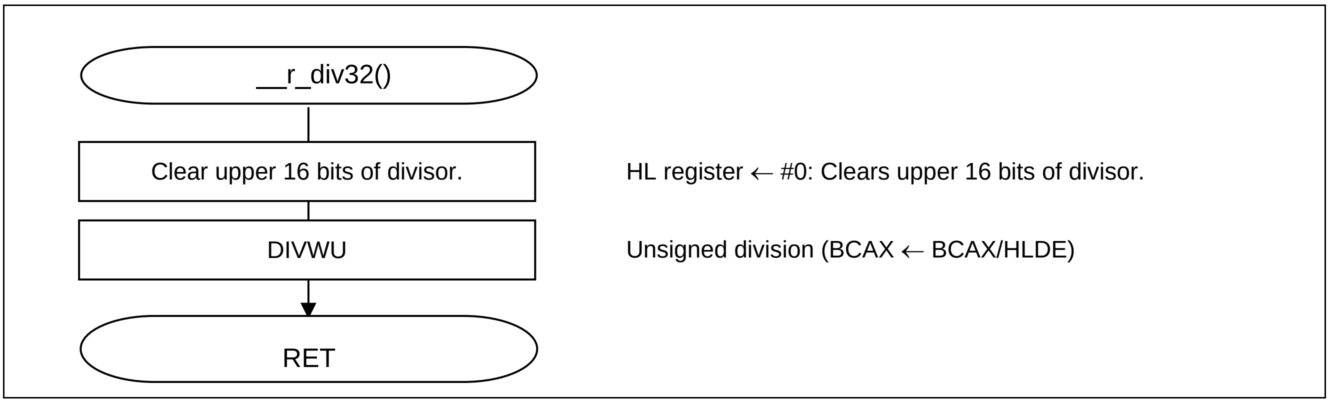


Figure 5.38 32-Bit Division

5.7.33 16-Bit Division

Figure 5.39 shows the flowchart of 16-bit division.

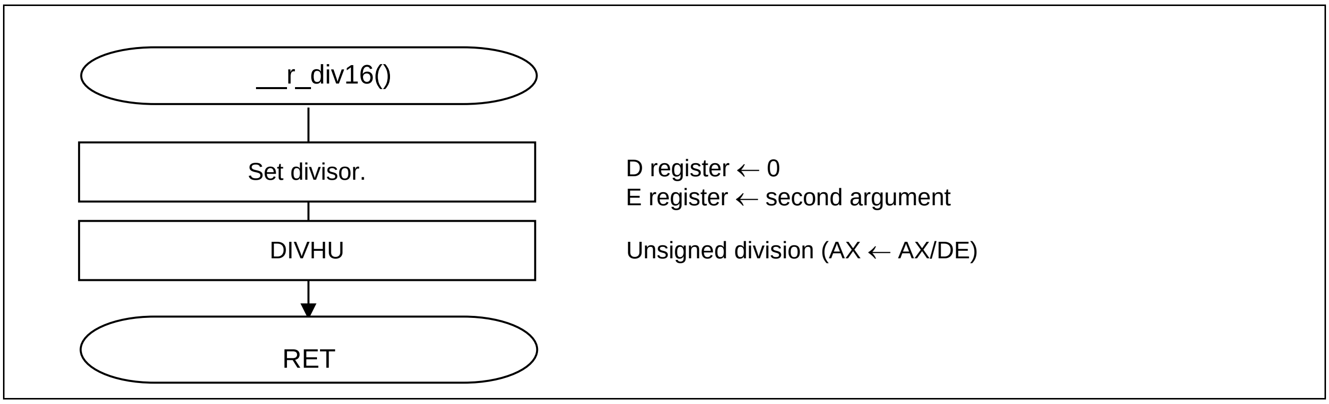


Figure 5.39 16-Bit Division

5.7.34 16-Bit Modulo Operation

Figure 5.40 shows the flowchart of 16-bit modulo operation.

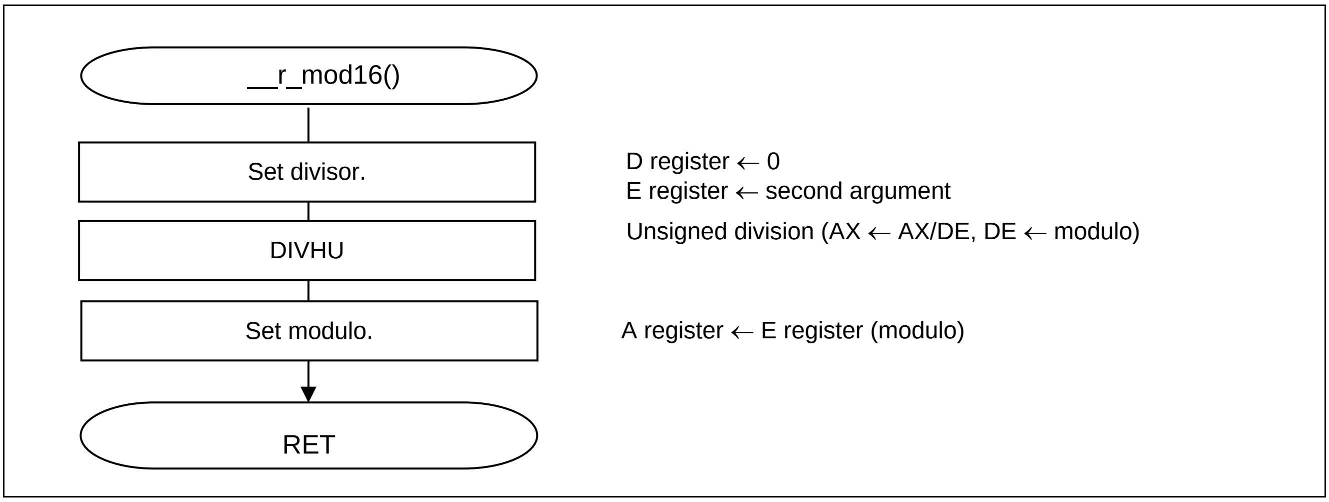


Figure 5.40 16-Bit Modulo Operation

5.7.35 Root-Mean-Square Calculation for 16-Bit Data

Figure 5.41 shows the flowchart for calculating the root-mean-square of 16-bit data.

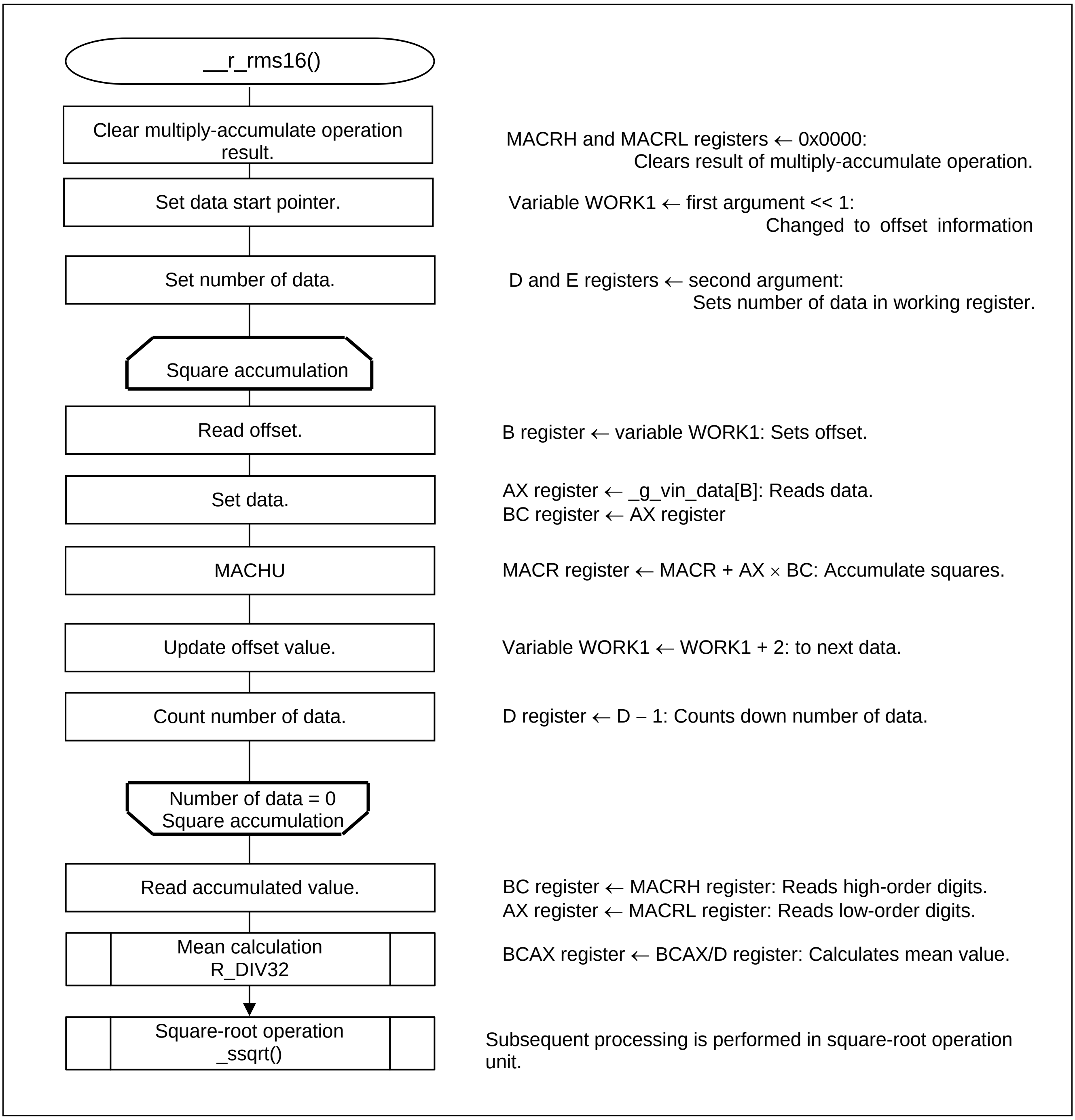


Figure 5.41 Root-Mean-Square Calculation for 16-Bit Data

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/G11 User's Manual: Hardware (R01UH0637E)

RL78 Family User's Manual: Software (R01US0015E)

(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://japan.renesas.com/

Inquiries
http://japan.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 19, 2018	–	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.

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