RL78/G16

Capacitive Touch Low Power Guide

Introduction

This application note describes the method for implementing low power consumption operation in touch measurement by the Capacitive Touch Sensing Unit (CTSUb) by using RL78/G16. This method uses the combination of intermittent operations of the CPU using the 12-bit interval timer (12-bit IT) and the suspension of the CTSUb to achieve low power consumption in touch measurement.

Target Device

RL78/G16

When implementing the touch application, conduct an extensive evaluation of the touch function according to the environment.

Contents

1. Outline

This application note describes the method for achieving low power consumption by using the suspension function of the CTSUb to perform intermittent operations in electrostatic capacitance touch measurements. This application note also shows the reference current consumption when touch measurement is performed with 100 ms cycle. In this application note, low power consumption to approximately 1/65 is implemented, compared to a case where touch measurement is performed by the CPU without using intermittent operations.

1.1 Assumed System

This sample code assumes the system shown in the red box in **Figure 1-1**. When the main system in a standby state, only the electrostatic capacitive touch button (power button) is measured at regular intervals to determine whether a touch is detected. The main system will transit to normal operation state by touch the power button.

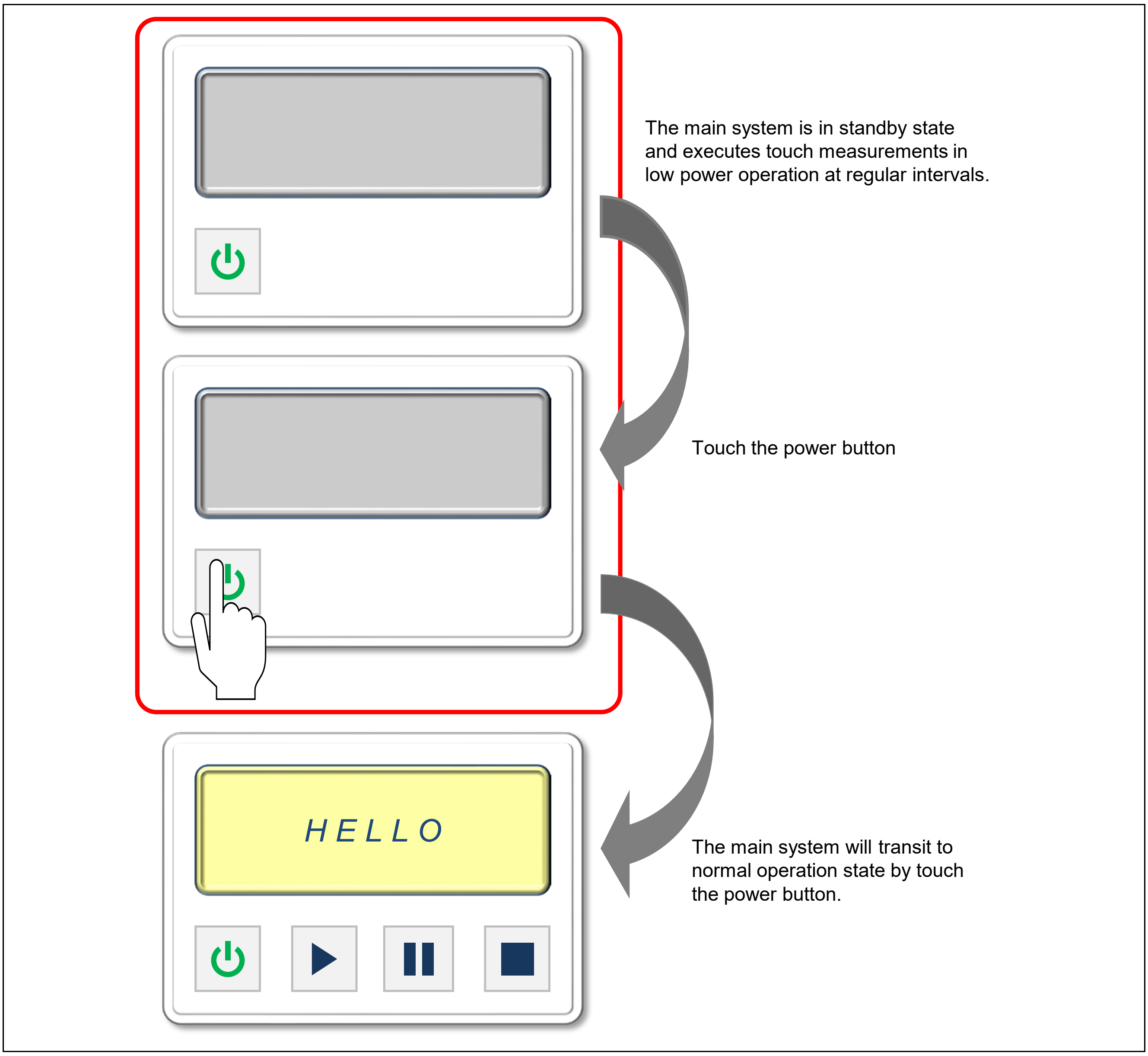


Figure 1-1 Model of the Capacitive Touch Low Power Consumption Operation

2. Overview of Low Power Consumption Operation for Touch Measurement

2.1 Operation Flow of Low-Power Consumption Touch Measurement

The capacitive touch sensing unit (CTSUb) in RL78/G16 can reduce power consumption in standby state by enabling the CTSU suspension.

This section provides a summary of the operation of peripherals and describes how power consumption is reduced, when the following processing is implemented in the touch application.

・Enable the suspension in CTSUb

・Intermittent operation of touch measurement by using the 12-bit interval timer (12bit IT)

Figure 2-1 shows an operation image of touch measurement with low power consumption.

During non-touch measurement, stopping the CPU operation by transitioning the CPU to STOP mode can reduce current consumption in section "a" in the figure as compared to normal operation mode.

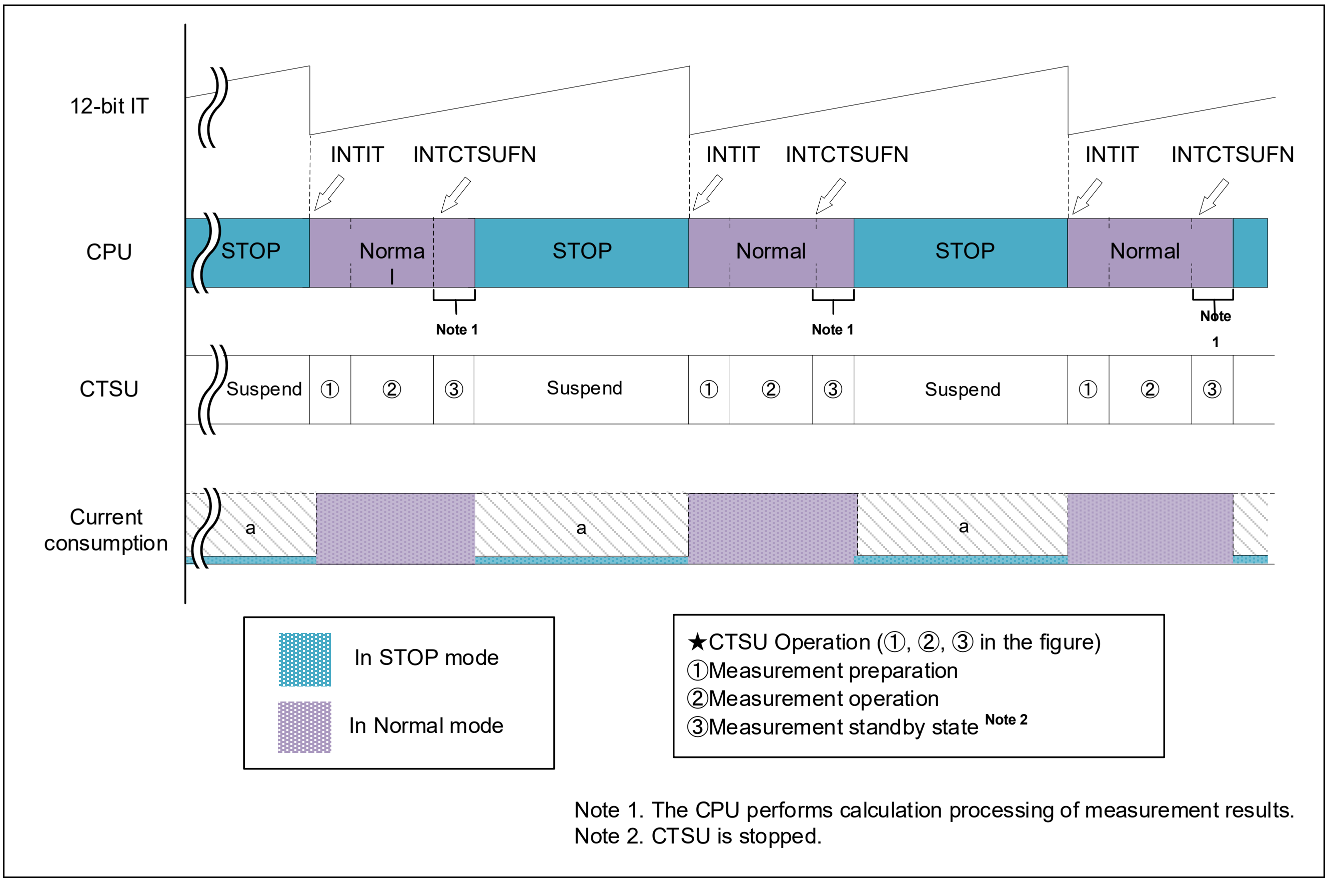


Figure 2-1 Operation Image of Touch Measurement with Low Power Consumption

〇Overview of operation

1. Set the touch-measurement interval with 12-bit IT and start counting.
2. To select a software trigger as the CTSU measurement start trigger and enable the CTSU suspension, the CTSUb turn to the SW suspend state.
   Then changes the state of CPU to STOP mode.
3. When the 12-bit IT interrupt source (INTIT) is generated, the CPU transits from STOP mode to Normal.
4. Setting touch measurement start processing by software, CTSUb releases the SW suspend status and touch measurement is started by a software trigger.
5. When the CTSUb ends measurement of all touch interfaces, it generates a CTSU measurement end interrupt (INTCTSUFN). Then CTSUb enters the Waiting for the start of measurement status.
6. RL78/G16 gets the result of measurement and determines the touch button state by the software. You can execute the user processing based on the touch button state.
7. Steps 2 to 6 are repeated.

2.2 Peripherals Implementing Low Power Consumption Touch Measurement

Table 2-1 lists the peripherals used for performing intermittent operations in touch measurement.

Table 2-1 Peripherals in the Sample Code

Used Peripherals	Functions
Capacitive Touch Sensing Unit (CTSUb)	Measures electrostatic capacitance of the touch sensor. Specifies the following settings to perform low power consumption operation. ・Select a software trigger ・Enable the suspension function
12-bit interval timer (12-bit IT)	Timer to count the touch measurement cycles. This timer generates an interrupt source signal (INTIT) every touch measurement cycle (100 ms).

3. Operation Check Environment / Conditions

Table 3-1 lists the operation check environment. **Table 3-2** lists the operation check conditions.

Table 3-1 Operation Check Environment

Item	Description
Microcontroller used	RL78/G16 (R5F121BCAFP)
Operating frequency	• Main system clock High-speed on-chip oscillator (f_{IH}): 16 MHz • CPU/peripheral hardware clock (f_{CLK}): 16 MHz • Subsystem clock Low-speed on-chip oscillator (f_{IL}): 15 kHz
Operating voltage	5.0 V SPOR detection voltage: At rising edge: TYP. 2.57 V (2.44 V to 2.68 V) At falling edge: TYP. 2.52 V (2.40 V to 2.62 V)
Target board	RL78/G16 Capacitive Touch Evaluation System (Product model: RTK0EG0047S01001BJ)
Integrated Development Environment (e² studio)	e² studio (2025-01) from Renesas Electronics Corp.
C compiler (e² studio)	CC-RL V1.15.00 from Renesas Electronics Corp.
Smart Configurator	V25.1.0 from Renesas Electronics Corp.
Board support package (BSP)	V1.80 from Renesas Electronics Corp.
QE for Capacitive Touch	V4.1.0 from Renesas Electronics Corp.

Table 3-2 Operation Check Conditions

Item	Description
Touch measurement cycle	Low power consumption operation mode:100 ms Normal operation mode: 20 ms
Sensor drive pulse frequency	1.0 MHz
Touch Sensor (TS pin)	TS08
Measurement mode	Self-capacitance multi-scan mode (CTSUMD[1:0] bits of CTSUCR1 register = 00b)
Measurement start trigger	Software trigger (CTSUCAP bit of CTSUCR0 register = 0)
Enables/Disables the CTSU suspension	low power consumption operation mode: Suspension is enabled. (CTSUSNZ bit of CTSUCR0 register = 1) Normal operation mode: Suspension is enabled. (CTSUSNZ bit of CTSUCR0 register = 0)
CTSU power supply capacity adjustment	Normal output (CTSUATUNE1 bit of CTSUCR0 register = 0)
Sensor stabilization wait time	32 cycles (SST[7:0] bits CTSUCRBL register = 0x1f)
Measurement count setting	7 (SNUM[7:6] bits CTSUSO1 register = 00b, SNUM[5:0] bits CTSUSO0 register = 000111b)

4. Hardware Descriptions

4.1 Example of Hardware Configuration

Figure 4-1 shows an example of the hardware configuration used in the application note.

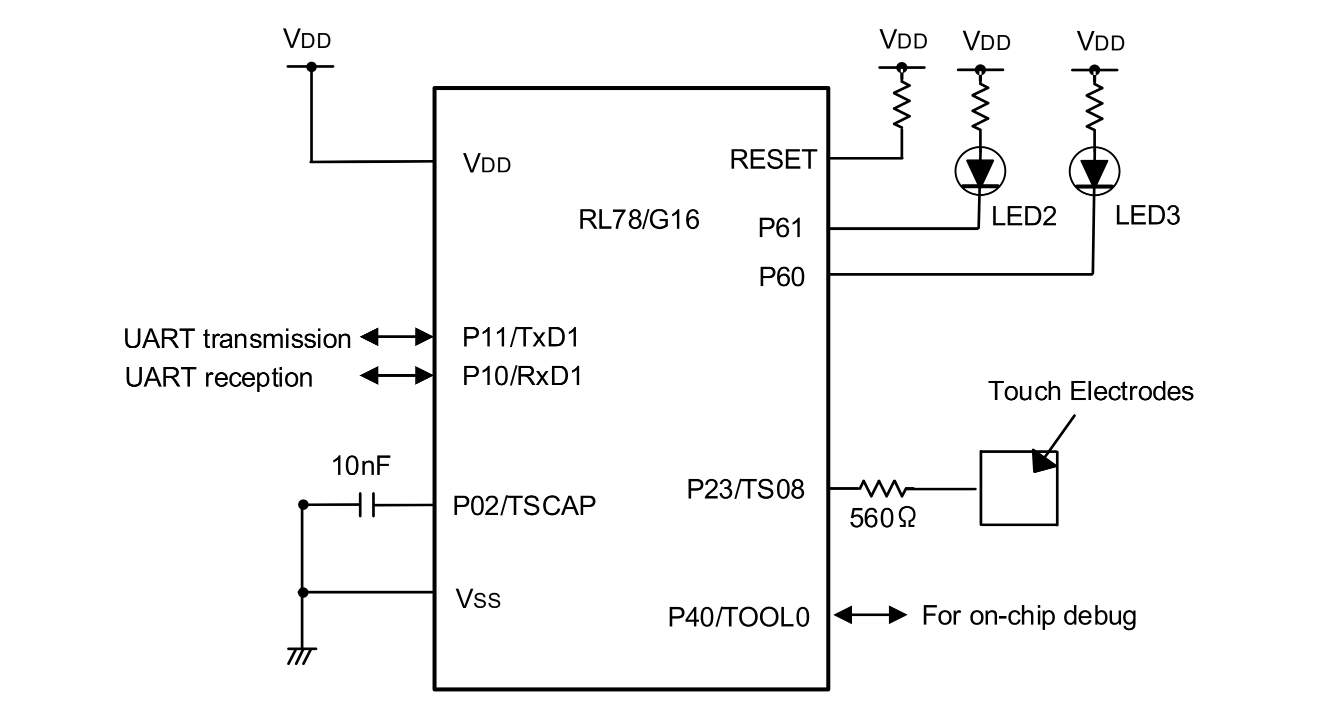


Figure 4-1 Hardware Configuration

Caution 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes appropriate pin handling and meets electrical characteristic requirements (connect each input-only port to V_{DD} or V_{SS} through a resistor).

Caution 2. V_{DD} must not be lower than the reset release voltage (V_{SPOR}) that is specified for the SPOR.

4.2 List of Pins to be Used

Table 4-1 lists the pins to be used and their functions.

Table 4-1 Pins to be Used and Their Functions

Pin name	I/O	Function
P23/TS08	I/O	Pins for electrostatic capacitance measurement
P10/TxD1	Output	UART serial data transmit pin
P11/RxD1	Input	UART serial data receive pin
P60	Output	LED3 control pin
P61	Output	LED2 control pin
P02/TSCAP	—	Secondary power supply for measurement (capacitor) connection pin

Caution: In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

5. Software Description

5.1 Sample Code Outline of Operation

The sample code operates as follows.

===== Up to initial offset tuning ===

1. After reset release by power-on, set each peripheral. Then, RM_TOUCH_Open function is executed to initialize the capacitive touch sensing unit (CTSU) Note.1.

2. Execute initial offset tuning.

===== Low power consumption mode ==

3. To manage the touch measurement cycle, start the 12-bit IT (measurement cycle: 100ms).

4. Set the CTSU to SW suspended state. Set the CTSUSNZ bit in the CTSUCR0 register to 1 to enable CTSU suspend function by setting.

5. CPU status transitions to STOP mode by executing the STOP instruction.

6. When the 12-bit IT interrupt source (INTIT) is generated, the CPU status transitions to Normal (Normal operation) mode.

7. Configure settings for starting measurement from the SW suspended status.
   Set the CTSUSNZ bit of CTSUCR0 register to 0 to disable the suspension function of CTSU.
   Then wait for at least 64 cycles of the base clock. In this sample code, wait for 128 us.

8. Execute RM_TOUCH_ScanStart function Note.2, and the CTSU starts touch measurement.

9. When the CTSU ends measurement, it generates a measurement end interrupt (INTCTSUFN) and enters the measurement standby state.

10. Get the measurement results by executing the RM_Touch_DataGet function.
    When the button status is determined to touch ON, stop the 12-bit IT and transit the Normal operation mode. When the button status is determined to touch OFF, return to flow “4”

===== Normal operation mode ===

11. Turn on the LED3 as a measurement-mode display LED.

12. Wait for 20 ms by software to adjust the measurement interval in Normal operation mode.

13. Execute RM_TOUCH_ScanStart function Note.2, and the CTSU starts touch measurement.

14. When the CTSU ends measurement, it generates a measurement end interrupt (INTCTSUFN) and enters the measurement standby state.

15. Get the measurement results by executing the RM_Touch_DataGet function. When the button status is determined to touch ON, turn on the LED2 and return to flow “12”. When the button status is determined to touch OFF, turn off the LED2 and return to flow “12”. When 3 seconds elapses in touch-OFF status, turn off the LED3 and return to flow “3” to transit Low power consumption mode. After that, the same operation as in this flow will be performed.

Note.1 In the processing of the RM_TOUCH_Open function, the CTSUCAP bit of the CTSUCR0 register is set to 0 to select a software trigger as the CTSU measurement start trigger.

Note.2 In the processing of the RM_TOUCH_ScanStart function, the CTSUSTRT bit of the CTSUCR0 register is set to 1. For details on setting conditions, refer to the description of the CTSUSNZ bit in section 15.3.2 CTSU control registers 0 (CTSUCR0) in the RL78/G16 User’s Manual: Hardware (R01UH0980).

5.2 Smart Configurator Settings

5.2.1 Components

Figure 5-1 shows a list of the Smart Configurator components and versions.

Selected components:

Component	Version	Configuration
Board Support Packages. - v1.80 (r_bsp)	1.80	r_bsp(used)
Capacitive Sensing Unit driver. (r_ctsu)	2.10	r_ctsu(used)
Interval Timer	1.6.0	Config_IT(IT: used)
Ports	1.6.0	Config_PORT(PORT: used)
Touch middleware. (rm_touch)	2.10	rm_touch(used)
UART Communication	1.8.0	Config_UART1(UART1: used)

Figure 5-1 Components of Smart Configurator

5.2.2 Component Settings

Table 5-1 describes the settings of each component in the sample code.

Table 5-1 Settings for Each Component

Component	Configuration	Settings
Capacitive Sensing Unit driver (CTSU)	r_ctsu	Specify to use all TSCAP pin and TS pins. - Parameter check: Use system default (default) - Interrupt setting: default
Touch middleware	rm_touch	- Support QE monitor using UART: Enable - Support QE tuning using UART: Enable - UART channel: UART1 - Parameter check: Use system default (default) - Type of chattering suppression: TypeA (default)
UART Communication	Config_UART1	- Transmit data pin (TxD1): P11, Receiver data pin (RxD1): P10[Note 1,2] - Operation clock: CK00, Clock source: f_{CLK}/2 - Transfer mode setting: Continuous transfer mode - Date length setting: 8bits - Transfer direction setting: LSB (default) - Parity setting: None (default) - Stop bit length setting: 1 bit (default) - Transfer data level setting: Non-reverse (default) - Transfer rate setting: 115200 bps - Interrupt setting: default - Callback function setting Transmission end: Enable Reception end: Enable Reception error: Enable
Interval Timer (12-bit IT)	Config_IT	- Operating clock: f_{SXP} [Note 3] - Interval time: 100 ms - Interrupt setting: default
Ports	Config_PORT	Set P60 and P61 to High-level output.

Note1. UART communication is used for tuning and monitoring in QE for Capacitive Touch. However, in RL78/G16, the UART communication through the serial array unit (SAU) cannot be used during STOP mode. So, the clock supply to the SAU is stopped during touch measurement in this sample code. Refer to 5.5 Folder/File Configuration for details.
Note2. Set the UART communication terminal in the “terminal” tab of the Smart Configurator.
Note3. Set the operating clock in the “Clock” tab of the Smart Configurator.

5.3 Capacitive Touch Settings

5.3.1 Touch Interface Configuration

Figure 5-2 shows the touch interface configuration. Measuring TS08 terminal by self-capacitance method.

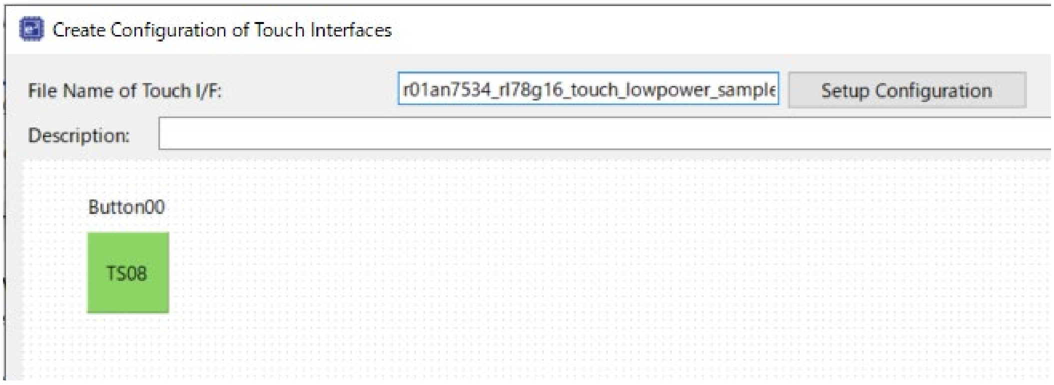


Figure 5-2 Touch Interface Configuration

5.3.2 Configuration (Methods) Settings

Figure 5-3 shows the Configuration (Methods) Settings of the touch interface.

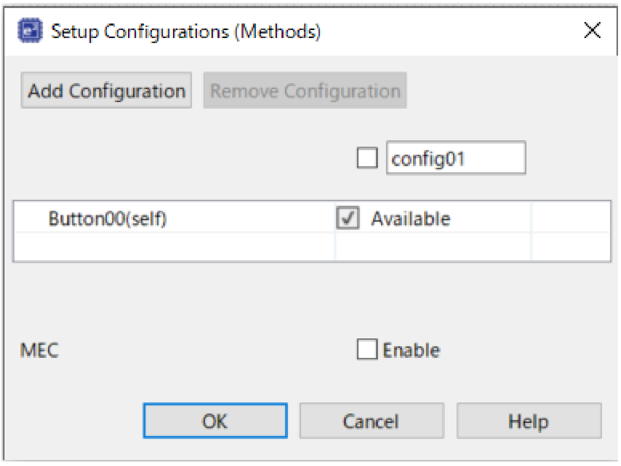
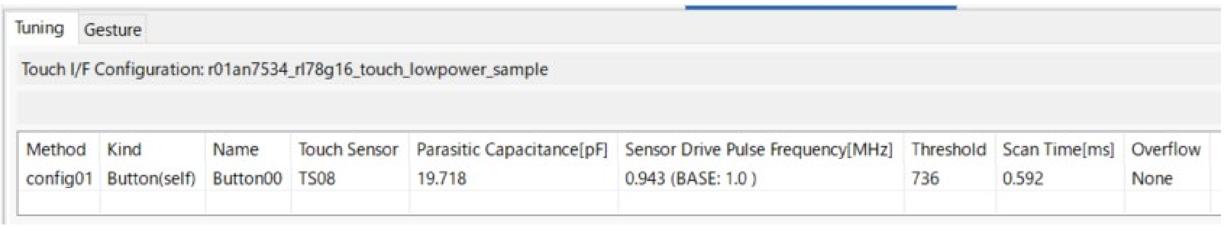


Figure 5-3 Setup Configuration (Methods)

5.3.3 Tuning Results

Figure 5-4 shows tuning results of touch interface in QE tuning. Sample code operates with the following setting values.

Since the values in QE tuning results depend on the operating environment at QE tuning, these values may change at QE tuning again.



Tuning | Gesture

Touch I/F Configuration: r01an7534_rl78g16_touch_lowpower_sample

Method	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow
config01	Button(self)	Button00	TS08	19.718	0.943 (BASE: 1.0)	736	0.592	None

Figure 5-4 QE Tuning Result

5.4 Setting of Option Byte

Table 5-2 shows the option byte settings.

Table 5-2 Option Byte Settings

Address	Setting Value	Contents
000C0H	11101111B (0xEF)	Disables the watchdog timer. Note (Counting stopped after reset)
000C1H	11111011B (0xFB)	SPOR detection voltage: At rising edge: 2.57 V (TYP) (2.44 V to 2.68 V) At falling edge: 2.52 V (TYP) (2.40 V to 2.62 V) Settings of P125/RESET pin: Enables RESET pin
000C2H	11101001B (0xE9)	HS mode High-speed on-chip oscillator clock (f_{IH}): 16 MHz
000C3H	10000101B (0x85)	Enables on-chip debugging

Note: The settings for the watchdog timer are listed in the r_bsp settings list on the Components tab. Leave the default (Unused) and do not change it.

The setting value of the option byte can be checked from the project properties after code generation. Open the project properties and select “C/C++ Build” -> “Settings” to open a “Tool Settings” tab, and select “Linker” -> “Device” and the “User option byte value” and “On-chip debug control value” are displayed.

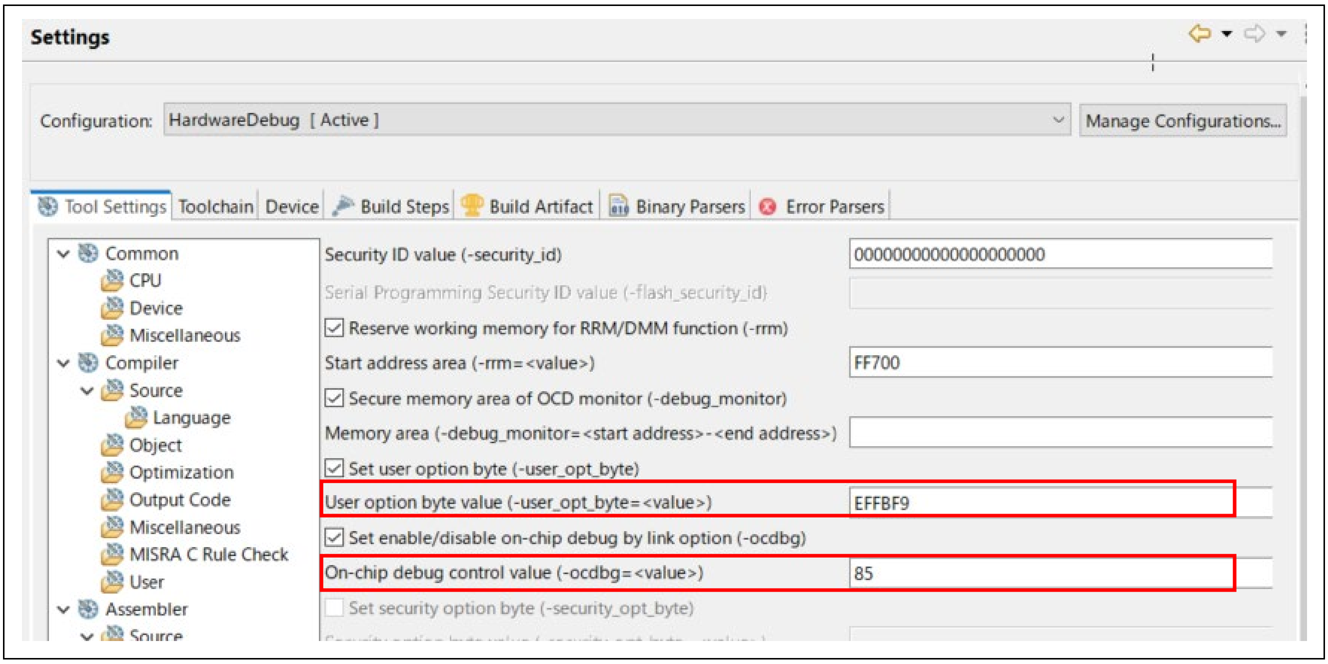


Figure 5-5 User Option Byte Value and On-chip Debug Control Value

5.5 Folder/File Configuration

Table 5-3 lists the sample code folders generated by Smart Configurator and QE for Capacitive Touch.
Table 5-4 lists the file added or changed in the sample code.

Table 5-3 Sample Code Folders

Folder Name	Description
qe_gen	Folder generated by QE for Capacitive Touch
smc_gen	Folder generated by Smart Configurator

Table 5-4 Files Added or Changed in the Sample Code

File Name	Storage	Outline	Change / Additional Contents
rl78g22_rssk_SNOOZE_sample.c	src folder	Main processing	Added the following processing: - Enable Maskable interrupts. - Added the touch measurement control processing function.
qe_touch_sample.c	qe_gen folder	Touch measurement control processing	Added the following processing: - Added processing to reduce power consumption. - LED control processing
r_cg_systeminit.c	src/smc_gen/general folder	Initial setting of PORT, UART, 12-bit IT	Comment out the default UART function settings Note.

Note. UART communication is used for tuning and monitoring in QE for Capacitive Touch. In this example project, the functions for UART communication are generated. However, in RL78/G16, the UART communication through the serial array unit (SAU) cannot be used during STOP mode. Therefore, this sample code performs the following process, the clock supply to the SAU is stopped during touch measurement.

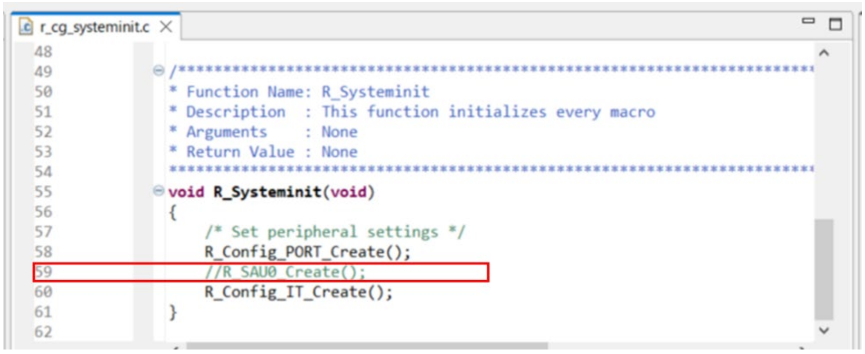


Figure 5-6 Comment Out to R_SAU0_Create Function in the “r_cg_systeminit.c”file

If you want to monitor using QE for Capacitive Touch for this sample project, you need to change the sample code. Make the following changes before building and compiling the project.

- Uncomment the “R_SAU0_Create();” on line 59 of the “r_cg_systeminit.c” file as shown in Figure 5-7.
- Comment out the low power mode measurement processing on line 92 of the “qe_touch_sample.c” file as shown in Figure 5-8.

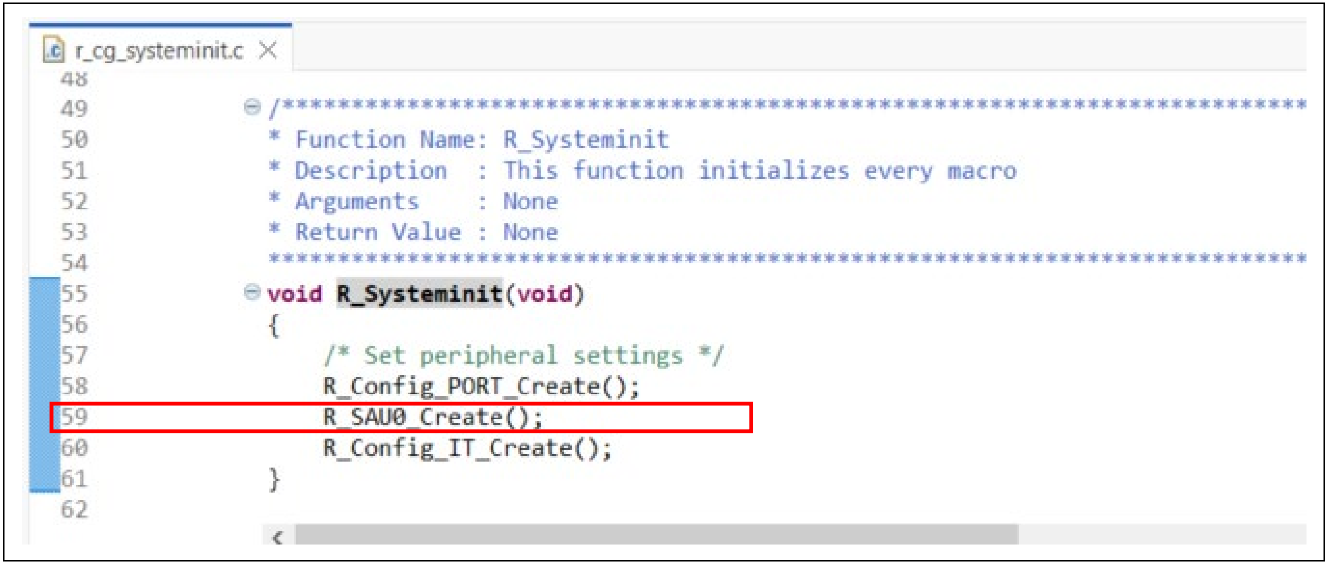


Figure 5-7 Enable R_SAU0_Create Function in the qe_touch_sample.c file

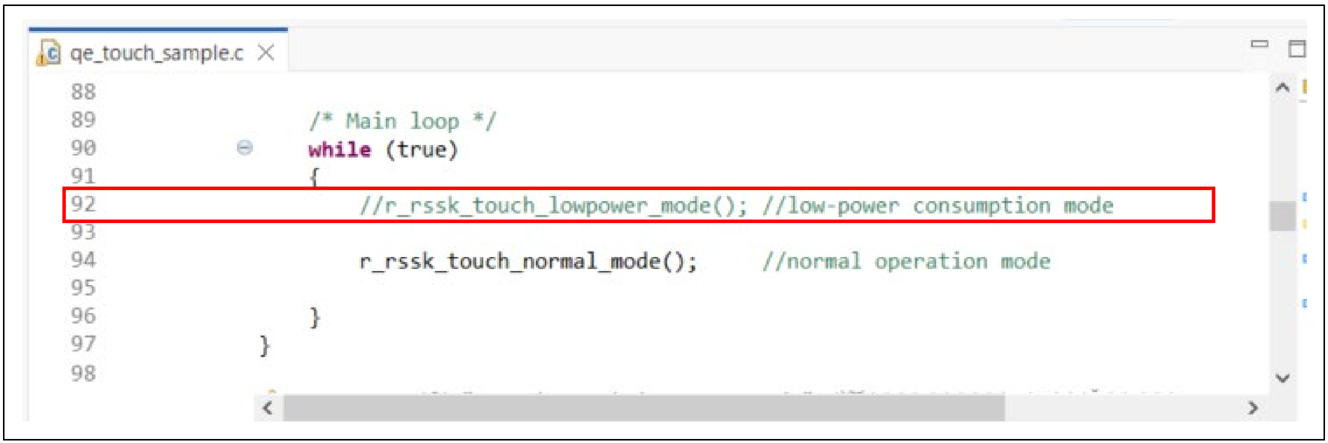


Figure 5-8 Disable to the Low Power Mode Measurement Processing in the qe_touch_sample.c file

5.6 Sample Code Details

5.6.1 List of Constants

Table 5-5 lists the global variable that is used by this sample program

Table 5-5 Constants

Constants Name	Setting Value	Description
LED2	(P6_bit.no1)	Pointer to port control register connected to LED2
LED3	(P6_bit.no0)	Pointer to port control register connected to LED3
LED_ON	(0)	Turn on the LED
LED_OFF	(1)	Turn off the LED
TOUCH_SCAN_INTERVAL_EXAMPLE	(20 * 1000)	Software delay value [20 ms]

5.6.2 List of Variables

Table 5-6 lists the global variable that is used by this sample program

Table 5-6 Global variable

Type	Variable name	Description
uint64_t	button_status	Variable to check the button status
uint8_t	g_qe_touch_flag	Measurement completion flag

5.6.3 List of Functions

Table 5-7 shows the functions used in the sample code. However, the functions that have not been changed after output by Smart Configurator or QE for Capacitive Touch are not listed.

Table 5-7 Functions

Function Name	Description	Source File
main	Main processing	rl78g16_rssk_lowpower_sample.c
qe_touch_main	Touch measurement control processing	qe_touch_sample.c
r_rssk_touch_lowpower_mode	Low power mode measurement processing	qe_touch_sample.c
r_rssk_touch_normal_mode	Normal mode measurement processing	qe_touch_sample.c

5.6.4 Specification of Functions

This section shows function specifications that are used in this sample code.

main	
Synopsis	Main processing
Header	r_smc_entry.h
Declaration	void main(void)
Explanation	Call the qe_touch_main()
Argument	- -
Return value	-

qe_touch_main		
Synopsis	Touch measurement control processing	
Header	qe_touch_config.h, r_smc_entry.h	
Declaration	void qe_touch_main(void)	
Explanation	This function configure the touch measurement settings and control the touch measurement. Execute the initial offset tuning, and then each operation mode is processed.	
Argument	-	-
Return value	-	

r_rssk_touch_lowpower_mode		
Synopsis	Low power mode measurement processing	
Header	qe_touch_config.h, r_smc_entry.h	
Declaration	static void r_rssk_touch_lowpower_mode(void)	
Explanation	This function performs intermittent operation by 12bit IT and controls the touch measurement using SW suspended state. Wait in STOP mode until 12-bit IT interrupt (INTIT) occurs. Performs touch measurement by CTSU and gets the measurement result at 100 ms cycle. When the button status is determined to touch ON, transit the Normal operation mode. When the button status is determined to touch OFF, repeat the touch measurement in Low power consumption mode.	
Argument	-	-
Return value	-	

r_rssk_touch_normal_mode		
Synopsis	Normal mode measurement processing	
Header	qe_touch_config.h, Config_IT.h, r_smc_entry.h	
Declaration	static void r_rssk_touch_normal_mode(void)	
Explanation	This function controls the touch measurement using the software timer in Normal (normal operation) mode. Performs touch measurement by CTSU and gets the measurement result at 20 ms cycle. When the button status is determined to touch ON, turn on the LED2 and repeat the touch measurement in Normal operation mode. When the button status is determined to touch OFF, turn off the LED2 and repeat the touch measurement in Normal operation mode. When 3 seconds elapses in touch-OFF status, turn off the LED3 and transit Low power consumption mode.	
Argument	-	-
Return value	-	

5.6.5 Flowcharts

5.6.5.1 Main Processing

Figure 5-9 shows the flowchart for the main processing.

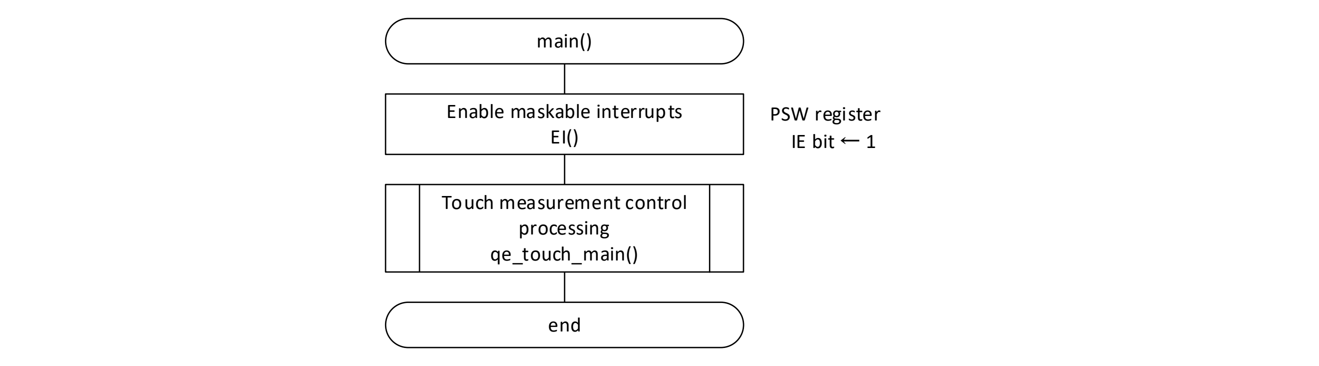


Figure 5-9 Main processing

5.6.5.2 Touch Measurement Control Processing

Figure 5-10 shows the flowchart for touch measurement control processing.

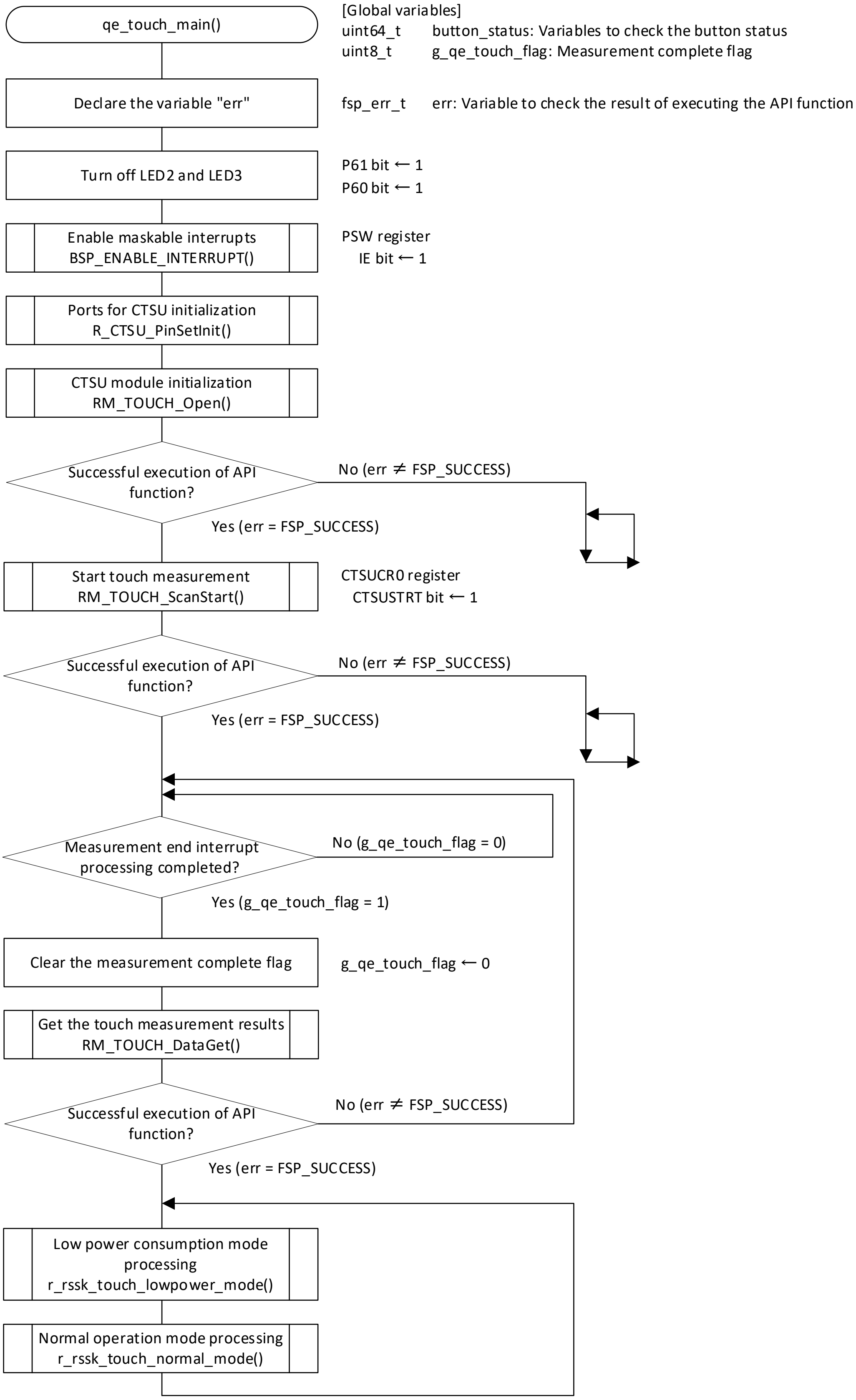


Figure 5-10 Touch measurement control processing

5.6.5.3 Low Power Consumption Mode Processing

Figure 5-11 shows the flowchart for low power consumption mode processing.

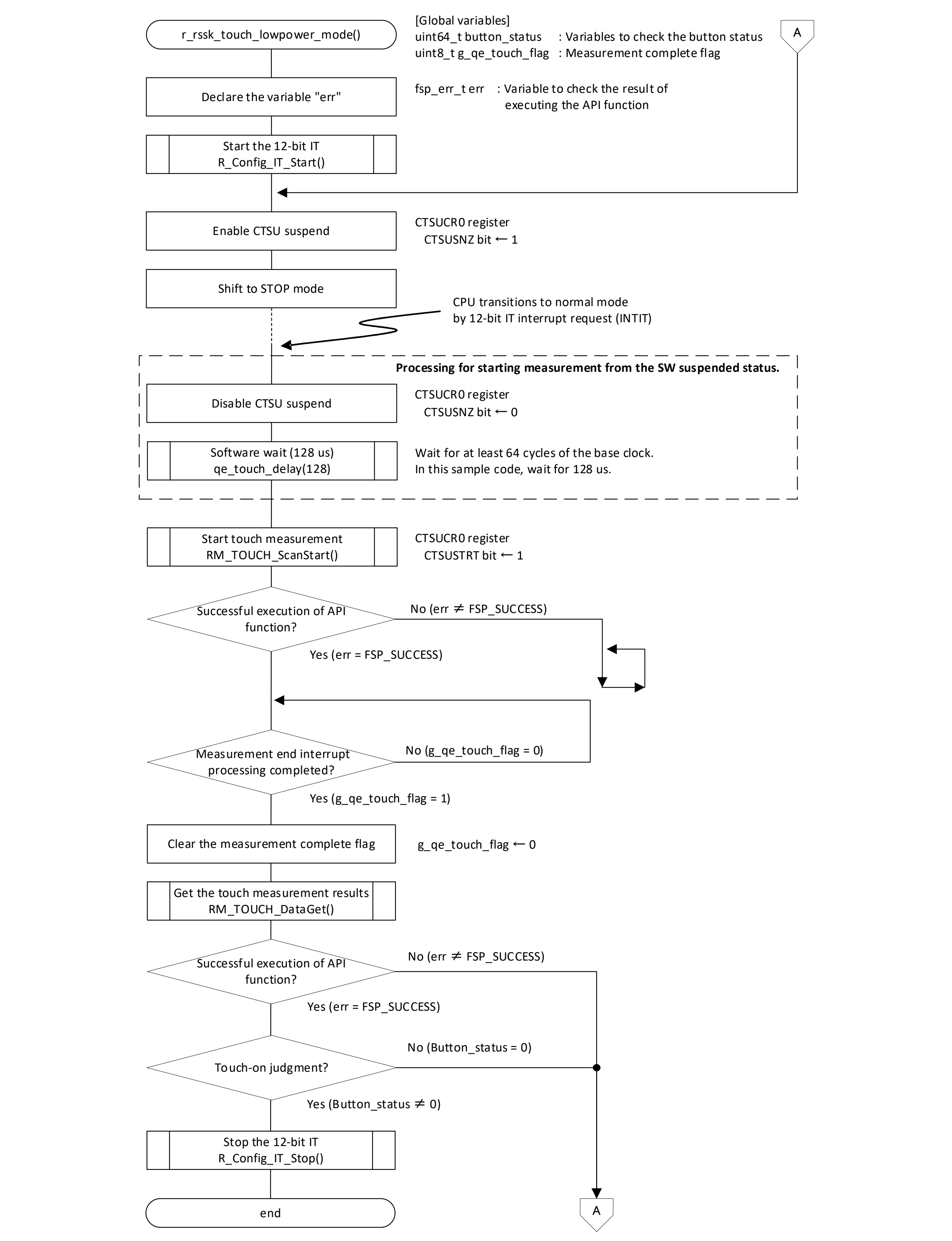


Figure 5-11 Low Power Consumption Mode Processing

5.6.5.4 Normal Operation mode Processing

Figure 5-12 shows the flowchart for touch measurement control processing.

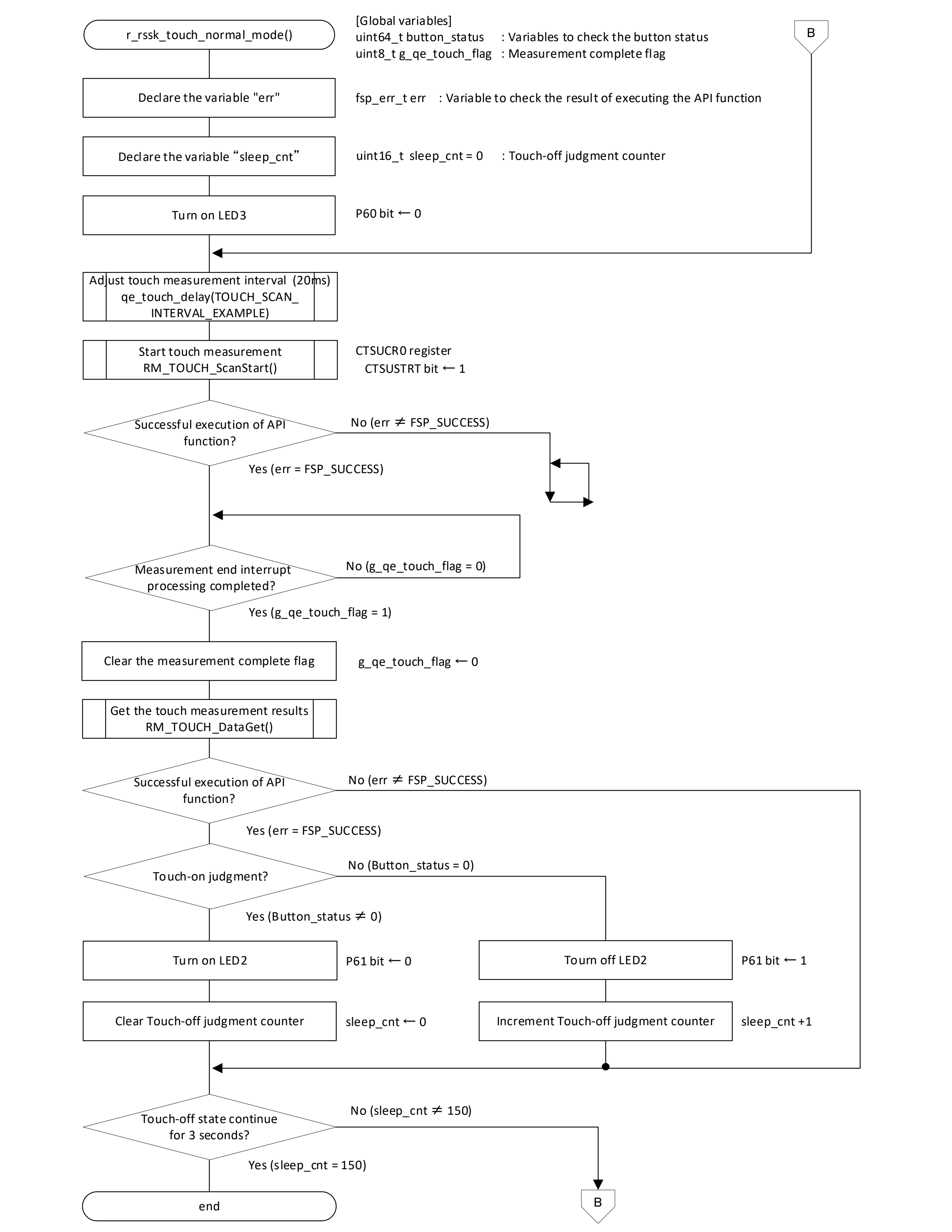


Figure 5-12 Normal Operation Mode Processing

6. Specification of Current Consumption Measurement

6.1 Environment to Measurement

Figure 6-1 shows the environment to measure current consumption.

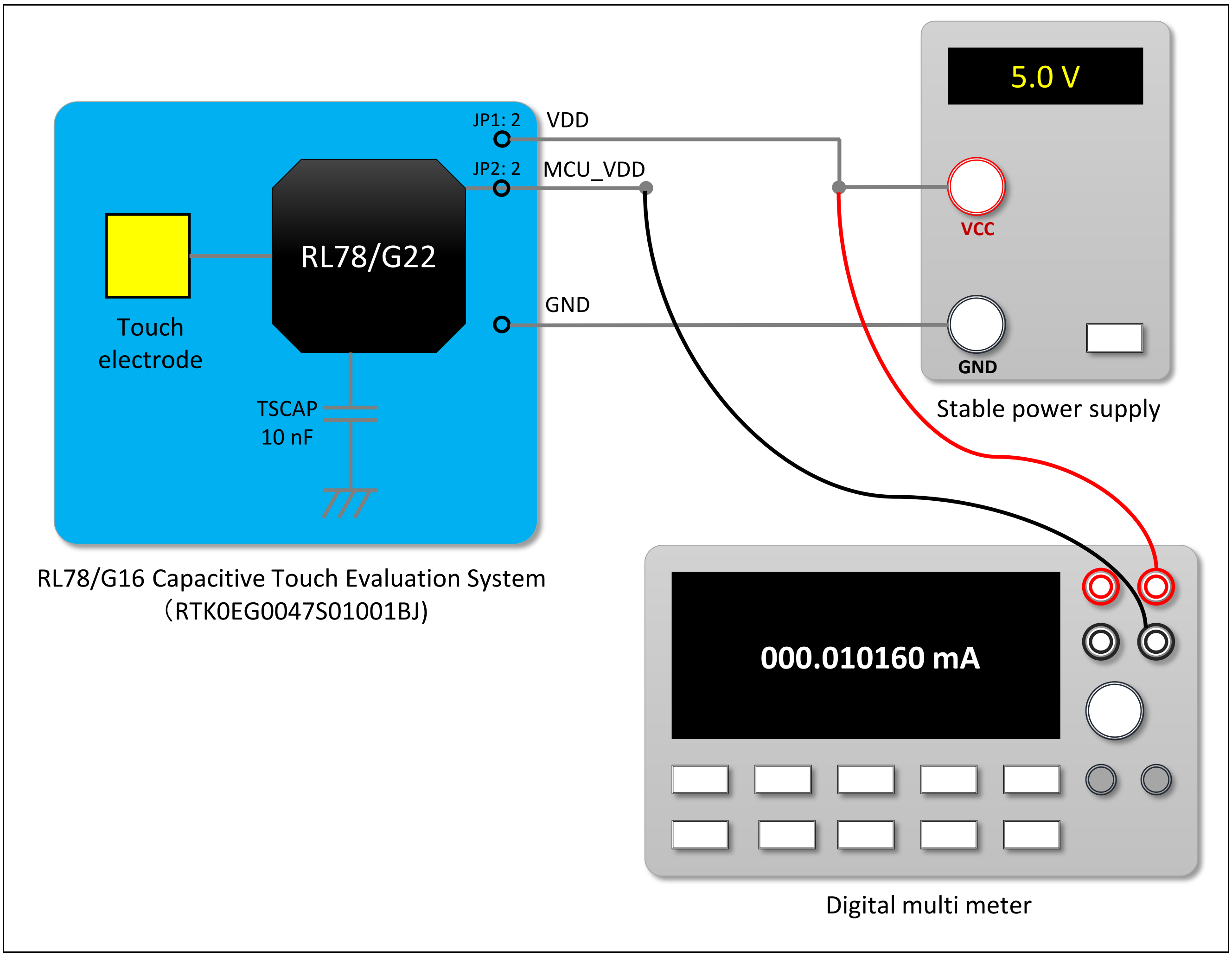


Figure 6-1 Environment to Measure Current Consumption

6.2 Equipment / Software

Table 6-1 shows the equipment and software used in current consumption measurement.

Table 6-1 Equipment / Software

Type	Name	Use
Digital multi meter	Keithley DMM7510	Measure current consumption.
Stable power supply	KENWOOD PWR18-2TP	Supply power to RL78/G16 Capacitive Touch Evaluation System (RTK0EG0047S01001BJ) CPU board.
Software	KickStart (V1.9.8.21)	Get result of current consumption measurement from Keithley DMM7510 and output the result to log-file.

6.3 CPU Board Jumper Settings

Table 6-2 shows the jumper settings of the CPU board of the RL78/G22 Capacitive Touch Evaluation System to measure current consumption.

Table 6-2 CPU Board Jumper Settings

Position	Circuit Group	Jumper	Use
JP1	VDD power	Shored pin 1-2	Power supply from USB connecter
JP2	MCU_VDD power	Open	Measure current consumption
SW4	Push Switch & LED / Application Header (CN1)	ON (2-3 pin, 5-6 pin)	Used to P61 for LED2 control

6.4 Setting of Unused Pins

Table 6-3 shows the settings specified in the Ports component of Smart Configurator to prevent current from flowing into unused pins during current consumption measurement by using RL78/G16 Capacitive Touch Evaluation System (RTK0EG0047S01001BJ). For any pins that also function as TS pins and that are not included in the following table, "Use" is selected in the CTSU module.

Table 6-3 Port Setting

Pin Number	Port Name	Destination of Connection	Setting	Remarks
4	P137	—	Input	Connect the pins to VDD via resistors
5	P122	CN1(pin number: 11 (LED_ROW1))	High Output	Set not to turn on the matrix LED on the electrode board
6	P121	CN1(pin number: 12 (LED_ROW3))	High Output	Set not to turn on the matrix LED on the electrode board
9	P60	LED3	High Output	Set not to turn on LED3
10	P61	LED2	High Output	Set not to turn on LED2
13	P12	CN1(pin number: 4 (LED_COL0))	High Output	Set not to turn on the matrix LED on the electrode board
14	P13	CN1(pin number: 5 (LED_COL1))	High Output	Set not to turn on the matrix LED on the electrode board
15	P14	CN1(pin number: 6 (LED_COL2))	High Output	Set not to turn on the matrix LED on the electrode board
16	P15	CN1(pin number: 7 (LED_COL3))	High Output	Set not to turn on the matrix LED on the electrode board
17	P41	CN1(pin number: 13 (LED_ROW0))	High Output	Set not to turn on the matrix LED on the electrode board

6.5 Settings of Current Consumption Measurement Software

Figure 6-2 shows the settings of current consumption measurement software (Keithley KickStart).

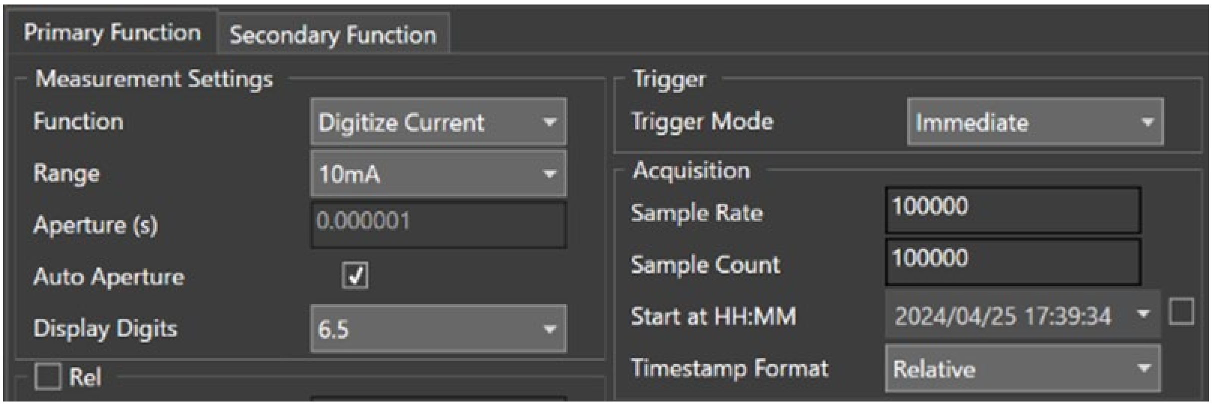


Figure 6-2 Settings of Current Consumption Measurement Software (Keithley KickStart)

7. Results of Current Consumption Measurement

This section shows a comparison of the results of current consumption measurement in low power consumption operation and those in normal operation.
In low power consumption operation, intermittent operation is performed in touch measurement by using the CTSU suspend function. In normal operation, touch measurement is always performed in Normal (normal operation) mode without using the SNOOZE function or intermittent operation. In both operations, current consumption is measured with LED2 and LED3 turned off.

7.1 Current Consumption Waveforms in Intermittent Operation

Figure 7-1 shows the current consumption waveforms in low power consumption operation with touch measurement every 100 ms.

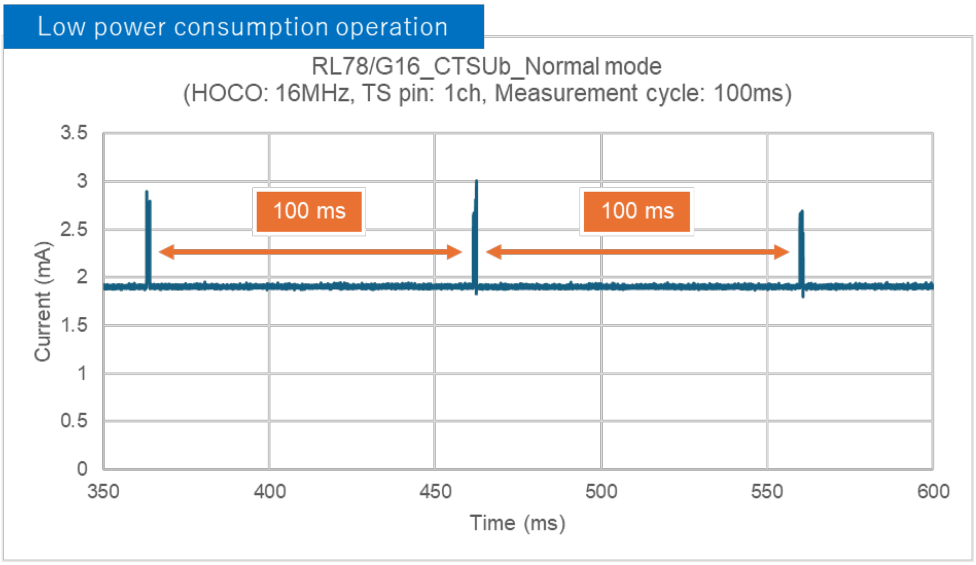


Figure 7-1 Current Consumption Waveforms with Touch Measurement every 100 ms (Low Power Consumption Operation)

Figure 7-2 shows the current consumption waveforms in normal operation with touch measurement every 100 ms.

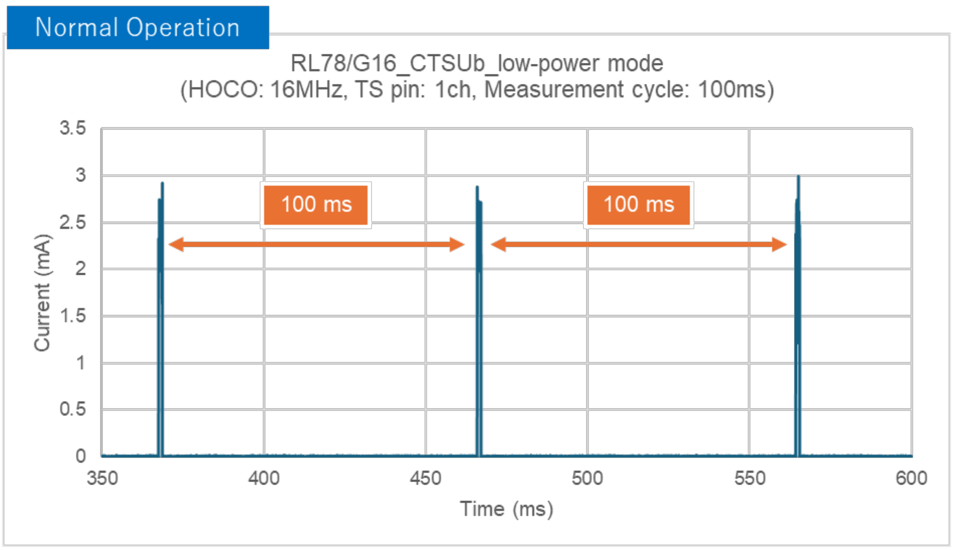


Figure 7-2 Current Consumption Waveforms with Touch Measurement every 100 ms (Normal Operation)

7.2 Current Consumption Waveforms During Touch Measurement Processing

Figure 7-3 shows the current consumption waveforms during touch measurement processing in low power consumption operation. **Table 7-1** shows the status of the CPU and CTSU in each measurement section. This figure shows the current consumption waveforms for a series of operations in which the CPU status transitions from STOP mode to SNOOZE mode to perform touch measurement processing, and then to Normal (normal operation) mode to perform touch measurement end processing + Touch On/Off judgment processing.

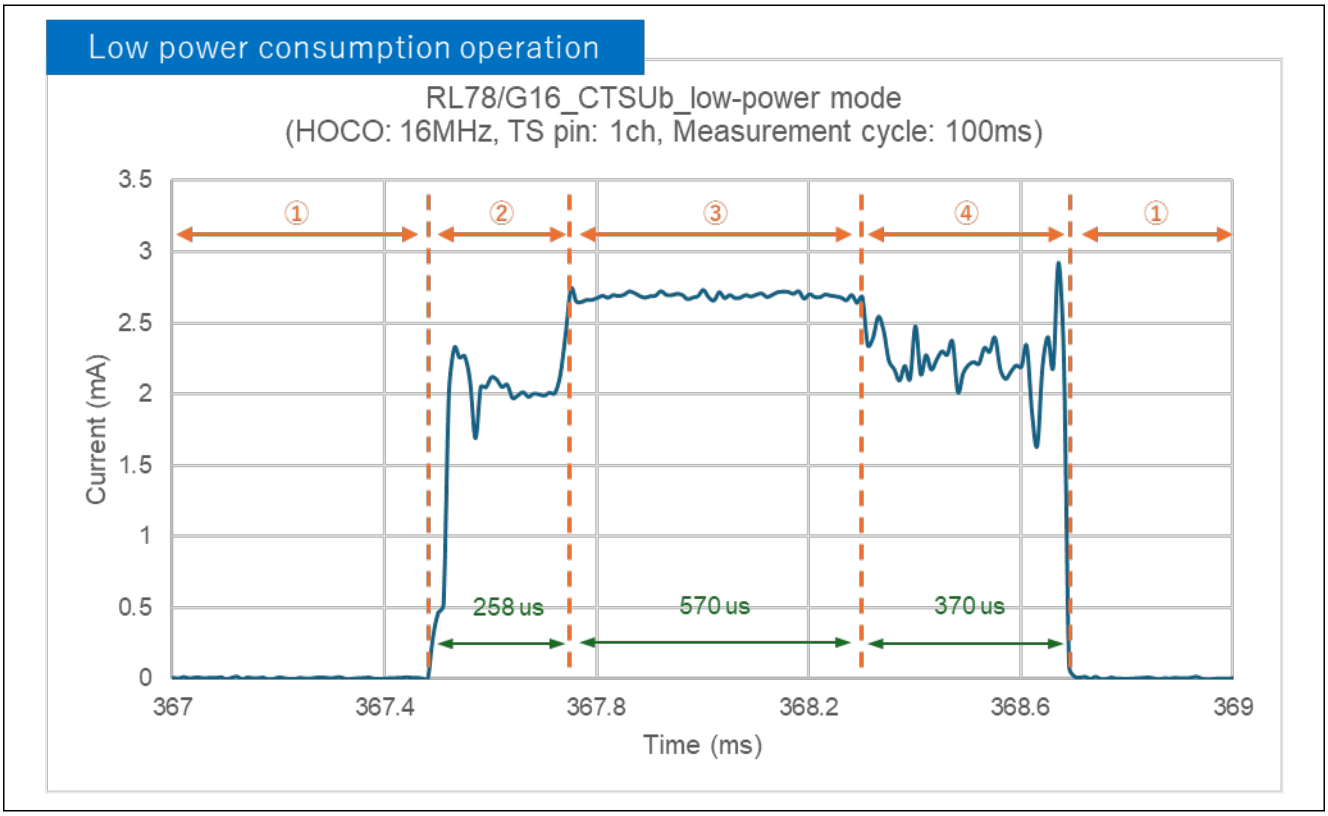


Figure 7-3 Current Consumption Waveforms during Touch Measurement Processing (Low Power Consumption Operation)

Table 7-1 Status of the CPU and CTSU in Each Measurement Section (Low power Consumption Operation)

Number in the Figure	Status of the CPU	Status of the CTSU
①	STOP mode	Suspend
②	Normal (Normal operation) mode	Measurement preparation
③	Normal (Normal operation) mode	Measurement operation
④	Normal (Normal operation) mode[Note 1]	Measurement standby[Note 2]

Note 1. The CPU performs calculation processing of measurement results.
Note 2. CTSU is stopped.

Figure 7-4 shows the current consumption waveforms during touch measurement processing in normal operation. **Table 7-2** shows the status of the CPU and CTSU in each measurement section. This figure shows the current consumption waveforms for a series of operations (touch measurement processing, touch measurement end processing, and touch on/off judgment processing) when the CPU is always running in Normal (normal operation) mode.

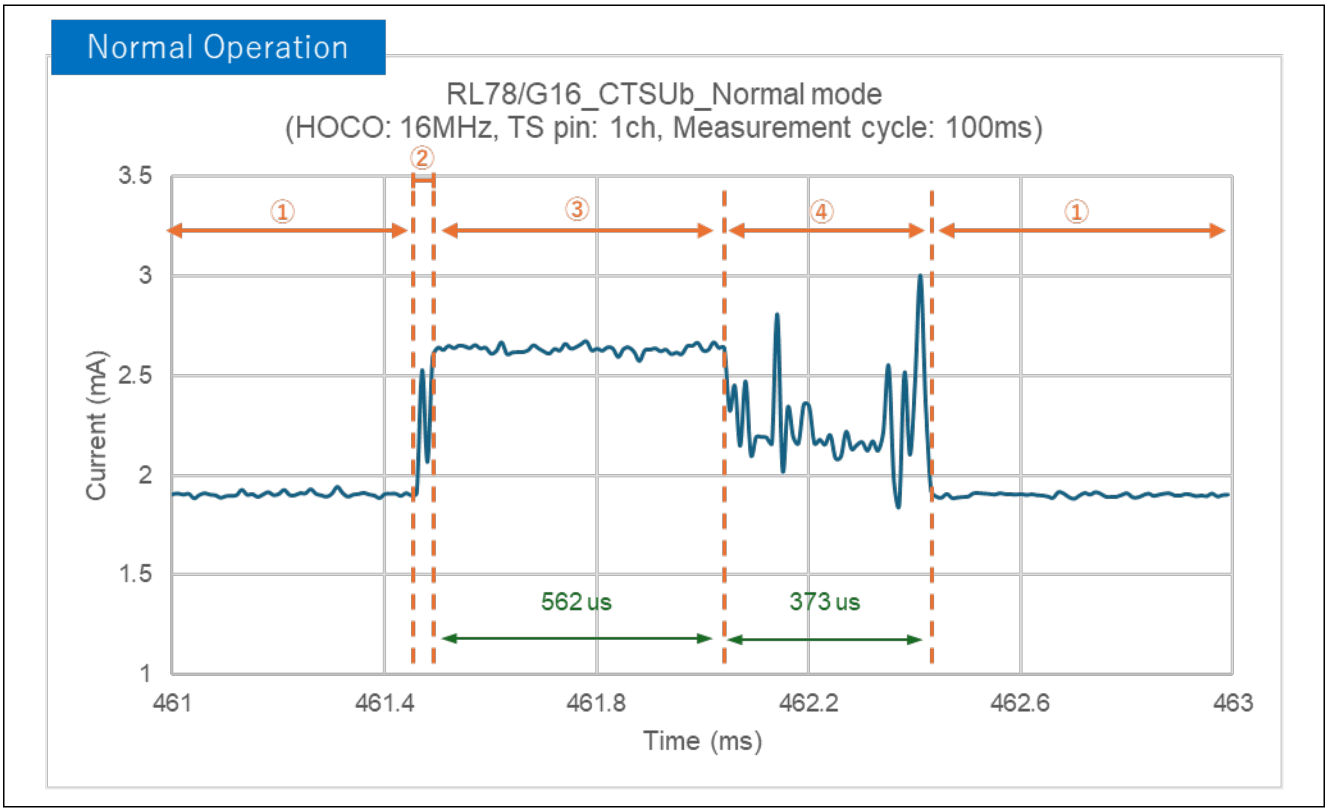


Figure 7-4 Current consumption waveforms during touch measurement processing (Normal operation)

Table 7-2 Status of the CPU and CTSU in each measurement section (Normal operation)

Number in the figure	Status of the CPU	Status of the CTSU
①	Normal (Normal operation) mode	Suspend
②	Normal (Normal operation) mode	Measurement preparation
③	Normal (Normal operation) mode	Measurement operation
④	Normal (Normal operation) mode[Note 1]	Measurement standby[Note 2]

Note 1. The CPU performs calculation processing of measurement results.
Note 2. CTSU is stopped.

7.3 Average Current Consumption per 100 ms

The following shows the average current consumption results obtained by the Keithley/KickStart software. shows the average current consumption for each CPU mode and the average current consumption per 100 ms in low power consumption operation.

Table 7-3 Average current consumption results (Low power consumption operation)

CPU mode	Time (ms)	Average current consumption (uA)
STOP mode	99.09	0.68
Normal (Normal operation) mode < touch measurement preparation >	0.26	1917
Normal (Normal operation) mode < touch measurement processing >	0.57	2682
Normal (Normal operation) mode < touch measurement end processing + touch on/off judgment processing >	0.37	2200
STOP mode + SNOOZE mode + Normal (Normal operation) mode	100	29.46

Average current consumption per 100 ms ≒ 29.46 uA

Table 7-4 shows the average current consumption per 100 ms in normal operation.

Table 7-4 Average current consumption results (Normal operation)

CPU mode	Time (ms)	Average current consumption (uA)
Normal (Normal operation) mode < touch measurement processing + touch measurement end processing + touch on/off judgment processing >	100	1908

Average current consumption per 100 ms ≒ 1.908 mA

Table 7-5 shows the difference in the average current consumption per 100 ms in each operation condition based on **Table 7-3** and **Table 7-4**.

Table 7-5 Difference in average current consumption for operation condition

Operation condition	Suspend function and intermittent operation	CPU mode transition	Average current consumption per 100 ms
Low power consumption operation	Used	STOP mode → SNOOZE mode → Normal (Normal operation) mode	29.46uA
Normal operation	Unused	Always in Normal (Normal operation) mode	1.908mA

Reduced to approximately 1/65

8. Sample Code

The sample code is available on the Renesas Electronics Website.

9. Usage Notes of the Sample Code (About the Timing of the result of touch judgement)

9.1 Effect of Debouncing Count of Touch-on Filter / Touch-off Filter

The timing when the touch measurement result is determined depends on the debouncing count of touch-on filter and touch-off filter (In QE for Capacitive Touch Ver. 4.0.0 and earlier, called the positive noise filter/negative noise filter). The touch measurement result is determined when the capacitance measurement value is either touch ON or touch OFF for a period of times (each filter setting value + 1 times) consecutively.

In the sample code, the debouncing count of touch-on filter and touch-off filter is set to "3" each. Therefore, the RM_TOUCH_DataGet function is called every touch measurement cycle to get the measurement result, and the touch measurement result is determined to be ON or OFF when the status of either touch ON or touch OFF is four times consecutively.

For details, refer to the application Note RL78 Family TOUCH Module Software Integration System [R11AN0485].

9.2 Effect of Moving Average Filter

The timing when the touch measurement result is determined depends on the setting of the moving average filter. In the sample code, the number of times the moving average is set to "4".

The number of times the moving average is set by the variable "num_moving_average" in the qe_touch_config.c file.

Figure 9-1 shows the example of moving average processing operation.

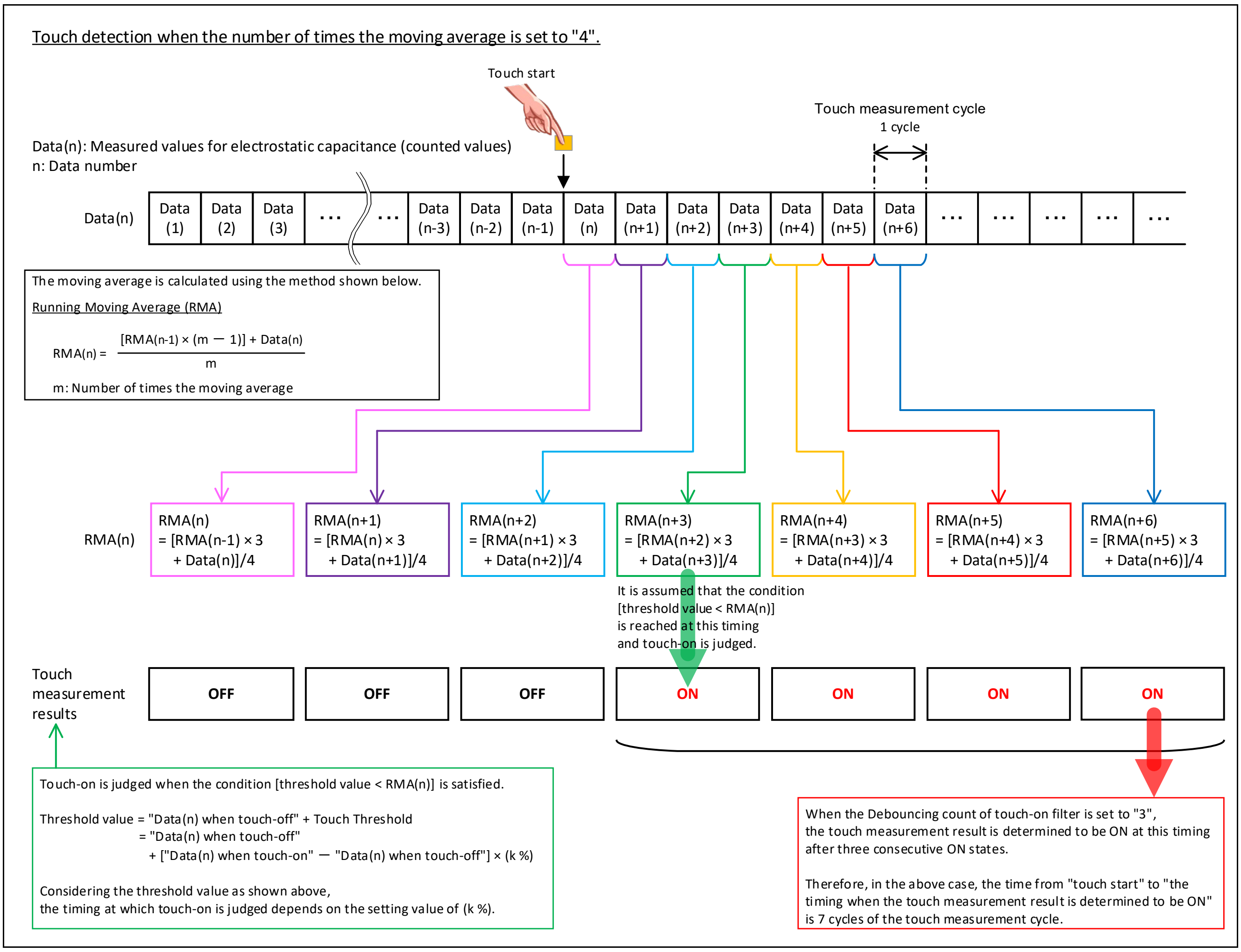


Figure 9-1 Example of Moving Average Processing Operation

10. Documents for Reference

○User's Manual

- RL78/G16 User’s Manual: Hardware [R01UH0978]
- RL78 Family User's Manual: Software [R01US0015]

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

○Technical Updates/Technical Brochures

(The latest information of the documents is available on the Renesas Electronics Website.)

○User's Manual: Development Tools

- RL78/G16 Capacitive Touch Evaluation System (RTK0EG0047S0100BJ) [R12UZ0110]

(The latest information of the documents is available on the Renesas Electronics Website.)

○Application Note

- Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide [R30AN0424]
- Capacitive Sensor Microcontrollers CTSU Capacitive Touch Electrode Design Guide [R30AN0389]

- RL78 Family Using QE and SIS to Develop Capacitive Touch Applications [R01AN5512]
- RL78 Family CTSU Module Software Integration System [R11AN0484]
- RL78 Family TOUCH Module Software Integration System [R11AN0485]

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

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Capacitive Sensing Unit related page
https://www.renesas.com/solutions/touch-key
https://www.renesas.com/qe-capacitive-touch

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.31.25	-	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. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.