# RL78/G23

## Capacitive Touch Low Power Guide (SNOOZE Mode Function)

### Introduction

This application note describes the method for implementing low-power operation in touch measurement by using the capacitive sensing unit (CTSU2L) of the RL78/G23. In this method, a combination of intermittent operation of the CPU under control of the 32-bit interval timer function (TML32) and the SNOOZE mode function of the CTSU2L is applied to achieve low power consumption in touch measurement.

### Target Device

RL78/G23

When implementing a touch application, extensively evaluate the touch function in terms of the target environment.

### Using the Low Power Consumption Application Notes

There are two ways of implementing low-power operation in touch measurement by using the capacitive sensing unit (CTSU2L) of the RL78/G23.

a. Only using the SNOOZE mode function
b. Using both the SNOOZE mode function and SNOOZE mode sequencer (SMS)

Use of the SMS function can further reduce power consumption in comparison with touch measurement by using only the SNOOZE mode function. However, this method keeps the SMS exclusively occupied in executing the processing for touch measurement so that the SMS is not available to execute other processing. Therefore, if you also want to use the SMS for other processing or to implement low-power operation with simple settings instead of using the SMS, read this application note.

If you want to use the SMS in implementing low-power operation, refer to the following application notes for details on the procedure for setting up the SMS for use.

- Sample code for using the SNOOZE mode function in combination with the SMS
  RL78/G23 Capacitive Touch Low Power Guide (SMS function) [R01AN6670]

- Setting up to use the SNOOZE mode function in combination with the SMS
  RL78 Family Capacitive Touch Low Power Application Development using SMS [R01AN7261]

## Contents

# 1. Overview

This application note describes the method for achieving low power consumption by using the SNOOZE mode function of the RL78/G23 in performing intermittent capacitive touch measurement operations. This application note also indicates reference current consumption when touch measurement proceeds in a 100-ms cycle. In the system described in this application note, power consumption is reduced to approximately 1/238 of that for a case where touch measurement is performed by the CPU without using intermittent operation.

Remark The SNOOZE mode function allows the starting of peripheral modules by external triggers in STOP mode without activating the CPU.

## 1.1 Assumed System

The low-power capacitive touch operation in the sample code is written on the assumption of the system shown in the red box in Figure 1-1. While the main system is on standby, measurement is only done for one capacitive touch button (the power button) at fixed intervals to determine whether a touch is detected. The system only begins normal operation when touch-on is detected for the power button.

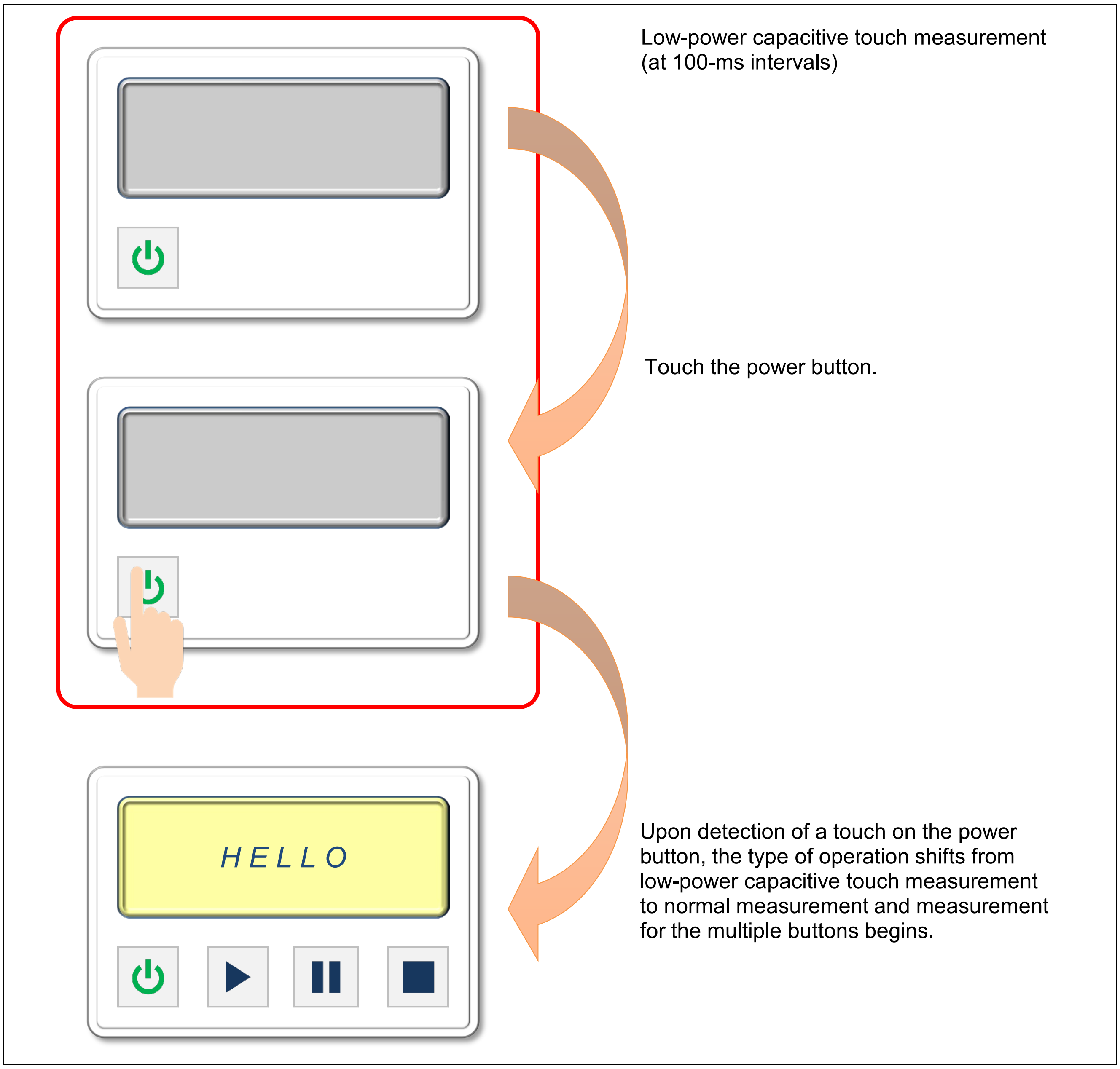



**Figure 1-1 Assumed System**

# 2. Touch Measurement Mechanism

## 2.1 Flow of Operation for Low-Power Touch Measurement (SNOOZE Mode Function)

This section provides a summary of the linkage of peripheral module operations and describes how power consumption is reduced when the SNOOZE mode function is used to control intermittent operations for touch measurement.

Figure 2-1 shows an image of touch measurement operations with low power consumption (SNOOZE mode function).

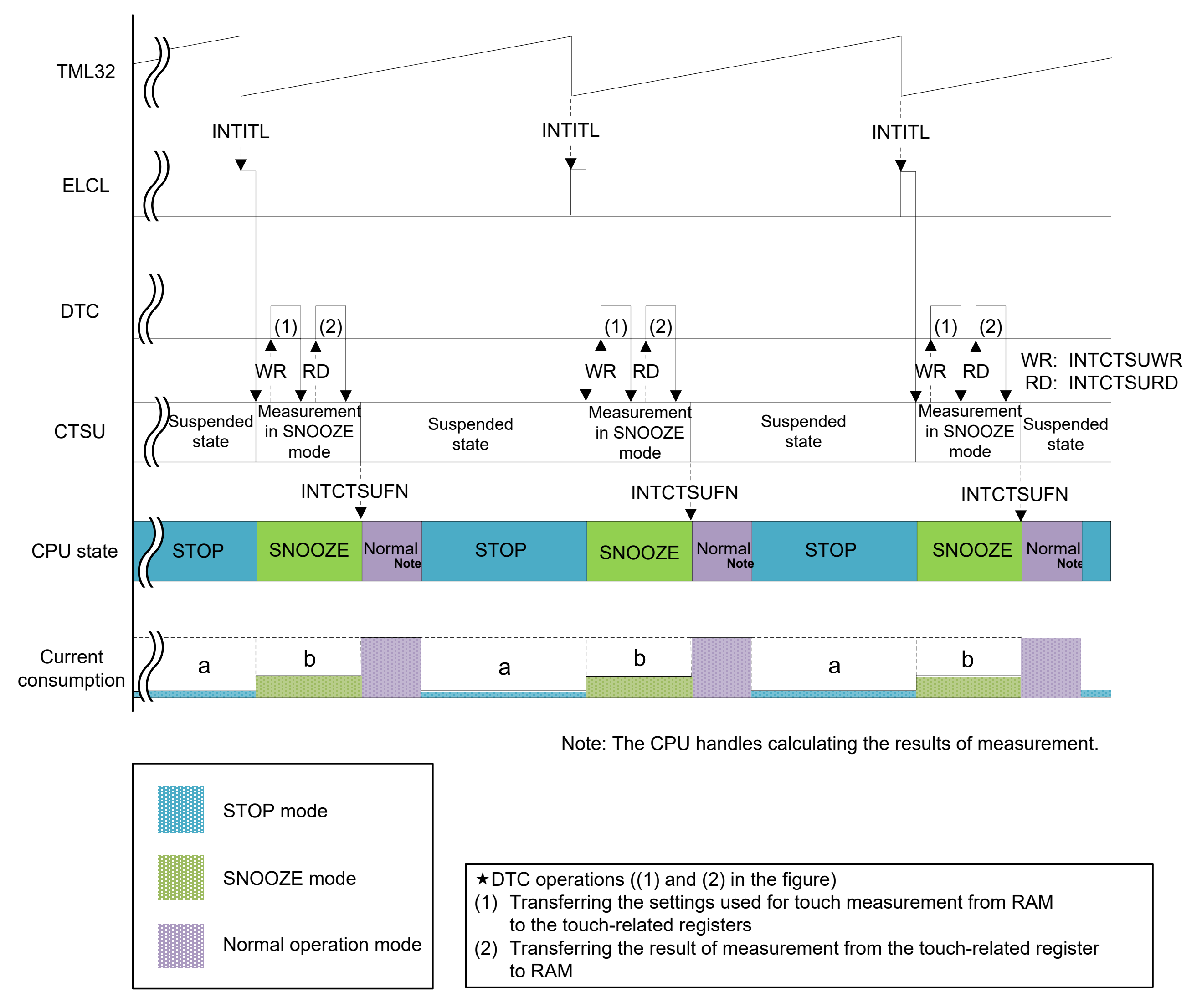



**Figure 2-1 Touch Measurement with Low Power Consumption (SNOOZE Mode Function)**

■ Overview of operations

The 32-bit interval timer (TML32) is used to control the timing of touch measurement.

1. Upon reception of an interrupt request signal from TML32 (INTITL), the CTSU makes a transition from the suspended state (waiting for an external trigger) to measurement in SNOOZE mode. The CPU makes a transition from STOP mode to SNOOZE mode at this time.

2. Upon reception of a request for writing settings to registers for each channel in the CTSU (INTCTSUWR), the DTC transfers the register settings used for touch measurement from RAM to the touch-related registers ((1) in the figure). The CTSU starts touch measurement in the corresponding channel according to the transferred register settings.

3. Upon reception of a measurement data transfer request from the CTSU (INTCTSURD), the DTC transfers the result of measurement from the touch-related register to RAM ((2) in the figure). Steps 2 and 3 are repeated the number of times corresponding to the number of measurement channels in the CTSU.
4. When the CTSU has completed measurements for all channels, it issues a CTSU measurement end interrupt request (INTCTSUFN) and enters the suspended state (waiting for a trigger). At this time, the CPU makes a transition from SNOOZE mode to Normal operation mode and starts operating.
5. Software performs touch-on/off judgment processing based on the results that have been read.
6. When the STOP instruction is executed, the CPU makes a transition from Normal operation mode to STOP mode. The CTSU then waits for the next interrupt request signal (INTITL).
7. Steps 1 to 6 are repeated.

■ Reduction of the current consumption

- While touch measurement is not being executed, stopping the CPU operation by shifting the CPU to STOP mode can reduce the current consumption in period “a” in the figure in comparison with normal operation.
- Touch measurement in SNOOZE mode without activating the CPU can reduce the current consumption in period “b” in the figure in comparison with touch measurement during normal operation.

## 2.2 Peripheral Modules Used in Implementing Low-Power Touch Measurement (SNOOZE Mode Function)

Table 2-1 lists the peripheral modules used in handling intermittent operation in touch measurement through the SNOOZE mode function and the functions of the modules.

**Table 2-1 Peripheral Modules and their Functions**

| Peripheral Module | Function |
|---|---|
| Capacitive sensing unit (CTSU2L) | Measures the capacitance generated on touch electrodes.<br>The following settings are made to obtain low-power operation for the unit.<br>• Selecting an external trigger<br>• Enabling the SNOOZE mode function |
| Data transfer controller (DTC) | Transfers the settings used for touch measurement from RAM to the touch-related registers. On completion of each round of touch measurement, the DTC transfers the result of measurement (counted value) from the touch-related register to RAM. |
| 32-bit interval timer (TML32) | Timer for counting the cycles of touch measurement.<br>This timer generates an interrupt request signal (INTITL) at 100-ms touch measurement intervals. |
| Logic and event link controller (ELCL) | Specifies the TML32 interrupt request signal (INTITL) as the external trigger for the CTSU. |

# 3. Environment and Conditions for Verifying Operation

Table 3-1 lists the elements of the environment for verifying operation. Table 3-2 lists the conditions for verifying operation.

**Table 3-1 Environment for Verifying Operation**

| Item | Description |
|---|---|
| Microcontroller used | RL78/G23 (R7F100GSN2DFB) |
| Operating frequency | • Main system clock<br>High-speed on-chip oscillator clock ($f_{IH}$): 32 MHz<br>• CPU and peripheral hardware clock ($f_{CLK}$): 32 MHz<br>• Subsystem clock<br>Low-speed on-chip oscillator clock ($f_{IL}$): 32.768 kHz<br>• Low-speed peripheral clock frequency ($f_{SXP}$): 32.768 kHz |
| Operating voltage | 5.0 V<br>Level of voltage detection by LVD0 in reset mode<br>For rising: 2.67 V typ. (2.59 V to 2.75 V)<br>For falling: 2.62 V typ. (2.54 V to 2.70 V) |
| Target board | RL78/G23 capacitive touch evaluation system<br>Product code: RTK0EG0030S01001BJ |
| Integrated development environment ($e^2$ studio) | $e^2$ studio (2025-01)<br>from Renesas Electronics Corporation |
| Smart Configurator | V25.1.0<br>from Renesas Electronics Corporation |
| C compiler ($e^2$ studio) | CC-RL V1.15.00<br>from Renesas Electronics Corporation |
| QE for Capacitive Touch | V4.1.0<br>from Renesas Electronics Corporation |

**Table 3-2 Conditions for Verifying Operation**

| Item | Description |
|---|---|
| Touch measurement cycle | 100 ms |
| Sensor driving pulse frequency | 2.0 MHz |
| Touch sensor (TS pin) | TS06 |
| Judgment method | Value majority mode (VMM) |
| Measurement mode | Self-capacitance method<br>(MD1 bit of the CTSUCRAL register = 0) |
| Scan mode | Multi-scan mode<br>(MD0 bit of the CTSUCRAL register = 1) |
| Measurement start trigger | External trigger (ELCL)<br>(CAP bit of the CTSUCRAL register = 1) |
| Enabling or disabling the SNOOZE mode function | The SNOOZE mode function is enabled.<br>(SNZ bit of the CTSUCRAL register = 1) |
| Boost power | Boost power on<br>(PUMPON bit of the CTSUCRAL register = 1) |
| Measurement power-supply | Measurement power-supply voltage = 1.5 V<br>(ATUNE0 bit of the CTSUCRAL register = 0) |
| Measurement power-supply current adjustment | 40 μA<br>(ATUNE1 bit of the CTSUCRAL register = 1,<br>ATUNE2 bit of the CTSUCRAH register = 0) |
| Non-measured pin output selection | Low-level output<br>(POSEL[1:0] bits of the CTSUCRAH register = 00b) |
| Sensor driving pulse selection | High-resolution pulse mode<br>(SDPSEL bit of the CTSUCRAH register = 1) |
| Sensor stabilization wait time | 32 cycles<br>(SST[7:0] bits of the CTSUCRBL register = 0x1f) |
| Measurement count setting | 7<br>(SNUM[7:6] bits of the CTSUSO1 register = 00b,<br>SNUM[5:0] bits of the CTSUSO0 register = 00111b) |

# 4. Hardware Descriptions

## 4.1 Example of Hardware Configuration

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

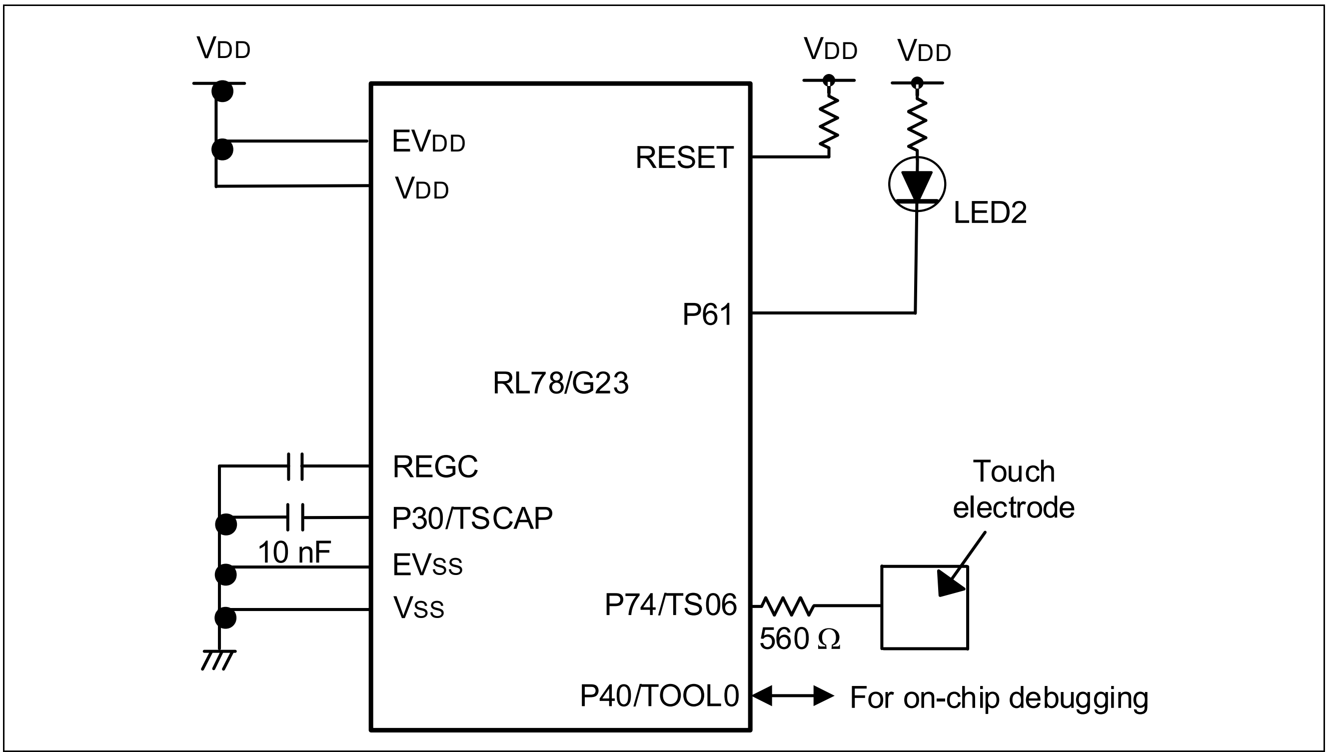



**Figure 4-1 Hardware Configuration**

Caution 1. This circuit diagram is simplified to show only an overview of the relevant connections. In actually designing the circuit for a user board, appropriately handle the pins such that the electrical characteristics are satisfied (connect each input-only port pin to VDD or VSS via a resistor).

Caution 2. Connect the pins having names beginning with EVSS to VSS and the pins having names beginning with EVDD to VDD.

Caution 3. VDD must not be lower than the reset release voltage (VLVD0) that is specified for LVD0.

## 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 |
|---|---|---|
| P74/TS06 | Input/output | Pin for use in measuring capacitance |
| P61 | Output | Pin for controlling LED2 |
| P30/TSCAP | — | Pin to connect a secondary power capacitor for use in measurement |

Caution This application note only describes handling of the pins that are to be used. In actually designing the circuit for a user board, appropriately handle all pins such that the electrical characteristics are satisfied.

# 5. Software Descriptions

## 5.1 Outline of Sample Code Operations

The sample code performs the following operations that correspond to low-power capacitive touch measurement as shown in Figure 1-1, Assumed System, in chapter 1.

1. After release from the power-on reset state, the RM_TOUCH_Open function is executed to initialize the capacitive sensing unit (CTSU).

2. The ELCL event source is set to channel 0 compare match (INTITL0) of the 32-bit interval timer (TML32). The event link destination is set as the CTSU.

3. After the touch measurement settings are made and the SNOOZE mode function is enabled by executing the RM_TOUCH_ScanStart function[Note], the CTSU is placed in the suspended state and waits for an external trigger.

4. Counting in the TML32 timer is started to time the 100-ms measurement intervals.

5. The CPU is placed in STOP mode by execution of the STOP instruction.

6. When the TML32 interrupt request (INTITL0) is generated, the CTSU is started by the external trigger from the ELCL that is generated in response. The CPU is then placed in SNOOZE mode and the CTSU starts touch measurement.

7. When the CTSU ends measurement, it generates a measurement end interrupt (INTCTSUFN) and enters the measurement standby state. The CPU enters Normal operation mode.

8. The results of measurement are obtained and judgment of the touch-on or touch-off state is made by execution of the RM_Touch_DataGet function. An LED (LED2) is turned on when touch-on is detected or turned off when touch-off is detected.

9. Steps 5 to 8 are repeated.

Note In the processing of the RM_TOUCH_ScanStart function, the CAP bit of the CTSUCRAL register is set to 1, the SNZ bit is set to 1, and the STRT bit is then set to 1. For details on setting conditions, refer to the description of the SNZ bit in section 30.2.3, CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH), in the RL78/G23 User's Manual: Hardware (R01UH0896).

## 5.2 Smart Configurator Settings

### 5.2.1 Components Used

Figure 5-1 shows the components used by the sample code as they appear in the Smart Configurator.

| 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) |
| ELCL Flexible Circuit | 1.0.0 | Config_ELCL(used) |
| Interval Timer | 1.6.0 | Config_ITL000(ITL000: used) |
| Ports | 1.6.0 | Config_PORT(PORT: used) |
| Touch middleware. (rm_touch) | 2.10 | rm_touch(used) |
| Voltage Detector | 1.5.0 | Config_LVD0(LVD0: used) |

**Figure 5-1 Components Used as Shown in the Smart Configurator**

### 5.2.2 Component Settings

Table 5-1 shows the settings for each component in the sample code.

**Table 5-1 Settings of Components**

| Component | Setting | Description |
|---|---|---|
| Capacitive sensing unit driver (CTSU) | r_ctsu | Set the DTC for data transfer in response to the generation of an interrupt request (INTCTSUWR or INTCTSURD). Specify use of the TSCAP pin and all TS pins. |
| Touch middleware | rm_touch | Default settings |
| ELCL flexible circuit[Note] | Config_ELCL | Select the 32-bit interval timer as the input source and the required external trigger (CTSU hardware trigger) as the output signal. Select L1L0 as the logic cell block to be used. |
| Interval timer | Config_ITL000 | Set the operating clock to fSXP, the clock frequency to fITL0/128, and the interval to 100 ms. |
| Ports | Config_PORT | Set P61 as an output initially at the high level. |
| Voltage detector | Config_LVD0 | Set the voltage for generating a reset to 2.62 V. |

Note The ELCL flexible circuit is supported by the graphical configurator in V. 24.1.0 and later versions of the Smart Configurator, so the settings of the logic and event link controller (ELCL) can be made in a graphical manner.
Figure 5-2 shows the settings for the ELCL on the [Components] tabbed page.

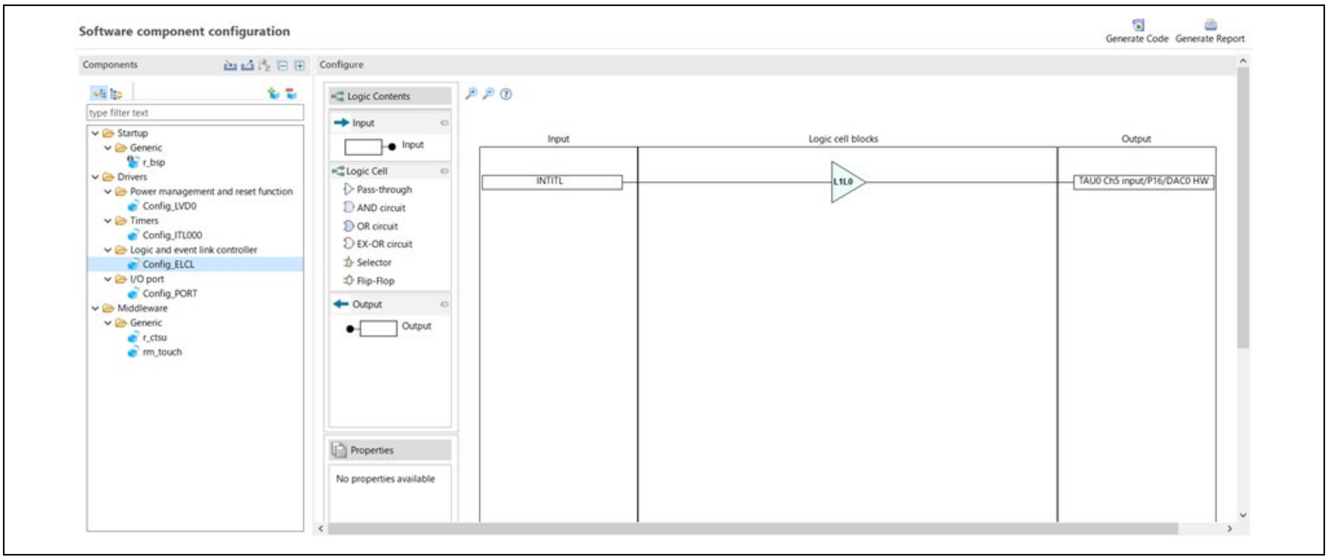



**Figure 5-2 Settings of the ELCL**

Remark For details of how to set up the ELCL, open [Help Contents] in the e² studio, select [e2 studio User Guide] → [Building Projects] → [Smart Configurator for RL78], and refer to the contents of "Adding Graphical Configurator Module".

When using CS+, refer to the equivalent contents from [Help] in the Smart Configurator.

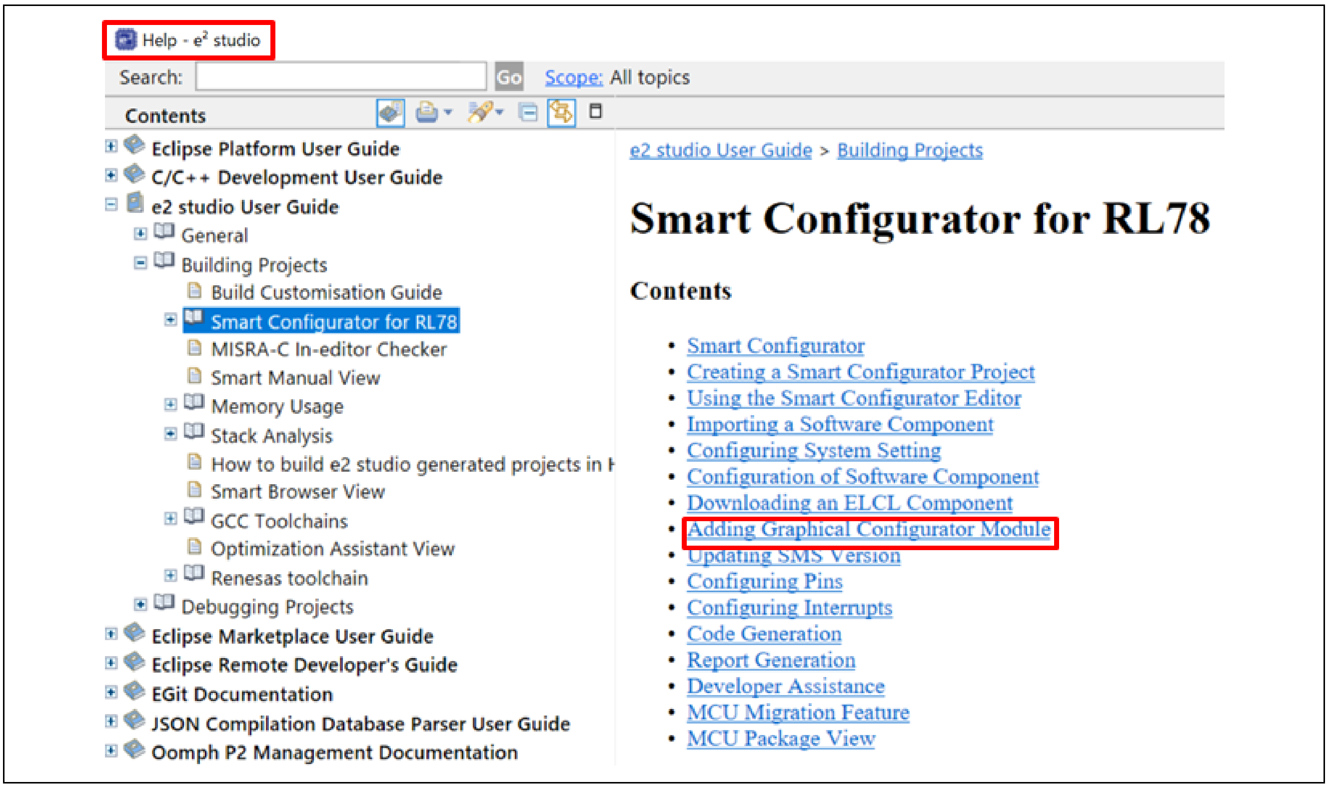



**Figure 5-3 Help Page on How to Use the Graphical Configurator for ELCL Settings (e² studio)**

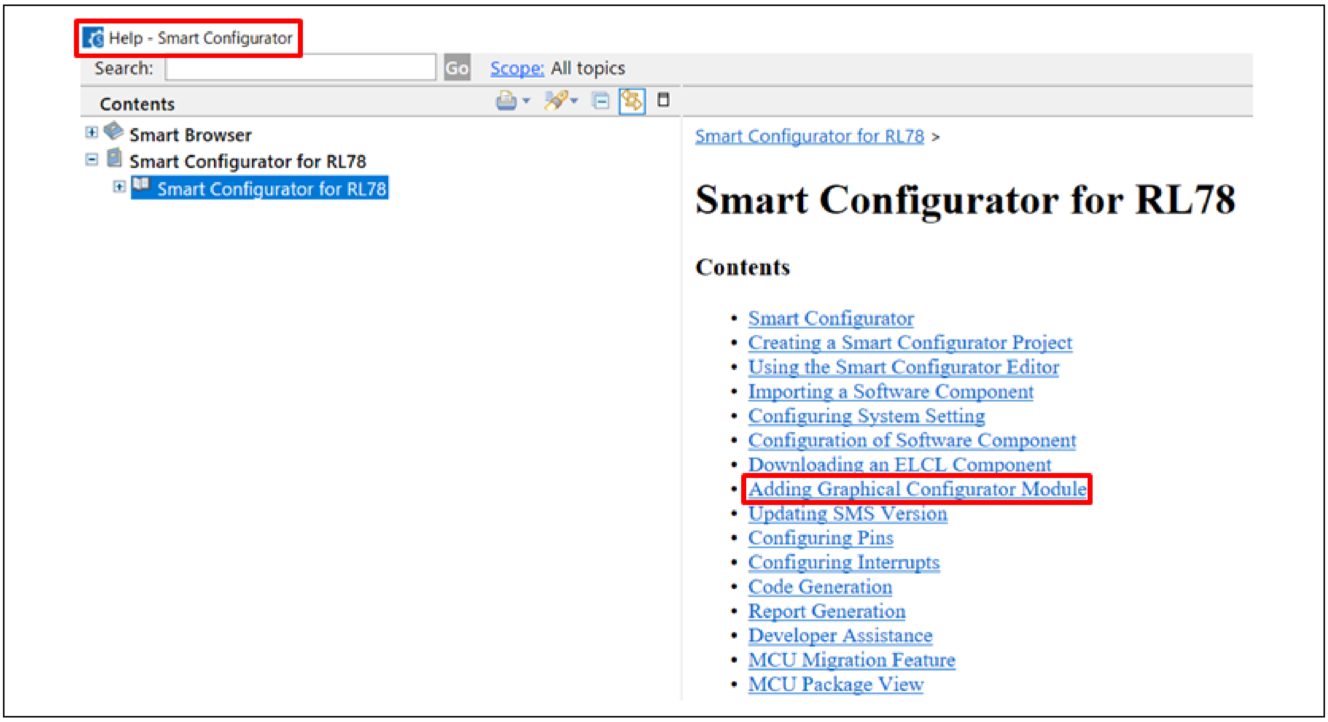



**Figure 5-4 Help Page on How to Use the Graphical Configurator for ELCL Settings (CS+)**

## 5.3 Capacitive Touch Settings

Figure 5-5 shows the touch interface configuration. Figure 5-6 shows the results of QE tuning. Measurement is done through TS06 with the self-capacitance method.

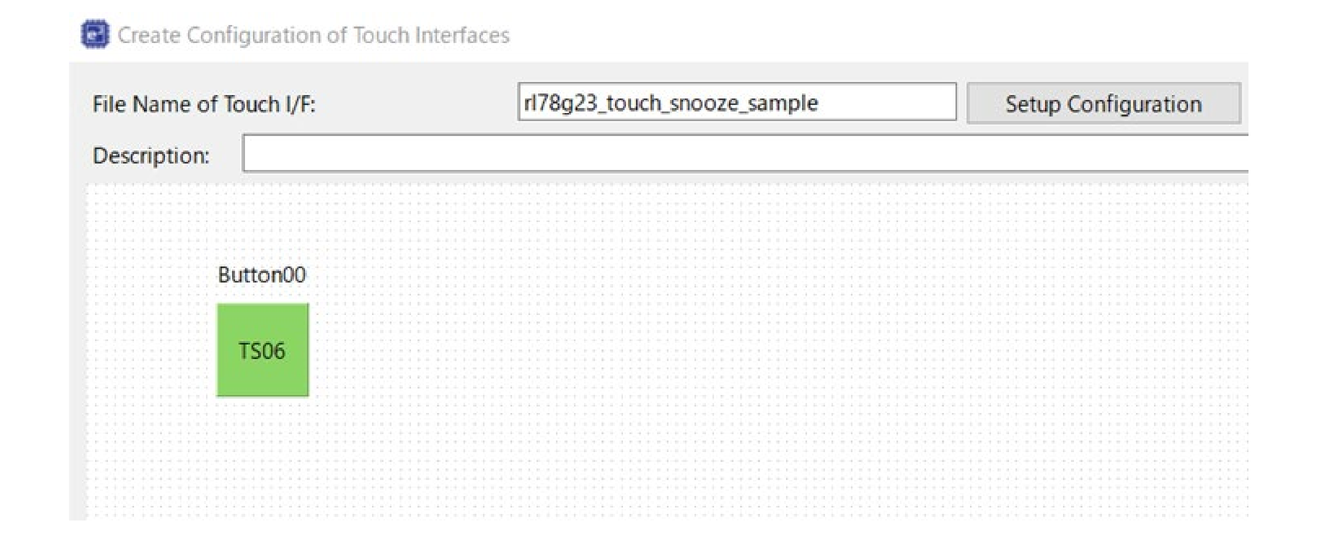
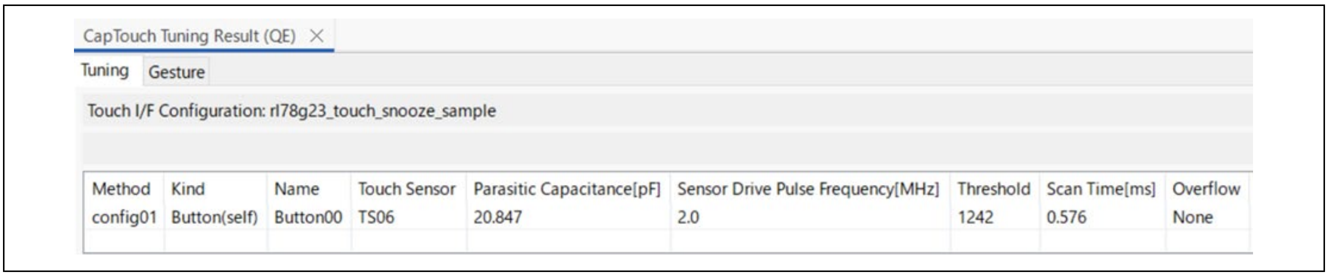



**Figure 5-5 Touch Interface Configuration**




| Method | Kind | Name | Touch Sensor | Parasitic Capacitance[pF] | Sensor Drive Pulse Frequency[MHz] | Threshold | Scan Time[ms] | Overflow |
|---|---|---|---|---|---|---|---|---|
| config01 | Button(self) | Button00 | TS06 | 20.847 | 2.0 | 1242 | 0.576 | None |

**Figure 5-6 Results of QE Tuning**

## 5.4 CTSU Startup Setting

Select "Use an external trigger" in QE for Capacitive Touch as shown in Figure 5-7. This setting enables the SNOOZE mode function of the CTSU2L, which reduces the power consumption as a standby state.

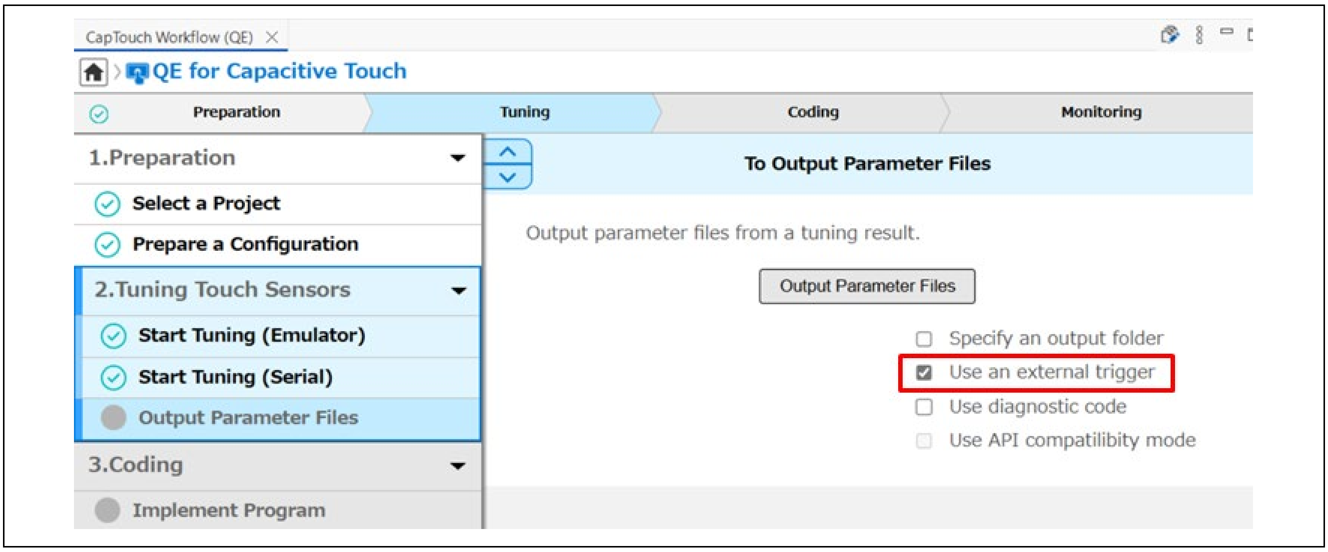



**Figure 5-7 Selecting an External Trigger in QE for Capacitive Touch**

## 5.5 Settings of the Option Bytes

Table 5-2 shows the settings of the option bytes.

**Table 5-2 Settings of the Option Bytes**

| Address | Setting | Description |
|---|---|---|
| 000C0H or 040C0H | 11101111B (0xEF) | Stops the watchdog timer[Note].<br>Counting is stopped after release from the reset state. |
| 000C1H or 040C1H | 11111100B (0xFC) | Level of voltage detection by LVD0 in reset mode<br>For rising: 2.67 V typ. (2.59 V to 2.75 V)<br>For falling: 2.62 V typ. (2.54 V to 2.70 V) |
| 000C2H or 040C2H | 11101000B (0xE8) | HS mode<br>High-speed on-chip oscillator clock ($f_{IH}$): 32 MHz |
| 000C3H or 040C3H | 10000100B (0x84) | Enables on-chip debugging |

Note: The setting for the watchdog timer is in the list of configuration items for r_bsp on the [Components] tabbed page. Do not change it from the default setting ("Unused").

## 5.6 Folder and File Configuration

Table 5-3 lists the sample code folders generated by the Smart Configurator and QE for Capacitive Touch. Table 5-4 lists the files of sample code in which contents have been added or changed from the generated code.

**Table 5-3 Sample Code Folders**

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

**Table 5-4 Files of Sample Code in Which Contents have been Added or Changed**

| File Name | Outline | Changed or Added Contents |
|---|---|---|
| rl78g23_rssk_snooze_sample.c | Main processing | The following processing was added.<br>• Enabling of maskable interrupts<br>• A call of the function for touch measurement control processing |
| mcu_clocks.c | Clock setting | The setting of the CMC register was changed. (For handling of unused pins P123 and P124) |
| qe_touch_sample.c | Touch measurement control processing | The following processing was added.<br>• Processing to reduce power consumption<br>• LED control processing<br>The setting of the CSC register was changed. (For handling of unused pins P123 and P124) |

## 5.7 Details of the Sample Code

### 5.7.1 List of Variables

Table 5-5 lists the global variables that are used in this sample code.

**Table 5-5 Global Variables**

| Type | Variable Name | Description |
|---|---|---|
| uint64_t | button_status | Variable to check the button state |
| uint8_t | g_qe_touch_flag | Measurement completion flag |

### 5.7.2 List of Functions

The specifications of the functions added or changed in this sample code are shown below.

main()

| | | |
|---|---|---|
| **Synopsis** | Main processing | |
| **Declaration** | void main(void) | |
| **Explanation** | Calls the qe_touch_main() function. | |
| **Argument** | — | — |
| **Return value** | — | |

qe_touch_main()

| | | |
|---|---|---|
| **Synopsis** | Touch measurement control processing | |
| **Declaration** | void qe_touch_main(void) | |
| **Explanation** | Configures the touch measurement settings to use the SNOOZE mode function and then controls the touch measurement.<br>Touch measurement is repeated and a result of measurement is obtained each time, and LED2 is then turned on if touch-on is detected or off if touch-off is detected. | |
| **Argument** | — | — |
| **Return value** | — | |

### 5.7.3 Flowcharts

#### 5.7.3.1 Main Processing

Figure 5-8 shows the flowchart for the main processing.

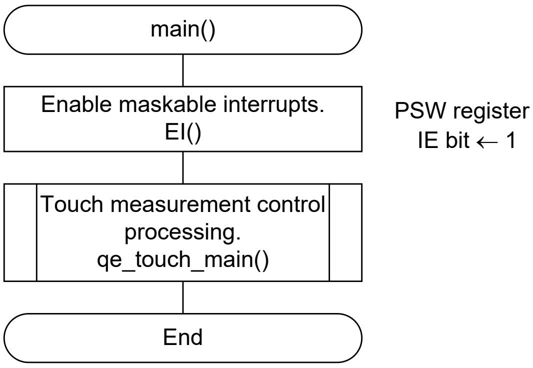



**Figure 5-8 Main Processing**

#### 5.7.3.2 Touch Measurement Control Processing

Figure 5-9 and Figure 5-10 are the flowchart for touch measurement control processing.

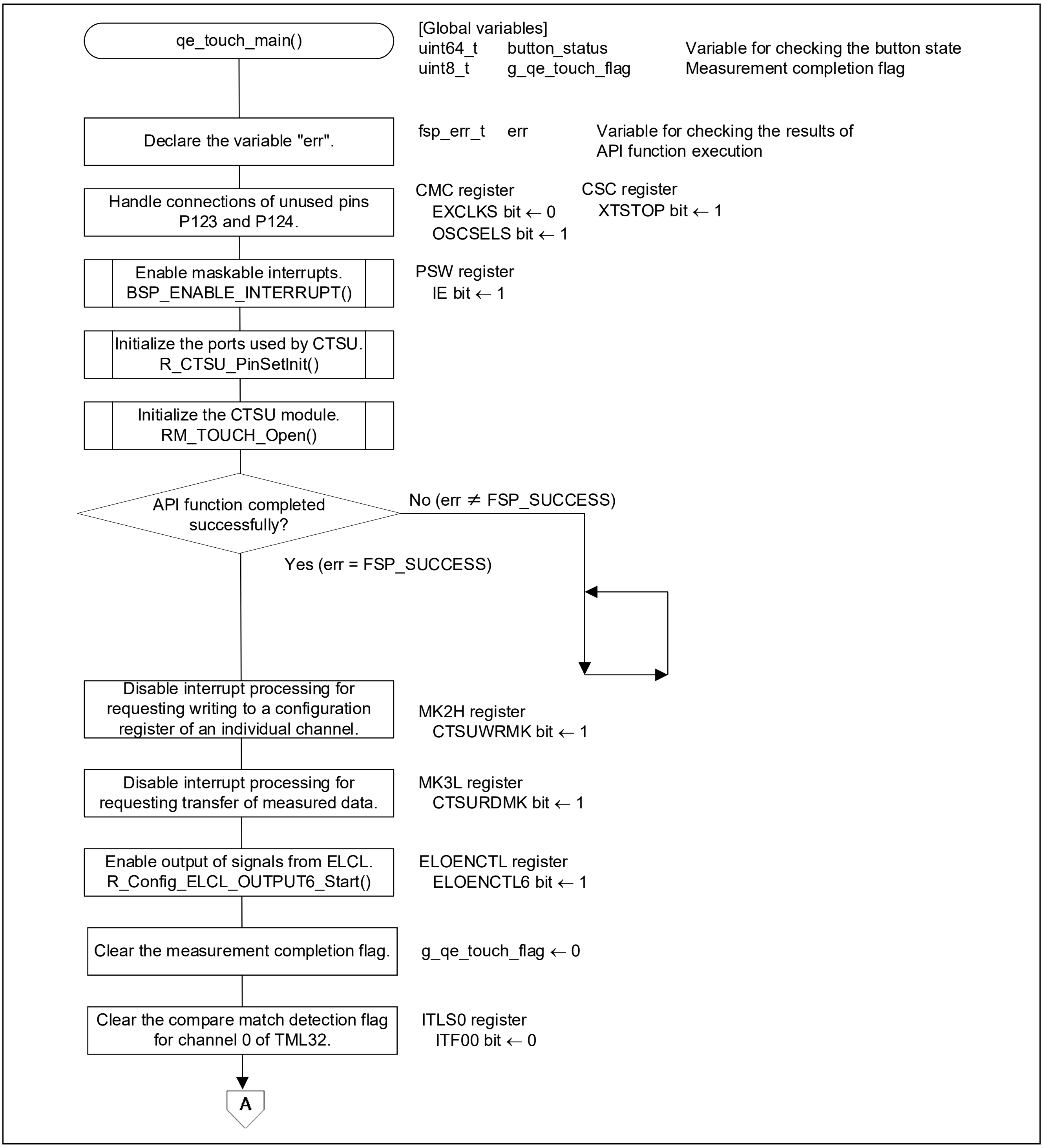



**Figure 5-9 Touch Measurement Control Processing (1/2)**

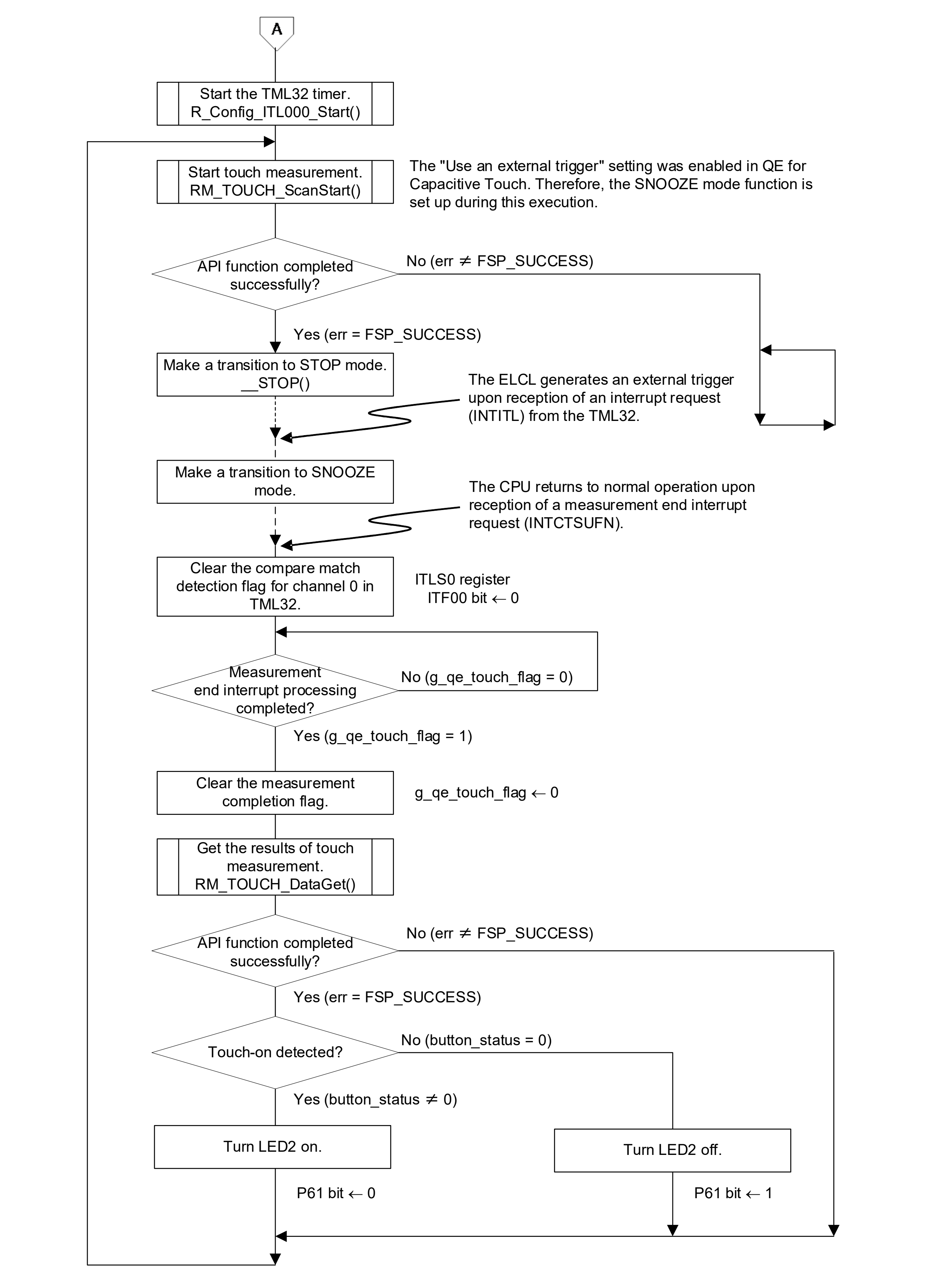



**Figure 5-10 Touch Measurement Control Processing (2/2)**

# 6. Current Consumption

## 6.1 Environment for Measuring Current Consumption

Figure 6-1 shows the environment used in measuring current consumption.

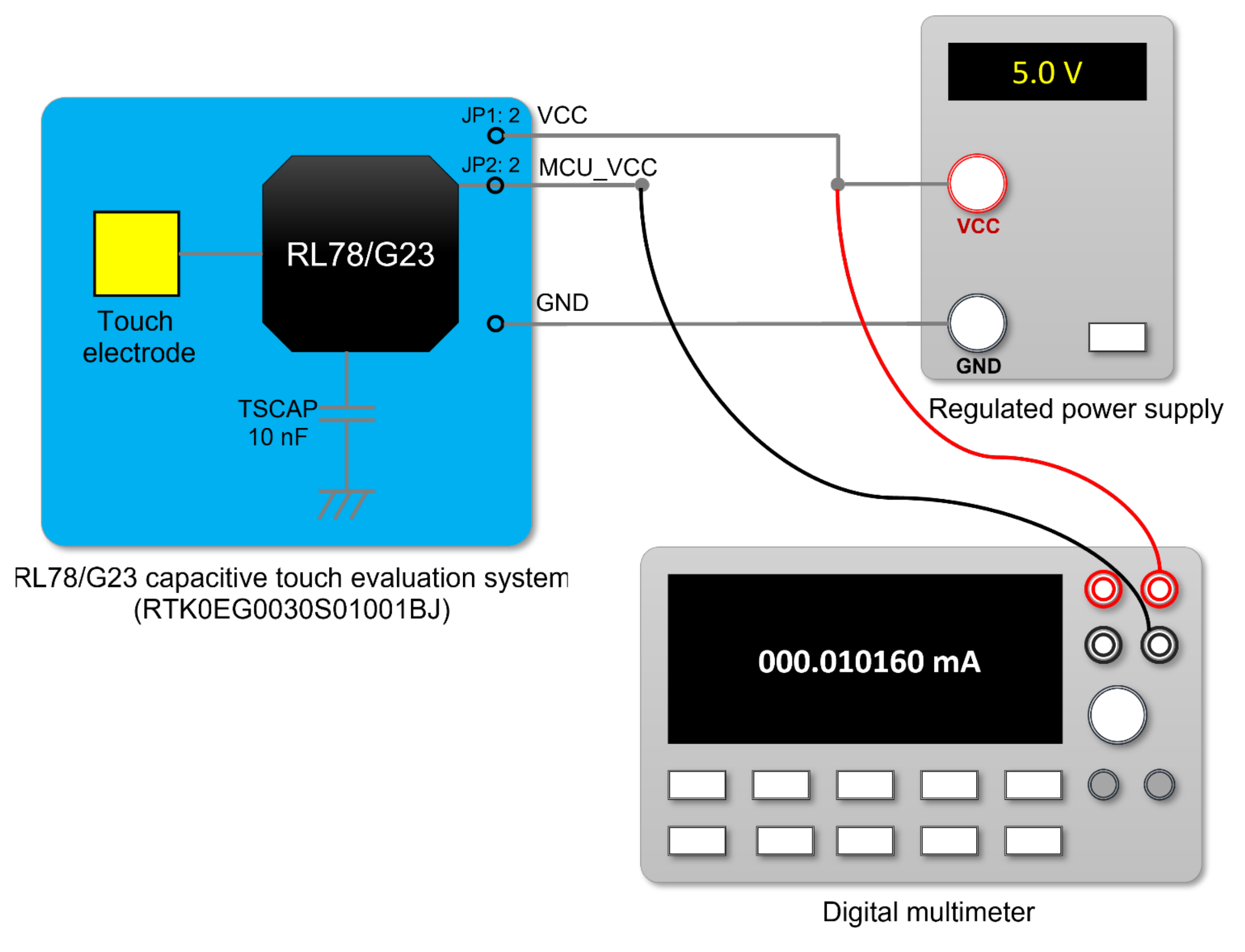



**Figure 6-1 Environment for Measuring Current Consumption**

## 6.2 Measurement Equipment and Software

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

**Table 6-1 Measurement Equipment and Software**

| Type | Name | Description |
|---|---|---|
| Digital multimeter | Keithley DMM7510 | Measures current consumption. |
| Regulated power supply | KENWOOD PWR18-2TP | Supplies power to the CPU board of the RL78/G23 capacitive touch evaluation system (RTK0EG0030S01001BJ). |
| Software | KickStart (V1.9.8.21) | Gets results of measurement of current consumption from the Keithley DMM7510 and outputs the results to a log file. |

## 6.3 CPU Board Jumper Settings

Table 6-2 shows the jumper settings on the CPU board of the RL78/G23 capacitive touch evaluation system when current consumption is to be measured.

**Table 6-2 CPU Board Jumper Settings**

| Jumper | Circuit Group | Setting | Description |
|---|---|---|---|
| JP1 | VCC power | Open (See Figure 6-1.) | Power supply from JP1: 2 |
| JP2 | MCU_VCC power | Open | For measuring current consumption |
| JP3 | Power-supply circuit | Pins 1-2 closed | — |
| JP4 | Power-supply circuit | Pins 1-2 closed | — |
| SW7 | Capacitive touch | Off (pins 1-2) | — |

## 6.4 Settings of Unused Pins

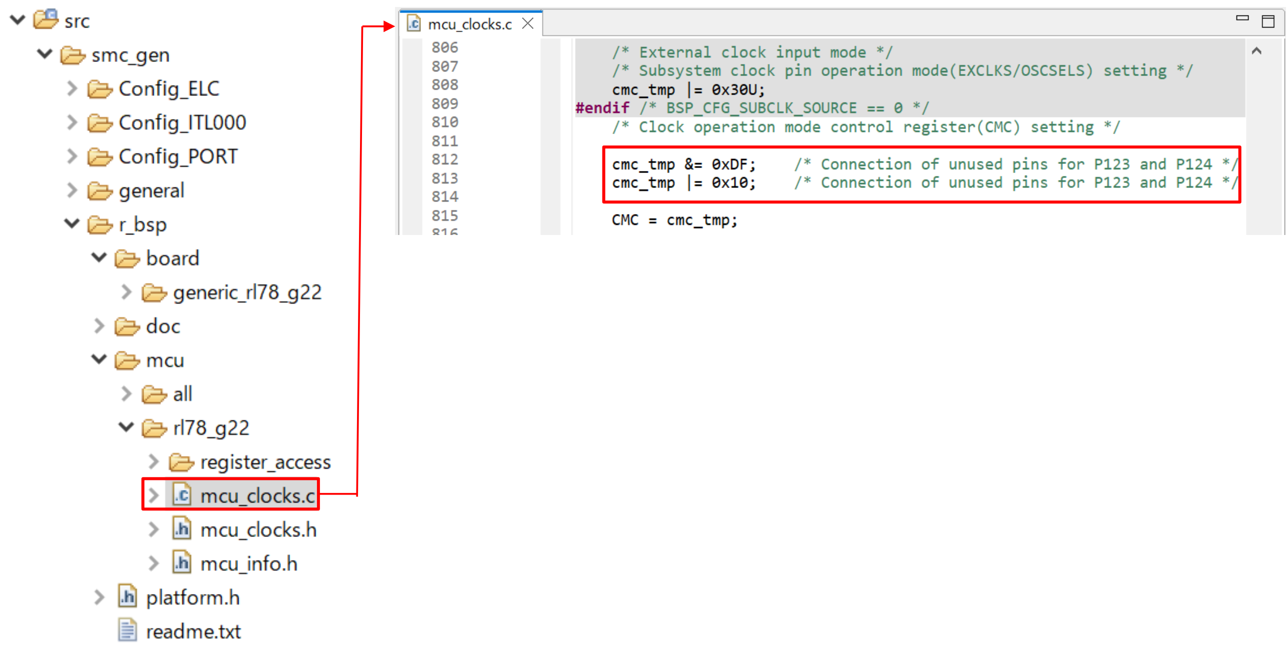
Table 6-3 shows the settings specified for the port component in the Smart Configurator to prevent the flow of current into unused pins during measurement of current consumption by the RL78/G23 capacitive touch evaluation system (RTK0EG0030S01001BJ).

**Table 6-3 Settings of Port Pins**

| Port Group | Port Name | Destination of Connection | Setting | Remarks |
|---|---|---|---|---|
| PORT0 | P04 | — | Low output | Open |
| | P07 | CN2 | | |
| PORT1 | P10 to P12 | CN3 | Low output | Open |
| | P13 to P15 | — | Low output | Open |
| | P16 | SW4 | Input | Connect to VDD via a resistor. |
| | P17 | — | Low output | Open |
| PORT2 | P20 | CN2 | Low output | Open |
| | P21 | — | | |
| PORT3 | P32 | CN1 | Low output | Open |
| | P33 | U3 | | |
| | P34 | U4 | | |
| | P35 | CN1 | | |
| | P36 to P37 | CN1 (LED_COL) | Low output | Set the pins such that none of the matrix LEDs on the electrode board of the evaluation system are turned on. |
| PORT4 | P41 to P42 | — | Low output | Open |
| | P43 to P46 | CN3 | | |
| | P47 | CN1 | | |
| PORT5 | P51 to P56 | — | Low output | Open |
| | P57 | SW3 | Input | Connect to VDD via a resistor. |
| PORT6 | P60 | LED1 | High output | Set the pin such that LED1 on the CPU board of the evaluation system is not turned on. |
| | P62 to P63 | — | Low output | Open |
| PORT8 | P80 to P87 | — | Low output | Open |
| PORT9 | P90 to P97 | — | Low output | Open |
| PORT10 | P100 to P101 | — | Low output | Open |
| | P102 | CN2 | | |
| | P103 | — | | |
| | P104 to P106 | CN1 (LED_ROW) | High output | Set the pins such that none of the matrix LEDs on the electrode board of the evaluation system are turned on. |
| PORT11 | P110 to P117 | — | Low output | Open |
| PORT12 | P120 | CN1 (LED_ROW) | High output | Set the pin such that none of the matrix LEDs on the electrode board of the evaluation system are turned on. |
| | P121 to P122 | — | Low output | Open |

| Port Group | Port Name | Destination of Connection | Setting | Remarks |
|---|---|---|---|---|
| PORT12 | P123 to P124Note | — | Input | Set the EXCLKS bit to 0 and the OSCSELS bit to 1 in the clock operation mode control register (CMC), set the XTSTOP bit in the clock operation status control register (CSC) to 1, and leave the pins open. |
| | P125 to P127 | — | Low output | Open |
| PORT13 | P130 | CN2 | Low output | Open |
| | P137 | CN3 | Input | Connect to VDD via a resistor. |
| PORT14 | P140 to P141 | CN1 (LED_COL) | Low output | Set the pins such that none of the matrix LEDs on the electrode board of the evaluation system are turned on. |
| | P142 | CN1 | Low output | Open |
| | P143 to P144 | CN1 | | |
| | P145 to P147 | — | | |
| PORT15 | P156 | — | Low output | Open |

Note The CSC register is set by the qe_touch_main function in qe_touch_sample.c. The CMC register is set by the mcu_clock_setup function in mcu_clocks.c as shown in Figure 6-2.




Caution If you change the version of the "r_bsp" component in the Smart Configurator, mcu_clocks.c will be overwritten and any code added by the user will be erased.
Therefore, whenever you change the version of the "r_bsp" component, you will need to add the code for the above settings.

**Figure 6-2 Editing the mcu_clocks.c File**

## 6.5 Settings of the Tool for Measuring Current Consumption

Figure 6-3 shows the settings of the tool for measuring current consumption (Keithley KickStart).

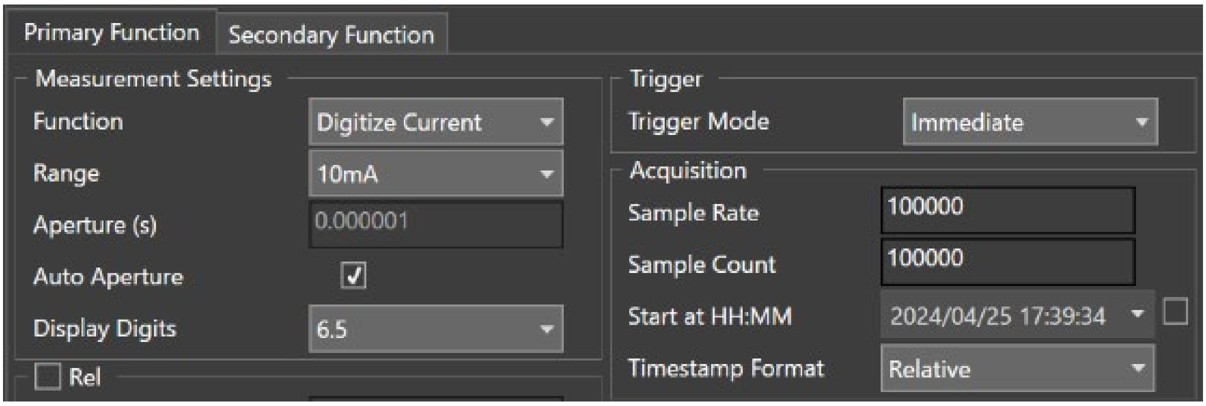



**Figure 6-3 Settings of the Tool for Measuring Current Consumption (Keithley KickStart)**

## 6.6 Results of Measuring Current Consumption

This section shows a comparison of the results of measuring current consumption in low-power operation and in normal operation.

For low-power operation, intermittent operations for touch measurement are handled by using the SNOOZE mode function.

For normal operation, touch measurement always proceeds in Normal operation mode without using the SNOOZE mode function or intermittent operations.

### 6.6.1 Waveforms of Current Consumption in Intermittent Operations

Figure 6-4 shows the waveforms of current consumption when touch measurement proceeds every 100 ms in low-power operation.

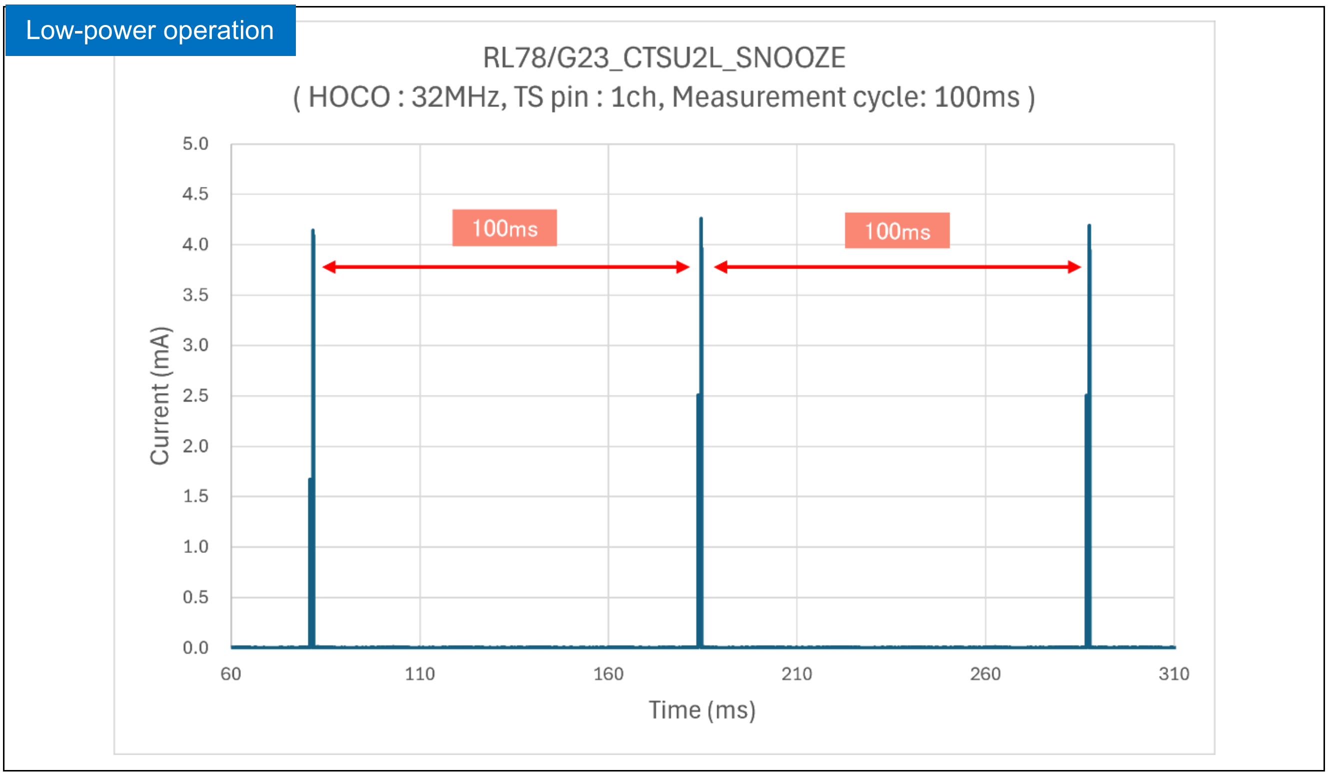



**Figure 6-4 Waveforms of Current Consumption When Touch Measurement Proceeds Every 100 ms (Low-Power Operation)**

Figure 6-5 shows the waveforms of current consumption when touch measurement proceeds every 100 ms in normal operation.

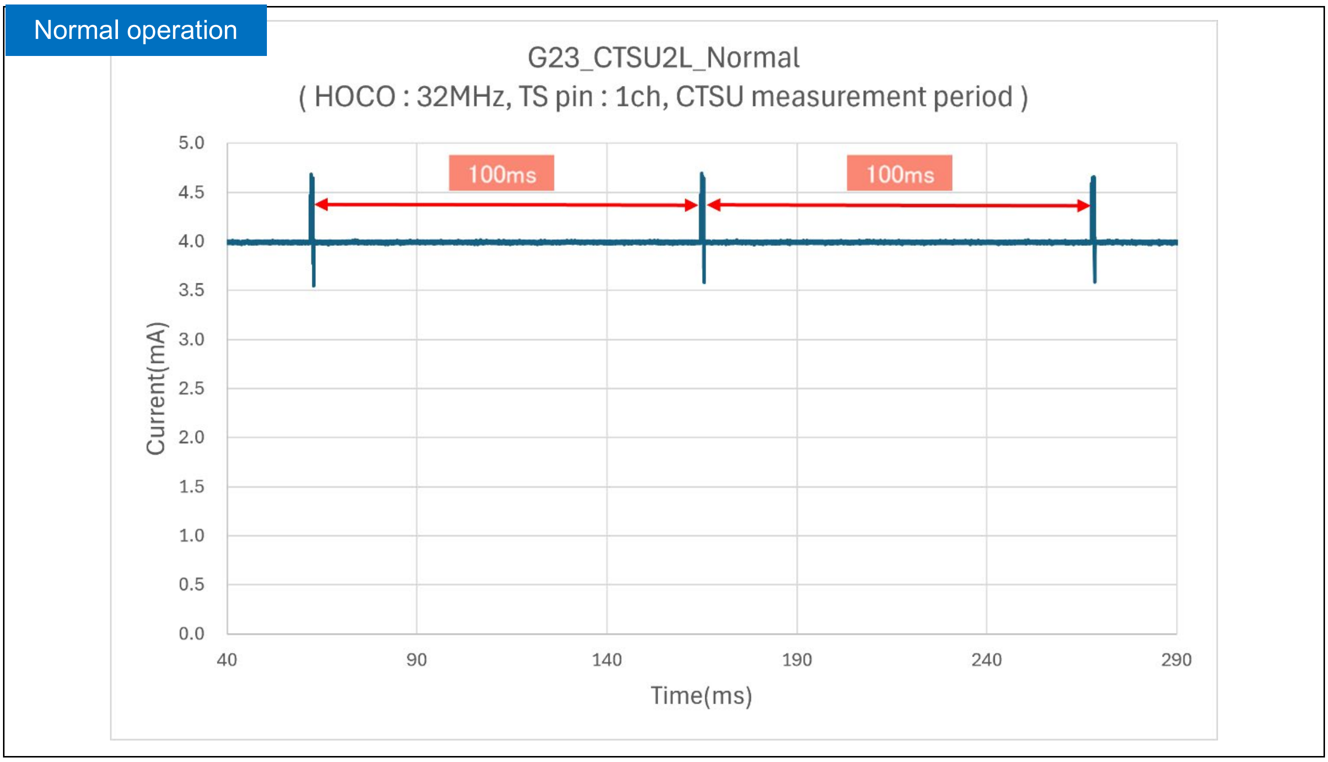



**Figure 6-5 Waveforms of Current Consumption When Touch Measurement Proceeds Every 100 ms (Normal Operation)**

### 6.6.2 Waveforms of Current Consumption during Touch Measurement Processing

Figure 6-6 shows the waveforms of current consumption during touch measurement processing in low-power operation. Table 6-4 lists the states of the CPU and CTSU in each measurement period. This figure shows the waveforms of current consumption for the series of operations in which the CPU makes the transition from STOP mode to SNOOZE mode to perform touch measurement processing and then to Normal operation mode to perform touch measurement end processing and touch-on/off judgment processing.

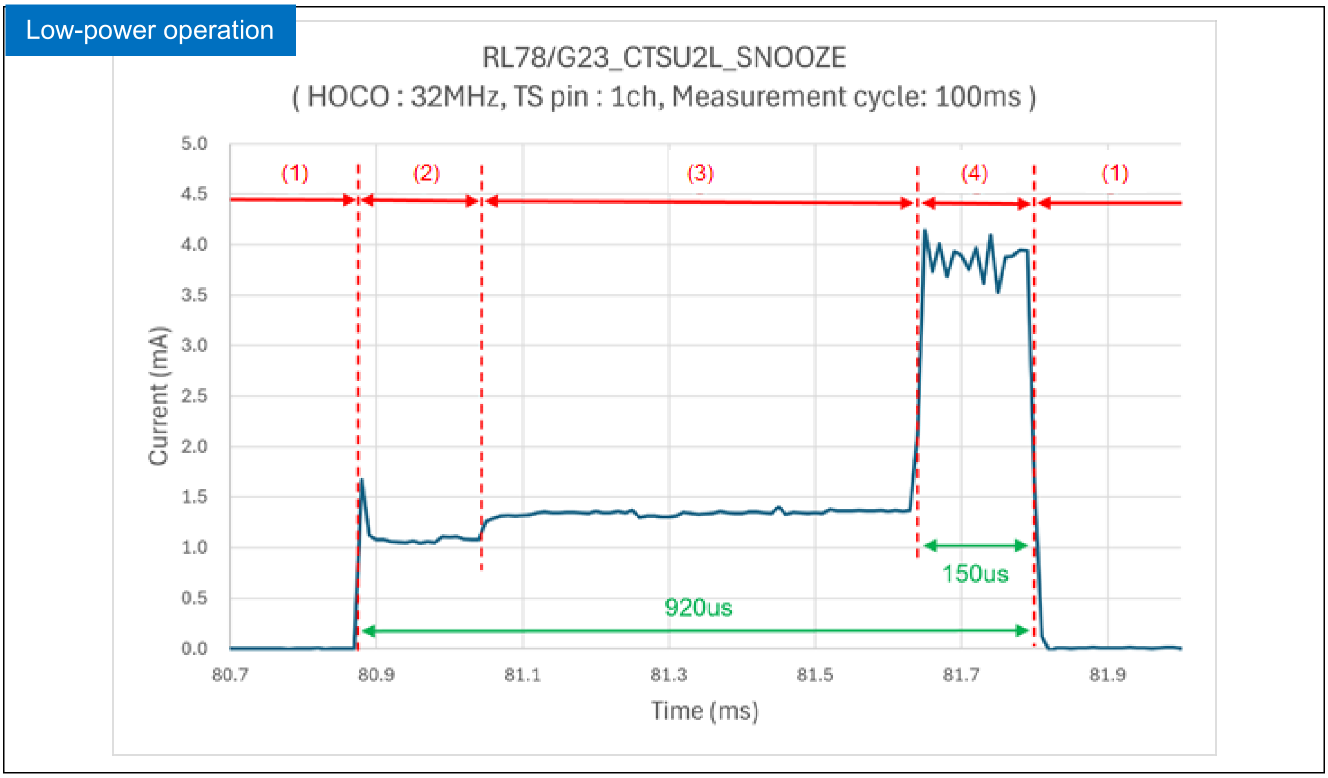



**Figure 6-6 Waveforms of Current Consumption during Touch Measurement Processing (Low-Power Operation)**

**Table 6-4 States of the CPU and CTSU in Each Measurement Period (Low-Power Operation)**

| Number in the Figure | State of the CPU | State of the CTSU Hardware | Remark |
|---|---|---|---|
| (1) | STOP mode | Suspended state | — |
| (2) | SNOOZE mode | Measurement in SNOOZE mode is in progress. | The operation is shifted to (3) and measurement begins 80 base-clock cycles after the clock supply (fCLK) to the CTSU is started. |
| (3) | SNOOZE mode | Measurement in SNOOZE mode is in progress. | The CTSU is operating. |
| (4) | Normal operation mode | Suspended state | The CTSU is stopped. The CPU handles calculating the results of measurement. |

Figure 6-7 shows the waveforms of current consumption during touch measurement processing in normal operation. Table 6-5 lists the states of the CPU and CTSU in each measurement period. This figure shows the waveforms of current consumption for the series of operations "touch measurement processing, touch measurement end processing, and touch-on/off judgment processing" when the CPU is always in Normal operation mode.

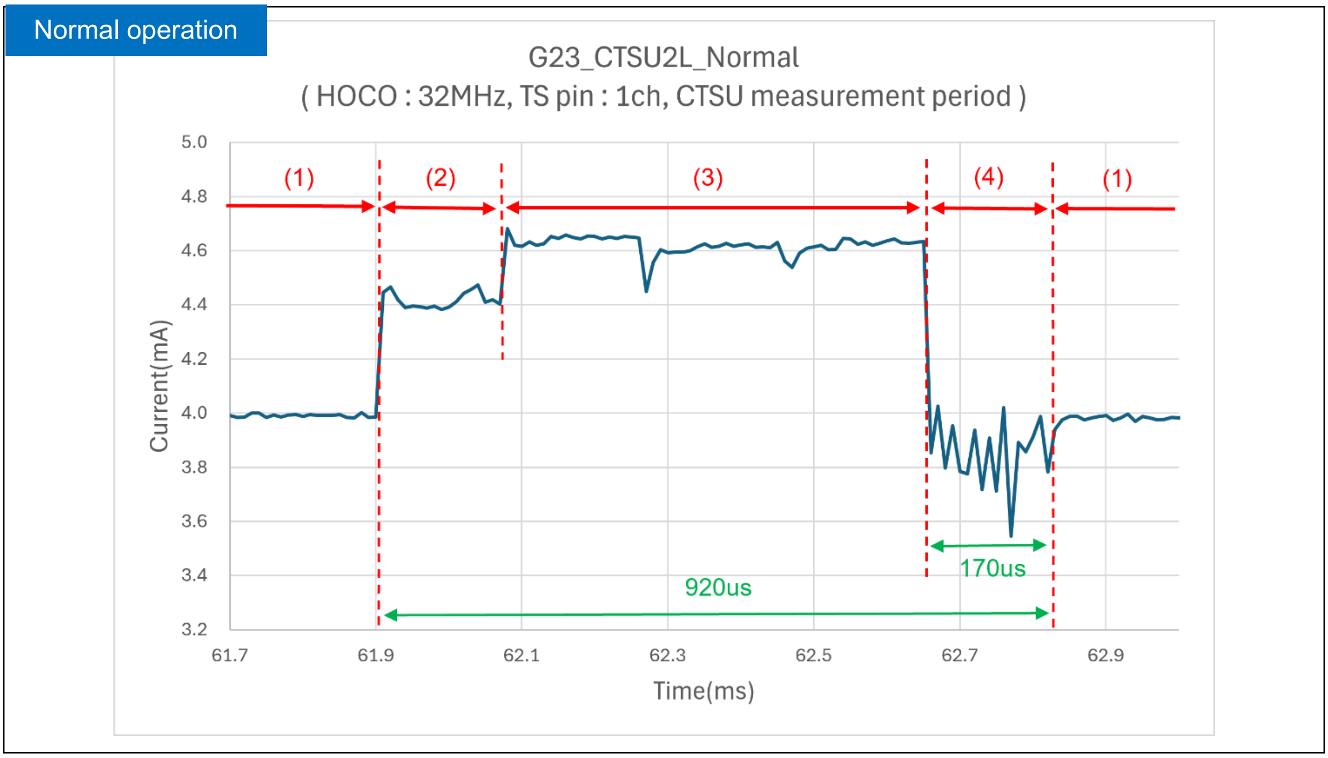



**Figure 6-7 Waveforms of Current Consumption during Touch Measurement Processing (Normal Operation)**

**Table 6-5 States of the CPU and CTSU in Each Measurement Period (Normal Operation)**

| Number in the Figure | State of the CPU | State of the CTSU Hardware | Remark |
|---|---|---|---|
| (1) | Normal operation mode | Waiting for the start of measurement | — |
| (2) | | Measurement in normal operation is in progress. | The operation is shifted to (3) and measurement begins 80 base-clock cycles after the clock supply (fCLK) to the CTSU is started. |
| (3) | | Measurement in normal operation is in progress. | The CTSU is operating. |
| (4) | | Waiting for the start of measurement | The CTSU is stopped.<br>The CPU handles calculating the results of measurement. |

### 6.6.3 Average Current Consumption per 100 ms

The following shows the results of average current consumption as obtained by the Keithley KickStart software. Table 6-6 shows the average current consumption in each CPU mode and the average current consumption per 100 ms in low-power operation.

**Table 6-6 Results of Average Current Consumption (Low-Power Operation)**

| CPU Mode | Time (ms) | Average Current Consumption (μA) |
|---|---|---|
| STOP mode | 99.08 | 0.811 |
| SNOOZE mode<br>(touch measurement processing) | 0.77 | 1302 |
| Normal operation mode<br>(touch measurement end processing + touch-on/off judgment processing) | 0.15 | 3868 |
| STOP mode + SNOOZE mode + Normal operation mode | 100 | 16.78 |

**Average current consumption per 100 ms ≈ 16.78 μA**

Table 6-7 shows the average current consumption per 100 ms in normal operation.

**Table 6-7 Results of Average Current Consumption (Normal Operation)**

| CPU Mode | Time (ms) | Average Current Consumption (μA) |
|---|---|---|
| Normal operation mode<br>(touch measurement processing + touch measurement end processing + touch-on/off judgment processing) | 100 | 3994 |

**Average current consumption per 100 ms ≈ 3.994 mA**

Table 6-8 indicates the difference in the average current consumption per 100 ms between the operation conditions that apply in Table 6-6 and Table 6-7.

**Table 6-8 Difference in Average Current Consumption between Operation Conditions**

| Operation Condition | SNOOZE Mode Function and Intermittent Operation | Transitions between CPU Modes | Average Current Consumption per 100 ms |
|---|---|---|---|
| Low-power operation | Used | STOP mode<br>→ SNOOZE mode<br>→ Normal operation mode | 16.78 μA |
| Normal operation | Not used | Always in Normal operation mode | 3.994 mA |

**Reduced to approximately 1/238**

# 7. Sample Code

The sample code is available on the Renesas Electronics Web site.

# 8. Notes on the Sample Code (Timing of Determining the Result of Touch Measurement)

## 8.1 Effect of Debouncing count of touch-on and touch-off filter

The timing when the touch measurement result is determined depends on the settings of the debouncing count of touch-on and touch-off filter. The result of touch measurement is determined when the measured capacitance value is detected as either the touch-on or touch-off value the required consecutive number (each noise filter setting value + 1) of times.

In the sample code, the number of cycles is set to "3" for both the debouncing count of touch-on and touch-off filter. Therefore, the RM_TOUCH_DataGet function is called in every touch measurement cycle to get the measured value and the result of touch measurement is determined as touch-on or touch-off when the state of either touch-on or touch-off is detected four consecutive times.

For details, see the application note RL78 Family TOUCH Module Software Integration System (R11AN0485).

## 8.2 Effect of Moving Average Setting

The timing of determining the result of touch measurement depends on the setting for moving average calculation.

In the sample code, the number of samples for the moving average is set to "4". This number of samples can be specified by using the variable "num_moving_average" in the qe_touch_config.c file.

Figure 8-1 shows examples of moving average processing.

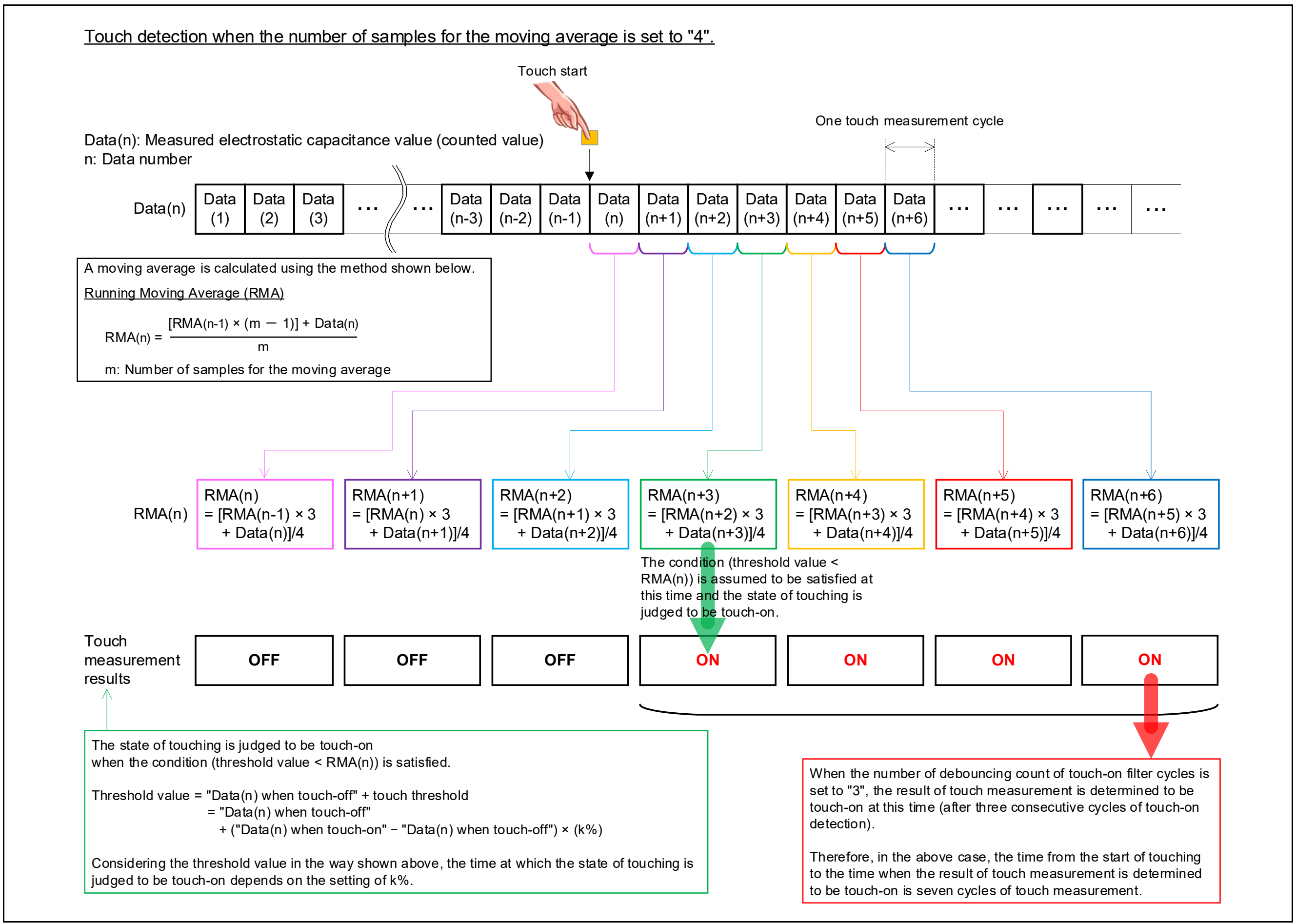



**Figure 8-1 Examples of Moving Average Processing**

# 9. Documents for Reference

- User's Manuals
  — RL78/G23 User's Manual: Hardware [R01UH0896]
  — RL78 Family User's Manual: Software [R01US0015]
  The latest versions of the documents are available on the Renesas Electronics Web site.

- Technical Updates and Technical News
  The latest information is available on the Renesas Electronics Web site.

  User's Manual: Development Environment
  — RL78/G23 Capacitive Touch Evaluation System User's Manual (RTK0EG0030S0100BJ) [R12UZ0095]

- Application Notes
  — 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]
  — RL78/G23 Capacitive Touch Low Power Guide (SMS function) [R01AN6670]
  — RL78 Family Capacitive Touch Low Power Application Development using SMS [R01AN7261]
  — Capacitive Sensor MCU QE for Capacitive Touch Advanced Mode Parameter Guide [R30AN0428]
  The latest versions of the documents are available on the Renesas Electronics Web site.

## Web site and Support

Renesas Electronics Web site
http://www.renesas.com/

Capacitive Sensing Unit related pages
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 | Oct.30.24 | - | First edition issued |
| 2.00 | Jun.09.25 | 1 | "Introduction" was modified. |
| | | 1 | "Using the Low Power Consumption Application Notes" was added. |
| | | 4 | Chapter 1, Overview, was added. |
| | | 4 | Section 1.1 was added and the contents of chapter 5, Current Consumption, in Rev. 1.00 were updated and moved to that section. |
| | | 5 | Chapter 2, Touch Measurement Mechanism, was added. |
| | | 5 | Figure 2-1, Touch Measurement with Low Power Consumption (SNOOZE Mode Function), was added. |
| | | 7 | Section 2.2 was added and the contents of section 1.2, Used Peripherals, in Rev. 1.00 were moved to that section. |
| | | 7-8 | Chapter 3, Environment and Conditions for Verifying Operation, was added. Table 3-1 and Table 3-2 were updated. |
| | | 9 | Chapter 4, Hardware Descriptions, was added. |
| | | 9 | Section 4.1, Example of Hardware Configuration, was added. |
| | | 10 | Section 5.1, Outline of Sample Code Operations, was added and the contents of chapter 3, Software Description, in Rev. 1.00 were moved to that section. |
| | | 10 | A note was added to section 5.1. |
| | | 10 | Section 5.2, Smart Configurator Settings, was added. |
| | | 10 | Section 5.2.1, Components Used, was added. |
| | | 10 | Figure 5-1 was updated. |
| | | 11 | Section 5.2.2, Component Settings, was added. |
| | | 11 | A note on the settings of the ELCL flexible circuit was added to section 5.2.2. |
| | | 11-12 | Figure 5-2 to Figure 5-4 were added. |
| | | 13 | Section 5.3, Capacitive Touch Settings, was added.<br>Figure 5-5 and Figure 5-6 were added. |
| | | 14 | Section 5.4, CTSU Startup Setting, was added. |
| | | 14 | Section 5.5, Settings of the Option Bytes, was added. |
| | | 15 | Section 5.6, Folder and File Configuration, was added. |
| | | 16 | Section 5.7, Details of the Sample Code, was added and the contents of section 3.2, List of Variables, and section 3.3, List of Functions, in Rev. 1.00 were moved to that section. |
| | | 17-19 | Figure 5-8 to Figure 5-10 were updated. |
| | | 20-21 | Table 6-1 and Table 6-2 were updated. |
| | | 22-23 | Section 6.4, Settings of Unused Pins, was added. |
| | | 23 | A note on register settings was added to section 6.4. |
| | | 24 | Figure 6-3 was updated. |
| | | 25 | The results of measurement of current consumption in normal operation were added to section 6.6 and the description in the section was updated. |
| | | 26 | Figure 6-5, Waveforms of Current Consumption When Touch Measurement Proceeds Every 100 ms (Normal Operation), was added. |

| | | | |
|---|---|---|---|
| | | 28 | Figure 6-7, Waveforms of Current Consumption during Touch Measurement Processing (Normal Operation), was added. |
| | | 28 | Table 6-5, States of the CPU and CTSU in Each Measurement Period (Normal Operation), was added. |
| | | 29 | Section 6.6.3, Average Current Consumption per 100 ms, was added to give the results for average current consumption (Table 6-6 and Table 6-7). |
| | | 29 | Table 6-8, Difference in Average Current Consumption between Operation Conditions, was added. |
| | | 30 | Section 8.1 was updated. |
| | | 30 | The number of samples for moving averages in the sample code was corrected. |
| | | 30 | The timing of determining the result of touch measurement was corrected in section 8.2 and Figure 8-1. |

# 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/.