RL78/G23

Self-capacitance Waterproof Touch Button Demo Sample Software

Introduction

The CTSU2, the enhanced version of the Renesas Capacitive Touch Sensing Unit (CTSU), supports an active shield as one of its anti-noise countermeasures. This document describes the self-capacitance waterproof touch button demo software, which implements the active shield function.

Target Device

RL78/G23 (R7F100GSN2DFB)

Contents

1. Overview

This document describes how to design in waterproof buttons based on self-capacitance detection, using the hardware and software used in the self-capacitance waterproof touch button demo set (referred to "demo set" herein).

2. Self-capacitance Touch Button: water resistance and noise immunity

Figure 2-1 shows parasitic capacitance when a water droplet adheres to the GND shield button. Capacitance detection touch buttons tend to generate a false detection when exposed to water. The self-capacitive button features a GND shield for enhanced noise immunity, yet if a drop of water forms over the touch electrode and the GND shield, it creates a capacitive component which increases the capacitance and causes a false detection.
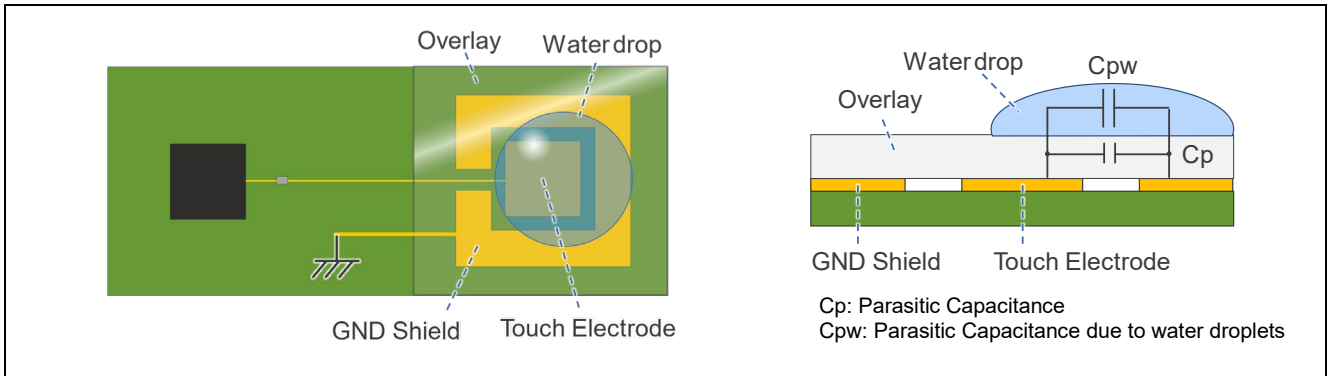
Figure 2-1 Parasitic Capacitance when Water Droplet Adheres to GND Shield Button

As the CTSU estimates the capacitance of a touch electrode from the charge/discharge current, when there is a potential difference in the capacitance component, a charge/discharge current is generated and the button may detect touch (be actuated) even if not actually touched. The CTSU2 resolves this issue by supporting an active shield that drives the shield pattern at the same potential as the electrode, enhancing both the noise immunity and water resistance of the self-capacitance method button. Figure 2-2 shows an image of parasitic capacitance cancellation by the CTSU2's active shield.
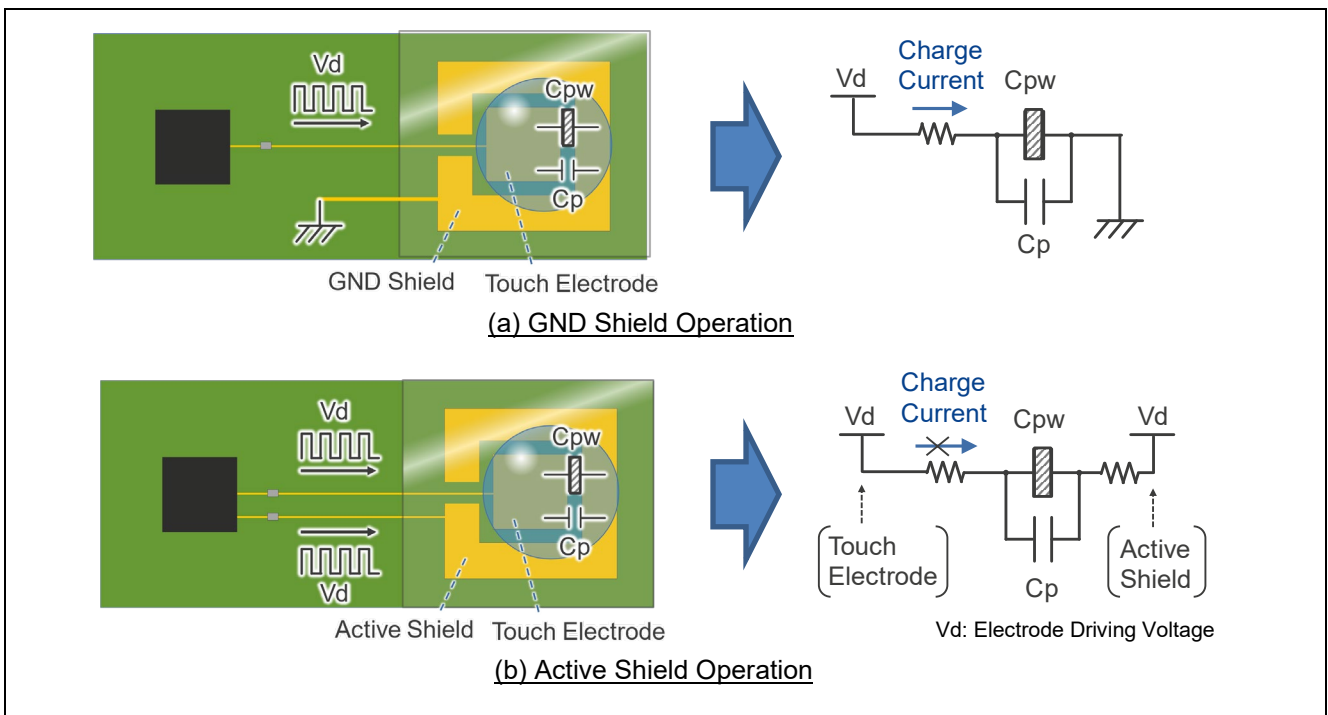
Figure 2-2 Image of Parasitic Capacitance Cancellation on CTSU2 Active Shield Button

Figure 2-3 shows the operations of the CTSU's non-measured pins. Because the CTSU's non-measured pins are fixed to low, a false detection may be generated even with no GND pattern surrounding the area if multiple buttons are exposed by water. In the CTSU2, on the other hand, the non-measured and measured pins are driven by the same potential, reducing the possibility of a false detection caused by adjacent button electrodes.
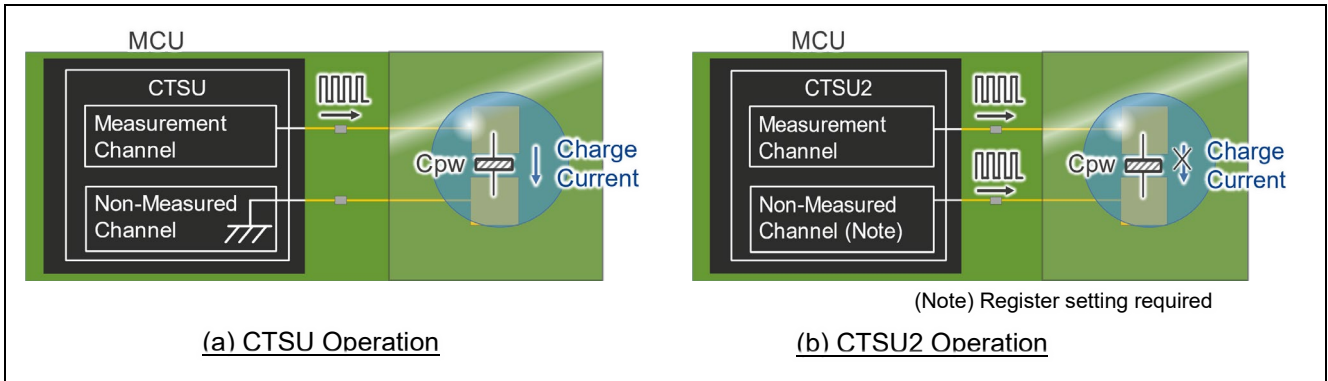

(Note) Register setting required

(a) CTSU Operation   (b) CTSU2 Operation

Figure 2-3 CTSU Non-Measured Pin Operations

3. Actual Device Operations

Figure 3-1 shows an external view of the electrode board used in this demo set. The demo board has a GND shield button that can be used with the conventional CTSU and an active shield button that can be used with the CTSU2, allowing a comparative evaluation of waterproof characteristics. In addition, software can be used to turn on the LED that corresponds to each button.
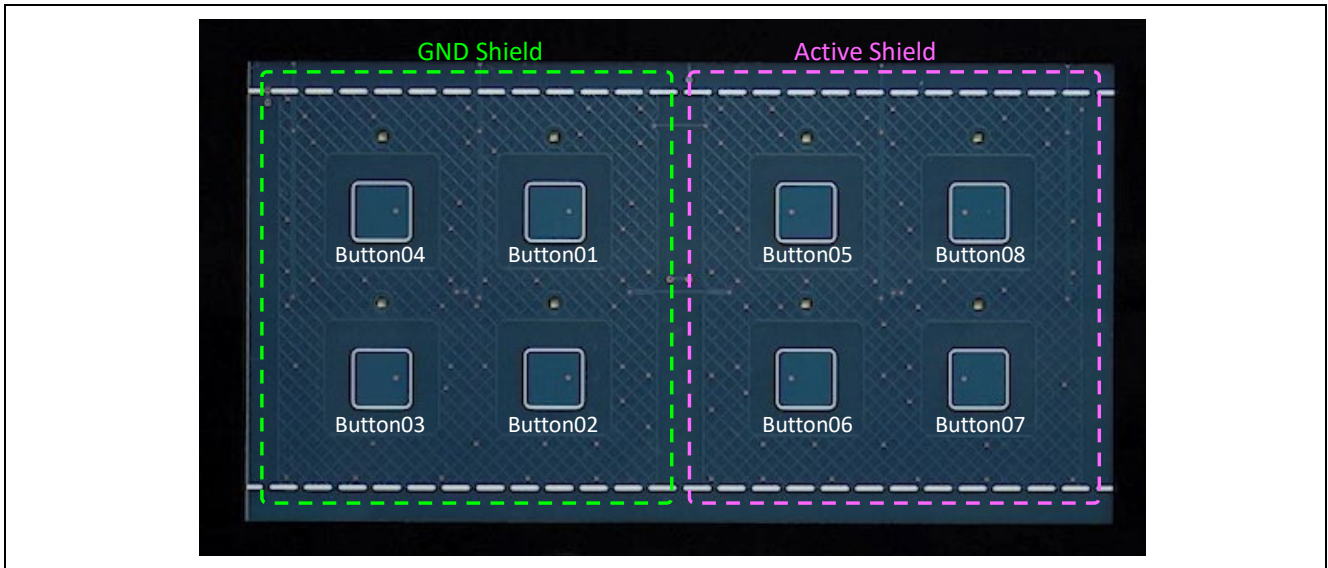
Figure 3-1   Self-Capacitance Touch Electrode Board for Waterproof Evaluation

Figure 3-2 shows the Demoset Block Diagram.
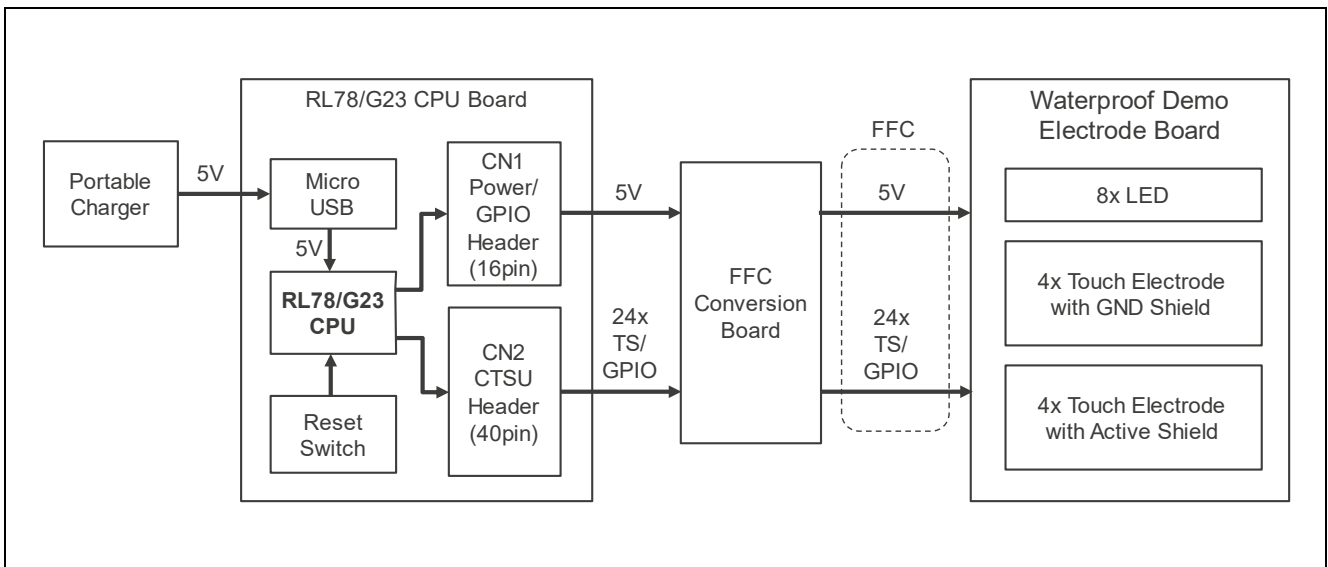

Figure 3-2   Demoset Block Diagram

Figure 3-3 shows the count value waveform when a water droplet adheres to a GND shield button. As the droplet covers both the GDN shield and the buttons, the count value exceeds the threshold and a touch is detected.

①   No-touch state, initial state
②   No-touch state, buttons completely covered with water. Touch detected when all buttons exceed threshold.
③   Button01 touch detected
④   Button02 touch detected
⑤   Button03 touch detected
⑥   Button04 touch detected
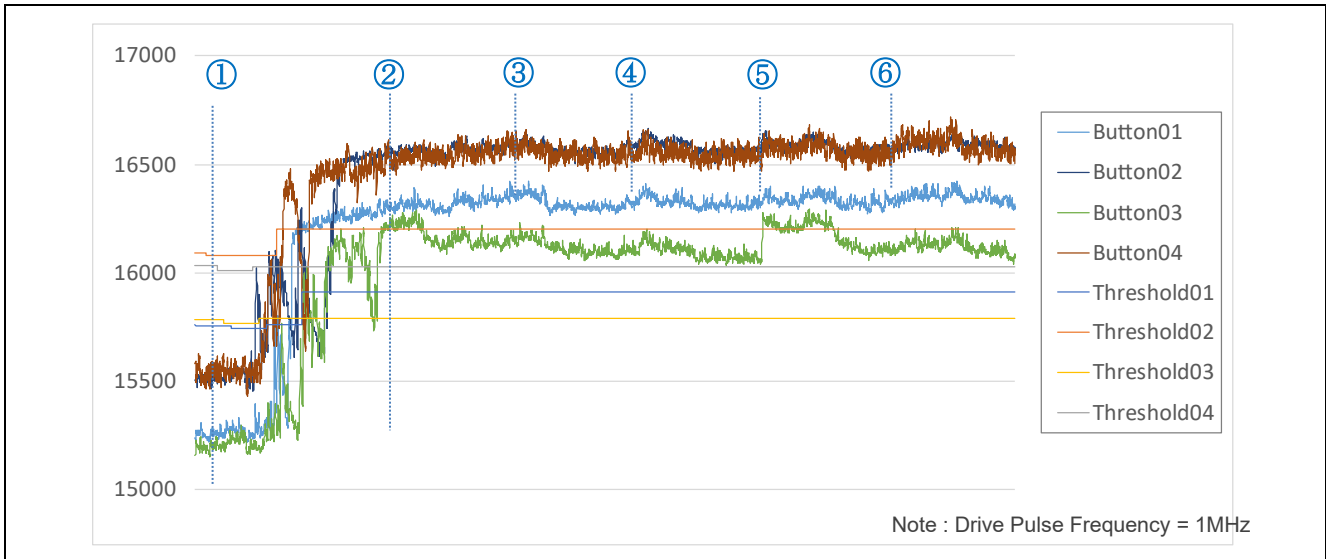
Figure 3-3   Count Value Waveform When Water Droplet Adheres to GND Shield Buttons
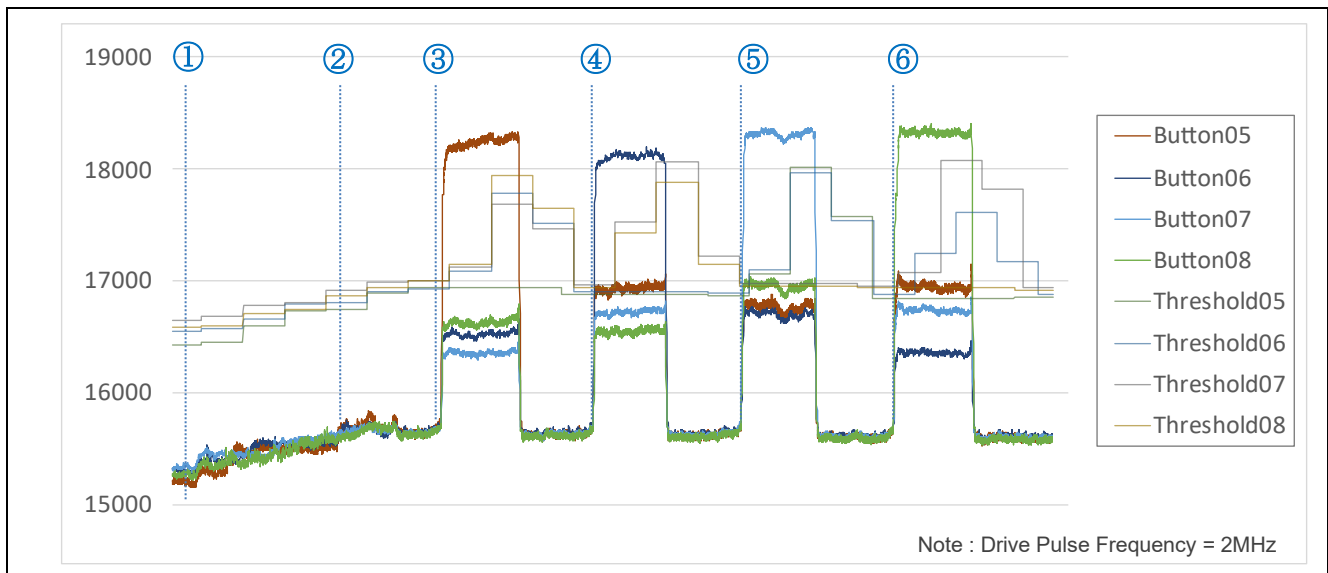
Figure 3-4 shows operations when a water droplet adheres to a GND shield button. Due to a false detection, all buttons detect touch (are actuated) and the corresponding LEDs light up.
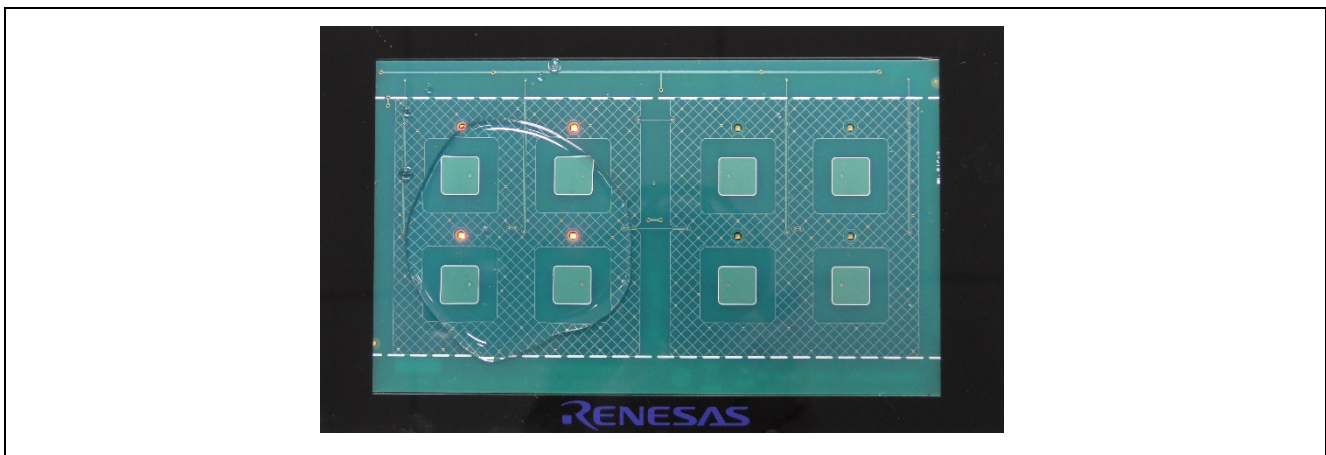
Figure 3-4   Operations when Water Droplet is Adhered to GND Shield Buttons

Figure 3-5 shows the changes in the count value and threshold when a droplet adheres to the active shield buttons. Using the active shield prevents false detections even when the buttons are covered by a droplet.

① No-touch state, initial state
② No-touch state, buttons completely covered with water.
   Touch is not detected (button not actuated) despite the increase in sensor count. The threshold follows the value due to drift correction.
③ Button05 touch detected
   When the count value for Button05 exceeds the threshold (Threshold05), touch is detected (button is actuated). Although the water droplet causes the count value of other buttons to increase, the value does not exceed the corresponding threshold, and therefore touch is not detected (button is not actuated).
④ Button06 touch detected
   Operations are the same as with Button05, but due to differing sensitivities, other buttons may exceed their thresholds and falsely detect touch (button is actuated). In such cases, false detection must be prevented by adjusting touch parameters, such as threshold or drift correction interval, or controlling button detection by software.
⑤ Button07 touch detected
⑥ Button08 touch detected

Figure 3-5   Active Shield Button Count Value Waveform

Figure 3-6 shows operations when a droplet is adhered to the active shield button. The active shield prevents the LED from turning ON as the button does not falsely detect touch (not actuated) if covered by a droplet.
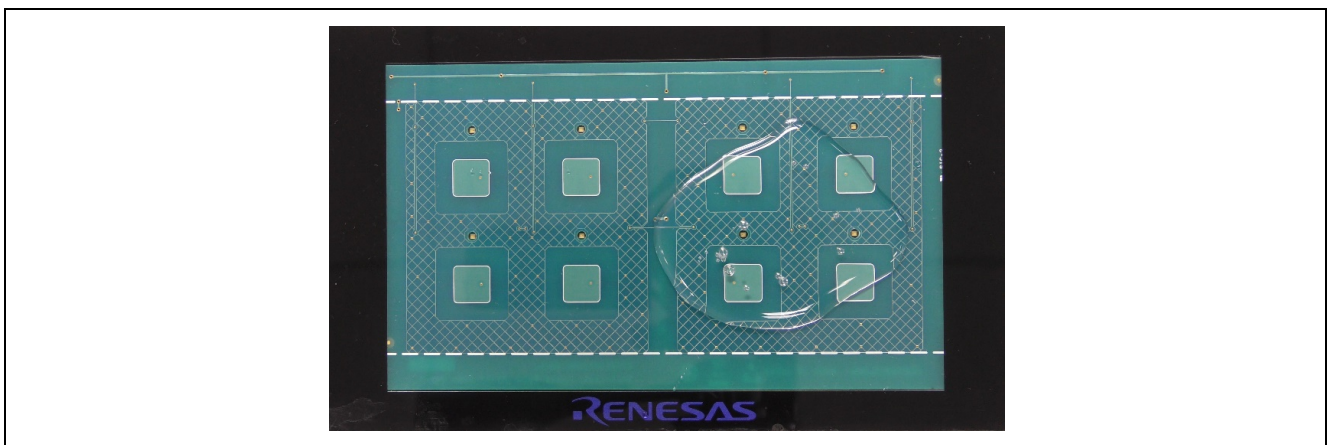
Figure 3-6   Operations when Water Droplet Adheres to Active Shield Buttons

4. Software Specifications

4.1 Operational Overview

Figure 4-1 shows the flowchart for the sample program.

In addition to the project output by the Smart Configurator and the project tuned by the QE for Capacitive Touch, this sample program includes two new functions: maximum count value difference button detection processing for the self-capacitance waterproof button and LED control processing. The periodic timer generates timing to improve touch response and control LEDs. The touch module's config01 and config02 indicate control structures. config01 is for the GND shield button group and config02 is for the active shield button group. For setting details, see 4.8 Touch Interface Settings.
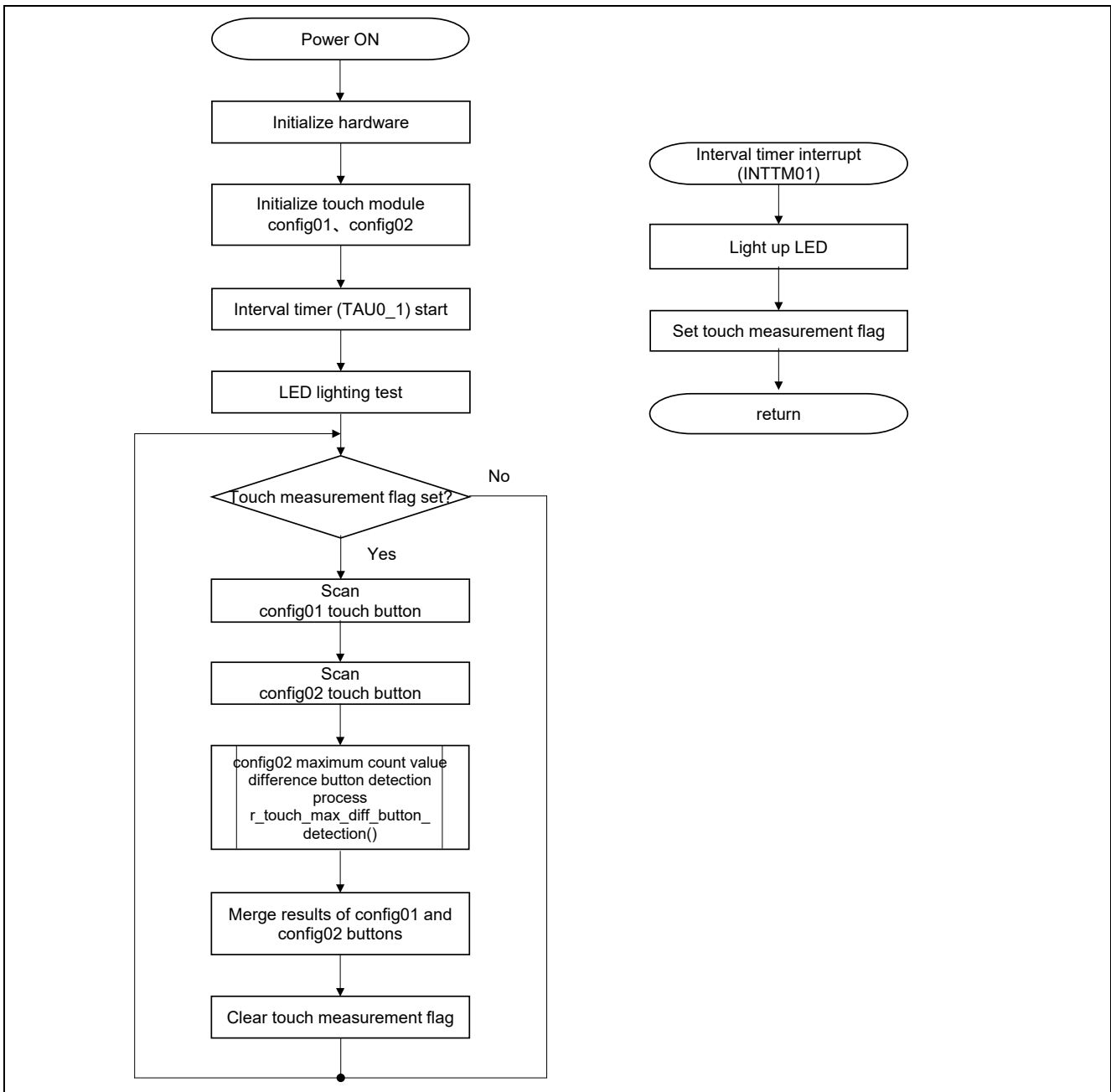
Figure 4-1   Sample Program Flowchart

Figure 4-2 shows the flowchart for the maximum count value difference button detection process. This process detects the channel with the maximum count value when the button defined in the control structure detects a touch, and judges it as the button being pressed. To improve response to continuous press, drift correction for all channels is disabled during button detection.
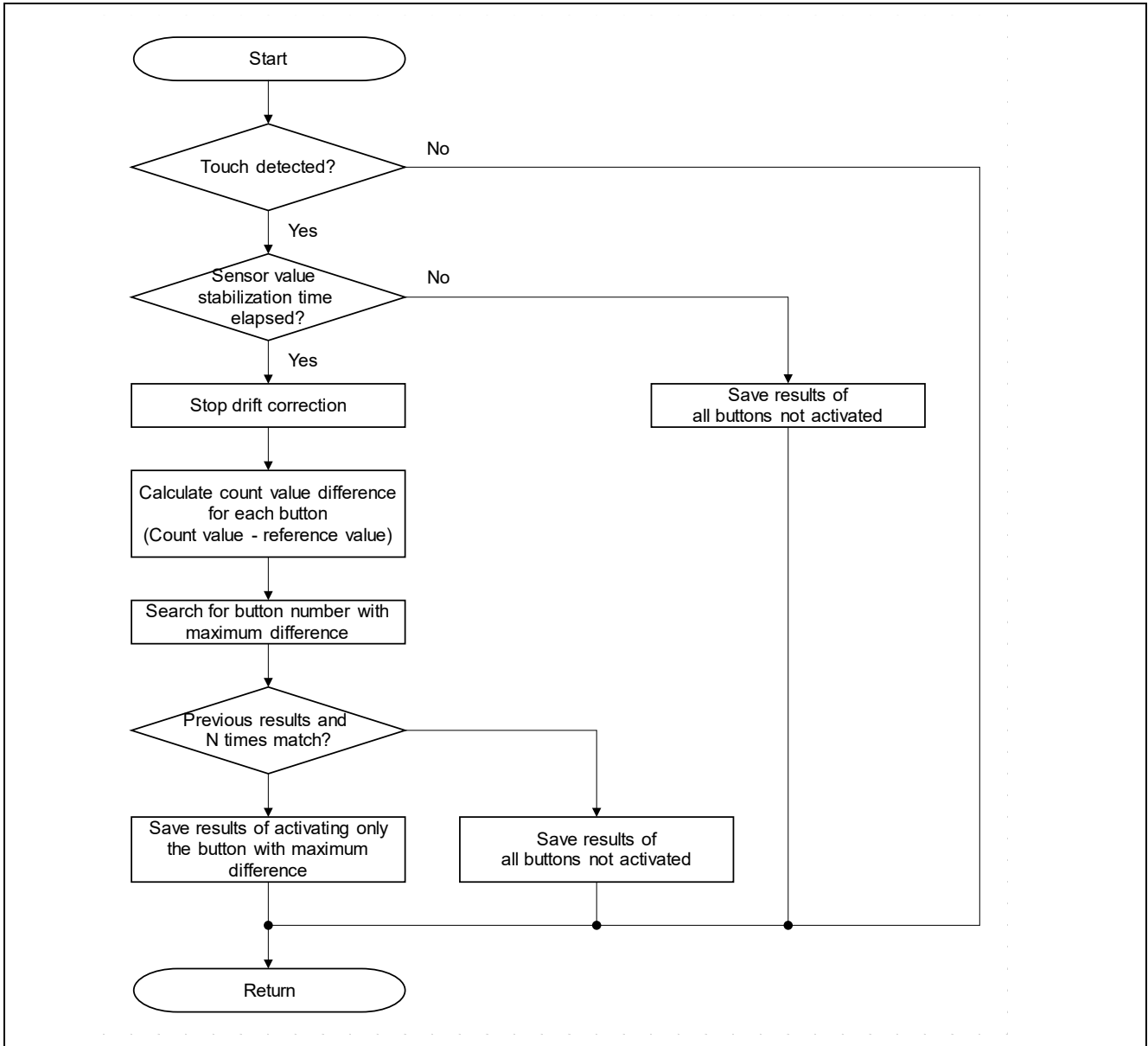

Figure 4-2   Flowchart for Maximum Count Value Difference Button Detection Process

4.2 Operating Environment

Table 4-1 details the conditions for the hardware development environment; Table 4-2 shows the same for the software development environment.

Table 4-1 Hardware Conditions

Item	Specification
CPU board	RL78/G23 Cap Touch CPU Board (RTK0EG0029C01001BJ) (RL78/G23 Capacitive Touch Evaluation System (RTK0EG0030S01001BJ) included)
MCU used	RL78/G23 (R7F100GSN2DFB)
Operating frequency	32MHz
Electrode board	• Self-Capacitance Touch Electrode Board for Waterproof Evaluation — Self-capacitance button (GND shield): 4 buttons — Self-capacitance button (active shield): 4 buttons — LEDs: 8 • FFC conversion board for Capacitive Touch Evaluation System • Overlay: acrylic 2mm
Power supply	5.0V

Table 4-2 Software Development Environment

Item	Specification
Integrated Development Environment (IDE)	Renesas e² studio Version: 2021-10 (21.10.0)
Compiler	Renesas CC-RL V1.10.00
RL78 Smart Configurator e² studio plugin	V1.1.0
Capacitive touch sensor development support tool	QE for Capacitive Touch V2.0.0
Emulator	Renesas E2 Emulator Lite

4.3　File Structure

Table 4-3 displays the project file structure. The types of files generated by the FSP Smart Configurator and QE for Capacitive Touch have been omitted for brevity. Similarly, the software descriptions for LED control have been omitted from this document.

Table 4-3　Project File Structure

Folder/File Name			Description of Change
Project			-
	qe_gen (file structure omitted)		-
	src		-
		rl78g23_ctsu_waterproofdemo.c	Main function was added to demo program.
		r_touch_waterproof_demo.c	Source code for controlling the demo.
		r_touch_waterproof_demo_led.c	Source code for LED control was added. Description not included in this document.
		r_touch_waterproof_demo_led.h	Header file for LED control was added. Description not included in this document.
	QE_Touch (file structure omitted)		-
	ra_cfg (file structure omitted)		-
	script (file structure omitted)		-

　-: no change

4.4　List of Functions

Table 4-4 lists the functions used in r_touch_waterproof_demo.c.

Table 4-4　r_touch_waterproof_demo.c Functions

Function Name	Description
r_touch_waterproof_demo_main()	Demo program main function
r_touch_max_diff_button()	Maximum count value difference button detection function
r_timer0_callback	Interval timer interrupt callback function

4.4.1　r_touch_waterproof_demo_main

This is the main routine for the sample program.

Format

　　void r_touch_waterproof_demo_main(void)

Argument

　　None

Return value

　　None

Description

　　This is the main routine for the sample program

4.4.1 r_touch_max_diff_button

Only the button that has the largest count value difference among the buttons defined in the control structure is actuated (touch detected).

Format

　　void r_touch_max_diff_button_detection(ctsu_ctrl_t * const p_ctrl, uint64_t * p_data)

Argument

　　ctsu_ctrl_t * const p_ctrl

　　　　Pointer to control structure (normally generated by QE for Capacitive Touch)

　　uint64_t * p_data

　　　　Pointer to button storing the button detection results

Return Value

　　None

Description

　　Only the button that has the largest count value difference among the buttons defined in the control structure is actuated (touch detected).

　　Specify the pointers for target button's control structure and button detection results in the argument.

　　Call this function after executing the touch module's RM_TOUCH_ScanStart() and RM_TOUCH_DagtaGet() functions.

4.4.2　　　r_timer0_callback

Interval timer interrupt callback function

Format

　　void r_timer0_callback(void)

Argument

　　None

Return value

　　None

Description

　　This function handles the interval timer interrupt. For this sample software, the function controls the LEDs and the touch detection processing execution flag.

4.5   List of Constants

Table 4-5 lists the constants used in r_touch_waterproof_demo.c.

Table 4-5   r_touch_waterproof_demo.c Constants

Constant Name	Initial Value	Description
TOUCH_SCAN_10ms_COUNT	2	Number of counts per 10ms interval
TOUCH_SCAN_INTERVAL	TOUCH_SCAN_10ms_COUNT	Sets the touch detection interval. The interval for this sample software is set to the set value x 5ms.
TOUCH_BUTTON_MVA_COUNT	4	Sets the moving average sample count for the touch middleware using the count value at which the sensor count value stabilizes. Normally, this is the same value as the control structure moving average count (num_moving_average of touch_cfg_t).
TOUCH_MAX_DIFF_BUTTON_DEBOUNCE_COUNT	4	Specifies the number of chatter removal samples for the maximum count value difference button detection processing. This constant is set independently of the number of consecutive match measurements in the control structure (on_freq and off_freq of touch_cfg_t).

4.6   List of Variations

Table 4-6 lists the variations used in r_touch_waterproof_demo.c.

Table 4-6   r_touch_waterproof_demo.c Variations

Variation Name	Type	Initial value	Description
button_status	uint64_t	0	Stores the touch button detection results.
g_touch_scan_wait	uint8_t	0	Counts the number of waits until the touch detection interval.
g_touch_scan_flag	uint8_t	0	Touch detection interval flag 0: Do not execute touch detection process 1: Execute touch detection process

4.7   MCU Resources Used in Application

Table 4-7 lists the MCU resources used in application.

Table 4-7   MCU Resources Used in Application

Module Name	Port Name	Use
PORT	P23 P26 P150 P151 P152 P153	LED control
CTSU	TS06 TS07 TS08 TS09	Touch detection pin (GND shield button group)
	TS01 TS12 TS13 TS14	Touch detection pin (Active shield button group)
	TS15	Active shield output pin
TAU0_1	─	5ms interval timer
UARTA1	RxDA1 TxDA1	QE for Capacitive Touch Monitoring View

4.8 Touch Interface Settings

Figure 4-3 shows the touch interface settings. Figure 4-4 shows the structure (method) settings. The touch interface structure (method) assigns the GND shield button group to config01 and the active shield button group to configu02.
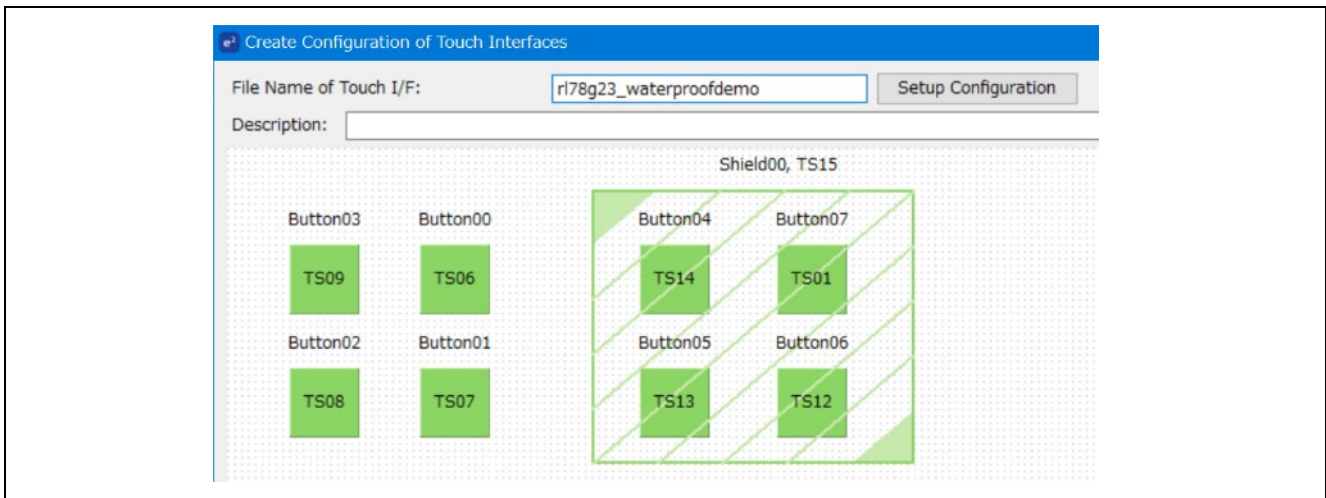
Figure 4-3   Touch Interface Settings
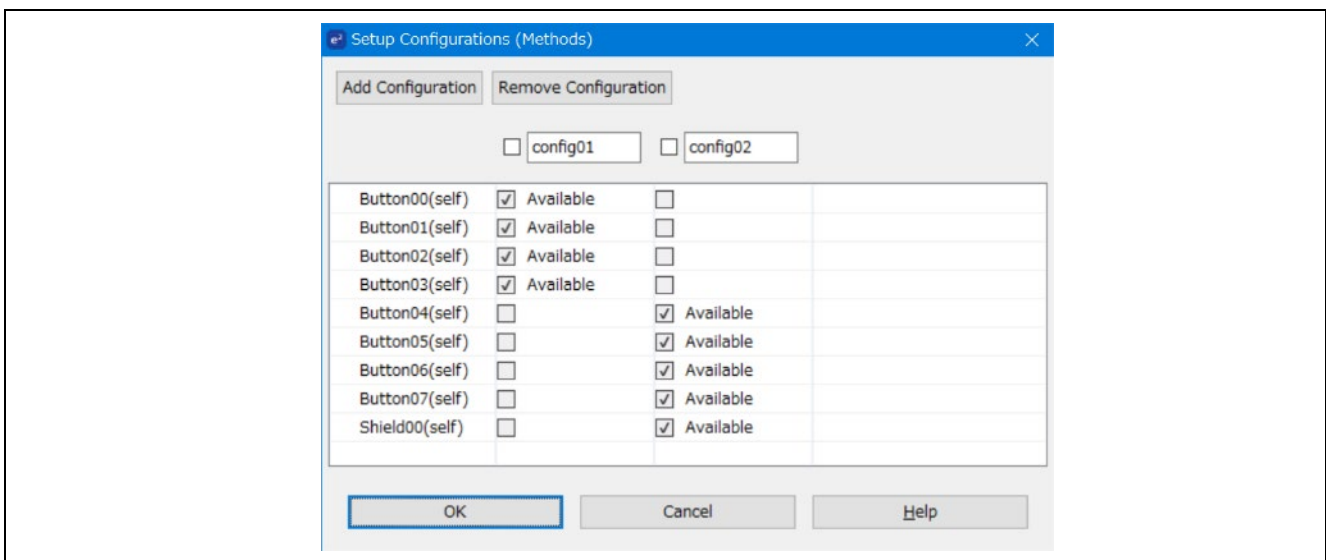
Figure 4-4   Structure (Method) Settings

4.9 Button Sensitivity Adjustment

Button sensitivity is adjusted with QE for Capacitive Touch. For details on how to make adjustments, please follow the QE for Capacitive Touch tutorial.

Revision History

| Rev. | Date | Description | |
		Page	Summary
1.0	Oct.25.21	-	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.

Notice

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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/.