Capacitive Touch Sensor Microcontrollers

Overview of Capacitive Touch Software

Outline

This application note describes the flow of touch judgement using the Capacitive Touch Sensing Unit (hereafter, referred to as CTSU) peripheral.

Based on the touch sample project created by QE for Capacitive Touch, it explains the flow from capacitance measurement to touch judgement with the CTSU driver and the TOUCH middleware functions.

Target Devices

RX Family, RA Family, RL78 Family MCUs and Renesas Synergy™ embedding the CTSU

In this application note, the (CTSU, CTSUa, CTSUb) are collectively referred to as CTSU1, and the (CTSU2,CTSU2L,CTSU2La,CTSU2SL,CTSU2SLa) are collectively referred to as CTSU2.

CTSU is used to collectively refer to CTSU1 and CTSU2.

Contents

1. Introduction

This application note describes the flow from reset to touch judgement using the CTSU peripheral.

The explanation is based on the touch sample project created by QE for Capacitive Touch.

It describes the flow from capacitance measurement to touch judgement using the CTSU driver and TOUCH middleware functions.

If you are developing a capacitive touch system for the first time, it is recommended that you read Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide beforehand.

For the CTSU modules and TOUCH middleware, refer to the following materials.

RL78 Family CTSU Module Software Integration System (R11AN0484)

RL78 Family TOUCH Module Software Integration System (R11AN0485)

RX Family QE CTSU Module Using Firmware Integration Technology (R01AN4469)

RX Family QE Touch Module Using Firmware Integration Technology (R01AN4470)

RA Flexible Software Package Documentation: CTSU (r_ctsu)

RA Flexible Software Package Documentation: Touch (rm_touch)

About developing tools and software components

The touch sample project (for RA2L1) used in this application note is created under the following environment.

- Integrated Development Environment e² studio 2025-01 (Build Id: R20241218-1723)
- Renesas QE for Capacitive Touch V4.1.0.v20250123-0520
- Renesas FSP Smart Configurator ARM V10.1.0.v20241205-1705
- Renesas FSP Smart Configurator Core V10.1.0.v20241218-1621

2. Processing Model

Unlike a typical switch with mechanical contacts, the Capacitive Touch Key captures minute changes in capacitance (a few pF or less) between the human body and the electrodes. The change is converted to ON/OFF similar to that of a mechanical switch.

To detect a touch with capacitive touch keys, the following are required:

- Settings for measurement of minute capacitance (a few pF or less)
- Conversion of the measurement result to touch ON/OFF
- Measure continuously

To achieve these, this software carries out the capacitive touch processing as shown below.

Figure 2-1 shows the entire flow of the capacitive touch processing.

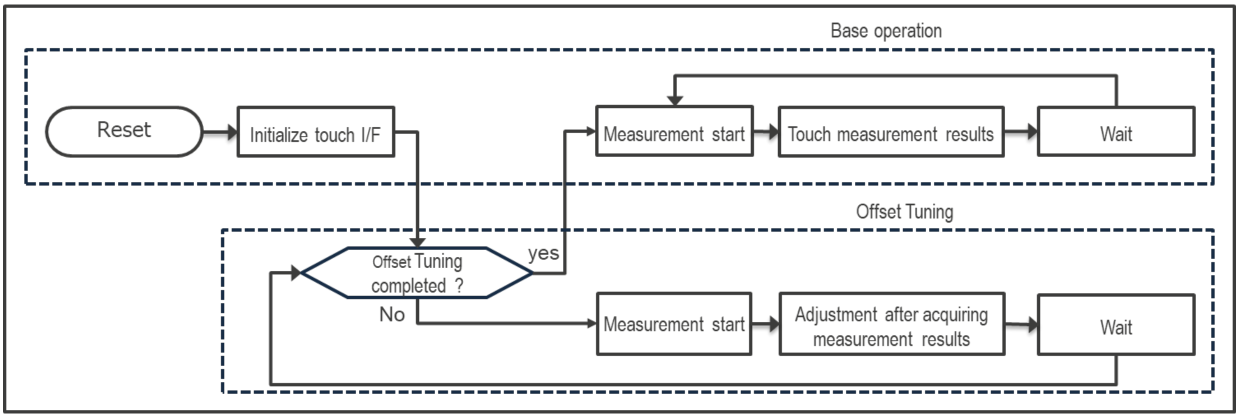

Figure 2-1 Entire flow of processing

The basic behavior is initialization for capacitance measurement, offset tuning, and after that, touch detection.

For the offset tuning after initialization, refer to “4. Offset Tuning.”

The basic behavior is explained based on the touch sample project and the configuration file created by QE for Capacitive Touch and the middleware and the drivers that are created by Smart Configurator.

2.1 Flow of Basic Behavior Model

Figure 2-2 shows the basic behavior model that judges a capacitive touch using the TOUCH modules.

The basic behavior is initialization for capacitance measurement and then touch detection, and its model is explained by using the function qe_touch_main in qe_touch_sample.c as below.

Touch detection process repeats the flow [Measurement start -> Acquisition of touch judgement result -> Wait].

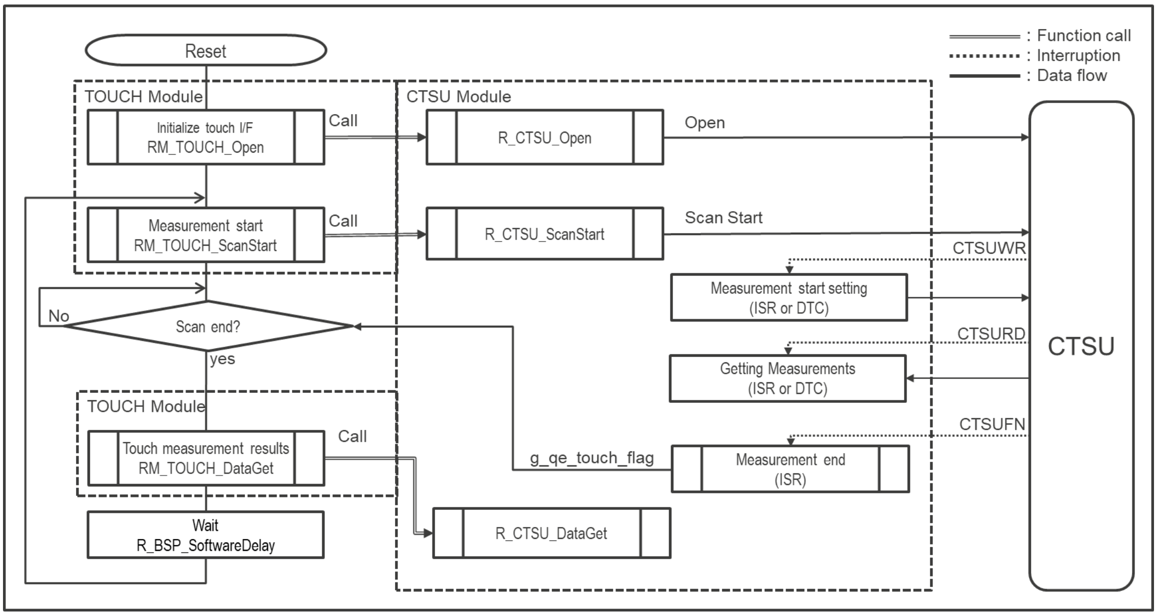

Figure 2-2 Basic behavior model

The following explains the main processing in Figure 2-2.

- **Initialization of touch interface**
  Configures initial settings for the touch processing with RM_TOUCH_Open.
  — Initializes the variables to be used for touch judgement and position detection.
  — Calls R_CTSU_Open.
  R_CTSU_Open carries out the following processes:
  — Initializes the variables and registers to be used for capacitance measurement.
  — Creates the Current Controlled Oscillator (hereafter, CCO) characteristic correction table.
  — Configures settings for start of auto tuning

- **Measurement start**
  Starts measuring capacitance according to the touch interface configuration.
  Executes R_CTSU_ScanStart from RM_TOUCH_ScanStart and makes the CTSU modules start measuring according to the touch interface configuration. When the measurement ends, notifies it by a flag.
  Additionally, Configures CTSU register settings and settings for processing measurement results (using interrupts or DTC).

- **Acquisition of touch judgement results**
  Acquires the touch judgement result by RM_TOUCH_DataGet.
  RM_TOUCH_DataGet makes a judgement based on the measurement value acquired by R_CTSU_DataGet.
  R_CTSU_DataGet performs offset tuning and applies correction to the measurement value. For details on correction, refer to “3.1.3 Correction Processing.”

- **SoftwareDelay**
  Waits to adjust the touch measurement cycle.

3. Flow of Touch Judgement Result Acquisition

This section describes the flow of acquiring touch judgement results. In this flow, measurement values and judgements are processed.

The function called first is RM_TOUCH_DataGet. RM_TOUCH_DataGet carries out the judgement process with the measurement value acquired by R_CTSU_DataGet.

Figure 3-1 shows the processing flow.

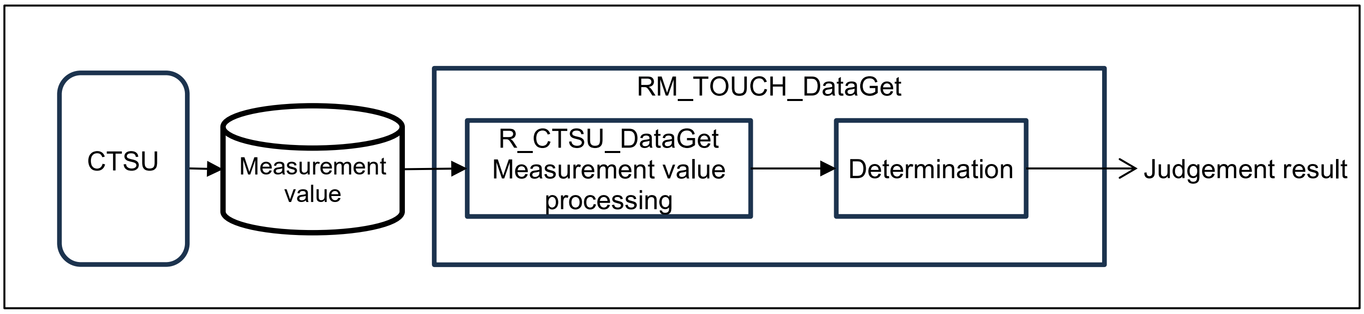

Figure 3-1 Touch judgement process

The measurement value processing is divided as below due to the difference of the processing flow.

- CTSU1
  Reads and corrects the result measured at one drive pulse frequency in the random pulse mode.

- CTSU2
  Reads and corrects the result measured at three drive pulse frequencies in the high-resolution pulse mode (multi-clock measurement).
  CTSU2 has two modes to process measurement values and judgements.
  — Value Majority Mode (VMM)
  — Judgement Majority Mode (JMM)

For the judgement process, refer to 3.2, which describes the following touch interfaces.

- Button touch judgement
- Slider position detection
- Wheel position detection

3.1 Measurement Value Processing

The measurement value is processed by R_CTSU_DataGet, and correction is applied to the acquired measurement value.

3.1.1 CTSU1

Figure 3-2 shows the measurement value processing in CTSU1.

CTSU1 measures in the random pulse mode, applies CCO correction and moving average to the measurement value, and gets one measurement value. For CCO correction, refer to 3.1.3.1. For moving average, refer to 3.1.3.2.

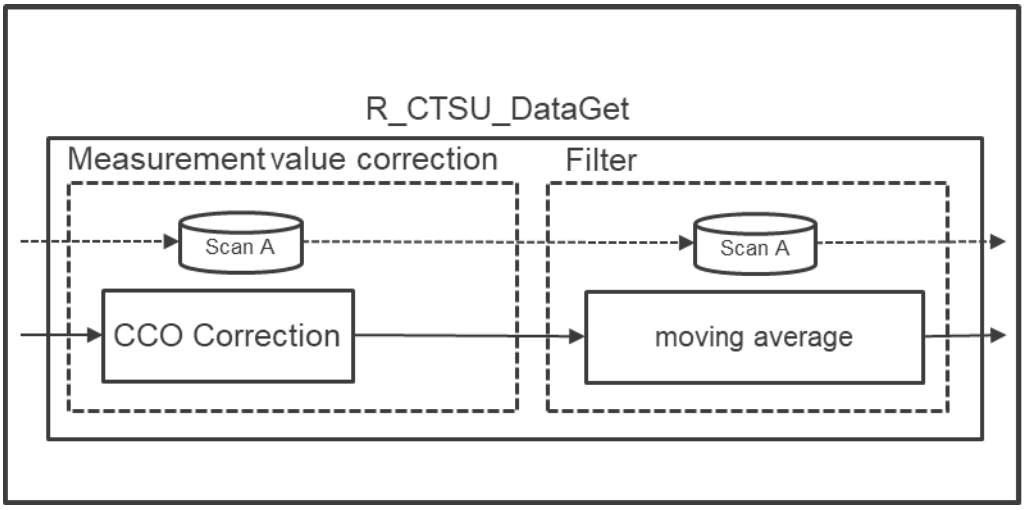

Figure 3-2 Measurement value processing in CTSU1

3.1.2 CTSU2

(1) Value Majority Mode (VMM)

Figure 3-3 shows the measurement value processing in VMM.

In VMM, measures at three drive pulse frequencies in high-resolution pulse mode.

CCO correction is applied to each count value. For CCO correction, refer to 3.1.3.1.

Multi-clock correction converts the measurement values based on one drive pulse frequency, and adds up two measurement values that are close each other to get one measurement value.

After that, performs moving average and gets one measurement value. For moving average, refer to 3.1.3.2.

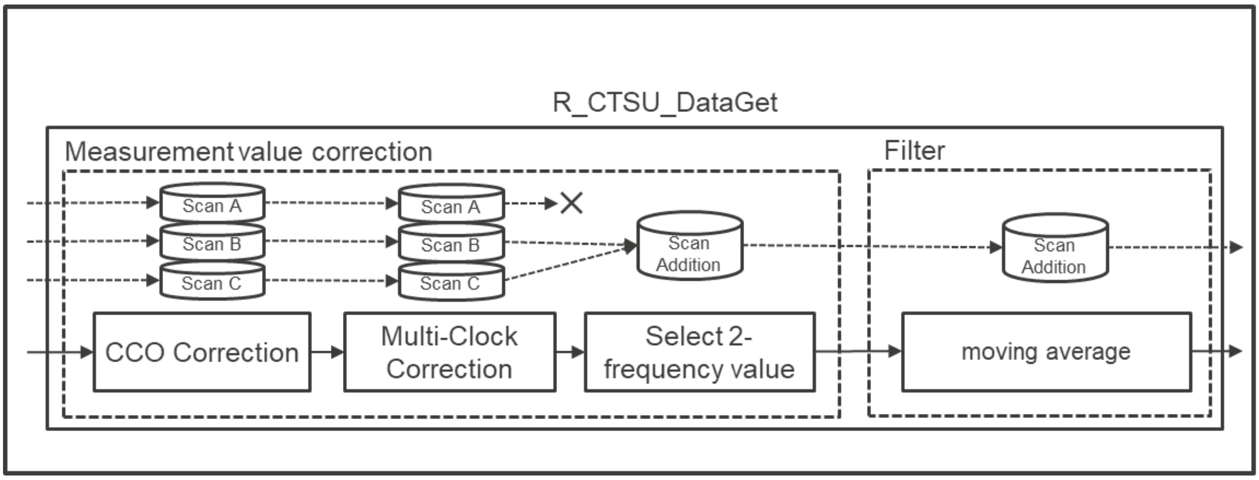

Figure 3-3 Measurement value processing in CTSU2 (VMM)

(2) Judgement Majority Mode (JMM)

Figure 3-4 shows the measurement value processing in JMM.

In JMM, measures at three drive pulse frequencies in the high-resolution pulse mode.

CCO correction is applied to each count value. For CCO correction, refer to the 3.1.3.1.

After that, performs moving average and gets three measurement values. For moving average, refer to 3.1.3.2.

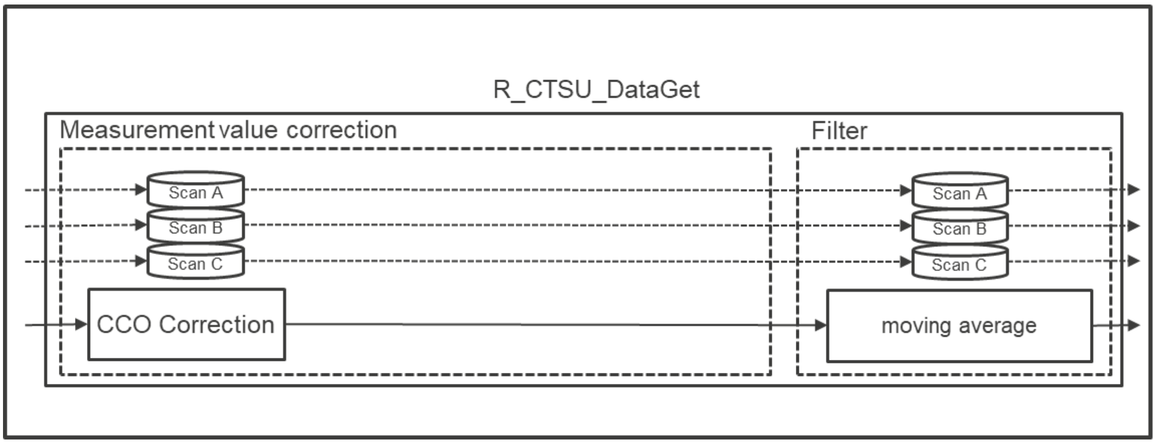

Figure 3-4 Measurement value processing in CTSU2 (JMM)

3.1.3 Correction Processing

3.1.3.1 CCO Correction

This is a process to ensure the accuracy of sensor measurement by improving the linearity of the output frequency for the CCO input current. The CCO characteristic correction table is used for the measurement results of the CCO that measures capacitance. This CCO characteristic correction table is created by R_CTSU_Open at the first initialization after the power is turned on.

3.1.3.2 Moving Average

To reduce noise, the acquired measurement value is smoothed by moving average processing.

3.2 Judgement Processing

In the judgement process, uses the measurement value output from R_CTSU_DataGet and makes a judgement for a button, slider, or wheel.

3.2.1 Button Touch Judgement

Judges whether the button is touched or not by detecting changes in capacitance. Figure 3-5 shows the judgement process.

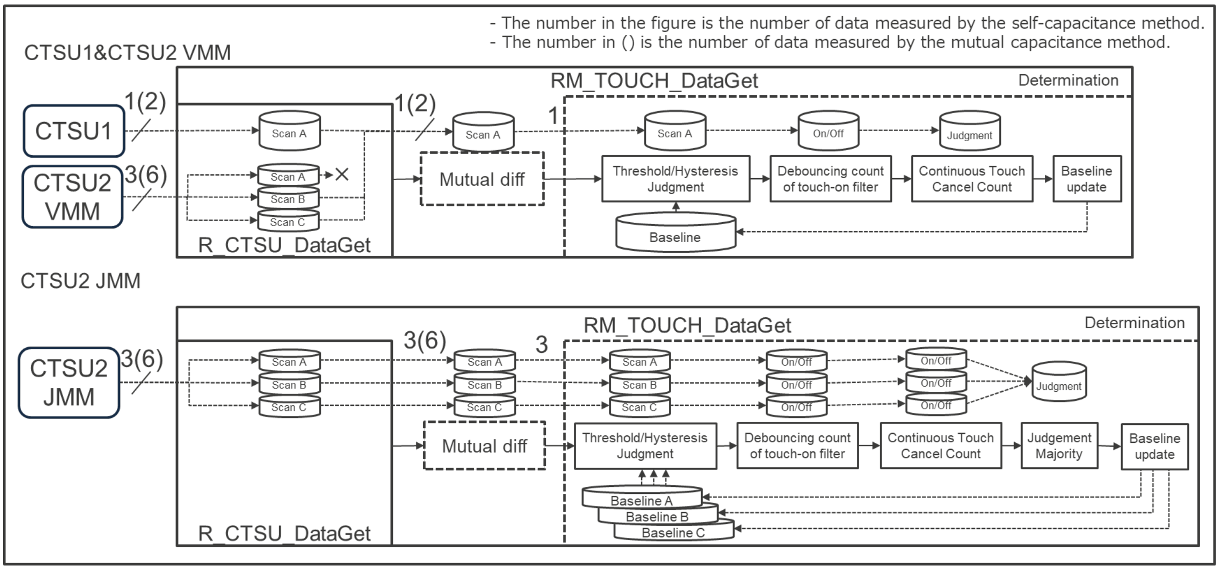

Figure 3-5 Judgement process

Unlike a button with mechanical contacts, a touch button has no ON/OFF state as hardware.

Therefore, the software judges a touch (touch ON/OFF) periodically.

The parameters related to touch judgement are “Baseline”, “Touch threshold”, and “Hysteresis”. A judgement is made based on the relation of four items: three parameters and “Measurement value”.

The self-capacitance button detects increase in the capacitance that occurs between the electrode and GND as shown in Figure 3-6.

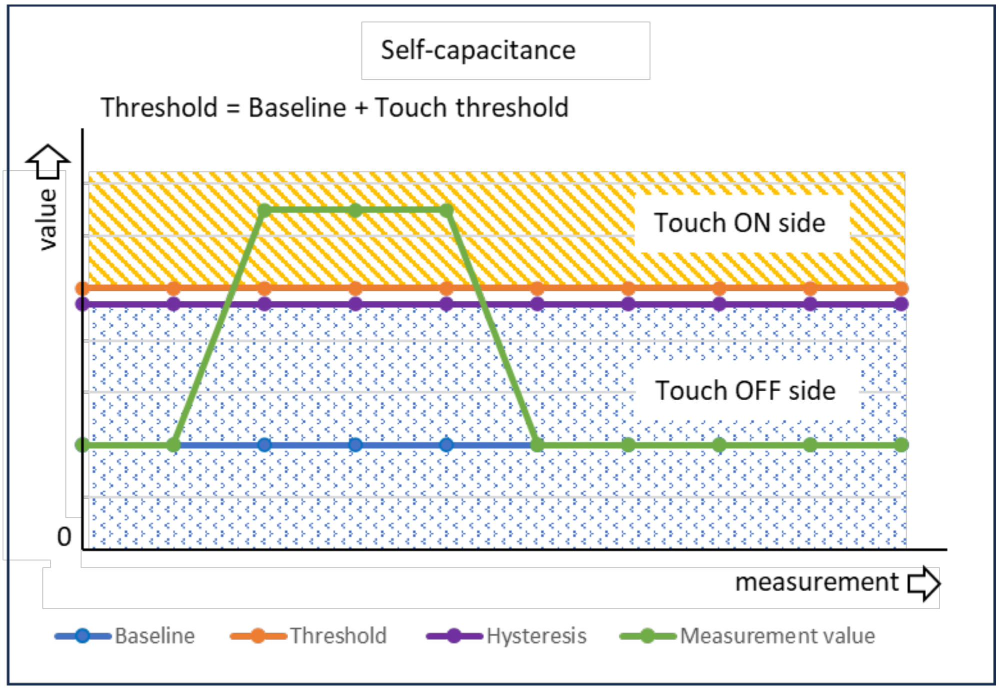

Figure 3-6 Relation of self-capacitance touch judgement

”Baseline” is a reference value for touch judgement. The first baseline is the value measured in non-touch (self-capacitance method: value after applying moving average, mutual-capacitance method: value after calculating the mutual capacitance).

”Touch threshold” is a value used for touch-ON judgement. The touch threshold is offset from the baseline.

”Hysteresis” is a value used for touch-OFF judgement. The hysteresis is offset in the touch-OFF direction from the touch threshold.

The mutual-capacitance button detects decrease in inter-electrode capacitance when being touched.

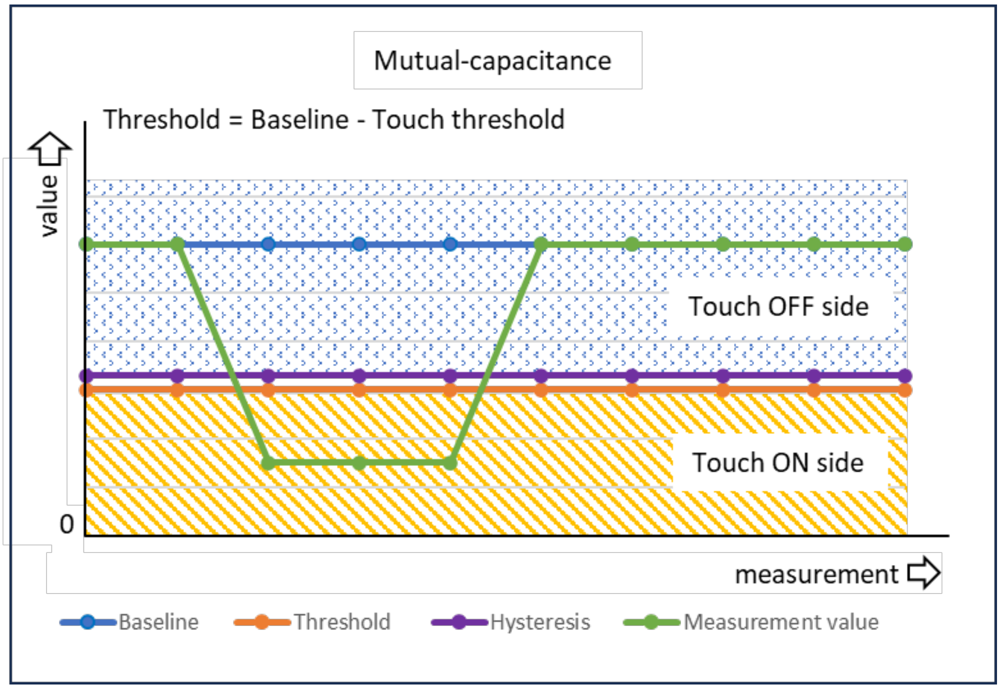

Figure 3-7 Relation of mutual-capacitance touch judgement

The following example is based on the self-capacitance button.

As an anti-chattering measure, chattering suppression (build option) is provided.

This build option is a function that complements the touch judgement function (debouncing count of touch-ON filter, or debouncing count of touch-OFF filter + hysteresis).

Set how to address the counter that counts the number of times the measurement value exceeds the touch threshold to TypeA or TypeB.

TypeA: Hold the number of times the measurement value has exceeded the touch threshold if it falls below the touch threshold but is within the hysteresis range.

TypeB: Reset the number of times the measurement value has exceeded the touch threshold if it falls below the touch threshold.

The difference between TypeA and TypeB is shown in Figure 3-8. TypeA is selected by default.

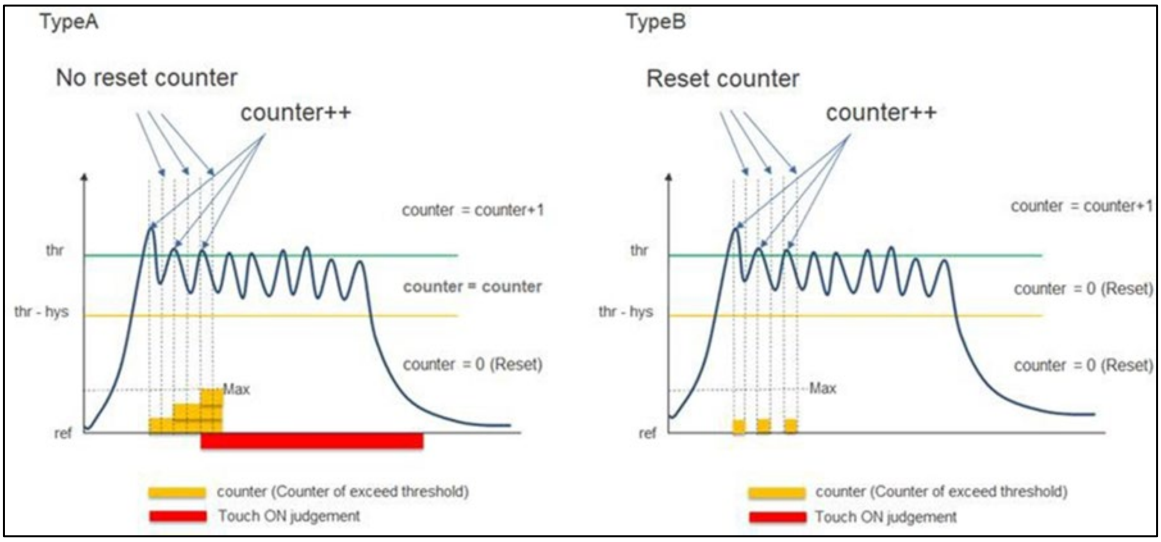

Figure 3-8 Example of chattering suppression

(1) Touch-ON Judgement

Touch-ON judgement is made by comparing the measurement value with the value of (baseline + touch threshold). Figure 3-9 shows the relation of touch-ON judgement (debouncing count of touch-ON filter = 3 (initial value)).

The condition to confirm touch-ON is that the measurement value exceeds the value of (baseline + touch threshold), and then the number of times exceeding the touch threshold exceeds the debouncing count of touch-ON filter.

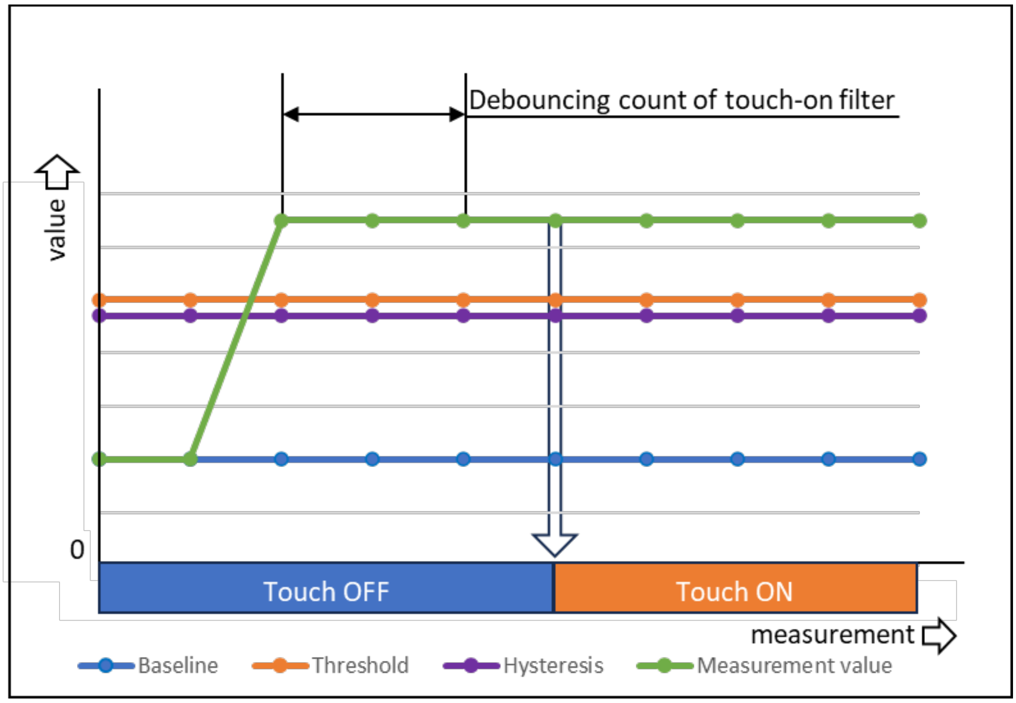

Figure 3-9 Relation of self-capacitance touch-ON judgement

(2) Touch-OFF Judgement

Touch-OFF judgement is made by comparing the measurement value with the value of (baseline + touch threshold – hysteresis). Figure 3-10 shows the relation of touch-OFF judgement (debouncing count of touch-OFF = 3 (initial value)).

The condition to confirm touch-OFF is that the measurement value falls below (baseline + touch threshold - hysteresis), and then the number of times falling below the hysteresis exceeds the debouncing count of touch-OFF filter.

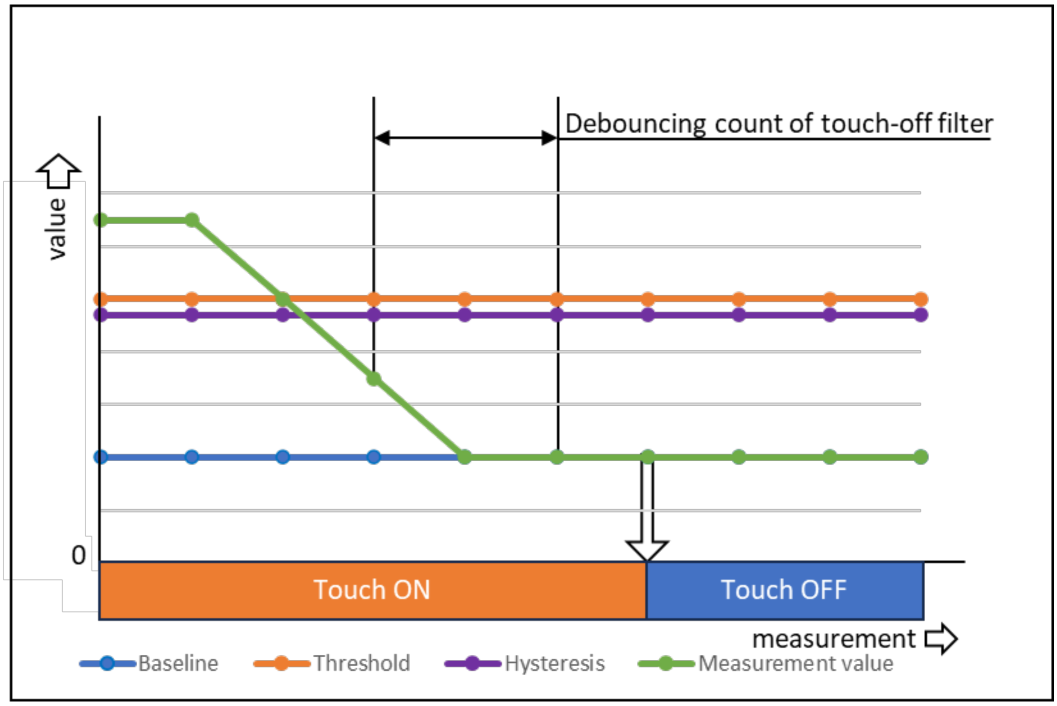

Figure 3-10 Relation of self-capacitance touch-OFF judgement

(3) Baseline and Drift Correction

The baseline is the average value in the non-contact state. Figure 3-11 shows the behavior of drift correction (update process of the baseline).

Drift correction is the operation to make the baseline follow changes in capacitance caused by environmental fluctuations. If it is non-touch state, the average value is calculated to update the baseline.

You can set the update duration with the configuration (drift_freq in touch_cfg_t). This update duration is common in buttons in the touch interface configuration.

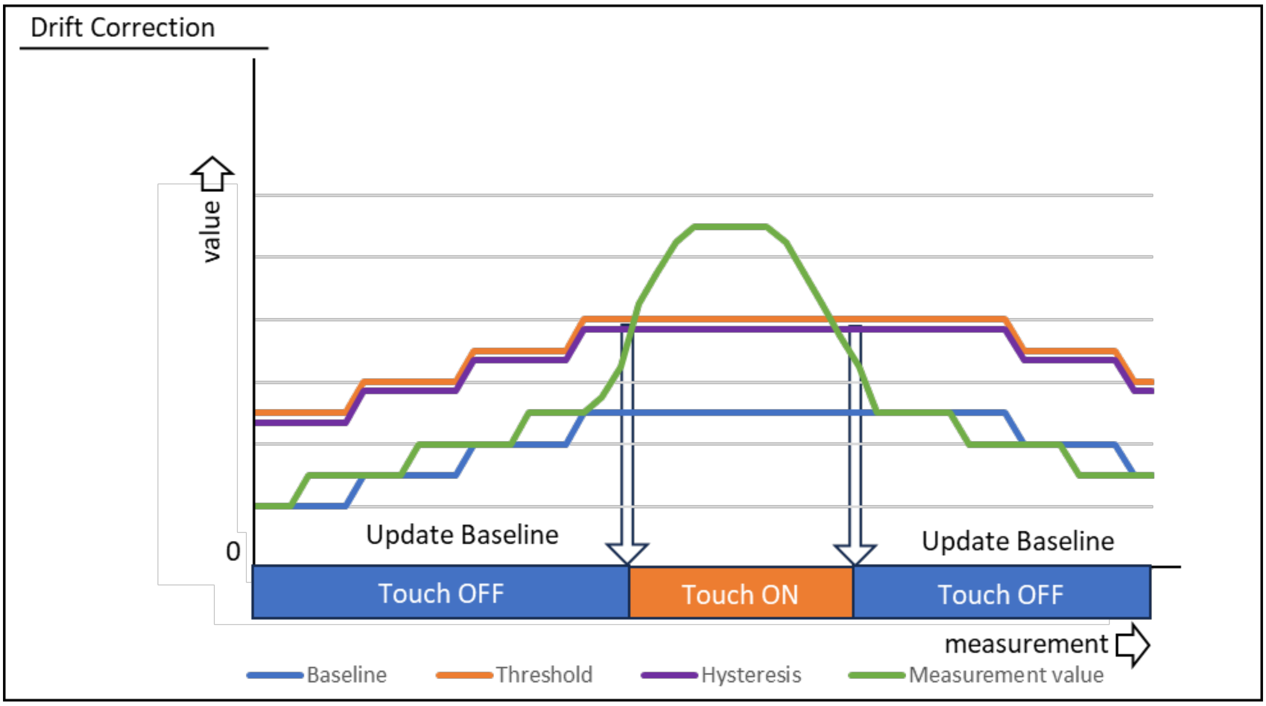

Figure 3-11 Behavior of drift correction

(4) Continuous Touch Cancel Count

The continuous touch cancel count is an operation of forcing touch-OFF when the touch-ON period exceeds a specified period. Figure 3-12 shows the relation of the continuous touch cancel count (debouncing count of touch-on filter = 3, continuous touch cancel count = 6).

Due to a sudden environmental change, it may not be possible to recover from the touch-ON status because the drift correction process cannot follow the change. To recover from this state, when touch-ON continues for a certain period, this function updates the baseline forcibly to turn the status to touch-OFF and activate the drift correction process.

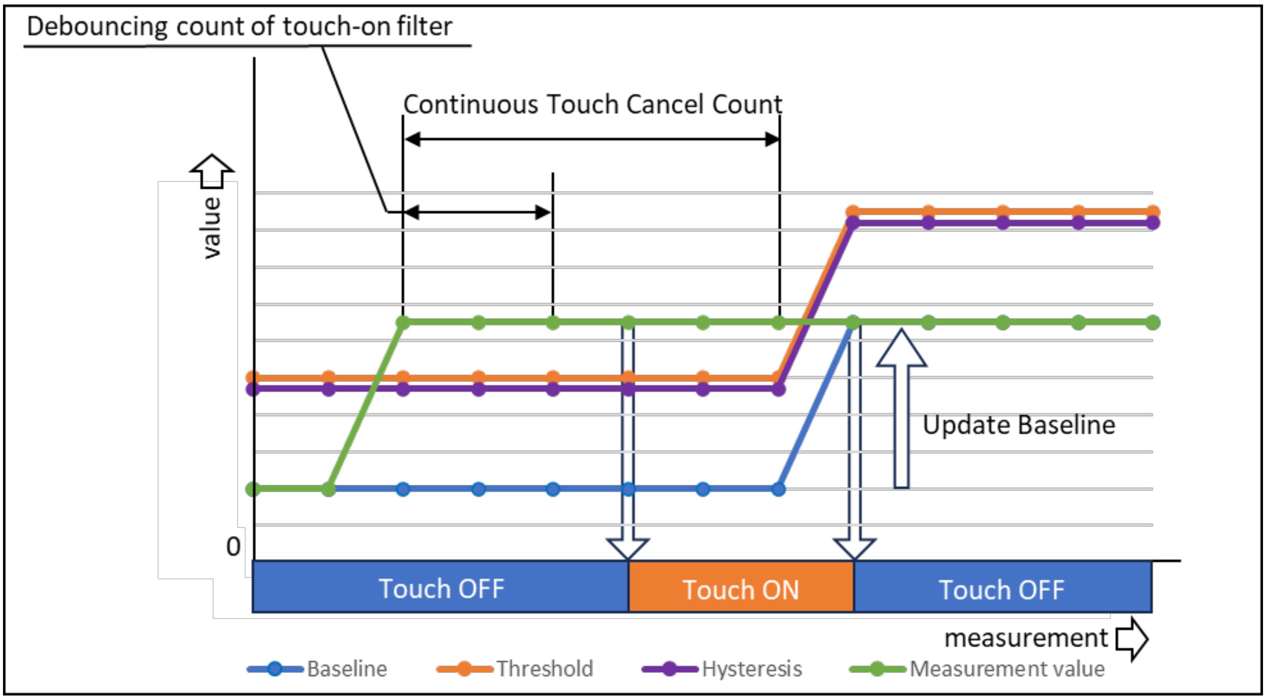

Figure 3-12 Relation of continuous touch cancel count

3.2.2 Slider Position Detection

The slider is an application of touch buttons, with touch buttons arranged in a straight line. The detection supports only the self-capacitance method. Figure 3-13 shows the position detection process.

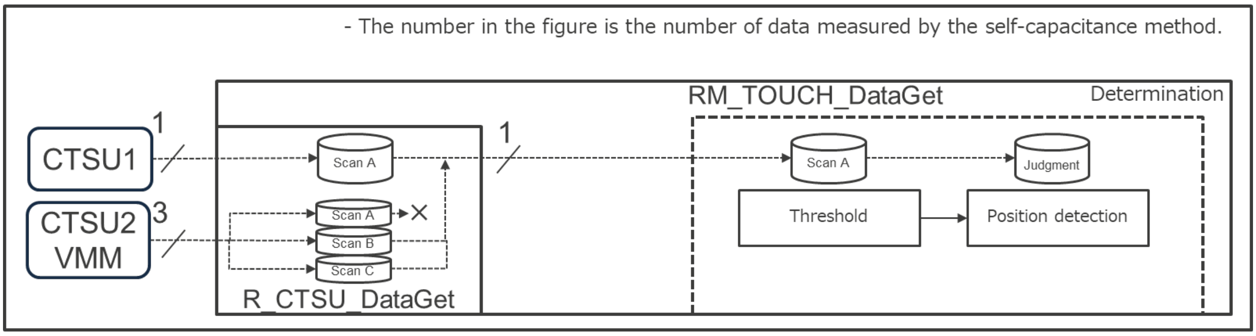

Figure 3-13 Position detection process

Use three to ten electrodes for one slider. Arrange the electrodes (TS pins) in a straight line to Configure a slider. Figure 3-14 shows the judgement for the slider.

The position of the finger on the slider is obtained from the measurement values of each TS pin.

The judgement result is output, showing the center of the touch with the value from 1 to 100 (slider resolution = 100). If there is no touch, 65535 is output.

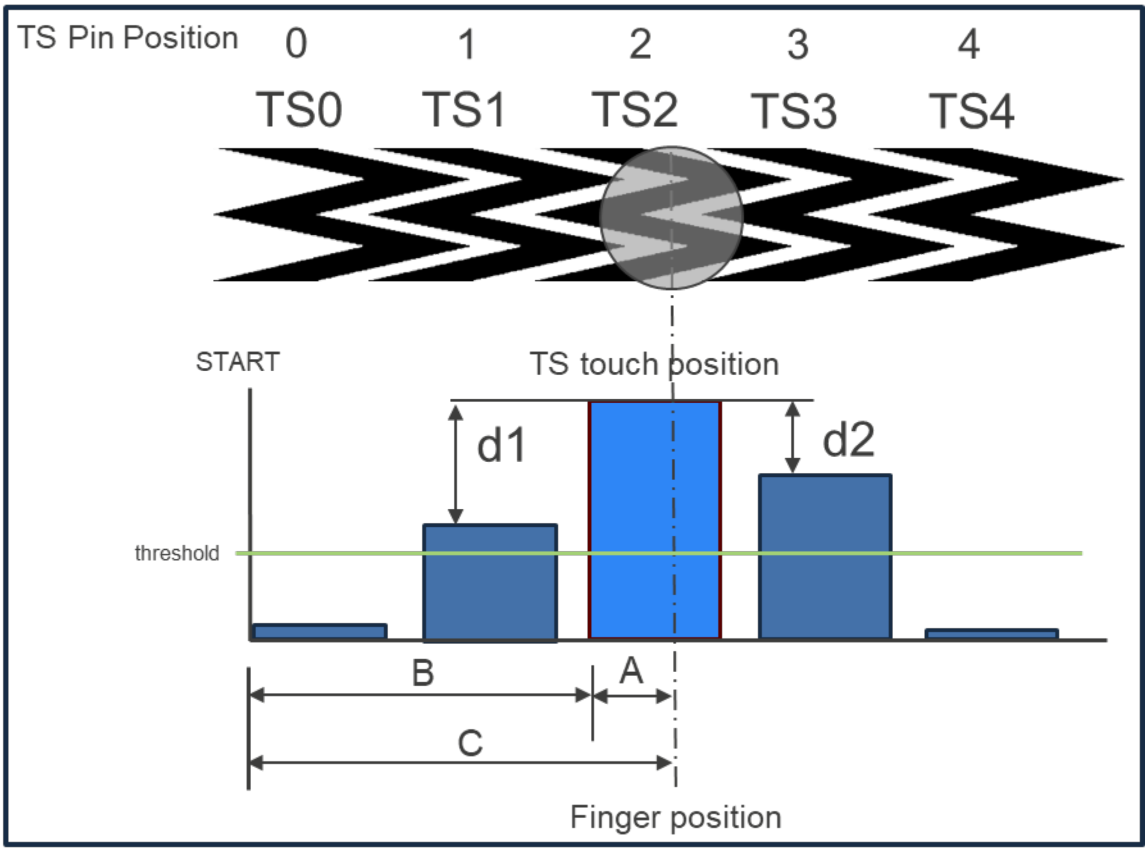

Figure 3-14 Judgement for Slider

The following explains the position detecting process using the interface in Figure 3-14 as an example. (The number of TS pins = 5.)

(1) Find the position of the TS pin with the largest measurement value (hereafter, TS touch position) among the electrodes of the slider.

(2) Calculate the differences of the measurement values (d1, d2) between the TS touch position and the both adjacent sides.

If the TS position is on the left or right end, calculate the differences by using the electrodes up to the second next from the end.

For example, if the TS touch position is on the left end TS0, calculation is as below.

- d1 is the difference between TS0 and TS1
- d2 is the difference between TS0 and TS2

If the TS touch position is on the right end TS4, calculation is as below.

- d1 is the difference between TS4 and TS3
- d2 is the difference between TS4 and TS2

(3) If the following condition is satisfied, start calculating the position. If not, end the calculation as no position detected.

- Slider threshold < d1 + d2

(4) The offset amount (B) from START to the TS touch position is as follows:

B = (Slider resolution / Number of TS pins) x TS touch position

(5) The moving amount (A) from the offset to the center of touching is as follows:

A = (Slider resolution / Number of TS pins) / (1+d2 / d1) / Number of TS pins

If the TS touch position is on the left or right end, calculate the position from the ratio of d1 and d2 with the TS touch position centered as a correction process.

(6) Sum A and B to obtain the finger position (C) on the slider.

C = A + B

(7) After that, correct it so that the result (slider output) will range from 1 to 100.

3.2.3 Wheel Position Detection

The wheel is an application of sliders, with a slider arranged in a circle. The detection supports only the self-capacitance method. Figure 3-15 shows the position detection process.

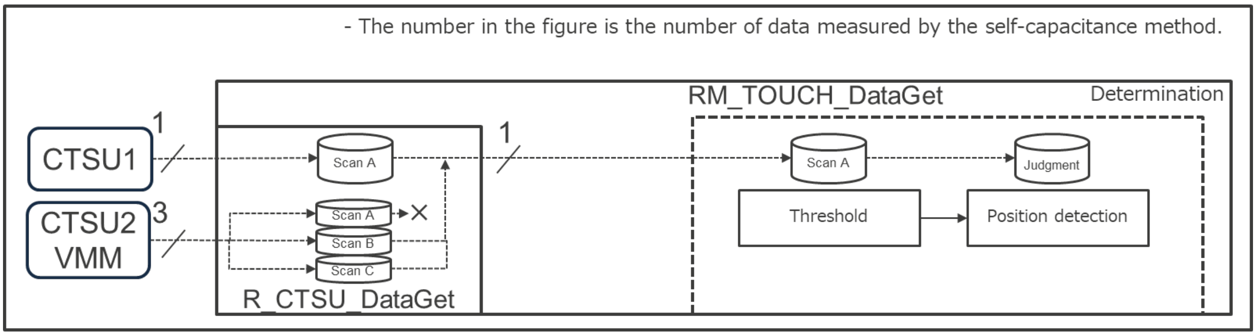

Figure 3-15 Position detection process

Use four or eight electrodes for a wheel. Arrange the electrodes (TS pins) along the circle to configure a wheel.

The position of the finger on the wheel is obtained from the measurement values of each TS pin.

The judgement result is output, showing the position of touching with the value from 1 to 360 (position resolution = 360). If there is no touch, 65535 is output.

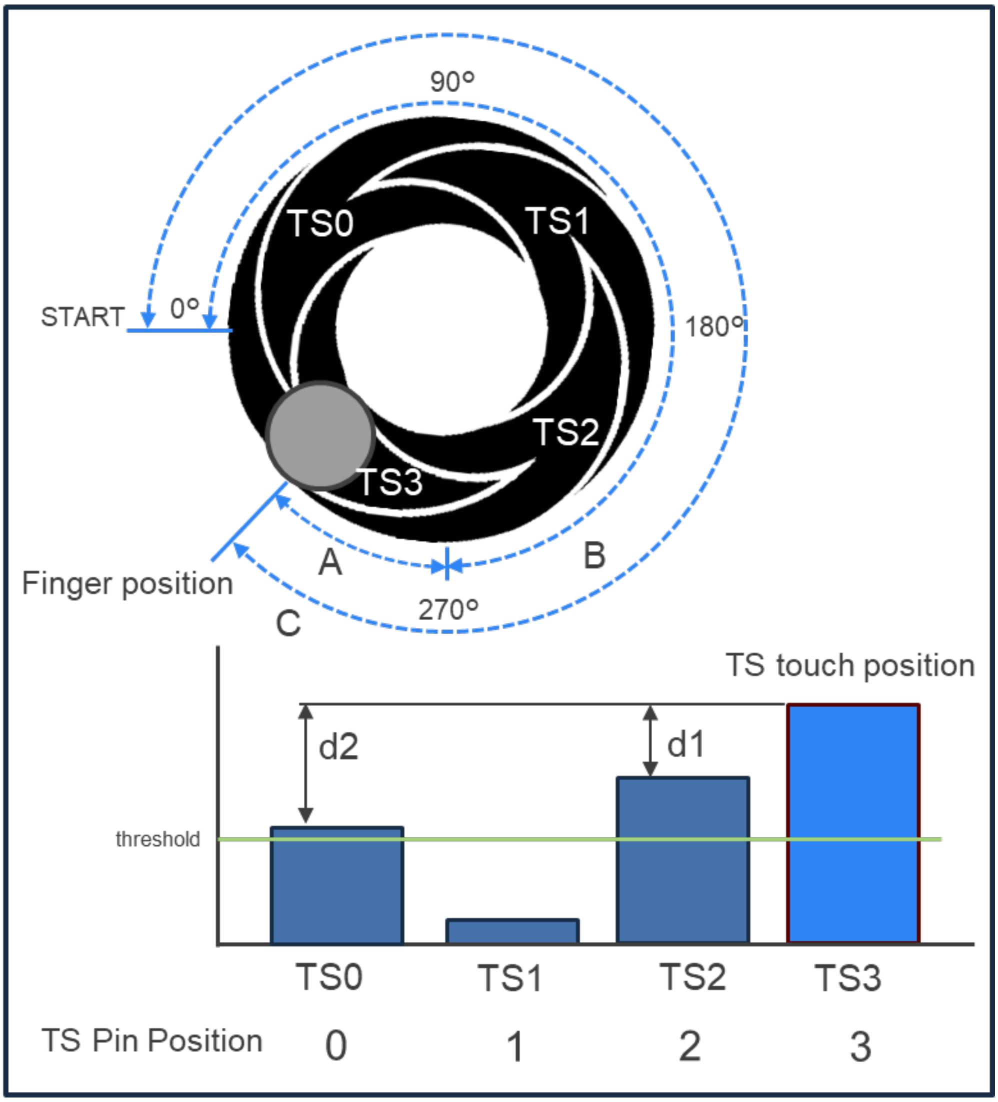

Figure 3-16 Wheel position detection

The following explains the position detection process using the interface in Figure 3-16 as an example. (The number of TS pins = 4.)

(1) Find the position of the TS pin with the largest measurement value (hereafter, TS touch position) among the electrodes of the wheel.

(2) Calculate the differences of the measurement values (d1, d2) between the TS touch position and the both adjacent sides.

If the TS position is on the starting position of the electrodes (TS0 in the Figure), d1 is the difference between TS0 and TS3, and d2 is the difference between TS0 and TS2.

If the TS position is on the ending position of the electrodes (TS3 in the Figure), d1 is the difference between TS3 and TS2, and d2 is the difference between TS3 and TS0.

(3) If the following condition is satisfied, start calculating the position. If not, end the calculation as no position detected.

Wheel threshold < d1 + d2

(4) The offset amount (B) from START to the TS touch position is as follows:

B = (Wheel resolution / Number of TS pins) x TS touch position

(5) The moving amount (A) from the offset to the center of touching is as follows:

A = (Wheel resolution / Number of TS pins) / 1 + d2 / d1)

(6) Sum A and B to obtain the finger position (C) on the wheel

C = A + B

(7) After that, correct it so that the result (slider output) will range from 1 to 360.

4. Offset Tuning

Offset tuning is to adjust the offset current to bring the measurement value close to the target value in the touch-OFF state.

This is done after the touch interface is initialized.

In offset tuning, a series of operation [Measurement start -> Acquisition of measured capacitance -> SoftwareDelay] is repeated until the measurement value gets close to the target value. In this period, the touch judgement process is not carried out.

Either offset tuning or touch judgement processing is performed depending on the internal variables shown in Table 4-1

Table 4-1 Offset tuning status

	Status	
p_instance_ctrl->tuning	CTSU_TUNING_INCOMPLETE	CTSU_TUNING_COMPLETE
Offset tuning	Yes	No
Touch judgement process	No	Yes

Ater the offset tuning is completed, the touch judgement process is carried out.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr.25.25	-	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 VIL (Max.) and VIH (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 VIL (Max.) and VIH (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

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/