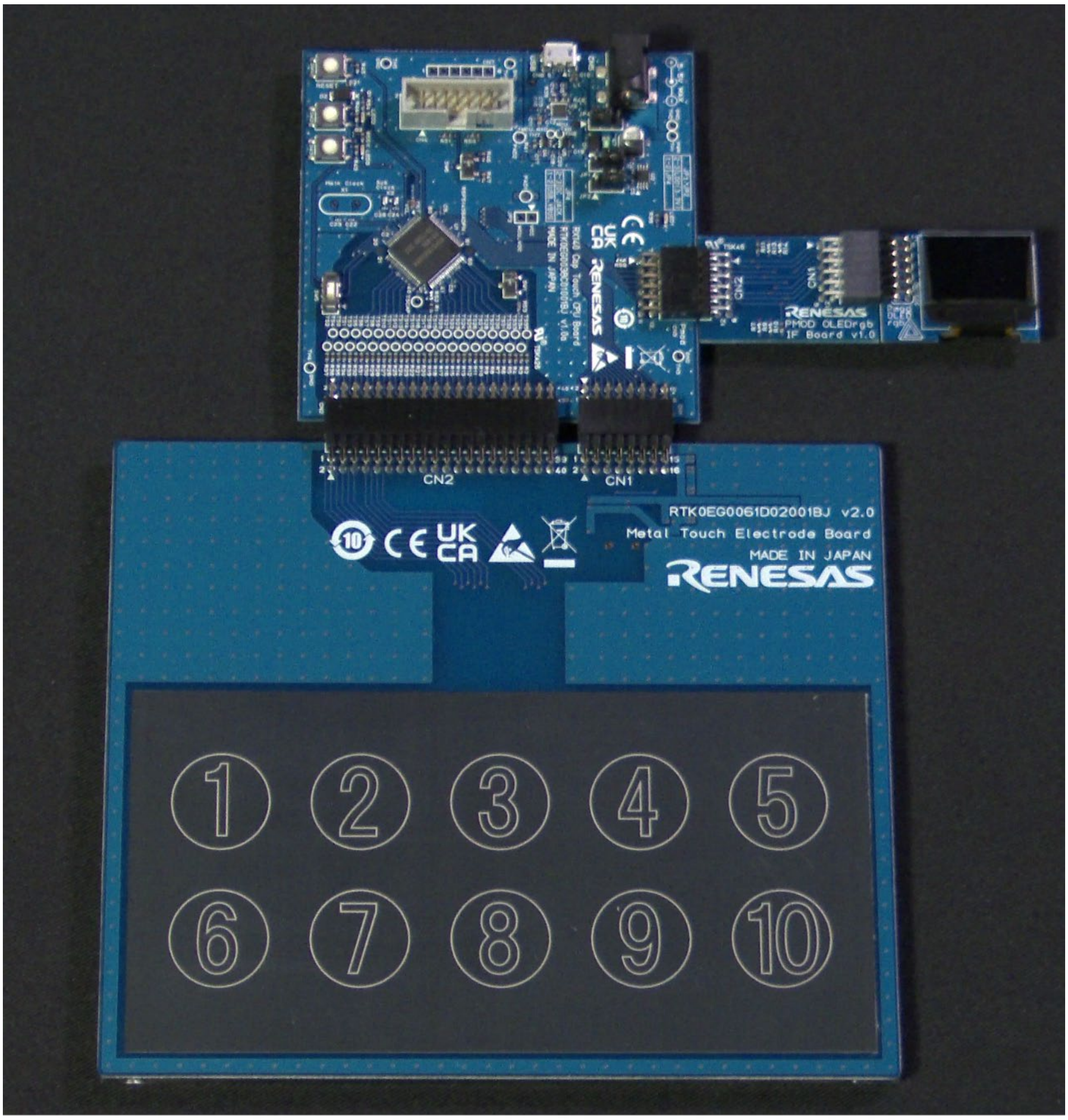
RX140 Group

Metal-touch sensor example

Introduction

This application note describes an example of a Capacitive Touch Sensor that is realized by using RX140 Capacitive Touch Unit (hereafter, CTSU) and a metal panel as an overlay on touch sensor electrodes (covering material to protect touch sensor electrodes and stabilize characteristics of capacitive touch detection). This is a method of measuring the change in capacitance caused by deflection when a finger presses the metal panel and judging the touch. Since the metal panel touch sensor measures deflection, it can judge a touch in multiple levels by judging its pressing strength with a single button. As a metal casing is available as it is, using this method can reduce the cost for preparing overlays separately for touch and allow the appearance of buttons to be sophisticated.



Contents

1. Overview

This application note is a guide for system designers, hardware designers, and software designers who develop a capacitive touch sensing system using a metal panel as an overlay.

The touch sensor with a metal panel overlay is different from conventional touch sensors, and it can detect the state and strength of a touch by capturing deflection based on the change in capacitance caused when the metal panel is pressed. Therefore, it can detect a touch with non-conductive gloves and is not affected by drops or stains. Compared to conventional mechanical buttons, it needs no mechanical moving parts, achieves higher resistance against water and stains, and makes it possible to implement button functions with higher reliability under a wider range of environments.

This document describes the detecting principle, features, configuration and schematics of the demo kit, information on designing the casings, and software configuration of the capacitive touch sensor with a metal panel. For demo kits, please contact us at the following:

https://www.renesas.com/buy-sample/locations

In this document, names of MCU-mounted Capacitive Touch Sensors CTSU/CTSUa/CTSUb are represented as CTSU1, and CTSU2/CTSU2L/CTSU2La/CTSU2SL are represented to as CTSU2.

For basic handling of the Capacitive Touch Sensor Unit, details on the unit itself, and features and differences of CTSU1 and CTSU2, refer to the application note “Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide”.

2. System Operation

This system operates and displays touch detection in the following two modes.

- Operating mode 1: Detects a button electrode being pressed, sounds a buzzer, and displays the press on the OLED.
- Operating mode 2: Detects the pressing strength in three levels, sounds a buzzer, and displays the pressing strength on the OLED.

Table 2-1 and Figure 2-1 show settings, operation, and display of the board. When using this system, make sure to connect pins 2 and 3 of JP1 and JP2 before supplying power.

Table 2-1 Board settings, operation, and display

Category	Item		Settings, Operation, Display	
Preparation	JP	Select the LDO power	JP1	2-3
			JP2	2-3
		Select the power supply	JP4	USB: 1-2 DC Jack: 2-3
		Use TS2 and TS4	SW5	OFF (lower)
	Power supply	Input 5V from USB or DC jack		
Operation	SW3	Switch the operating mode	Default: Operating mode 2	
Display	LED3	Display the operating mode	OFF: Operating mode 1 ON: Operating mode 2	
	OLED	Display the operating mode and button pressing state	Operating mode 1	Pressing state: 0, 1
			Operating mode 2	Pressing strength: 0, 1, 2, 3

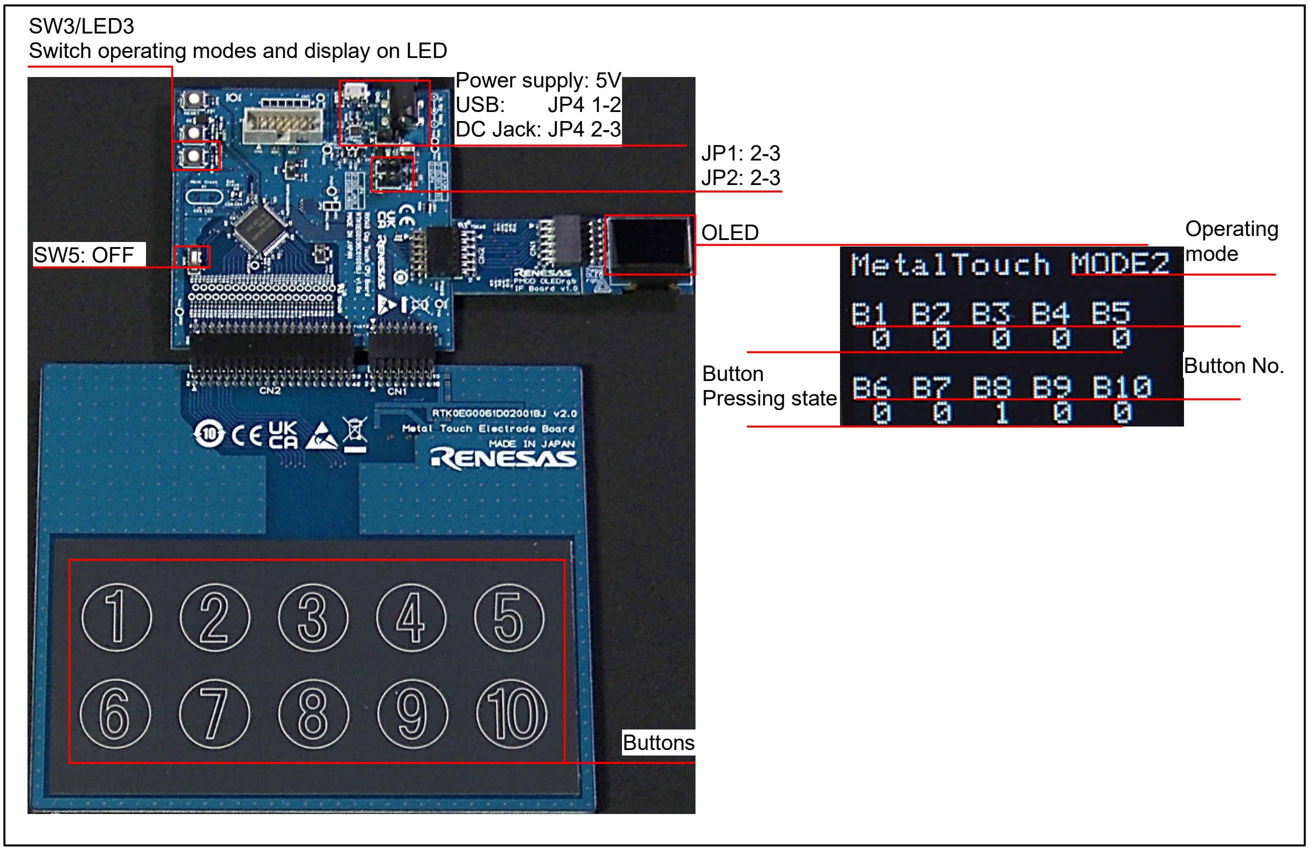


Figure 2-1 Settings, operation, and display

3. Environment for Operation Confirmation

Table 3-1 Environment for Operation Confirmation

Item		Name
CPU board		RX140 Capacitive Touch Evaluation System CPU board (RTK0EG0038C01001BJ)
	MCU	RX140 (R5F51406ADFN)
Electrode board		Metal Touch Electrode Board (RTK0EG0061D02001BJ)
Display connection board		PMOD OLEDrgb IF Board
Display		Digilent Pmod OLEDrgb (410-323)
IDE		e2 studio Version 2025-01 RX Smart Configurator V25.1.0 QE for Capacitive Touch V4.1.0
	FIT modules	QE TOUCH module V3.1.0 QE CTSU module V3.1.0 DTC module V4.5.0
Tool chain		CC-RX V3.07.00
Emulator		E2 emulator Lite

4. Related Documents

- R30AN0424 Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide
- R12UZ0102 RX140 Group Capacitive Touch Evaluation System User's Manual
- R01AN4516 RX Family Using QE and FIT to Develop Capacitive Touch Applications
- R01UH0905 RX140 Group User's Manual: Hardware

5. Operating Principle

5.1 Basic Model of Capacitor

Figure 5-1 shows the configuration of a parallel-plate capacitor, which is a basic model of capacitors.

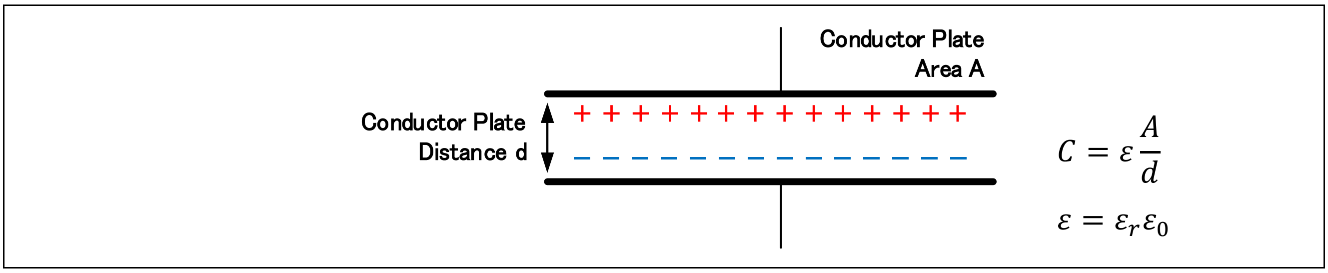


Figure 5-1 Parallel-plate capacitor

C: Capacitance of the parallel-plate capacitor [F]

A: Area of two plates overlapping [m^2]

ε_r: Relative permittivity of the material between the electrodes (unitless)

ε_0: Electric constant $(\varepsilon_0 \approx 8.854 \times 10^{-12})$ [F/m]

d: Distance between two plates [m]

A parallel-plate capacitor is composed of two plate electrodes, and its capacitance is proportional to the size of the two electrodes and the permittivity of the material between the two electrodes and is inversely proportional to the distance between the electrodes.

5.2 Capacitive Touch Sensor with Non-Conductive Overlay

Figure 5-2 shows the configuration of a typical capacitive touch sensor using a self-capacitance method with a non-conductive overlay. In a typical capacitive touch sensor, the capacitor consists of the electrode formed on the board and a finger. The electrode is connected to GND on the board through the parasitic capacitance C_p. When a finger comes close to the electrode, the capacitance C_s between the electrode and the finger is added to the parasitic capacitance C_p between the electrode and GND of the PCB. By measuring the amount of this increase in capacitance, a touch can be detected. In this method, a non-conductive overlay such as an acrylic board or glass plate is placed on the electrode. When a material with a higher relative permittivity than air is used as an overlay, improvement of sensitivity is expected. Furthermore, using a material with high insulation performance can reduce risks of damage or malfunctions of the mounted devices caused by electrostatic discharge (ESD).

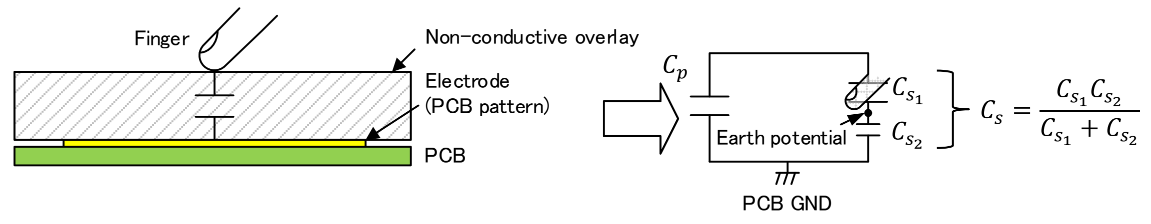


Figure 5-2 Configuration of typical capacitive touch sensor

C_p : Parasitic capacitance between the electrode and PCB GND [F]

C_s : Capacitance between the finger and PCB GND [F]

C_{s_1}: Parasitic capacitance between the finger and the earth potential [F]

C_{s_2}: Parasitic capacitance between the earth potential and PCB GND [F]

5.3 Capacitive Touch Sensor with Conductive Overlay (Metal Panel Touch Sensor)

Figure 5-3 shows the configuration of the metal panel touch sensor with a conductive overlay. The distance between the metal panel and the board is fixed with spacers, and the metal panel is connected to GND of the board. When applying force to the metal panel, a small strain occurs. This strain changes the distance between the metal panel and the electrode formed on the board, causing change in capacitance C_s between the metal panel and the electrode. As this system operates as a sensor that measures the distance between the metal panel and the electrode with capacitance, it can measure strength of the pressing force against the metal panel with the change in capacitance.

Since the metal panel is stable at the GND potential, the parasitic capacitance of human body does not affect on measurement, and a touch can be detected even if the user wears non-conductive gloves. Water drops on the surface of the metal panes do not affect as well.

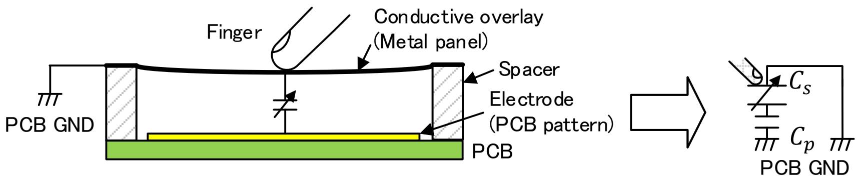


Figure 5-3 Configuration of metal panel touch sensor

C_p : Parasitic capacitance between the electrode and PCB GND [F]

C_s : Capacitance between the metal panel and PCB GND [F]

5.4 Measurement Principles of CTSU

Figure 5-4 shows the measurement circuit of CTSU.

CTSU measures the capacitance by outputting a sensor drive pulse from TS terminal and measuring the charge/discharge current. The following equation is established when the electrode-side current is I, sensor drive pulse frequency is F, parasitic capacitance is C_P, capacitance of detection target is C_s, and sensor drive pulse voltage is V.

$$I = F(C_s + C_P)V$$

The current I is the sum of the current I_1 supplied from VDC (Voltage Down Converter) for measurement and the current I_2 supplied from the offset DAC.

For the current I_1 supplied from the measurement VDC, a proportional current I_{OUT} is applied to the current-controlled oscillator (hereinafter, CCO) via the current mirror. CCO outputs a pulse-frequency proportional to I_{OUT} to the sensor counter. The sensor counter measures the sensor drive pulse for a fixed period and stores the measurement value of the current I_{OUT} in the sensor counter register.

The amount of current (measured value) differs when only the parasitic capacitance C_p is used and when a C_s is applied due to finger-contact etc. By using this, the change in capacitance is detected from the change in the current measurement value.

The capacitor connected to TSCAP terminal has the role of stabilizing the internal voltage.

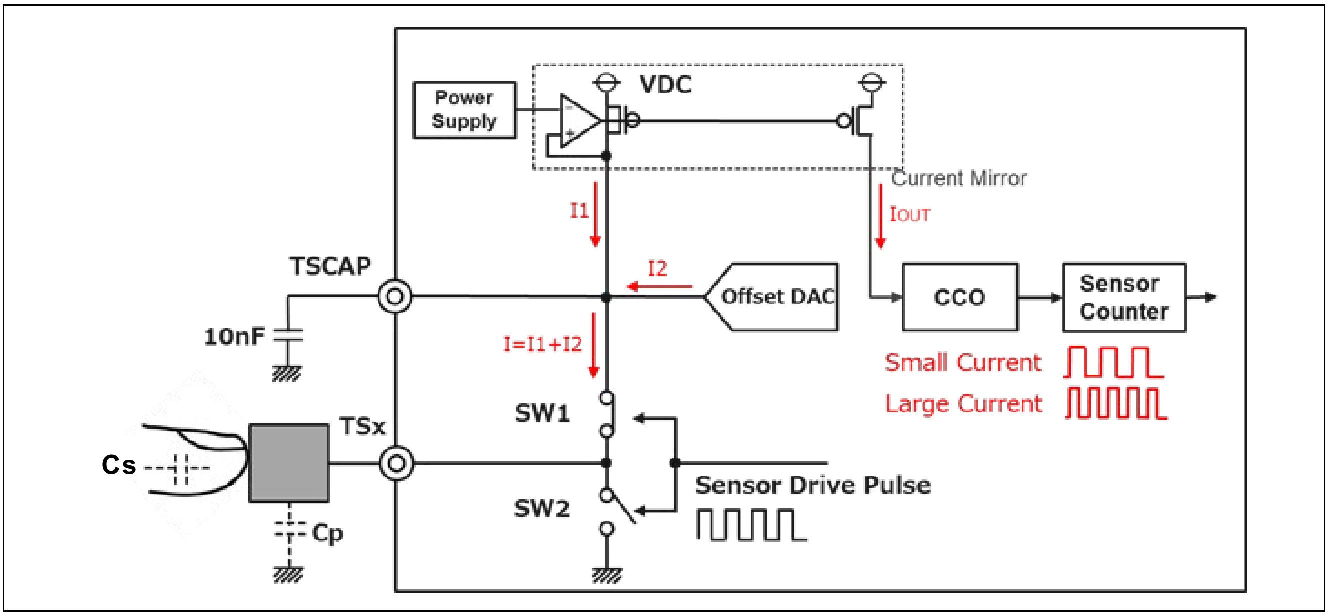


Figure 5-4 Measurement circuit of CTSU

6. System Configuration

Figure 6-1 shows the system configuration of the demo kit. This demo kit consists of four boards: CPU board, touch electrode board, PMOD OLEDrgb IF board, and PMOD OLEDrgb (display).

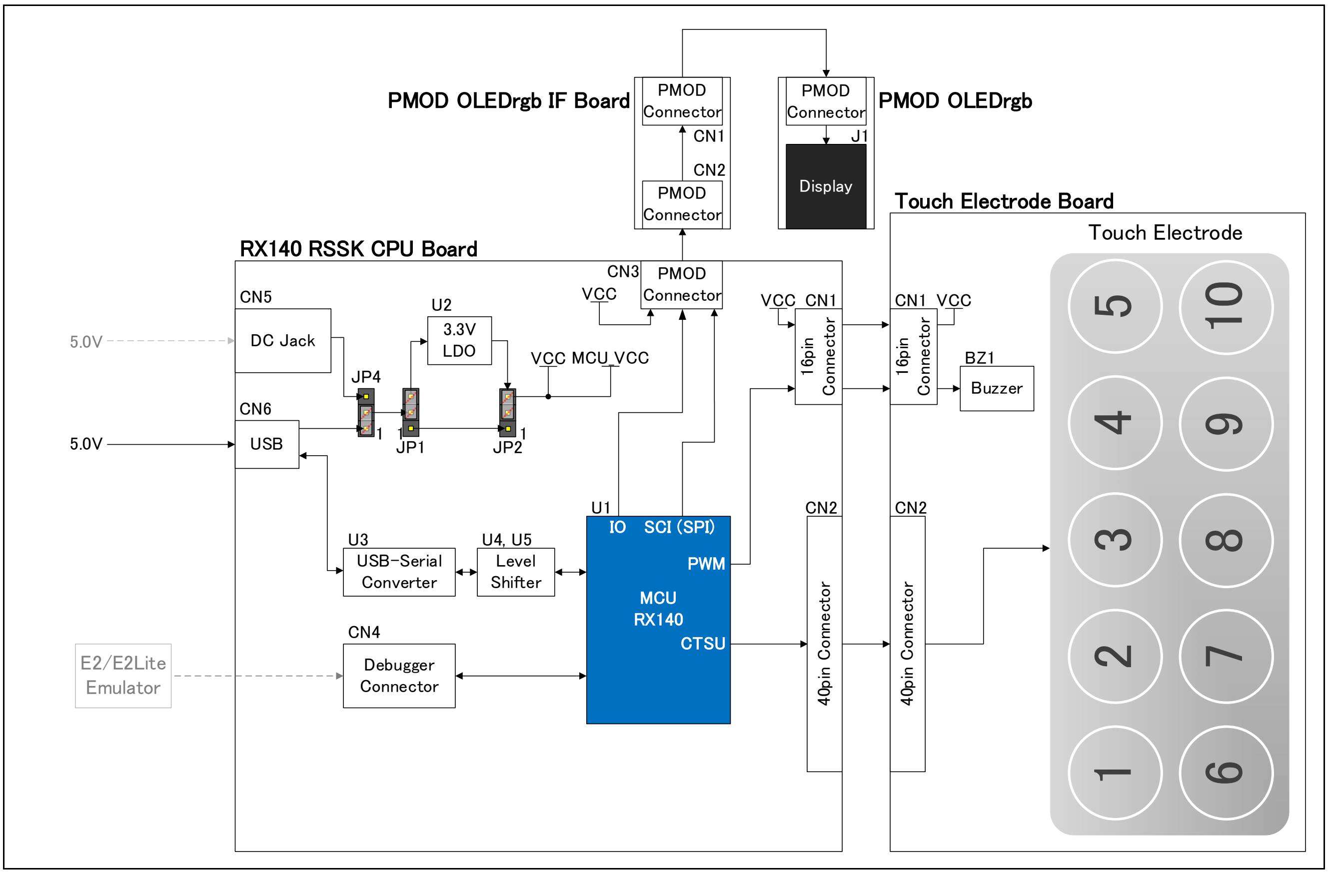


Figure 6-1 System configuration of demo kit

6.1 Basic Configuration

6.1.1 CPU Board (RX140 RSSK)

Figure 6-2 shows the external appearance of the CPU board (RX140 RSSK). The CPU board is equipped with RX140 and supplies power to each board, communicates to a PC, measures capacitance of the touch electrode board, judges a touch, and communicates to PMOD OLEDrgb. As the maximum rating of PMOD OLEDrgb is 3.5V, close the pins 2 and 3 of JP1 and JP2 with jumper sockets and use the 3.3V LDO of U2. (Refer to Table 2-1.)

For details on the CPU board, refer to “RX140 Group Capacitive Touch Evaluation System User’s Manual.”

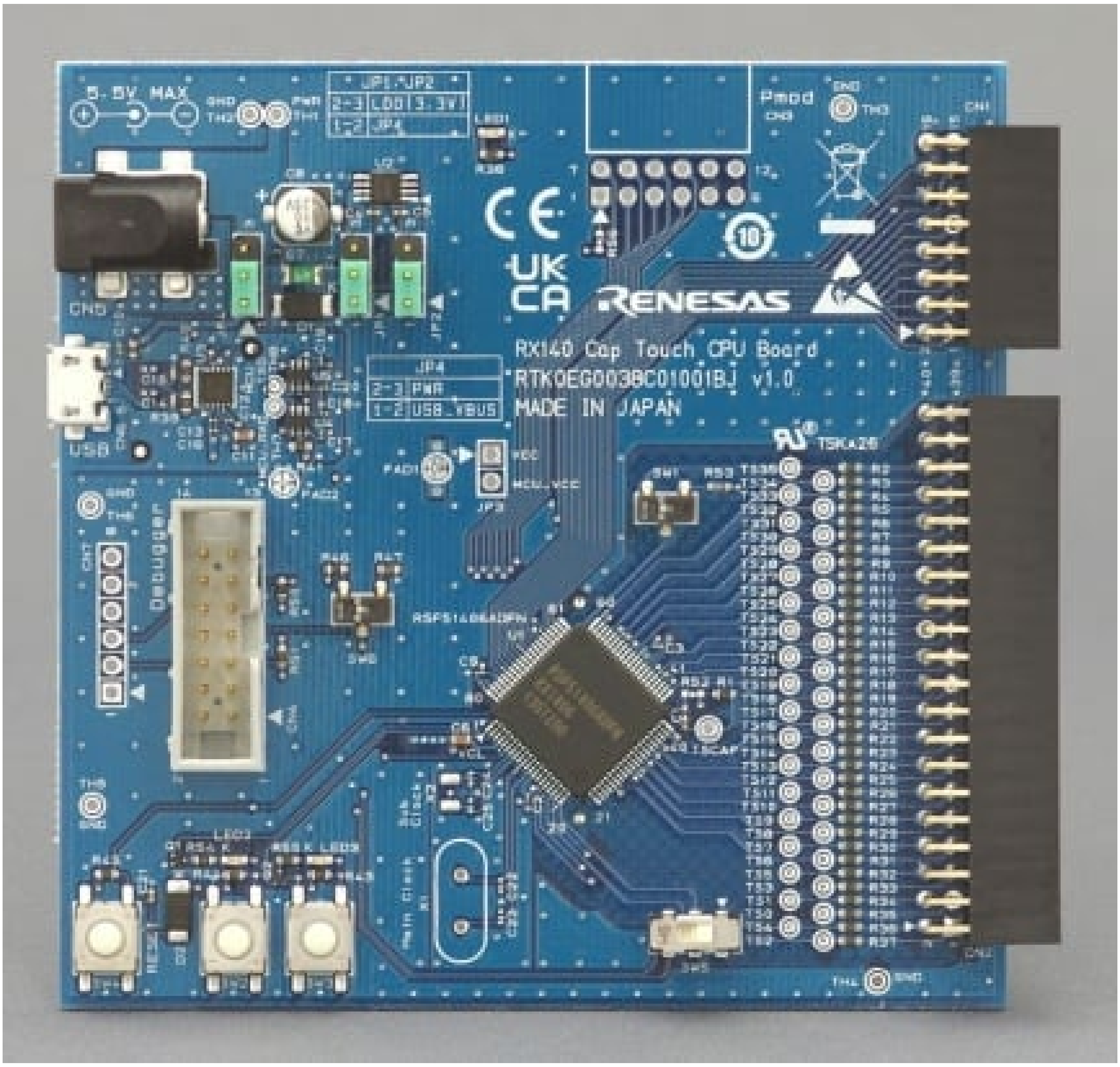


Figure 6-2 External appearance of CPU board (RTK0EG0038C01001BJ)

6.1.2 Touch Electrode Board

Figure 6-3 shows the external appearance of the touch electrode board. The touch electrode board mounts ten touch electrodes and a buzzer that sounds when a touch is detected. Figure 6-4, Figure 6-5, and Figure 6-6 show diagrams of the touch electrode board. The touch electrode board is a double-sided board. On the component side, ten circular electrodes with a diameter of 20mm are mounted so that the centers of each electrode are 30mm apart. On the solder side, a shield is placed to protect the electrodes from noise. The shield is mesh-formed to prevent an increase in parasitic capacitance between the electrodes and GND. On the four corners of the metal panel mounting positions, spring-loaded pins (TP3 to TP6) are placed to secure conduction between the metal panel and GND.

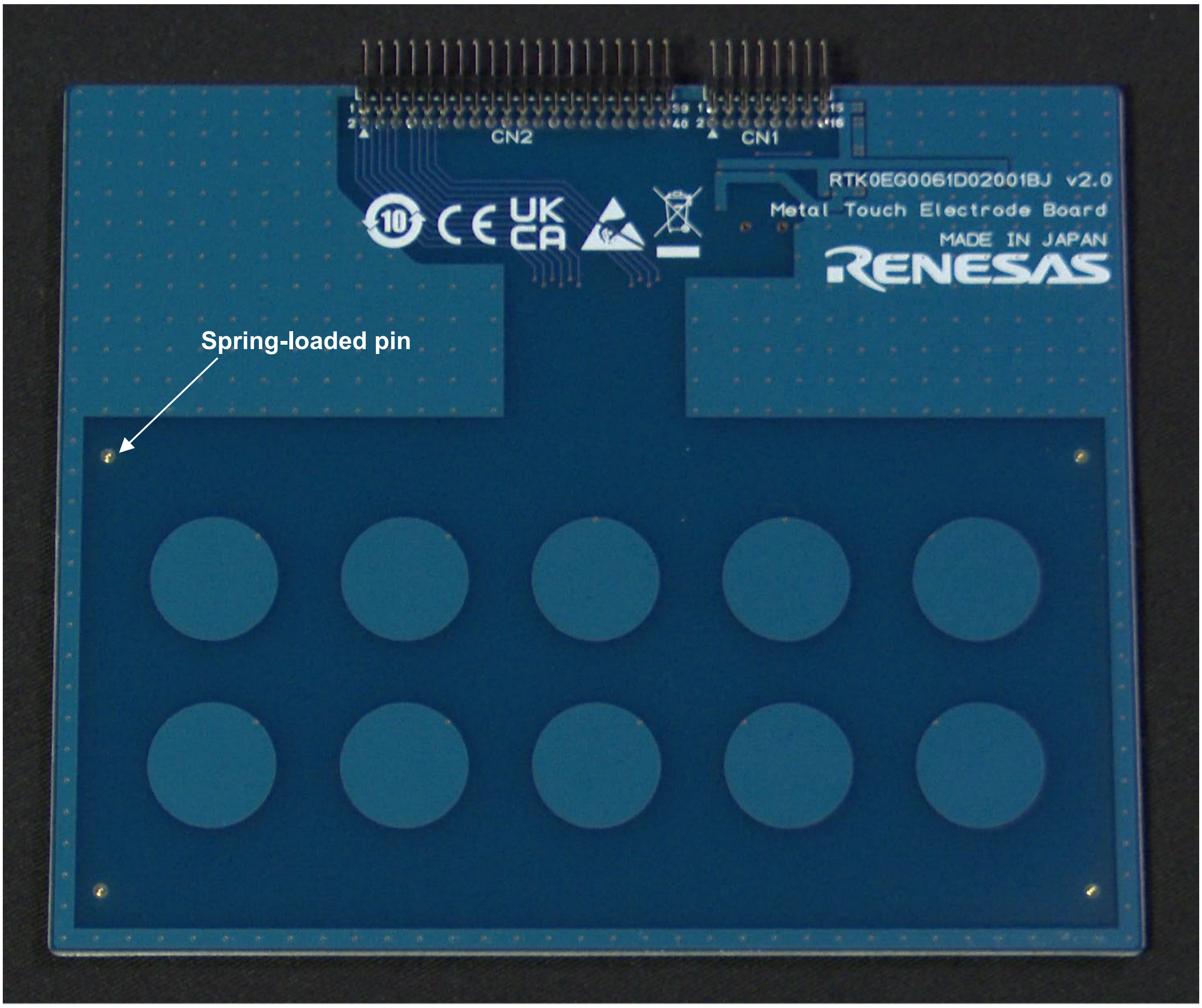


Figure 6-3 External appearance of touch electrode board (before attaching metal panel)

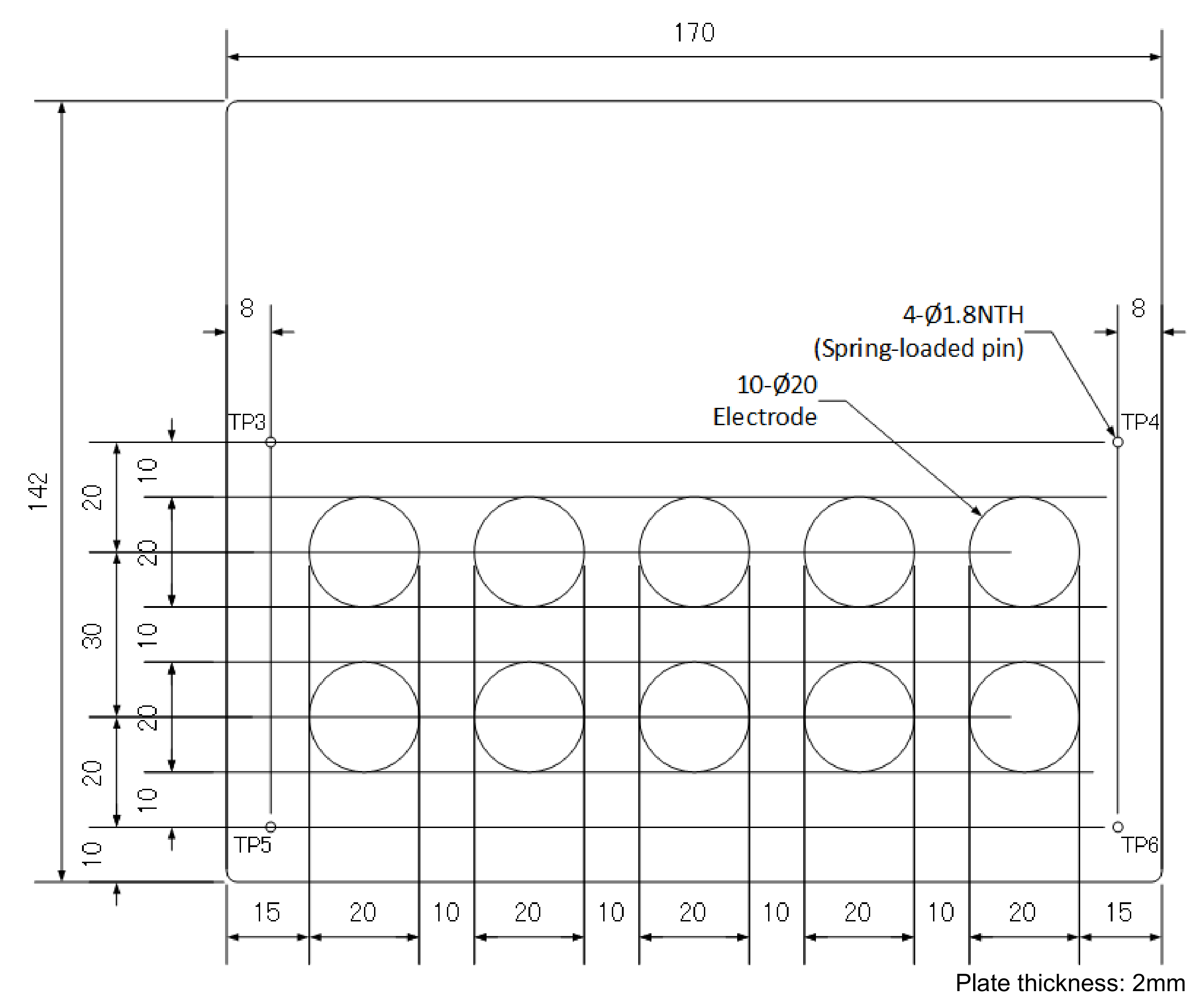


Figure 6-4 Dimension of electrode section of touch electrode board (unit: mm)

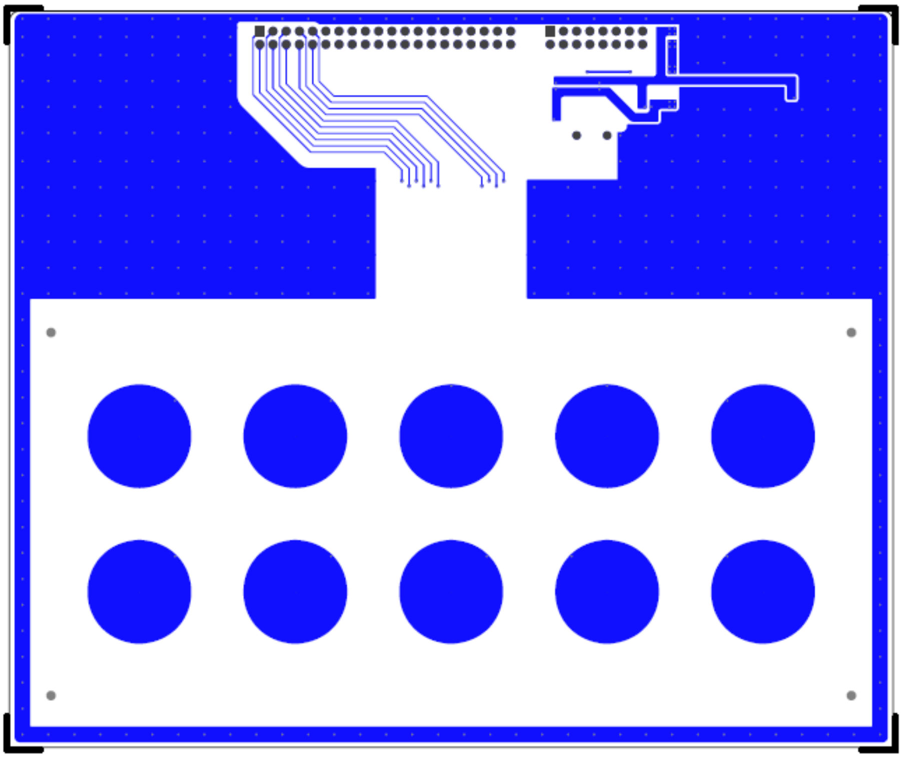


Figure 6-5 Component-side pattern of touch electrode board

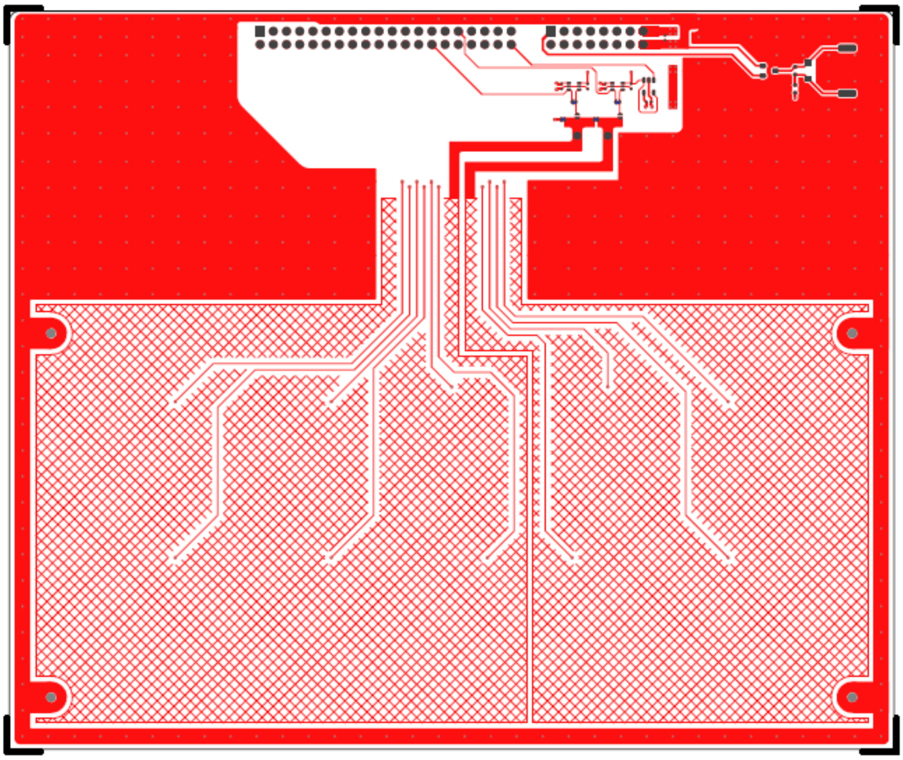


Figure 6-6 Solder-side pattern of touch electrode board (view of component surface)

6.1.3 PMOD OLEDrgb IF Board

The PMOD OLEDrgb IF board is a board to connect the CPU board and PMOD OLEDrgb. The CPU board provides a PMOD connector on CN3, which supports PMOD Type 6A (Extended I2C). As PMOD OLEDrgb supports PMOD Type 2A (Extended SPI), this IF board connects PMOD OLEDrgb and the SPI-compatible pin of the CPU.

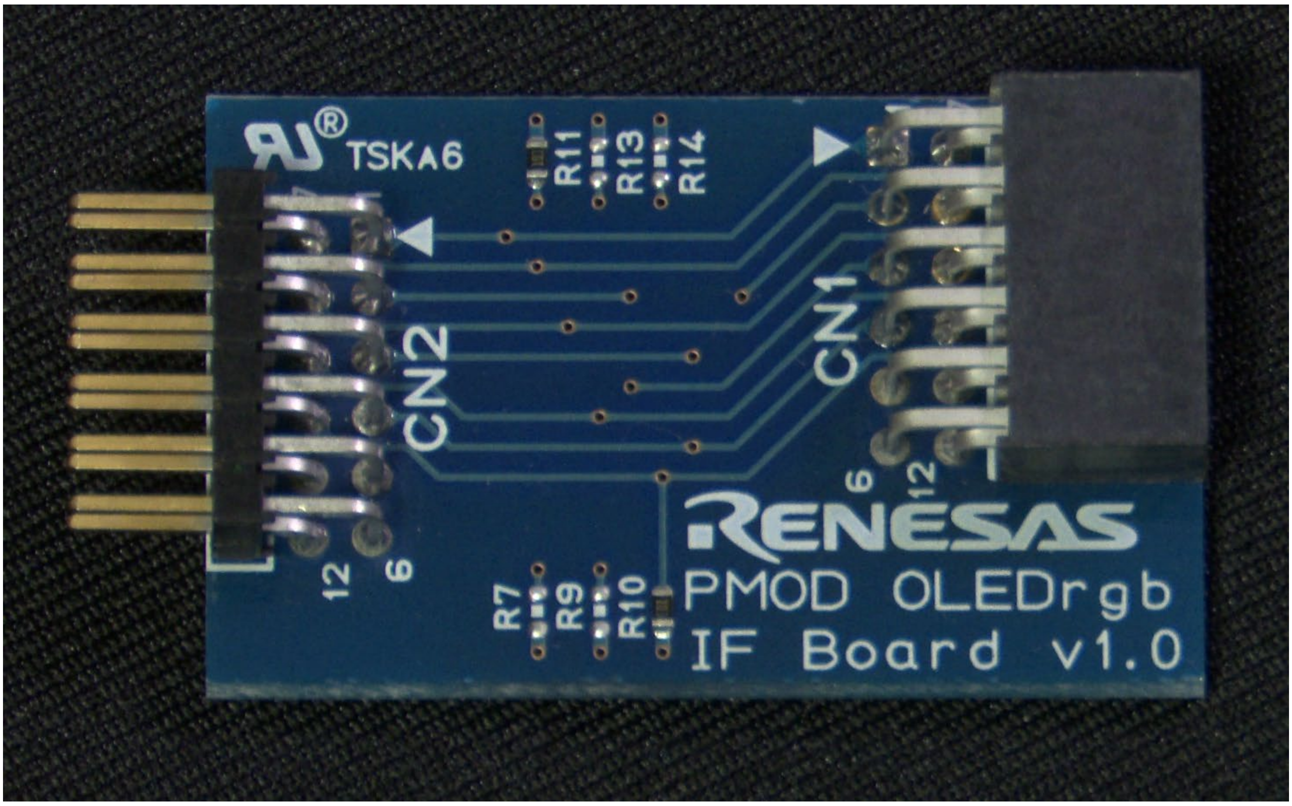


Figure 6-7 External appearance of PMOD OLEDrgb IF board

6.1.4 PMOD OLEDrgb (Display)

PMOD OLEDrgb is a display module manufactured by Digilent, supporting PMOD Type 6A (Extended I2C). In this system, it displays pressed buttons and the pressing strength.

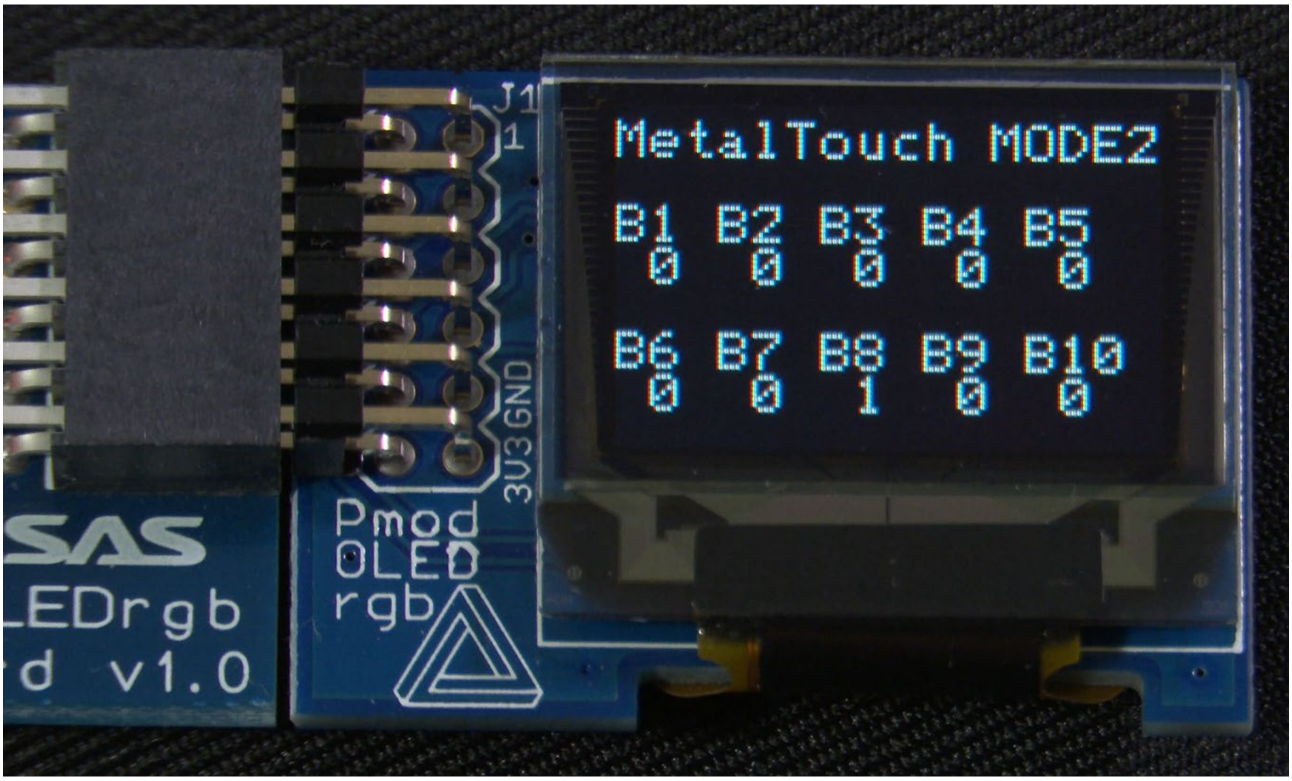


Figure 6-8 External appearance and display example of PMOD OLEDrgb

6.2 Machine Configuration

6.2.1 Configuration of Sensing Part

Figure 6-9 shows the configuration of the sensing part of the metal panel touch sensor. The sensing part mainly consists of three layers: The first layer is a conductive overlay (metal panel) layer, and metal plates are used in general. The second layer is a spacer layer, which has a role of fixing the distance between the first and the third layers. The third layer is a PWB layer, forming a capacitor between the electrodes formed on the board and the first layer (metal panel).

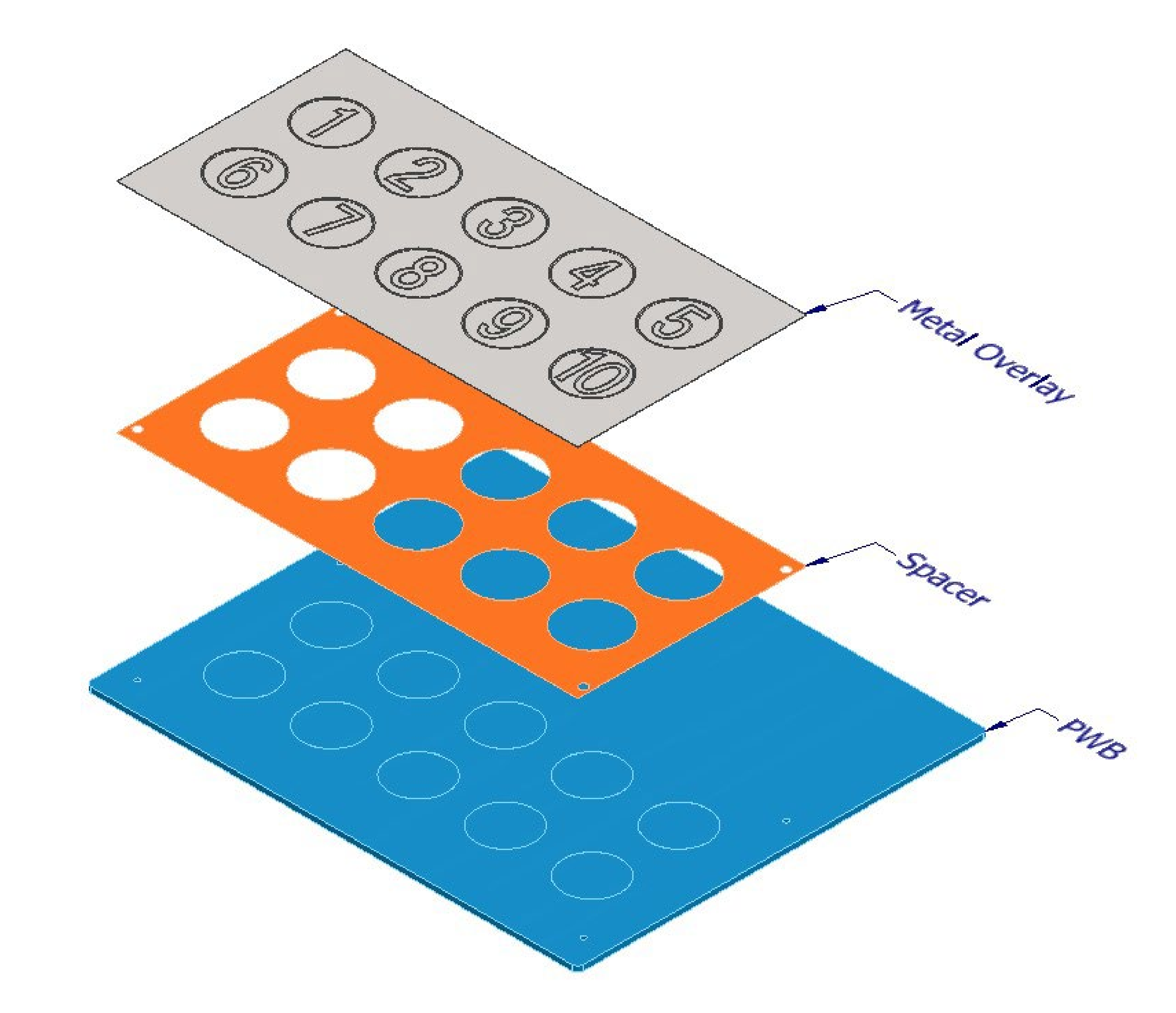


Figure 6-9 Configuration of sensing part of metal panel touch sensor

6.2.2 Metal Panel Layer

For the metal panel, use a material such as stainless steel, aluminum, etc. Process the front side of the panel by laser processing, etching, printing, and so on in order to indicate the location of sensors. Connect the metal panel to the fixed potential like the board GND. If the potential with the electrodes on the board layer is not stable, the reliability of touch judgement and noise immunity are significantly reduced. In this system, the front side of the metal panel is printed by etching. The back side is etched according to the location of the electrodes. The following are expected by etching:

・When pressure is applied on any buttons that are not being pressed or any surrounding area, deflection is suppressed, and detection errors are reduced.

・By making the metal panel of the button section thinner, the panel is more likely to deflect, and sensitivity is improved.

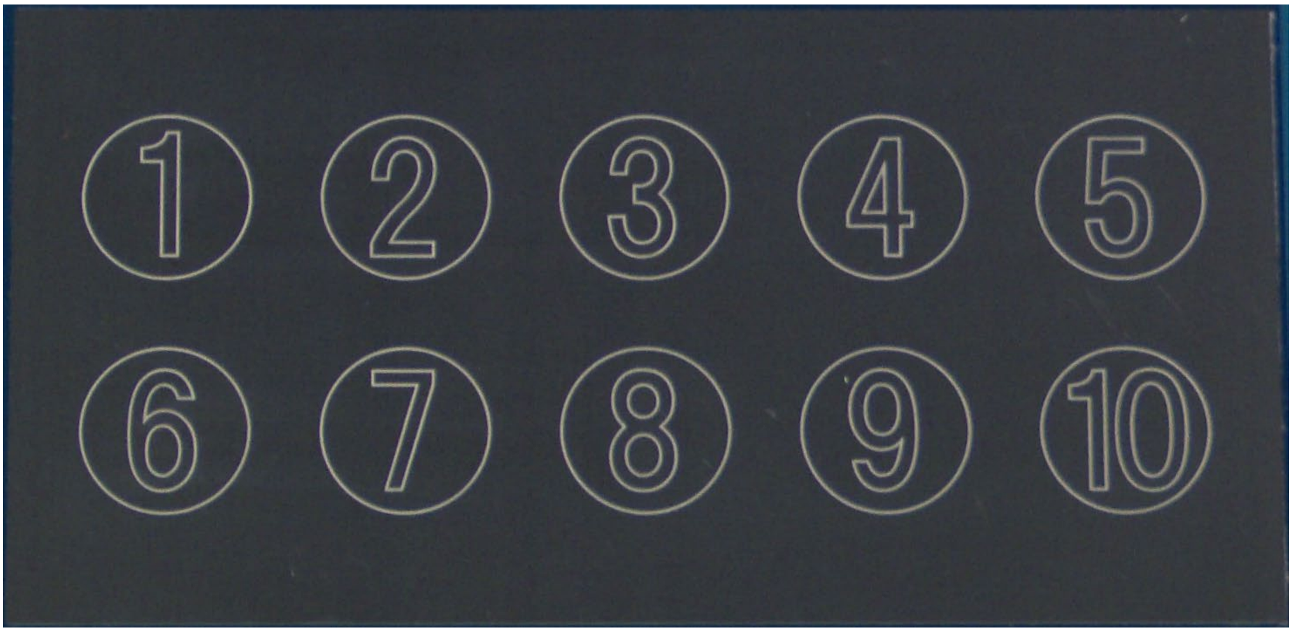


Figure 6-10 External appearance of metal panel layer (print by etching)

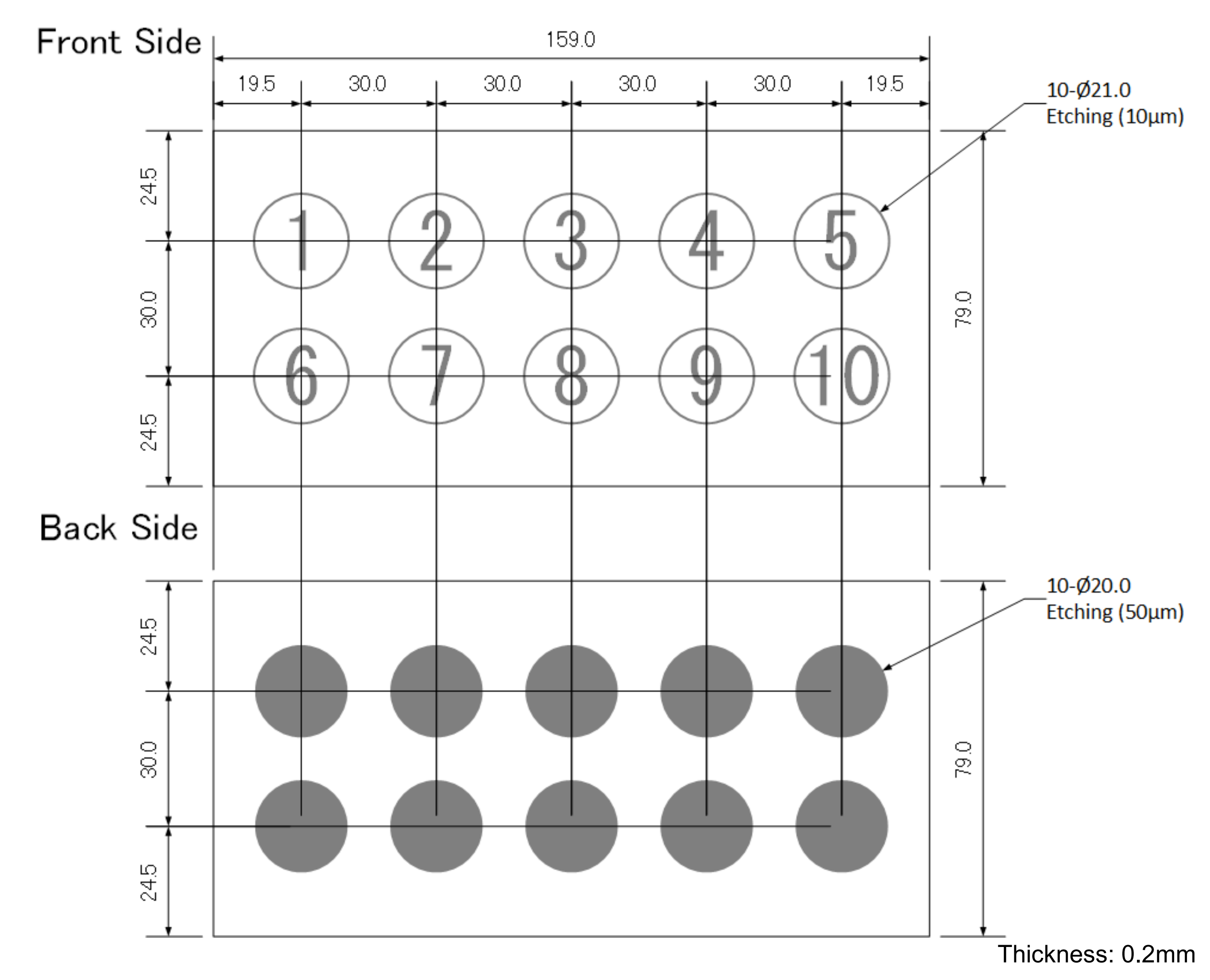


Figure 6-11 Dimension of metal panel layer (unit: mm, except where noted)

6.2.3 Spacer Layer

The spacer layer is used to fix the distance between the metal panel layer and the board layer. A non-conductive sheet is generally used as a spacer. Make holes according to the location of the sensors. For the spacer, use a robust material that does not deform when the metal panel layer is pressed. Optical-clear adhesive (OCA) film, glass epoxy sheet (FR4), and acrylic resin are recommended for the spacer material.

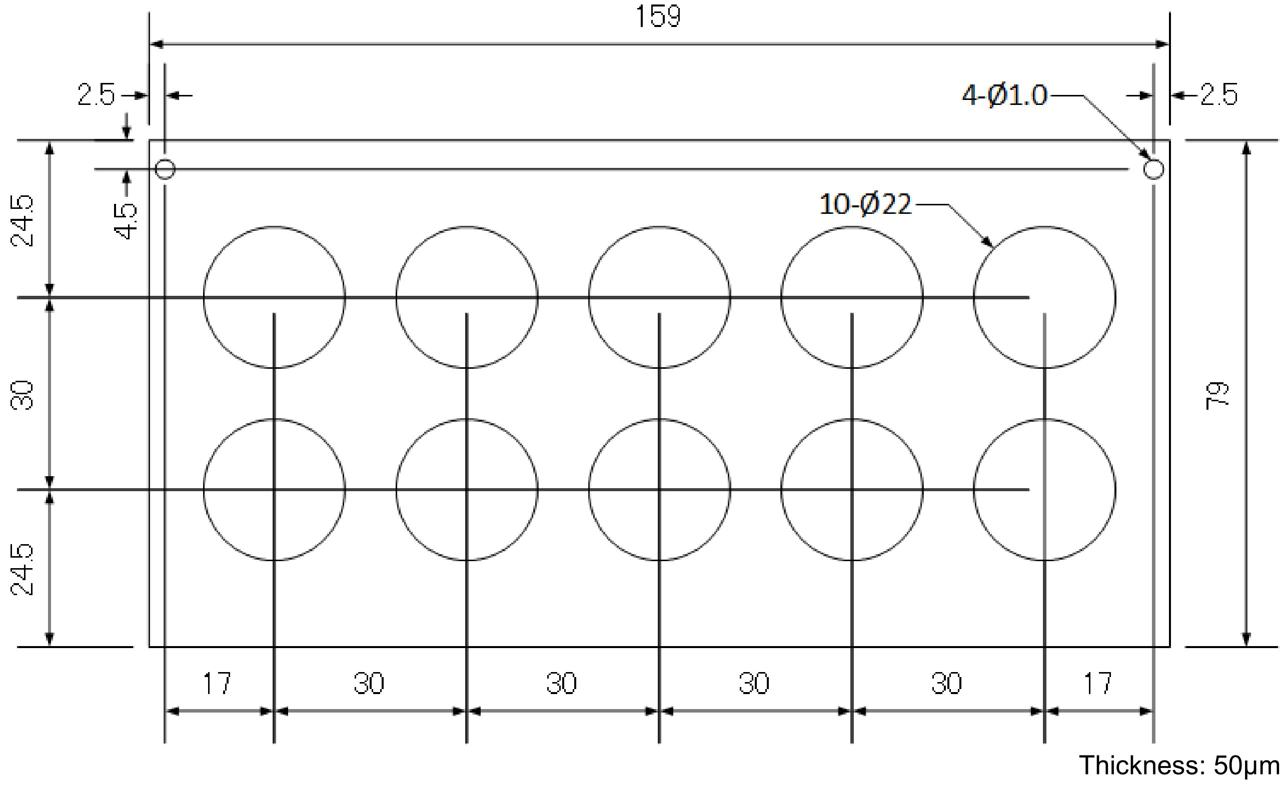


Figure 6-12 Dimension of spacer layer (unit: mm)

6.2.4 Board Layer

The board layer is a board that has electrodes formed by the pattern, forming a capacitor between it and the metal panel layer.

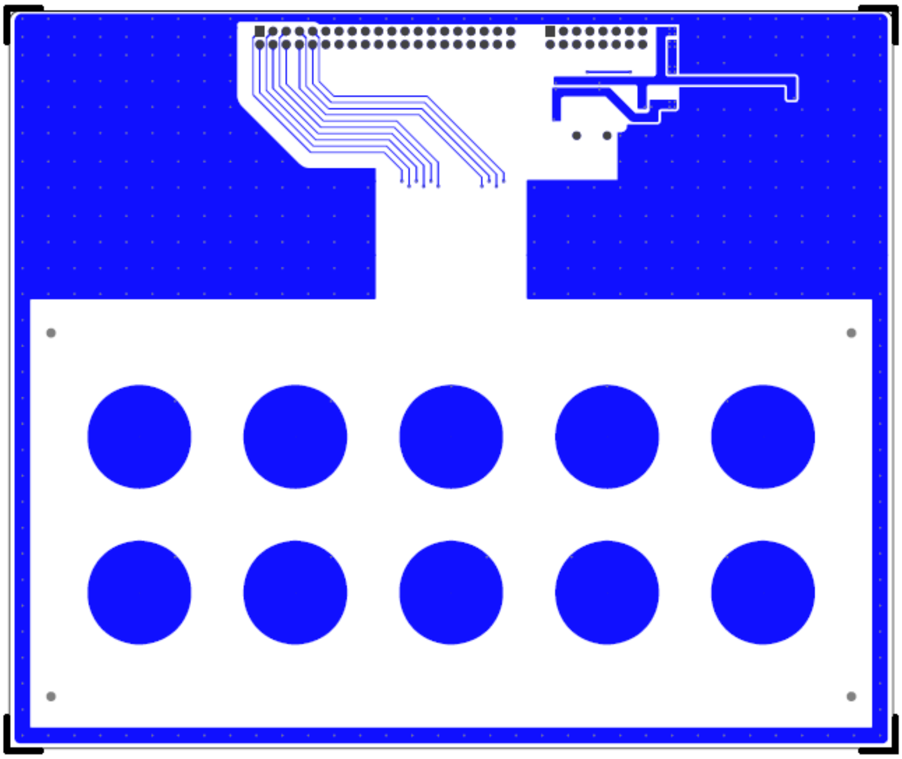


Figure 6-13 Board layer

6.2.5 Sensor Design

This section describes precautions on designing the metal panel touch sensor. Figure 6-14 shows the configuration of the metal panel touch sensor.

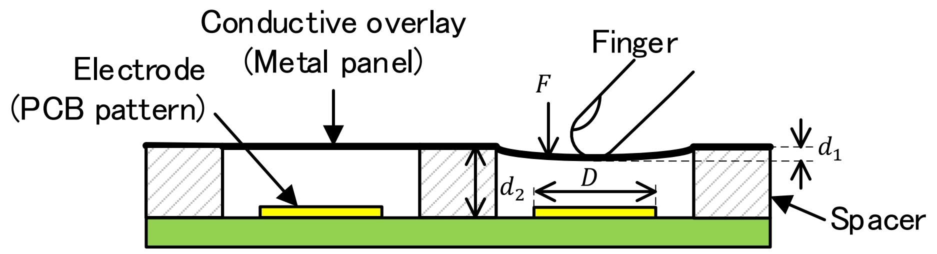


Figure 6-14 Configuration of metal panel touch sensor

F: Force pressing the conductive overlay [N]

D: Diameter of the electrode [m]

d_1: Deflection of the conductive overlay due to stress [m]

d_2: Height of the spacer layer [m]

When designing, consider the minimum of the pressing force F that you wish to detect and the maximum that does not cause irreversible deformation as constraints and select materials for the conductive overlay and the spacer. The deflection d_1 over the force F is determined by the material of the conductive overlay. Next, determine the electrode diameter D and the spacer layer height d_2 so that the change in capacitance will be appropriate to the force.

(1) Selection of Conductive Overlay

The metal panel should be soft enough to be deflected when pressed by a user. At the same time, it should be hard enough not to deform irreversibly when pressed. As an index of softness, there is a characteristic of Young's modulus, which is a proportionality constant for strain and stress. As an index of strength, there is a characteristic of yield strength. Table 6-1 shows Young's modulus and yield strength of general materials.

The metal materials used most commonly in consumer and industrial fields are stainless steel and aluminum. Under the sufficient yield strength, when the Young's modulus is larger, the deflection of the material due to the pressing force and the change in capacitance are larger.

Table 6-1 Young's modulus and yield strength of metal materials (reference value)

Material	Young's modulus [GPa]	Yield strength [MPa]
Stainless steel (SUS303, SUS304)	193	210
Aluminum (A5052)	70	215
Brass (C2600)	110	130
Titanium (TP340)	106	215

(2) Selection of Spacer

The distance between the metal panel and the board layer is determined by the spacer thickness d_2. This determines the capacitance $C_{s_{def}}$[F] with no force applied. $C_{s_{def}}$ is calculated with the following equation and parameters in Figure 6-14.

$$C_{s_{def}} = \varepsilon_r \varepsilon_0 \frac{\pi D^2}{4d_2} \quad (1)$$

ε_r: Relative permittivity of the material between the electrodes (assuming air, $\varepsilon_r \cong 1$)

ε_0: Electric constant ($\varepsilon_0 \approx 8.854 \times 10^{-12}$) [F/m]

From the equation (1), the relation between the spacer thickness d_2 and the capacitance $C_{s_{def}}$ with no force applied is shown in Figure 6-15 (reference value), where the relative permittivity $\varepsilon_r \cong 1$, and the circular electrode diameter $D = 20$ [mm]. Figure 6-15 shows that the change in capacitance is larger when the thickness of the spacer is smaller. Therefore, the thickness of the spacer should be small to get high sensitivity. However, as it is more difficult to set the threshold for judging strength if the change is too large, the thickness of the spacer should not be too small when the function of judging strength is required.

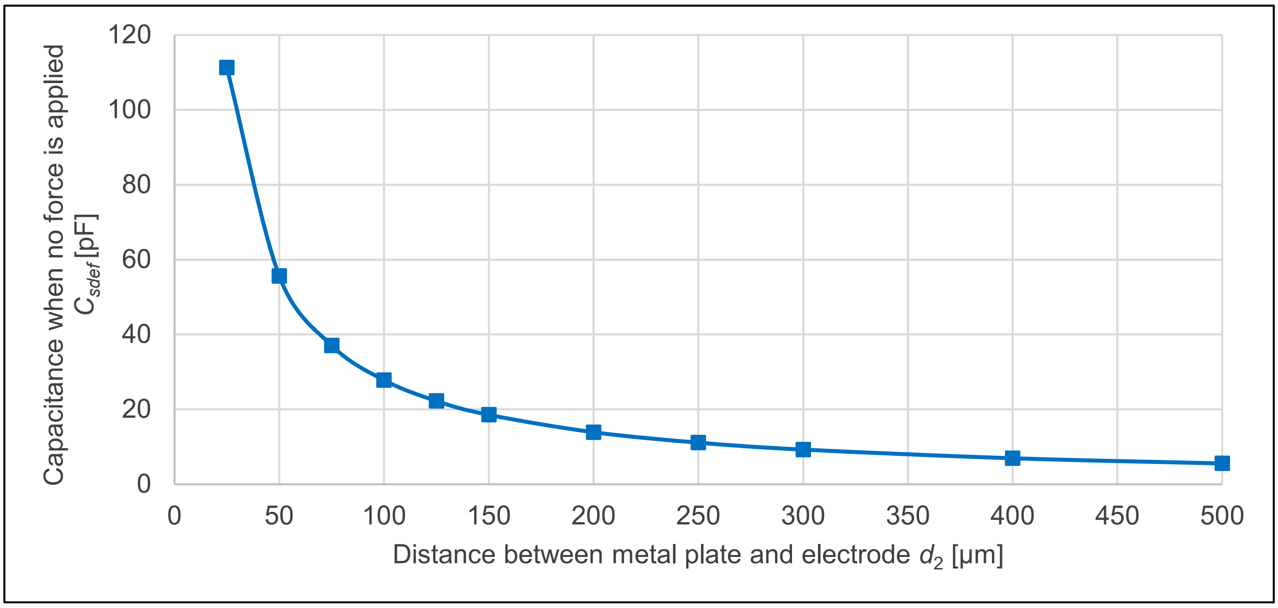


Figure 6-15 Relation between spacer thickness d_2 and capacitance $C_{s_{def}}$ (reference value)

The following equation expresses the capacitance change percentage ΔC_s[%] when the button section is pressed.

$$\Delta C_s = \frac{C_{s_{push}} - C_{s_{def}}}{C_{s_{def}}} \times 100 \tag{2}$$

The capacitance $C_{s_{push}}$ [F] when the button section is pressed is calculated by the following equation.

$$C_{s_{push}} = \varepsilon_r \varepsilon_0 \frac{\pi D^2}{4(d_2 - d_1)} \tag{3}$$

From the equation (2), the relation between the deflection d_1 with the button section pressed and the capacitance change percentage ΔC_s is shown in Figure 6-16 (reference value), where the relative permittivity ε_r ≅1, the circular electrode diameter $D=20$[mm], and the spacer thickness d_2 = 0.05mm.

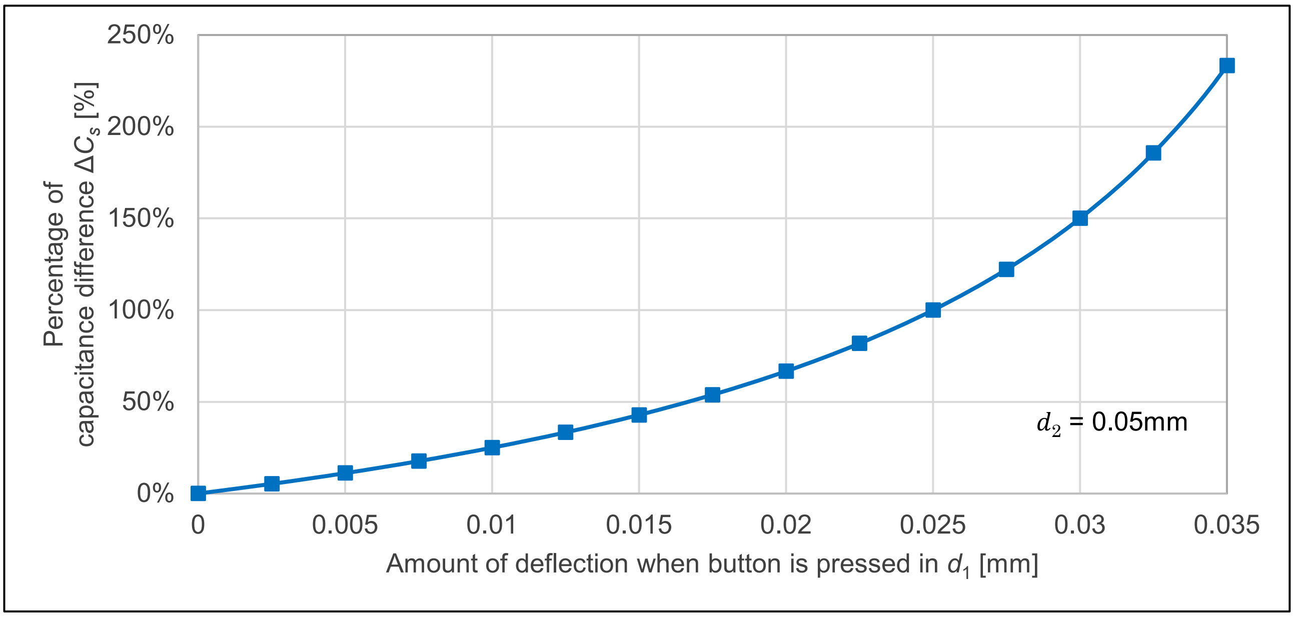


Figure 6-16 Relation between capacitance change percentage ΔC_s and the deflection d_1 of metal panel due to stress (reference value)

6.2.6 How to Assemble

Stack the metal panel layer, spacer layer, and board layer and fix them to each other. To fix each layer, adhesives are generally used. The adhesives should be strong and have no elasticity. Inappropriate assembly may cause a crosstalk between buttons or a detection error when any place other than buttons is pressed. The following describes precautions on design for appropriate assembly.

(1) If the rigidity of the adhesive or spacer layer is not sufficient, or if the layers are not glued evenly

If the rigidity of the adhesive or the spacer layer is not sufficient compared to the metal panel, or if the layers are not glued evenly, the area other than the sensor deforms when the sensing part is pressed. This may cause a detection error. To avoid this, select appropriate materials for the metal panel, adhesive, and spacer layer. In addition, to glue them evenly, it is recommended that you glue the flat area on the board considering the thickness of the board pattern.

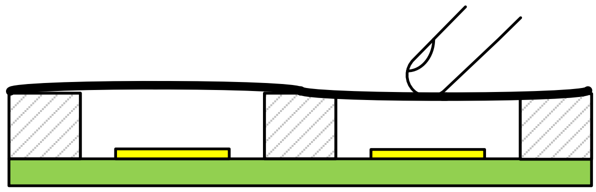


Figure 6-17 If the rigidity of the adhesive or spacer layer is not sufficient, or if the layers are not glued evenly

(2) If the rigidity of the mechanism is not sufficient

If the rigidity of the mechanism is not sufficient, the entire mechanism deflects due to a press. As a result, the area other than the sensor deforms when the sensing part is pressed, which may cause a detection error. To avoid this, it is recommended that you increase the thickness of the board layer or attach a reinforcement on the solder side to increase the stiffness of the board. To avoid an increase in capacitance, use non-conductive materials such as glass epoxy, acrylic resin, etc. as reinforcement.

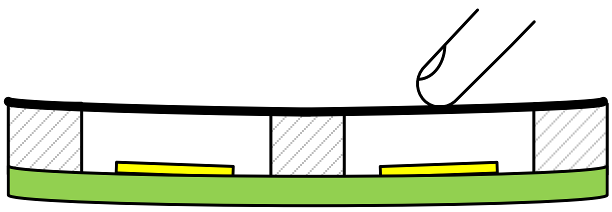


Figure 6-18 If the rigidity of the mechanism is not sufficient

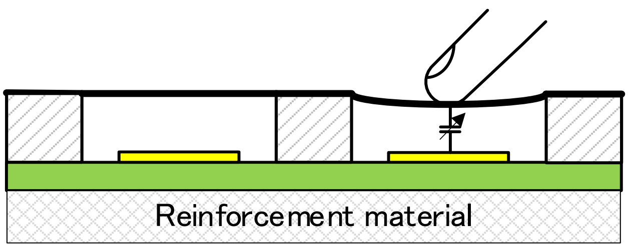


Figure 6-19 Adding reinforcement

7. Sample Program

7.1 Operation Overview

Figure 7-1 shows a flowchart of processing in this sample program.

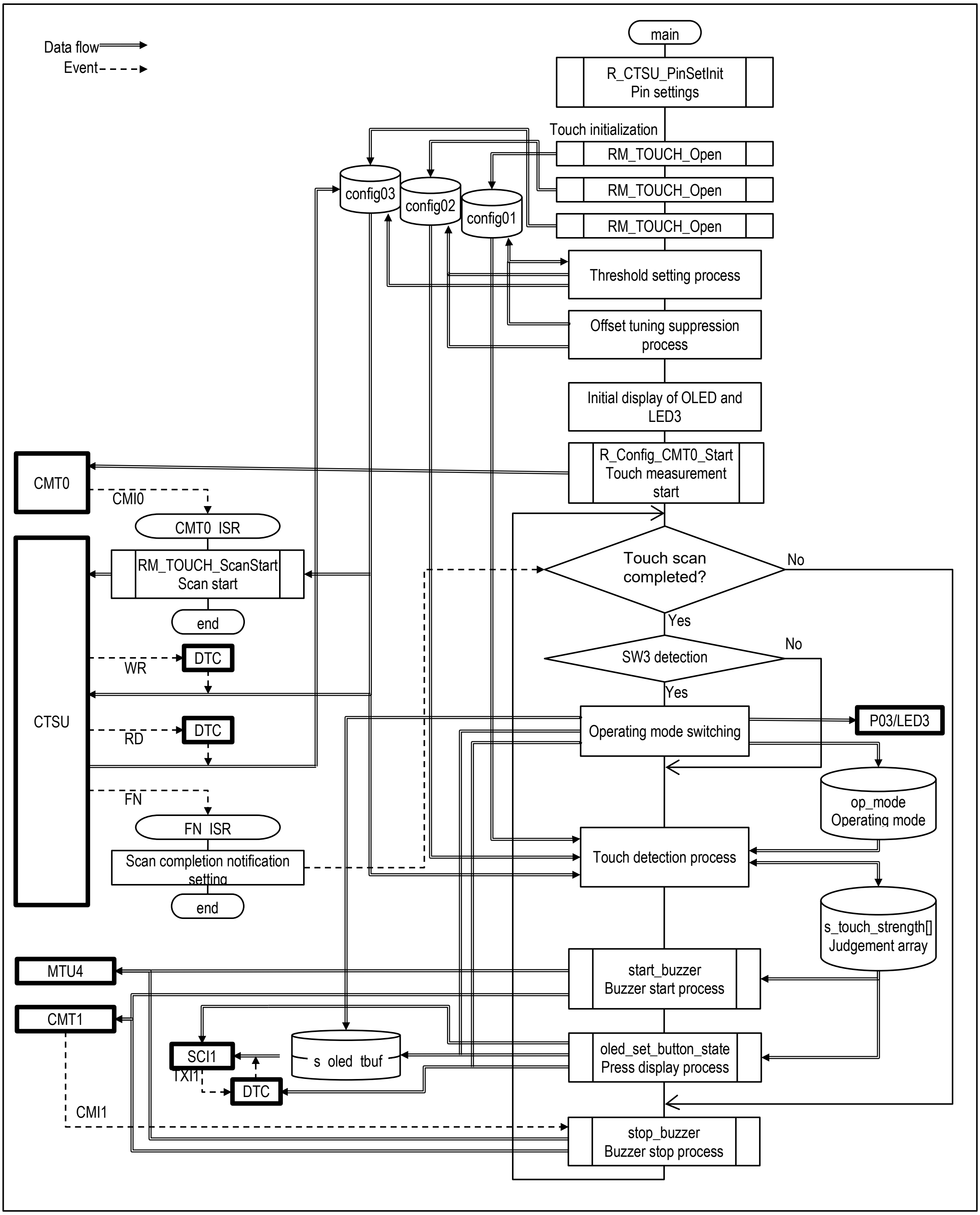


Figure 7-1 Overview of processing flow

The following describes an overview of each process in Figure 7-1. Figure 7-2 shows the timing chart of the touch measurement process and operation of peripherals.

Initial Settings

Configure initial settings for each function:

- Configure initial settings of config01, 02, and 03 that are created in “7.2.1.2 Settings in QE for Capacitive Touch” using RM_TOUCH_Open.
- Set the thresholds for touch detection for config01, 02, and 03 based on the threshold definition in “Table 7-19 usr_config.h definitions”.
  For detail, refer to “7.3.1 Settings of Touch Judgement Threshold”.
- Set the offset tuning suppression for config01 and 02.
- Set the initial display of Pmod_OLDrgb.
- Start the Timer of touch measurement cycle.

SW3 Detection

The state of SW3 is determined by three consecutive matching of the pin state, and the operating mode is switched when the pressed state is detected. Then LED3 and PMOD OLEDrgb display will be updated accordingly.

Touch Detection Process

If detecting a touch scan completion, get the result of touch judgement according to the operating mode.

- Operating mode 1: Judge a press of each button based on the measurement result of config03.
- Operating mode 2: Judge the pressing strength of each button based on the measurement result of config03 under the judgement conditions of config01, 02, and 03.

For details on press judgement in each operating mode, refer to the 7.3.1 Touch Detection Process

Starting/Stopping Buzzer Process

In the buzzer start process, set the musical note assigned to the press-detected button to MTU4 and sound the buzzer. If detecting multiple buttons being pressed at the same time, sound the buzzer corresponding to the button with the largest number. In addition, if detecting next another button being pressed while buzzing, sound another buzzer corresponding to that new button. When setting a buzzer, the timer CMT1 specifying the sounding time starts counting.

In the buzzer stop process, when detecting over time of the sounding time by compare-match CMI1 of CMT1 and stop MTU4 and CMT1.

For details, refer to “7.4 Starting/Stopping Buzzer Process”.

Press Display Process

Display the pressing states for each button in the judgement array s_data on Pmod OLEDrgb.

For details, refer to “7.5 Display Process”.

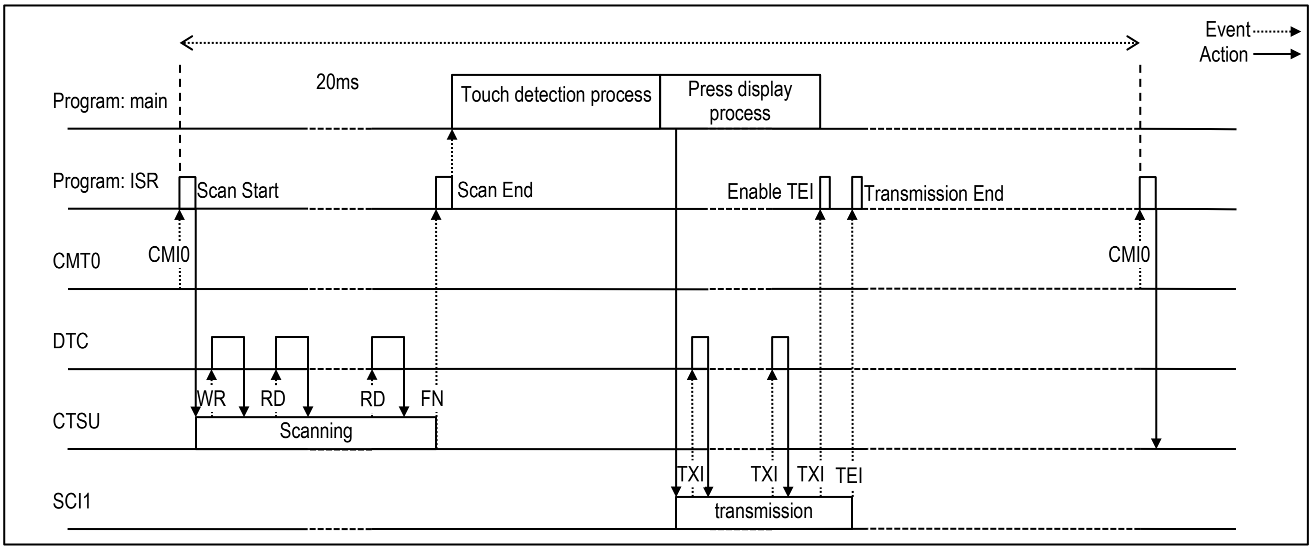


Figure 7-2 Peripherals timing chart

7.2 Functions and Settings of MCU Used

Table 7-1 lists the peripheral functions used in this example, Table 7-2 lists the pins used, and Table 7-3 lists the clock settings. Unused pins are set to output Low.

The settings of the peripheral functions are generated by using the FIT module or the code generation function of Smart Configurator. The settings for each peripheral function are described below.

Table 7-1 Peripheral functions used

Peripheral function	Usage
CTSU	Capacitive touch detection
CMT0	Cycle of capacitive touch detection
DTC	CTSU control Communication to the display with SCI1
SCI1	Display control
PORT	Control of display, SW3, and LED3
MTU4	Sounding a buzzer
CMT1	Buzzer control

Table 7-2 Pins used

No.	Pin name	I/O	Usage
2	P03	O	LED3
14	P35	I	Detection of SW2 (not used in this program)
15	P34	I	Detection of SW3 being pressed
16	TS0	-	Measurement of the electrode Button02 on the electrode board
17	TS1	-	Measurement of the electrode Button07 on the electrode board
18	TS2	-	Measurement of the electrode Button01 on the electrode board
19	TS3	-	Measurement of the electrode Button03 on the electrode board
20	TS4	-	Measurement of the electrode Button06 on the electrode board
21	P21	-	PMOD pin 8, Pmod OLEDrgb RES#
22	P20	I	PMOD pin 2
23	SCK1	O	PMOD pin 4, Pmod OLEDrgb SPI clock output.
24	SMOSI1	O	PMOD pin 3, Pmod OLEDrgb SPI data output
25	TS5	-	Measurement of the electrode Button08 on the electrode board
26	TS6	-	Measurement of the electrode Button04 on the electrode board
27	P13	O	PMOD pin 1, Pmod OLEDrgb CS#
28	P12	O	PMOD pin 7, Pmod OLEDrgb D/C#
29	TS7	-	Measurement of the electrode Button09 on the electrode board
30	TS8	-	Measurement of the electrode Button05 on the electrode board
31	TS9	-	Measurement of the electrode Button10 on the electrode board
38	TSCAP	-	CTSU TSCAP pin
62	MTIOC4C	O	Buzzer of the electrode board
65	PD1	I	USB serial IC (not used in this program)
66	PD0	O	
78	P07	O	PMOD pin 9, PmodOLEDrgb VCCEN
80	P05	O	PMOD pin 10, PmodOLEDrgb PMODEN

Note: Unused pins are set to output 0.

Table 7-3 Clock settings

Item	Setting
Clock used	HOCO clock (48MHz)
SCKCR (FCLK)	x1 (48MHz)
SCKCR (ICLK)	x1 (48MHz)
SCKCR (PCLKB)	x1/2 (24MHz)
SCKCR (PCLKD)	x1 (48MHz)

7.2.1 Detection of Capacitive Touch

For detection of capacitive touch, the following modules are used.

- FIT QE Touch module: rm_touch_qe
- FIT QE CTSU module: r_ctsu_qe
- FIT DTC module: r_dtc_rx

CMT0 is used for the cycle of touch detection.

The following shows the settings of the modules and the peripheral functions, and the settings in QE for Capacitive Touch.

7.2.1.1 Settings of Modules Used

The following shows the settings of each module in Smart Configurator.

Table 7-4 FIT QE Touch module rm_touch_qe settings

Item		Setting
Configuration		
	Parameter check	Use system default
	Support for QE monitoring using UART	Sensor monitor not used
	Support for QE monitoring using UART	Serial tuning not used
	Type of chattering suppression	TypeA: Counter if exceeded threshold is held within hysteresis range

Table 7-5 QE CTSU module r_ctsu_qe settings

Item		Setting
Configuration		
	Parameter check	Use system default
	Data transfer of INTCTSUWR and INTCTSURD	DTC
	Select automatic judgement code	Disable
	Interrupt level for INTCTSUWR	Level 2
	Interrupt level for INTCTSURD	Level 2
	Interrupt level for INTCTSUFN	Level 2
Resource: CTSU		
	TSCAP pin	Used
	TS0 pin	Used
	TS1 pin	
	TS2 pin	
	TS3 pin	
	TS4 pin	
	TS5 pin	
	TS6 pin	
	TS7 pin	
	TS8 pin	
	TS9 pin	
	Other TS pins	Not used

Table 7-6 FIT DTC module r_dtc_rx settings

Item		Setting
Configuration		
	Parameter check	Use system default
	DTCER control	Clear all DTCER registers in R_DTC_Open()
	Address mode	Full address mode
	Transfer data read skip	Enable transfer data read skip
	DMAC FIT check	DMAC FIT module is not used with DTC FIT module
	Sequence transfer	Sequence transfer not used

Table 7-7 CMT0 settings

Item		Setting
Count clock setting		PCLK/32
Compare match setting	Interval value	20ms
	Compare match interrupt (CMI0)	Enable
	Enable multiple interrupts (CMI0)	Disable
	Priority	Level 15 (highest)

7.2.1.2 Settings in QE for Capacitive Touch

After configuring the MCU operating conditions, the peripheral functions, and the pins shown in Table 7-1 to Table 7-3 with Smart Configurator, follow the workflow of QE for Capacitive Touch to set touch measurement.

(1) Touch Interface Configuration

Create a touch interface configuration according to the electrode board. In this example, the electrodes are structured as shown in Figure 7-3, and three types of configurations (methods) are created for the single electrode group.

The config03 is used for touch measurement and used for press judgement in the operating mode 1. The config01 and the config02 are used together with config03 for judgement of the pressing strength in the operating mode 2.

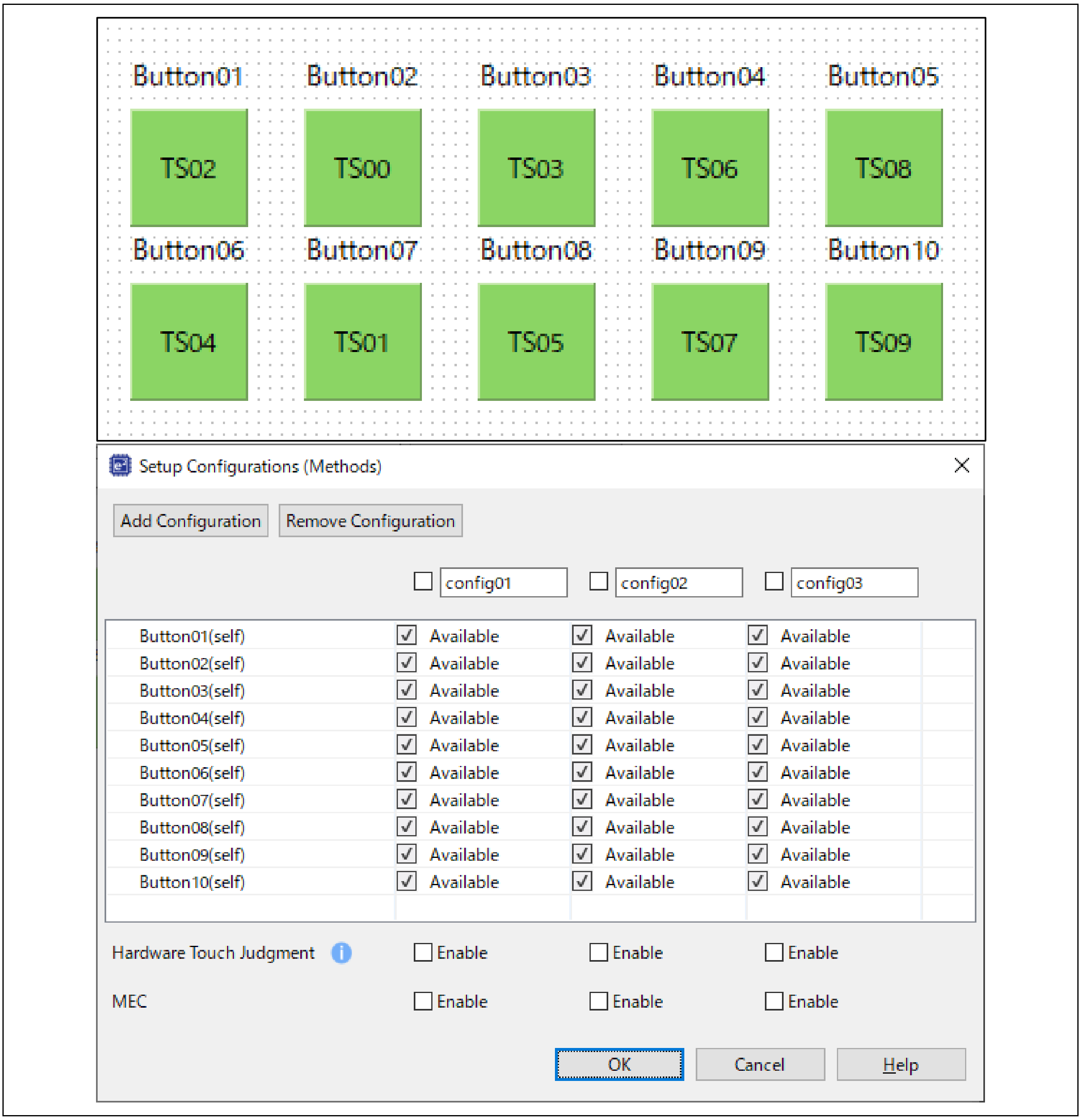


Figure 7-3 Electrode configuration of touch interface configuration

(2) Adjustment of Touch Sensor

Table 7-8 shows assignment of pressing strength to each method. The touch sensors are adjusted by the strength assigned to config01. As the adjustment of the touch sensors corresponding to config02 and config03 is reconfigured by the program, it does not depend on the pressing strength.

After adjusting the touch sensors, output a file of the adjustment result to the folder “qe_gen” in “Table 7-17 Source file configuration”, and rebuild the project.

Table 7-8 Assignment of pressing strength to methods in operating mode 2

Method	Pressing strength
config01	High
config02	Middle
config03	Low

7.2.2 Control of Display

To control Pmod OLEDrgb, SCI1 is used for the master transmission only in the SPI clock synchronous mode. PORT0,1,2 are used for pin control, and the FIT DTC module is used for data transfer setting of SCI1.

Table 7-9 shows the SCI settings, and Table 7-10 shows the PORT settings based on “Table 7-2 Pins used”. For the FIT DTC module settings, refer to Table 7-6.

Table 7-9 SCI1 settings

SPI clock synchronous mode
Master transmission

Item		Setting
Transfer direction setting		MSB-first
Data inversion setting		Normal
Transfer speed setting	Transfer clock	Internal clock
	Bit rate	6000 kbps
	Enable modulation duty correction	Disable
Clock setting		Not used
Data handling setting	Transfer data handling	Data handled by DTC
Interrupt setting	TXI1, TEI1 priority	Level 15 (highest)
Callback function setting		Not used

Table 7-10 PORT settings for Pmod OLEDrgb control

Item	Setting					
Port selection	PORT0		PORT1		PORT2	
Used port	P05	P07	P12	P13	P20	P21
	Out		Out	Out Output 1	In Pull-up	Out Output 1

7.2.3 Sounding a Buzzer

To sound a buzzer, MTU4 is used in the PWM mode 1 to output PWM waveforms from PE1. CMT1 is used to set the sounding time.

Table 7-11, Table 7-12, and Table 7-13 show the MTU4 settings, musical scale settings, and CMT1 time settings, respectively.

Table 7-11 PWM mode timer 1 (MTU4) settings

Item		Setting
Synchronous mode setting		Not used
TCNT4 counter setting	Counter clear source	TGR4 compare match (Use TGRC4 as a cycle register)
	Counter clock Selection	PCLK
General register setting	TGRC4	Output compare register
	TGRD4	Output compare register
Output setting	MTIOC4A pin	Output disabled
	MTIOC4C pin	Output initial 0, toggle at compare match
	When TGRD compare match	Toggle output from MTIOC4C pin
PWM output setting	PWM period	Refer to Table 7-12 Apply musical note C5 as initial value
	TGRA initial value	
	TGRB initial value	
	TGRC initial value	
	TGRD initial value	
A/D conversion start trigger setting		Not used
Interrupt setting		Not used

Table 7-12 MTU4 musical scale setting

index	Musical scale	Frequency [Hz]	Setting		
			PWM period [ms]	TGRA, TGRC	TGRB, TGRD
0	A4	880	1.136363636	27272	13636
1	B4	987.7666025	1.012384907	24297	12148
2	C5	1046.502261	0.955564108	22933	11466
3	D5	1174.659072	0.851310839	20431	10215
4	E5	1318.510228	0.758431735	18202	9101
5	F5	1396.912926	0.715864233	17180	8590
6	G5	1567.981744	0.637762527	15306	7653
7	A5	1760	0.568181818	13636	6818
8	B5	1975.533205	0.506192453	12148	6074
9	C6	2093.004522	0.477782054	11466	5733

Table 7-13 CMT1 settings

Item		Setting
Count clock setting		PCLK/512
Compare match setting	Interval value	100ms
	Compare match interrupt (CMI1)	Enable
	Enable multiple interrupts (CMI1)	Disable
	Priority	Level 0 (disabled)

7.2.4 SW3 and LED3

To switch and display the operating modes, SW3 and LED3 are used. Table 7-14 shows the PORT settings based on “Table 7-2 Pins used”.

Table 7-14 Port settings of SW3 and LED3

Item	Setting	
Port selection	PORT0	PORT3
Used port	P03	P34
	Out Output 1	In

7.2.5 Settings of Other Pins

Table 7-15 shows the PORT settings of SW2 and USB serial IC that are not used in this program.

Unused pins are set to output Low.

Table 7-15 PORT settings of SW2 and USB serial IC connection

Item	Setting		
Port selection	PORTD		PORT3
Used port	PD0	PD1	P35
	Out CMOS output Output 1	In	In

7.3 Touch Detection Process

A touch is measured with config03. When detecting measurement completion, perform the touch detection process. The touch judgement depends on the threshold settings.

7.3.1 Settings of Touch Judgement Threshold

The thresholds for touch judgement of each pressing strength are set in the initialization process. From the result of adjusting touch sensors of config01 with QE for Capacitive Touch, set the thresholds for each button based on the percentage of touch judgement threshold in “Table 7-19 usr_config.h definitions” as follows.

(1) Calculation of the dynamic range of change in the measured value and the hysteresis ratio

In touch sensor adjustment, the threshold for touch judgement is set to 60% of the change in the measured value during adjustment. The hysteresis is set to 5% lower than the touch judgement threshold.

For config01, the touch judgement threshold and hysteresis are stored as follows.

```
g_qe_touch_instance_config01.p_ctrl->binfo.p_threshold[]  // touch threshold
g_qe_touch_instance_config01.p_ctrl->binfo.p_husteresis[] // hysteresis
```

From the above, the dynamic range of the measurement value is defined by the following formula, where the button number is n (0 to 9). In addition, scaling is performed by 2^{SCALE} for integer calculation. SCALE is set to 9 to fit the calculation result into 32 bits.

$$\mathrm{DR[n]} = \frac{\mathrm{p\_threshold[n]}}{60\%}$$

$$\mathrm{DR[n]} \cdot 2^{\mathrm{SCALE}} = \frac{100 \cdot \mathrm{p\_threshold[n]} \cdot 2^{\mathrm{SCALE}}}{(100 \cdot 60\%)}$$

For (3) Calculation of the hysteresis, retrieve the touch judgement threshold and hysteresis of the first button of config01.

```
thr = g_qe_touch_instance_config01.p_ctrl->binfo.p_threshold[0];
hys = g_qe_touch_instance_config01.p_ctrl->binfo.p_hysteresis[0];
```

(2) Calculation of the touch judgement threshold

For the calculated dynamic ranges of change in the measured value for each button, calculate the touch thresholds for each config as below, based on the touch judgement threshold rate in “Table 7-19 usr_config.h definitions”. ‘m’ represents the number of config.

$$\begin{aligned}\mathrm{config_{0m}.P\_threshold[n]} &= \mathrm{DR[n]} \cdot \mathrm{threshold\ rate\ [m]} \\ &= \left(\frac{(\mathrm{DR[n]} \cdot 2^{\mathrm{SCALE}}) \cdot \mathrm{D\_CFG\_TOUCH\_THRESHOLD_m}}{100}\right) 2^{-\mathrm{SCALE}}\end{aligned}$$

(3) Calculation of the hysteresis

From the touch judgement threshold calculated in (2), calculate the hysteresis as below.

$$\begin{aligned}\mathrm{config_{0m}.p\_hysteresis[n]} &= \mathrm{config_{0m}.p\_threshold[n]} \frac{\mathrm{hys}}{\mathrm{thr}} \\ &= \frac{\mathrm{config_{0m}.p\_threshold[n]} \cdot \mathrm{hys}}{\mathrm{thr}}\end{aligned}$$

7.3.2 Touch Detection Process

The touch detection process differs depending on the operating mode.

Operating mode 1: Judge a press

From the measurement result of config03, judge a press on each button. Store 1 if it is pressed, and store 0 if not pressed in the press judgement result.

Operating mode 2: Judge the pressing strength

Judge the pressing strength of each button based on the measurement result of config03 under the judgement conditions of config03, 02, and 01. Store the judgement in the press judgement result. For assignment of the pressing strength for each config, refer to Table 7-8.

For touch judgement, the same baseline is used, so the baseline of config03 is applied to config01, 02 after the process.

Figure 7-4 shows a flowchart of touch detection.

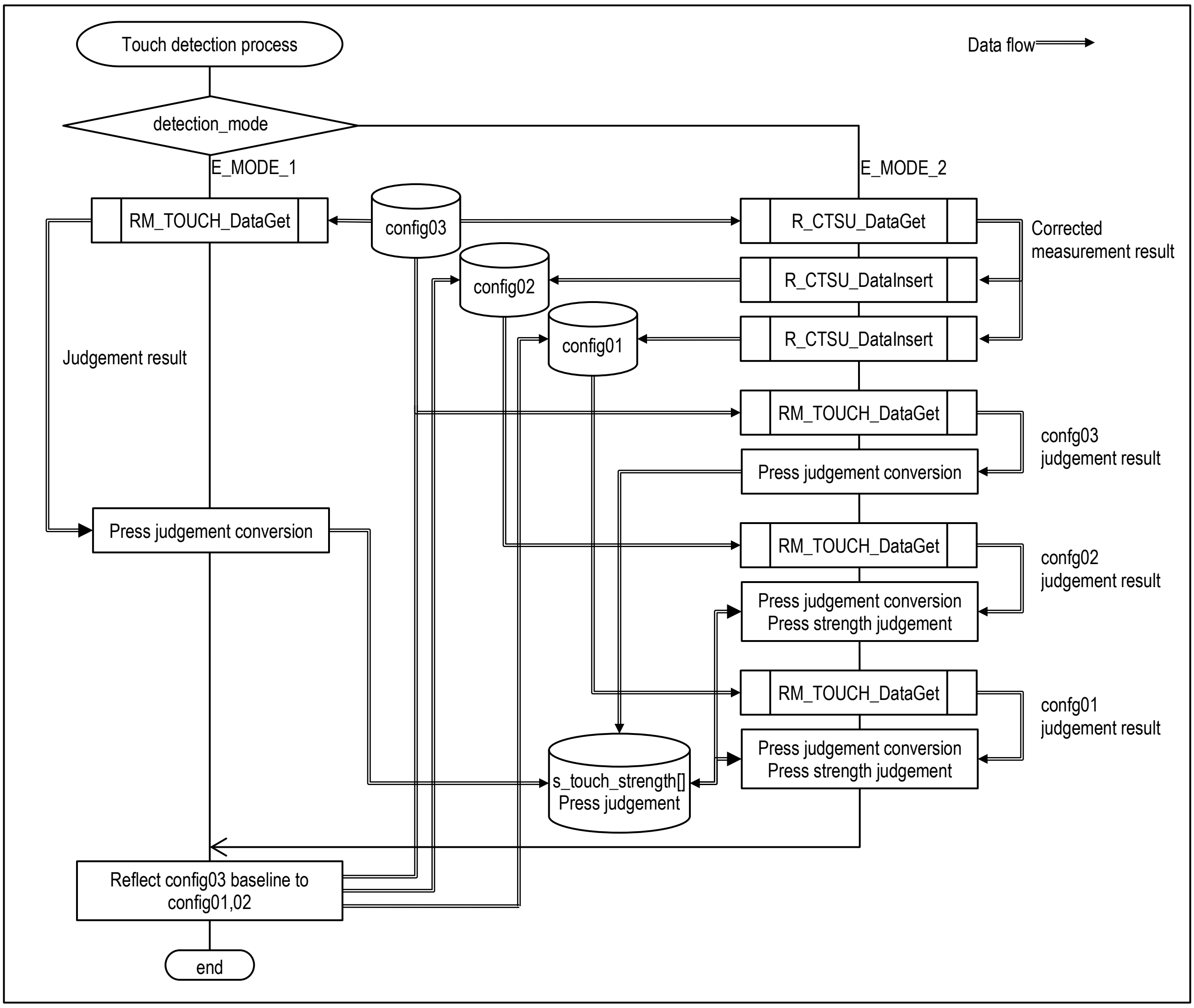


Figure 7-4 Flowchart of touch detection

7.4 Starting/Stopping Buzzer Process

When a press is detected, sound a buzzer with the musical note assigned to the detected button, and stop the buzzer after the preset sounding time.

If detecting multiple buttons being pressed at the same time, sound a buzzer with the musical note for the largest number button. In addition, if detecting next another button being pressed while buzzing, sound another buzzer corresponding to that new button. When setting a buzzer, clear CMT1 and start counting.

Stop the buzzer when CMT1 compare match is detected, which has been set in setting of the buzzer.

Figure 7-5 shows a flowchart of starting and stopping a buzzer.

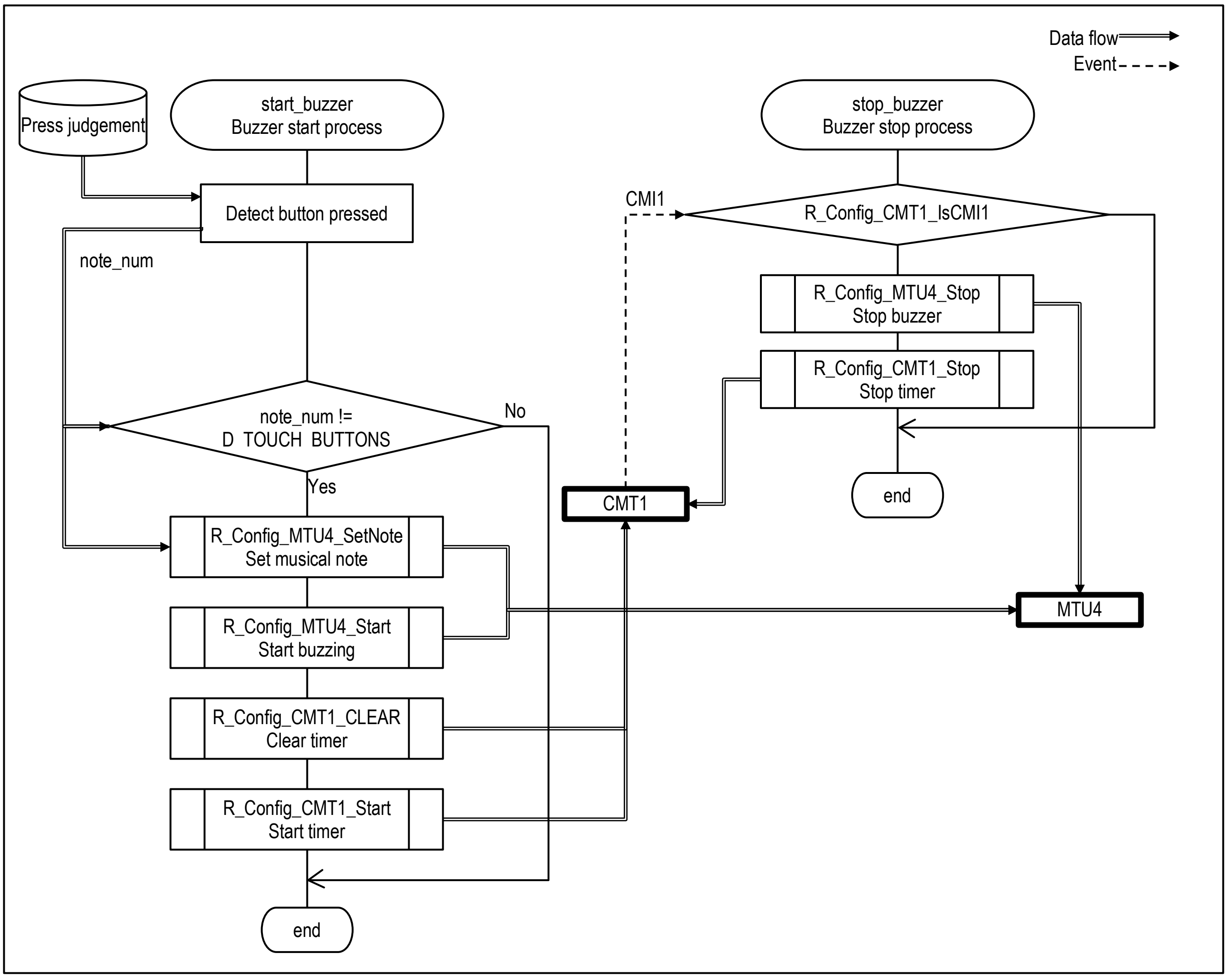


Figure 7-5 Flowchart of starting/stopping a buzzer

7.5 Display Process

Process the display on Pmod OLEDrgb. For details on control, refer to the following Digilent web page.

Pmod OLEDrgb Reference Manual - Digilent Reference

In this program, characters are displayed in 8 rows and 16 columns with 8x6-dot fonts as shown in Figure 7-6.

Figure 7-7 shows a timing chart of transferring display data of the pressing state, and Table 7-16 shows the initial settings of Pmod OLEDrgb.

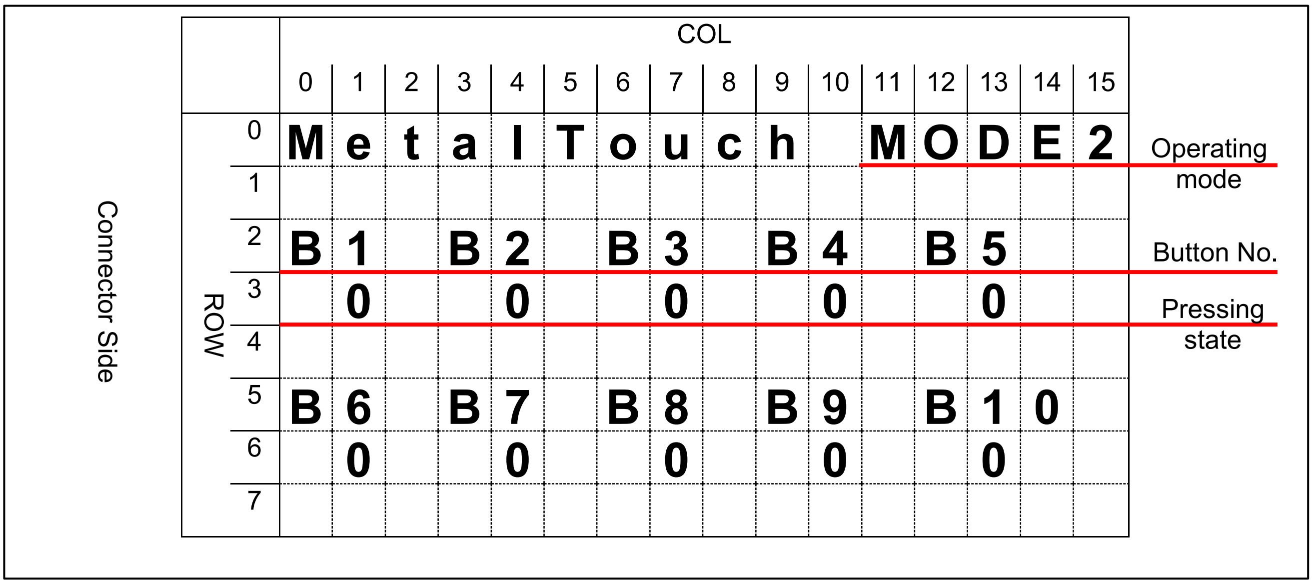


Figure 7-6 Pmod OLEDrgb display layout

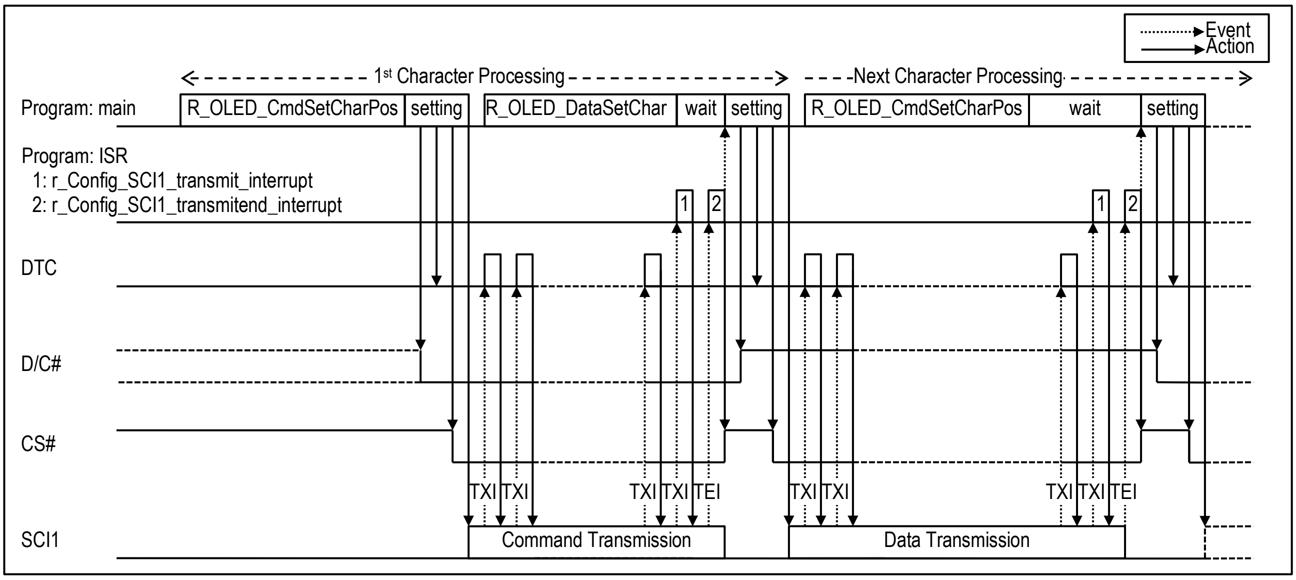


Figure 7-7 Timing chart of transferring display data

Table 7-16 Pmod OLEDrgb initial settings

Item	Setting	Command/Option
Remap & Color Depth setting	[0] Vertical address increment: 1 [1] RAM column 0 to 95 maps to pin seg (SA,SB,SC) 95 to 0 :1 [2] Normal order SA,SB,SC (e.g. RGB): 0 [3] Disable left-right swapping on COM: 0 [4] Scan from COM [N-1] to COM0: 1 [5] Enable COM Split Odd Even: 1 [7:6] 256 color format: 0	A0h, 33h
Display start line by row	0	A1h, 00h
Display offset	0	A2h, 00h
Display mode	Normal display	A4h
Multiplex ratio	62	A8h, 3Eh
Master configuration	External V_{CC} supply	ADh, 8Eh
Power save mode	Disable power save mode	B0h, 0Bh
Phase 1 and 2 period adjustment	[3:0] Phase 1 period in N DCLK: 1 [7:4] Phase 2 period in N DCLK: 3	B1h, 31h
Display clock divider / Oscillator frequency	Define the divide ratio (D) of the display clocks (DCLK) [3:0]: 0 Fosc frequency increases as setting value increases [7:4]: 15	B3h, F0h
Second pre-charge speed for “A”	Second pre-charge speed ranges.	8Ah, 64h
Second Pre-charge Speed for “B”		8Bh, 78h
Second Pre-charge Speed for “C”		8Ch, 64h
Pre-charge level	Pre-charge voltage level [5:1]	BBh, 3Ah
V_{COMH}	0.83 x V_{CC}	BEh, 3Eh
Master current control	Master current attenuation factor [3:0]: 7/16	87h, 06h
Contrast for “A”	Contrast for all color in each segment.	81h, 91h
Contrast for “B”		82h, 50h
Contrast for “C”		83h, 7Dh
Deactivate scrolling	Deactivates the scrolling action.	2Eh
Clear window	Column address of start: 0 Row address of start: 0 Column address of end: 95 Row address of end: 63	25h, 00h, 00h ,5Fh ,3Fh

7.6 Program Configuration

7.6.1 Source File Configuration

Table 7-17 Source file configuration

Folder name, file name	Description
Src	
├ main.c	Main function and subroutine
├ r_oled_api.h	API for Pmod OLEDrgb transfer command/data creation
├ r_oled_api.c	
├ usr_config.h	Setting of the threshold to judge the pressing strength
├ smc_gen	Generated by Smart Configurator
│ ├ Config_CMT0	
│ ├ Config_CMT1	
│ ├ Config_MTU4	
│ ├ Config_PORT	
│ ├ Config_SCI1	
│ ├ general	
│ ├ rm_touch_qe	
│ ├ r_bsp	
│ ├ r_config	
│ ├ r_ctsu_qe	
│ ├ r_dtc_rx	
│ ├ r_pincfg	
qe_gen	Generated by QE for Capacitive Touch
├ qe_touch_config.c	
├ qe_touch_config.h	
└ qe_touch_define.h	

7.6.2 Macro Definitions

Table 7-18 main.c definitions

Definition name	Initial value	Description
Touch setting		
D_TOUCH_BUTTONS	10	Number of buttons
D_TOUCH_STRENGTH_MAX	CTSU_CFG_NUM_SELF_ELEMENTS / D_TOUCH_BUTTONS	Number of levels of pressing strength
D_TOUCH_THRESHOLD_QE	60	Threshold rate for touch judgement in QE [%]
D_TOUCH_HYSTERESIS	5	Hysteresis ratio setting in QE [%]
OLED display setting		
D_DISP_BUFFERBYTES	D_OLED_CHR_WIDTH * D_OLED_CHR_HEIGHT * D_OLED_COLS	Number of bytes of Pmod OLEDrgb transfer data buffer
D_DISP_TITLE_STR	"MetalTouch MODE"	Character strings of title
D_DISP_TITLE_ROW	0	Starting position of title
D_DISP_TITLE_COL	0	
D_DISP_KEYNO_STR	"B1 "	Character strings of button No.
D_DISP_KEYNO_ROW1	2	Row 1 of button No. display
D_DISP_KEYNO_ROW2	5	Row 2 of button No. display
D_DISP_KEYNO_COL	0	Starting column of button No. display
D_DISP_KEYNO_LEN	3	Character length of button No.
D_DISP_MODE_ROW	0	Position of operating mode display
D_DISP_MODE_COL	15	
D_DISP_KEY_ROW1	D_DISP_KEYNO_ROW1 + 1	Row1 of detection result display
D_DISP_KEY_ROW2	D_DISP_KEYNO_ROW2 + 1	Row 2 of detection result display
D_DISP_KEY_COL	1	Starting column of detection result display
D_DISP_KEY_LEN	3	Column length of detection result display

Table 7-19 usr_config.h definitions

Definition name	Initial value	Description
D_CFG_TOUCH_THRESHOLD1	75	Rate of touch judgement threshold for each pressing strength. The number corresponds to config No.
D_CFG_TOUCH_THRESHOLD2	50	
D_CFG_TOUCH_THRESHOLD3	25	
D_CFG_TOUCH_SCAN_INDEX	2	Index of element to be scanned by gp_touch_instance[] that is declared in main.c.

Table 7-20 r_oled_api.h definitions

Definition name	Initial value	Description
D_OLED_COLORS	256	Color depth (fixed)
D_OLED_DOT_WIDTH	96	Number of horizontal dots of Pmod OLEDrgb
D_OLED_DOT_HEIGHT	64	Number of vertical dots of Pmod OLEDrgb
D_OLED_CHR_WIDTH	6	Number of horizontal dots of character
D_OLED_CHR_HEIGHT	8	Number of vertical dots of character
D_OLED_ROWS	D_OLED_DOT_HEIGHT / D_OLED_CHR_HEIGHT	Number of rows to be displayed
D_OLED_COLS	D_OLED_DOT_WIDTH / D_OLED_CHR_WIDTH	Number of columns to be displayed

7.6.3 Structure, Unions, and Enumeration Types

Table 7-21 main.c

Enumeration type name		e_mode_t	
Description		Operating mode	
Member	**Name**	**Value**	**Description**
	E_MODE_1	0	Operating mode 1
	E_MODE_2	1	Operating mode 2

Table 7-22 r_oled_api.h

Union type name		u_oled_rgb_t	
Description		Specifies 8-bit color for character/character string to be displayed	
Member	**Type**	**Name**	**Description**
	uint8_t	color	8-bit color specification
	struct	rgb	RGB specification
	uint8_t:2	r	Red
	uint8_t:3	g	Green
	uint8_t:3	b	Blue

7.6.4 Functions

Table 7-23 main.c

Function name		main		
Description		main function		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	--	
Function name		static oled_init		
Description		Pmod OLEDrgb initial settings		
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	uint8_t *	pbuf	Pointer to the transfer commands storage destination
Return value		uint8_t *	Pointer to the available transfer data storage destination	
Function name		static oled_set_operation_mode		
Description		Displays operating mode on Pmod OLEDrgb		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	e_mode_t	mode	Operating mode
	I/O	uint8_t *	pbuf	Pointer to the transfer commands storage destination
Return value		uint8_t *	Pointer to the available transfer data storage destination	
Function name		static oled_set_button_state		
Description		Displays the pressing state on Pmod OLEDrgb		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	data[]	Pressing state array
	I/O	uint8_t *	pbuf	Pointer to the transfer commands storage destination
Return value		uint8_t *	Pointer to the available transfer data storage destination	
Function name		static start_buzzer		
Description		Sounds a buzzer for the button with press detected		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	data[]	Button pressing strength array
Return value		void	-	
Function name		static stop_buzzer		
Description		Process of stopping buzzer		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		static set_transmissiondata_dtc_txi1		
Description		DTC setting for transmission to OLED		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	const void *	src_addr	Address of the buffer storing the transfer data
	I	uint32_t	count	Number of bytes of transfer data
Return value		void	-	

Table 7-24 r_oled_api

Function name		R_OLED_CmdInit		
Description		Creates a group of commands for PmodOLEDrgb initial settings		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t *	buf	Pointer to the transfer commands storage destination
Return value		int32_t	Number of bytes of the command group stored in buf	
Function name		R_OLED_CmdDisplay		
Description		Creates ON/OFF command for PmodOLEDrgb display		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	on	True: display ON False: display OFF
	I	uint8_t *	buf	Pointer to the transfer commands destination
Return value		int32_t	Number of bytes of the command stored in buf	
Function name		R_OLED_CmdSetCharPos		
Description		Creates command for starting position of character display		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	row	Row specification: 0 to 15
	I	uint8_t	col	Column specification: 0 to 7
	I	uint8_t *	buf	Pointer to the transfer commands destination
Return value		int32_t	Number of bytes of the command stored in buf	
Function name		R_OLED_DataSetChar		
Description		Creates character data to be displayed on PmodOLEDrgb		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	code	ASCII code of the character to be displayed
	I	u_oled_rgb_t	color	Color specification
	I	uint8_t *	buf	Pointer to the transfer data storage destination
Return value		int32_t	Number of bytes of the data stored in buf	
Function name		R_OLED_DataSetStr		
Description		Create character string data to be displayed on PmodOLEDrgb		
Argument	**I/O**	**Type**	**Name**	**Description**
	I	const uint8_t *	code	Pointer to the ASCII character string to be displayed
	I	uint8_t	len	Length of character string (excluding NULL-terminator)
	I	u_oled_rgb_t	color	Color specification
	I	uint8_t *	buf	Pointer to the transfer data storage destination
Return value		int32_t	Number of bytes of the data stored in buf	

Table 7-25 Config_MTU4 user-defined function

Function name	R_Config_MTU4_SetNote			
Description	Sets the musical note for a buzzer			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	uint32_t	note	Index that indicates the musical note (Refer to Table 7-12)
Return value		void	-	

Table 7-26 Config_CMT0_user.c interrupt processing function

Function name		r_Config_CMT0_cmi0_interrupt		
Description		Starts touch scan triggered by CMI0		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	

Table 7-27 Config_CMT1 user-defined function

Function name		R_Config_CMT1_CLEAR		
Description		Clears the count value (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_CMT1_IsCMI1		
Description		Gets CMT1 compare match		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		bool	True: compare match is detected False: not detected	

Table 7-28 Config_PORT user-defined functions (1/2)

Function name		R_Config_PORT_GetSwitchState		
Description		Detects SW3 being pressed		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		bool	True: pressed False: other	
Function name		R_Config_PORT_OLED_SET_CS		
Description		Asserts Pmod OLEDrgb CS# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_PORT_OLED_CLEAR_CS		
Description		Negates Pmod OLEDrgb CS# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_PORT_OLED_SET_DC_DATA		
Description		Selects data transfer of Pmod OLEDrgb D/C# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	

Table 7-29 Config_PORT user-defined functions (2/2)

Function name		R_Config_PORT_OLED_SET_DC_CMD		
Description		Selects command transfer of Pmod OLEDrgb D/C# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_PORT_OLED_SET_RES		
Description		Asserts Pmod OLEDrgb RST# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_PORT_OLED_CLEAR_RES		
Description		Negates Pmod OLEDrgb RST# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_PORT_OLED_SET_VCCEN		
Description		Asserts Pmod OLEDrgb VCCN (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_Config_PORT_OLED_SET_PMODEN		
Description		Asserts Pmod OLEDrgb PMODEN (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		R_PORT_OLED_CS		
Description		Gets the states of Pmod OLEDrgb CS# (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		bool	False: assert True: negate	
Function name		R_Config_PORT_SET_LED3		
Description		Sets LED3 (macro function)		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	int32_t	a	0: OFF 1: ON
Return value		void	-	

Table 7-30 Config_SCI1_user.c interrupt processing functions

Function name		r_Config_SCI1_transmit_interrupt		
Description		Triggered by TXI1, disables TXI1 interrupt, and enables TEI1 interrupt		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	
Function name		r_Config_SCI1_transmitend_interrupt		
Description		Triggered by TEI1, negates Pmod OLEDogb CS#, disables TEI1 interrupt, and sets transfer stop		
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value		void	-	

8. Importing a Project

After importing the sample project, make sure to confirm build and debugger setting.

8.1 Importing a Project into e2 studio

Follow the steps below to import your project into e^2 studio. Pictures may be different depending on the version of e^2 studio to be used.

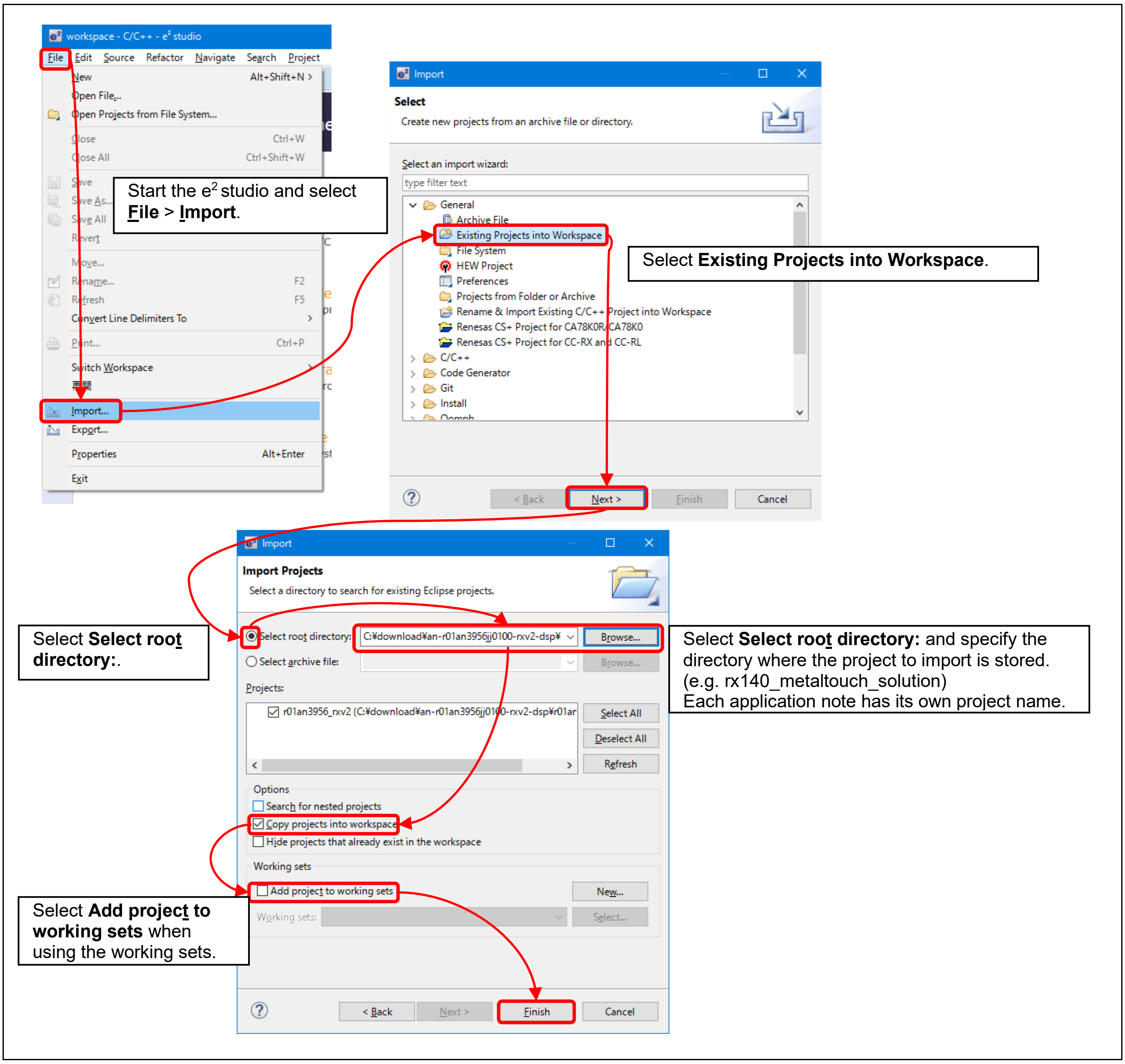


Figure 8-1 Importing a project into e^2 studio

8.2 Importing a Project into CS+

Follow the steps below to import your project into CS+. Pictures may be different depending on the version of CS+ to be used.

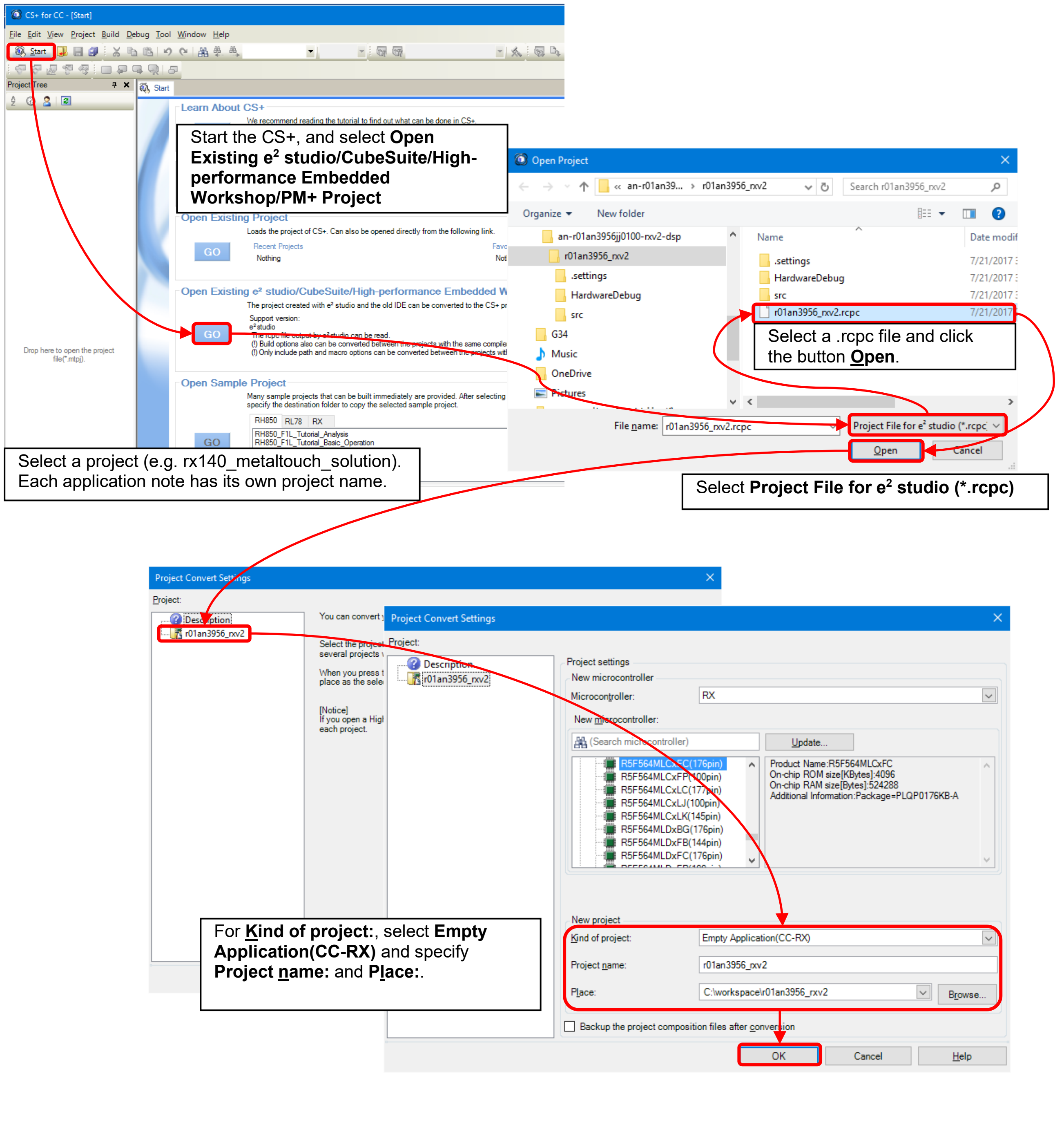


Figure 8-2 Importing a project into CS+

9. Evaluation Results with Sample Program

9.1 Memory Usage and Number of Execution Cycles

9.1.1 Build Conditions

Table 9-1 shows the build conditions for the sample program.

Table 9-1 Build Conditions

Item		Setting
Compiler		-isa=rxv2 -define=QE_TOUCH_CONFIGURATION -utf8 -nomessage -output=obj -obj_path=${workspace_loc:/${ProjName}/${ConfigName}} -debug -outcode=utf8 -nologo
Linker		-noprelink -output="rx140_metaltouch_solution.abs" -form=absolute -nomessage -vect=_undefined_interrupt_source_isr -list -show=symbol -nooptimize -rom=D=R,D_1=R_1,D_2=R_2 -cpu=RAM=00000000-0000ffff,FIX=00080000-00083fff, FIX=00086000-00087fff,FIX=00088000-0008dfff,FIX=000a0000-000bffff, ROM=00100000-00101fff,FIX=007fc000-007fc4ff,FIX=007ffc00-007fffff, ROM=fffc0000-ffffffff -nologo
	Section	SU,SI,B_1,R_1,B_2,R_2,B,R/04,PResetPRG,C_1,C_2,C,C$*,D*,W*,L,P/0FFFC0000, EXCEPTVECT/0FFFFFF80,RESETVECT/0FFFFFFFC

Note: The included path settings other than user settings are omitted.

9.1.2 Memory Usage

The amount of memory usage of this program is shown in Table 9-2.

Table 9-2 Amount of memory usage

Item			Size [byte]	
			Setting	Actual amount of usage
ROM				18317
	Code			14546
	Data			3771
RAM			11361	10453
	Data			10081
	Stack	USER	1024	220
		INT	256	152

9.1.3 Number of Execution Cycles and Processing Time

Table 9-3 shows the number of CPU execution cycles, and Figure 9-1shows a distribution of processing times in touch detection period of 20ms.

Table 9-3 Execution Cycles and Processing Time

ICLK=48MHz

Process	Maximum execution cycles (Processing time)		Condition
	Operating Mode 1	Operating Mode 2	
SW3 press detection	34cycle (0.708μs)		On pressing detection
Operating mode switching	1436cycle (29.917μs)		
Touch detection	11750cycle (244.792μs)	19477cycle (405.771μs)	
Buzzser start	287cycle (5.979μs)		On starting buzzer
Touch display	45689cycle (951.854μs)		
Subtotal during touch detection period	61130cycle (1273.542μs)	68932cycle (1436.083μs)	On touch detection
Buzzser stop	192cycle (4.000μs)		On stopping buzzer
CMI0 interrupt	1143 (23.813μs)		
CTSUFN interrupt	88 (1.833μs)		

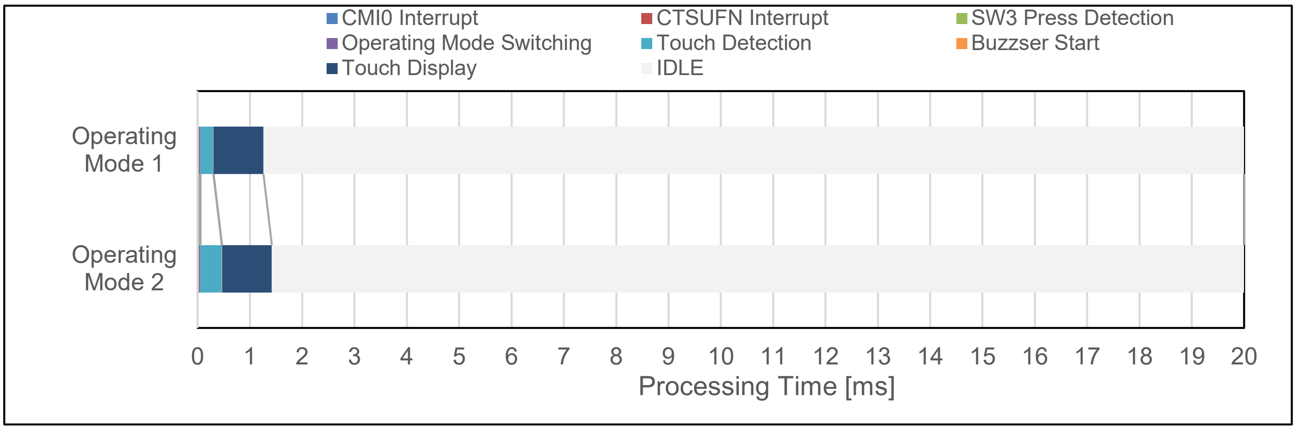


Figure 9-1 Distribution of processing time in touch detection period of 20ms

9.2 Evaluation Result

9.2.1 Conditions

Table 9-4 shows the conditions for adjusting the touch sensors with QE for Capacitive Touch, and Table 9-5 shows the threshold settings for each strength level.

Table 9-4 Adjustment conditions for touch sensor with QE for Capacitive Touch

Item	Setting
config01 pressing strength	500g
Measurement voltage setting	Normal voltage (1.5V)
Current range	40μA
Sensor drive pulse frequency	F0: 0.500MHz F1: 0.430MHz, F2: 0.570MHz
Measurement time	0.128ms
Judgment type	VMM

Table 9-5 Setting of rates of touch judgement threshold

Method	Pressing strength	Rate of touch judgement threshold
config01	High	75%
config02	Medium	50%
config03	Low	25%

9.2.2 Button Sensitivity

Figure 9-2 shows the measurement values and the touch judgement thresholds when putting a weight of 500g on each button.

The touch judgement thresholds are obtained from the measurement value at tuning with a weight of 500g.

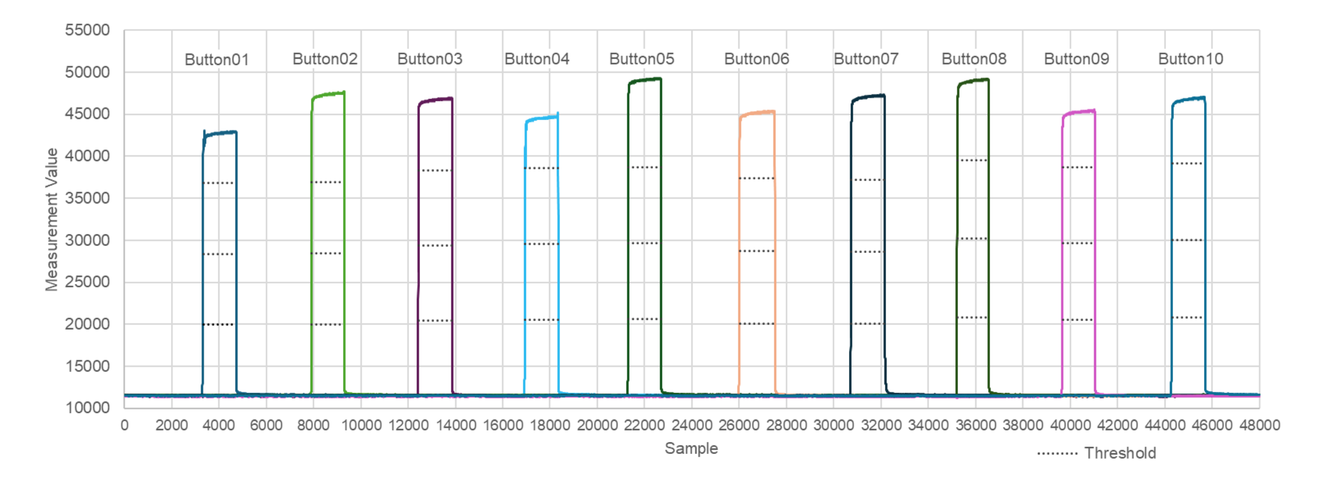


Figure 9-2 Measurement values of each button with 500g-weight and touch judgement thresholds

When putting a weight of 500g on one button, the ratio of the largest change in the measurement values of the other buttons to the threshold for touch judgement of low pressing strength is shown in Figure 9-3.

The values in Figure 9-3 are defined as follows.

$$\text{Variation ratio } [\%] = \frac{\text{Change in measurement value}}{\text{Threshold for touch judgement of the low pressing strength}} \cdot 100$$

If the variation ratio of the other buttons exceeded 100%, it would be misjudged. However, the ratio is small enough, so it is confirmed that no judgement error occurs.

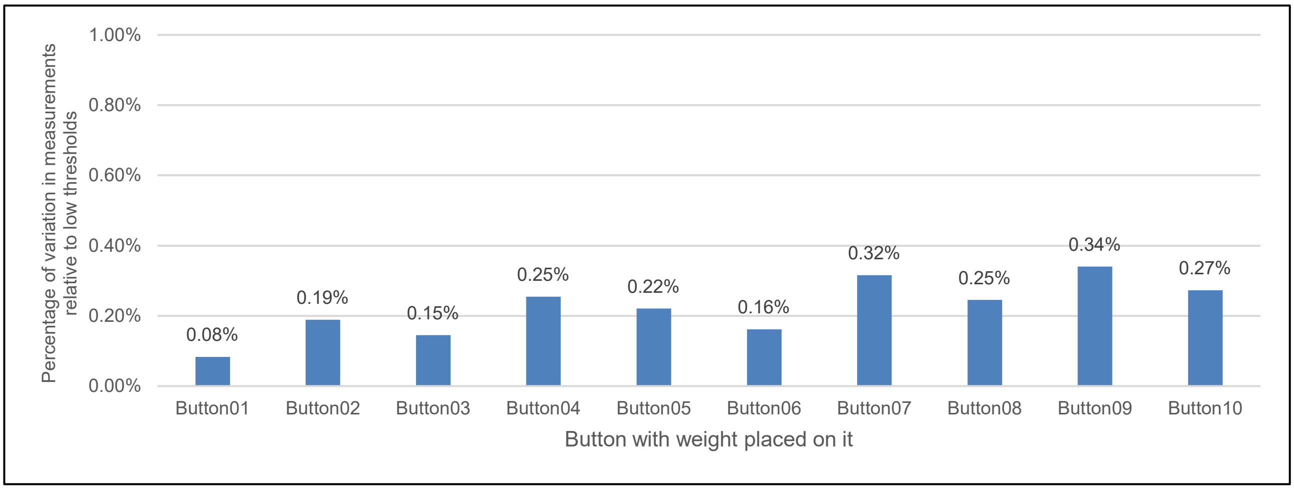


Figure 9-3 Impact on the other buttons when putting 500g-wieght

9.2.3 EMC Test Result

With this system, we conducted a self evaluation conforming to EN 61000-4. Table 9-6 shows the result.

Table 9-6 EMC test result

Item	Test level		Performance criterion
EN 61000-4-2:2009 (electrostatic discharge)	Direct discharge	±8kV	A
	Indirect discharge	±8kV	A
EN 61000-4-3:2020 (radio-frequency electromagnetic field)	10 V/m		A
EN 61000-4-4:2012 (electrical fast transient/burst)	±4kV (5kHz)		A
	±4kV (100kHz)		A
EN 61000-4-6:2014 (conducted disturbances induced by radio-frequency fields.)	10V		A

The performance criterion A indicates that a touch can be judged as expected during the test and after the test.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr.10.25	—	First edition, issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

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

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/.