

Application Note

RX Family CTSU 3D Gesture Demo Set Sample Software

R01AN4101EJ0210 Rev.2.10 Mar 30, 2018

Introduction

This application note describes the 3D Gesture Demo Software specification using a sample application based on the Capacitive Touch Sensing Unit (CTSU), the hardware that detects human touch by measuring capacitance generated between touch electrodes and the human hand.

Target Devices

RX130, RX231

Related Documents

- 1. RX Family CTSU API Reference Guide (R30AN0215EJ)
- 2. RX Family CTSU Mutual-capacitance Touch Measurement (R30AN0217EJ)
- 3. RX113 Group Design Guide for Renesas Touch Solution (R30AN0218EJ)
- 4. RX231 Group CTSU Application Example: 3D Gesture Demo Set (Hardware) (R01AN4219EJ)
- 5. RX130 Group CTSU Application Example: 3D Gesture Demo Set (Hardware) (R01AN4320EJ)
- 6. RX Family CTSU 3D Gesture Demo Set Evaluation Tool '3D Monitor' (R20AN0501EJ)



Contents

1. Overview	3
1.1 Software Structure	3
1.2 File Structure	4
2. Operation Confirmation Environment	
2.1 Regular version	
2.2 Small Version	5
	~
3. CTSU API Setting	
3.1 Regular Version	
3.2 Small Version	6
4. 3D Position Calculation	7
4.1 3D Position Definition	
4.2 3D Position Calculation Principle	7
4.3 3D Position Calculation Pre-Measurement Conditions	
4.4 3D Position Calculation	
4.4.1 Z position	8
4.4.2 X position	9
4.4.3 Y position	10
4.5 Moving Average Filter	10
4.6 Drift Correction	10
4.7 Noise Environment Judgement	11
4.7.1 High noise environment detection judgement	11
4.7.2 Normal environment recovery judgement	12
5. 3D Position Calculation Software Specification	
5.1 File Configuration	
5.2 Constants	
5.3 Structures	
5.4 API Function Specifications	17
6. Gesture Recognition Library Specification	21
6.1 File Configuration	
6.2 Constants	
6.3 Structures	
6.4 API Function Specifications	
7. Sample Application	23



1. Overview

Two versions of the 3D Gesture Demo Set are available: RX231_3D Gesture Demo Set (referred to as the regular version) and RX130_3D Gesture Demo Set (referred to as the small version). For descriptions of the corresponding hardware, please refer to related documents 4 and 5 listed in the introduction.

This application note describes sample software that runs on both versions of the 3D Gesture Demo Set.

1.1 Software Structure

Figure 1.1 shows software structures for normal and small versions of the application.

The CTSU API controls the CTSU to measure electrostatic capacitance. The API was automatically generated using Workbench6, the integrated development environment for capacitance touch.

The 3D position calculation API calculates the 3D position from the results of the electrostatic capacitance measured by the CTSU API.

The gesture recognition library is used to analyze changes in the 3D position and determine gestures.

The application notifies the user of the 3D position calculation and gesture recognition results either via UART communication between the 3D Gesture Demo Set LED display and the BLE module or via USB communication with the user's PC.

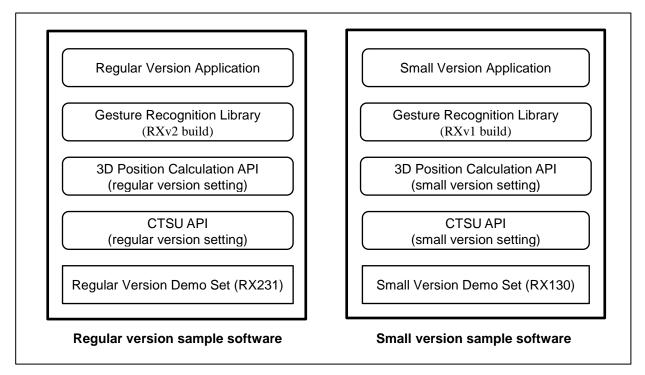


Figure 1.1 Software Structure



1.2 File Structure

Table 1.1 shows the application's file structure; the CTSU API has been left out for brevity.

Table 1.1	File Structure

Folder/File Name	Description		
r_01an4101jj0210-ctsu	Project folder		
TouchAPI_3DGestureRegular	Workbench6 automatically generated software		
TouchApiBase.mtpj	Project file		
TouchApiBase.rcpe	Common project file		
Gesture_Recognition_RXv2.lib	Gesture recognition library (RXv2 build)		
Source	Source folder		
r_cap_position.c	3D position calculation API source file		
Config_SCI12.c	SCI12 source file		
Config_SCI12_user.c	SCI12 user source file		
r_main.c	Application file		
Include	Include folder		
r_cap_position_config.h	3D position calculation API config file (regular version setting)		
r_cap_position_measure.h	3D position calculation API previous measured value file (regular version setting)		
r_cap_position_if.h	3D position calculation API interface file		
r_cap_position.h	3D position calculation API header file		
r_cap_gesture_if.h	Gesture recognition interface file		
Config_SCI12.h	SCI12 header file		
wbSetting	Workbench6 setting folder		
TouchAPI_3DGestureSmall	Workbench6 automatically generated software		
TouchApiBase.mtpj	Project file		
TouchApiBase.rcpe	Common project file		
Gesture_Recognition_RXv1.lib	Gesture recognition library (RXv1 build)		
Source	Source folder		
r_cap_position.c	3D position calculation API source file		
Config_SCI1.c	SCI1 source file		
Config_SCI1_user.c	SCI1 user source file		
r_main.c	Application file		
Include	Include folder		
r_cap_position_config.h	3D position calculation API config file (small version setting)		
r_cap_position_measure.h	3D position calculation API previous measured value file (small version setting)		
r_cap_position_if.h	3D position calculation API interface file		
r_cap_position.h	3D position calculation API header file		
r_cap_gesture_if.h	Gesture recognition interface file		
Config_SCI1.h	SCI1 header file		
wbSetting	Workbench6 setting folder		



2. Operation Confirmation Environment

2.1 Regular version

Table 2.1 lists the operating conditions of the regular version software.

Table 2.1 Operating Environment (regular version)

Item	Description
Evaluation board (in custom case)	RTK5RX2310D00000BR
MCU used	RX231
Operating frequency	54MHz
Operating voltage	3.3V
Integrated Development Environment	CS+ v6.01.00
C compiler	CC-RX v2.08.00
Capacitance touch IDE	Workbench6 V01.07.00.00

Figure 2.1 shows the connection diagram for the Gesture Demo Set.

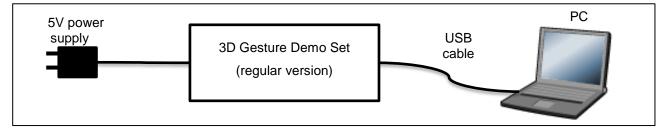


Figure 2.1 Demo Set Connection Diagram (regular version)

2.2 Small Version

Table 2.2 describes the operating environment of the small version of the software.

Table 2.2	Operating Environment (small version)
-----------	--

Item	Description
Evaluation board (in custom case)	RTK0EG0014D00001BJ
MCU used	RX130
Operating frequency	32MHz
Operating voltage	5.0V
Integrated Development Environment	CS+ v6.01.00
C compiler	CC-RX v2.08.00
Capacitance touch IDE	Workbench6 v01.07.00.00

Figure 2.2 shows the connection diagram for the Gesture Demo Set.

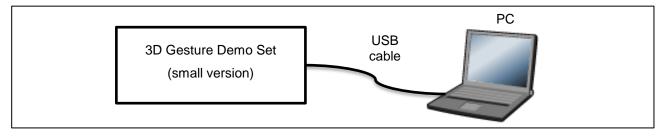


Figure 2.2Demo Set Connection Diagram (small version)



3. CTSU API Setting

3.1 Regular Version

Set the frequency of the sensor drive pulse generated based on the CTSU base clock to 3.375 MHz. Setting the frequency to a low value will shorten the 3D position detection distance.

Edit r_touch,h to set the measurement cycle to 10 ms.

```
#define TOUCH_MAIN_CYCLE_US 10000
```

 $Edit \ r_usb_pcdc_apl.c \ to \ enable \ communication \ with \ the \ evaluation \ application.$

Add sample application function SampleAplReceive() to function usb_psmpl_MainTask()

3.2 Small Version

Set the frequency of the sensor drive pulse generated based on the CTSU base clock to 4.000 MHz. Setting the frequency to a low value will shorten the 3D position detection distance.

Edit r_touch,h to set the measurement cycle to 10 ms.

#define TOUCH_MAIN_CYCLE_US 10000

Edit r_cgc.h to enable USB connection with the PC. #define WORKBENCH_SERIAL_CONNECT (1)

Edit r_mpc.c to enable BLE module and UART transmissions, or button-type input.

Function initial_port_set() PORT3.PDR.BYTE = PORT3.PDR.BYTE | 0x1E; PORT2.PDR.BYTE = 0xBF; PORTA.PDR.BYTE = PORTA.PDR.BYTE | 0xE7;

Edit r_sci.c to enable communication with the evaluation application. Add sample application function SampleApIReceive() to function r_sci6_receive_interrupt().



4. 3D Position Calculation

4.1 3D Position Definition

As shown in Figure 4.1, with 0 as the origin at the center of the sensor board surface, the X axis indicates left and right directions horizontal to the board, the Y axis indicates top and bottom directions horizontal to the board, and the Z axis indicates the vertical direction from the board. The 3D position of the hand is expressed as (x, y, z). Values for x and y are integers, and z is a natural number. Positions are measured in millimeters (mm). The upper right of the board is the first quadrant (x > 0, y > 0), the upper left is the second quadrant (x < 0, y > 0), the lower left is the third quadrant (x < 0, y < 0).

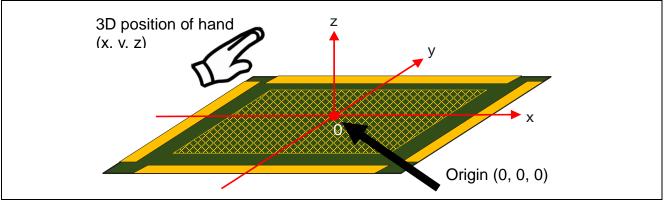


Figure 4.1 3D Position Definition

4.2 3D Position Calculation Principle

Calculation of the 3D position uses the capacitance measurement results (referred to as "count value" herein) of the top, bottom, left and right receiver electrodes (4 in total).

The reference value is the count value measured when there is nothing (hand or other conductive object) located near the evaluation board. When a hand approaches the board, the count value of the 4 receiver electrodes decrease according to the position of the hand. The 3D position of the hand is calculated from the difference between the reference value and the count value. If the count values are not decreasing, the 3D position cannot be calculated in that spot.

The relationship between this count value and the 3D position differ depending on the hardware, and values cannot be calculated using general formulas. Therefore, before developing with the sample software, it is important to first obtain a count value, or count values measured at several positions, from the hardware you are using when there is no conductive object in the vicinity.

Figure 4.2 shows the reference and count values when a hand is placed near, then removed from, the electrodes.

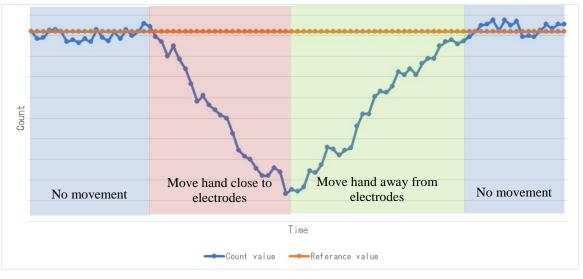


Figure 4.2 Reference Value and Count Value



4.3 3D Position Calculation Pre-Measurement Conditions

The required conditions of the positions before measurement is obtained are as follows.

[Z axis]

- Count values of 4 electrodes in 3 to 10 locations where x and y are 0
 - Calculation accuracy increases as the number of measured positions increases.
 - The farthest measurement position is the calculation range of z. We recommend obtaining at least 100 counts equaling the difference with the reference value.

[X axis]

• Count values of right and left electrodes in 1 location 30mm or more in the plus direction of the X axis from the position measured on the Z axis

[Y axis]

- Count values of top and bottom electrodes in 1 location 30mm or more in the plus direction of the Y axis from the position measured on the Z axis
 - Not required if the electrodes are vertically and horizontally symmetrical. This software does not measure in the Y axis in this case.

Please describe the measured count values in the 3D position calculation API pre-measurement file.

4.4 3D Position Calculation

4.4.1 Z position

For count value calculations, \triangle top represents the top electrode difference, \triangle bottom represents the bottom electrode, \triangle right represents the right electrode, and \triangle left represents the left electrode. The average of these differences is represented by \triangle average.

Calculate \triangle average at each Z position from the pre-measured count values, and then calculate Z position using the linear interpolation approximation formula (\triangle average_REF).

Figure 4.3 shows \triangle average_REF of the evaluation board. The vertical axis is \triangle average and the horizontal axis is Z position.

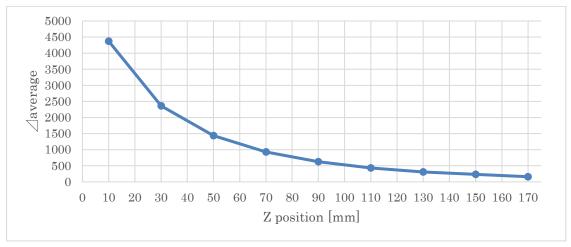


Figure 4.3 Evaluation Board ∠average_REF

For example, if the pre-measured value obtained is

 \triangle average = 500 at (0, 0, 100), and \triangle average = 250 at (0, 0, 150),

then when \triangle average is 325,



$$z = 100 + ((500 - 325) / ((500 - 250) / (150 - 100)))$$

= 100 + (175 / 5)
= 135

4.4.2 X position

The difference between rightarrow right and rightarrow left is rightarrow horizontal.

Although \angle horizontal and X position are proportional, the slope (\angle x_slope) differs depending on Z position.

Figure 4.4 shows \angle horizontal of the evaluation board. The vertical axis is \angle horizontal and the horizontal axis is X position.

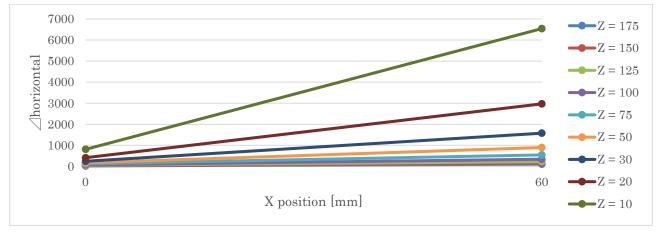
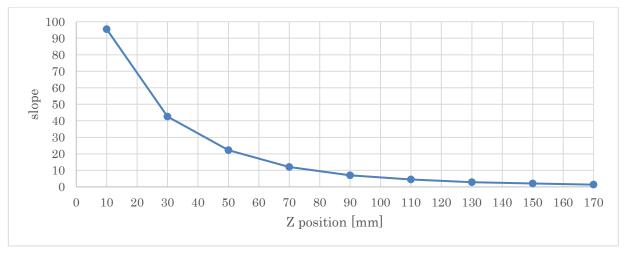


Figure 4.4 Evaluation Board ∠horizontal

Calculate $\[these x]$ slope at each Z position from the pre-measured count values, and then X position is calculated using the linear interpolation approximation formula ($\[these x]$ slope_REF).

Figure 4.5 shows $\[t]x_slope_REF$ of the evaluation board. The vertical axis is $\[t]x_slope$ and the horizontal axis is Z position.





Finally, correct the sensitivity difference of left and right electrodes from \triangle horizontal at the pre-measured X position 0. Because \triangle horizontal varies according to the Z position, linear interpolation approximation formula is used in the same way. Let this correction value be \triangle horizontal_offset.



For example, if the pre-measured values obtained are:

Z position is 100mm and $\angle x_{slope} = 8$,

Z position is 150mm and $\angle x_slope = 3$,

and Z position is calculated as 135mm,

 $\Delta x_{slope} = 3 + ((150 - 135) * ((8 - 3) / (150 - 100)))$

= 3 + (15 * (5 / 50))= 3 + 1.5

And, when \angle horizontal is 180 and \angle horizontal_offset is 18 when Z position is 135mm,

$$\mathbf{x} = (180\text{-}18) / 4.5 = 36$$

= 4.5

4.4.3 Y position

The difference between \triangle top and \triangle bottom is \triangle vertical.

The calculation method for y is the same as that of x.

4.5 Moving Average Filter

The moving average filter is processed according to the obtained count values to enhance noise immunity.

The filter averages the count values up to the number of times before the moving average setting value.

The calculations described in sections 4.4 and 4.6 use the average values obtained with the moving average filter applied.

Describe the moving average setting value in the config file of the 3D position calculation API. The higher the value, the better the noise immunity, but the slower the response time.

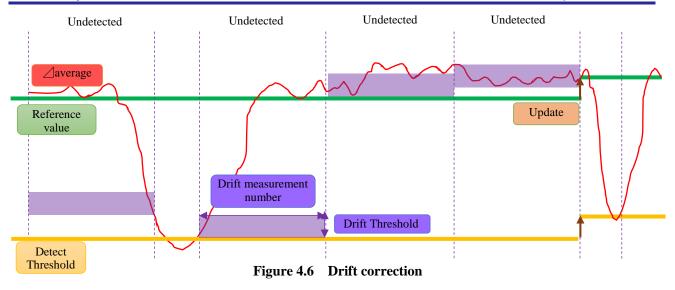
4.6 Drift Correction

Drift correction is a process of updating the reference value so as to follow the change of the surrounding environment when judging non-detection.

When the noncontiguous determination continues for the number of drift measurements, if the difference between the maximum and minimum values of \triangle average is within the drift threshold value, the reference value is updated to the current count value (moving average filter applied).

The user should describe the drift measurement number and the drift threshold in the config file of the 3D position calculation API. As the number of drift measurements is decreased and the drift threshold value is increased, the ability to adjust to changes in the surrounding environment improves. However, if drift correction is performed even when the hand is moving slowly, the result of 3D position calculation may be affected.





4.7 Noise Environment Judgement

3D position calculation is realized by capturing very minute changes in count values. In addition to the various noise immunity measures built into the CTSU, software-based noise filters are also implemented in the CTSU API, but it is currently difficult to completely remove noise.

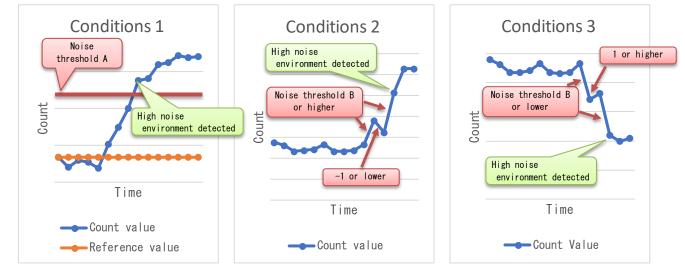
Therefore, if the software determines that the noise level will greatly affect 3D position calculation, it stops the calculation. The high noise environment continues to be judged until a normal environment recovery is confirmed.

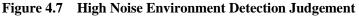
4.7.1 High noise environment detection judgement

When the count value meets one of the following two conditions, a high noise environment is detected. This judgement is conducted separately for each count value obtained from the 4 electrodes.

- 1. (Count value reference value) is A or more
- 2. (Count value previous count value) & (previous count value 2nd preceding count value) & (2nd preceding count value 3rd preceding count value) is (B or higher) & (-1 or lower) & (B or higher).
- 3. (Count value previous count value) & (previous count value 2nd preceding count value) & (2nd preceding count value 3rd preceding count value) is (B or lower) & (1 or higher) & (B or lower).
 - A: Noise threshold value A B: Noise threshold value B

Figure 4.7 shows figures describing the conditions of the high noise environment detection judgement process.







Please describe noise threshold A and noise threshold B in the 3D position calculation API configuration file. The smaller the value, the more effective in detecting a high noise environment. However, this also increases the possibility of erroneously determining a normal environment as a high noise environment.

4.7.2 Normal environment recovery judgement

A "normal environment recovery" is judged when the following conditions are meet.

When the average count value of four electrodes for the past N times shows the difference between the maximum and minimum values as M or less. If the value at this time is larger than the average reference value of 4 electrodes, update the reference value.

N: recovery measurement times M: recovery threshold

This judgment is carried out every time a count value is obtained after saving the count value for N times after a high noise environment is detected.

Figure 4.8 shows figures describing the judgement process of normal environment recovery.



Figure 4.8 Judgement of Normal Environment Recovery

Describe the recovery measurement times and recovery threshold in the 3D position calculation API configuration file. As the number of recovery measurements decreases and the recovery threshold increases, the normal environment recovery will be faster, the possibility of erroneously determining a recovery within a high noise environment increases.



5. 3D Position Calculation Software Specification

5.1 File Configuration

Table 5.1 shows the source file used in this sample software.

	Description	Natao
File Name	Description	Notes
r_cap_position.c	3D position calculation API source file	-

Table 5.2 shows the header files.

File Name	Description	Notes
r_cap_position_config.h	3D position calculation API config file	-
r_cap_position_measure.h	3D position calculation API pre- measurement file	-
r_cap_position_if.h	3D position calculation API interface file	-
r_cap_position.h	3D position calculation API header file	For use in 3D position calculation API

Table 5.2Header Files

5.2 Constants

Table 5.3 shows the r_cap_positon_if.h constants.

Table 5.3 Constants (r_cap_position_if.h)

Constant Name	Setting Value	Description
CAPPOS_NODETECT	(0x3FFF)	Value displayed when no 3D position calculation results are detected
CAPPOS_SUCCESS	(0x00)	3D position calculation API completed normally
CAPPOS_ERROR	(0x01)	3D position calculation API error occurred
CAPPOS_NORMAL	(0x00)	Value displayed when result of noise environment is normal
CAPPOS_NOISY	(0x01)	Value displayed when result of noise environment is noise detection

Table 5.4 shows the r_cap_position_config.h constants.

The default setting value is the value of the evaluation board; change as necessary to fit the usage environment.

 Table 5.4
 Constants
 (r_cap_position_config.h)

Constant Name	Setting	Setting	Description
	Value	Value	
	(Regular)	(Small)	
CAPPOS_CNF_MOVAVG_NUM	(15)	(15)	Moving average setting value
CAPPOS_CNF_DRIFT_NUM	(150)	(150)	Drift measurement number
CAPPOS_CNF_DRIFT_THR	(25)	(25)	Drift threshold
CAPPOS_CNF_RESUME_NUM	(25)	(25)	resume measurement number
CAPPOS_CNF_RESUME_THR	(20)	(20)	Resume threshold
CAPPOS_CNF_NOISE_THR_A	(150)	(150)	Noise threshold A (difference with reference value)
CAPPOS_CNF_NOISE_THR_B	(50)	(50)	Noise threshold B (difference with past value)
CAPPOS_CNF_MAX_X	(100)	(100)	X direction calculation range [mm]
CAPPOS_CNF_MAX_Y	(100)	(100)	Y direction calculation range [mm]



Table 5.5 shows the r_cap_position_measure.h constants.

The default setting value is the value of the evaluation board; change as necessary to fit the usage environment.

Constant Name	Setting Value	Setting Value	Description
	(Regular)	(Small)	
CAPPOS_MEAS_Z0	(10)	(10)	Measured Z direction position [mm]
	(10)	(10)	Set to 1 or higher.
CAPPOS_MEAS_Z1	(20)	(20)	Measured Z direction position [mm]
	(30)	(20)	Set to a value higher than CAPPOS_MEAS_Z0
CAPPOS_MEAS_Z2	(50)	(20)	Measured Z direction position [mm]
	(50)	(30)	Set to a value higher than CAPPOS_MEAS_Z1
CAPPOS_MEAS_Z3			Measured Z direction position [mm].
	(70)	(40)	Set to a value higher than CAPPOS_MEAS_Z2.
	(70)	(40)	If positions were measured in 3 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_3" constants.
CAPPOS_MEAS_Z4			Measured Z direction position [mm].
	(90)	(50)	Set to a value higher than CAPPOS_MEAS_Z3.
	(00)	(00)	If positions were measured in 4 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_4" constants.
CAPPOS_MEAS_Z5			Measured Z direction position [mm].
	(110)	(60)	Set to a value higher than CAPPOS_MEAS_Z4.
	(110)	(00)	If positions were measured in 5 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_5" constants.
CAPPOS_MEAS_Z6			Measured Z direction position [mm].
	(130)	(70)	Set to a value higher than CAPPOS_MEAS_Z5.
	(/	< - /	If positions were measured in 6 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_6" constants.
CAPPOS_MEAS_Z7			Measured Z direction position [mm].
	(150)	(80)	Set to a value higher than CAPPOS_MEAS_Z6.
	. ,	. ,	If positions were measured in 7 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_7" constants.
CAPPOS_MEAS_Z8			Measured Z direction position [mm].
	(170)	(90)	Set to a value higher than CAPPOS_MEAS_Z7.
			If positions were measured in 8 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_8" constants.
CAPPOS_MEAS_Z9			Measured Z direction position [mm].
	(0)	(100)	Set to a value higher than CAPPOS_MEAS_Z8. If positions were measured in 9 locations, set to 0 for this constant and all
			"CAPPOS_MEAS_*_9" constants.
CAPPOS MEAS X			Measured X direction position [mm]
	(60)	(60)	If the X direction is not measured, set to 0 for this constant.
CAPPOS_MEAS_Y			Measured Y direction position [mm]
	(0)	(0)	If the Y direction is not measured, set to 0 for this constant.
CAPPOS_MEAS_TOP	(44637)	(37390)	Top electrode count value when nothing is in vicinity of board
CAPPOS_MEAS_TOP	(40961)	(38310)	Bottom electrode count value when nothing is in vicinity of board
CAPPOS_MEAS_BIM	(40961)	(37725)	Right electrode count value when nothing is in vicinity of board
CAPPOS_MEAS_RGT	, ,	, ,	
	(37431)	(40357)	Left electrode count value when nothing is in vicinity of board
CAPPOS_MEAS_TOP0	(40782)	(33307)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_TOP1	(42607)	(35353)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_TOP2	(43417)	(36185)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_TOP3	(43844)	(36614)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z3)



CTSU 3D Gesture Demo Set Sample Software

CAPPOS_MEAS_TOP4	(44108)	(36842)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_TOP5	(44272)	(36982)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_TOP6	(44373)	(37100)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_TOP7	(44432)	(37199)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_TOP8	(44505)	(37221)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_TOP9	(0)	(37279)	Top electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_BTM0	(35517)	(34539)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_BTM1	(38022)	(36341)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_BTM2	(39181)	(37075)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_BTM3	(39823)	(37490)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z3)
CAPPOS_MEAS_BTM4	(40202)	(37721)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_BTM5	(40440)	(37860)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_BTM6	(40589)	(37975)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_BTM7	(40679)	(38061)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_BTM8	(40769)	(38127)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_BTM9	(0)	(38197)	Bottom electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_RGT0	(35751)	(33743)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_RGT1	(37803)	(35698)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_RGT2	(38757)	(36484)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_RGT3	(39273)	(36901)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z3)
CAPPOS_MEAS_RGT4	(39580)	(37139)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_RGT5	(39791)	(37288)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_RGT6	(39922)	(37407)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_RGT7	(40009)	(37495)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_RGT8	(40079)	(37534)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_RGT9	(0)	(37597)	Right electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_LFT0	(33743)	(36378)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_LFT1	(35399)	(38354)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_LFT2	(36183)	(39150)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_LFT3	(36628)	(39562)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z3)
CAPPOS_MEAS_LFT4	(36886)	(39802)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_LFT5	(37048)	(39938)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_LFT6	(37161)	(40047)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_LFT7	(37222)	(40110)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_LFT8	(37286)	(40176)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_LFT9	(0)	(40248)	Left electrode count value measured when coordinates are (0, 0, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_X_RGT0	(32791)	(34470)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_X_RGT1	(36733)	(35920)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_X_RGT2	(38231)	(36622)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_X_RGT3	(38998)	(36998)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z3)
CAPPOS_MEAS_X_RGT4	(39423)	(37212)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_X_RGT5	(39704)	(37340)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_X_RGT6	(39868)	(37453)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_X_RGT7	(39971)	(37513)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_X_RGT8	(40062)	(37591)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_X_RGT9	(0)	(37616)	Right electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_X_LFT0	(36513)	(39557)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_X_LFT1	(36884)	(39824)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_X_LFT2	(36990)	(39943)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_X_LFT3	(37078)	(40033)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z3)
CAPPOS MEAS X LFT4	(37152)	(40108)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_X_LFT5	(37229)	(40149)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_A, 0, CAPPOS_MEAS_24)
	(0.220)	N	Loss of the second second to the second



CAPPOS_MEAS_X_LFT6	(37275)	(40211)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_X_LFT7	(37310)	(40216)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_X_LFT8	(37350)	(40284)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_X_LFT9	(0)	(40299)	Left electrode count value measured when coordinates are (CAPPOS_MEAS_X, 0, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_Y_TOP0	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_Y_TOP1	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_Y_TOP2	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_Y_TOP3	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z3)
CAPPOS_MEAS_Y_TOP4	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_Y_TOP5	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_Y_TOP6	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_Y_TOP7	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_Y_TOP8	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_Y_TOP9	(0)	(0)	Top electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z9)
CAPPOS_MEAS_Y_BTM0	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z0)
CAPPOS_MEAS_Y_BTM1	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z1)
CAPPOS_MEAS_Y_BTM2	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z2)
CAPPOS_MEAS_Y_BTM3	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z3)
CAPPOS_MEAS_Y_BTM4	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z4)
CAPPOS_MEAS_Y_BTM5	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z5)
CAPPOS_MEAS_Y_BTM6	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z6)
CAPPOS_MEAS_Y_BTM7	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z7)
CAPPOS_MEAS_Y_BTM8	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z8)
CAPPOS_MEAS_Y_BTM9	(0)	(0)	Bottom electrode count value measured when coordinates are (0, CAPPOS_MEAS_Y, CAPPOS_MEAS_Z9)

5.3 Structures

Table 5.6 shows the st_position_result_t structures.

Table 5.6	st_cappos_	_result_	t Structures
-----------	------------	----------	--------------

Туре	Member	Description
int16_t	coord_x	x position calculation result [mm]
int16_t	coord_y	y position calculation result [mm]
int16_t	coord_z	z position calculation result [mm]
uint8_t	noise	Result of noise environment detection

Table 5.7 shows the st_captouch_data_t structures.

Туре	Member	Description
uint16_t	top	Top electrode count value
uint16_t	btm	Bottom electrode count value
uint16_t	rgt	Right electrode count value
uint16_t	lft	Left electrode count value



5.4 API Function Specifications

5.4.1 R_CAPPOS_Create

Initialization API

Format

uint8_t R_CAPPOS_Create(void)

Argument

Note

Return Value

CAPPOS_SUCCESS	Successful completion
CAPPOS_ERROR	Abnormal completion

Description

This API initializes the 3D position calculation API variables and creates the 3D position calculation coefficients from the 3D position calculation API config file. If there is a problem in the 3D position calculation API config file, the function returns CAPPOS_ERROR.

Remarks

This function is called only once immediately after system startup.



5.4.2 R_CAPPOS_GetAve

Get moving average value API

Format	
uint8_t	R_CAPPOS_GetAve(st_captouch_data_t * p_captouch_data)
Argument	
*p_captouch_data	4 electrode counter value data pointer
Return Value	
CAPPOS_SUCCESS CAPPOS_ERROR	Successful completion Abnormal completion

Description

This API stores the moving average value of the four electrodes stored by the 3D position calculation API in the argument *p_captouch_data.

When the argument is NULL, the function returns CAPPOS_ERROR.



5.4.3 R_CAPPOS_GetRef

Get reference value API

Format	
uint8_t	R_CAPPOS_GetRef(st_captouch_data_t * p_captouch_data)
Argument	
*p_captouch_data	4 electrode counter value data pointer
Return Value	
CAPPOS_SUCCESS CAPPOS_ERROR	Successful completion Abnormal completion

Description

This API stores the reference value of the four electrodes stored by the 3D position calculation API in the argument *p_captouch_data.

When the argument is NULL, the function returns CAPPOS_ERROR.



5.4.4 R_CAPPOS_Read

3D Position Calculation API

Format

uint8_t R_CAPPOS_Read(st_cappos_result_t * p_cappos_result, const st_captouch_data_t * p_captouch_data)

Argument

*p_cappos_result *p_captouch_data	3D position calculation result pointer 4 electrode counter value data pointer
Return Value	
CAPPOS_SUCCESS	Successful completion

CAPPOS_SUCCESS Successful completion CAPPOS_ERROR Abnormal completion

Description

This API stores the results of the 3D position calculated from the value of the 4-electrode count of argument *p_captouch_data in argument *p_cappos_result.

When high noise environment, CAPPOS_NOISY is stored in argument *p_cappos_result. When the argument is NULL, or when R_CAPPOS_Create() is not called before the API call, the function returns CAPPOS_ERROR.

Remarks

If the API is not called the number of times indicated by the moving average setting value (CAPPOS_CNF_MOVAVG_NUM), CAPPOS_NODETECT is stored in argument *p_cappos_result.

Call this API after confirming that return value of R_Set_Cap_Touch_Result_Create() is 0_SUCCESS in the CTSU API's electrostatic capacitance measurement period.

Calling this API periodically also automatically implements the moving average filter (refer to 4.5), the drift correction (refer to 4.6), and the high noise environment judgement (refer to 4.7).



6. Gesture Recognition Library Specification

6.1 File Configuration

Table 6.1 shows the library files.

File Name	Description	Notes
Gesture_Recognition_RXv2.lib	Gesture recognition library file (RXv2 build)	For regular version
Gesture_Recognition_RXv1.lib	Gesture recognition library file (RXv1 build)	For small version

Table 6.2 shows the header file.

	Table 6.2	Header	File
--	-----------	--------	------

File Name	Description	Notes
r_cap_gesture_if.h	Gesture recognition API interface file	-

6.2 Constants

Table 6.3 shows the constants. These are defined by enumerated type e_gesture_result_t.

Constant Name	Setting Value	Description
GESTURE_RESULT_NONE	(0)	No recognition
GESTURE_RESULT_RIGHT_SWIPE	(1)	Right swipe
GESTURE_RESULT_LEFT_SWIPE	(2)	Left swipe
GESTURE_RESULT_FRONT_SWIPE	(3)	Front swipe
GESTURE_RESULT_BACK_SWIPE	(4)	Back swipe
GESTURE_RESULT_DOWN_SWIPE	(5)	Push
GESTURE_RESULT_CW_SLOW	(6)	Draw circle (Clockwise, Slow)
GESTURE_RESULT_CW_FAST	(7)	Draw circle (Clockwise, Fast)
GESTURE_RESULT_RESERVE	(8)	Reserved
GESTURE_RESULT_CCW_SLOW	(9)	Draw circle (Counter clockwise, Slow)
GESTURE_RESULT_CCW_FAST	(10)	Draw circle (Counter clockwise, Fast)
GESTURE_RESULT_RESERVE2	(11)	Reserved2

Table 6.3 Constants

6.3 Structures

Table 6.4 shows the st_cappos_input_t structure.

Туре	Member	Description
int16_t	coord_x	x position calculation result [mm]
int16_t	coord_y	y position calculation result [mm]
int16_t	coord_z	z position calculation result [mm]
uint8_t	noise	Result of noise environment detection



6.4 API Function Specifications

6.4.1 R_GESTURE_Detect

Gesture Recognition API

形式

uint16_t R_GESTURE_Detect(e_gesture_result_t * p_result, const st_cappos_input_t * p_cappos_input)

引数

e_gesture_result_t	*p_result	Gesture recognition result pointer
st_cappos_input_t	*p_cappos_input	3D Position Calculation input pointer

戻り値

0 Successful completion

解説

This API stores the result of gesture recognition processing in p_result with the 3D position calculation result as input.



7. Sample Application

The 3D Gesture sample application has been added based on the CTSU API application file r_main.c.

• Demo Set initialization

Initialization function SampleInitialize() has been added before the main function loop.

• Results notification

Function SampleApplication(), which notifies the user of the 3D position calculation and gesture recognition results after return value _0_SUCCESS has been confirmed in R_Set_Cap_Touch_Result_Create () of the main function loop.

Based on this, SampleApplication() is called for each touch measurement cycle.

• LED display

Table 7.1 shows the display for each LED.

Table 7.1 LED Display

Regular ver. LED	Small ver. LED	ON	OFF
LED1 (green)	D10 (orange)	Initialization complete	Initialization incomplete
LED2 (orange)	D9 (green)	3D position calculation results detected	3D position calculation results not detected
LED3 (green)	D8 (orange)	Noise environment detected	Normal environment
LED4 (orange)	D7 (green)	Communicating with evaluation app	Not communicating with evaluation app

• UART communication with BLE module

The regular version uses SCI12 (TXD12: PE1, RXD12: PE2) for UART communication, the small version uses SCI1 (TXD1: P26, RXD1: P30).

Table 7.2 lists UART settings.

Setting	Value
Baudrate	250000 bps
Data length	8 bits
Parity	none
Stop bit	1 bit
Flow control	none

Table 7.2UART Settings

When 0xAA is received from the BLE module, 8 bytes of data are transmitted, with the following values:

4th byte: X coordinate value

5th byte: Y coordinate value

6th byte: Z coordinate value

1st, 2nd, 3rd, 7th and 8th bytes: 0x00



• USB communication with evaluation tool

Function SampleAplReceive(), which is called from the USB receive function in the CTSU API, has been added to the regular version.

Function SampleAplReceive(), which is called from the SC16 receive interrupt handler in the CTSU API, has been added to the small version.

Communication with the evaluation tool is conducted in the format shown in Figure 7.1

Market 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 2 HEAD Lurt TOP BTM RGT LUT TOP BTM RGT RGT <td< th=""><th>$RX \rightarrow PC$</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></td<>	$RX \rightarrow PC$								
wead mean D CMD Model Cast X Y Z TOP BTM RGT LFT TOP.R BTM.R RGT.R LFT.R Ex Dx55 Constant value D									
HEAD Header Qx55 Constant value ID ID unsigned char Count up for each communication GMD Command 0x03 Result of measurement NOISE Result of measurement noise environment noise environment 0 norise environment 1 Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 9 Counter Clockwise(fast) 8 Reserve 0 Counter Clockwise(fast) 8 Reserve X, Y, Z Result of 3D position calucuration aigned short 3D coordinate value 0x3FFF No detection TOP, BTM, RGT, LFT Value of R_CAPPOS_GetAve unsigned short Reserve value Constant value Corstant value PC → RX Value of R_CAPPOS_GetAve wurd <u>tell 2 3 4</u> value <u>Constant value</u> <tr< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>_</td></tr<>									_
0x55 Constant value ID unsigned char Count up for each communication CMD Command 0x03 Result of noise environment 0 normal environment 1 noise environment GEST Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Outback (slow) 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve settion 12 Value of R_OPOS_GetAve 13 Reserve value 25 Counter Clockwise(fast) 8 Reserve settion 9 Counter Clockwise(fast) 11 Reserve 25 Result of 30 position calucuration 13 Reserve settion 10 Counter Clockwise(fast) 11 Reserve settion 25 Counter Clockwise(fast) 11 Reserve settion 26 Counter Clockwise(fast) 10 Counter Clockwise(fast)	val HEAD ID CMD NOISE GEST X	Z TOP BTM	I RGT	LFT	TOP_R	BTM_R	RGT_R	LFT_R	END
0x55 Constant value ID unsigned char Count up for each communication CMD Command 0x03 Result of noise environment 0 normal environment 1 noise environment GEST Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Outback (slow) 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve settion 12 Value of R_OPOS_GetAve 13 Reserve value 25 Counter Clockwise(fast) 8 Reserve settion 9 Counter Clockwise(fast) 11 Reserve 25 Result of 30 position calucuration 13 Reserve settion 10 Counter Clockwise(fast) 11 Reserve settion 25 Counter Clockwise(fast) 11 Reserve settion 26 Counter Clockwise(fast) 10 Counter Clockwise(fast)									
ID ID unsigned char Count up for each communication OMD Command 0x03 Result of measurement 0 normal environment 0 noise environment 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Push 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve 8 Reserve 9 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve 12 Reserve 9 Counter Clockwise(fast) 11 Reserve 12 Reserve 9 Counter Clockwise(fast) 11 Reserve 0 Counter Clockwise(fast) 11 Reserve 0 Counter Clockwise(fast) 13 Counter Clockwise(fast) 14 Reserve 9 Counter Clockwise(fast) 11 Reserve									
unsigned char Count up for each communication CMD Command 0x03 Result of measurement NOISE Result of noise environment 0 nomal environment 1 noise environment GEST Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Push 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve 9 Counter Clockwise(fast) 11 Reserve 9 Counter Clockwise(fast) 10 Counter Clockwise(fast) 11 Reserve 9 Counter Clockwise(fast) 10 Counter Clockwise(fast) 11 Reserve 9 Counter Clockwise(fast) 12 Value of R_CAPPOS_GetAve unsigned short Reference value Constant value PC → RX Value of R_CAPPOS_GetAve unsigned short Reference value <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>									
OMD Command 0x03 Result of measurement NOISE Result of noise environment 0 normal environment 1 noise environment GEST Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Push 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve X, Y, Z Result of 3D position calucuration asigned short 3D coordinate value 0x3FFF No detection TOP, BTM, RGT, LFT Value of R_CAPPOS_GetAve unsigned short Reserve sug Value of R_CAPPOS_GetAve unsigned short Reference value PC → RX Value of R_CAPPOS_GetAve breve 1 2 value of R_CAPPOS_GetAve Counter Clockwise(Inclustrian) twig with D_D_D_D_D_D_D_D_D_D_D_D_D_D_D_D_D_D_D_									
Ox03 Result of measurement NOISE Result of noise environment 0 normal environment 1 Result of gesture recognition 0 No detection 1 Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Push 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve 8 Reserve 9 Counter Clockwise(fast) 11 Reserve X, Y, Z Result of 3D position calucuration signed short 3D coordinate value 0x3FFF No detection TOP, R JTM, RGT, LFT Value of R_CAPPOS_GetAve unsigned short Delimiter 0x0a Constant value PC RX Preserve http: 1 2 3 4 unsigned short Constant value PC RX Header 0x05 Constant value			n						
NOISE Result of noise environment 0 normal environment 1 noise environment GEST Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Push 6 Clockwise(slow) 7 Counter Clockwise(slow) 8 Reserve 9 Counter Clockwise(slow) 10 Counter Clockwise(slow) 11 Reserve X, Y, Z Result of 3D position calucuration signed short 3D coordinate value No detectiton TOP, BTM, RGT, LFT Value of R, CAPPOS GetAve unsigned short Reference value END Constant value PC → RX ^{byde} 1 Vat Leader vat Interve Unsigned short Reference value									
0normal environment noise environment1noise environment noise environment0Result of gesture recognition0No detection1Right swipe2Left swipe3Front swipe4Back swipe5Olockwise(slow)6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(fast)11Reserve9Counter Clockwise(fast)12Reserve9Counter Clockwise(fast)13Reserve9Counter Clockwise(fast)14Reserve15Ditter Clockwise(fast)16Counter Clockwise(fast)17Calle Clockwise(fast)18Co									
1noise environmentGESTResult of gesture recognition0No detection1Right swipe2Lisf swipe3Front swipe4Back swipe5Push6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(fast)10Counter Clockwise(fast)11Reserve9Counter Clockwise(fast)10Counter Clockwise(fast)11Reserve8Ocounter Clockwise(fast)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueENDDelimiter0xGaConstant valuePC \rightarrow RXValue of SCAPPOS_GetReftheta I2theta I2theta ICountconstant valuePC \rightarrow RXtheta IDtheta ICounttheta IIDunsigned charCount up for each communication			t						
GEST Result of gesture recognition 0 No detection 1 Right swipe 2 Left swipe 3 Front swipe 4 Back swipe 5 Push 6 Clockwise(slow) 7 Counter Clockwise(fast) 8 Reserve 9 Counter Clockwise(fast) 11 Reserve 9 Counter Clockwise(fast) 11 Reserve X. Y. Z Result of 3D position calucuration signed short 3D coordinate value 0x3FFF No detection TOP, BTM, RGT, LFT Value of R_CAPPOS_GetAve unsigned short Reference value END Delimiter 0x0a Constant value PC → RX Value of R_CAPPOS_GetRef val vale val vale Value Onstant value PC → RX Constant value PC ID unsigned char Count up for each communication									
0No detection1Right swipe2Left swipe3Front swipe3Front swipe4Back swipe5Push6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position caluourationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueENDDelimiter0x0aConstant valuePC \rightarrow RXbyte $\frac{1}{pEAD}$ byte $\frac{1}{pEAD}$ ValueConstant valueDIDunsigned charConstant valueDIDunsigned charConstant value	•								
1Right swipe2Left swipe3Front swipe4Back swipe5Push6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position caluourationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReserve valueConstant valueConstant valuePC \rightarrow RXConstant valuePC \rightarrow RXLeaderbyte $1 = 2 - 3 - 4$ val $y = per A = 0$ DConstant valueDC \rightarrow RXConstant valueDDDunsigned charConstant valueDIDunsigned charConstant valuePCRAHEADHeader0x55Constant valueDDIDunsigned charCount up for each communication			on						
2Left swipe3Front swipe4Back swipe5Push6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(fast)10Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, R STM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueDo detectionDelimitervalue of n_CAPPOS_GetRefunsigned shortConstant value0x0aConstant valuePC \rightarrow RXbyte $\frac{1}{\mu EAD}$ HeadervalConstant valuePC \rightarrow DDunsigned charConstant valueDIDunsigned charCount up for each communication	-								
3Front swipe4Back swipe5Push6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueTOP_R, BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetAveunsigned shortReference valueFNDDelimiter0x0aConstant valuePC \rightarrow RXKeference valueHEADHeader0x55Constant valueIDIDunsigned charCountury for each communication		0 1							
4Back swipe5Push5Push6Clockwise(slow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueTOP,R, BTM,R, RGTR, LFT,RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter0x0aConstant valuePC \rightarrow RXValueHEADHeader0x55Constant valueIDIDunsigned charCount up for each communication									
5Push6Clockwise(sow)7Counter Clockwise(fast)8Reserve9Counter Clockwise(fast)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortmeasurement valueTOP, RTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueDolimiterConstant valuePC \rightarrow RXLbyte124HEADID0IDIDIDunsigned charConstant valueIDConstant valueIDIDUnsigned charConstant value		•							
6Clockwise(slow) Counter Clockwise(fast)7Counter Clockwise(fast)8Reserve9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectitonTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReserve talue0x0aConstant valuePC \rightarrow RXbyte12A4LADHeader0x55Constant valueIDIDunsigned charIDunsigned charCount at value									
7Counter Clockwise(fast)8Reserve9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortmeasurement valueTOP,R, BTM, RGT, R, LFT,RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter0x0aConstant valuePC \rightarrow RXHeaderHEADHeader0x55Constant valueIDIDunsigned charCount value									
8Reserve9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueTOP,R, BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter0x0aConstant valuePC \rightarrow RXbyte $1 \ge 3 4$ valtheADHeader0x55Constant valueIDIDunsigned charCount up for each communication									
9Counter Clockwise(slow)10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortReference valueTOP,R, BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter0x0aConstant valuePC \rightarrow RXbyte $1 \ 2 \ 3 \ 4$ walHEADIHEADHeader0x55Constant valueIDIDunsigned charCount up for each communication									
10Counter Clockwise(fast)11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value $0x3FFF$ No detectionTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortmeasurement valueTOP, R, BTM_R, RGT, LFT, RValue of R_CAPPOS_GetRefunsigned shortReference valueDelimiterDelimiter0x0aConstant valuePC \rightarrow RXHeadertheADHeader0x55Constant valueIDIDunsigned charGount up for each communication									
11ReserveX, Y, ZResult of 3D position calucurationsigned short3D coordinate value $0x3FF$ No detectitonTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortmeasurement valueTOP_R, BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter $0x0a$ Constant valuePC $\rightarrow RX$ byte123valHeader $0x55$ Constant valueIDIDunsigned charGount up for each communication									
X, Y, ZResult of 3D position calucurationsigned short3D coordinate value0x3FFFNo detectitonTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortmeasurement valueTOP,R, BTM, RGT, R, LFT,RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter0x0aConstant valuePC \rightarrow RXbyte123valHeader0x55Constant valueIDIDunsigned charCount up for each communication									
signed short3D coordinate value $0x3FFF$ No detectitonTOP, BTM, RGT, LFTValue of R_CAPPOS_GetAveunsigned shortmeasurement valueTOP, R BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter $0x0a$ Constant valuePC \rightarrow RX $\frac{1}{14EAD}$ byte $1 2 3 4$ valHEADHeader $0x55$ Constant valueIDIDunsigned charCount up for each communication									
0x3FFF No detectiton TOP, BTM, RGT, LFT Value of R_CAPPOS_GetAve unsigned short measurement value TOP,R, BTM,R, RGT,R, LFT_R Value of R_CAPPOS_GetRef unsigned short Reference value END Delimiter 0x0a Constant value PC → RX PC byte 1 2 3 HEAD ID Constant value ID ID ID unsigned char Count up for each communication		-	uration						
TOP, BTM, RGT, LFT Value of R_CAPPOS_GetAve unsigned short measurement value TOP, R, BTM_R, RGT_R, LFT_R Value of R_CAPPOS_GetRef unsigned short Reference value END Delimiter 0x0a Constant value PC → RX HEAD ID CMD byte 1 2 3 4 val HEAD ID Constant value ID ID Constant value ID unsigned char Count up for each communication Count up for each communication									
unsigned shortmeasurement valueTOP_R, BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter $0x0a$ Constant valuePC $\rightarrow RX$ Valuebyte123 $HEAD$ DComstant valueHeader $0x55$ Constant valueIDIDConstant valueIDIDConstant value									
TOP_R, BTM_R, RGT_R, LFT_RValue of R_CAPPOS_GetRefunsigned shortReference valueENDDelimiter $0x0a$ Constant valuePC \rightarrow RXbyte123 $HEAD$ IDCMDENDHeader $0x55$ Constant valueIDIDIDunsigned charCount up for each communication									
unsigned shortReference valueENDDelimiter $0x0a$ Constant valuePC \rightarrow RXbyte123HEADIDCMDIDConstant valueIDIDunsigned charCount up for each communication									
ENDDelimiter $0x0a$ Constant value $PC \rightarrow RX$ byte123 $HEAD$ IDCMDCMDENDHeader $0x55$ Constant valueIDIDunsigned charCount up for each communication									
OxOa Constant value PC \rightarrow RX byte 1 2 3 4 val HEAD ID CMD END Header 0x55 Constant value ID ID Constant value unsigned char Count up for each communication									
PC \rightarrow RX byte 1 2 3 4 val HEAD ID CMD END Header 0x55 Constant value ID ID ID unsigned char Count up for each communication									
byte 1 2 3 4 val HEAD ID CMD END Header 0x55 Constant value ID ID ID unsigned char Count up for each communication	uxua	Constant Value		1					
0x55 Constant value ID ID unsigned char Count up for each communication	byte <u>1 2 3 4</u>								
0x55 Constant value ID ID unsigned char Count up for each communication	HEAD	Header							
ID I									
unsigned char Count up for each communication									
			n						
	CMD	Command		1					
0x01 Start communication									
0x02 Stop communication									
END Delimiter									
0x0a Constant value									

Figure 7.1 Communication Format

Website and Support

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

Inquiries

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

All trademarks and registered trademarks are the property of their respective owners.



Revision History

			Description	
	Rev.	Date	Page	Summary
_	1.00	Jan 16, 2018	-	First edition issued
_	2.00	Fer 28, 2018	-	full-fledged revisions
_	2.10	Mar 30, 2018	14, 15, 16	Change the setting value of Table 5.5

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.
- 2. Processing at Power-on

The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not
 access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
- 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.
- 5. Differences between Products

Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

— The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

- 1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others,
- 4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems: surgical implantations: etc.), or may cause serious property damage (space system: undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet. user's manual or other Renesas Electronics document

- 6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you
- 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics

(Rev.4.0-1 November 2017)

RENESAS

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.

SALES OFFICES

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A. Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited 9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3 Tel: +1-905-237-2004 Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH Arcadiastrasse 10, 40472 Düsseldorf, Germany Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd. Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd. Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China Tel: +86-21-2226-0888, Fax: +86-21-2226-0999 Renesas Electronics Hong Kong Limited entury Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Unit 1601-1611, 16/F., Tower 2, Grand Cent Tel: +852-2265-6688, Fax: +852 2886-9022 Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan Tel: +886-2-8175-9600, Fax: +886 2-8175-9670 Renesas Electronics Singapore Pte. Ltd. 80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949 Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd. Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd. No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd. 17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea Tel: +82-2-558-3737, Fax: +82-2-558-5338