# RA2E1 Group

## RA2E1 Sensor & Touchless key Demo Sample Software

### Introduction

This application note explains demo software for RA2E1 Sensor & Touchless key demo.

### Target Device

RA2E1 Group

### Related Document

(1) RA2E1 Group Sensor & Touchless key Demo Board (r12an0113ej0100-ra2e1)

(2) HS300x Datasheet (https://www.renesas.com/us/en/document/dst/hs300x-datasheet)

(3) ZMOD4410 Programming Manual (https://www.renesas.com/us/en/document/mas/zmod4410-programming-manual-read-me?language=en)

## Contents

## 1. Outline

This software performs LED, buzzer, and UART output (UART-USB conversion) according to the result of contactless button (touchless key) operation by the capacitive touch sensor and sensor control using I2C communication. More detail of this board, please refer RA2E1 Group Sensor & Touchless Key Demo Board (R12AN0113EJ0100).

Figure 1.1 shows system configuration.

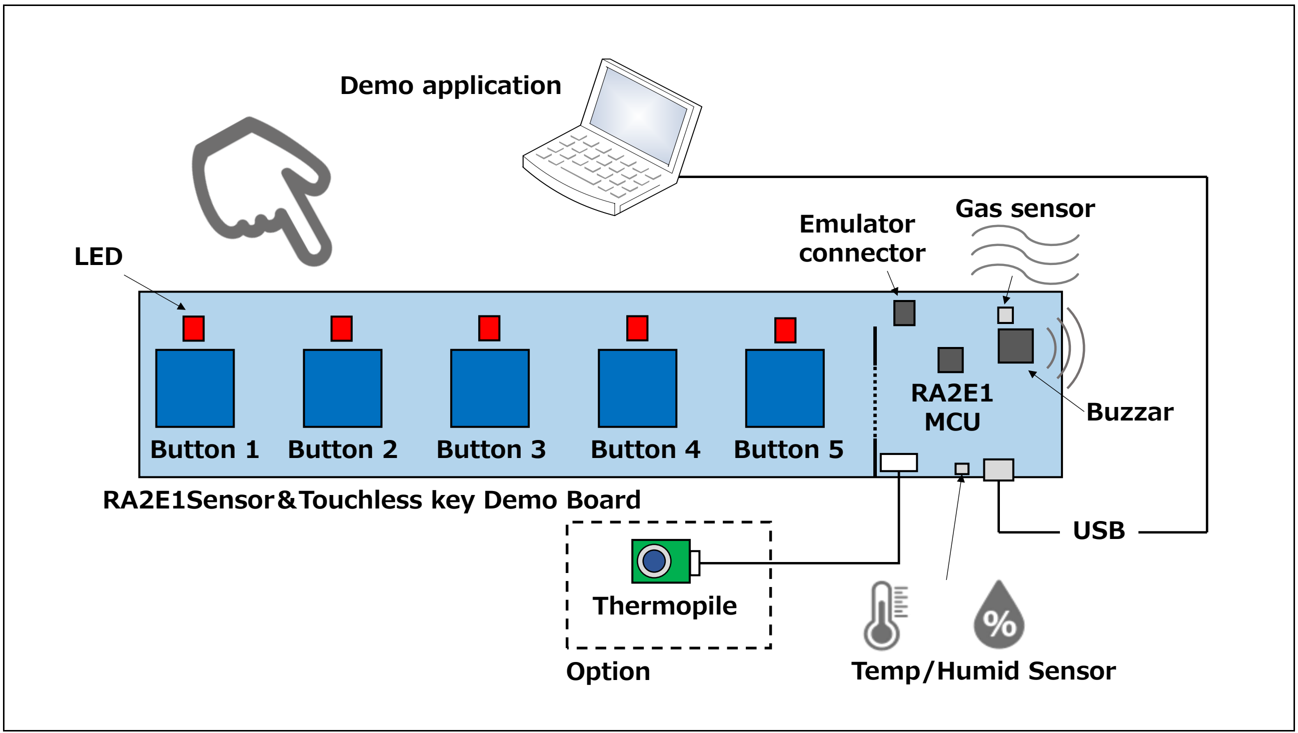



**Figure 1.1 System configuration**

## 2. Confirmed Operation Environment

Table 2.1 shows confirmed operation environment for this software.

**Table 2.1 Confirmed operation environment**

| Item | Description |
|---|---|
| Demo board | RTK0EA0005D00001BJ |
| MCU | RA2E1 |
| Operating Frequency | 48MHz |
| Operating Voltage | 5V |
| Integrated Development Environment | $e^2$ Studio 2021-01 |
| C Compiler | GCC 9.2.1 |
| FSP | 2.3.0 |

# 3. Software Functions

Functions of this software are listed below.

(1) Contactless button (touchless key) operation by the capacitive touch sensor

(2) Temperature / Relative humidity measurement by HS3001

(3) Indoor Air Quality (IAQ) measurement by ZMOD4410

(4) Temperature measurement by Thermopile Sensor

## 3.1 Contactless button (Touchless key) operation by capacitive touch sensor

By Capacitive Touch Sensing Unit (CTSU2) in RA2E1 MCU.

Bring your finger about 15mm above the button and it will be judged ON and buzzer will be output.

LED turns on according to the mode.

## 3.1 Temperature / Relative humidity measurement by HS3001

By build in I2C in MCU.

Measures temperature and humidity to first decimal place and sends it by UART.

## 3.2 Indoor Air Quality (IAQ) measurement by ZMOD4410

By build in I2C in MCU.

Measures Indoor air quality (IAQ) to first decimal place and sends it by UART.

## 3.3 Temperature measurement by Thermopile sensor

By build in I2C in MCU.

The surface temperature of the target object is measured to the first decimal place in measurement area of 16(4x4) elements and sends it by UART.

# 4. Software functions

## 4.1 Software structure

Figure 4.1 shows the software structure.

Use RA smart configurator to add following FSP modules and create an application.

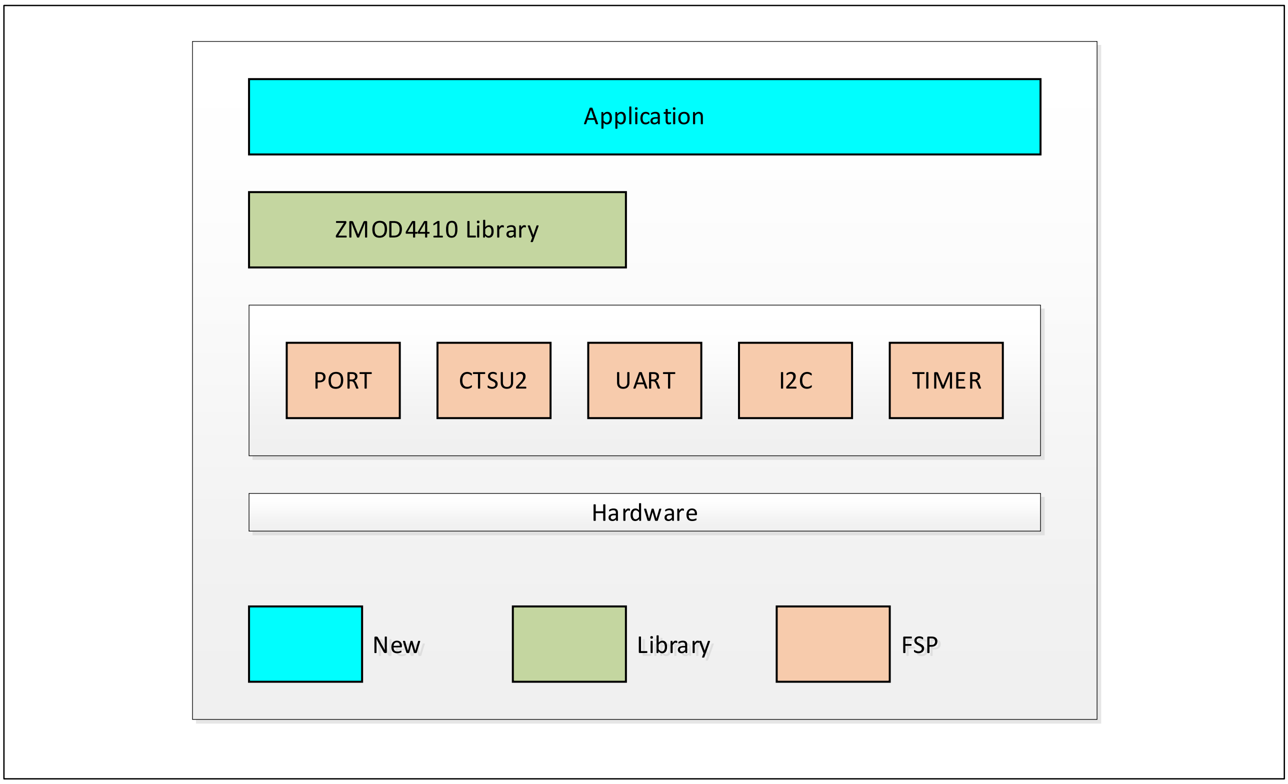
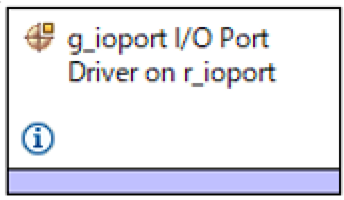
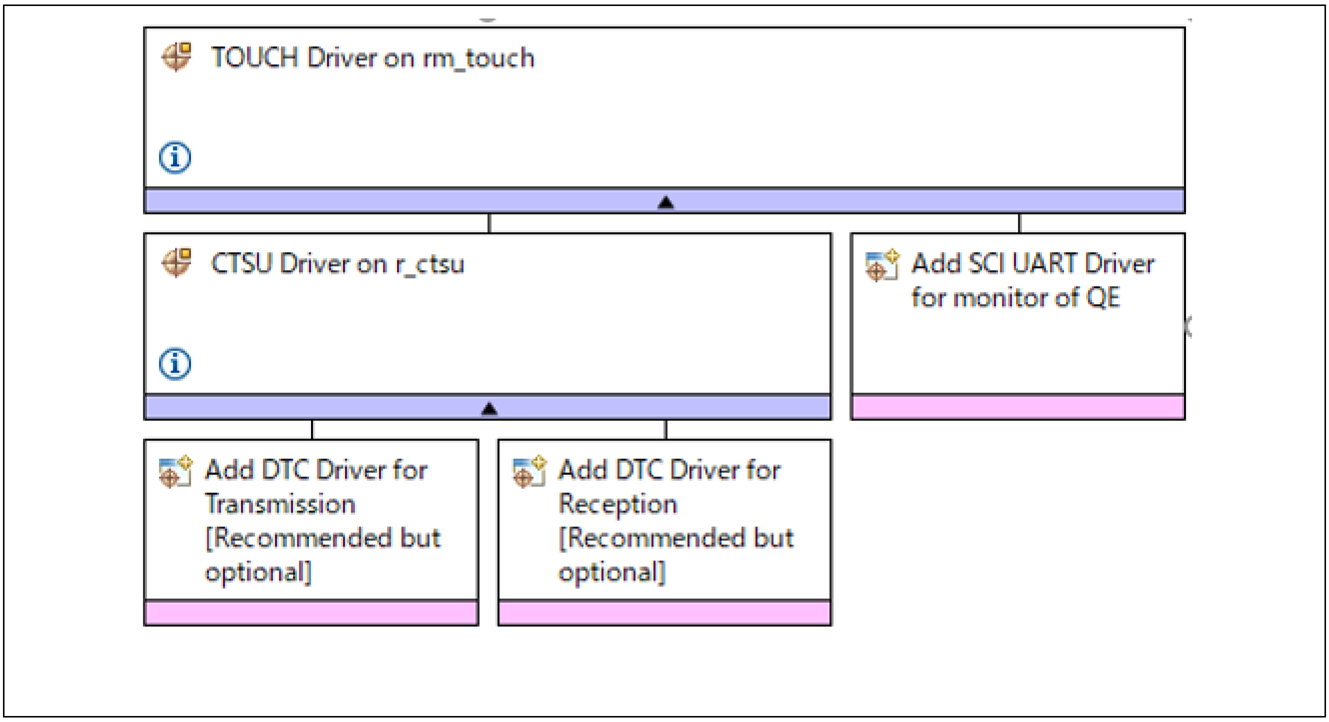



**Figure 4.1 Software structure**

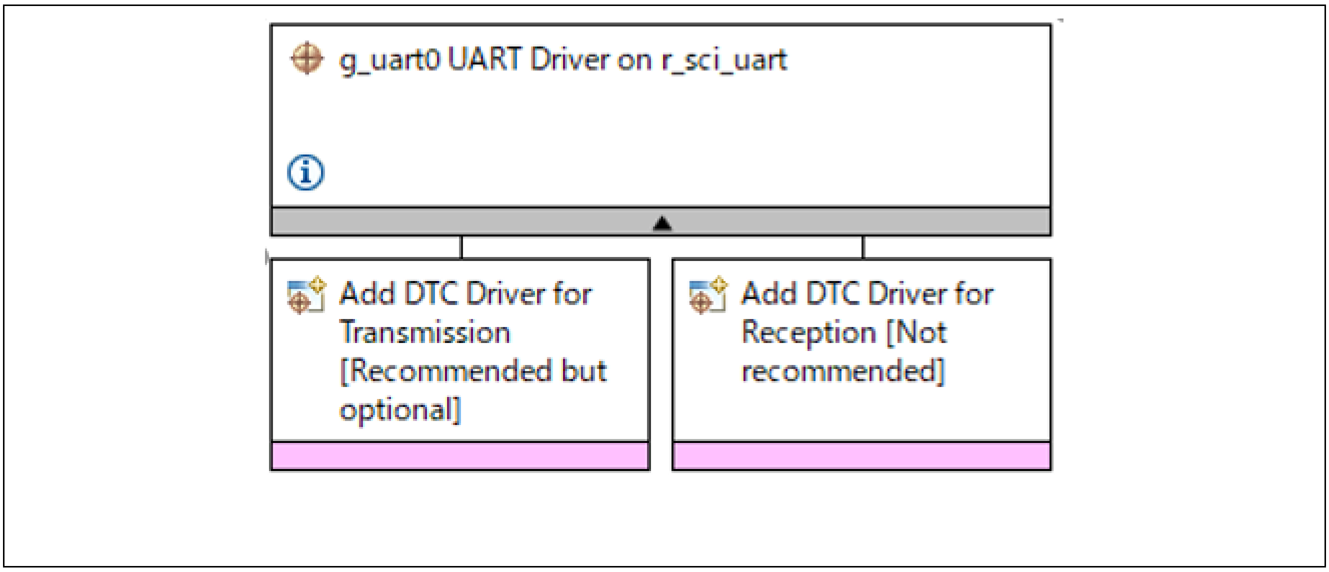
PORT (r_ioport)



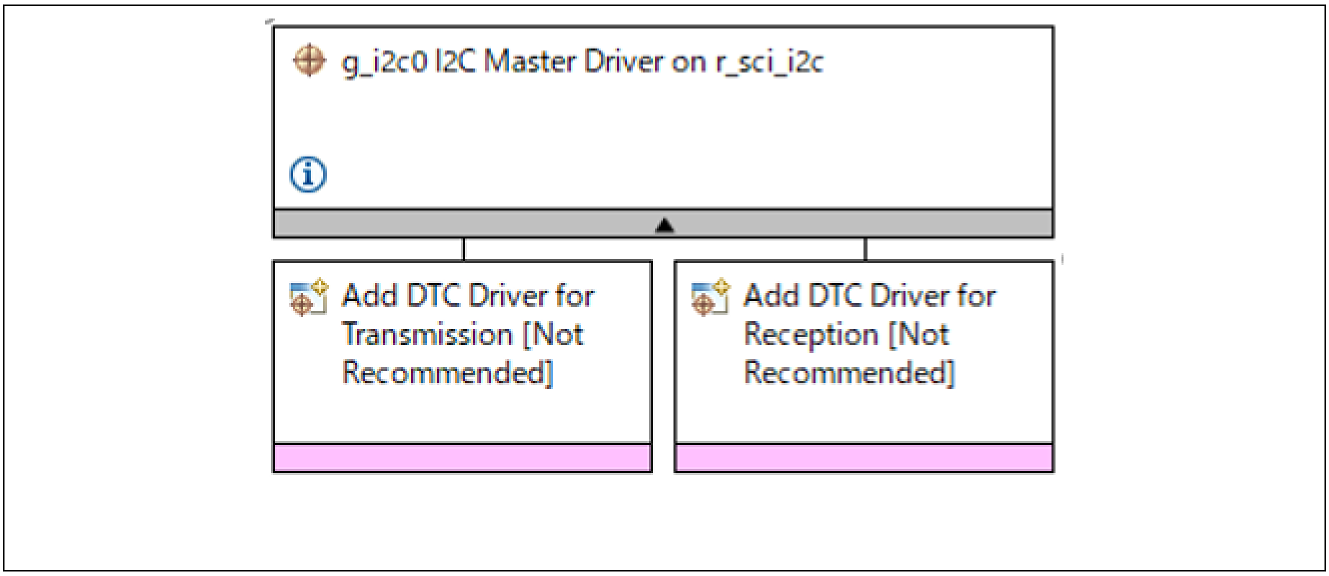
CTSU2 (rm_touch)



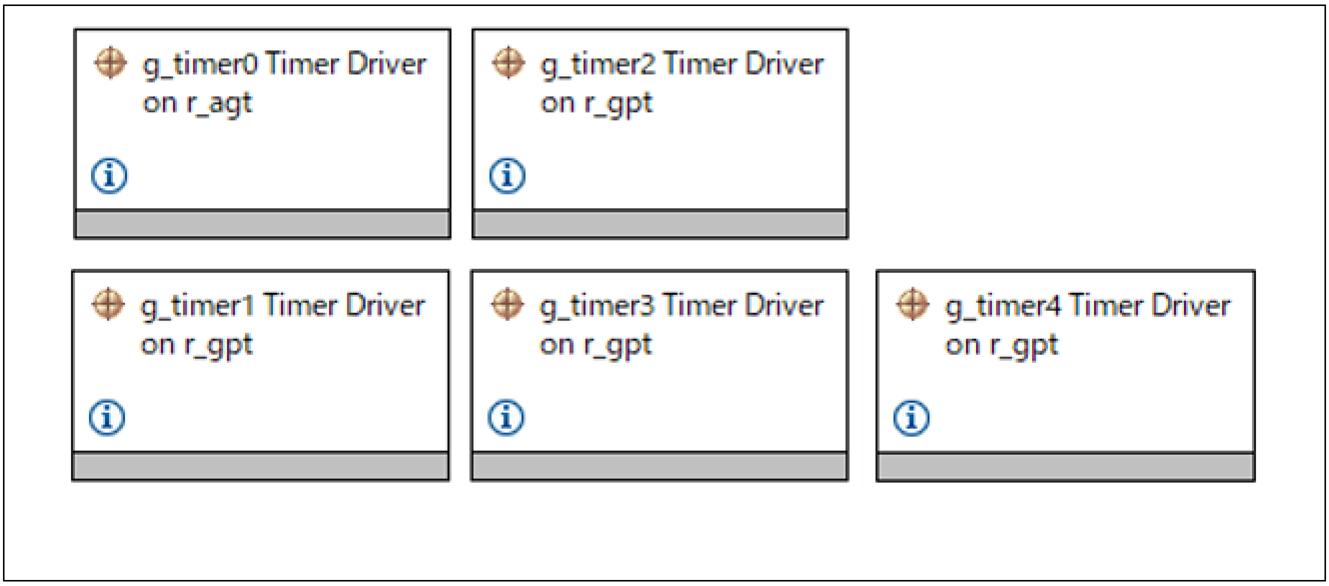

UART (r_sci_uart)



I2C (r_sci_i2c)




TIMER (r_agt, r_gpt)



## 4.2 File structure

Figure 4.1 shows source file tree.

FSP files and libraries are omitted.

**Figure 4.1 Source file tree**

```
├─ src
│    ├─ demo
│    │     │    touch_main.c ・・・・・・・・・・Main Source file
│    │     │    touch_main.h ・・・・・・・・・・Main Include file
│    │     ├─ I2C
│    │     │         r_i2c.c     ・・・・・・・・・・I2C Source file
│    │     │         r_i2c.h     ・・・・・・・・・・I2C Header file
│    │     ├─ SENSOR
│    │     │         r_sensor.c   ・・・・・・・・・Sensor Source file
│    │     │         r_sensor.h ・・・・・・・・・Sensor Header file
│    │     │         ├─ HS3001
│    │     │         │      r_hs3001.c   ・・・・HS3001 Source file
│    │     │         │      r_hs3001.h   ・・・・HS3001 Header file
│    │     │         ├─ Thermo
│    │     │         │      r_thermo.c   ・・・・Thermo Sensor Source file
│    │     │         │      r_thermo.h   ・・・・Thermo Sensor Header file
│    │     │         ├─ ZMOD4410
│    │     │         │      r_zmod4410.c   ・・・ZMOD4410 Source file
│    │     │         │      r_zmod4410.h   ・・・ZMOD4410 Header file
│    │     │         │      ├─ Library ・・・・ZMOD4410 library folder
                                (The folder is empty.  Copy the files by referring to chapter 4.3.)
```

Table 4.1 shows source files.

**Table 4.1 Source Files**

| File name | Description |
|---|---|
| touch_main.c | Main Source file |
| r_i2c.c | I2C Source file |
| r_sensor.c | Sensor Main Source file |
| r_hs3001.c | HS3001 Source file |
| r_thermo.c | Thermo Sensor Source file |
| r_zmod4410.c | ZMOD4410 Source file |

Table 4.2 shows header files.

**Table 4.2 Header files**

| File name | Description |
|---|---|
| voice_main.h | Main Header file |
| r_i2c.h | I2C Header file |
| r_sensor.h | Sensor Main Header file |
| r_hs3001.h | HS3001 Header file |
| r_thermo.h | Thermo Sensor Header file |
| r_zmod4410.h | ZMOD4410 Header file |

## 4.3 How to get the ZMOD4410 library

Download the library for the ZMOD4410 from the following link.

https://www.renesas.com/us/en/products/sensor-products/gas-sensors/zmod4410-indoor-air-quality-sensor-platform

The library to download is the following.

ZMOD4410 - 2nd Gen - Air Quality & eCO2 Firmware - Recommended for New Designs

Unzip the downloaded library and copy the following files to the Library folder.

**Table 4.3 Library files**

| Folder name | File name |
|---|---|
| REN_ZMOD4410-AirQuality-eCO2-FW-2nd-Gen-2p1p2_SWR_20201019<br>\Renesas_ZMOD4410_IAQ_2nd_Gen_Example_2.1.2<br>\Renesas_ZMOD4410_IAQ_2nd_Gen_Example<br>\ZMOD4410_Firmware\gas-algorithm-libraries<br>\iaq_2nd_gen\Arm Cortex-M\M23\arm-none-eabi-gcc\ | iaq_2nd_gen.h<br>lib_iaq_2nd_gen.a<br>lib_zmod4xxx_cleaning.a<br>zmod4xxx_cleaning.h |
| REN_ZMOD4410-AirQuality-eCO2-FW-2nd-Gen-2p1p2_SWR_20201019<br>\Renesas_ZMOD4410_IAQ_2nd_Gen_Example_2.1.2<br>\Renesas_ZMOD4410_IAQ_2nd_Gen_Example<br>\ZMOD4410_Firmware\zmod4xxx_example\src\ | zmod4xxx.c<br>zmod4xxx.h<br>zmod4xxx_types.h<br>zmod4410_config_iaq2.h |

## 4.4 Constants

Table 4.4 shows the list of constants.

**Table 4.4 Constants**

| Constants name | Setting | Description |
|---|---|---|
| VD_PRV_INFINITE_LOOP | while(1) | Error Loop |
| VD_PRV_STS_ERR | (-1) | Error code |
| VD_PRV_BTN_MD_OFF | (0x0001) | Run mode Off |
| VD_PRV_BTN_MD_LOW | (0x0002) | Run mode Low |
| VD_PRV_BTN_MD_MID | (0x0003) | Run mode Middle |
| VD_PRV_BTN_MD_HI | (0x0004) | Run mode Hi |
| VD_PRV_BTN_MD_AUTO | (0x0005) | Run mode Auto |
| VD_PRV_AUTO_MD_0 | (0x0010) | Run mode Auto_0 |
| VD_PRV_AUTO_MD_1 | (0x0020) | Run mode Auto_1 |
| VD_PRV_AUTO_MD_2 | (0x0030) | Run mode Auto_2 |
| VD_PRV_AUTO_MD_3 | (0x0040) | Run mode Auto_3 |
| VD_PRV_AUTO_MD_4 | (0x0050) | Run mode Auto_4 |
| LED | | |
| VD_PRV_LED_ON | (0) | LED On |
| VD_PRV_LED_OFF | (1) | LED Off |
| VD_PRV_LED_POW | (0x0001) | LED Pow bit |
| VD_PRV_LED_LOW | (0x0002) | LED Low bit |
| VD_PRV_LED_MID | (0x0004) | LED Middle bit |
| VD_PRV_LED_HI | (0x0008) | LED Hi bit |
| VD_PRV_LED_AUTO | (0x0010) | LED Auto bit |
| VD_PRV_LED_MD_OFF | (0) | LED mode off bit |
| VD_PRV_LED_MD_LOW | (0x0003) | LED mode low bit |
| VD_PRV_LED_MD_MID | (0x0005) | LED mode middle bit |
| VD_PRV_LED_MD_HI | (0x0009) | LED mode hi bit |
| VD_PRV_LED_MD_A0 | (0x0011) | LED mode auto0 bit |
| VD_PRV_LED_MD_A1 | (0x0013) | LED mode auto1 bit |
| VD_PRV_LED_MD_A2 | (0x0015) | LED mode auto2 bit |
| VD_PRV_LED_MD_A3 | (0x0019) | LED mode auto3 bit |
| VD_PRV_LED_MD_A4 | (0x0019) | LED mode auto4 bit |
| CTSU | | |
| VD_PRV_BTN_1 | (1) | Button1 bit |
| VD_PRV_BTN_2 | (2) | Button2 bit |
| VD_PRV_BTN_3 | (3) | Button3 bit |
| VD_PRV_BTN_4 | (4) | Button4 bit |
| VD_PRV_BTN_5 | (0) | Button5 bit |
| VD_PRV_BTN_POW | (0x0002) | Button Pow |
| VD_PRV_BTN_LOW | (0x0004) | Button Low |
| VD_PRV_BTN_MID | (0x0008) | Button Middle |
| VD_PRV_BTN_HI | (0x0010) | Button Hi |
| VD_PRV_BTN_AUTO | (0x0001) | Button Auto |
| VD_PRV_BTN_BIT_POW | (0x0001) | Button bit Pow |
| VD_PRV_BTN_BIT_LOW | (0x0002) | Button bit Low |
| VD_PRV_BTN_BIT_MID | (0x0004) | Button bit Middle |
| VD_PRV_BTN_BIT_HI | (0x0008) | Button bit Hi |
| VD_PRV_BTN_BIT_AUTO | (0x0010) | Button bit Auto |
| BUZZER | | |
| VD_PRV_BZ_POW | (22943) | Buzzer Count Pow (1046Hz) |
| VD_PRV_BZ_LOW | (20441) | Buzzer Count Low (1174Hz) |
| VD_PRV_BZ_MID | (18208) | Buzzer Count Middle (1318Hz) |
| VD_PRV_BZ_HI | (17190) | Buzzer Count Hi (1396Hz) |
| VD_PRV_BZ_AUTO | (15314) | Buzzer Count Auto (1567Hz) |
| VD_PRV_BZ_A0 | (22943) | Buzzer Count Auto_0 |

| | | |
|---|---|---|
| VD_PRV_BZ_A1 | (20441) | Buzzer Count Auto_1 |
| VD_PRV_BZ_A2 | (18208) | Buzzer Count Auto_2 |
| VD_PRV_BZ_A3 | (17190) | Buzzer Count Auto_3 |
| VD_PRV_BZ_A4 | (15314) | Buzzer Count Auto_4 |
| VD_PRV_BZ_CNT_1 | (1) | Buzzer Count (N * 100ms) |
| TOOL | | |
| VD_PRV_MODE_RUN | (1) | Send Enable |
| VD_PRV_MODE_STOP | (0) | Send Disable |
| VD_PRV_DATA_HEAD | (0x88) | Data Head Mark |
| VD_PRV_SEND_LEN | (57) | Send length |
| VD_PRV_SEND_TIME | (5) | Send Interval (N * 20ms) |
| UART | | |
| VD_PRV_WAIT | (1) | Wait Send Complete |
| VD_PRV_NO_WAIT | (0) | No Wait Send Complete |
| VD_PRV_KEY_ERR | (0xff) | Error |
| VD_PRV_KEY_CR | (0x0d) | CR ASCII CODE |
| VD_PRV_KEY_LF | (0x0a) | LF ASCII CODE |
| VD_PRV_UART_RX_MAX | (64) | Receive Buffer Size |
| VD_PRV_UART_CMD_MAX | (64) | Command Buffer Size |
| THERMO Sensor | | |
| VD_PRV_THR_R_LEN | (35) | Receive Data Length |
| VD_PRV_THR_DATA_OFS | (2) | Data Offset |
| VD_PRV_THR_DATA_CNT | (16) | Data Count |
| VD_PRV_THR_THR | (400) | Thresh (40.0°C) |
| VD_PRV_THR_LVL1 | (0) | Level1 Count |
| VD_PRV_THR_LVL2 | (2) | Level2 Count |
| VD_PRV_THR_LVL3 | (4) | Level3 Count |
| GAS Sensor | | |
| VD_PRV_IAQ_LVL1 | (1.99f) | Level1 Thresh |
| VD_PRV_IAQ_LVL2 | (2.99f) | Level2 Thresh |
| VD_PRV_IAQ_LVL3 | (3.99f) | Level3 Thresh |
| VD_PRV_IAQ_LVL4 | (4.99f) | Level4 Thresh |

## 4.5 Global Variables

Table 4.5 shows the global variables.

**Table 4.5 Global Variables**

| Variables name | Type | Description |
|---|---|---|
| gs_timer1_cnt | int16_t | Timer1 count |
| gs_timer3_cnt | int16_t | Timer3 count |
| gs_timer4_cnt | int16_t | Timer4 count |
| gs_sts_prev | int16_t | Status previous |
| gs_btn_mode | int16_t | Run mode |
| gs_btn_mode_prev | int16_t | Run mode previous |
| gs_btn_nml_prev | int16_t | Normal mode previous |
| gs_btn_prev | int16_t | Button previous |
| gs_auto_mode_prev | int16_t | Auto mode previous |
| gs_iaq_fdata | float | IAQ float data |
| gs_iaq_idata | int16_t | IAQ int data |
| gs_temp_idata | int16_t | Temperature int data |
| gs_humi_idata | int16_t | Humidity int data |
| gs_btn_data | uint16_t | Button data |
| gs_data_num | int16_t | Send data number |
| Command | | |
| gs_mode_flg | uint16_t | Send mode flag |
| CTSU | | |
| gs_btn_status | uint64_t | Button status |
| gs_btn_dif | uint16_t[] | Button difference data |
| UART | | |
| gs_uart_rx_len | int16_t | Receive length |
| gs_uart_rd_idx | int16_t | Buffer Read index |
| gs_uart_wr_idx | int16_t | Buffer Write index |
| gs_uart_cmd_len | int16_t | Command length |
| gs_uart_rx_buf | uint8_t[] | Receive Buffer |
| gs_uart_cmd_buf | uint8_t[] | Command Buffer |
| gs_send_time | int16_t | Send Interval |
| gs_send_buf | uint8_t[] | Send Buffer |
| gs_thr_data | uint8_t[] | Thermo sensor data Buffer |

## 4.6 Function Specifications

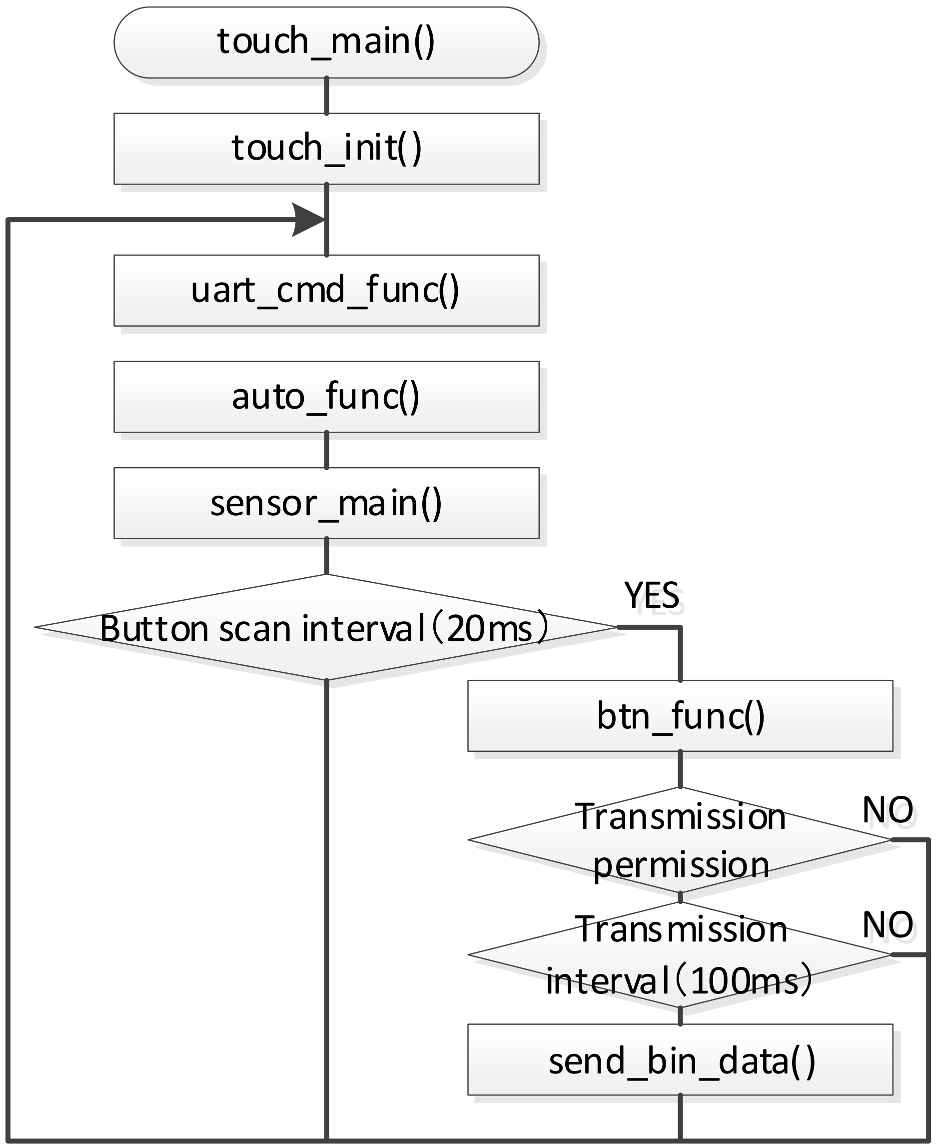
Table 4.6 shows the list of functions.

**Table 4.6 Function list**

| Function name | Description |
|---|---|
| touch_main | Main function |
| touch_init | Initialize |
| touch_loop | Main loop |
| btn_func | Button function |
| btn_func_normal_0 | Normal mode Off function |
| btn_func_normal_L | Normal mode Low function |
| btn_func_normal_M | Normal mode Middle function |
| btn_func_normal_H | Normal mode High function |
| btn_func_auto | Auto mode function |
| btn_bit_set | Button bit set |
| btn_check | Button check |
| normal_mode_func | Normal mode function |
| auto_func | Auto mode function |
| auto_mode0_func | Auto mode0 function |
| auto_mode1_func | Auto mode1 function |
| auto_mode2_func | Auto mode2 function |
| auto_mode3_func | Auto mode3 function |
| auto_mode4_func | Auto mode4 function |
| led_set | LED data set |
| buzzer_start | Buzzer (buzzer wave) start |
| timer_open | Timer initialize |
| timer4_wait | Timer4 (20ms wait timer) start |
| timer3_stop | Timer3 (300ms buzzer timer) stop |
| ctsu_open | CTSU initialize |
| ctsu_getkey | Get key code |
| uart_open | UART initialize |
| uart_cmd_func | UART command function |
| uart_getchr | UART Get Key code |
| uart_putmsg | UART Send message |
| uart_wait_tx_end | UART Wait Send complete |
| cmd_chk | Command check |
| send_bin_data | Send binary data |
| thermo_data_cnt | Thermo sensor data count |
| thermo_data_clr | Thermo sensor data count clear |

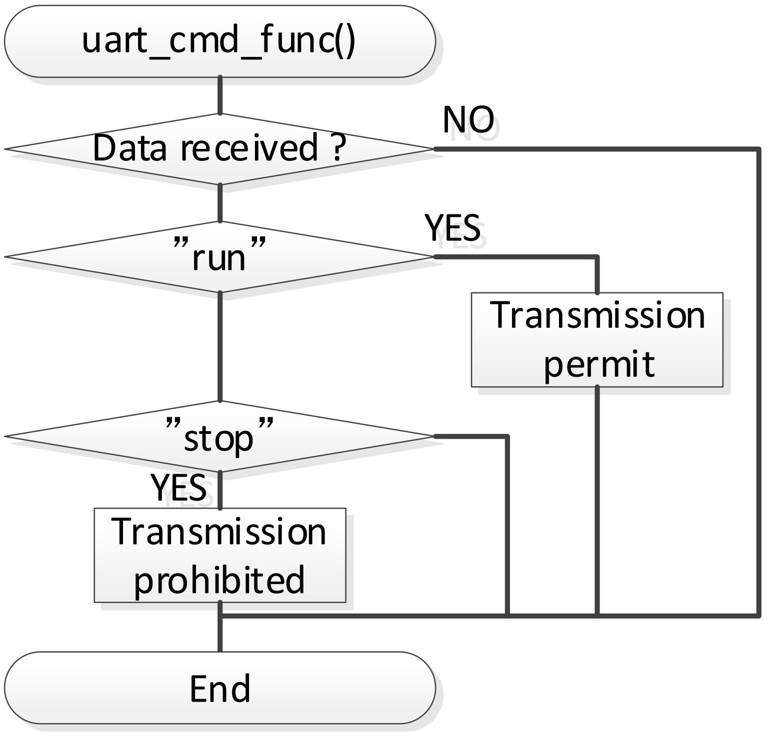
## 4.7 Overall processing Flow

The following describes the overall processing flow.



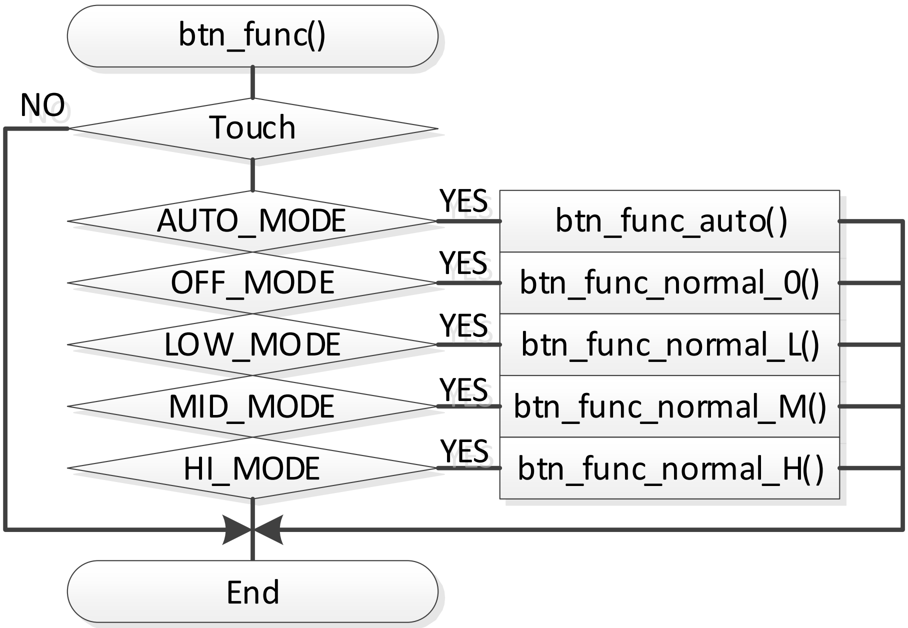
## 4.8 UART command processing Flow

The following shows UART processing flow.



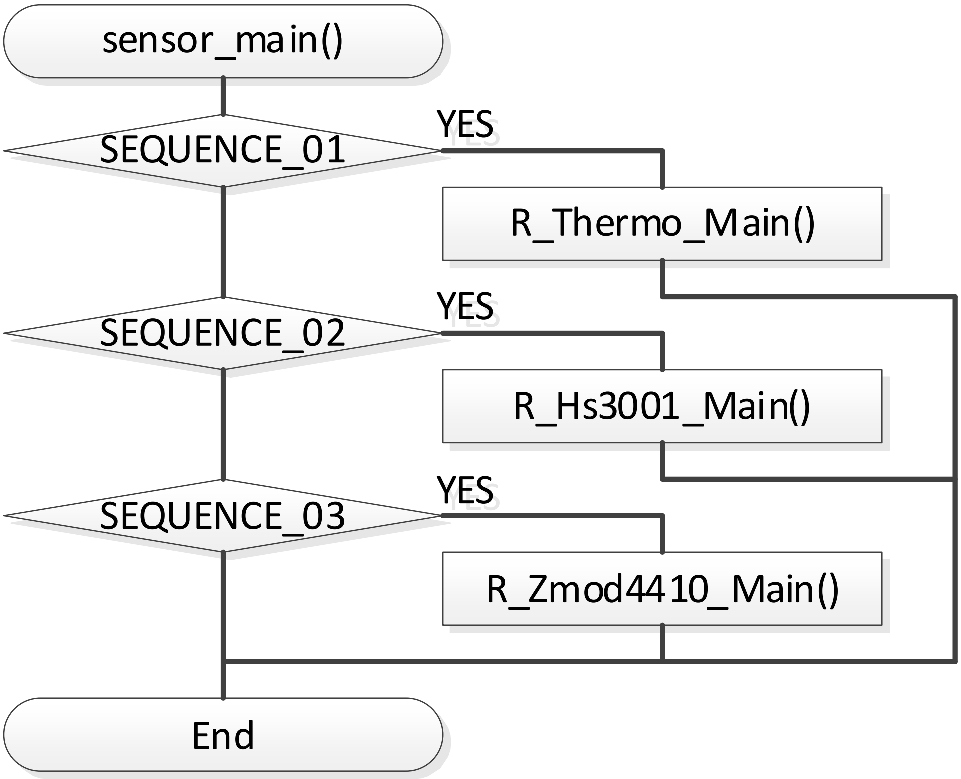
### 4.9 Touchless button operation processing flow

Following shows the touchless button operation processing flow.



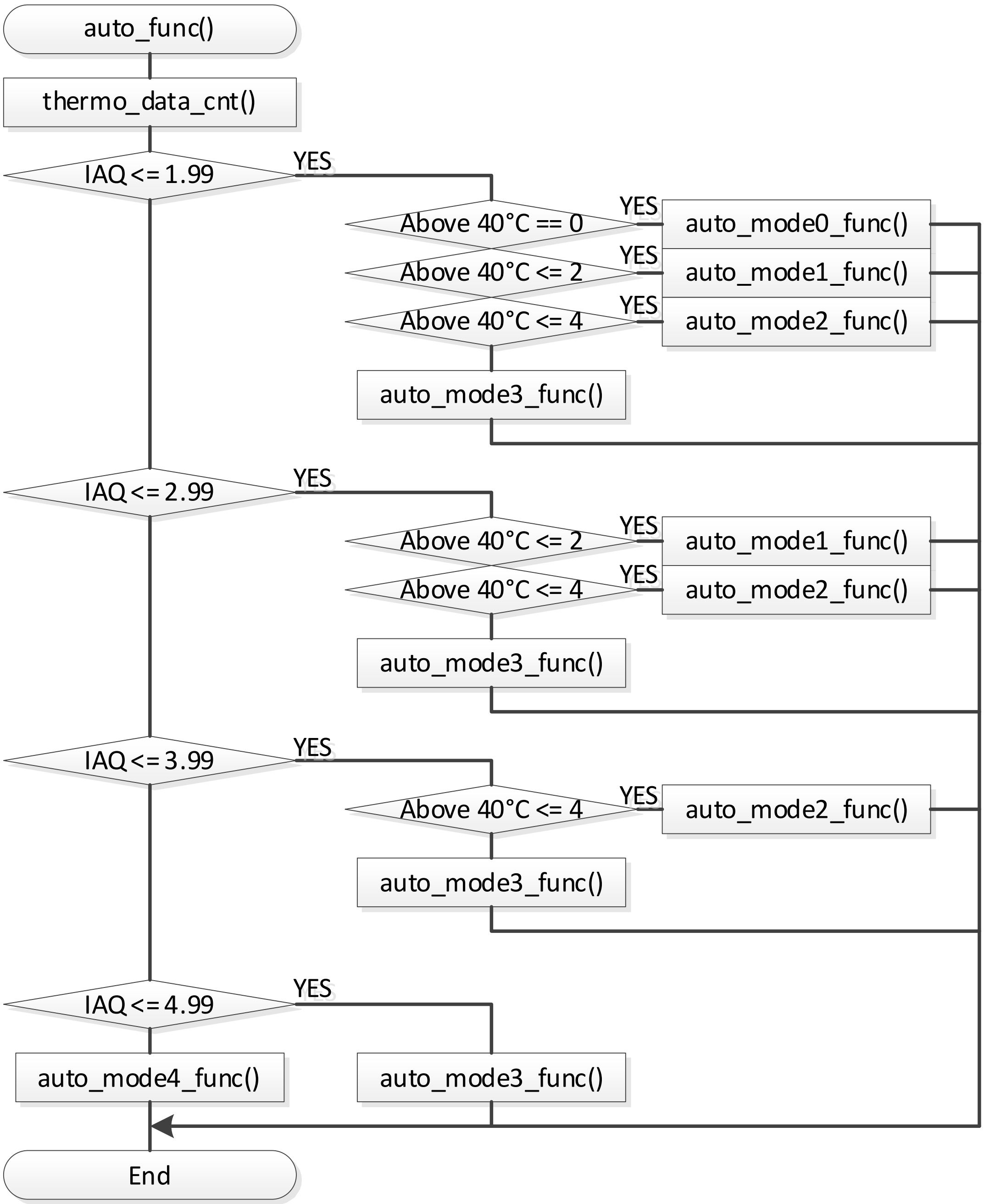
## 4.10 Sensor measurement processing flow

Following shows the sensor measurement processing flow.



## 4.11 AUTO mode processing flow

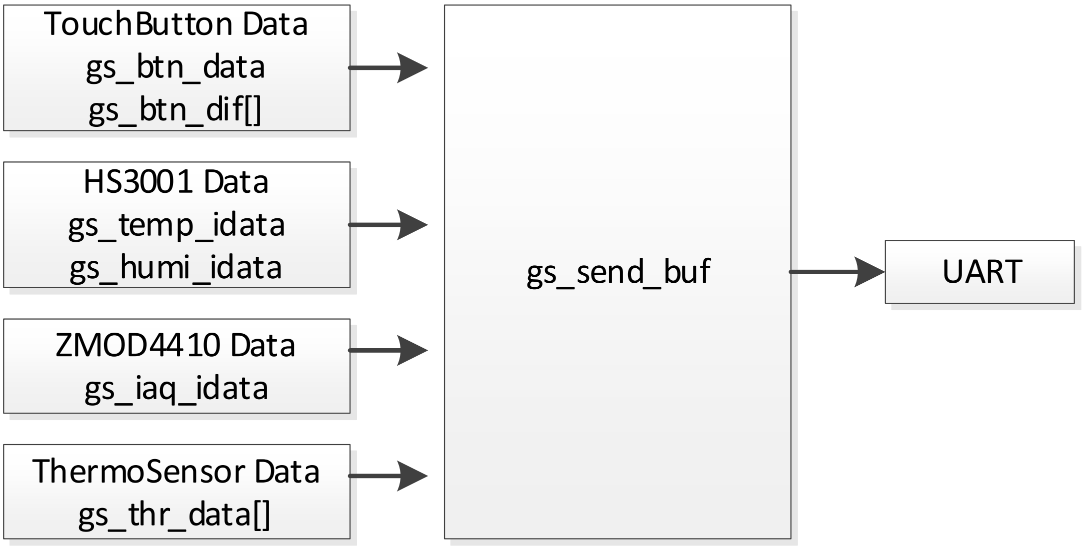
Following shows the AUTO mode processing flow.



## 4.12 Data flow

Following shows the data flow.

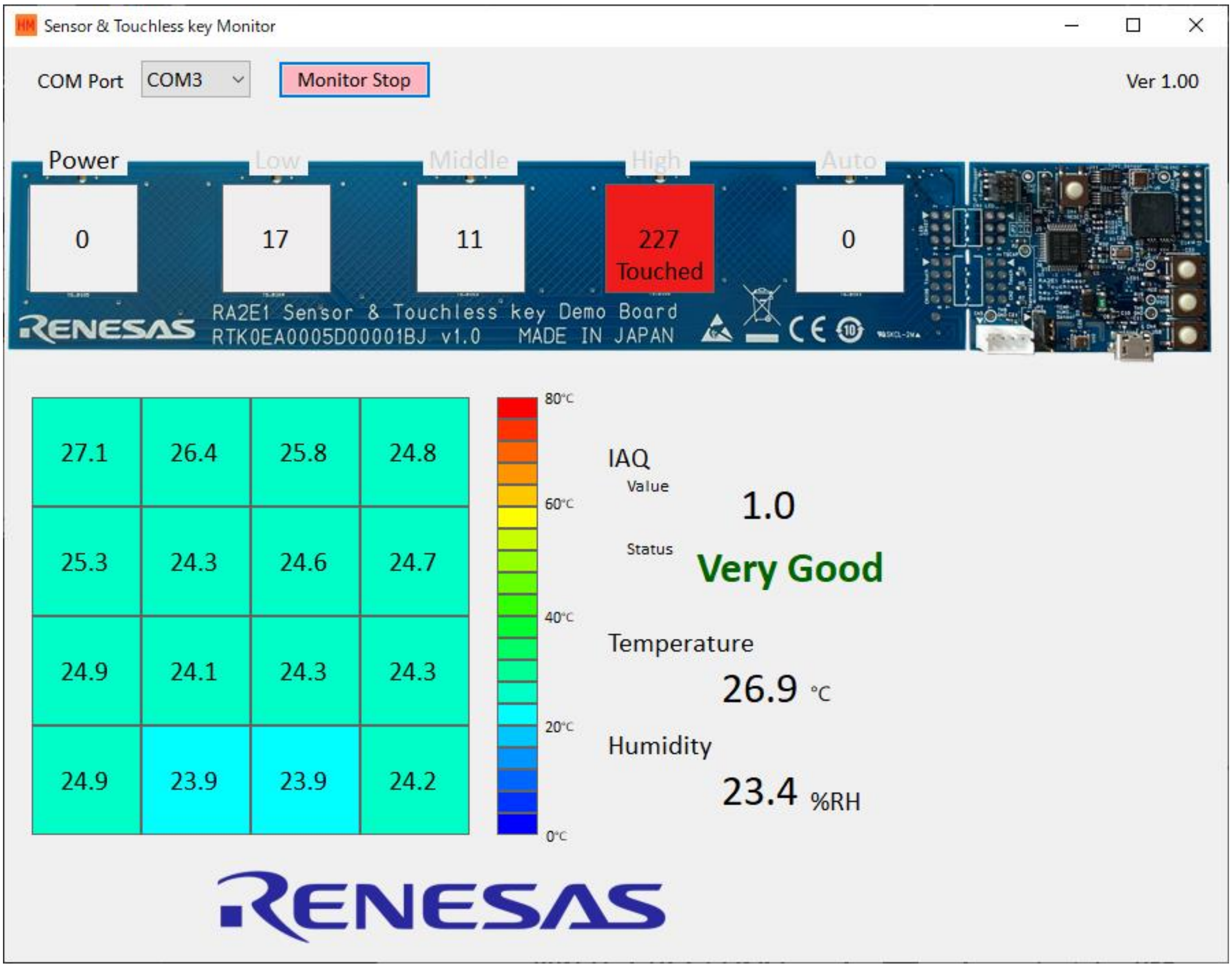
Copies the button data and sensor data to the transmission buffer and send to UART every 100ms.




Transmitted data format is as follows.

| OFFSET | Item | Byte | Description |
|---|---|---|---|
| 0 | Data Head Mark | 1 | Shows data header (0x88) |
| 1 | Data Number | 1 | Loop between 0x00 and 0xFF |
| 2 | Button Status | 1 | Button status; When each button is touched, the following bit is set to 1<br>Bit0:Button1, Bit1: Button2, Bit2: Button3, Bit3: Button4, Bit4: Button5 |
| 3 | Mode | 1 | OFF(0x01), LOW(0x02), MID(0x03), HI(0x04),<br>AUTO_MODE0(0x15), AUTO_MODE1(0x25),<br>AUTO_MODE2(0x35), AUTO_MODE3(0x45),<br>AUTO_MODE4(0x55) |
| 4 | Button1 Data | 2 | DIFF value (0 to 65535) |
| 6 | Button2 Data | 2 | DIFF value (0 to 65535) |
| 8 | Button3 Data | 2 | DIFF value (0 to 65535) |
| 10 | Button4 Data | 2 | DIFF value (0 to 65535) |
| 12 | Button5 Data | 2 | DIFF value (0 to 65535) |
| 14 | Temperature | 2 | Value obtained by multiplying the data with one decimal place by ten<br>(example : when data is 12.3°C value will be 123) |
| 16 | Humidity | 2 | Value obtained by multiplying the data with one decimal place by ten<br>(example : when data is 45.6%, value will be 456) |
| 18 | IAQ | 2 | Value obtained by multiplying the data with one decimal place by ten<br>(example : when data is 12.3, value will be 123) |
| 20 | Dummy | 1 | Dummy |
| 21 | Thermo sensor | 1 | Number of byte of following thermo sensor data(35) |
| 22 | PTAT | 2 | Thermo sensor data<br>Value obtained by multiplying the data with one decimal place by ten<br>(example : when data is 12.3°C, value will be 123) |
| 24 | P00 Data - P15 Data | 32<br>(2 * 16) | Thermo sensor data<br>Value obtained by multiplying the data with one decimal place by ten<br>(example : when data is 12.3°C, value will be 123) |
| 56 | PEC | 1 | Checksum of Thermo sensor data |
| | Total byte | 57 | |

Transmitted data is supposed to be used in application on PC as shown below.




More info and detail of this application, please refer Sensor & Touchless key Demo Evaluation Tool “Sensor & Touchless key Monitor” (R20AN0614EJ0100).

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Feb.11.21 | - | First release |
| | | | |

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

## Corporate Headquarters


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


## Trademarks



## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.