

RX200, RX100 Series

R11AN0139EU0100

Rev 1.00

Getting Started with Capacitive Touch

May 8, 2017

Introduction

Capacitive touch interfaces are becoming an increasingly popular for companies to reduce cost and complexity by exchanging a button with a capacitive touch sensor. It enables more complex eye pleasing designs, without drastically increasing cost like a traditional electromechanical button or switch. However, implementing capacitive touch systems incurs new design challenges. Renesas provides a suite of software to help aid in integrating the solution into a product. The following text provides a general overview of the software suite, to help get an application up and running.

Target Device

RX231, RX230, RX130, RX113 with onboard capacitive touch sensing units.

Related Documents

- [Touch Module Using Firmware Integration Technology \(R01AN3820EU\)](#)
- [CTSU Module Using Firmware Integration Technology \(R01AN3821EU\)](#)
- [Touch Button Module Using Firmware Integration Technology \(R01AN3822EU\)](#)
- [Touch Slider Module Using Firmware Integration Technology \(R01AN3823EU\)](#)
- [Capacitive Touch Layout Guidelines \(R01AN3825EU\)](#)
- [Workbench6 User Manual \(R20UT3842EJ\)](#)
- [Tuning of a RX based Capacitive Touch Project \(R30AN0291EU\)](#)
- [CTSU Hardware Overview \(R01AN3824EU\)](#)
- [CTSU Basis of Cap touch detection \(R30AN0218EJ\)](#)

Contents

1. Overview of the Capacitive Touch Hardware	3
2. Overview of the Capacitive Touch Software Environment	4
2.1 Required Software	4
2.2 Overview of CTSU Firmware	5
3. Renesas Tuning Tools – Workbench6	6
3.1 Outputs from the Tuning Process	6
3.1.1 Creating a New Base Project	7
3.1.2 Adding Capacitive Touch Functionality to an Existing FIT Project.....	8
3.2 Workbench6 and Python	8
4. Connecting to Workbench6	9
4.1 e2Studio through the E1 Debugger.....	9
4.2 Serial Communication Interface (SCI6).....	9
4.3 Serial Over Built-in USB Using CDC (RX113/RX231).....	9
5. Overview of the Generated Software	10
5.1 Overview of the FIT Style Project	10
5.2 Overview of the Touch API Project	11

1. Overview of the Capacitive Touch Hardware

Renesas’ capacitive touch solution takes advantage of interactions a human body has with an electrostatic field generated by the microcontroller unit (MCU) to determine user input. The peripheral, referred to as the capacitive touch sensing unit (CTSUS), measures a finger interacting with the capacitive sensor starts by generating an electrostatic field. Here the peripheral takes ambient measurements via the CTSUS by measuring the parasitic (stray) capacitance between conductors or between two pads. The CTSUS monitors the human interaction and via some specialized circuitry produces digital counts. These digital counts vary depending upon the interactions with the environment; as a user interacts with the sensor, these counts will eventually exceed a threshold, which can be represented in software as a touch. An overview of this process is shown in Figure 1.

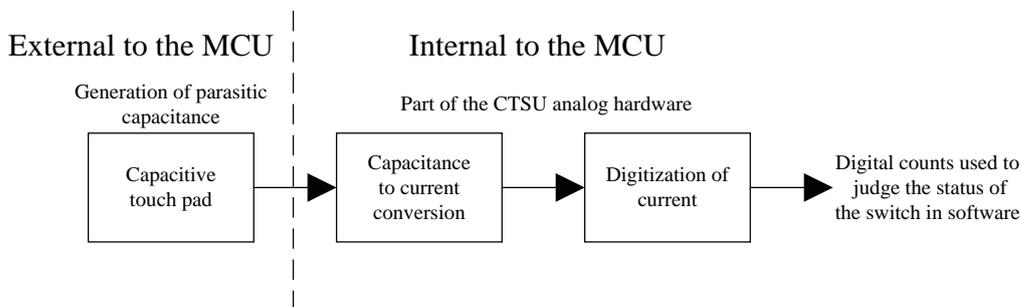


Figure 1: Block diagram describing the high-level overview of the different parts of a capacitive touch system.

The two main methods in which the field is generated is via the:

- Self-Capacitance Method – The electrostatic field is generated between the pad connected to the MCU and its surrounding conductive surfaces. As your finger interacts with the pad, this causes the total capacitance in the system to increase.
- Mutual-Capacitance Method – The electrostatic field is generated between two pads, each connected to a sensing channel. As your finger interacts with the pad in this configuration, it causes the total capacitance in the system to decrease.

In either method, the CTSUS measures the change in capacitance by converting the interaction that the human body has to a current via a switched capacitor filter. This interaction affects the overall capacitance, and the resulting current from this interaction is how the CTSUS determines interaction with the electrode. The overview of the process is shown in Figure 2: Overview of the current to counts conversion process, and is made up of the output from the switched capacitor filter, the power supply circuit, current oscillator and a counter.

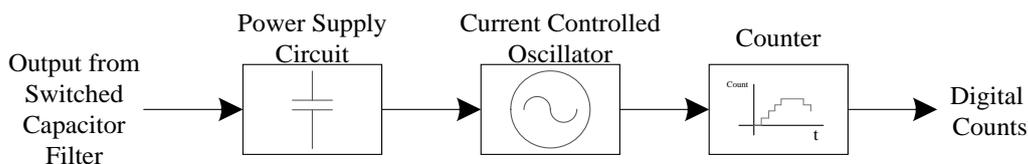


Figure 2: Overview of the current to counts conversion process

The conversion process starts with the power supply circuit taking the output from the switch capacitor filter and low pass filters it with a connected capacitor (referred to as TSCAP). This smoothed current feeds the current oscillator whose frequency is controlled in proportion to the measured current. This oscillation is then passed along to a counter that counts the number of oscillations producing a count. An overview of the results after each stage is shown in Figure 3 for self-capacitance mode, where the two rows of charts demonstrates how the current amplitude affects the digital counting process.

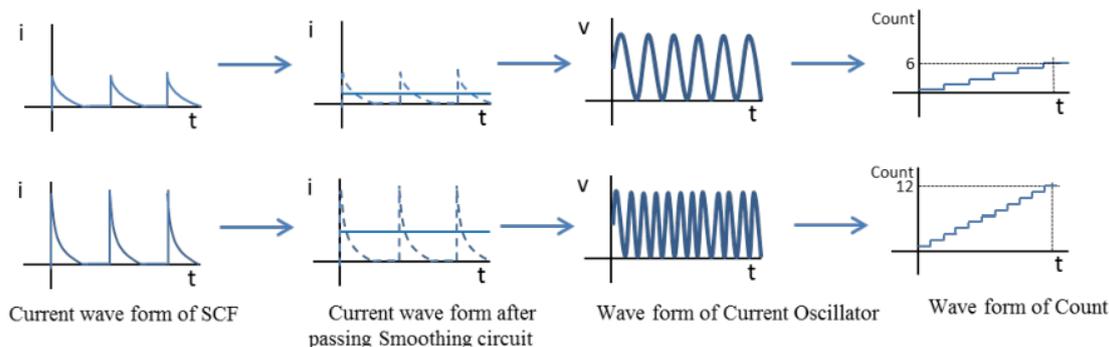


Figure 3: Waveforms describing the current and counts after each part of the conversion process with currents from self-capacitance mode.

This count value is then saved in a register for use by firmware to determine the touch-status of the sensor.

2. Overview of the Capacitive Touch Software Environment

Since the performance of the CTSU is affected by the physics of the board, Renesas offers Workbench6, which is software that customizes peripheral settings to the application. After the user has tuned the board using the 'First Step Guide (in R20UT3842EJ)', the Renesas capacitive touch solution through Workbench creates example projects that allow the user to monitor their board performance and aid in the fine-tuning of the values to the application. The settings and firmware from these projects integrate into the end user application, completing the workflow found in Figure 4 below.

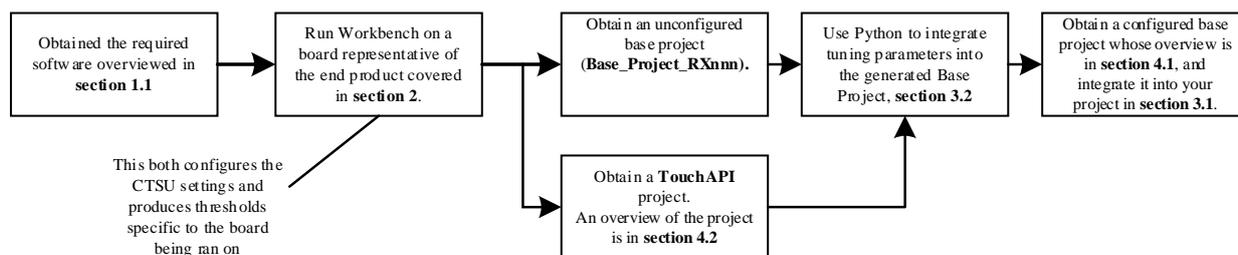


Figure 4: Workflow of working with Renesas' Capacitive Touch Software

This document exists as an overview to guide a user through the different parts of implementing capacitive touch to their end application. It will cover a Workflow and links to other documentation to help integrate a tuned driver into their software.

2.1 Required Software

Before starting, it is required to have the following components installed and working, they are obtained from the Renesas Electronics America website:

- e2studio (version 5.2 or later) integrated development environment
- Renesas RX Compiler v2.0 or later
- Latest version of Workbench6 (V1.07 or later), used to calibrate/monitor the CTSU.
- Python 2.7 (Must be 2.7.xx) - Available via the download section of Python's website.
- E1 debugger to be able to flash and communicate with the target device
- Optional is a USB cable to connect and communicate with the target depending upon the connection method chosen.

2.2 Overview of CTSU Firmware

A brief overview of the firmware architecture is shown below in Figure 5. This describes the driver stack produced after the software tunes the CTSU to the board.

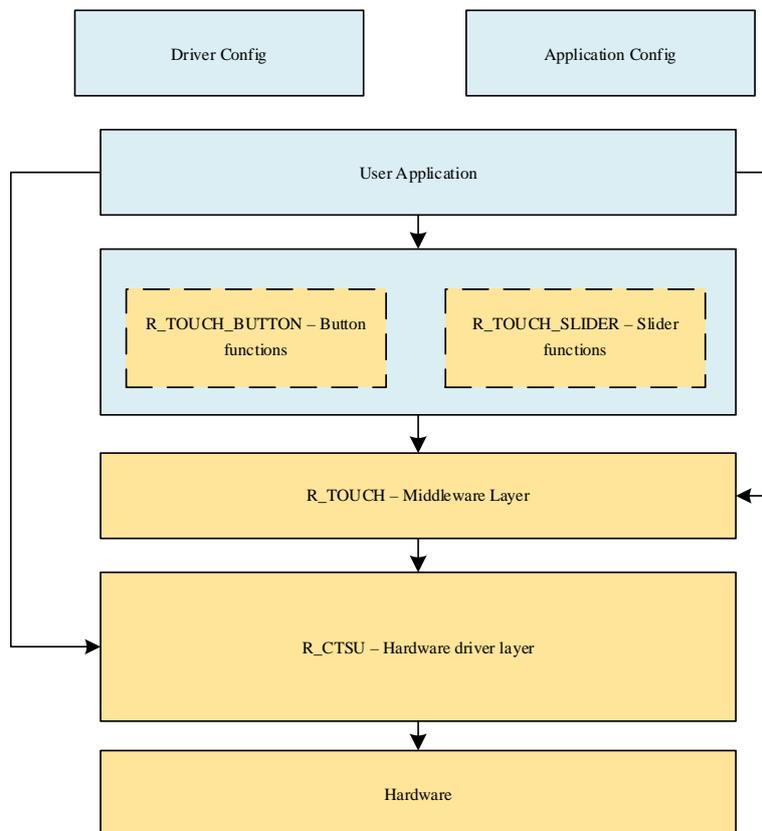


Figure 5: Software stack up of the Capacitive Touch Firmware driver

Described below are the three main parts of the driver:

- **The Hardware Layer** – This layer is what is responsible for converting the changes in touch to digital counts for use by the CTSU layer. An overview of how the peripheral functions is contained in R01AN3824EU and R30AN0218EJ.
- **The CTSU Layer** – The CTSU layer is responsible configuring the peripheral. It contains functions that facilitate the opening, closing, reading and control of the hardware peripheral and puts limits on the digital counts and tuning offsets. Additional information on API specific function calls is contained in R01AN3821EU.
- **The Touch Layer** – The data taken from the CTSU layer, and is then maintained and monitored at the touch layer. Here information regarding the number and configuration of sensor channels, as well as factors related to each touch sensor is contained. The touch layer holds filtering, debouncing and hysteresis configuration of a touch event; additional API information regarding operating the touch layer is found in R01AN3820EU.

In addition to the three main parts, the touch is abstracted to into both a button and slider layer. These parts are available when the user chooses to integrate the Base_Project_RXnnn, into their project where nnn is the RX device number e.g. 130. In the rest of this document, Base_Project_RXnnn will be referred to as Base Project.

- **The button layer** – This software layer takes the touch layer output and through software creates a button abstraction. This allows the creation of useful software features like individualized functions for each sensor pressed; touch events (pressed, released, etc.) and an additional software debounce for each button construct. More information regarding operating the button layer can be found in R01AN3822EU.
- **The slider layer** – This software layer takes the touch output and through software creates a slider, holding not only functions for opening and closing the slider, but information regarding the slider construction as well (wheel based, circle based, number of elements, etc.). More information regarding operating the slider layer can be found in R01AN3823EU.

3. Renesas Tuning Tools – Workbench6

After installing Workbench, the user should first explore Workbench's Document Manager which has a variety of application notes specific to interacting with the peripheral, where this is located is depicted in Figure 6.



Figure 6: The document manager (highlighted in red) contains useful documentation regarding the operation of Workbench and the Renesas Capacitive Touch boards

Before getting started, the user is encouraged to open the document manager and review the 'Workbench6 User Manual (R20UT3842EJ).' In that user manual, it is recommended for the user to review Section 2.2, which describes the tuning process in detail. The goal is to get adequate separation from the base parasitic values after repeatedly touching the board.

After configuring the touch channels through the Graphical User Interface (GUI) found under the 'First Step Guide' and choosing various operating parameters, Workbench will produce ideal settings for the CTSU and generate the appropriate thresholds values for the touch software layer. Should these need to be fine-tuned later, the user is encouraged to connect to Workbench6 and to use both the measurement and Touch Parameters window to fine-tune the generated parameters. More information about tuning a project after it has been generated by Workbench is found in R30AN0291EU.

3.1 Outputs from the Tuning Process

Upon completion of the tuning process, two projects are generated by Workbench6. :

- **Base_Project**– The base project creates the various button, and slider structures by referencing the r_touch and r_ctsu driver layers. In addition, it houses the capacitive touch configuration folder separately, for ease of usage and portability of settings. Because this project is Firmware Integration Technology (FIT) based, it has various API's that are user callable. For more information on these APIs, refer to the Related Documents section at the beginning of this document. It is recommended that users start with this project for any non-evaluation based development.
- **TouchAPI** – The TouchAPI project is for evaluation purposes. It allows ease of use with Workbench's GUI for tuning as well as evaluation. It is recommended that users use this project only for tuning their application with Workbench6. More documentation on the TouchAPI project is in the Workbench document manager.

After obtaining a tuned project, the next step is to create a user application is to either continue with the Base Project or integrate it with the user application.

3.2 Creating a New Base Project

This section details the steps needed to create a Base Project and include it in an application as the basis for a new project.

1. If e2studio has never been started, open it and complete the initial setup phase.
2. End all e2studio processes.
3. Start Workbench6, locate the blue button that displays the 'First Step Guide START' to calibrate the CTSU, which depends on the MCU operating environment and capacitive touch sensor board layout. The 'First Step Guide START,' is located in Figure 7.

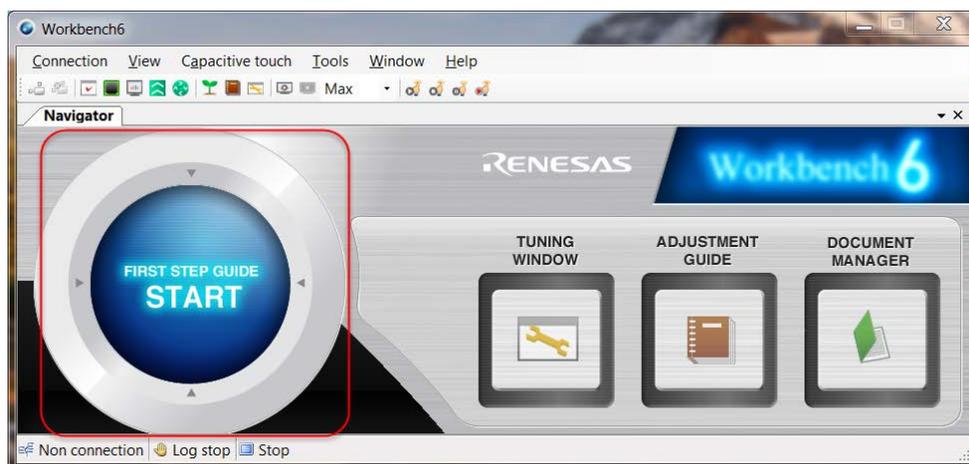


Figure 7: Location of 'First Step Guide START' in the Workbench GUI.

4. When complete, Workbench6 will output a TouchAPI project and derive a Base_Project, which is tunable using the python scripts in the workbench installation folder. The Base Project is generated at the same location specified by the user in the first step guide.
5. Jump to Workbench6 and Python in section 3.2.1 and follow the instructions contained there for configuring the Base_Project generated by Workbench.
6. Import the configured Base Project into the Workspace. The Help Contents in e2Studio provides a guide on Importing Existing Projects.
7. The settings for CTSU sensors will be found in the Base_Project project under the path, src/cts. Some settings for the sensors will be set to a default value.
8. The user can insert code for handling events under src/app. The notification function invoked when the user interacts with the button is located under src/app/button_actions.c. When this function is invoked, an identifier is passed to it, along with the event that occurred. An example application code is given in 'Overview of the FIT Style Project,' section 5.1.

3.2.1 Workbench6 and Python

To facilitate porting tuned parameters used in the TouchAPI project, which aid in the proper setup of the Base Project, Workbench contains a set of python scripts. To execute these scripts manually the user must:

1. Ensure that python is installed and is able to be invoked from a command line prompt.
2. Open a command prompt window and navigate to where workbench is installed, e.g. 'cd \Program Files (x86)\Renesas\Workbench6'
3. Here you will find a collection of python scripts by entering a 'dir *.py' command. To interact with the parsing script type, 'python -B widget_parse_RX_raw.py --help'. This will display a list of commands.
4. Here you will be asked to provide three things for the script to be able to successfully parse the project and add it to the output directory.
 - --input "install directory of where workbench resides"
 - --wbp "complete workbench in which to draw the tuning parameters from," this is the file path associated with the TouchAPI project.
 - --output "complete output directory path in which you want the project setup in"

The completed syntax for the command to be executed all on one line:

```
python -B widget_parse_Rx_raw.py --input "C:\Program Files
(x86)\Renesas\Workbench6\" --wbp
"C:\Workspaces\CaptouchWorkspaces\TouchAPI_20170322105123" --output
"C:\Workspaces\CaptouchWorkspaces\Test"
```

5. Go to the output directory to view the generated Base Project. This is the project to be used when creating or integrating into an end application.

3.2.2 Adding Capacitive Touch Functionality to an Existing FIT Project

This section outlines the process for integrating the generated FIT style drivers into an existing user project, after the board has been tuned using Workbench6.

1. Ensure that the user project uses FIT, and is based on r_bsp v3.40 or later.
2. Copy the modules r_ctsu_rx, r_touch_rx, r_touch_button, and r_touch_slider to the FIT project from the Base Project.
3. Add the following locations to the compile include paths:
 - '\${workspace_loc}/\${ProjName}/r_ctsu_rx'
 - '\${workspace_loc}/\${ProjName}/r_touch_rx'
 - '\${workspace_loc}/\${ProjName}/r_touch_button'
 - '\${workspace_loc}/\${ProjName}/r_touch_slider'
4. Copy the following files from the Base Project's r_config folder to the User Project:
 - r_ctsu_rx_config.h
 - r_touch_rx_config.h
 - r_touch_button_config.h
 - r_touch_slider_config.h
5. Copy the folders src/app and src/ctsu from the Base Project and place them in the User Project.
6. Ensure that the clock settings in r_bsp_config.h match for both the Base and User Project.
7. Initialize all CTSU pins using the r_mpc_rx and r_gpio_rx FIT modules. Then open the touch, button, and slider layers using the APIs from the corresponding modules.
8. Call the scan and update functions to operate the CTSU and perform touch detection.

4. Connecting to Workbench6

The user can monitor the operation of the CTSU and the firmware by connecting the Base Project to Workbench6. By doing this, the user is able to observe the touch interaction in real time, under various environmental conditions such as electrical noise or water.

The following interfaces are supported:

- e2Studio through the E1 debugger
- Serial communication (SCI6)
- Serial over built in USB using CDC (on-chip USB Peripheral available with RX113 and RX231 only).

It is **strongly encouraged**- during development- to leave at least one of the aforementioned methods for connecting to Workbench6 for monitoring the application project. This is to allow count data to be easily extracted from the MCU and analyzed.

All tools for monitoring the operation of the CTSU in Workbench6 are located under the menu item, Capacitive Touch. Monitoring is initiated with Capacitive Touch > Start Monitor. However, the interface for communicating with the target must first be selected.

4.1 e2Studio through the E1 Debugger

The following procedure applies to connect to Workbench6 using the E1 Debugger:

1. Build, download and connect the Base Project to the target using an E1 debugger.
2. Open Workbench6 and use Connection > e2 studio connection
3. Provide the location for the Base Project and Workspace.
4. Ensure the checkbox for “Connect to e2 studio that is already running” is checked. Press OK.
5. The bottom left corner will now notify you by displaying, “Connected (e2 studio)”.

4.2 Serial Communication Interface (SCI6)

The following procedure applies to connect Workbench6 using SCI communication:

1. Open the file src/Base_Project.c in the Base Project.
2. Change #define SCI_SERIAL_USE (false) to #define SCI_SERIAL_USE (true).
3. Connect the PC to the target. Build, and download the Base Project to the target. Reset, and run the target MCU.
4. Open Workbench6 and use Connection > Serial Port Connection
5. Select the correct COM Port that connects to the target MCU over SCI6.
6. Baud rate must be fixed to 38400 and select OK.
7. The bottom left corner will now notify you by displaying, “Connected (Serial Port).”

4.3 Serial Over Built-in USB Using CDC (RX113/RX231)

The following procedure applies to connect Workbench6 using the SCI communication over Built-in USB using the RX113 as an example:

1. Open the file src/Base_Project.c in the Base Project.
2. Change #define USB_SERIAL_USE (false) to #define USB_SERIAL_USE (true).
3. Connect the PC to the target. Build, and download the Base Project to the target. Reset, and run the target MCU.
4. If not installed, locate and install the USB Drivers in Base_Project_RX113/r_usb_pcdc_mini/utilities. When installed, the MCU will be assigned a COM port and list as Renesas RX113 CDC Sample.
5. Open Workbench6 and use Connection > Serial Port Connection.
6. Select the COM Port listing as Renesas RX113 CDC Sample.
7. Any baud rate can be selected.
8. The bottom left corner will now notify you by displaying, “Connected (Serial Port).”

5. Overview of the Generated Software

After integrating the generated firmware into the desired project, it is important to understand both the project structure and some basic parts of the generated project whether a project is just starting out, being added to, or simply evaluating performance.

5.1 Overview of the FIT Style Project

The folder structure of the Base Project separates the configuration and source files for the various layers depicted in Figure 5 (represented by r_ctsu_rx, r_touch_rx, r_touch_button and r_touch_slider) into their own folders in addition to the needed SCI, BSP and GPIO FIT modules. Pictured in Figure 8, is the folder structure for a base project generated for the RX130.

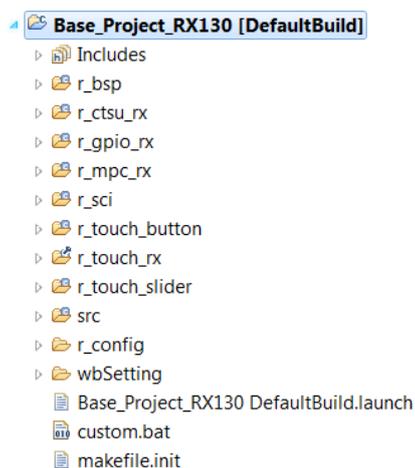


Figure 8: Folder structure of the Base_Project_RX130 project

The main function is held in the Base_Project_RX130.c file under the 'src' directory. This file demonstrates how to configure the pins and opens the various software layers, preparing them for use by the rest of the driver stack.

This project creates a button layer, which interacts with the touch layer for a button like functionality. The user can modify the debounce time for button touches. In addition, it allows for the user to:

- Modify the maximum hold time for touch calibration.
- Modify callback function name. Callback functions are user generated to create button specific functions.

These options are contained in the src/ctsu/button_config.c file. The application code that is referenced by the callback is shown below, and can be modified to be unique for each button if desired. This function is contained under the src/app/button_actions.c file.

```
void App_TOUCH_Button_Notification(void * p_args)
{
    touch_button_callback_arg_t * p_arg = p_args;
    uint32_t button_id = p_arg->id;
    touch_button_event_t button_event = (touch_button_event_t)p_arg->event;

    (void)button_id;
    (void)button_event;

    if (TOUCH_BUTTON_EVENT_REQUEST_DELAY==button_event)
    {
        R_BSP_SoftwareDelay(1, BSP_DELAY_MILLISECS);
    }
}
```

Notice the creation of a button event variable that contains the enumerations described below. These define the different types of events that the button can undergo, and if the application requires it can allow different actions taken based on these events.

```

typedef enum e_touch_button_event
{
    TOUCH_BUTTON_EVENT_RELEASED = 1,           // Button is in the released state

    TOUCH_BUTTON_EVENT_PRESSED = 2,           // Button is in the pressed state

    TOUCH_BUTTON_EVENT_HOLD = 4,             // Button released within valid
                                           // timing

    TOUCH_BUTTON_EVENT_MULTI_TOUCH = 8,      // More than one touch sensor is
                                           // being touched

    TOUCH_BUTTON_EVENT_REQUEST_DELAY = 0x10, // Lower level asking for a delay
} touch_button_event_t;

```

Lastly, the touch threshold and hysteresis settings are contained in the `/src/ctsu/r_touch.h`, file.

5.2 Overview of the Touch API Project

The TouchAPI project is used for evaluation and for tuning purposes only. The folder structure is shown in Figure 9.

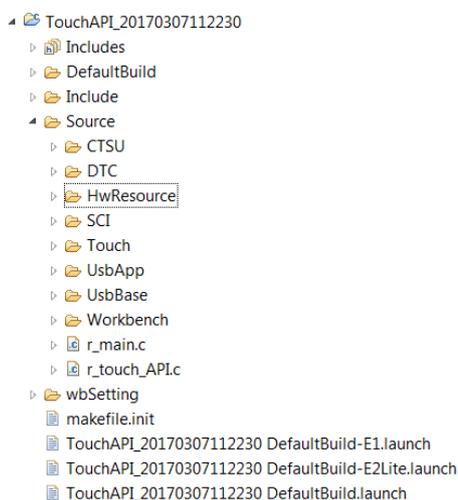


Figure 9: Project structure for the TouchAPI project

The individual tuning thresholds and hysteresis values are contained under the `/Include/Touch/r_touch.h` file. The `r_main.c` file contains the results from the touch event, specifically in the `ts_result` structure. This structure holds the results from a combination of the button, matrix, wheel or slider depending upon how the project was tuned. In the following example, it's checking `button[0]` against the first channel (0x0001). Its rate of scan is based off the `cmt0` interrupt, which controls the setting of a global timing flag.

```
ts_result = R_Get_Cap_Touch_Result( method );

/*The following predicate statement will need to be modified based on the
channel being checked against*/
if (0 != (ts_result.button[0] & 0x0001))
{
/* Touch feedback    Led etc...    */

/* Touch feedback    Led etc...    */
}
else
{
/* No Touch Detected */
}
```

By placing a watch on the ts_result structure, and looking at the members of the structure the application can see which sensor channel that was configured and is being interacted with if it wasn't recorded during the tuning process.

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

Rev.	Date	Description	
		Page	Summary
1.00	May 8, 2017	-	Initial Version Released

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 disputes 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, drawing, chart, program, algorithm, 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 otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
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; and 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.
Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate 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. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
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 majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

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

Renesas Electronics America Inc.

2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
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 II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.

12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141