

Renesas Synergy™ Platform

Mutual Capacitive Touch Software Application Design with Synergy S124 and S3A7 MCUs

Introduction

This application note provides guidelines to users creating a mutual capacitive touch application using the Synergy Software Package (SSP) and Synergy MCUs with an on-chip Capacitive Touch Sensing Unit (CTSU). The application examples included in the package are based on AE-CAP1-S124 and AE-CAP1-S3A7, with the concepts and operation of the CTSU common to all Synergy MCUs.

The target users are application and system developers who have worked with the e² studio and/or IAR Embedded Workbench® for Renesas Synergy™ (IAR EW for Synergy) integrated software developer environment (ISDE). Users should have experience importing a Synergy application to e² studio and opening a new workspace in IAR EW for Synergy, where you configure, generate, build, download, and execute a Synergy application. New Synergy MCU users are recommended to exercise the Blinky Project Tutorial in the *SSP User's Manual* before using this application note. It takes approximately two hours for the average reader to exercise sample projects using this application note.

The software included in this application project is built on the Synergy capacitive touch application example kit, AE-CAP1. This kit includes five PCB boards.



Figure 1. AE-CAP1 Kit

Renesas Synergy S124 and S3A7 MCUs are used in this kit. The methods used to construct and implement the mutual capacitive-touch application design in this application note also apply to other Synergy MCUs.

Required Resources

To build and run the mutual capacitive touch application example, you need:

- AE-CAP1 Kit
- e² studio version 7.5.1 or later, or IAR EW for Synergy version 8.23.3 or later
- SSC v7.5.1 or later
- Synergy Software Package (SSP) version 1.7.0 or later
- Capacitive Touch Workbench for Renesas Synergy™ (CTW for Synergy) version 1.05.0033 or later
- SEGGER J-Link® USB driver
- Two micro USB cables
- Serial to USB conversion cable

Visit the Renesas Synergy™ Software site (www.renesas.com/synergy/software) to download the required software.

Contents

1. Application Overview	4
1.1 Mutual-Capacitive Touch with Synergy MCU.....	4
1.2 Synergy Software Package Frameworks	4
1.3 AE-CAP1 and CTW for Synergy Operational Overview.....	5
1.3.1 Capacitive Touch Application Architecture.....	5
1.3.2 Capacitive Touch Tuning.....	5
2. Capacitive Touch Framework Architecture	6
2.1 Overview.....	6
2.2 Button Framework	7
2.3 Slider/Wheel Framework	7
2.4 Wheel and Slider Position Reporting.....	7
2.5 Capacitive Touch Framework.....	7
2.6 CTSU HAL Driver	8
3. AE-CAP1 Capacitive Touch Application Examples	8
3.1 Overview.....	8
3.2 Threads in the Projects	8
3.3 S124 Capacitive Touch System Configurations	9
3.3.1 Capacitive Touch Button Framework Configurations.....	9
3.3.2 Capacitive Touch Slider Framework Settings	9
3.3.3 Capacitive Touch Framework Settings	9
3.3.4 CTSU HAL Driver Settings	10
3.4 S3A7 Capacitive Touch System Configurations	11
3.4.1 Capacitive Touch Button Framework Configuration Settings	11
3.4.2 Capacitive Touch Slider Framework Settings	11
3.4.3 CTSU HAL Driver Settings	12
3.5 Capacitive Touch Application User Callbacks.....	13
3.5.1 Button Callbacks.....	13
3.5.2 Capacitive Touch Framework Error Callback.....	14
4. Running the AE-CAP1 Mutual-capacitive Touch Application Projects.....	14
4.1 Downloading and Building the Project.....	14
4.2 Downloading to e ² studio.....	14
4.3 Downloading to IAR EW for Synergy	14
5. Unzip AE_CAP1_BSP.zip.....	14
5.1 Importing and Building the Project	14
5.2 Running the AE_CAP1_S124_MT Application Project	15
5.2.1 Powering Up the Board	15
5.2.2 Verifying the AE_CAP1_S124_MT Application Project	16

5.3	Running the AE_CAP1_S124_MT_UART_Monitor Application Project	16
5.3.1	Powering Up the Board	16
5.3.2	Verifying the AE_CAP1_S124_MT_UART_Monitoring Application Project.....	17
5.3.2.1	Standalone Mode	17
5.3.2.2	Communication with CTW for Synergy	17
5.4	Running the AE_CAP1_S3_MT Application Project	18
5.4.1	Powering Up the Board	18
5.4.2	Install the USB CDC Device Driver on Windows	19
5.4.3	Verifying the AE_CAP1_S3A7_MT Application Project	20
5.4.3.1	Standalone Mode	20
5.4.3.2	Communication with CTW for Synergy	20
6.	Synergy Capacitive Touch Solution Resources	21
	Revision History	23

1. Application Overview

1.1 Mutual-Capacitive Touch with Synergy MCU

Mutual capacitance sensing uses two electrodes separated by dielectric media to hold an electric charge. The detected capacitance is the field between the two electrodes: one transmitter and one receiver. Modulating the transmission node induces an electric field that is then detected at the reception node. When a user's finger is applied, it disrupts or absorbs some of the field and decreases the energy detected at the reception node. Mutual capacitance is better suited for the matrix method which optimizes the channel count.

To learn more about the hardware design of mutual-capacitive touch design, see the Capacitive Touch Hardware Design and Layout Guide.

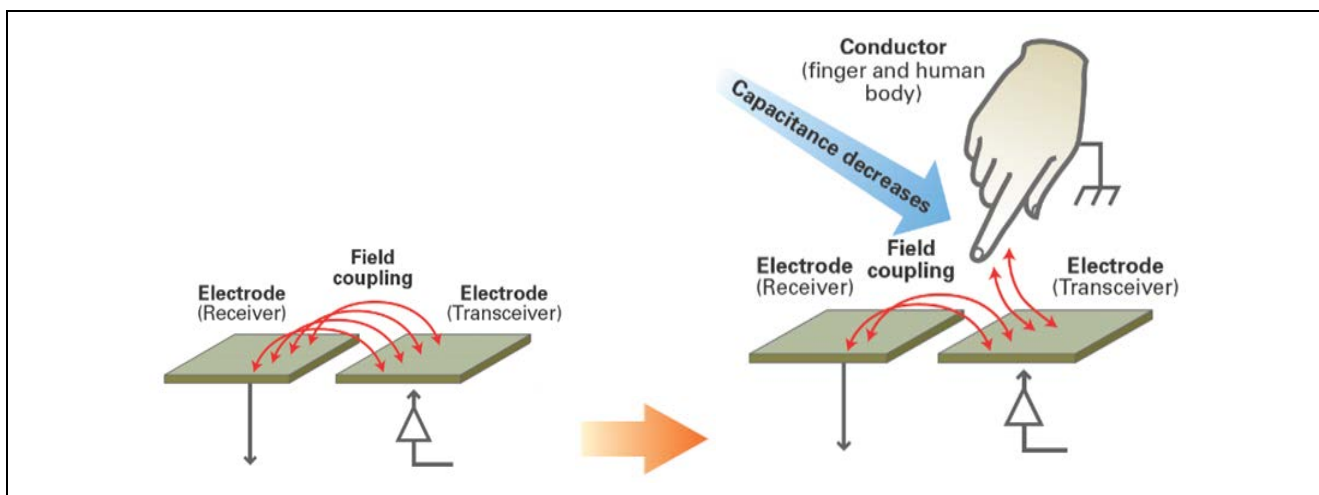


Figure 2. Mutual-Capacitance Sensing

The Capacitive Touch Sensing Unit (CTS) on S124 and S3A7, as well as other Synergy MCUs, such as the RX130 and RX113, have Renesas Touch Gen 2 IP. For details on the CTSU peripheral and its internal operations, see the S124 and S3A7 MCU User's Manuals.

1.2 Synergy Software Package Frameworks

Figure 3 shows the major software systems used in this application project, with the blue box showing SSP components.

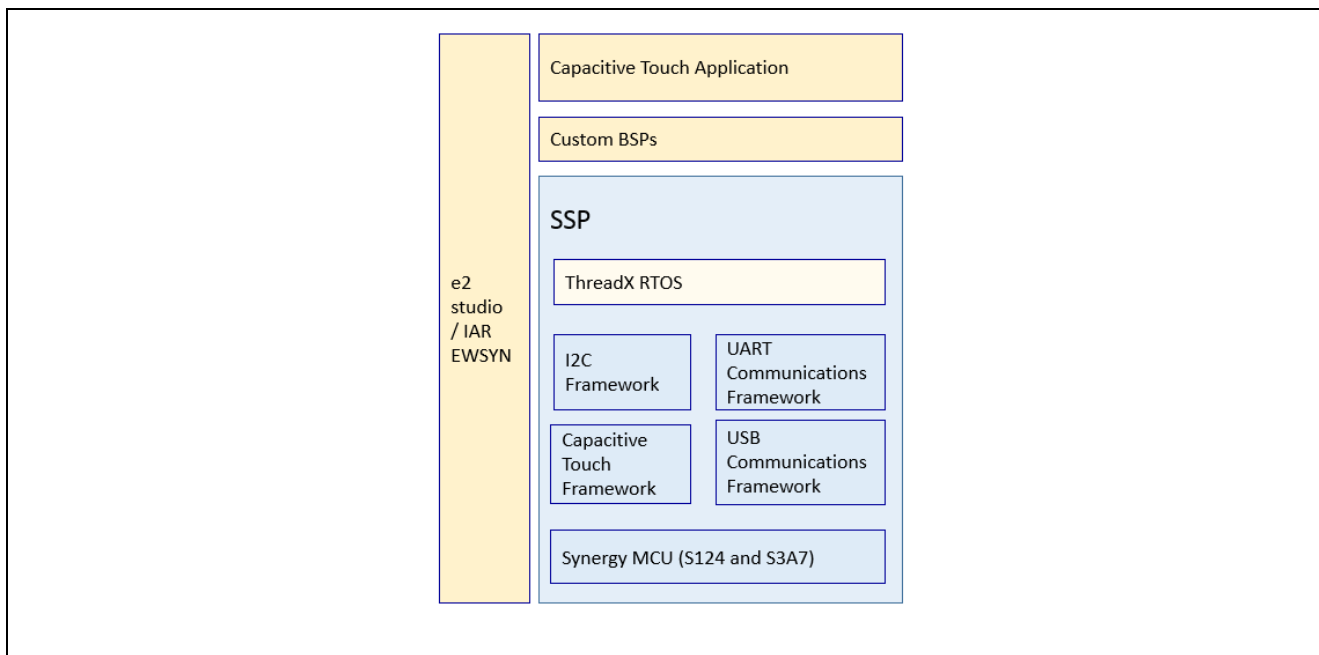


Figure 3. SSP Modules

1.3 AE-CAP1 and CTW for Synergy Operational Overview

Figure 4 is a high-level architectural overview of the Synergy capacitive touch application project and its interaction with CTW for Synergy.

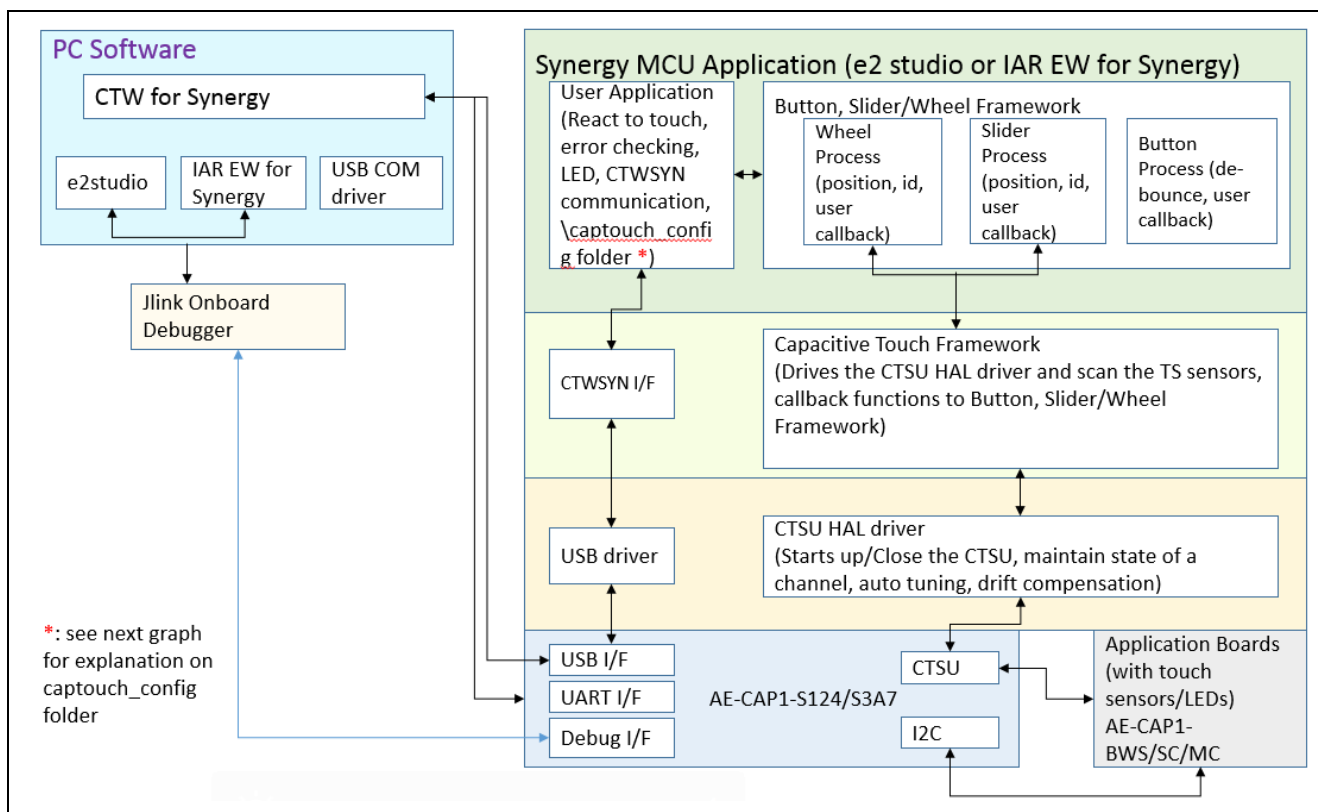


Figure 4. Capacitive Touch System

1.3.1 Capacitive Touch Application Architecture

The capacitive touch application project communicates with the CTW for Synergy to monitor the operation of the capacitive touch sensor and perform fine-tuning based on the application environment.

On the PC side, there are several software components users employ in their capacitive touch applications:

- CTW for Synergy, the software tool that handles capacitive touch tuning and monitoring
- e² studio or the IAR Embedded Workbench® for Renesas Synergy™ (IAR EW for Synergy)
- USB COM, a USB driver provided with this application project

The embedded software performs two main functions:

- Touch detection through various capacitive touch frameworks
 Note: Mutual capacitive touch requires only the button framework because the Synergy capacitive touch solution supports only the mutual capacitive touch button component. The Slider and Wheel capacitive touch button framework is not supported.
- Communicates with the CTW for Synergy through a UART or USB for some of the software projects provided.

For application functionality, see the block in Figure 4 labeled, “Synergy MCU Application (e² studio or IAR EW for Synergy).” Table 1 summarizes the application projects included and lists their functionality.

1.3.2 Capacitive Touch Tuning

Before operating a capacitive touch application project, perform a tuning process. The tuning process sets the touch threshold among other parameters to enable the CTSU to operate properly based on the hardware system. Determining the touch threshold maximizes the sensitivity of the capacitive touch application.

Figure 5 shows software component interaction during the tuning process.

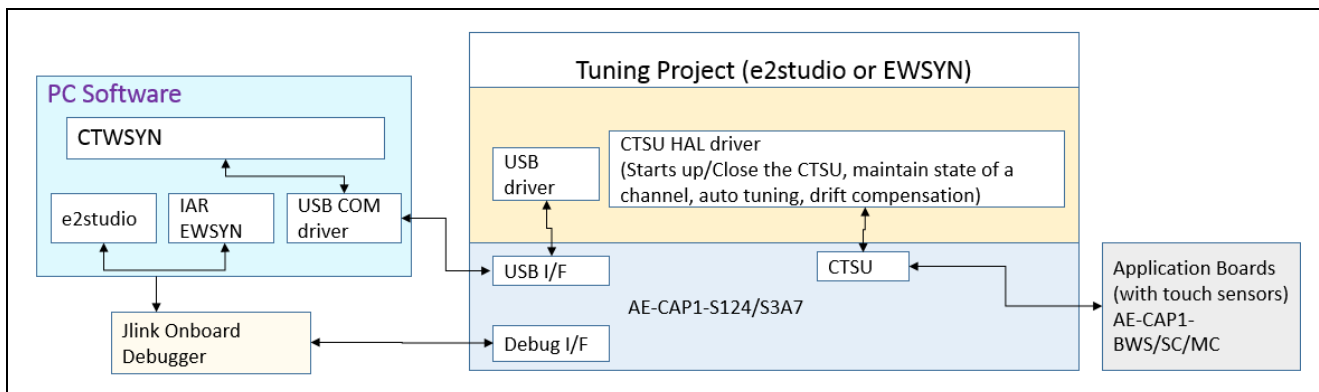


Figure 5. Tuning Setup

This tuning project communicates with CTW for Synergy to generate the `\captouch_config` folder used in the application project. Users need to use the identical PCLKB frequency for tuning and the application project.

This application note focuses on creating application projects once touch tuning has been successfully performed. To learn the tuning process, see Tuning the Capacitive Touch Solution application project listed in section 6.

2. Capacitive Touch Framework Architecture

2.1 Overview

Part of the Renesas Synergy™ Platform, the Synergy™ Software Package (SSP) is a complete, integrated solution designed to provide easy to use, scalable, high quality software for embedded systems.

Figure 6 shows the structural relationship of the software interface to capacitive touch sensing.

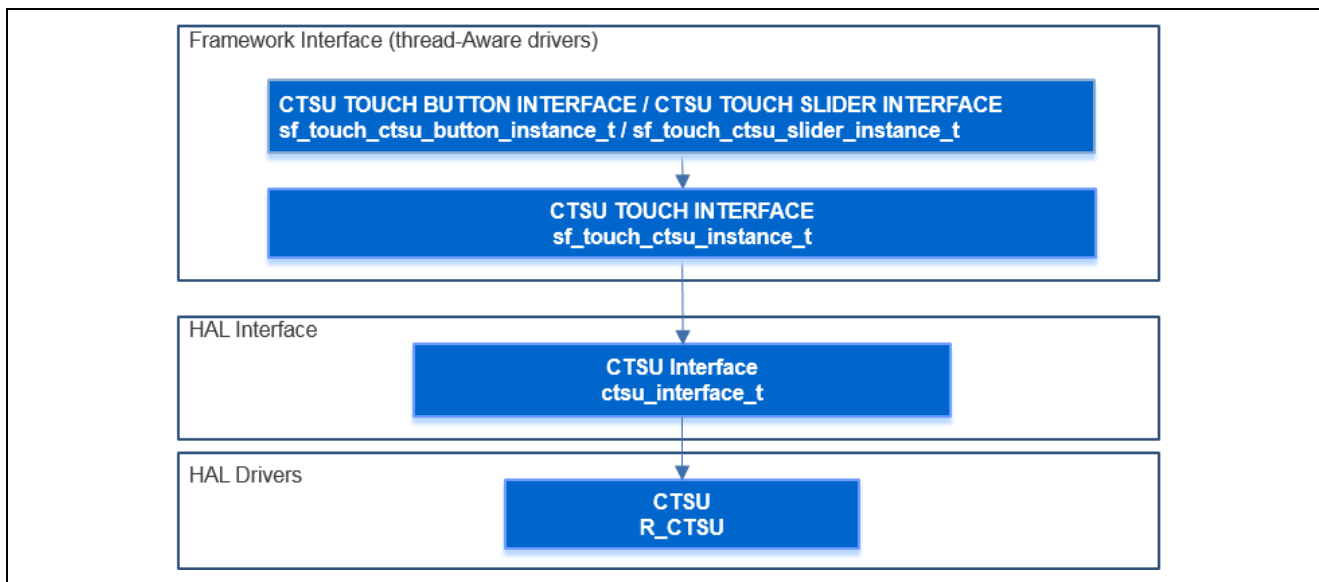


Figure 6. SSP CTSU Framework

Capacitive Touch solution consists of the following SSP frameworks:

- Capacitive Touch Button Framework
- Capacitive Touch Slider/Wheel Framework
- Capacitive Touch Framework

For capacitive touch API usage, see the Capacitive Touch Button Framework Module Guide and the Capacitive Touch Slider Framework Module Guide listed in section 6.

2.2 Button Framework

- The Button Framework enables users to configure the number of buttons, debounce settings, and a callback function that allows a user to act on touch events.
- The Button Framework is used to interpret the CTSU data from all buttons present in the system. The Button Framework also initializes the Capacitive Touch Framework layer.
- The Button Framework registers a callback with the Capacitive Touch Framework layer that occurs each time processed data is available. The Capacitive Touch Button Framework then uses this processed data to perform debounce and to decide which configured event (Press, Release, and so on) is valid for each button.
- The Button Framework invokes the user callback for each button in the order in which they are present in the button configuration table.
- Button Framework layer uses the Capacitive Touch Framework layer to implement a button interface. Using this Button Framework, a user can configure multiple buttons using the configuration structure generated from CTW for Synergy. An action on each button results in a callback with an argument that indicates the button id and event type.

2.3 Slider/Wheel Framework

- The Framework is a generic API implemented on `sf_touch_ctsu_slider` for Capacitive Touch Slider and Wheel applications using the ThreadX® RTOS.
- This Framework requires the Capacitive Touch Framework and the CTSU Driver modules on the Synergy microcontroller hardware.
- The Framework is used to interpret CTSU data for all the slider configurations initialized by the system. It also initializes the Capacitive Touch Framework layer.
- Framework registers a callback with the Capacitive Touch Framework layer, which is called each time processed data is available.
- Uses data inputs (raw values) to determine whether a touch or release occurred and, if so, where it occurred. If there is a state change, the Framework invokes the callback for each slider/wheel, in the order in which they are present in the slider/wheel configuration table, with the corresponding event and position.
- The Framework executes the callback at the update rate (`sf_touch_ctsu_configuration_update_hz`) between the touch and release events.

2.4 Wheel and Slider Position Reporting

The slider position reading from the SSP is linearly distributed on the slider capacitive touch component from 0 to 500. A wheel shape is similar in concept to the slider, but instead of a linear interpretation, the software assumes it is a circle with a layout between 0 and 360 degrees.

2.5 Capacitive Touch Framework

- The Framework is a ThreadX-aware CTSU interface that is used to drive the CTSU HAL driver. The framework can be used to run the CTSU hardware and read back the results of the scans.
- The Framework Interface creates a private thread which drives a hardware scan of a capacitive touch panel and updates the panel at a periodic rate.
- The Framework reads the scanned results using the HAL layer CTSU driver. When a scan is completed, the callback registered by the application layer is invoked. If multiple upper layers are using this framework (for example, button, slider, wheel), this layer invokes the callbacks for each of those layers in the order that they were initialized in this layer.

2.6 CTSU HAL Driver

- The CTSU Driver is used to initialize the CTSU peripheral to detect a change in capacitance on any of the configured (and enabled) channels. It performs requisite filtering and generates data that can be used by higher level widget layers, like buttons, wheel, and sliders.
- To support the different types of data required by these widget layers, the implementation provides a `Read()` function that allows upper level layers to read different types of processed data based on their need.
- The driver also provides a callback when each scan is complete and when new processing data is available. These callbacks can be used by upper layers to read the data.
- The CTSU driver allows the user to configure the CTSU channels for all supported modes of operation, including both Mutual and Self-Capacitance Capacitive Touch.
- The driver scans the configured channels, moves the data using the DTC, perform filtering, drift compensation, and auto-tuning, and notifies the user via a callback once each iteration has completed.
- The driver only supports one configuration at a time, but the user can reopen the driver with multiple channel configurations, as required by the application.

3. AE-CAP1 Capacitive Touch Application Examples

3.1 Overview

The included mutual capacitive touch application software allows user interaction with the mutual capacitive touch board, AE-CAP1-MC.

Table 1 summarizes the software projects included in this application project. All projects support Windows® 10/Windows 7 with e² studio and IAR EW for Synergy, unless otherwise marked.

Table 1. Application Project Summary

Software Projects	Project Description
AE_CAP1_S3A7_MT_App	S3A7 application project for AE-CAP1-MC <ul style="list-style-type: none"> • 20 mutual capacitive touch buttons • LED indication on the AE-CAP1-MT board • Communication with CTW for Synergy through USB
AE_CAP1_S124_MT_App	S124 application project for AE-CAP1-MC <ul style="list-style-type: none"> • 20 mutual capacitance touch buttons • LED indication on the AE-CAP1-MC board
AE_CAP1_S124_MT_UART_Monitor_App	S124 application project for the AE-CAP1-MC board <ul style="list-style-type: none"> • 20 mutual capacitance touch buttons • LED indication on the AE-CAP1-S124 board • Communication with CTW for Synergy through UART

3.2 Threads in the Projects

All sample projects include a CTSU thread for the capacitive touch application. To communicate with CTW for Synergy, this implementation includes a USB communication thread for the S3A7 and a UART communication thread for the S124.

3.3 S124 Capacitive Touch System Configurations

3.3.1 Capacitive Touch Button Framework Configurations

Figure 7 shows some common parameter settings for the Capacitive Touch Button Framework used in the S124 mutual capacitive software projects.

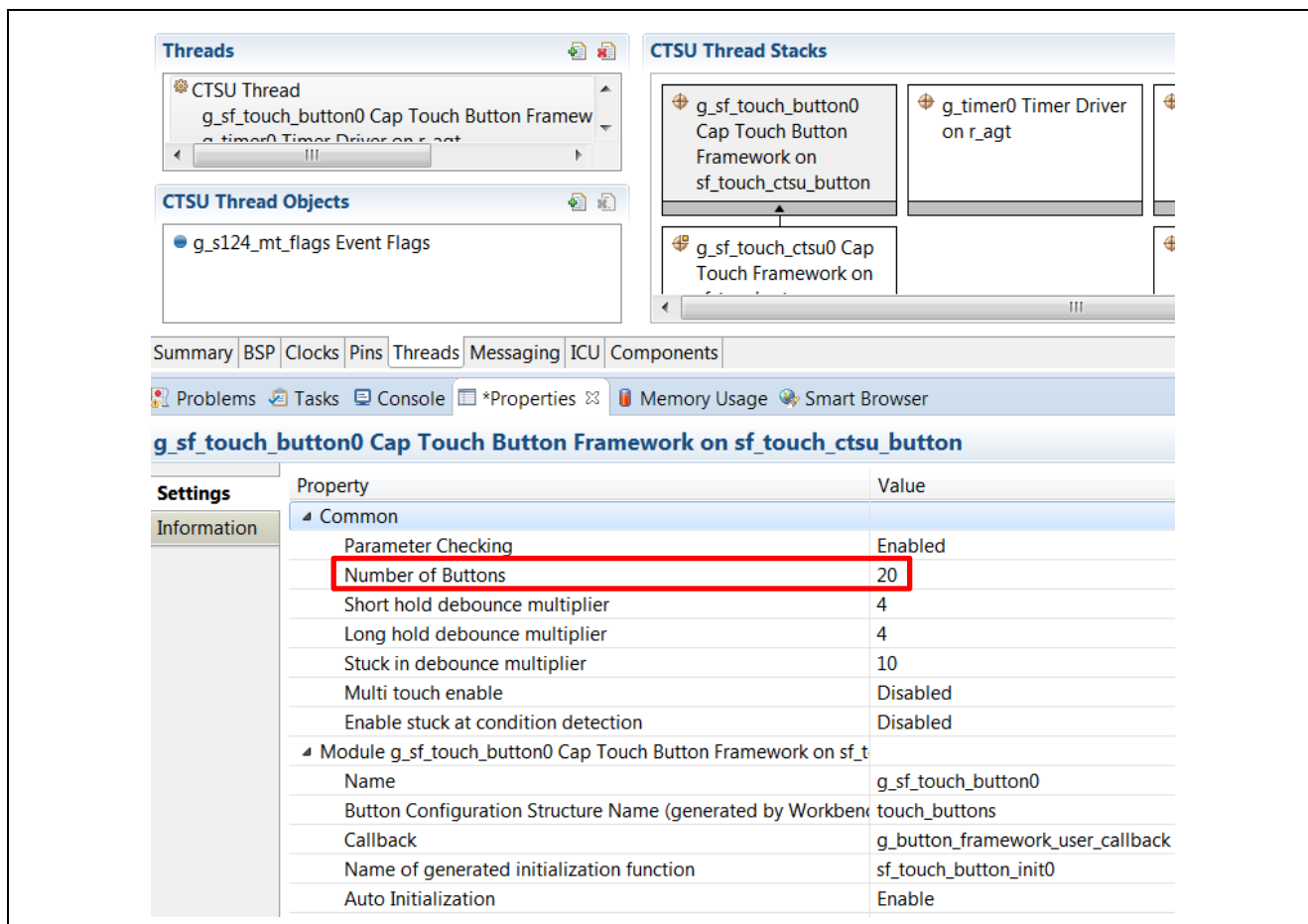


Figure 7. Capacitive Touch Button Framework Settings for S124 in the Mutual-Capacitive Application

While performing the Capacitive Touch Button Framework settings, see the following:

- The AE-CAP1-MC application board uses a 5-row x 4-column matrix setting, with each intersection constructed as a mutual capacitive touch button and the **Number of Buttons** set to 20.
- The user implements the callback function `g_button_framework_user_callback` in the application code.
- The debounce multiplier is internally multiplied by 7 in the application.

3.3.2 Capacitive Touch Slider Framework Settings

For the first release of the AE-CAP1 kit, the mutual-capacitive slider/wheel is not implemented.

3.3.3 Capacitive Touch Framework Settings

Figure 8 shows the common parameter settings for the Capacitive Touch framework in S124 mutual capacitive touch software projects. Users can set the proper thread priority in their software applications.

Note: The property, **Update Hz**, needs to be less than the RTOS tick rate, which is set to 100 Hz in the application.

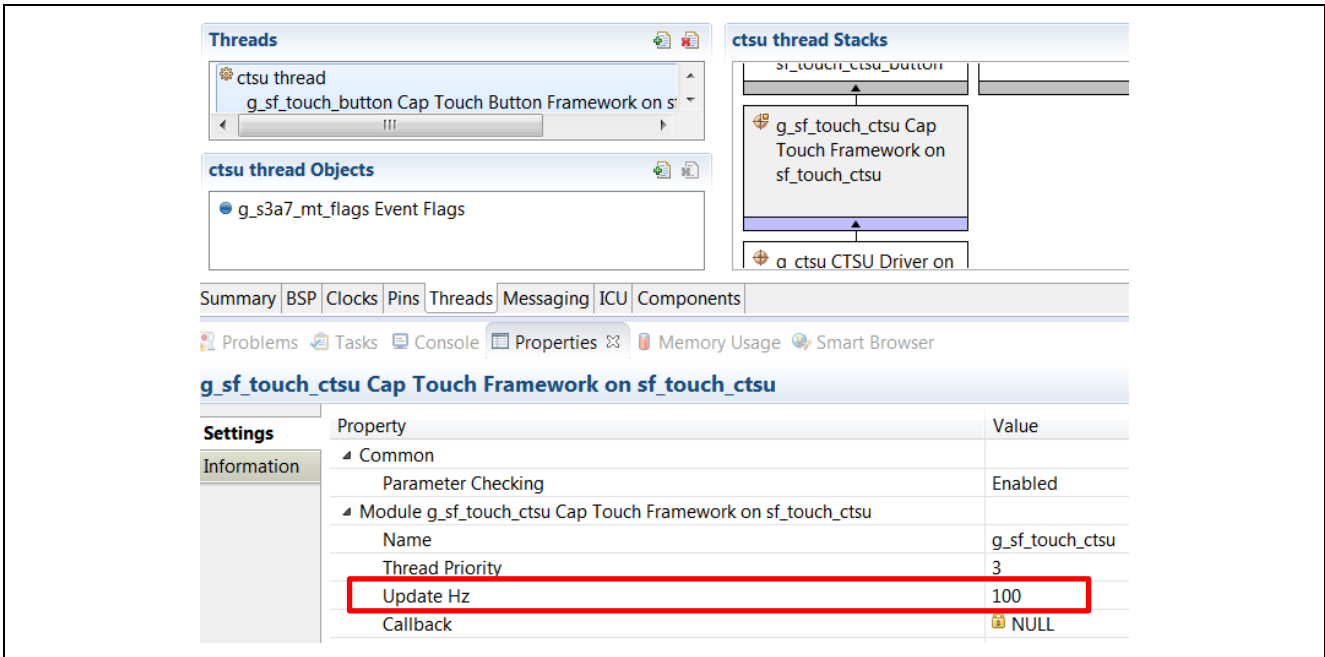


Figure 8. Capacitive Touch Framework Settings for Both S124 and S3A7

3.3.4 CTSU HAL Driver Settings

Figure 9 shows some common parameter settings for the CTSU HAL driver in S124 software projects.

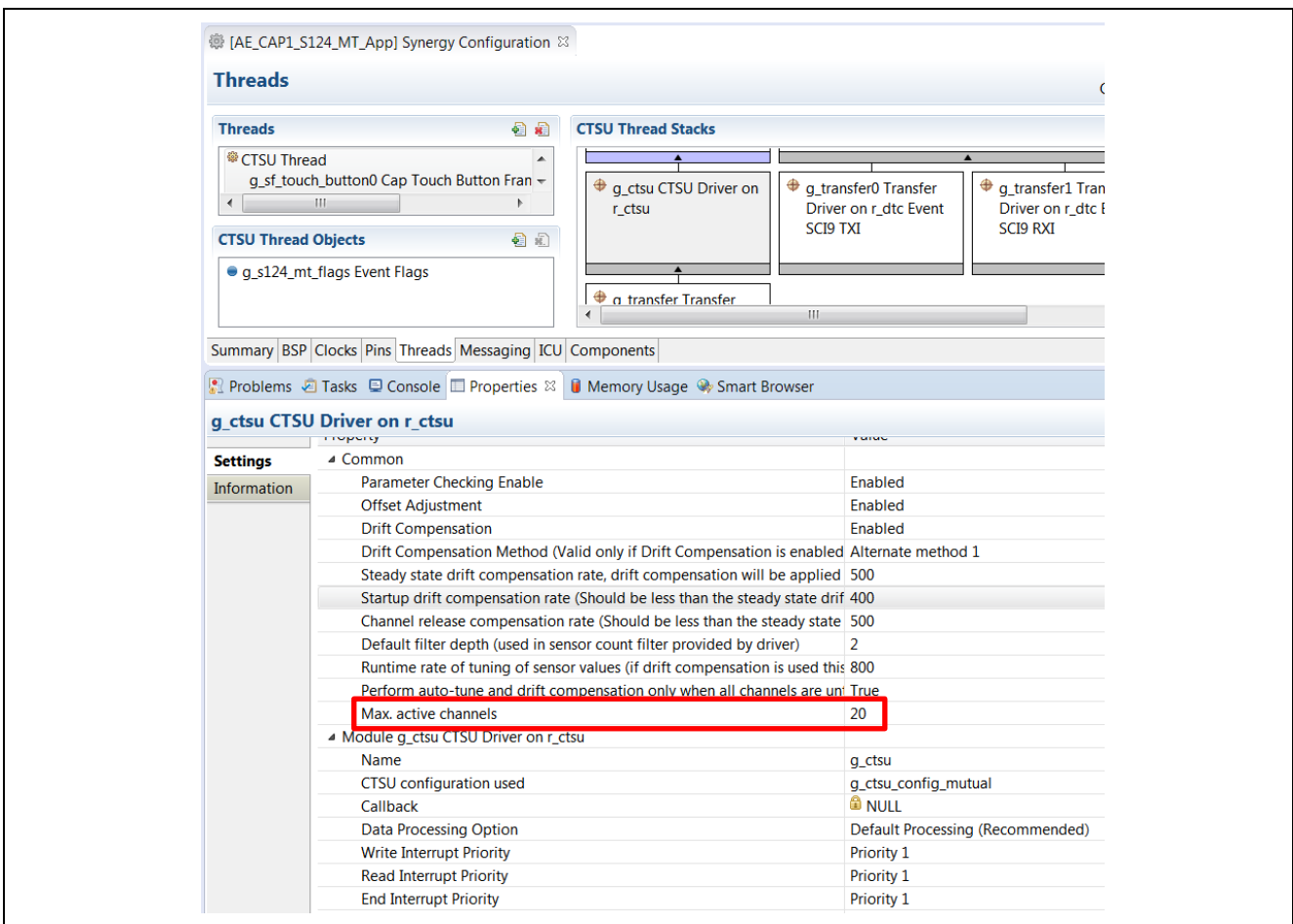


Figure 9. CTSU HAL Driver Settings for S124 in Mutual-capacitive Application

While performing the CTSU HAL Driver settings, see the following:

- Figure 9 shows the property, **Max. active channels**, matches the **Number of Buttons** in Figure 7. See the *SSP User's Manual* to understand various property settings.
- The CTSU configuration is used to set `g_ctsu_config_mutual`, the name of the data structure generated from the tuning process that holds the mutual-capacitance tuning data.
- Startup drift compensation is changed from the SSP default value of 5 to 400. A setting of 5 with the application project corresponds to a drift compensation rate of 50 ms and a setting of 400 corresponds to a drift compensation rate of 4 seconds.
- Users can adjust the **Default Filter Depth** based on the application noise environment. Increasing the **Default Filter Depth** increases noise resistance at a minimal cost in processing time.
- Users can set the **Write, Read, End Interrupt Priority** based on their application.

3.4 S3A7 Capacitive Touch System Configurations

3.4.1 Capacitive Touch Button Framework Configuration Settings

Figure 10 includes some of the common parameter settings for the Capacitive Touch Button framework in the S3A7 software projects.

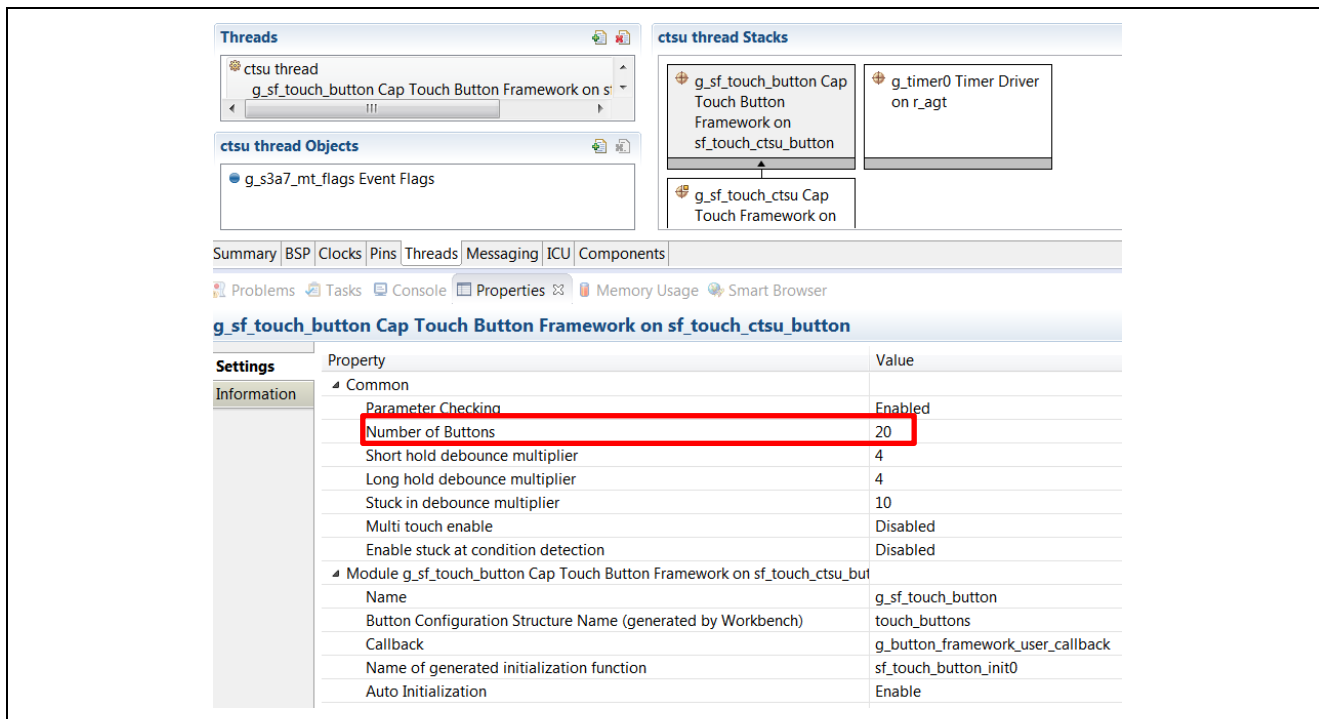


Figure 10. Capacitive Touch Button Framework for S3A7 in Mutual-capacitive Touch Application

While performing the Capacitive Touch Button Framework settings, see the following:

- Similar to the S124 Capacitive Touch system, the mutual-capacitive touch buttons on the AE-CAP1-MC are organized as a 5-row x 4-column matrix with 20 buttons at cross points.
- Users implement the callback function, `g_button_framework_user_callback`, in the application code.
- The debounce multiplier is internally multiplied by 7 in the application.

3.4.2 Capacitive Touch Slider Framework Settings

For the first release of the AE-CAP1 kit, there is no mutual-capacitive slider/wheel implementation.

Between S124 and S3A7 projects, the Capacitive Touch Framework layer shares common settings. See Figure 8 for relevant settings.

3.4.3 CTSU HAL Driver Settings

Figure 11 shows some of the common parameter settings for the CTSU HAL driver in the S3A7 mutual-capacitive touch software projects.

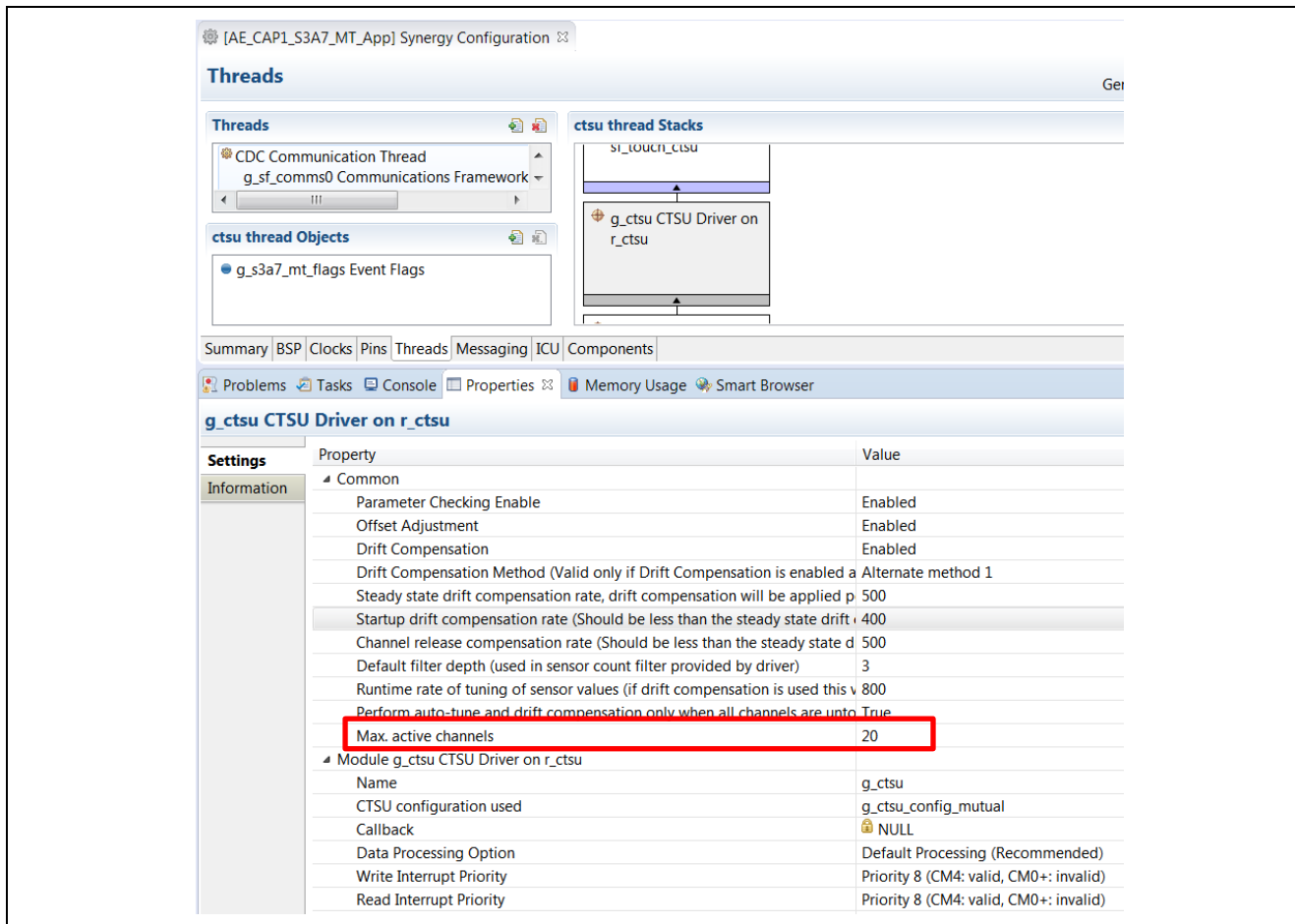


Figure 11. CTSU HAL Driver Settings for S3A7 in Mutual-Capacitive Touch Application

While performing the CTSU HAL Driver settings, see the following:

- Figure 11 shows that the property **Max. active channels** setting matches the **Number of Buttons** setting in Figure 10. See the *SSP User's Manual* to understand these property settings.
- Note that the CTSU configuration is set to `g_ctsu_config_mutual`, the data structure generated from the tuning process that holds the self-capacitance tuning data.
- Startup drift compensation is changed from the default value of 5 to 400. A setting of 5 with the application project corresponds to a drift compensation rate of 50 ms, and a setting of 400 corresponds to a drift compensation rate of 4 seconds
- Users can adjust the **Default Filter Depth** based on the application noise environment. Increasing the **Default Filter Depth** increases the noise resistance at a minimal cost in processing time.
- Users can set the **Write, Read, End Interrupt Priority** based on their application.

3.5 Capacitive Touch Application User Callbacks

3.5.1 Button Callbacks

The application project uses two steps to handle the button callback.

1. Set the button callback event.

```
void g_button_framework_user_callback(sf touch ctsu button callback args t * p_args)
{
    button_callback_args = p_args;
    tx_event_flags_set(&g_s3a7_mt_flags, BUTTON_EVENT, TX_OR);
}
```

Figure 12. Button Framework User Callbacks

2. Pick up the button event in the CTSU thread entry function. Figure 13 shows the process button events in the CB_Self_Button().

```
while (true)
{
    status = tx_event_flags_get(&g_s3a7_mt_flags, ALL_EVENTS, TX_OR_CLEAR, &events_got, TX_WAIT_FOREVER);
    if (status == TX_SUCCESS)
    {
        switch(events_got)
        {
            case BUTTON_EVENT:
                CB_Mutual0Button ();
                break;
            case TIMER_EVENT:
                handle_timer();
                break;
            default:
                break;
        }
    }
    tx_thread_sleep (10);
}
```

Figure 13. Handle Button Callback

3. Figure 14. shows possible events generated from the Button callback.

```
/* handle mutual button touch events */
void CB_Mutual0Button (void)
{
    switch(button_callback_args->event)
    {
        case TOUCH_BUTTON_STATE_INITIAL:
            break;
        case TOUCH_BUTTON_STATE_PRESSED:
            light_up_correspondingLED(button_callback_args);
            break;
        case TOUCH_BUTTON_STATE_LONG_HOLD:
            break;
        case TOUCH_BUTTON_STATE_SHORT_HOLD:
            break;
        case TOUCH_BUTTON_STATE_STUCK:
            break;
        case TOUCH_BUTTON_STATE_RELEASED:
            offLEDS();
            break;
        case TOUCH_BUTTON_STATE_CLOSING:
            break;
        case TOUCH_BUTTON_STATE_DISABLED:
            break;
        case TOUCH_BUTTON_STATE_MULTI_TOUCH:
            break;
        default:
            break;
    }
}
```

LED indication on the application board

Figure 14. Button Events Processing

3.5.2 Capacitive Touch Framework Error Callback

```

void g_sf_touch_button_err_callback(void * p_instance, void * p_data)
{
    ssp_err_t ssp_err_g_sf_touch_button;
    SSP_PARAMETER_NOT_USED (p_instance);
    ssp_err_g_sf_touch_button = *((ssp_err_t*)p_data);

    switch(ssp_err_g_sf_touch_button){
        case SSP_ERR_CTSU_OFFSET_ADJUSTMENT_FAILED:
            Blink_Red_Led = 1;
            break;
        case SSP_ERR_CTSU_SC_OVERFLOW:
            break;
        case SSP_ERR_CTSU_RC_OVERFLOW:
            break;
        case SSP_ERR_CTSU_SAFETY_CHECK_FAILED:
            break;
        default:
            break;
    }
}

```

Figure 15. Automatic Error Checking

You can add other error handling, based on available events.

4. Running the AE-CAP1 Mutual-capacitive Touch Application Projects

4.1 Downloading and Building the Project

The included Mutual_Capacitive Touch Source Code.zip file contains the complete project. Use the following sections to build the projects.

4.2 Downloading to e² studio

1. Unzip the AE_CAP1_BSP.zip. In the package there are two board support package (BSP) files: Renesas.Synergy_board_s3a7_ae_cap1.1.7.0.pack and Renesas.Synergy_Board_s124_ae_cap1.1.7.0.pack.
2. Place these files in your e² studio installation folder: \<your e2_studio folder>\internal\projectgen\arm\Packs.

4.3 Downloading to IAR EW for Synergy

5. Unzip AE_CAP1_BSP.zip.

In the package there are two BSP files:

Renesas.Synergy_board_s3a7_ae_cap1.1.7.0.pack and
Renesas.Synergy_Board_s124_ae_cap1.1.7.0.pack.

Place these files in your IAR EW for Synergy SSC folder: \<your SSC folder>\internal\projectgen\arm\Packs.

5.1 Importing and Building the Project

For instructions on importing the completed project into e² studio or IAR EW for Synergy, see *Renesas Synergy™ Project Import Guide* (r11an0023eu0121-synergy-ssp-import-guide.pdf). After you import the project, the following files structures should appear.

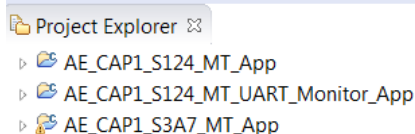


Figure 16. Mutual Capacitive Touch Software Projects

5.2 Running the AE_CAP1_S124_MT Application Project

5.2.1 Powering Up the Board

Using Figure 17 and Figure 18, bring up the AE-CAP1-S124 and AE-CAP1-MT system.

Follow hardware settings in Figure 17 to bring up the AE-CAP1-S124 and AE-CAP1-BWS.

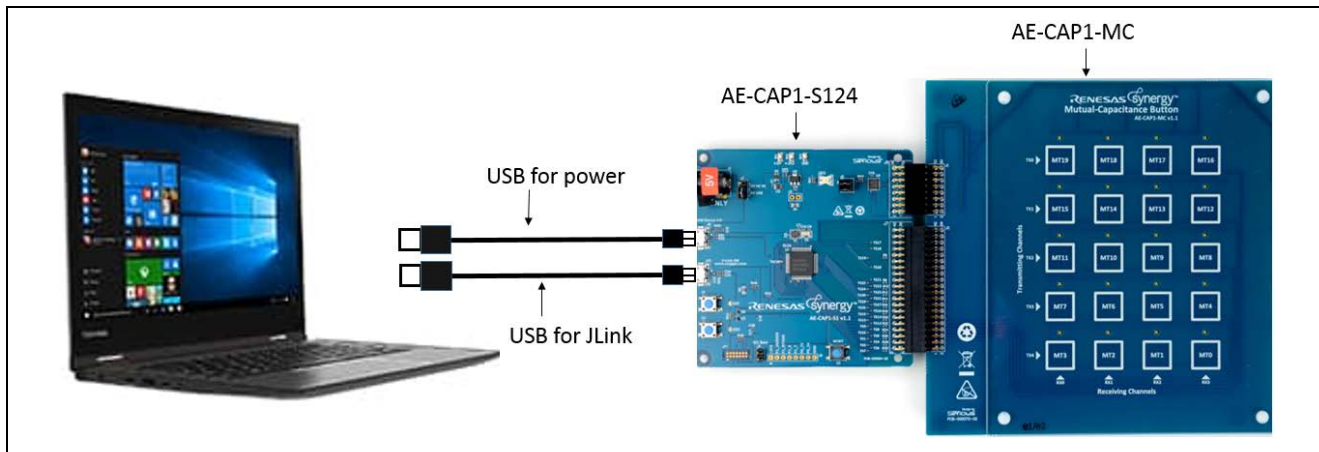


Figure 17. Project AE_CAP1_S124_MT_App Hardware Setup

Keep jumper J4 open, orient jumper J2 towards USB Device, and install jumpers for J5 as shown in Figure 18.

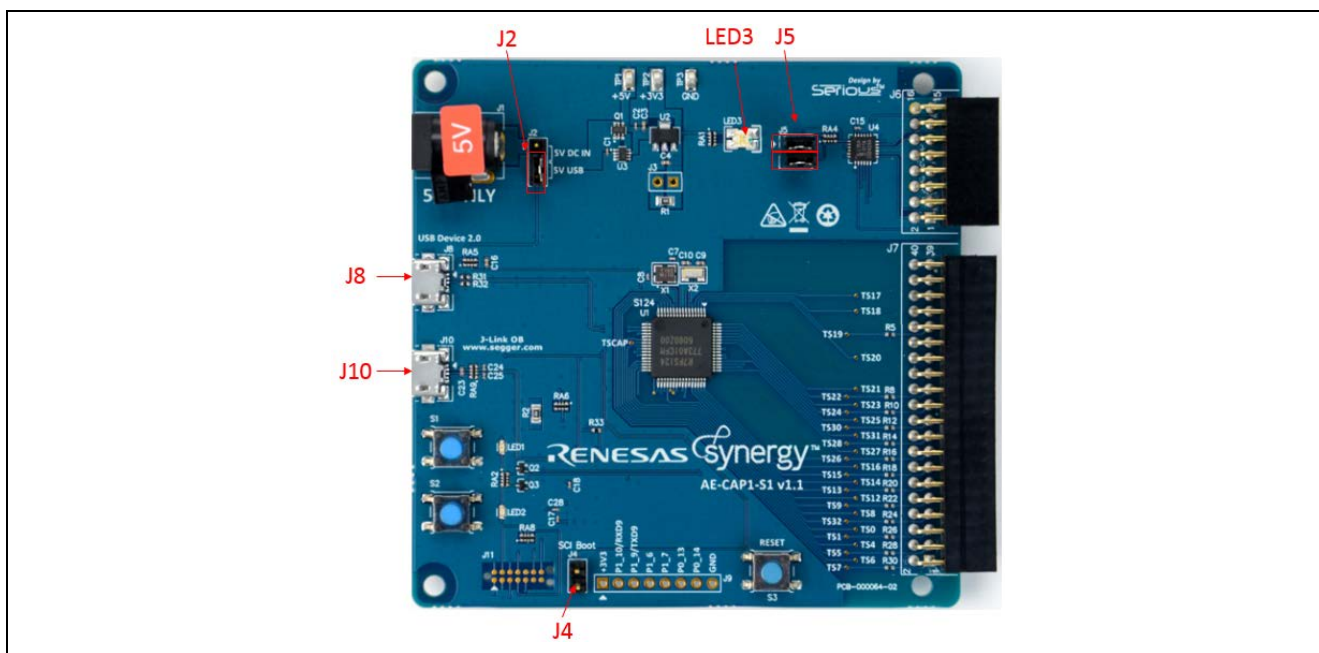


Figure 18. AE-CAP1-S1 Setting for Application Board LED Control

1. Connect AE-CAP1-S1 from J8 to PC using the included USB cable to provide power to the system. LED3 turns on to indicate a good power connection.
2. Connect AE-CAP1-S124 from J10 to PC using the other USB cable included to provide J-Link connection.
3. Build, download, and run the project.

5.2.2 Verifying the AE_CAP1_S124_MT Application Project

When you start the application running, the following system events occur in sequence:

1. The system performs an auto-tuning test on the BWS board (about 2 seconds).
2. When this auto-tuning is finished, LED1 (red) and LED2 (green) on the AE-CAP1-S1 board blinks 5 times. **Do not touch board before LED1 and LED2 start to blink.**
3. If auto tuning is successful, LED2 (green) continues to blink and the board is ready to accept user touch inputs.
4. Press the LED on top of the button to light the LED. Press again to release the button and turn off the LED.
5. If auto tuning fails, only LED1 (red) blinks to indicate a hardware issue. If you get a hardware issue:
 - A. Press the **Reset** button to restart the system.
 - B. Verify that you are using the correct application board, AE-CAP1-MT.
 - C. Make sure that the boards you are using are placed on a non-conductive surface.

5.3 Running the AE_CAP1_S124_MT_UART_Monitor Application Project

5.3.1 Powering Up the Board

Using Figure 19 and Figure 20, bring up the AE-CAP1-S124 and AE-CAP1-MT system.

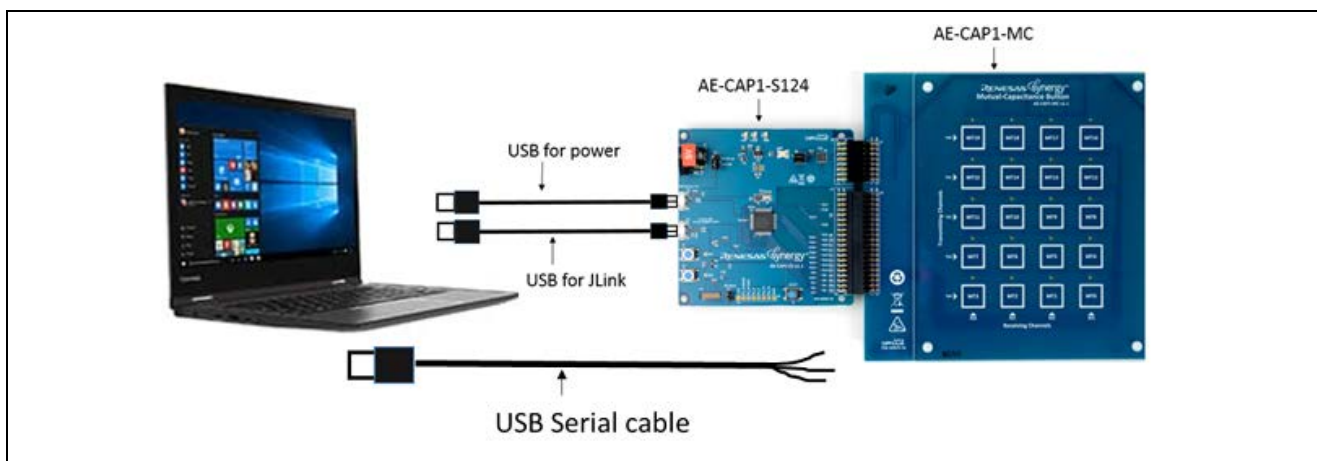


Figure 19. Project AE_CAP1_S124_MT_UART_Monitor_App Hardware Setup

1. On AE-CAP1-S1, keep jumper J4 open, orient jumper J2 towards USB Device, and un-install jumpers from J5 (leave J5 open). See Figure 18 for J4, J2, and J5 jumper locations.
2. Connect the AE-CAP1-S1 board from J8 to the PC using the included USB cable to provide power to the system.
3. Connect the AE-CAP1-S124 board from J10 to the PC using the other USB cable included to provide the J-Link connection.
4. Connect the AE-CAP1-S124 board J9 connector to the PC via the Serial-to-USB converter (see Figure 20). **The USB Serial conversion cable is not included in the kit.**

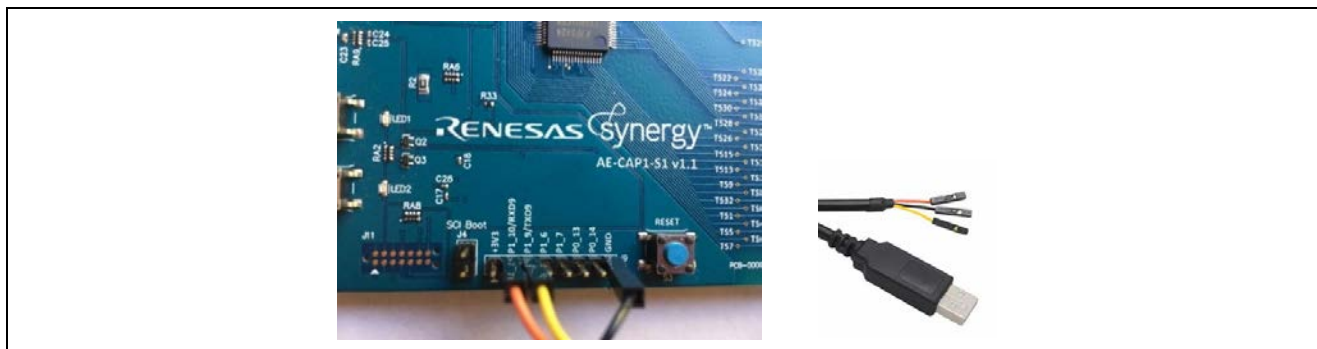


Figure 20. Serial Connection

5. Build, download, and run the project.

5.3.2 Verifying the AE_CAP1_S124_MT_UART_Monitoring Application Project

5.3.2.1 Standalone Mode

Notes:

- If the system is not connected to the debugger:
 - Power on the system and press the **Reset** button to restart the system.
- If the system is operating in debug mode:
 - There is no need to press the **Reset** button.

Once the board is running, the following system events occur in sequence:

1. The system performs a BWS auto-tuning test (about 2 seconds).
2. When auto-tuning completes, LED1 (red), LED2 (yellow), and LED3 (green) on the AE-CAP1-S3 board blink 5 times. **Do not touch the board before LED1, LED2, and LED3 start to blink.**
3. If auto-tuning is successful, LED3 (green) continues to blink and the board is ready to accept user touch inputs. **Note that the LED on the AE-CAP1-MT board is not active with the demo code.**
4. If auto-tuning fails, only LED1 (red) blinks to indicate a hardware issue. If you get a hardware issue, do the following:
 - A. Press the **Reset** button to restart the system.
 - B. Verify that you are using the correct application board, AE-CAP1-MT.
 - C. Make sure that you placed the boards on a non-conductive surface.

5.3.2.2 Communication with CTW for Synergy

1. Launch the CTW for Synergy program, select **Capacitive touch > Start monitor** (see Figure 21).

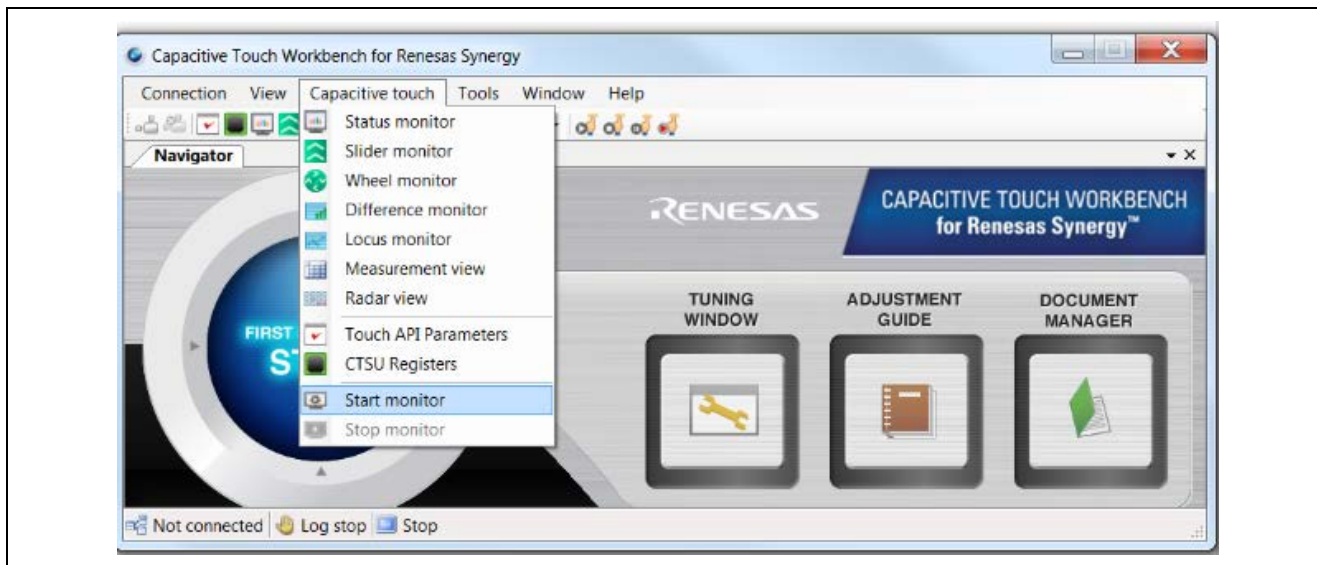


Figure 21. Start the Monitoring Process

2. Select 115200 bps as the UART baudrate to establish communication.

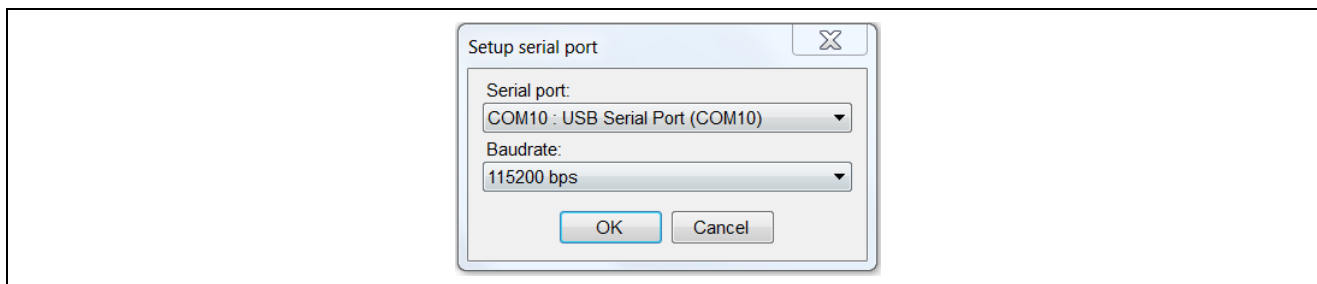


Figure 22. Select the Enumerated COM Port

Table 2 maps markings on the AE-CAP1-MC board to the touch button assignments in the CTW for Synergy S124:

Table 23. CTW for Synergy S124 Mutual Button Mapping with AE-CAP1-MC Board Silk Screen Marking

CTW for Synergy Assignment	AE-CAP1-MC Mark	CTW for Synergy Assignment	AE-CAP1-MC Mark	CTW for Synergy Assignment	AE-CAP1-MC Mark
Mtx0	MT16	Mtx7	MT9	Mtx14	MT2
Mtx1	MT12	Mtx8	MT5	Mtx15	MT19
Mtx2	MT8	Mtx9	MT1	Mtx16	MT15
Mtx3	MT4	Mtx10	MT18	Mtx17	MT11
Mtx4	MT0	Mtx11	MT14	Mtx18	MT7
Mtx5	MT17	Mtx12	MT10	Mtx19	MT3
Mtx6	MT13	Mtx13	MT6		

Figure 23 shows an example of status monitoring with Mtx0. When you press the MT16 button on the AE-CAP1-MC, you can monitor your detected touch input. Right-click anywhere in the graph area to select other channels to monitor.

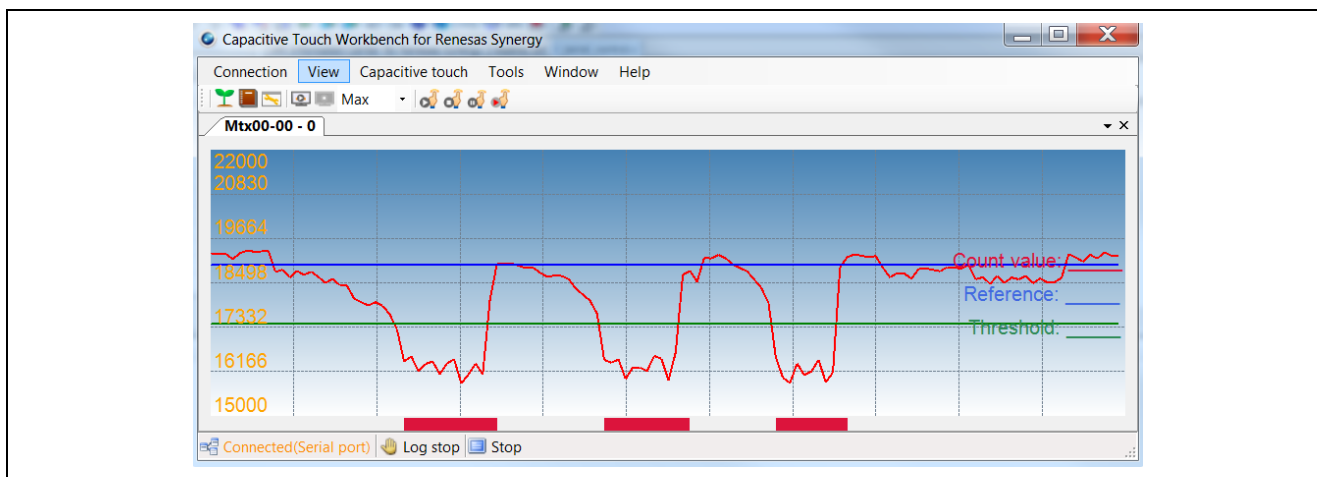


Figure 23. Monitoring the Touch Activity with the CTW for Synergy

5.4 Running the AE_CAP1_S3_MT Application Project

5.4.1 Powering Up the Board

Using Figure 24 and Figure 25, bring up the AE-CAP1-S3A7 and AE-CAP1-MC system as follows.

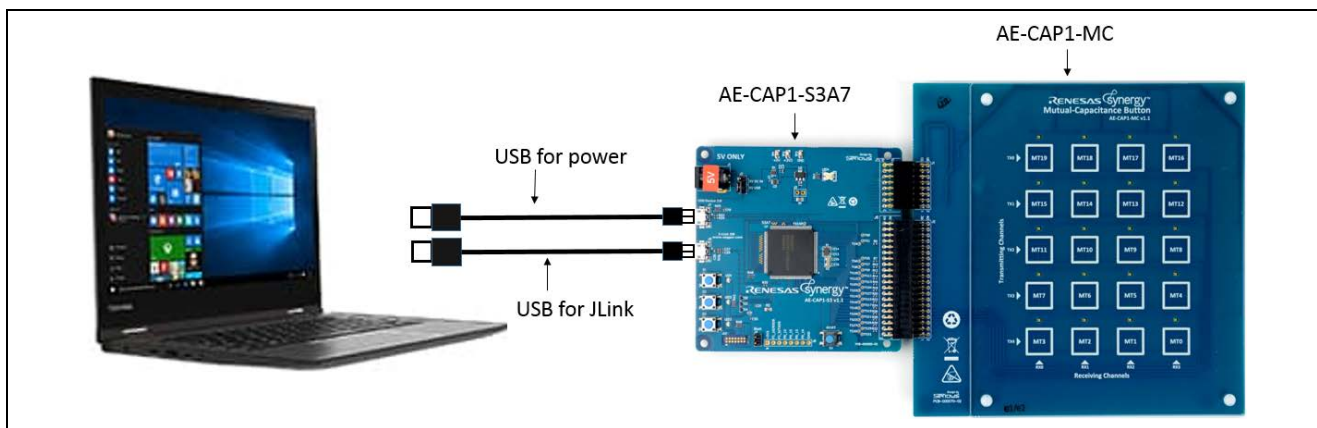


Figure 24. Project AE_CAP1_S3A7_MT_App Hardware Setup

1. Leave jumper J4 open and orient jumper J2 towards the USB Device.

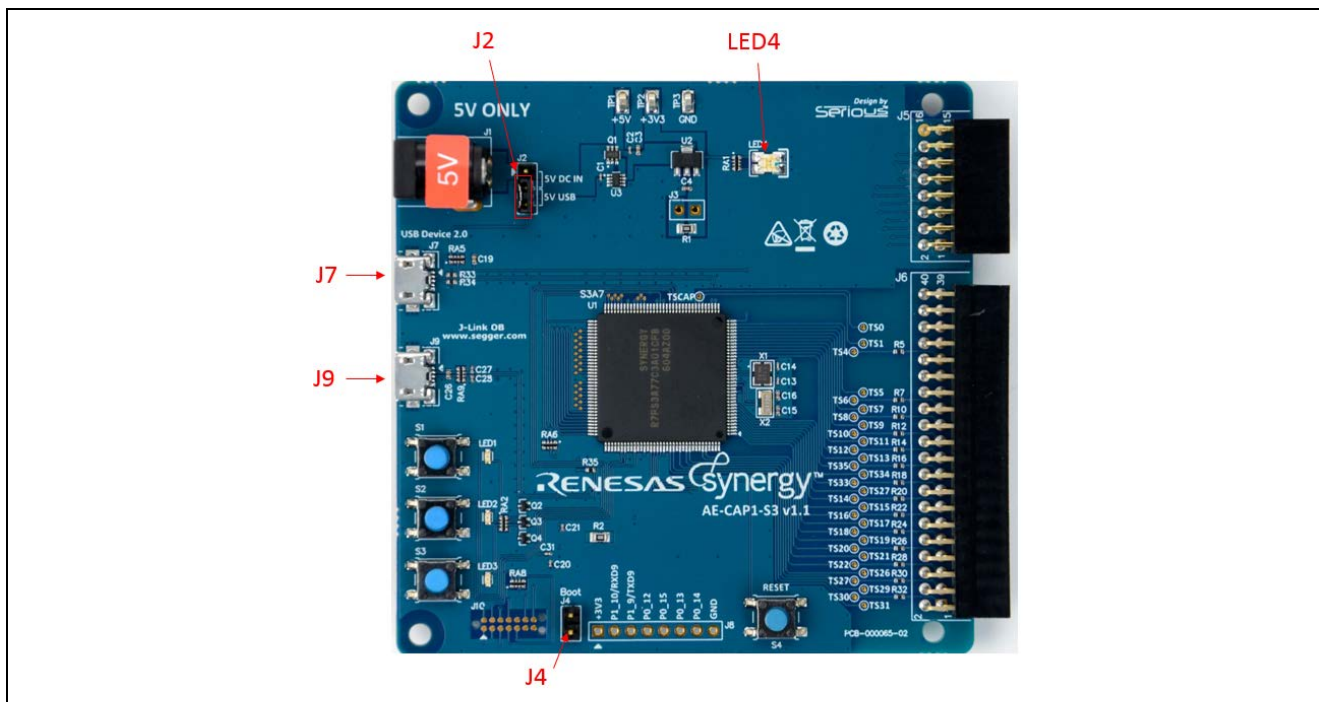


Figure 25. AE-CAP1-S3 Setting

2. Connect the AE-CAP1-S3 board from J7 to the PC using the included USB cable to provide power to the system.
3. Connect the AE-CAP1-S3A7 board J9 connector to the PC using the other USB cable included to provide the J-Link connection.
4. Build, download, and run the application project.

5.4.2 Install the USB CDC Device Driver on Windows

As Figure 25 shows, the S3A7 software uses USB (J9 connector) to communicate with the CTW for Synergy, with the following caveats:

- For a Windows® 10 PC, there is no need to install a USB driver. The USB Serial Device enumeration occurs automatically with the sample projects included in the package.
- For a Windows 7 PC, perform the following steps to install the USB driver.

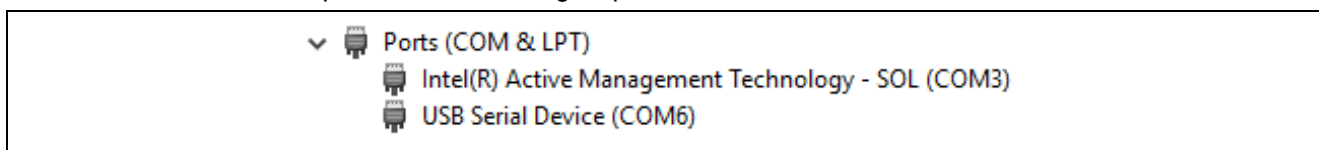


Figure 26. USB Serial Device COM Port on Windows 10

1. Install the USB CDC/ACM device driver attached to this application project, `Windows_USB_serial_driver.zip`. Unzip the package to the folder: `\Windows_USB_serial_driver`.
2. After downloading and running the application, the CDC/ACM device appears in the Windows® 7 Device Manager as **UNKNOWN DEVICE** in the **Universal Serial Bus Controller** group.
3. Right-click on the USB **UNKNOWN DEVICE** and select **Update Driver Software**.
4. When prompted for the driver location, navigate to the `\Windows_USB_serial_driver` location created in step 1. Once the driver is updated, the new COM device appears in the Device Manager.



Figure 27. Communications Port on Windows 7

5.4.3 Verifying the AE_CAP1_S3A7_MT Application Project

If you are working with a Windows 7 PC, see section 5.4.2 to install the USB driver for Windows 7. Once the USB CDC enumeration is properly installed for your Windows 7 or Windows 10 PC, use the following steps to continue operation.

5.4.3.1 Standalone Mode

Notes:

- If the system is not connected to the debugger:
 - Power on the system and press the **Reset** button to restart the system.
- If the system is operating in debug mode:
 - There is no need to press the **Reset** button.

Once the board is running, the following system events occur in sequence:

1. The system performs a BWS auto-tuning test (about 2 seconds).
2. When auto-tuning completes, LED1 (red), LED2 (yellow) and LED3 (green) on the AE-CAP1-S3 board blinks 5 times. **Do not touch the board before LED1, LED2, and LED3 start to blink.**
3. If auto-tuning is successful, LED3 (green) continues to blink and the board is ready to accept user touch inputs.
4. Press the LED on top of the button to light the LED. Press again to release the button and turn off the LED.
5. If auto-tuning fails, only LED1 (red) blinks to indicate a hardware issue. If you get a hardware issue, do the following:
 - A. Press the **Reset** button to restart the system.
 - B. Verify that you are using the correct application board, AE-CAP1-MT.
 - C. Make sure you placed the boards on a non-conductive surface.

5.4.3.2 Communication with CTW for Synergy

Figure 21 shows how to start the monitoring process with CTW for Synergy. You will see a different COM port enumeration if you are using Windows 10 instead of Windows 7. See section 5.4.2 to understand enumeration.

Figure 22 shows you select the COM in a similar way to communicate with CTW for Synergy. **Make sure you connect J7 USB Device port to PC when working with this program.** See Table 4 to map the marking on the AE-CAP1-MC to the touch sensor channel on CTW for Synergy:

Table 4. CTW for Synergy S3A7 Mutual Button Mapping with AE-CAP1-MC Board Silk Screen Marking

CTW for Synergy Assignment	AE-CAP1-MC Mark	CTW for Synergy Assignment	AE-CAP1-MC Mark	CTW for Synergy Assignment	AE-CAP1-MC Mark
Mtx0	MT4	Mtx7	MT13	Mtx14	MT18
Mtx1	MT0	Mtx8	MT9	Mtx15	MT7
Mtx2	MT12	Mtx9	MT17	Mtx16	MT3
Mtx3	MT8	Mtx10	MT6	Mtx17	MT15
Mtx4	MT16	Mtx11	MT2	Mtx18	MT11
Mtx5	MT5	Mtx12	MT14	Mtx19	MT19
Mtx6	MT1	Mtx13	MT10		

Figure 28 shows an example of the status monitoring with Mtx0. When you press the MT4 button on the AE-CAP1-MC, you can monitor your detected touch input. Right-click anywhere in the graph area to select other channels to monitor.



Figure 28. Monitoring the Touch Activity with CTW for Synergy

6. Synergy Capacitive Touch Solution Resources

Software Tools and Kits

- Visit www.renesas.com/synergy/tools to learn about development tools.
- Visit www.renesas.com/synergy/gallery to download software tools and utilities.
- Learn about:
 - Synergy kits at www.renesas.com/synergy/kits
 - Synergy Microcontrollers at www.renesas.com/synergy/mcus
 - Other Synergy Application Projects that run on the AE-CAP1 kit at: www.renesas.com/synergy/ae-cap1
 - Synergy Software at www.renesas.com/synergy/software
 - Renesas Synergy Solutions Gallery at www.renesas.com/synergy/solutionsgallery

Documentation

- Download *Capacitive Touch Hardware Design and Layout Guidelines for Synergy, RX200, and RX100* from www.renesas.com/synergy/docs to learn about the hardware design guidelines for Renesas capacitive touch.
- Download *Tuning the Capacitive Touch Solution (R20AN0448EU)* to learn about capacitive touch tuning with the CTW for Synergy.
- Download *Capacitive Touch Button Framework Module Guide* from www.renesas.com/us/en/software/D6003125.html
- Download *Capacitive Touch Slider Framework Module Guide* from: www.renesas.com/us/en/software/D6003126.html

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	May.04.17	—	Initial release
1.01	Nov.28.17	—	Updated for SSP v1.3.0 and CTW for Synergy version 1.05.0000.28 and later
1.02	Dec.05.17	—	Source code updated
1.03	Apr.27.18	—	Updated for SSP v1.4.0
1.04	Sep.27.18	—	Updated for SSP v1.5.0
1.05	Mar.22.19	—	Updated for SSP v1.6.0
1.06	Oct.14.19	—	Updated for SSP v1.7.0

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.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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/.