CTSU vs CTSU2 comparative

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by
Nira Tubert Comas

In partial fulfilment
of the requirements for the degree in
Telecommunications Technologies and Services Engineering

Advisor: Manuel M. Dominguez Pumar

Barcelona, January 2022

Abstract

The project consists of a comparative of two Capacitive Touch Sensing Units using a touch user interface.

The first one (CTSU) is applied to an existing Renesas Xtreme Family (RX) microcontroller board.

A new software has been implemented for the second one (CTSU2), which is applied to Renesas Advanced Family (RA) microcontroller board.

Tests in its basic configuration, when configuring multi-frequency and configuring parallel scan have been run and compared for both boards.

Based on those test results, two final touch control configurations are selected and ElectroMagnetic Compatibility tests have been executed.

Gathering all the results, it can be seen that, when using parallel scan on a RA microcontroller, the measurement time decreases significantly and Signal to Noise Ratio (SNR) results are acceptable, although they are worse than on other configurations. Using multifrequency configuration EMC tests have passed successfully. In conclusion, the results obtained from CTSU2 exceed those from CTSU.

Resum

El projecte consisteix en una comparativa de dues Capacitive Touch Sensing Units utilitzant una interfície d'usuari tàctil.

La primera (CTSU) s'aplica a una placa existent de microcontroladors Renesas Xtreme Family (RX).

S'ha implementat un nou programari per a la segona (CTSU2), que s'aplica a una placa de microcontrolador Renesas Advanced Family (RA).

S'han realitzat tests en la seva configuració bàsica, configurant multifreqüència i escaneig paral·lel, i s'han comparat per a ambdues plaques.

A partir d'aquests resultats, s’han seleccionat dues configuracions finals de control tàctil i s'han realitzat proves de Compatibilitat ElectroMagnètica.

Recollint tots els resultats, es pot observar que, quan s'utilitza l'escaneig paral·lel en un microcontrolador RA, el temps de mesura disminueix significativament i els resultats de relació senyal/soroll (SNR) són acceptables, tot i que pitjors que en altres configuracions. Utilitzant la configuració de multifreqüència, les proves d'EMC s’han superat correctament. En conclusió, els resultats obtinguts de CTSU2 superen els de CTSU.

Resumen

El proyecto consiste en una comparativa de dos Capacitive Touch Sensing Units utilizando una interfaz de usuario táctil.

La primera (CTSU) se aplica a una placa existente de microcontroladores Renesas Xtreme Family (RX).

Se ha implementado un nuevo software para la segunda (CTSU2), que se aplica a una placa de microcontrolador Renesas Advanced Family (RA).

Se han realizado tests en su configuración básica, configurando multifrecuencia y escaneo paralelo, y se han comparado para ambas placas.

A partir de estos resultados, se han seleccionado dos configuraciones finales de control táctil y se han realizado pruebas de Compatibilidad ElectroMagnética.

Recogiendo todos los resultados, puede observarse que, cuando se utiliza el escaneo paralelo en un microcontrolador RA, el tiempo de medida disminuye significativamente y los resultados de relación señal/ruido (SNR) son aceptables, aunque peores que en otras configuraciones. Utilizando la configuración de multifrecuencia, las pruebas de EMC se han superado correctamente. En conclusión, los resultados obtenidos de CTSU2 superan a los de CTSU.

Revision history and approval record

Revision	**Date**	**Purpose**
0	07/10/2021	Document creation
1	21/01/2022	Document revision

DOCUMENT DISTRIBUTION LIST

Name		**Position**
Nira Tubert Comas		Project Author
Manuel M. Dominguez Pumar		Project Supervisor
Ignasi Villagrasa Isasi		Company Supervisor

Written by:		Reviewed and approved by:	
Date	07/10/2021	Date	21/01/2022
Name	Nira Tubert Comas	Name	Manuel M. Dominguez Pumar
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Statement of purpose

The project scope is to evaluate new technology improvements, based on the Capacitive Touch Sensing Unit (CTSU) technology from Renesas.

It will be taken as reference the CTSU2 applied to the Renesas Advanced Family (RA) microcontrollers, in respect to former CTSU in a real application environment using the Renesas Xtreme Family (RX) microcontroller.

The project is carried out at the company E.G.O. Appliance Controls S.L.U, a company dedicated to the development, production and sale of electronic circuits for different types of applications. So, for doing the comparative, an EGO’s existing touch user interface named T4, applied to a washer, will be taken as a reference.

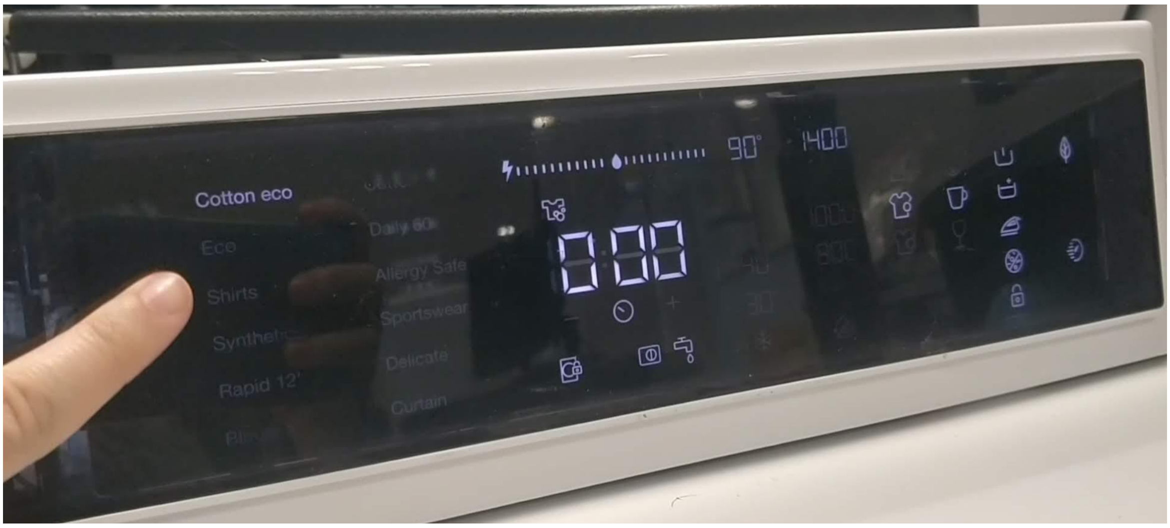


Figure 1. T4 touch user interface

The project main goals are:

1. To implement software to the new board according to the RA and CTSU2 specification.
2. Get basic functionality for a board sample with RA microcontroller (touch control routines).
3. Run a set of tests for RA CTSU technology board including:
    a. Tests in former CTSU configuration.
    b. Tests configuring multi-frequency scan.
    c. Tests configuring parallel scanning function.
4. EMC tests. Based on those test results, a final touch control configuration for the RA CTSU technology board will be selected. With this configuration a set of EMC tests will be run.

1.2. Background

Until now CTSU has been the reference technology for EGO on current existing UI products. Even if it's an improved technology in respect to first touch control generation, there are some aspects to consider that can lead to quality issues:

- Scan rate / sensitivity: As more touch channels are required, it's important to increase the readability rate per channel.
- In some environments, coupled Cp's (parasitic capacitance) often mask the touch reading leading to loss of sensitivity.
- Coupled noise coming from signal lines and/or electronic radiated emitters in the proximity lead to unstable reading systems.

As a consequence, the problems reported above impact the final performance of the touch control system, both in normal operation and during EMC assessment.

CTSU2 offers several improvements that help increase performance and reduce these potential problems.

- Improved noise immunity
  - Improved HW IP
  - Multiple frequency scan
  - Shield electrode support
- Improved sensitivity
  - Improved ICO to manage temperature drift
  - Internal Temperature Drift correction
- New features
  - Parallel scan to support multiple touch via frequency converter
  - Safety function included

1.3. Requirements and specifications

Here is a list of software and tools used in the project:

- e² studio version 2021-07 (21.7.0): Eclipse-based Renesas integrated development environment (IDE).
- Renesas FSP Smart Configurator Core (8.5.0.v20210624-1718): Enhanced software package designed to provide easy-to-use, scalable, high-quality software for embedded system designs using Renesas RA Family of Arm Microcontrollers.
- QE for Capacitive Touch (1.2.0.v20201223-0532): Development assistance tool for Capacitive Touch Sensors.
- CS+ (V6.01.00): Formerly CubeSuite+, an integrated development environment.
- Workbench6 Software (1.07.00.00): The IDE used with Renesas RX 32-bit microcontrollers for capacitive touch applications.

- TouchAPI version 2018/01/22-0A.

The project consists of verifying different tests, so there are no specific margins or values to be reached in order to evaluate it.

1.4. Methods and procedures

As mentioned before, the project aims to compare the technology used on a current product of the company, so it will be performed in the framework of its development project.

There exists the T4 touch user interface and there is a current existing board based on RX130 microcontroller and CTSU touch technology.

For RA, the development environment and library are different. The target is to develop the firmware needed to configure the test setups and make the measurements, so there is a reference, but the adaptation will be done from scratch as well as the tests, as they will have to be adapted for making the same measurements and have conclusive results.

1.5. Work plan with tasks, milestones and a Gantt diagram

a) Work Packages

<table>
<tr><td>Project: Microcontroler family and tools knowledge acquisition</td><td colspan="2">WP ref: (WP1)</td></tr>
<tr><td>Major constituent: SW</td><td colspan="2">Sheet 1 of 3</td></tr>
<tr><td rowspan="2">Short description:
Knowledge to be acquired in different fields (specified in internal tasks)</td><td colspan="2">Planned start date: 04/10/2021
Planned end date: 26/12/2021</td></tr>
<tr><td colspan="2">Start event: 04/10/2021
End event: 26/12/2021</td></tr>
<tr><td>Internal task T1: Microcontroler family knowledge
Internal task T2: SW Life cycle development tools. JIRA/Git
Internal task T3: Use of Renesas e2studio IDE platform for the development/configuration of the software
Internal task T4: Use of Renesas QE for Capacitive touch, the development assistance tool for capacitive touch sensors
Internal task T5: Use of On chip debugger (Renesas E2 Lite) in the process of development/debugging
Internal task T6: Use of an oscilloscope for electronic inspection during development
Internal task T7: Use of EMC specific equipment for analysing system response in hazardous environments
Internal task T8: Test Report elaboration</td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

Project: RA software development	WP ref: (WP2)	
Major constituent: SW	Sheet 1 of 3	
Short description: To adapt the software to the new board according to the RA and CTSU2 specification, get basic functionality for a board sample.	Planned start date: 11/10/2021 Planned end date: 15/12/2021	
	Start event: 11/10/2021 End event: 15/12/2021	
Internal task T1: Self touch buttons configuration Internal task T2: Self tests SW adaptation Internal task T3: Mutual touch buttons configuration Internal task T4: Mutual tests SW adaptation Internal task T5: Multi-frequency SW adaptation Internal task T6: Parallel scanning SW adaptation Internal task T7: Board configuration	Deliverables: RA software	Dates: 15/12/2021

Project: RX software adaptation	WP ref: (WP3)	
Major constituent: SW	Sheet 2 of 3	
Short description: To adapt current software for testing SW	Planned start date: 01/11/2021 Planned end date: 15/12/2021	
	Start event: 15/10/2021 End event: 02/11/2021	
Internal task T1: Current SW understanding Internal task T2: Self tests SW adaptation Internal task T3: Mutual tests SW adaptation	Deliverables: RX software	Dates: 02/11/2021

Project: Tests in former CTSU configuration	WP ref: (WP4)	
Major constituent: simulation	Sheet 2 of 3	
Short description: Injected currents and fast transients.	Planned start date: 15/11/2021 Planned end date: 28/11/2021	
	Start event: 15/11/2021 End event: 28/11/2021	
Internal task T1: Self measurement parameters for RX Internal task T2: Mutual measurement parameters for RX Internal task T3: Self measurement parameters for RA Internal task T4: Mutual measurement parameters for RA	Deliverables: Former CTSU tests	Dates: 28/11/2021

Project: Tests configuring multi-frequency scan	WP ref: (WP5)	
Major constituent: simulation	Sheet 2 of 3	
Short description: Comparative test with and without multi-frequency	Planned start date: 29/11/2021 Planned end date: 12/12/2021	
	Start event: 29/11/2021 End event: 12/12/2021	
Internal task T1: Tests setup Internal task T2: Run test Internal task T3: Test analysis and parameterization Internal task T4: Gathering tests results	Deliverables: Multi frequency scan tests	Dates: 12/12/2021

Project: Tests configuring parallel scanning function	WP ref: (WP6)	
Major constituent: simulation	Sheet 3 of 3	
Short description: Comparative test with and without parallel scanning	Planned start date: 13/12/2021 Planned end date: 26/12/2021	
	Start event: 13/12/2021 End event: 26/12/2021	
Internal task T1: Tests setup Internal task T2: Run test Internal task T3: Test analysis and parameterization Internal task T4: Gathering tests results	Deliverables: Parallel scan tests	Dates: 26/12/2021

Project: Tests configuring EMC immunity level	WP ref: (WP7)	
Major constituent: simulation	Sheet 3 of 3	
Short description: Apply EMC test and verify immunity levels according IEC standards	Planned start date: 27/12/2021 Planned end date: 09/01/2022	
	Start event: 27/12/2021 End event: 09/01/2022	
Internal task T1: Tests setup Internal task T2: Run test Internal task T3: Test analysis and parameterization Internal task T4: Gathering tests results	Deliverables: EMC tests	Dates: 09/01/2022

Project: Final conclusion	WP ref: (WP8)	
Major constituent: documentation	Sheet 3 of 3	
Short description: Set the benefits and drawbacks of using the new technology and the future possibilities based on the run tests and conclusions.	Planned start date: 10/01/2022 Planned end date: 21/01/2022	
	Start event: 10/01/2022 End event: 21/01/2022	
Internal task T1: Tests comparison Internal task T2: Reach conclusion Internal task T3: Conclusion documentation	Deliverables: Final conclusion	Dates: 21/01/2022

Table 1. Work packages

b) Milestones

Here is a list of the different checkpoints followed during the project:

WP#	Task#	Short title	Milestone / deliverable	Date (week)
2	7	RA software	Software	15/12/2021
3	5	RX software	Software	02/11/2021
4	4	Former CTSU tests	Tests results	28/11/2021
5	4	Multi frequency scan tests	Tests results	12/12/2021
6	4	Parallel scan tests	Tests results	26/12/2021
7	4	EMC tests	Tests results	09/01/2022
8	3	Final conclusion	Final conclusion	21/01/2022

Table 2. Milestones

c) Time Plan (Gantt diagram)

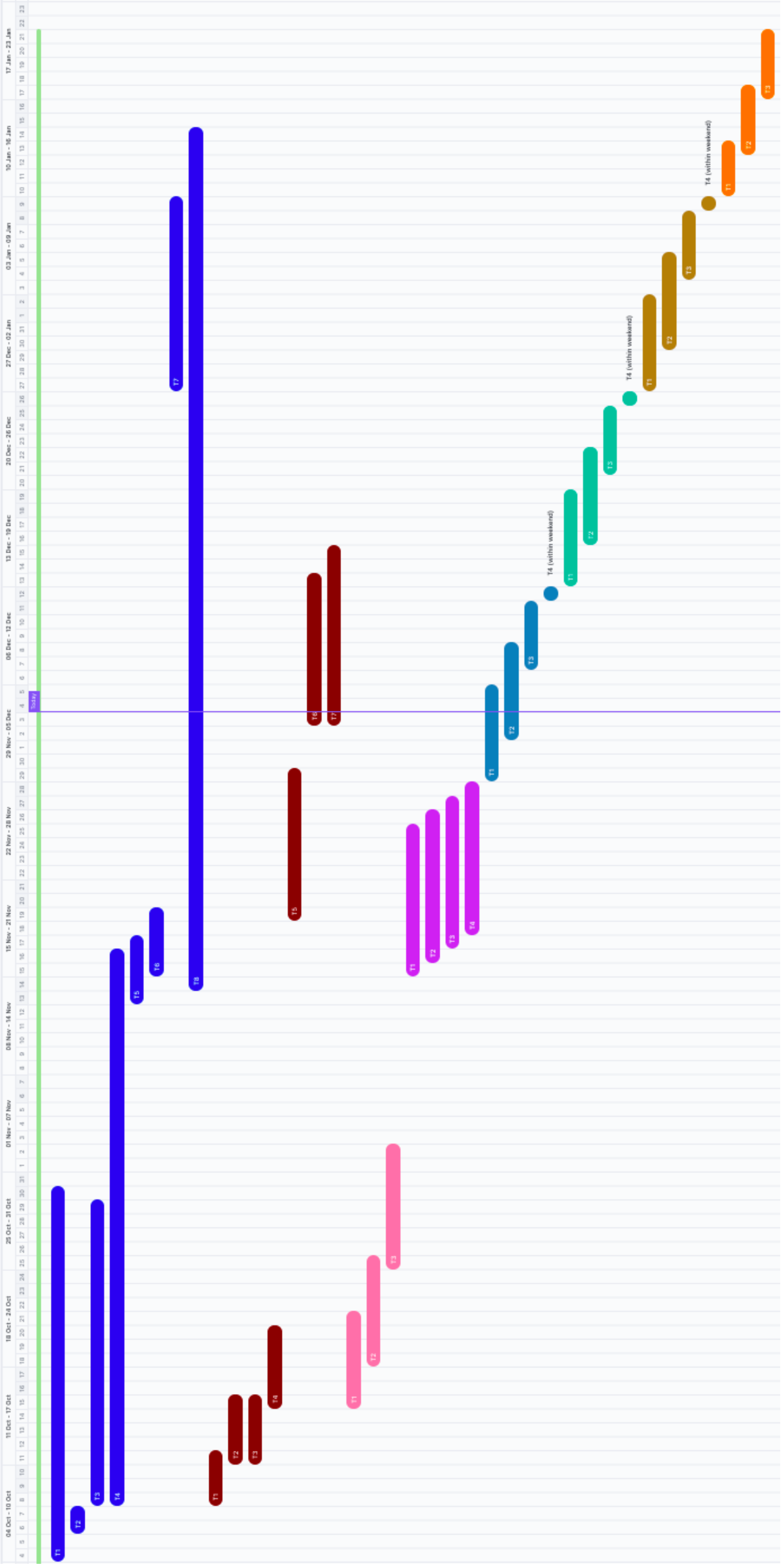


Figure 2. Gantt diagram

1.6. Initial plan deviations

The project had to be done with a specific board using a RA2E1 microcontroller (gathering the hardware of the touch screen that will be used on the washing machine) which was already requested by the company.

However, the specific board took longer than expected to arrive, so I started to make the software based on RA2L1 (quite similar to RA2E1) and this work could later be added to the final software. I also could focus on RX software adaptation ending earlier than expected.

Once the board arrived I started testing it, but I found problems with the connection.

There was a design mistake in the PCB and the person in charge of it was off work, making it longer for me to find the error.

I corrected the hardware connections and changed them to fit the project.

A redistribution of WP2 tasks had to be done, starting with buttons configuration and software adaptation and ending with board configuration (which originally was the first task).

2. State of the art of the technology used or applied in this thesis

2.1. Capacitive Touch Switch

The capacitive touch switch detects the status of the switch (ON or OFF) by measuring a small capacitive (1pF or less) change which exists between the electrodes and the human body. For the capacitive touch detection method which Renesas has developed, it adopts the switched capacitor filter in order to achieve high sensitivity and noise immunity, and judges the status of switch by converting the capacitance to current, amplifying and digitalization. (Figure 3)

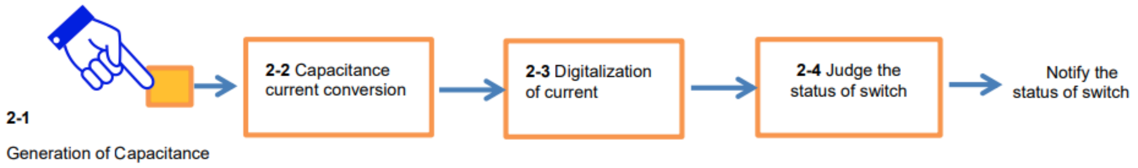


Figure 3. Flow of capacitance touch switch detection [2]

2.2. Self-capacitance Method

The mechanism of existing electrostatic capacity is shown in Figure 4. Parasitic capacity (*Cp*) exists between an electrode on the space and the conductive materials (ground pattern or metal frame, etc.) of its surroundings. At this time, finger capacity (*Cf*) is newly generated between the human body and electrode if the human body approaches, and it is grounded to the ground through the sole capacity (*Cs*) which is generated between the sole of feet and the ground through the human body as a conductor (the red line in the figure).

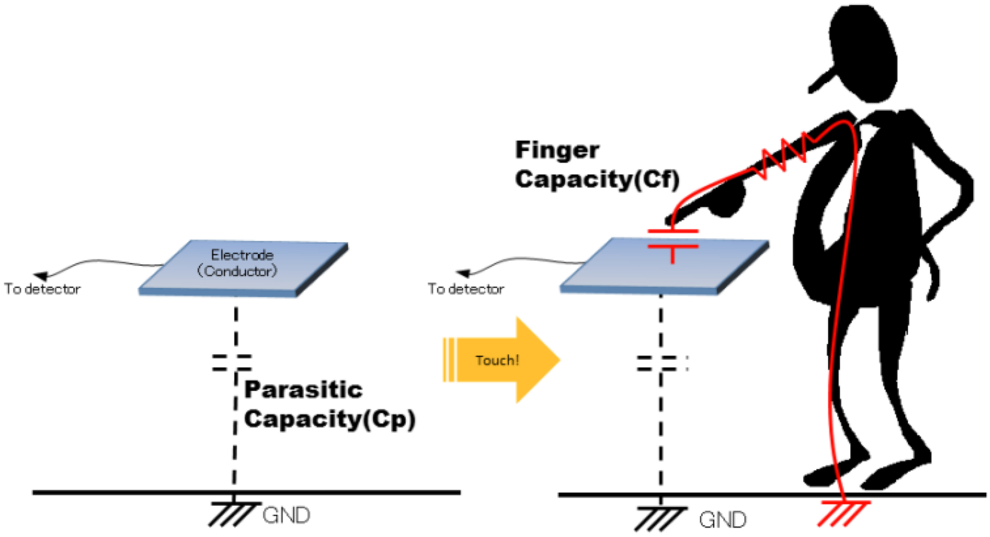


Figure 4. Generation of electrostatic capacity (self-capacitance method) [2]

Total capacity that is occurring on the electrode is shown in the following equation:

$$Total\,Capacity\ =\ Cp\ +\ Cf$$

The capacitive touch switch judges the status of the switch by detecting the increase of *Cf* of capacitance by the human body and measures the capacitance on the electrode by cyclic measurement. By setting a threshold for the amount of increase in *Cf*, you can determine whether the touch button is ON or OFF.

The CTSU outputs a digital count value proportional to capacitance C of the connected electrode, and determines whether the touch button is ON or OFF by software. When the electrode is connected to the CTSU, it performs as a switched capacitor controlled by the sensor drive pulse and estimates capacitance from the charge/discharge current to *C*. The CTSU measurement block has a current-frequency conversion function which inputs a current equivalent to the charge/discharge current and outputs a frequency proportional to the amount of current.

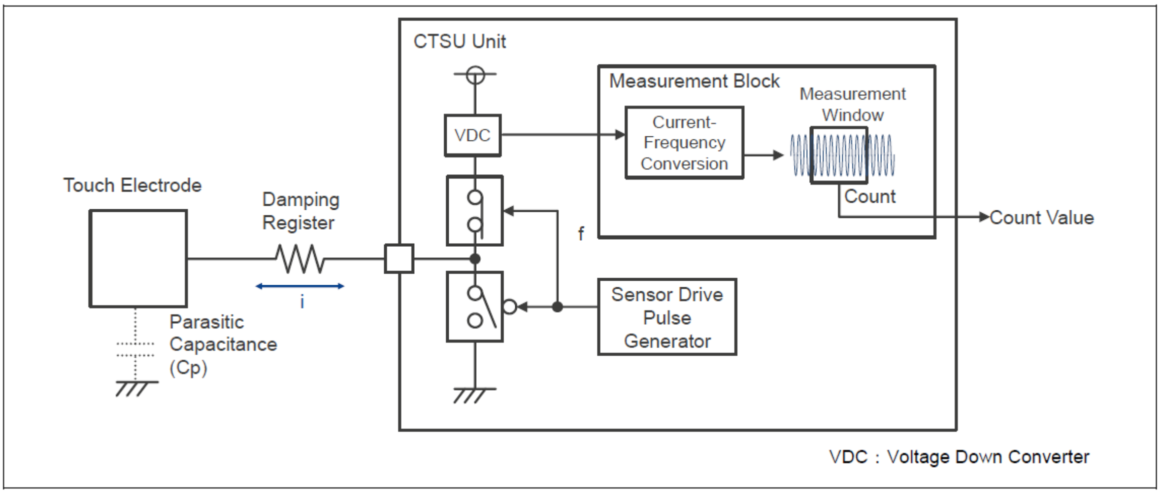


Figure 5. Internal Configuration Overview of Self-capacitance CTSU [1]

2.3. Mutual-capacitance Method

There is a mutual capacity method with a pair of electrodes in contrast with the self-capacity method with a single electrode. The example of the mutual capacity method is shown in Figure 6. The mutual capacity method is configured by receiving electrode, transmitting electrode, and pulse generators. When AC pulse is inputted to the transmitting electrode, Field Coupling is generated between the receiving electrodes. If a human body approaches in this state, the part of the electric field moves to the human body and the electric field between electrodes decreases.

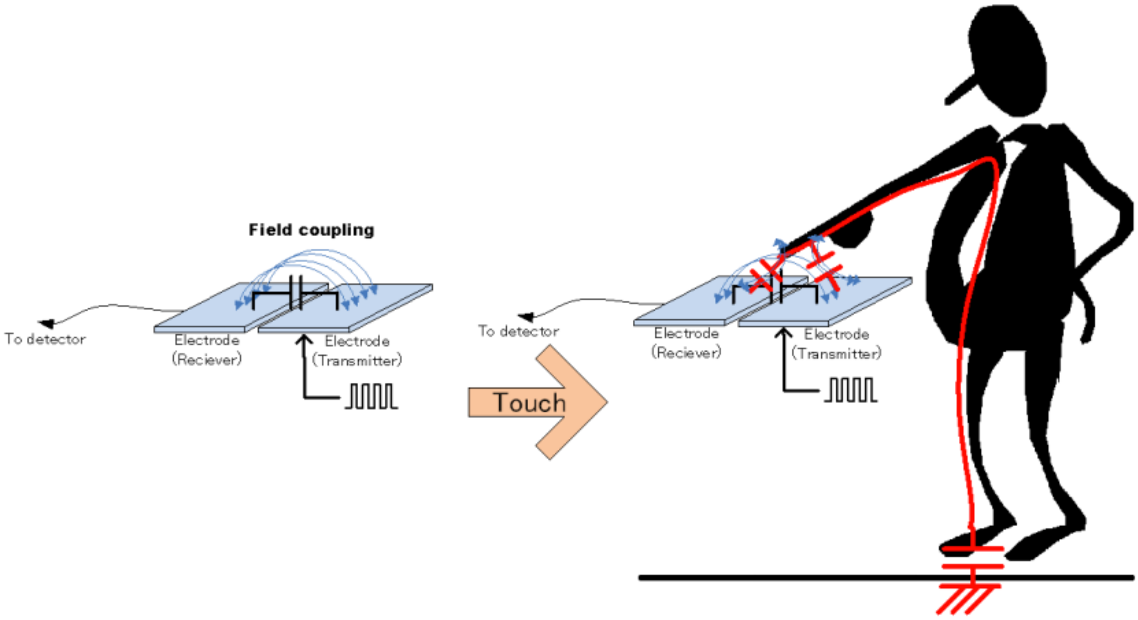


Figure 6. Mutual capacity method [2]

By setting a threshold for the amount of increase in Cf, you can determine whether the touch button is ON or OFF.

The CTSU outputs a digital count that is negatively proportional to the mutual capacitance of Rx and Tx connected to the electrode, and determines whether the touch button is ON or OFF by software.

In order to measure the capacitance Cm existing on the two connected electrodes, the CTSU obtains Cm by inverting the phase relationship between the pulse output and the switched capacitor, measuring the self capacitance twice, then calculating the difference of the two values by software.

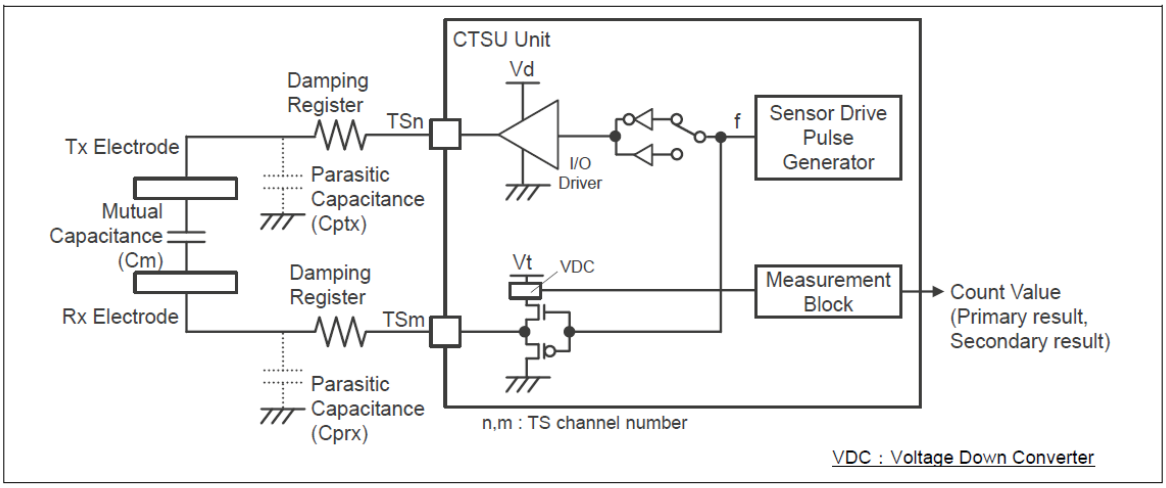


Figure 7. Internal Configuration Outline for Mutual-Capacitance Method [1]

2.4. Capacitance-Current Conversion

Switched capacitor filter (SCF) is structured by a capacitor, a power supply, two switches and the control signal to toggle the two switches in ON or OFF mode alternatively.

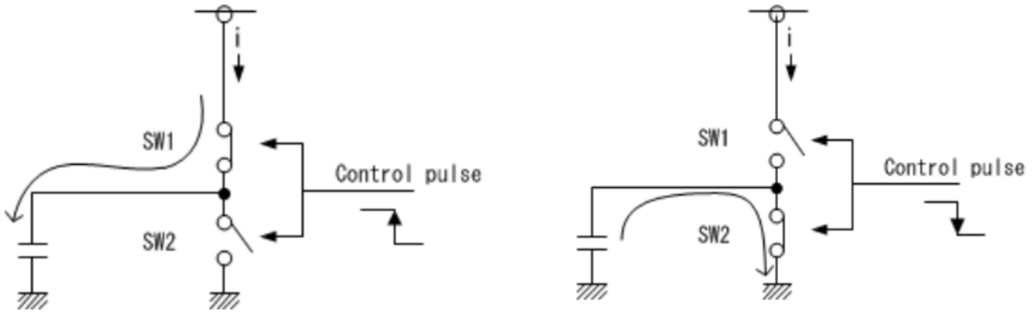
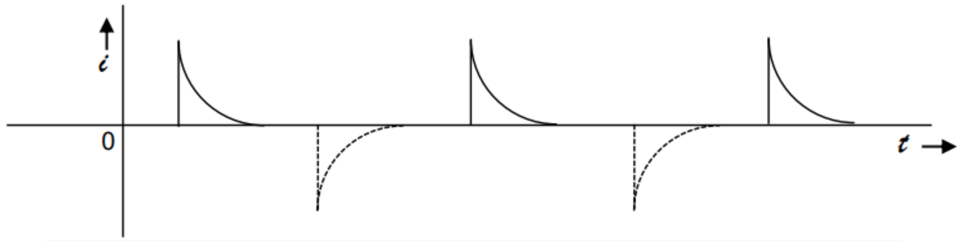


Figure 8. SCF configuration and Charge and discharge operation of capacitor [2]

When SW1 turns ON, SW2 turns OFF and the capacitor is charged as described in Figure 8 (left). After switching SW1 to OFF, SW2 is switched to ON, and the capacitor is discharged as described in Figure 8 (right).



SW1	ON	OFF	ON	OFF	OFF
SW2	OFF	ON	OFF	ON	ON

Figure 9. Status of SW1 and SW2 and the relationship of electric current i [2]

The electric charge of the capacitor is 0 at the moment SW1 turns ON. As the charge of the capacitor progresses, the amperage decreases, and the current stops at full charge. After that, the electric charge of the capacitor flows into the ground when SW2 turns ON. This current (Figure 9 dashed line) does not appear on the power supply side because SW1 is turned OFF. When cyclically repeats this operation, a certain amount of current flows intermittently synchronised with turning ON and OFF.

Amperage changes when changing the capacity of the capacitor or the ON/OFF cycle of the switches:

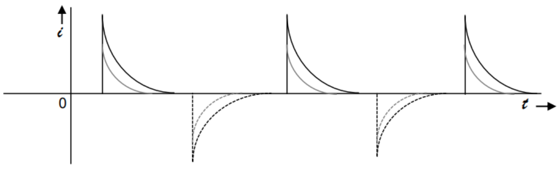


Figure 10. Double external capacitor capacity [2]

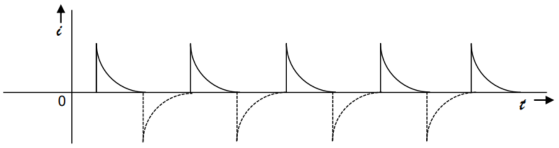


Figure 11. Double SW switching frequency [2]

From these relationships, the following equation holds by writing the circuit current as *i*, the switch frequency as *f*, capacitor capacity as *c*, and circuit voltage as *v*.

$$i = fcv$$

The conversion ratio of capacitance and amperage can be changed by adjusting *f* and *v*.

Figure 12 shows an image of a CTSU measurement. When one cycle of the sensor drive pulse frequency is shorter than the C charge/discharge time and the charge/discharge is insufficient, not enough current flows to C and the count value is smaller than the ideal value. When parasitic capacitance is large, it may be possible to take a measurement by lowering the sensor drive pulse frequency. When the sensor drive pulse frequency is lowered, the CTSU can measure a maximum of 50pF. Note that when the sensor drive pulse frequency is decreased, the number of measurements per unit time by the current-frequency conversion function also declines. The sensitivity of the electrode is likely to decrease as well. The unit time can be increased by adjusting the register setting value in the CTSU, but the amount of time required to complete the measurement will also increase. When designing a capacitive electrode circuit, conditions for button sensitivity, measurement time and noise immunity must be met.

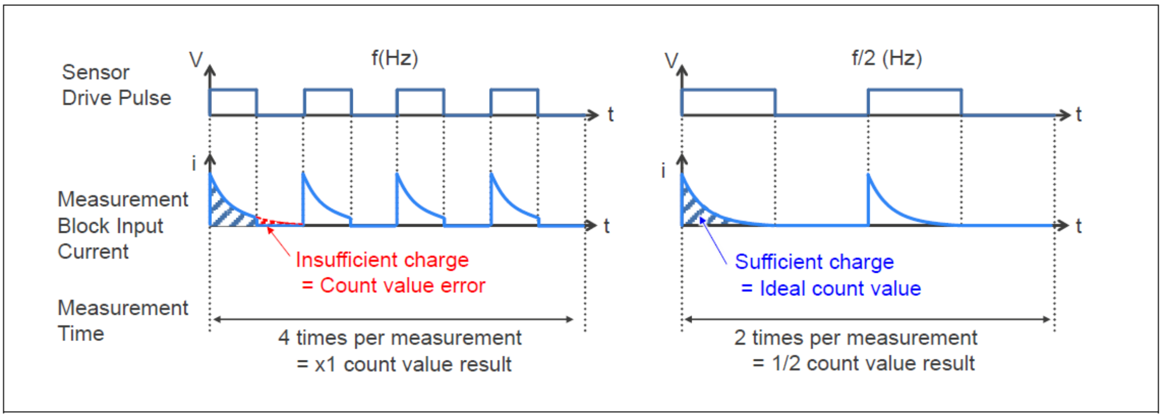


Figure 12. Image of CTSU Measurement [1]

2.5. Digitalization of Current

The capacity of the capacitor, which is converted to amperage, is digitalized by means of the circuit that changes oscillating frequency in accordance with amperage and the counter that counts pulse outputting from the circuit.

The flow of digitalization is shown in Figure 13. The electric current from SCF flows alternatively because the capacitor with SCF charges and discharges continually. This alternative current is smoothed by the power supply circuit connected to SCF. The current is sent to the current oscillator that varies oscillation frequency in proportion to amperage, and converted to oscillation frequency. This pulse is sent to the counter, and the counter holds it by measuring the number of pulses for a certain time. The graph in the figure shows the example when the frequency of SCF is fixed and the capacitor capacity is doubled, then the amperage and the frequency of the current oscillator become double. Eventually, the count value measured by the counter also doubles.

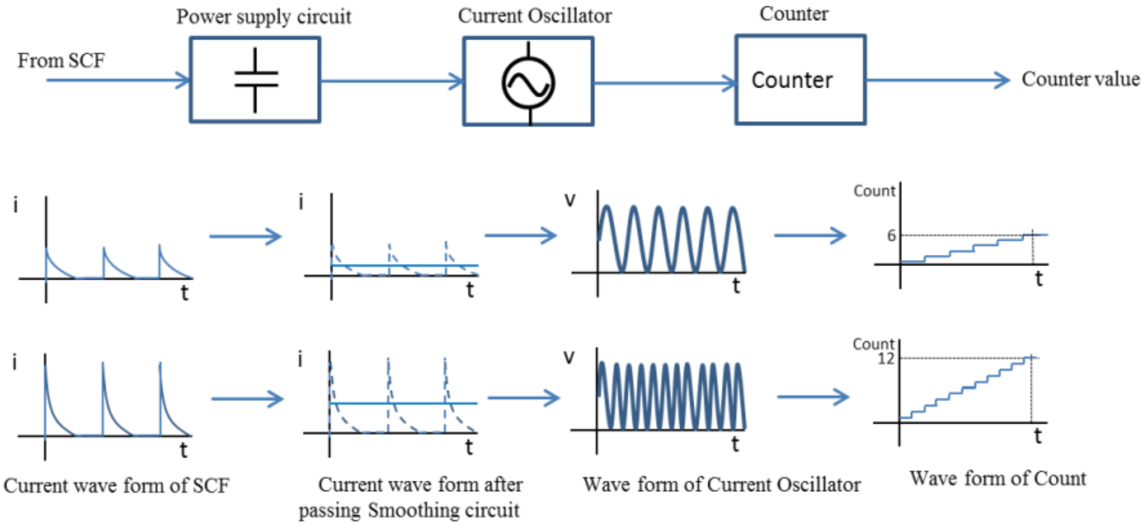


Figure 13. Flow of current digitalization [2]

Measurement of the capacitance is carried out at a regular interval as the measured timing shown in the figure. Count value obtained by measurement (blue line in figure 14) increases with the increase of capacitance when finger is approached, and decreases and becomes to a certain value when the finger is away again. At this time, set the count value when the finger is away as a standard value (green dashed line) and the value that added a certain value from the standard value as threshold value. Then, it will be possible to switch ON/OFF as a capacitive touch switch by switching ON when the measured count value exceeds the threshold value and switching OFF when the value is under the threshold value.

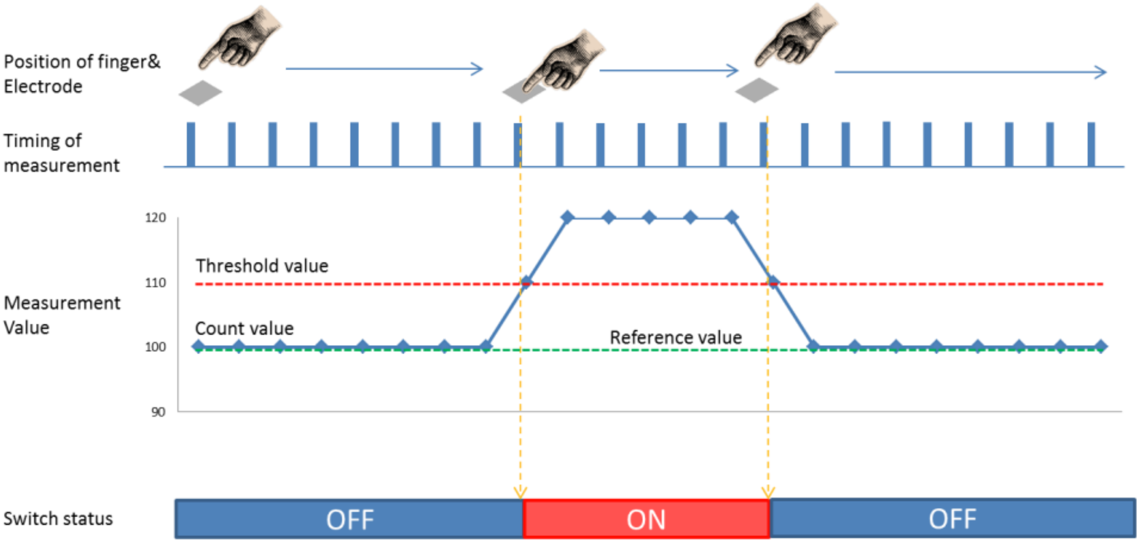


Figure 14. ON/OFF switch judgement [2]

The sensitivity adjustment of the capacitive touch switch can be done by changing the threshold value. Also, chattering suppression of the switch and the reaction rate can be adjusted by changing the cycle of measurement timing and averaging the multiple times of count value.

2.6. Mutual-capacitance parallel scan mode

This mode provides fast measurement time by parallel scanning the Rx lines with a CFC circuit. Operation is otherwise identical to normal CTSU mutual scanning.

- Scan Order
  - The hardware scans all Rx pins simultaneously for each Tx pin.
  - For example, if sensors TS10, TS11, and TS03 are specified as Rx sensors, and sensors TS02, TS07, and TS04 are specified as Tx sensors, the hardware will scan them in the following sensor-pair order: TS02-(TS03, TS10, TS11), TS04-(TS03, TS10, TS11), TS07-(TS03, TS10, TS11)
- Element
  - An element refers to the index of a sensor-pair within the scan order. Using the previous example, TS07-TS10 is element 7.
- Scan Time
  - Because the Rx lines are scanned in parallel, CFC mutual-capacitance scan is the same amount of times faster than a basic mutual matrix scan as the number of Rx lines. In other words, on a matrix with N receive lines, CFC mutual scanning is N times faster than basic mutual scanning.

2.7. Multi-frequency Measurements

The CTSU2 peripheral takes measurements at three different frequencies to avoid synchronous noise. After standardising the results obtained at the three frequencies in accordance with the first frequency reference value, the measured value is determined based on majority in a process referred to as "normalisation."

The three values standardised to the first frequency reference value are called correction data.

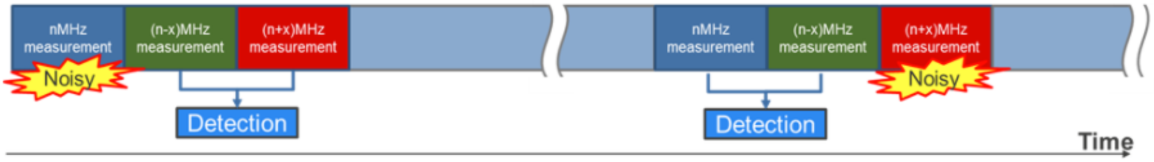


Figure 15. Multi-frequency Measurements [6]

2.8. Debounce

When pressing a button the voltage may fluctuate between states several times during the transition period. Debounce is used to remove that small ripple of current that makes a series of short contacts, by waiting a specified amount of time between sampling periods.

2.9. EMC (Electromagnetic compatibility)

2.9.1. Electrical fast transient/burst immunity test

Its objective is to evaluate the immunity of electrical and electronic equipment when subjected to electrical fast transient/bursts (sequence of a limited number of distinct pulses or an oscillation of limited duration) on supply, signal, control and earth ports.

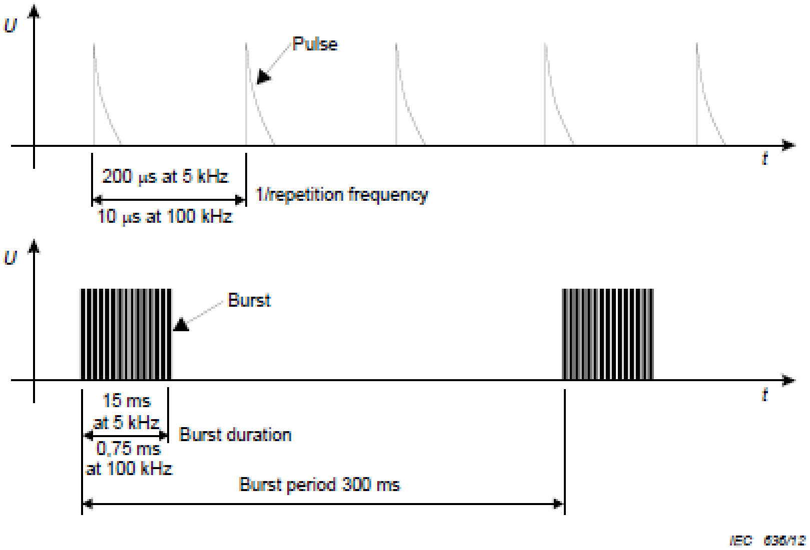


Figure 16. Representation of an electrical fast transient/burst [7]

2.9.2. Immunity to conducted disturbances, induced by radio-frequency fields

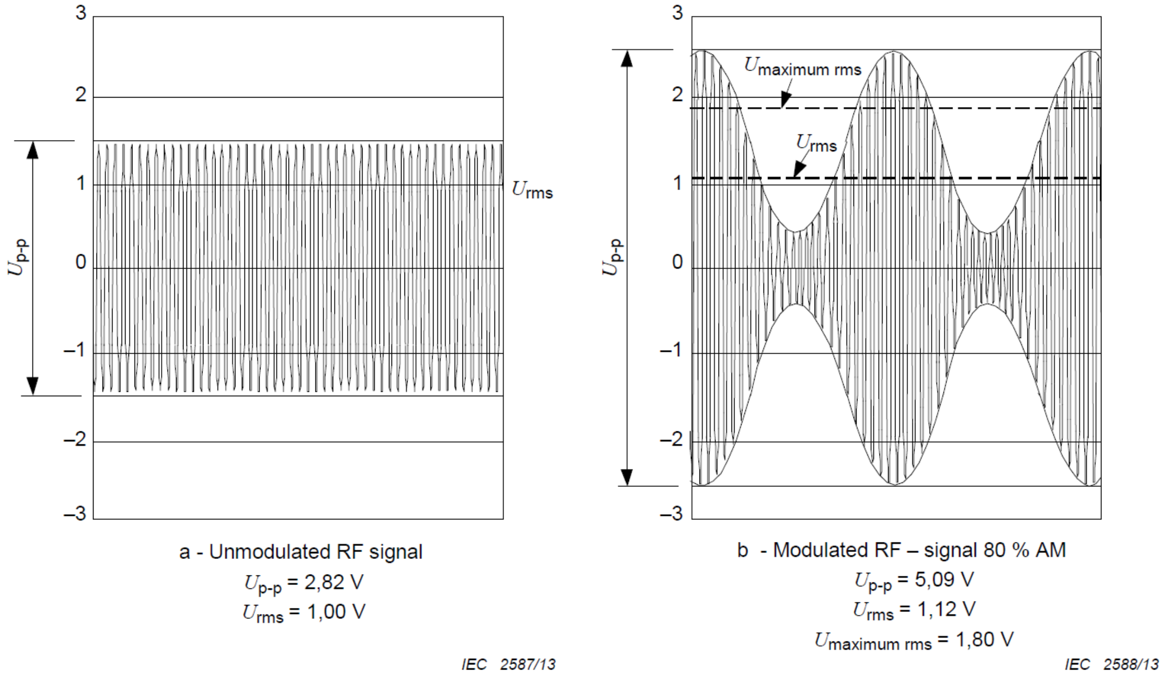
Its objective is to evaluate the functional immunity of electrical and electronic equipment to electromagnetic disturbances coming from intended radio-frequency (RF) transmitters in the frequency range 150 kHz up to 80 MHz.

The open circuit test levels (e.m.f.) of the unmodulated disturbing signal, expressed in r.m.s., are given in Table 3.

Frequency range 150 kHz to 80 MHz		
Level	**Voltage level (e.m.f.)**	
	Uo V	Uo dB (µV)
1	1	120
2	3	129,5
3	10	140
X[a]	Special	
[a] "X" can be any level, above, below or in between the others. The level has to be specified in the dedicated equipment specification.		

Table 3. Test levels [8]

The test levels are set at the EUT port of the coupling devices. For testing of the equipment, this signal is 80 % amplitude modulated with a 1 kHz sine wave to simulate actual threats. The effective amplitude modulation is shown in Figure 17.



a - Unmodulated RF signal
$U_{\text{p-p}}$ = 2,82 V
U_{rms} = 1,00 V

b - Modulated RF – signal 80 % AM
$U_{\text{p-p}}$ = 5,09 V
U_{rms} = 1,12 V
$U_{\text{maximum rms}}$ = 1,80 V

Figure 17. Open circuit waveforms at the EUT port of a coupling device for test level 1 [8]

2.10. FINE Interface

FINE is used to communicate with the serial programmer in boot mode (FINE Interface).

Figure 18 shows an Example of Pin Connections in Boot Mode (FINE Interface).

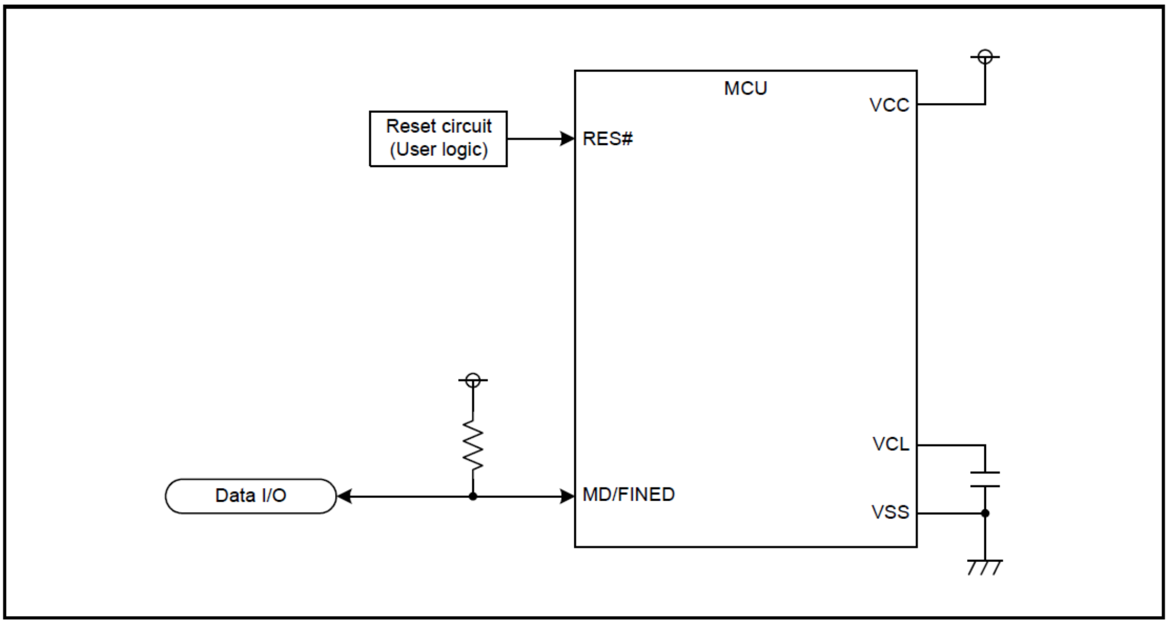


Figure 18. Example of Pin Connections in Boot Mode (FINE Interface)

2.11. SWD Interface

SWD Interface Connection Figure 19 shows a recommended circuit for connection through the SWD interfaces.

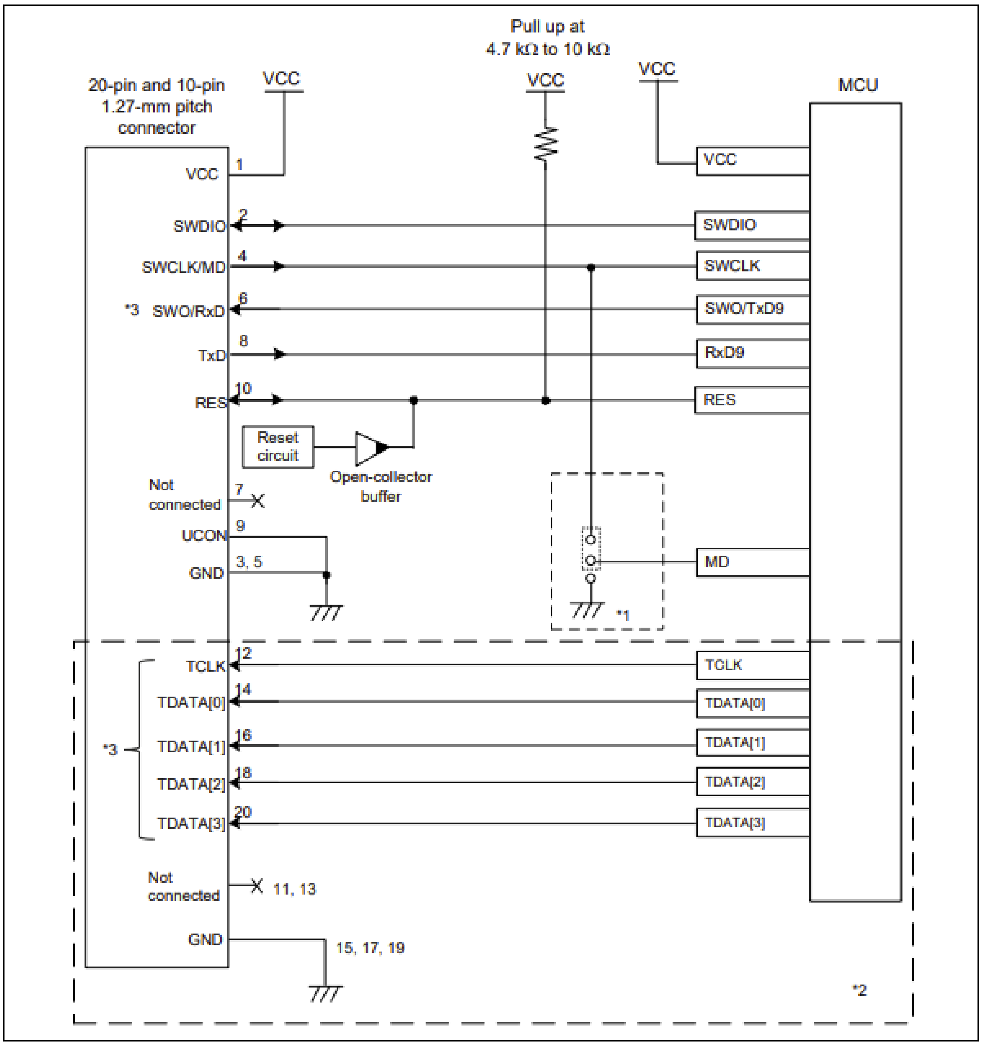


Figure 19. Example of Connection through the SWD Interfaces

Notes:

1. For DLM control, the MD pin must be connected to the emulator. When DLM control is not required, the MD pin can operate even if it is not connected to the emulator. When the MD pin is not connected to the emulator, a special circuit for the pin must be configured on the user system. For details on handling of the MD pin, refer to section2.5,Notes on Connection.
2. If a 10-pin connector is mounted on the user system, pins 11 to 20 are not used.
3. E2 Lite emulator does not support outputting trace data in Emulator Debugger. On the other hand, E2 emulator only handles outputting trace data from SWO pin.

3. Methodology / project development

The initial part of the project gathers some knowledge acquisition as well as some software/hardware tests in order to understand the state of the art of the technology used with practical results. For this purpose, RA2L1 microcontroller has been used on Capacitive Touch Evaluation System with Self-capacitance Button/Wheel/Slider Electrode Board and Mutual-capacitance Button/Proximity Sensor Board.

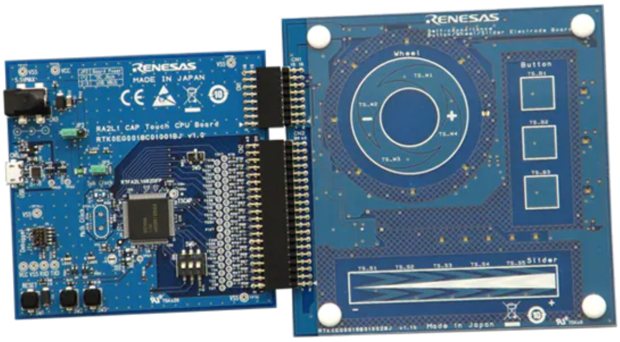


Figure 20. CAP Touch CPU Board + Self-capacitance Button/Wheel/Slider Electrode Board

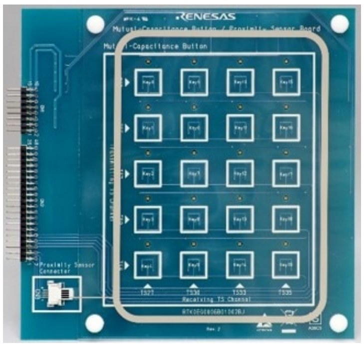
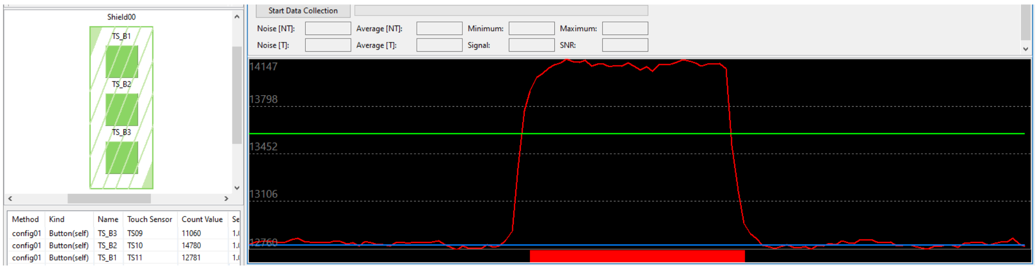


Figure 21. Mutual-capacitance Button/Proximity Sensor Board

The first test was to achieve a basic configuration of some self buttons. The next figure shows the monitoring of one of the three configured buttons. Measurement count (red), threshold (green) and reference (blue) values can be identified as well as the "pressed button" state (red bar).



Method	Kind	Name	Touch Sensor	Count Value	Se
config01	Button(self)	TS_B3	TS09	11060	1.
config01	Button(self)	TS_B2	TS10	14780	1.
config01	Button(self)	TS_B1	TS11	12781	1.

Figure 22. Self button monitoring

When monitoring different channels at the same time different calibrations of them can be seen, setting each of them in a different offset. The sequence of pressed buttons on the next image is the button TS_B3 (yellow), TS_B2 (green) and TS_B1 (red), keeping the last one pressed.

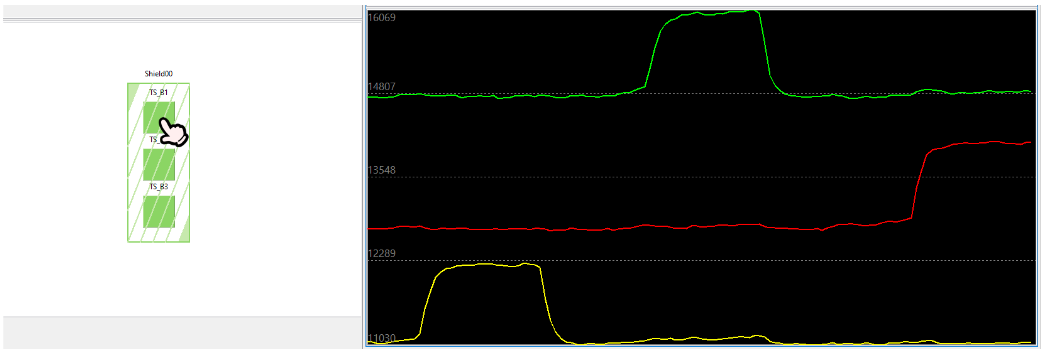


Figure 23. Self button sequence monitoring

The next step is to compare self buttons' behaviour with mutual buttons. In order to see the effect of Rx and Tx sensor pins the configuration will be a square of four buttons.
The next figure shows the configuration of the channels, where TS00 and TS08 are Tx and TS23 and TS32 are Rx.

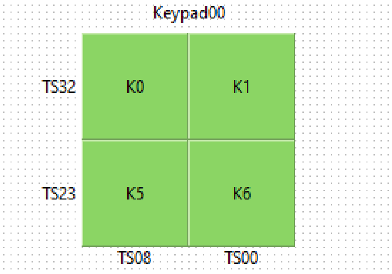


Figure 24. Mutual 2x2 configuration

As expected, in figure 25 it can be seen that the button count is negatively proportional to the mutual capacitance of Rx and Tx. So, when using mutual buttons, the "pressed" status will be when the count value crosses the threshold negatively.

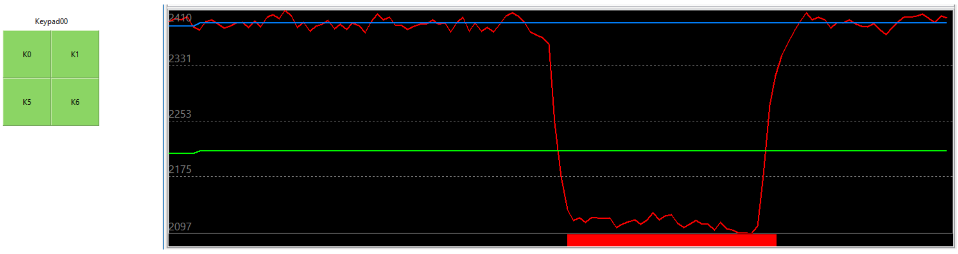
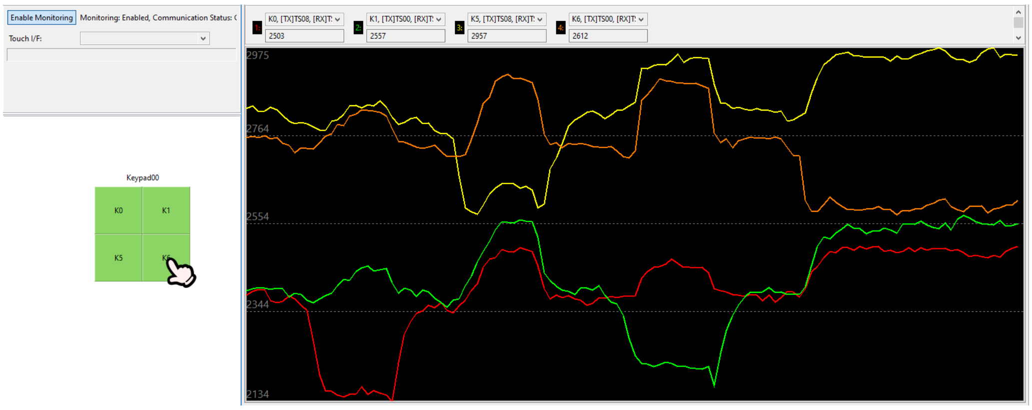


Figure 25. Mutual button monitoring

The effect on the other channels when pressing one of them is shown in figure 26. The sequence of touch of the next image is the button K0 (red), K5 (yellow), K1 (green) and K6 (orange), keeping the last one pressed. It can be noticed that the sensor count value increases on the non-touched channels because of the proximity of each to the other.


Monitoring: Enabled, Communication Status: (

1: K0, [TX]TS08, [RX]T: 2503
2: K1, [TX]TS00, [RX]T: 2557
3: K5, [TX]TS08, [RX]T: 2957
4: K6, [TX]TS00, [RX]T: 2612

Figure 26. Mutual button sequence monitoring

Some measurements can be done with the QR tool. Nevertheless, to gather the same kind of measurements, the comparative SNR will be calculated with a specifically developed software, as the two projects use different frameworks and we can not use this tool on both of them.

For calculating measurement times, a pin toggle has been used, and this toggle has been visualised with the oscilloscope (yellow measurement). There is also the touch sensor pin representation (blue signal) where we can see three different measures, as this configuration uses only one self channel with multifrequency (three different frequencies).

This blue representation is an invasive method, as it is being taken from the touch pin itself, but it allows us to see a nearly representation of the signal and a conceptual view.

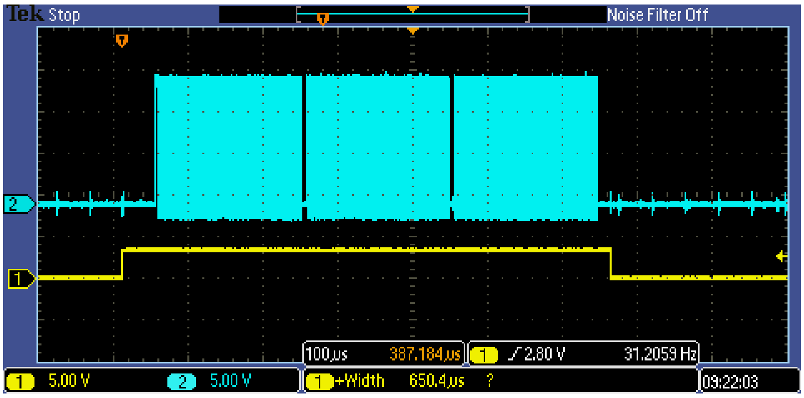


Figure 27. One button with multifrequency configuration measurement time

When measuring a mutual button on a configuration of four mutual buttons square (figure 28), it can be seen that two buttons share the Tx pin, so we can observe two different measures of it (with three measurements on each because of multifrequency). In the case of the Rx pins (figure 29), all measures in the same Rx are together.

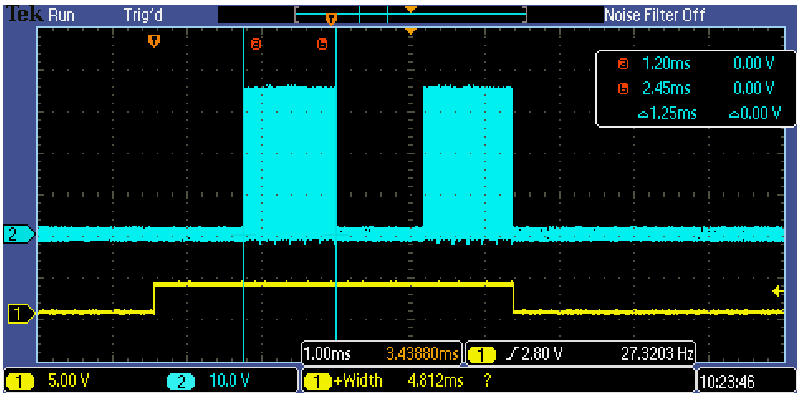


Figure 28. Four buttons with multifrequency configuration measurement time (Tx pin)

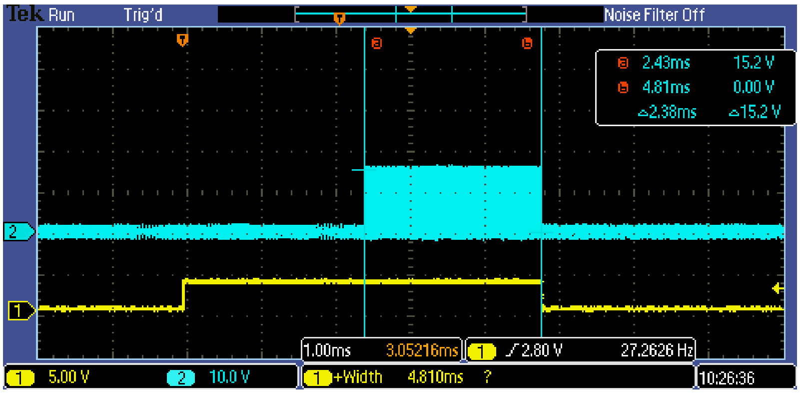


Figure 29. Four buttons with multifrequency configuration measurement time (Rx pin)

In the case of parallel scan, Tx has just one measurement for each pin (apart from multifrequency), as we measure together all Rx pins for each Tx pin.

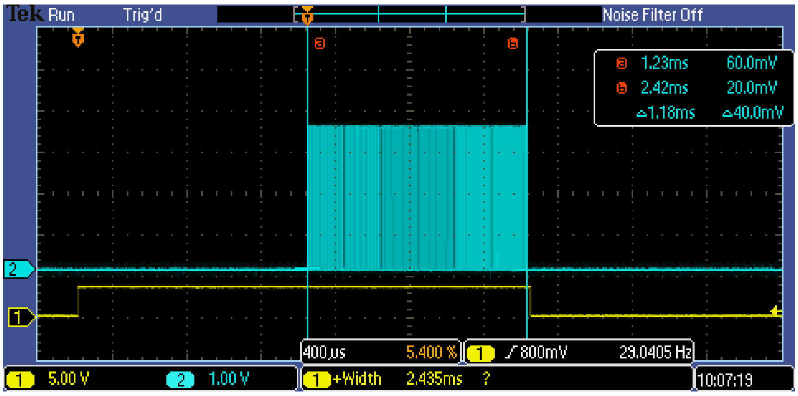


Figure 30. Four buttons with multifrequency and parallel scan configuration measurement time (Tx pin)

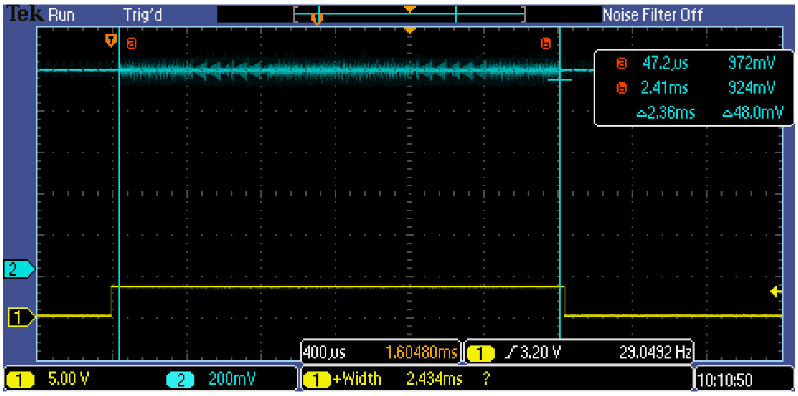


Figure 31. Four buttons with multifrequency and parallel scan configuration measurement time (Rx pin)

A table of different measurement times for the different button configurations has been made:

Button number	**1**	**2 (1x2)**	**3 (1x3)**	**3 (3x1)**	**4 (2x2)**	**5 (1x5)**	**9 (3x3)**
Self	650us	1.24ms	1.84ms	-	-	-	-
Mutual	1.24ms	2.43ms	3.62ms	3.62ms	4.81ms	6ms	10.75ms
Mutual + parallel scan	1.25ms	2.43ms	3.62ms	1.25ms	2.43ms	6ms	3.64ms

Table 4. RA measurement times comparative

Note: button number (Rx x Tx)

It can be noticed that for mutual buttons the time is doubled in respect of on self buttons, because two pins (Rx and Tx) are being scanned for each channel. When using parallel scan the time is reduced if there are multiple Rx pins for the same Tx pin, and this reduction is in order of how many of these Rx are.

3.1. Board configuration for measurements

The first thing to look at on the RA2E1 microcontroller board was the connection, as it was not communicating with the debugger. RA2E1 was expected to support FINE connection but it does not, so another debugging interface had to be used. For implementing SWD Interface Connection some of the pins had to be changed (see section 1 of appendices for resulting schematic).

The next thing to look at were the used ports for its configuration on the code (see section 2 of appendices to see the resulting touch foil of both boards).

For SNR and time tests, of all the channels on the board, only 2 self buttons and 3 mutual buttons will be used. This configuration allows us to compare SNR results between buttons and also allows us to compare times. Mutual buttons must be parallel in reception in order to compare parallel scan from the configuration without parallel scan, as parallel scan works scanning several reception ports within scanning the same transmission port.

So the figure 32 represents the configuration for the project with RX130 microcontroller, where S49 (TS01) and S48 (TS25) are self buttons, TS22 is a transmitter pin and TS10, TS09 and TS08 are receiver pins, resulting on the mutual buttons S2, S10 and S18.

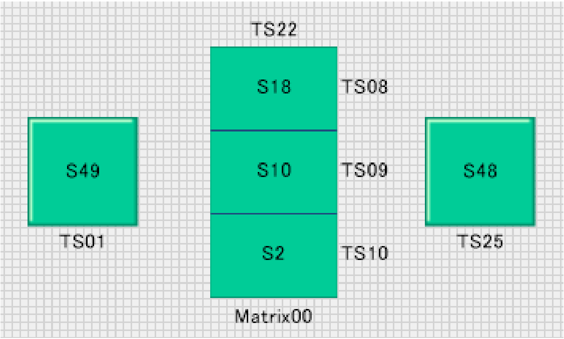


Figure 32. Two self, three mutual configuration for RX

Then, to have an equivalent configuration while using RA2E1 it would result like the following figures:

Without parallel scan:

With parallel scan:

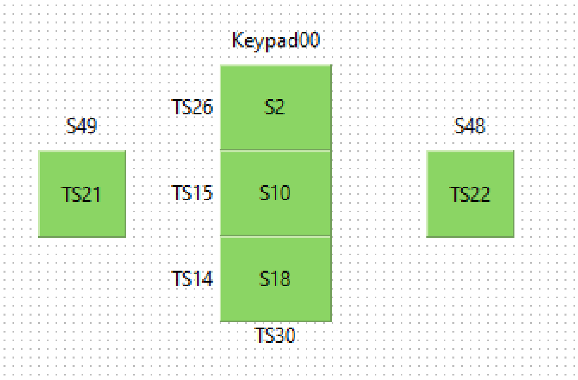


Figure 33. Two self, three mutual without parallel scan configuration for RA

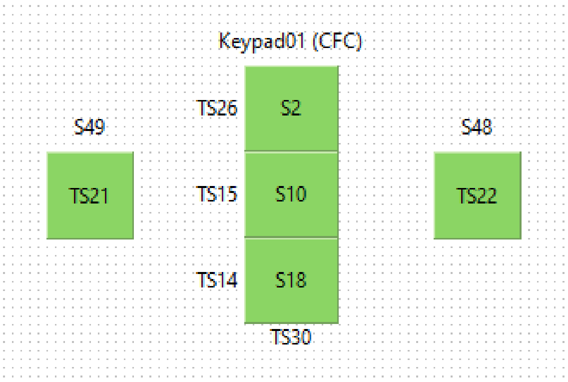


Figure 34. Two self, three mutual with parallel scan configuration for RA

In figures 33 and 34, self channel S49 is the TS21 pin and S48 is TS22. On mutual channels the transmitter pin is TS30 and the receiver pins are TS26, TS15 and TS14. On parallel scan configuration we can see that CFC configuration is activated.

This configuration should be compared with and without multifrequency and parallel scan. With the equivalent scan time that we have on RX130 configuration, the measurement exceeds its maximum size when using multifrequency because, as we have seen in section 2.7, two detection values are added resulting in a high measurement value. This, added to having a higher value because of the scan time, results in overflow.
With this being said, a new comparison must be done decreasing the scan time. When decreasing the scan time, it will decrease the sensitivity because it compresses the integration period.

All this brings us to this different configurations to be compared:

- RX
- RA with RX scan time
- RA with lower scan time
- RA with multifrequency scan
- RA with parallel scan
- RA with multifrequency and parallel scan

As the project consists of a comparison between an existing board and a new version of it with a different microcontroller, the next step is to understand the original configuration and extract the information needed to implement an equivalence in the new board.

Config.	**RX / RA Single freq. with high scan time**					**RA Single freq. / RA Multifreq.**					**RA Single freq. + parallel / RA Multifreq. + parallel**				
Channel	S49	S48	S18	S10	S2	S49	S48	S18	S10	S2	S49	S48	S18	S10	S2
DPF	4	4	1	1	1	4	4	1	1	1	4	4	0.25	0.25	0.25
ScT	576	576	1152	1152	1152	128	128	256	256	256	128	128	256	256	256

Table 5. Parameters comparative for each configuration

Notes: DPF: Drive Pulse Frequency (MHz)
ScT: Scan Time (us)

3.2. Sensibility measurements

The following function has been used for SNR calculation:

$$SNR = \frac{max\ touch - reference}{noise} \quad \text{where} \quad noise = max\ no\ touch - min\ no\ touch$$

The sensor value has been extracted from the code. Several measurements have been made and the extracted sensor value has been compared to obtain the maximum and minimum value. From these values, noise can be obtained.

This noise will be calculated while the sensor is not in touch. Once we have saved these values, touch measurements need to be done in order to find *max touch* value and *reference* value. With these results we can obtain SNR using the previous function (see section 3 of appendices for the code example).

3.3. Time measurements

The same method as the one followed to obtain figure 27 is used to get the time measurements. A toggle is programmed and measured using an oscilloscope. In this case the pin itself is not measured, but all the channels *scan time* is measured (see section 3 of appendices for the code example).

For RX, the pin toggle is set up by implementing a *state change* right before the measurement starts, and another *state change* when the interrupt jumps (at the end of the measurement).

For RA, the first *state change* is implemented right before the measurement starts. Once this measurement ends, it results in a flag change and the other *state change* comes after it. In the end, it is not needed to check the end of the measure like it was with the RX.

3.4. Memory usage measurement

There is an option on both used frameworks (CS+ and e2 studio) which allows extracting the Memory Usage. This has been done for each of the different configurations in order to compare it.

3.5. EMC tests

On the next table are the two RA configurations chosen to do the EMC tests:

Configurations	**All channels scan time (seconds)**
RX	62,49
RA single frequency + parallel scan	4,95
RA multifrequency + parallel scan	12,89

Table 6. Measurement time for all channels on EMC chosen configurations

In order to compare both boards, debounce and threshold changes may be done by approximating the measurement time to 65 seconds. In this time, one measurement can be done with RX configuration, 13 measurements with RA single frequency + parallel scan configuration, and 5 measurements with RA multifrequency + parallel scan configuration, so the initial debounce parameters are set based on these values. Threshold is set to 60%.

The drift frequency is used to recalculate the reference value during the execution of the program, and refers to the number of samples between the reference value's recalculations. On the RA multifrequency with parallel scan configuration, this value is set to 150, as RA makes 5 times more samples than RX, which has a drift frequency of 30.

Electrical fast transient/burst immunity tests are configured to a voltage peak level of 4kV, a 100kHz repetition frequency and 60s steps.

Immunity to conducted disturbances, induced by radio-frequency fields tests, are set to a voltage level of 6V and a frequency range of 1MHz up to 10MHz.

The EMC immunity level results are considered passed if it does not detect any fake touch.

4. Results

After all the measurements, some tables are made in order to gather the information into a conclusive result.

Results can be summarised in SNR results (with and without leds activation), time results, memory results and EMC immunity level results.

As it has been said before, the evaluation is based in different configurations: in the first one the board is using the current microcontroller used by the company, RX130, while the other tests are using RA2E1 microcontroller, first using the equivalent values to the RX configuration, later reducing the scan time in order to balance the next results and, finally, applying multi frequency and parallel scan.

As leds add some noise, SNR results need to be found with and without the leds.

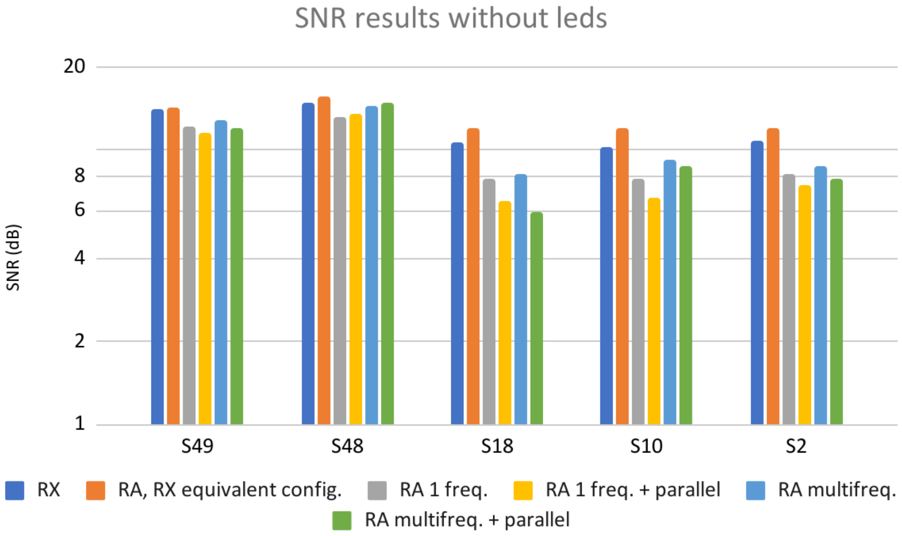


Figure 35. SNR results without leds

In the case of not using leds, we can see better results on self buttons than on mutual buttons. We can also see better SNR on RA2E1 than on RX130 when using the same scan time. When reducing the scan time we can see better SNR results when there is no parallel scan and when using multifrequency scan.

For SNR with leds comparison, we will only compare RA results, due to the different configurations of the leds with respect to RX.

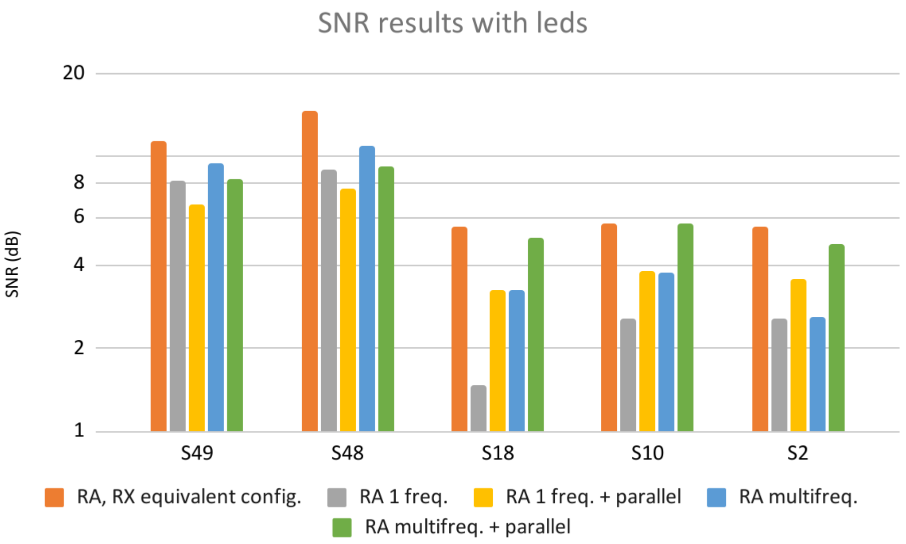


Figure 36. SNR results with leds

We can clearly see a decrease of the SNR value when using leds. The differences mentioned before between configurations are still the same but, in this case, are less noticeable because the SNR levels are closer now.

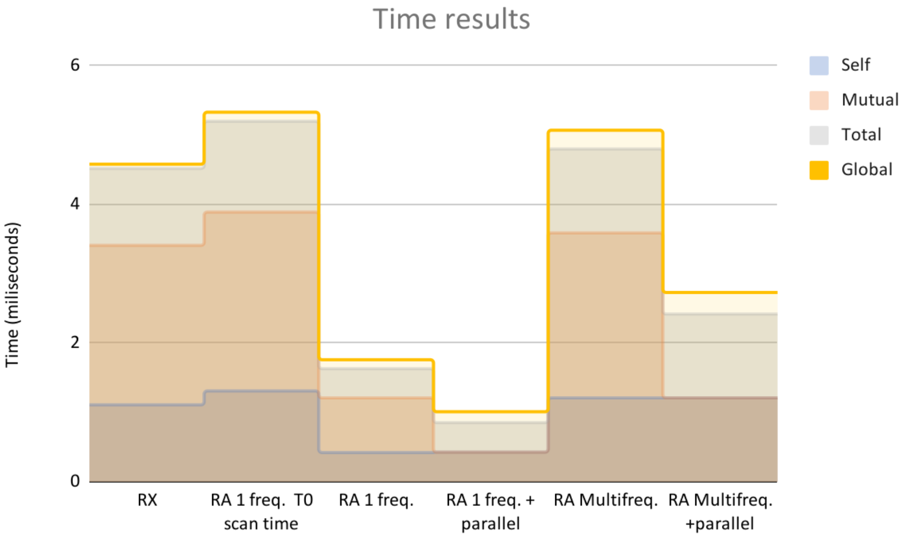


Figure 37. Time results

We can see that, with equivalent configurations, RX130 and RA2E1 use almost the same time for the measurements. When using multifrequency, the time is three times the time of one frequency measurement (because we are using three frequencies). When using parallel scan, the mutual measurement time is the third part of it when it is not using parallel scan, because we have three mutual buttons in parallel on reception and it scans all of them at the same time.

We can see that with an equivalent configuration, RA has better SNR than RX, using almost the same measurement time. When using additional configurations as multifrequency scan, SNR results increase. And when using parallel scan, it makes the measurement much faster, although SNR decreases.

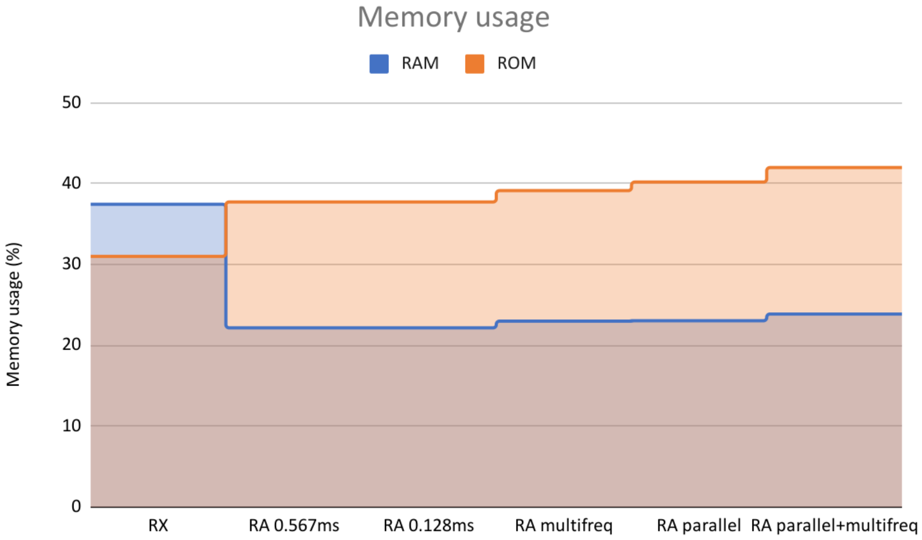


Figure 38. Memory usage results

It can be seen that memory usage does not change too much from one to the other configuration, when adding multifrequency and parallel scan the memory usage increases a little, but it will not be a decisive factor for the results of our comparison.

With the configurations set as specified in section 3.5, EMC immunity level results have been run for RA CTSU technology board.

This results, compared to RX CTSU technology board EMC tests results, are gathered on the next table:

EMC immunity level results	**RX**	**RA Single frequency + parallel**	**RA Multifrequency + parallel**
Electrical fast transient/burst immunity test	Passed	Passed	Passed
Immunity to conducted disturbances, induced by radio-frequency fields	Passed	Failed	Passed

Table 7. EMC immunity level results

With the single frequency with parallel scan configuration we have reduced the measurement time to an 8% in respect to RX, and SNR has decreased an average of 26% while passing burst EMC test, but not passing injected EMC tests.

With the multifrequency with parallel scan configuration we have reduced the measurement time to a 21% in respect to RX, and SNR has decreased an average of 20% while passing burst and injected EMC tests.

In the end, for the purpose of this board, the optimal configuration is to use multifrequency with parallel scan.

5. Budget

This project consists of a software development of a hardware board, so the budget will consist of software, hardware and engineering costs.

It is usual for some softwares to be free until some memory usage is exceeded. In this case, CS+ and Workbench were free, while e2studio had a cost of 1700€/year.

Software	**Price**	**Utilization (months)**	**Amortization (years)**	**Cost**
e2studio,1 user, PROFESSIONAL FLOATING LICENCE	1.700,00 €	4	1	566,67 €
CubeSuite+	0,00 €	-	-	-
WWB6 - Wireless Workbench® 6	0,00 €	-	-	-
			Total	566,67 €

Table 8. Software costs

The hardware part is divided in components and tools lists.

As the T4 (RX) board is already in production, several units are bought, resulting in a low price in comparison to the T4 (RA) board.

Components list	**Cost**
RA2L1 CAP Touch CPU Board + Self-capacitance Button/Wheel/Slider Electrode Board (RTK0EG0022S01001BJ)	140,00 €
Mutual-capacitance Button/Proximity Sensor Board - Touch Sensor Development Tools RX130 Capacitive Evaluation Kit (RTK0EG0003S02001BJ)	197,00 €
Renesas E1 emulator (R0E000010KCE00)	149,00 €
T4 board	15,00 €
Renesas E2 Lite (RTE0T0002LKCE00000R)	72,70 €
T4 (RA) board	200,00 €
Total	773,70 €

Table 9. Component costs

Tools	**Price**	**Utilization (months)**	**Amortization (years)**	**Cost**
PC	700,00 €	4	5	46,67 €
Oscilloscope	200,00 €	4	15	4,44 €
			Total	51,11 €

Table 10. Tools costs

Different engineering tasks have been implemented for the project, the table 11 summarises each part. It includes the Social Security costs that the company must assume.

Engineering	Hours	Salary	Total
Junior Software Development	405	9 €	3.645 €
Support for Software Development	20	57 €	1.140 €
Layout (PCB routing support)	35	30 €	1.050 €
Hardware Development	22	30 €	660 €
		Net total	6.495 €
		Social Security (33%)	2.143 €
		Total	8.638 €

Table 11. Engineering costs

This different parts result on the next amount for the final cost of the project:

Description	Cost
Software	566,67 €
Hardware	824,81 €
Engineering	8.638 €
Total	**10.029,83 €**

Table 12. Project total cost

6. Conclusions and future development

Creating a new software for the new T4 board using RA2E1 microcontroller has been accomplished. It is a functional board which can be implemented and used in future products of the company.

It has been demonstrated that we get a lot of advantages from using CTSU2 instead of CTSU. For the purpose of this board, the optimal configuration is to use multifrequency with parallel scan.

For further developments some more comparatives and improvements can be done. As an example, as CTSU2 includes shield electrode support, shield and safety function analysis could be done.

Some radiated and conducted emissions EMC tests could also be run. This kind of tests consist of measuring the electromagnetic field strength of the emissions that are unintentionally generated by the board in order to assure that it complies with the emissions limits.

As we have seen that the single frequency configuration measurement time is much lower than the one on multifrequency configuration, the debounce parameter could be incremented from 5 to 15 samples, resulting in an increment of the stability of the system. It can be the case that this would be preferred, so some improvements should be studied in order that this configuration passes EMC injected tests.

Finally, it is possible to configure non-touch signs detections (if the immunity to radiated emissions is confirmed previously), as CTSU2 is able to detect the movement of the finger to one or two dimensional directions with better results than CTSU, due to the sensibility improvements resulting from the better time results.

Bibliography

[1] Renesas Electronics Corporation, “Capacitive Sensor Microcontrollers CTSU Capacitive Touch Electrode Design Guide”. R30AN0389EJ0100 Rev.1.00, TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan, Apr.12.21. Available:

https://www.renesas.com/eu/en/document/apn/capacitive-sensor-microcontrollers-ctsu-capacitive-touch-electrode-design-guide?language=en?language=en

[2] RX113 Group, CTSU Basis of Cap touch detection. R30AN0218EJ0100 Rev.1.00 Dec 25, 2014. Available:

https://www.renesas.com/us/en/document/apn/ctsu-basis-cap-touch-detection-rev100?language=en

[3] RX130 Group. User’s Manual: Hardware. RENESAS 32-Bit MCU. RX Family ⁄ RX100 Series. R01UH0560EJ0300 Rev.3.00 Aug 09, 2018. Available:

https://www.renesas.com/us/en/document/mah/rx130-group-users-manual-hardware-rev300?language=en&r=1054356

[4] Renesas RA2L1 Group. User’s Manual: Hardware. 32-bit MCU. Renesas Advanced (RA) Family. Renesas RA2 Series. R01UH0853EJ0110 Rev.1.10 Feb 26, 2021. Available:

https://www.renesas.com/us/en/document/mah/ra2l1-group-users-manual-hardware?language=en&r=1398061

[5] Renesas RA2E1 Group. User’s Manual: Hardware. 32-bit MCU. Renesas Advanced (RA) Family. Renesas RA2 Series. R01UH0852EJ0110 Rev.1.10 Dec 28, 2020. Available:

https://www.renesas.com/us/en/document/man/ra2e1-group-users-manual-hardware

[6] RA Flexible Software Package Documentation Release v3.5.0. Available:

https://renesas.github.io/fsp/index.html

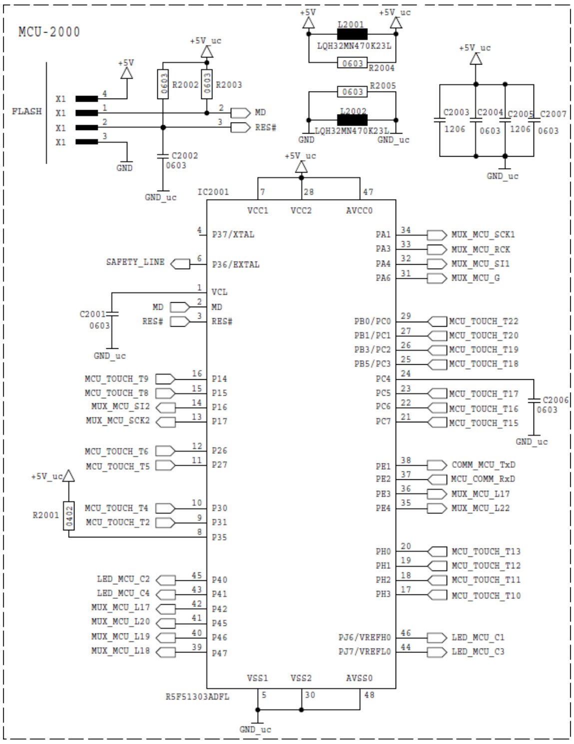
[7] *BASIC EMC PUBLICATION, Electromagnetic compatibility (EMC) – Part 4-4: Testing and measurement techniques – Electrical fast transient/burst immunity test*. IEC 61000-4-4, Edition 3 April 2012.

[8] *BASIC EMC PUBLICATION, Electromagnetic compatibility (EMC) – Part 4-6: Testing and measurement techniques – Immunity to conducted disturbances, induced by radio-frequency fields.* IEC 61000-4-6, Edition 4 October 2013.

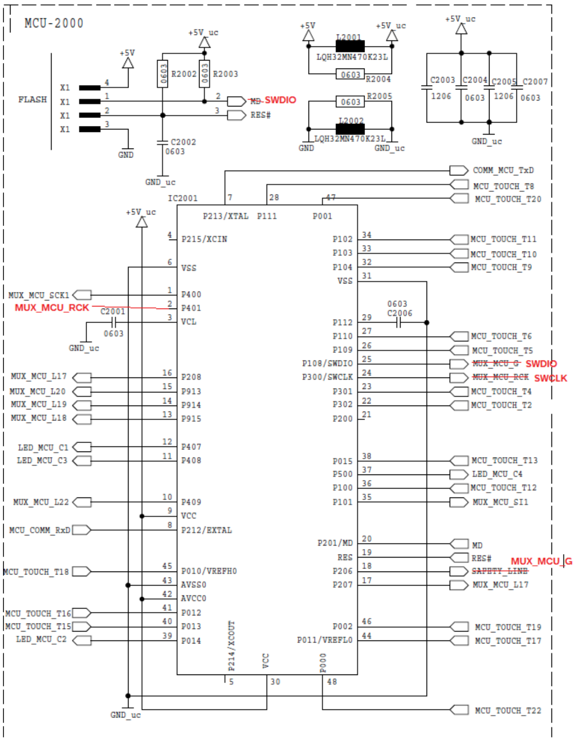
Appendices

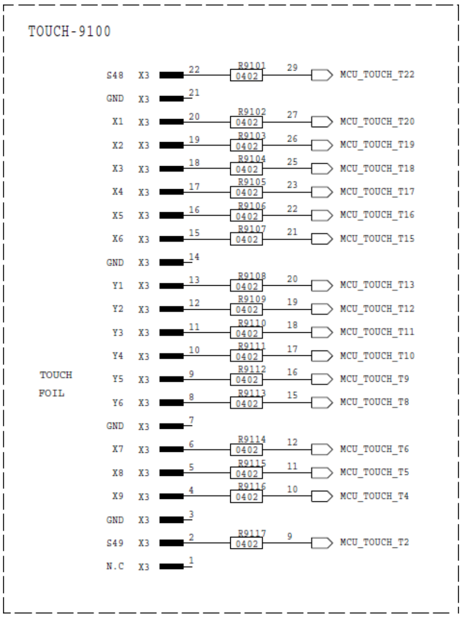
1. T4 using RX130 vs T4 using RA2E1 schematics

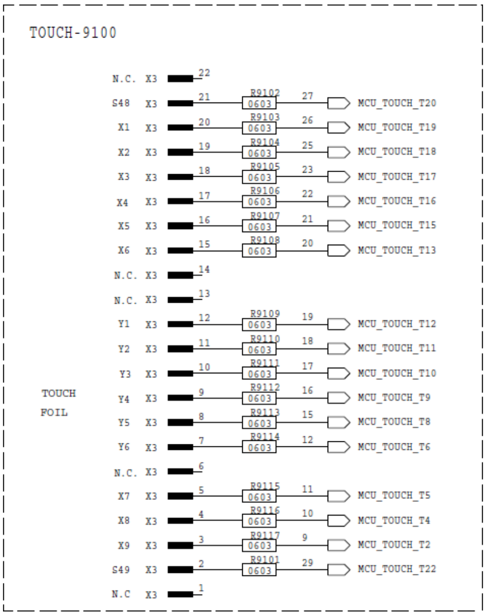
RX:



RA:

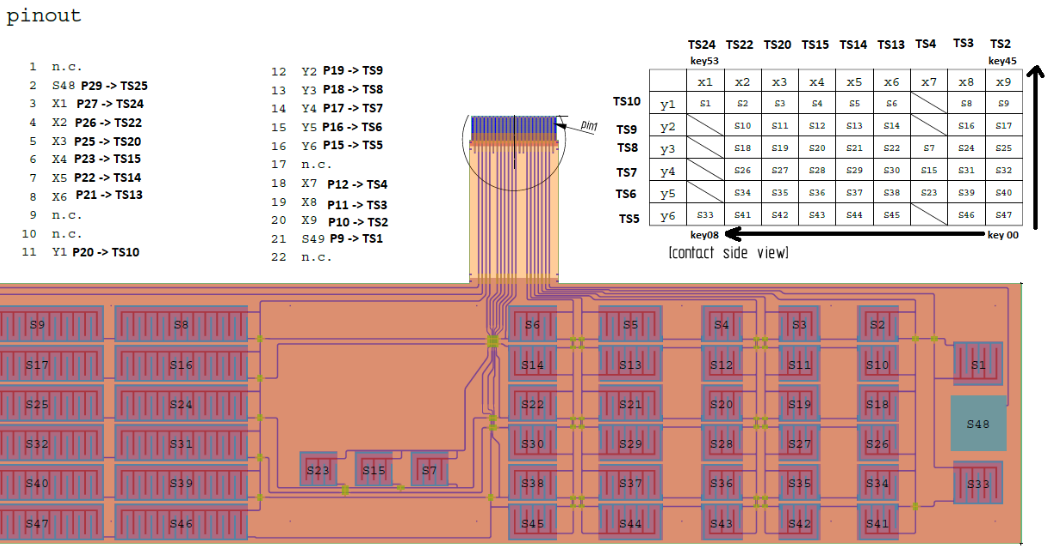






2. Resulting touch foils for RX and RA

Touch foil with RX130:



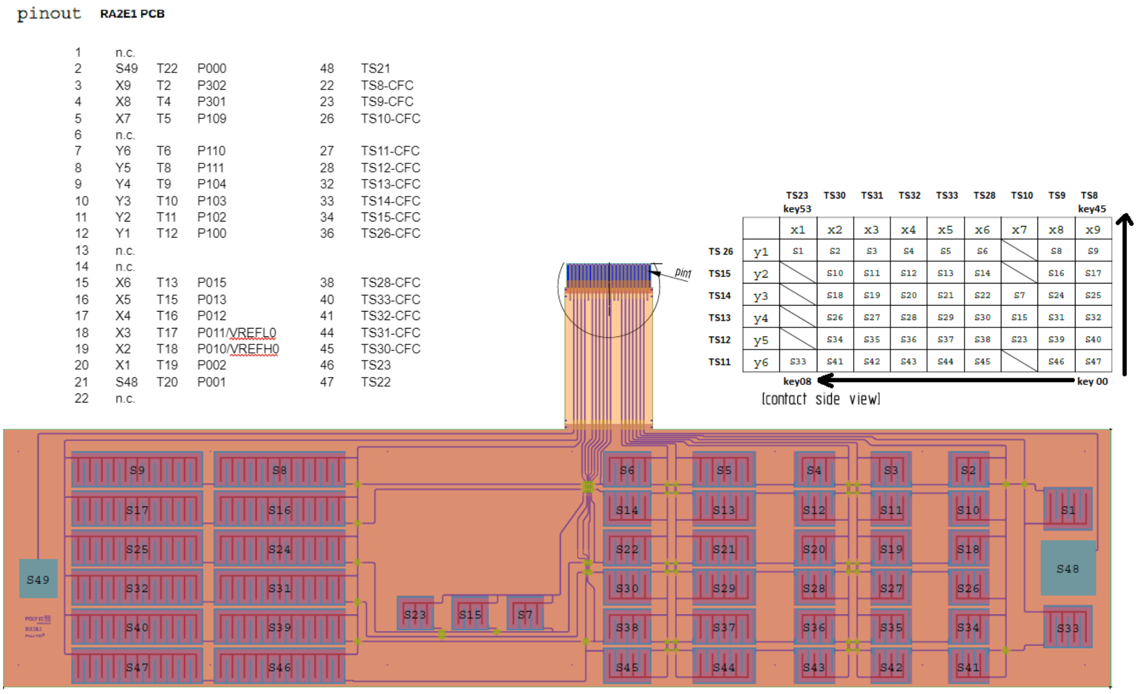
pinout

1	n.c.	
2	S48	**P29 -> TS25**
3	X1	**P27 -> TS24**
4	X2	**P26 -> TS22**
5	X3	**P25 -> TS20**
6	X4	**P23 -> TS15**
7	X5	**P22 -> TS14**
8	X6	**P21 -> TS13**
9	n.c.	
10	n.c.	
11	Y1	**P20 -> TS10**
12	Y2	**P19 -> TS9**
13	Y3	**P18 -> TS8**
14	Y4	**P17 -> TS7**
15	Y5	**P16 -> TS6**
16	Y6	**P15 -> TS5**
17	n.c.	
18	X7	**P12 -> TS4**
19	X8	**P11 -> TS3**
20	X9	**P10 -> TS2**
21	S49	**P9 -> TS1**
22	n.c.	

		TS24	TS22	TS20	TS15	TS14	TS13	TS4	TS3	TS2
		key53								key45
		x1	x2	x3	x4	x5	x6	x7	x8	x9
TS10	y1	S1	S2	S3	S4	S5	S6		S8	S9
TS9	y2		S10	S11	S12	S13	S14		S16	S17
TS8	y3		S18	S19	S20	S21	S22	S7	S24	S25
TS7	y4		S26	S27	S28	S29	S30	S15	S31	S32
TS6	y5		S34	S35	S36	S37	S38	S23	S39	S40
TS5	y6	S33	S41	S42	S43	S44	S45		S46	S47

key08 ← key 00
(contact side view)

Touch foil with RA2E1:



pinout **RA2E1 PCB**

1	n.c.				
2	S49	T22	P000	48	TS21
3	X9	T2	P302	22	TS8-CFC
4	X8	T4	P301	23	TS9-CFC
5	X7	T5	P109	26	TS10-CFC
6	n.c.				
7	Y6	T6	P110	27	TS11-CFC
8	Y5	T8	P111	28	TS12-CFC
9	Y4	T9	P104	32	TS13-CFC
10	Y3	T10	P103	33	TS14-CFC
11	Y2	T11	P102	34	TS15-CFC
12	Y1	T12	P100	36	TS26-CFC
13	n.c.				
14	n.c.				
15	X6	T13	P015	38	TS28-CFC
16	X5	T15	P013	40	TS33-CFC
17	X4	T16	P012	41	TS32-CFC
18	X3	T17	P011/VREFL0	44	TS31-CFC
19	X2	T18	P010/VREFH0	45	TS30-CFC
20	X1	T19	P002	46	TS23
21	S48	T20	P001	47	TS22
22	n.c.				

		TS23	TS30	TS31	TS32	TS33	TS28	TS10	TS9	TS8
		key53								key45
		x1	x2	x3	x4	x5	x6	x7	x8	x9
TS 26	y1	S1	S2	S3	S4	S5	S6		S8	S9
TS15	y2		S10	S11	S12	S13	S14		S16	S17
TS14	y3		S18	S19	S20	S21	S22	S7	S24	S25
TS13	y4		S26	S27	S28	S29	S30	S15	S31	S32
TS12	y5		S34	S35	S36	S37	S38	S23	S39	S40
TS11	y6	S33	S41	S42	S43	S44	S45		S46	S47

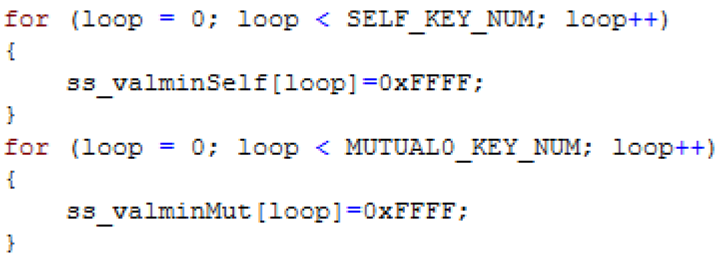
key08 ← key 00
(contact side view)

3. Code used for measurement extractions

For tracking code changes Git software has been used.

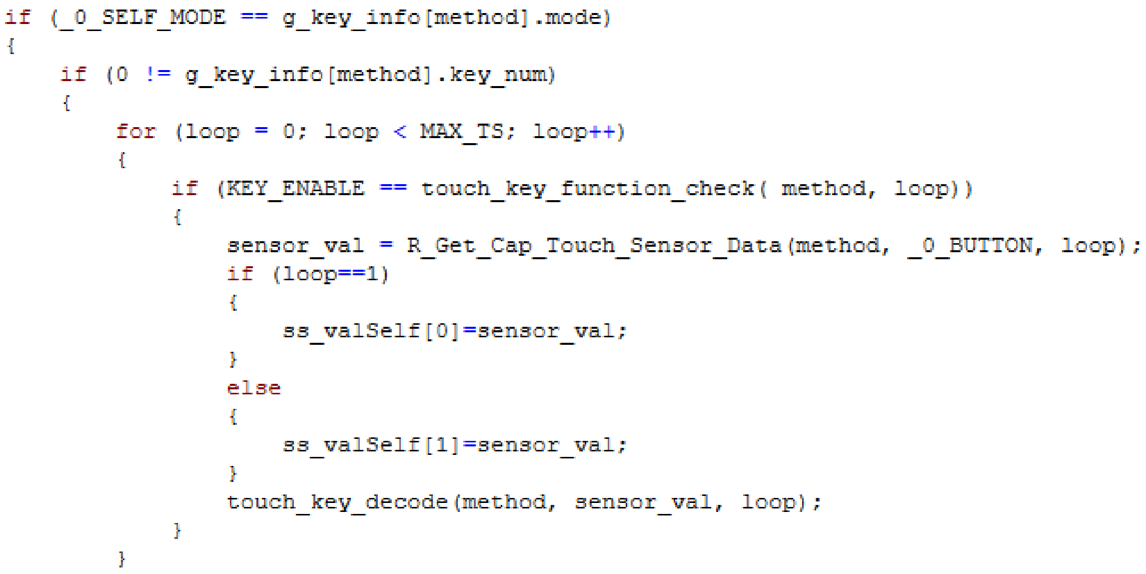
SNR measurement code for RX

Set minimum value to maximum for later comparison:

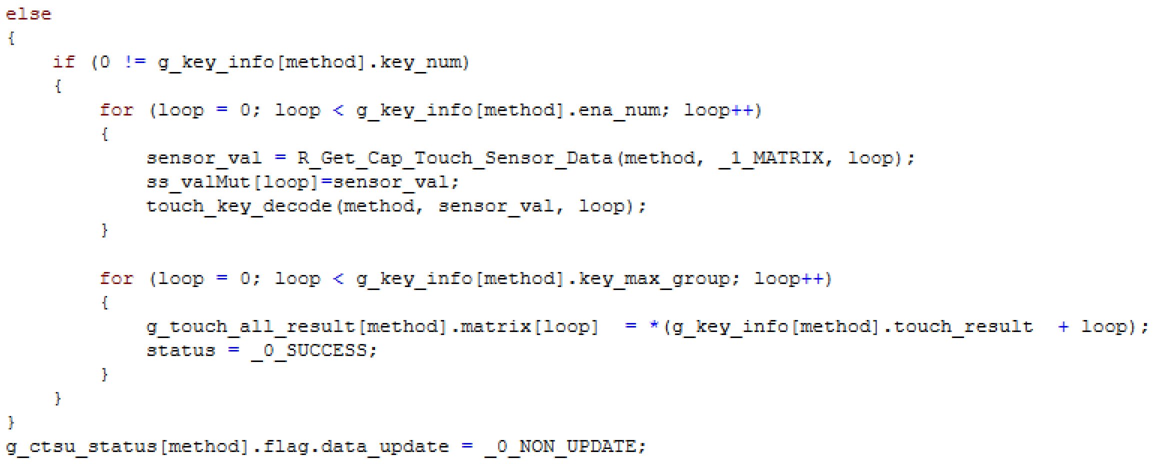


```
for (loop = 0; loop < SELF_KEY_NUM; loop++)
{
    ss_valminSelf[loop]=0xFFFF;
}
for (loop = 0; loop < MUTUAL0_KEY_NUM; loop++)
{
    ss_valminMut[loop]=0xFFFF;
}
```

Sensor value extraction:

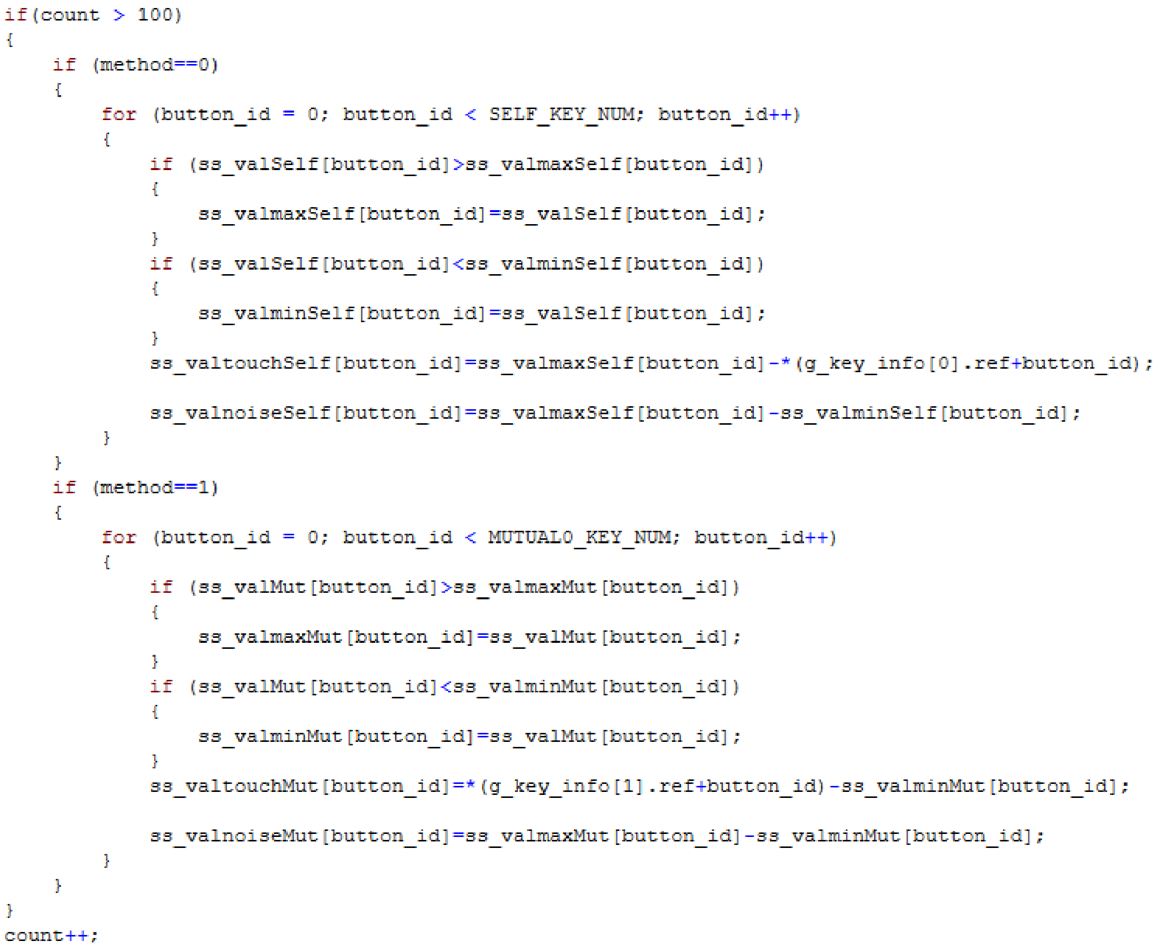


```
if (_0_SELF_MODE == g_key_info[method].mode)
{
    if (0 != g_key_info[method].key_num)
    {
        for (loop = 0; loop < MAX_TS; loop++)
        {
            if (KEY_ENABLE == touch_key_function_check( method, loop))
            {
                sensor_val = R_Get_Cap_Touch_Sensor_Data(method, _0_BUTTON, loop);
                if (loop==1)
                {
                    ss_valSelf[0]=sensor_val;
                }
                else
                {
                    ss_valSelf[1]=sensor_val;
                }
                touch_key_decode(method, sensor_val, loop);
            }
        }
```



```
else
{
    if (0 != g_key_info[method].key_num)
    {
        for (loop = 0; loop < g_key_info[method].ena_num; loop++)
        {
            sensor_val = R_Get_Cap_Touch_Sensor_Data(method, _1_MATRIX, loop);
            ss_valMut[loop]=sensor_val;
            touch_key_decode(method, sensor_val, loop);
        }

        for (loop = 0; loop < g_key_info[method].key_max_group; loop++)
        {
            g_touch_all_result[method].matrix[loop]  = *(g_key_info[method].touch_result  + loop);
            status = _0_SUCCESS;
        }
    }
}
g_ctsu_status[method].flag.data_update = _0_NON_UPDATE;
```

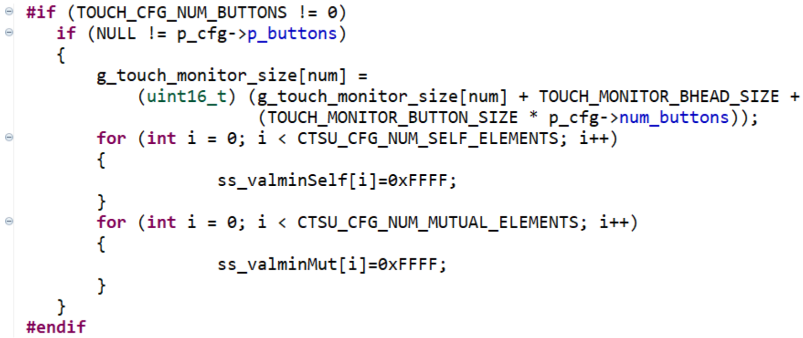
Noise and touch values measurement:



```
if(count > 100)
{
    if (method==0)
    {
        for (button_id = 0; button_id < SELF_KEY_NUM; button_id++)
        {
            if (ss_valSelf[button_id]>ss_valmaxSelf[button_id])
            {
                ss_valmaxSelf[button_id]=ss_valSelf[button_id];
            }
            if (ss_valSelf[button_id]<ss_valminSelf[button_id])
            {
                ss_valminSelf[button_id]=ss_valSelf[button_id];
            }
            ss_valtouchSelf[button_id]=ss_valmaxSelf[button_id]-*(g_key_info[0].ref+button_id);

            ss_valnoiseSelf[button_id]=ss_valmaxSelf[button_id]-ss_valminSelf[button_id];
        }
    }
    if (method==1)
    {
        for (button_id = 0; button_id < MUTUAL0_KEY_NUM; button_id++)
        {
            if (ss_valMut[button_id]>ss_valmaxMut[button_id])
            {
                ss_valmaxMut[button_id]=ss_valMut[button_id];
            }
            if (ss_valMut[button_id]<ss_valminMut[button_id])
            {
                ss_valminMut[button_id]=ss_valMut[button_id];
            }
            ss_valtouchMut[button_id]=*(g_key_info[1].ref+button_id)-ss_valminMut[button_id];

            ss_valnoiseMut[button_id]=ss_valmaxMut[button_id]-ss_valminMut[button_id];
        }
    }
}
count++;
```

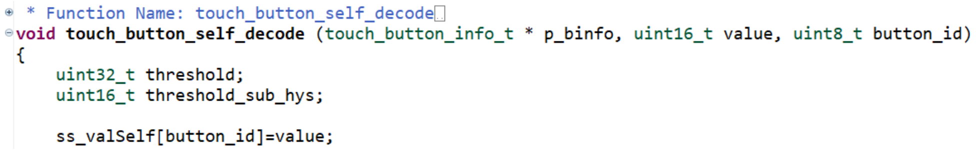
SNR measurement code for RA

Set minimum value to maximum for later comparison:

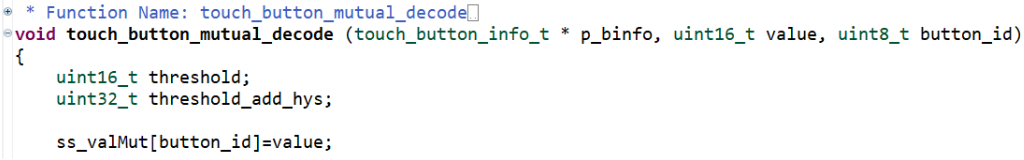


```
#if (TOUCH_CFG_NUM_BUTTONS != 0)
    if (NULL != p_cfg->p_buttons)
    {
        g_touch_monitor_size[num] =
            (uint16_t) (g_touch_monitor_size[num] + TOUCH_MONITOR_BHEAD_SIZE +
                        (TOUCH_MONITOR_BUTTON_SIZE * p_cfg->num_buttons));
        for (int i = 0; i < CTSU_CFG_NUM_SELF_ELEMENTS; i++)
        {
                    ss_valminSelf[i]=0xFFFF;
        }
        for (int i = 0; i < CTSU_CFG_NUM_MUTUAL_ELEMENTS; i++)
        {
                    ss_valminMut[i]=0xFFFF;
        }
    }
#endif
```

Sensor value extraction:

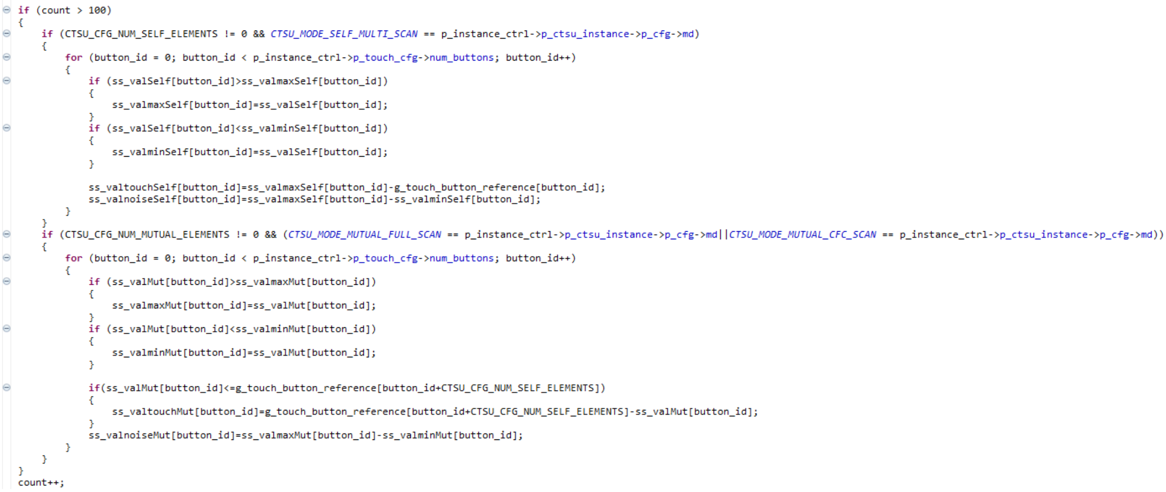


```
 * Function Name: touch_button_self_decode
void touch_button_self_decode (touch_button_info_t * p_binfo, uint16_t value, uint8_t button_id)
{
    uint32_t threshold;
    uint16_t threshold_sub_hys;

    ss_valSelf[button_id]=value;
```



```
 * Function Name: touch_button_mutual_decode
void touch_button_mutual_decode (touch_button_info_t * p_binfo, uint16_t value, uint8_t button_id)
{
    uint16_t threshold;
    uint32_t threshold_add_hys;

    ss_valMut[button_id]=value;
```

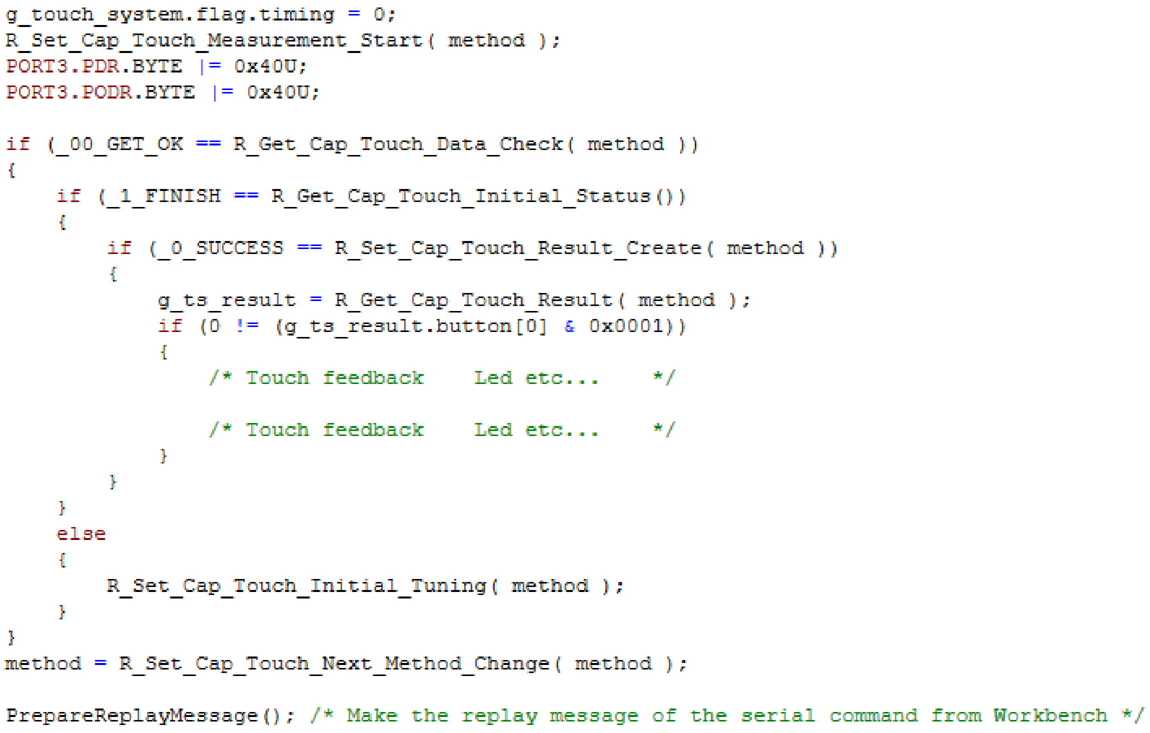
Noise and touch values measurement:



```
if (count > 100)
{
    if (CTSU_CFG_NUM_SELF_ELEMENTS != 0 && CTSU_MODE_SELF_MULTI_SCAN == p_instance_ctrl->p_ctsu_instance->p_cfg->md)
    {
        for (button_id = 0; button_id < p_instance_ctrl->p_touch_cfg->num_buttons; button_id++)
        {
            if (ss_valSelf[button_id]>ss_valmaxSelf[button_id])
            {
                ss_valmaxSelf[button_id]=ss_valSelf[button_id];
            }
            if (ss_valSelf[button_id]<ss_valminSelf[button_id])
            {
                ss_valminSelf[button_id]=ss_valSelf[button_id];
            }

            ss_valtouchSelf[button_id]=ss_valmaxSelf[button_id]-g_touch_button_reference[button_id];
            ss_valnoiseSelf[button_id]=ss_valmaxSelf[button_id]-ss_valminSelf[button_id];
        }
    }
    if (CTSU_CFG_NUM_MUTUAL_ELEMENTS != 0 && (CTSU_MODE_MUTUAL_FULL_SCAN == p_instance_ctrl->p_ctsu_instance->p_cfg->md||CTSU_MODE_MUTUAL_CFC_SCAN == p_instance_ctrl->p_ctsu_instance->p_cfg->md))
    {
        for (button_id = 0; button_id < p_instance_ctrl->p_touch_cfg->num_buttons; button_id++)
        {
            if (ss_valMut[button_id]>ss_valmaxMut[button_id])
            {
                ss_valmaxMut[button_id]=ss_valMut[button_id];
            }
            if (ss_valMut[button_id]<ss_valminMut[button_id])
            {
                ss_valminMut[button_id]=ss_valMut[button_id];
            }

            if(ss_valMut[button_id]<=g_touch_button_reference[button_id+CTSU_CFG_NUM_SELF_ELEMENTS])
            {
                ss_valtouchMut[button_id]=g_touch_button_reference[button_id+CTSU_CFG_NUM_SELF_ELEMENTS]-ss_valMut[button_id];
            }
            ss_valnoiseMut[button_id]=ss_valmaxMut[button_id]-ss_valminMut[button_id];
        }
    }
}
count++;
```

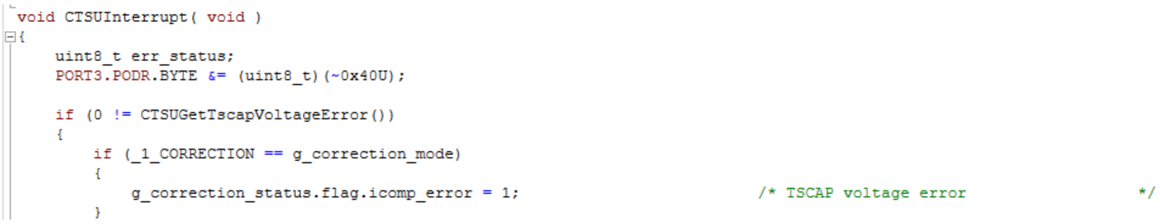
Toggle code for RX

Change before measurement:



```
g_touch_system.flag.timing = 0;
R_Set_Cap_Touch_Measurement_Start( method );
PORT3.PDR.BYTE |= 0x40U;
PORT3.PODR.BYTE |= 0x40U;

if (_00_GET_OK == R_Get_Cap_Touch_Data_Check( method ))
{
    if (_1_FINISH == R_Get_Cap_Touch_Initial_Status())
    {
        if (_0_SUCCESS == R_Set_Cap_Touch_Result_Create( method ))
        {
            g_ts_result = R_Get_Cap_Touch_Result( method );
            if (0 != (g_ts_result.button[0] & 0x0001))
            {
                /* Touch feedback    Led etc...    */

                /* Touch feedback    Led etc...    */
            }
        }
    }
    else
    {
        R_Set_Cap_Touch_Initial_Tuning( method );
    }
}
method = R_Set_Cap_Touch_Next_Method_Change( method );

PrepareReplayMessage(); /* Make the replay message of the serial command from Workbench */
```

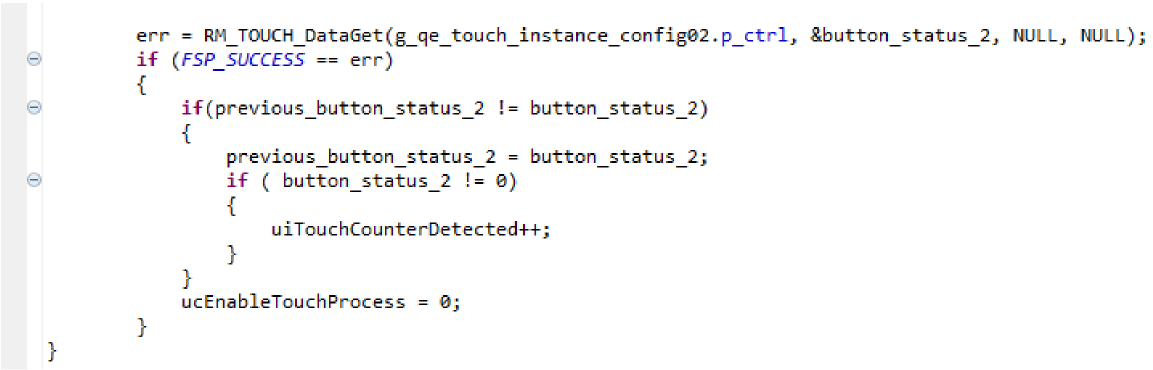
Change when interrupt jumps:



```
void CTSUInterrupt( void )
{
    uint8_t err_status;
    PORT3.PODR.BYTE &= (uint8_t)(~0x40U);

    if (0 != CTSUGetTscapVoltageError())
    {
        if (_1_CORRECTION == g_correction_mode)
        {
            g_correction_status.flag.icomp_error = 1;                        /* TSCAP voltage error                  */
        }
```

Toggle code for RA

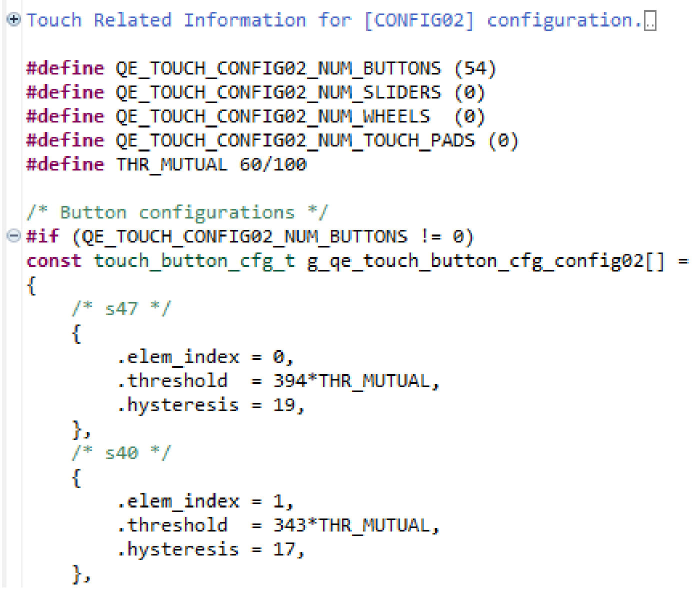
```
void qe_touch_main(void)
{
    fsp_err_t err;

        /* for [CONFIG01] configuration */
        pin_high(BSP_IO_PORT_04_PIN_07);
        err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
        if (FSP_SUCCESS != err)
        {
            while (true) {}
        }
        while (0 == g_qe_touch_flag) {}
        pin_low(BSP_IO_PORT_04_PIN_07);
        g_qe_touch_flag = 0;

        err = RM_TOUCH_DataGet(g_qe_touch_instance_config01.p_ctrl, &button_status, NULL, NULL);
        if (FSP_SUCCESS == err)
        {
            if(previous_button_status != button_status)
            {
                previous_button_status = button_status;
                if ( button_status != 0)
                {
                    uiTouchCounterDetected++;
                }

            }
        }

        /* for [CONFIG02] configuration */
        pin_high(BSP_IO_PORT_04_PIN_07);
        err = RM_TOUCH_ScanStart(g_qe_touch_instance_config02.p_ctrl);
        if (FSP_SUCCESS != err)
        {
            while (true) {}
        }
        while (0 == g_qe_touch_flag) {}
        pin_low(BSP_IO_PORT_04_PIN_07);
        g_qe_touch_flag = 0;
```



```
        err = RM_TOUCH_DataGet(g_qe_touch_instance_config02.p_ctrl, &button_status_2, NULL, NULL);
        if (FSP_SUCCESS == err)
        {
            if(previous_button_status_2 != button_status_2)
            {
                previous_button_status_2 = button_status_2;
                if ( button_status_2 != 0)
                {
                    uiTouchCounterDetected++;
                }
            }
            ucEnableTouchProcess = 0;
        }
}
```

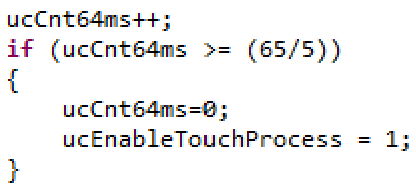
Code changes for RA EMC tests

Threshold change to be adjusted to 60%.



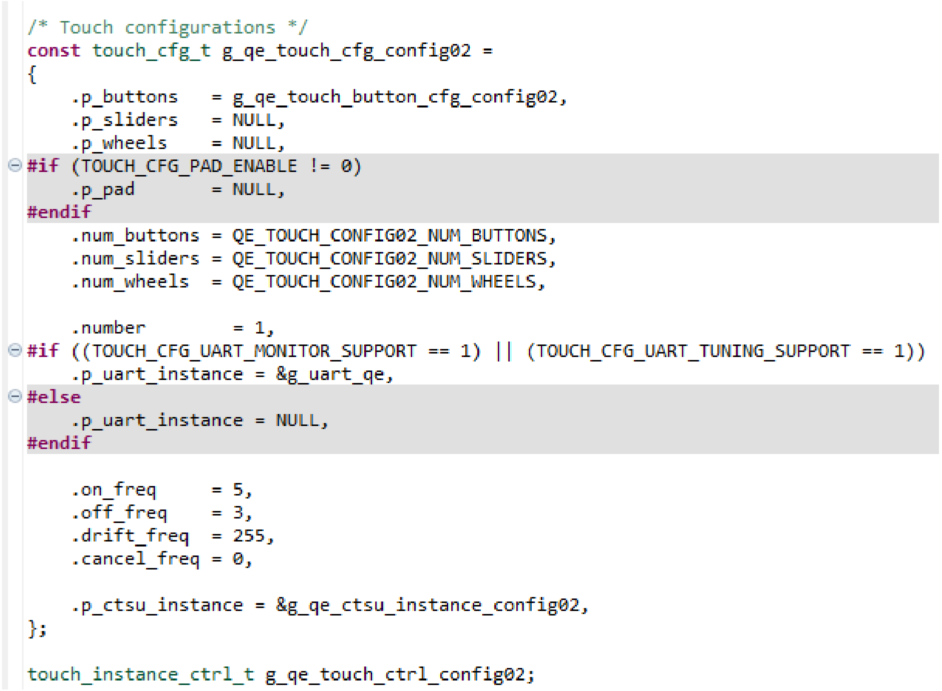
```
Touch Related Information for [CONFIG02] configuration.□

#define QE_TOUCH_CONFIG02_NUM_BUTTONS (54)
#define QE_TOUCH_CONFIG02_NUM_SLIDERS (0)
#define QE_TOUCH_CONFIG02_NUM_WHEELS  (0)
#define QE_TOUCH_CONFIG02_NUM_TOUCH_PADS (0)
#define THR_MUTUAL 60/100

/* Button configurations */
#if (QE_TOUCH_CONFIG02_NUM_BUTTONS != 0)
const touch_button_cfg_t g_qe_touch_button_cfg_config02[] =
{
    /* s47 */
    {
        .elem_index = 0,
        .threshold  = 394*THR_MUTUAL,
        .hysteresis = 19,
    },
    /* s40 */
    {
        .elem_index = 1,
        .threshold  = 343*THR_MUTUAL,
        .hysteresis = 17,
    },
```

For debounce changes, ucCnt stands for enabling touch process time



```
ucCnt64ms++;
if (ucCnt64ms >= (65/5))
{
     ucCnt64ms=0;
     ucEnableTouchProcess = 1;
}
```

on_freq is the debounce parameter.



```
/* Touch configurations */
const touch_cfg_t g_qe_touch_cfg_config02 =
{
    .p_buttons    = g_qe_touch_button_cfg_config02,
    .p_sliders    = NULL,
    .p_wheels     = NULL,
#if (TOUCH_CFG_PAD_ENABLE != 0)
    .p_pad        = NULL,
#endif
    .num_buttons = QE_TOUCH_CONFIG02_NUM_BUTTONS,
    .num_sliders = QE_TOUCH_CONFIG02_NUM_SLIDERS,
    .num_wheels  = QE_TOUCH_CONFIG02_NUM_WHEELS,

    .number        = 1,
#if ((TOUCH_CFG_UART_MONITOR_SUPPORT == 1) || (TOUCH_CFG_UART_TUNING_SUPPORT == 1))
    .p_uart_instance = &g_uart_qe,
#else
    .p_uart_instance = NULL,
#endif

    .on_freq      = 5,
    .off_freq     = 3,
    .drift_freq   = 255,
    .cancel_freq  = 0,

    .p_ctsu_instance = &g_qe_ctsu_instance_config02,
};

touch_instance_ctrl_t g_qe_touch_ctrl_config02;
```

4. Tests executions

Jira Software is the leading agile project management tool used by the company. Test results have been reported and exported from Jira:

SNR, memory and time test execution results:

Test Cycle

Key	SP00014-C1	**Name**	RX tests		
Description	RX tests with and without leds to find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RX				
Planned start date	2021-11-01, 00:00	**Planned end date**	2022-01-15, 00:00	**Iteration**	-
Status	DONE	**Version**	-		

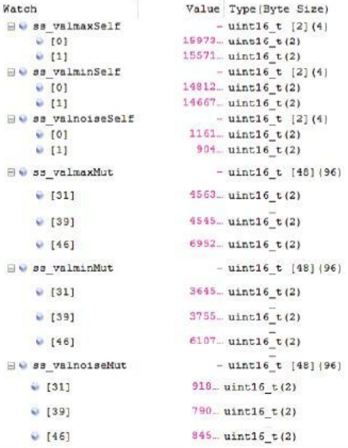
Test Executions

Key	SP00014-T13	**Status**	PASS	**Name**	RX tests in former CTSU configuration with leds (all channels)
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RX				
Precondition	Config: all channels				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2022-01-13, 08:57	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	CS+	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

Test Script

1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values

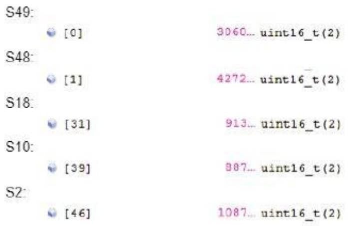
Actual Result



```
Watch                                Value Type(Byte Size)
⊟ ss_valmaxSelf                         - uint16_t [2](4)
      [0]                          15973… uint16_t(2)
      [1]                          15571… uint16_t(2)
⊟ ss_valminSelf                         - uint16_t [2](4)
      [0]                          14812… uint16_t(2)
      [1]                          14667… uint16_t(2)
⊟ ss_valnoiseSelf                       - uint16_t [2](4)
      [0]                           1161… uint16_t(2)
      [1]                            904… uint16_t(2)
⊟ ss_valmaxMut                          - uint16_t [48](96)
      [31]                          4563… uint16_t(2)
      [39]                          4545… uint16_t(2)
      [46]                          6952… uint16_t(2)
⊟ ss_valminMut                          - uint16_t [48](96)
      [31]                          3645… uint16_t(2)
      [39]                          3755… uint16_t(2)
      [46]                          6107… uint16_t(2)
⊟ ss_valnoiseMut                        - uint16_t [48](96)
     [31]                            918… uint16_t(2)
     [39]                            790… uint16_t(2)
     [46]                            845… uint16_t(2)
```

	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal

Actual Result



```
S49:
      [0]                 3060… uint16_t(2)
S48:
      [1]                 4272… uint16_t(2)
S18:
      [31]                 913… uint16_t(2)
S10:
      [39]                 887… uint16_t(2)
S2:
      [46]                1087… uint16_t(2)
```

	Issues	-

3	**Status**	PASS
	Details	Calculate SNR (touch/noise value)
	Test Data	-
	Expected Result	SNR

Actual Result

RX all channels	Max NT	Min NT	Noise	Max T	SNR
Self but					
S49	15973	14812	1161	3060	2,64
S48	15571	14667	904	4272	4,73
Mutual but					
S18	4563	3645	918	913	0,99
S10	4545	3755	790	887	1,12
S2	6952	6107	845	1087	1,29

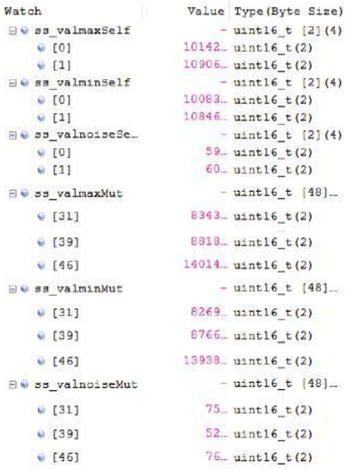
	Issues	-

Key	SP00014-T3	**Status**	PASS	**Name**	RX tests in former CTSU configuration without leds (all channels)
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RX				
Precondition	Config: all channels				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2022-01-13, 08:56	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	CS+	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

Test Script

1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values

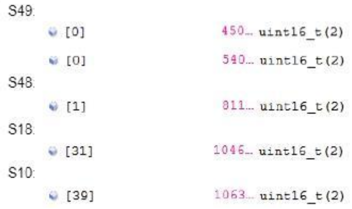
Actual Result



```
Watch                      Value  Type(Byte Size)
⊟ ss_valmaxSelf                -  uint16_t [2](4)
     [0]                  10142…  uint16_t(2)
     [1]                  10906…  uint16_t(2)
⊟ ss_valminSelf                -  uint16_t [2](4)
     [0]                  10083…  uint16_t(2)
     [1]                  10846…  uint16_t(2)
⊟ ss_valnoiseSe…               -  uint16_t [2](4)
     [0]                     59…  uint16_t(2)
     [1]                     60…  uint16_t(2)
⊟ ss_valmaxMut                 -  uint16_t [48]…
     [31]                  8343…  uint16_t(2)
     [39]                  8818…  uint16_t(2)
     [46]                 14014…  uint16_t(2)
⊟ ss_valminMut                 -  uint16_t [48]…
     [31]                  8269…  uint16_t(2)
     [39]                  8766…  uint16_t(2)
     [46]                 13938…  uint16_t(2)
⊟ ss_valnoiseMut               -  uint16_t [48]…
     [31]                    75…  uint16_t(2)
     [39]                    52…  uint16_t(2)
     [46]                    76…  uint16_t(2)
```

	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal

Actual Result



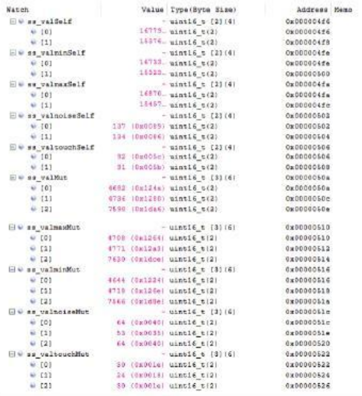
```
S49:
     [0]                 450…  uint16_t(2)
     [0]                 540…  uint16_t(2)
S48:
     [1]                 811…  uint16_t(2)
S18:
     [31]               1046…  uint16_t(2)
S10:
     [39]               1063…  uint16_t(2)
```

		 S2 [46] 1285… uint16_t(2)
	Issues	-
	Status	PASS
	Details	Calculate SNR (touch/noise value)
	Test Data	-
	Expected Result	SNR
3	**Actual Result**	

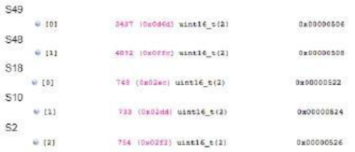
RX all channels	Max NT	Min NT	Noise	Max T	SNR
Self but					
S49	10142	10083	59	540	9,15
S48	10906	10846	60	811	13,52
Mutual but					
S18	8343	8269	75	1046	13,95
S10	8818	8766	52	1063	20,44
S2	14014	13938	76	1285	16,91

	Issues	-

Key	SP00014-T1	**Status**	PASS	**Name**	RX tests in former CTSU configuration without leds
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RX				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2022-01-13, 08:53	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	CS+	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

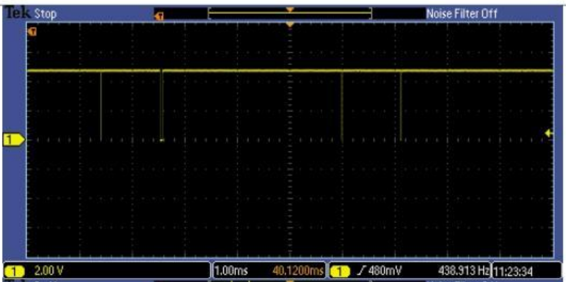
	Test Script	
	Status	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
1	**Actual Result**	

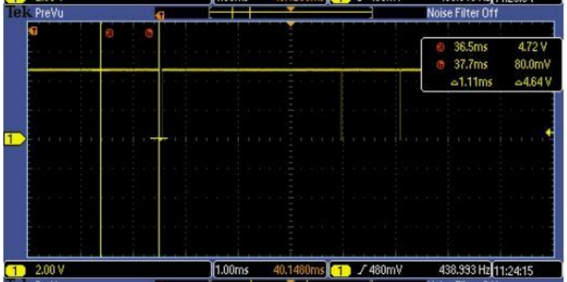
```
Watch                          Value  Type(Byte Size)      Address  Memo
⊟ ss_valSelf                       - uint16_t [2](4)    0x000004f6
    [0]                       16775… uint16_t(2)        0x000004f6
    [1]                       15376… uint16_t(2)        0x000004f8
⊟ ss_valminSelf                    - uint16_t [2](4)    0x000004fe
    [0]                       16733… uint16_t(2)        0x000004fe
    [1]                       15323… uint16_t(2)        0x00000500
⊟ ss_valmaxSelf                    - uint16_t [2](4)    0x000004fa
    [0]                       16870… uint16_t(2)        0x000004fa
    [1]                       15457… uint16_t(2)        0x000004fc
⊟ ss_valnoiseSelf                  - uint16_t [2](4)    0x00000502
    [0]                 137 (0x0089) uint16_t(2)        0x00000502
    [1]                 134 (0x0086) uint16_t(2)        0x00000504
⊟ ss_valtouchSelf                  - uint16_t [2](4)    0x00000506
    [0]                  92 (0x005c) uint16_t(2)        0x00000506
    [1]                  91 (0x005b) uint16_t(2)        0x00000508
⊟ ss_valMut                        - uint16_t [3](6)    0x0000050a
    [0]                4682 (0x124a) uint16_t(2)        0x0000050a
    [1]                4736 (0x1280) uint16_t(2)        0x0000050c
    [2]                7590 (0x1da6) uint16_t(2)        0x0000050e

⊟ ss_valmaxMut                     - uint16_t [3](6)    0x00000510
    [0]                4708 (0x1264) uint16_t(2)        0x00000510
    [1]                4771 (0x12a3) uint16_t(2)        0x00000512
    [2]                7630 (0x1dce) uint16_t(2)        0x00000514
⊟ ss_valminMut                     - uint16_t [3](6)    0x00000516
    [0]                4644 (0x1224) uint16_t(2)        0x00000516
    [1]                4718 (0x126e) uint16_t(2)        0x00000518
    [2]                7566 (0x1d8e) uint16_t(2)        0x0000051a
⊟ ss_valnoiseMut                   - uint16_t [3](6)    0x0000051c
    [0]                  64 (0x0040) uint16_t(2)        0x0000051c
    [1]                  53 (0x0035) uint16_t(2)        0x0000051e
    [2]                  64 (0x0040) uint16_t(2)        0x00000520
⊟ ss_valtouchMut                   - uint16_t [3](6)    0x00000522
    [0]                  30 (0x001e) uint16_t(2)        0x00000522
    [1]                  24 (0x0018) uint16_t(2)        0x00000524
    [2]                  30 (0x001e) uint16_t(2)        0x00000526
```

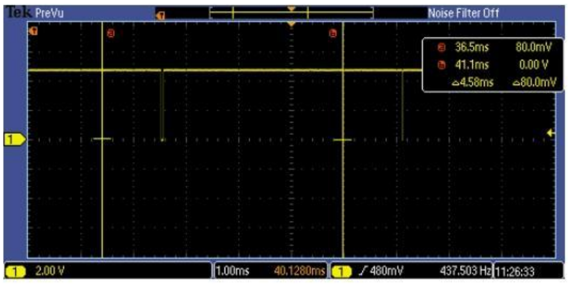
	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
	Actual Result	

```
S49
    [0]        3437 (0x0d6d) uint16_t(2)        0x00000506
S48
    [1]        4092 (0x0ffc) uint16_t(2)        0x00000508
S18
    [0]         748 (0x02ec) uint16_t(2)        0x00000522
S10
    [1]         733 (0x02dd) uint16_t(2)        0x00000524
S2
    [2]         754 (0x02f2) uint16_t(2)        0x00000526
```

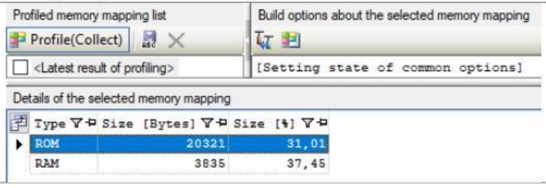
	Issues	-
	Status	PASS
	Details	Calculate SNR (touch/noise value)

<table>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td>3</td><td>Actual Result</td><td>
<table>
<tr><td>RX</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>16870</td><td>16733</td><td>137</td><td>3437</td><td>25,09</td></tr>
<tr><td>S48</td><td>15457</td><td>15323</td><td>134</td><td>4092</td><td>30,54</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>4708</td><td>4644</td><td>64</td><td>748</td><td>11,69</td></tr>
<tr><td>S10</td><td>4771</td><td>4718</td><td>53</td><td>549</td><td>10,36</td></tr>
<tr><td>S2</td><td>7630</td><td>7566</td><td>64</td><td>754</td><td>11,78</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>4</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check measurement time</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>Measurement time</td></tr>
<tr><td></td><td>Actual Result</td><td>






<table>
<tr><td>Measurement time (ms)</td><td></td></tr>
<tr><td>Self</td><td>1,11</td></tr>
<tr><td>Mutual</td><td>3,41</td></tr>
<tr><td>Total</td><td>4,52</td></tr>
<tr><td>Global</td><td>4,58</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td></td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check memory usage</td></tr>
</table>

5	**Test Data**	-
	Expected Result	Memory usage
	Actual Result	
	Issues	-

Profiled memory mapping list

☐ <Latest result of profiling>

Build options about the selected memory mapping

[Setting state of common options]

Details of the selected memory mapping

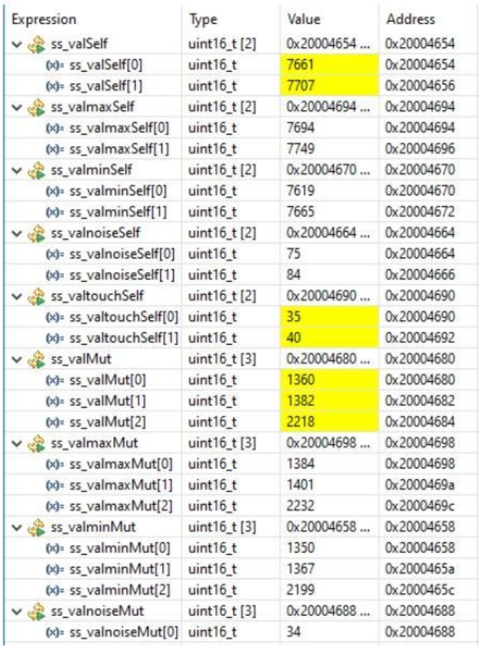
Type	Size [Bytes]	Size [%]
ROM	20321	31,01
RAM	3835	37,45

Test Cycle

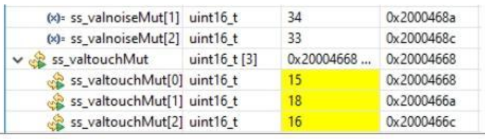
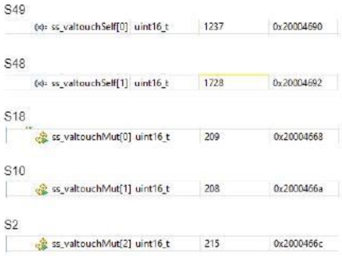
Key	SP00014-C2	**Name**	RA tests		
Description	Ra tests with and without leds (using different configurations) to find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Planned start date	2021-11-01, 00:00	**Planned end date**	2022-01-15, 00:00	**Iteration**	-
Status	DONE	**Version**	-		

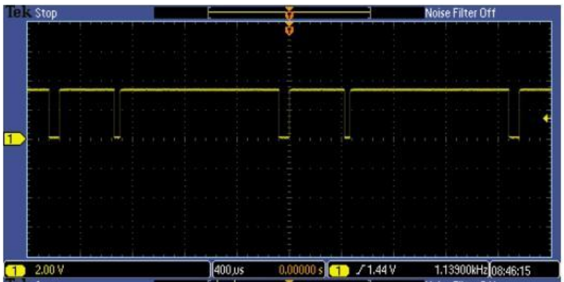
Test Executions

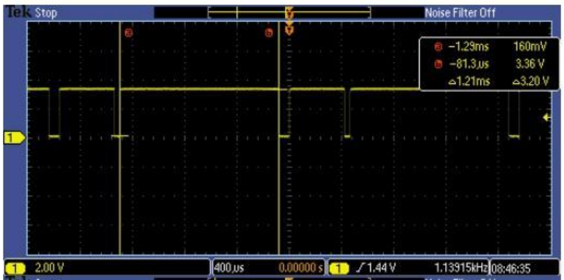
Key	SP00014-T5	**Status**	PASS	**Name**	RA tests in former CTSU configuration (0.128ms for self scan time, 0.256ms for mutual scan time, without leds)
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:43	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

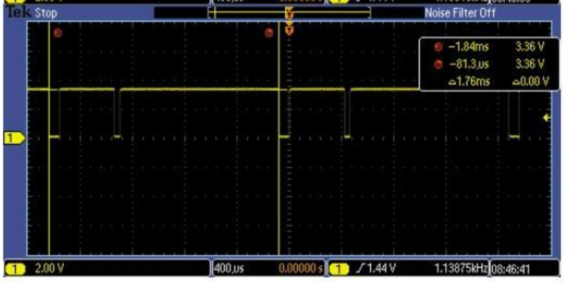
	Test Script	
1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
	Actual Result	

Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x20004654 ...	0x20004654
ss_valSelf[0]	uint16_t	7661	0x20004654
ss_valSelf[1]	uint16_t	7707	0x20004656
ss_valmaxSelf	uint16_t [2]	0x20004694 ...	0x20004694
ss_valmaxSelf[0]	uint16_t	7694	0x20004694
ss_valmaxSelf[1]	uint16_t	7749	0x20004696
ss_valminSelf	uint16_t [2]	0x20004670 ...	0x20004670
ss_valminSelf[0]	uint16_t	7619	0x20004670
ss_valminSelf[1]	uint16_t	7665	0x20004672
ss_valnoiseSelf	uint16_t [2]	0x20004664 ...	0x20004664
ss_valnoiseSelf[0]	uint16_t	75	0x20004664
ss_valnoiseSelf[1]	uint16_t	84	0x20004666
ss_valtouchSelf	uint16_t [2]	0x20004690 ...	0x20004690
ss_valtouchSelf[0]	uint16_t	35	0x20004690
ss_valtouchSelf[1]	uint16_t	40	0x20004692
ss_valMut	uint16_t [3]	0x20004680 ...	0x20004680
ss_valMut[0]	uint16_t	1360	0x20004680
ss_valMut[1]	uint16_t	1382	0x20004682
ss_valMut[2]	uint16_t	2218	0x20004684
ss_valmaxMut	uint16_t [3]	0x20004698 ...	0x20004698
ss_valmaxMut[0]	uint16_t	1384	0x20004698
ss_valmaxMut[1]	uint16_t	1401	0x2000469a
ss_valmaxMut[2]	uint16_t	2232	0x2000469c
ss_valminMut	uint16_t [3]	0x20004658 ...	0x20004658
ss_valminMut[0]	uint16_t	1350	0x20004658
ss_valminMut[1]	uint16_t	1367	0x2000465a
ss_valminMut[2]	uint16_t	2199	0x2000465c
ss_valnoiseMut	uint16_t [3]	0x20004688 ...	0x20004688
ss_valnoiseMut[0]	uint16_t	34	0x20004688

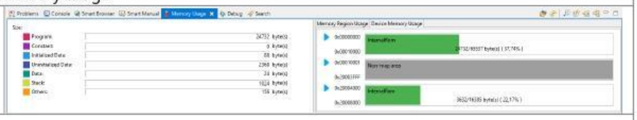
<table>
<tr><td></td><td></td><td>
(x)= ss_valnoiseMut[1] uint16_t 34 0x2000468a
(x)= ss_valnoiseMut[2] uint16_t 33 0x2000468c
ss_valtouchMut uint16_t [3] 0x20004668 ... 0x20004668
ss_valtouchMut[0] uint16_t 15 0x20004668
ss_valtouchMut[1] uint16_t 18 0x2000466a
ss_valtouchMut[2] uint16_t 16 0x2000466c</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>2</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check touch value (max-reference difference while touching)</td></tr>
<tr><td></td><td>Test Data</td><td>ss_valtouchSelf, ss_valtouchMut</td></tr>
<tr><td></td><td>Expected Result</td><td>Maximum measured signal</td></tr>
<tr><td></td><td>Actual Result</td><td>
S49
(x)= ss_valtouchSelf[0] uint16_t 1237 0x20004690
S48
(x)= ss_valtouchSelf[1] uint16_t 1728 0x20004692
S18
ss_valtouchMut[0] uint16_t 209 0x20004668
S10
ss_valtouchMut[1] uint16_t 208 0x2000466a
S2
ss_valtouchMut[2] uint16_t 215 0x2000466c</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>3</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td></td><td>Actual Result</td><td>
<table>
<tr><td>RA 1 freq</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>7694</td><td>7619</td><td>75</td><td>1237</td><td>16,49</td></tr>
<tr><td>S48</td><td>7749</td><td>7665</td><td>84</td><td>1728</td><td>20,57</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>1384</td><td>1350</td><td>34</td><td>209</td><td>6,15</td></tr>
<tr><td>S10</td><td>1401</td><td>1367</td><td>34</td><td>208</td><td>6,12</td></tr>
<tr><td>S2</td><td>2232</td><td>2199</td><td>33</td><td>215</td><td>6,52</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>4</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check measurement time</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>Measurement time</td></tr>
<tr><td></td><td>Actual Result</td><td></td></tr>
</table>







Measurement time (ms)	
Self	0,42
Mutual	1,21
Total	1,63
Global	1,76

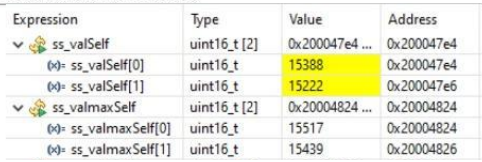
	Issues	-
5	**Status**	PASS
	Details	Check memory usage
	Test Data	-
	Expected Result	Memory usage
	Actual Result	
	Issues	-

Size:

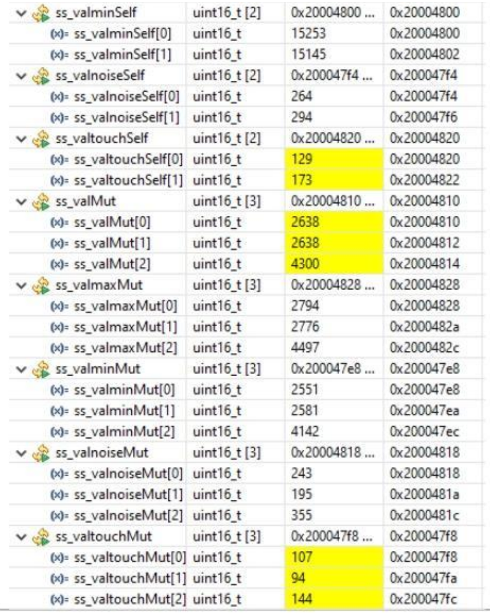
Program	24732 byte(s)
Constant	0 byte(s)
Initialized Data	88 byte(s)
Uninitialized Data	2368 byte(s)
Data	24 byte(s)
Stack	1024 byte(s)
Others	156 byte(s)

Key	SP00014-T10	**Status**	PASS	**Name**	RA tests configuring multi-frequency scan with leds
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-11				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:43	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

Test Script

	Status	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
		

Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x200047e4 ...	0x200047e4
(x)= ss_valSelf[0]	uint16_t	15388	0x200047e4
(x)= ss_valSelf[1]	uint16_t	15222	0x200047e6
ss_valmaxSelf	uint16_t [2]	0x20004824 ...	0x20004824
(x)= ss_valmaxSelf[0]	uint16_t	15517	0x20004824
(x)= ss_valmaxSelf[1]	uint16_t	15439	0x20004826

<table>
<tr><td>1</td><td>Actual Result</td><td>
<table>
<tr><td>ss_valminSelf</td><td>uint16_t [2]</td><td>0x20004800 ...</td><td>0x20004800</td></tr>
<tr><td>(x)= ss_valminSelf[0]</td><td>uint16_t</td><td>15253</td><td>0x20004800</td></tr>
<tr><td>(x)= ss_valminSelf[1]</td><td>uint16_t</td><td>15145</td><td>0x20004802</td></tr>
<tr><td>ss_valnoiseSelf</td><td>uint16_t [2]</td><td>0x200047f4 ...</td><td>0x200047f4</td></tr>
<tr><td>(x)= ss_valnoiseSelf[0]</td><td>uint16_t</td><td>264</td><td>0x200047f4</td></tr>
<tr><td>(x)= ss_valnoiseSelf[1]</td><td>uint16_t</td><td>294</td><td>0x200047f6</td></tr>
<tr><td>ss_valtouchSelf</td><td>uint16_t [2]</td><td>0x20004820 ...</td><td>0x20004820</td></tr>
<tr><td>(x)= ss_valtouchSelf[0]</td><td>uint16_t</td><td>129</td><td>0x20004820</td></tr>
<tr><td>(x)= ss_valtouchSelf[1]</td><td>uint16_t</td><td>173</td><td>0x20004822</td></tr>
<tr><td>ss_valMut</td><td>uint16_t [3]</td><td>0x20004810 ...</td><td>0x20004810</td></tr>
<tr><td>(x)= ss_valMut[0]</td><td>uint16_t</td><td>2638</td><td>0x20004810</td></tr>
<tr><td>(x)= ss_valMut[1]</td><td>uint16_t</td><td>2638</td><td>0x20004812</td></tr>
<tr><td>(x)= ss_valMut[2]</td><td>uint16_t</td><td>4300</td><td>0x20004814</td></tr>
<tr><td>ss_valmaxMut</td><td>uint16_t [3]</td><td>0x20004828 ...</td><td>0x20004828</td></tr>
<tr><td>(x)= ss_valmaxMut[0]</td><td>uint16_t</td><td>2794</td><td>0x20004828</td></tr>
<tr><td>(x)= ss_valmaxMut[1]</td><td>uint16_t</td><td>2776</td><td>0x2000482a</td></tr>
<tr><td>(x)= ss_valmaxMut[2]</td><td>uint16_t</td><td>4497</td><td>0x2000482c</td></tr>
<tr><td>ss_valminMut</td><td>uint16_t [3]</td><td>0x200047e8 ...</td><td>0x200047e8</td></tr>
<tr><td>(x)= ss_valminMut[0]</td><td>uint16_t</td><td>2551</td><td>0x200047e8</td></tr>
<tr><td>(x)= ss_valminMut[1]</td><td>uint16_t</td><td>2581</td><td>0x200047ea</td></tr>
<tr><td>(x)= ss_valminMut[2]</td><td>uint16_t</td><td>4142</td><td>0x200047ec</td></tr>
<tr><td>ss_valnoiseMut</td><td>uint16_t [3]</td><td>0x20004818 ...</td><td>0x20004818</td></tr>
<tr><td>(x)= ss_valnoiseMut[0]</td><td>uint16_t</td><td>243</td><td>0x20004818</td></tr>
<tr><td>(x)= ss_valnoiseMut[1]</td><td>uint16_t</td><td>195</td><td>0x2000481a</td></tr>
<tr><td>(x)= ss_valnoiseMut[2]</td><td>uint16_t</td><td>355</td><td>0x2000481c</td></tr>
<tr><td>ss_valtouchMut</td><td>uint16_t [3]</td><td>0x200047f8 ...</td><td>0x200047f8</td></tr>
<tr><td>(x)= ss_valtouchMut[0]</td><td>uint16_t</td><td>107</td><td>0x200047f8</td></tr>
<tr><td>(x)= ss_valtouchMut[1]</td><td>uint16_t</td><td>94</td><td>0x200047fa</td></tr>
<tr><td>(x)= ss_valtouchMut[2]</td><td>uint16_t</td><td>144</td><td>0x200047fc</td></tr>
</table></td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td></td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check touch value (max-reference difference while touching)</td></tr>
<tr><td></td><td>Test Data</td><td>ss_valtouchSelf, ss_valtouchMut</td></tr>
<tr><td></td><td>Expected Result</td><td>Maximum measured signal</td></tr>
<tr><td>2</td><td>Actual Result</td><td>S49

(x)= ss_valtouchSelf[0] uint16_t 2342 0x20004820
S48:

(x)= ss_valtouchSelf[1] uint16_t 3606 0x20004822
S18

(x)= ss_valtouchMut[0] uint16_t 513 0x200047f8
S10

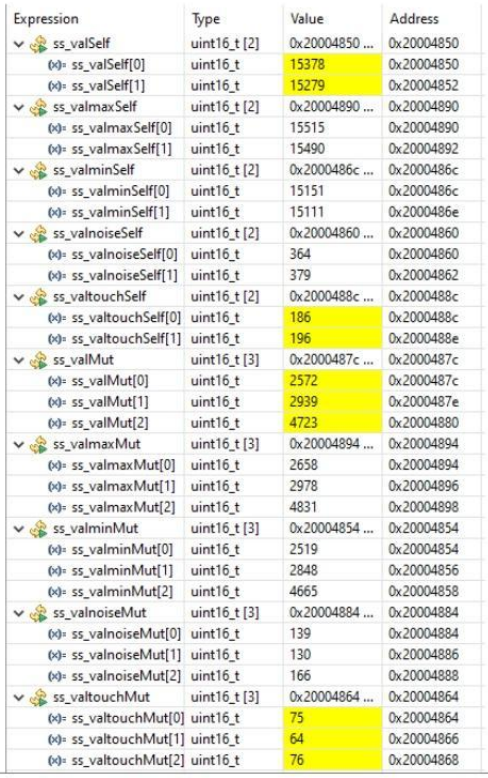
(x)= ss_valtouchMut[1] uint16_t 463 0x200047fa
S2

(x)= ss_valtouchMut[2] uint16_t 648 0x200047fc</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>3</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td></td><td>Actual Result</td><td>
<table>
<tr><td>RA</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>15517</td><td>15253</td><td>264</td><td>2342</td><td>8,87</td></tr>
<tr><td>S48</td><td>15439</td><td>15145</td><td>294</td><td>3606</td><td>12,27</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>2794</td><td>2551</td><td>243</td><td>513</td><td>2,11</td></tr>
<tr><td>S10</td><td>2776</td><td>2581</td><td>195</td><td>463</td><td>2,37</td></tr>
<tr><td>S2</td><td>4497</td><td>4142</td><td>355</td><td>648</td><td>1,83</td></tr>
</table></td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
</table>

Key	SP00014-T12	**Status**	PASS	**Name**	RA tests configuring multi-frequency scan with parallel scanning function with leds
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-11				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:44	**Estimated Time**	00:00	**Actual time**	00:00

Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

	Test Script	
1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
	Actual Result	



Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x20004850 ...	0x20004850
ss_valSelf[0]	uint16_t	15378	0x20004850
ss_valSelf[1]	uint16_t	15279	0x20004852
ss_valmaxSelf	uint16_t [2]	0x20004890 ...	0x20004890
ss_valmaxSelf[0]	uint16_t	15515	0x20004890
ss_valmaxSelf[1]	uint16_t	15490	0x20004892
ss_valminSelf	uint16_t [2]	0x2000486c ...	0x2000486c
ss_valminSelf[0]	uint16_t	15151	0x2000486c
ss_valminSelf[1]	uint16_t	15111	0x2000486e
ss_valnoiseSelf	uint16_t [2]	0x20004860 ...	0x20004860
ss_valnoiseSelf[0]	uint16_t	364	0x20004860
ss_valnoiseSelf[1]	uint16_t	379	0x20004862
ss_valtouchSelf	uint16_t [2]	0x2000488c ...	0x2000488c
ss_valtouchSelf[0]	uint16_t	186	0x2000488c
ss_valtouchSelf[1]	uint16_t	196	0x2000488e
ss_valMut	uint16_t [3]	0x2000487c ...	0x2000487c
ss_valMut[0]	uint16_t	2572	0x2000487c
ss_valMut[1]	uint16_t	2939	0x2000487e
ss_valMut[2]	uint16_t	4723	0x20004880
ss_valmaxMut	uint16_t [3]	0x20004894 ...	0x20004894
ss_valmaxMut[0]	uint16_t	2658	0x20004894
ss_valmaxMut[1]	uint16_t	2978	0x20004896
ss_valmaxMut[2]	uint16_t	4831	0x20004898
ss_valminMut	uint16_t [3]	0x20004854 ...	0x20004854
ss_valminMut[0]	uint16_t	2519	0x20004854
ss_valminMut[1]	uint16_t	2848	0x20004856
ss_valminMut[2]	uint16_t	4665	0x20004858
ss_valnoiseMut	uint16_t [3]	0x20004884 ...	0x20004884
ss_valnoiseMut[0]	uint16_t	139	0x20004884
ss_valnoiseMut[1]	uint16_t	130	0x20004886
ss_valnoiseMut[2]	uint16_t	166	0x20004888
ss_valtouchMut	uint16_t [3]	0x20004864 ...	0x20004864
ss_valtouchMut[0]	uint16_t	75	0x20004864
ss_valtouchMut[1]	uint16_t	64	0x20004866
ss_valtouchMut[2]	uint16_t	76	0x20004868

	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
	Actual Result	

S49

ss_valtouchSelf[0]	uint16_t	2434	0x2000488c

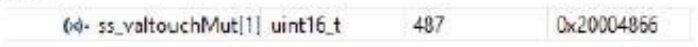
S48

ss_valtouchSelf[1]	uint16_t	3101	0x2000488e

S18

ss_valtouchMut[0]	uint16_t	443	0x20004864

S10



ss_valtouchMut[1]	uint16_t	487	0x20004866

S2

ss_valtouchMut[2]	uint16_t	599	0x20004868

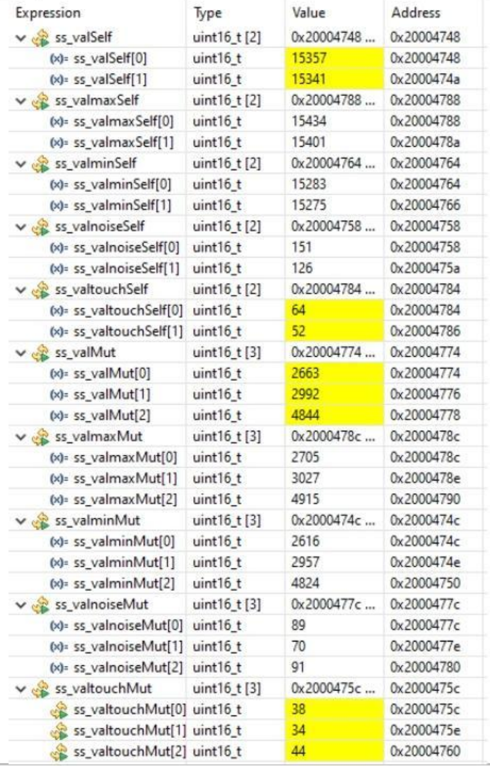
	Issues	-
3	**Status**	PASS
	Details	Calculate SNR (touch/noise value)
	Test Data	-
	Expected Result	SNR
	Actual Result	

RA parallel	Max NT	Min NT	Noise	Max T	SNR
Self but					
S49	15515	15151	364	2434	6,69
S48	15490	15111	379	3101	8,18
Mutual but					

	S18	2658	2519	139	443	3,19
	S10	2978	2848	130	487	3,75
	S2	4831	4665	166	499	3,01
Issues	-					

Key	SP00014-T11	**Status**	PASS	**Name**	RA tests configuring multi-frequency scan with parallel scanning function without leds	
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA					
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)					
Coverage (issues)	SP00014-11					
Coverage (confluence pages)						
Coverage (web links)						
Actual end date	2021-12-27, 11:44	**Estimated Time**	00:00	**Actual time**	00:00	
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution	
Executed by	Nira Tubert					
Issues	-					

Test Script

	Status	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
1	**Actual Result**	

Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x20004748 ...	0x20004748
ss_valSelf[0]	uint16_t	15357	0x20004748
ss_valSelf[1]	uint16_t	15341	0x2000474a
ss_valmaxSelf	uint16_t [2]	0x20004788 ...	0x20004788
ss_valmaxSelf[0]	uint16_t	15434	0x20004788
ss_valmaxSelf[1]	uint16_t	15401	0x2000478a
ss_valminSelf	uint16_t [2]	0x20004764 ...	0x20004764
ss_valminSelf[0]	uint16_t	15283	0x20004764
ss_valminSelf[1]	uint16_t	15275	0x20004766
ss_valnoiseSelf	uint16_t [2]	0x20004758 ...	0x20004758
ss_valnoiseSelf[0]	uint16_t	151	0x20004758
ss_valnoiseSelf[1]	uint16_t	126	0x2000475a
ss_valtouchSelf	uint16_t [2]	0x20004784 ...	0x20004784
ss_valtouchSelf[0]	uint16_t	64	0x20004784
ss_valtouchSelf[1]	uint16_t	52	0x20004786
ss_valMut	uint16_t [3]	0x20004774 ...	0x20004774
ss_valMut[0]	uint16_t	2663	0x20004774
ss_valMut[1]	uint16_t	2992	0x20004776
ss_valMut[2]	uint16_t	4844	0x20004778
ss_valmaxMut	uint16_t [3]	0x2000478c ...	0x2000478c
ss_valmaxMut[0]	uint16_t	2705	0x2000478c
ss_valmaxMut[1]	uint16_t	3027	0x2000478e
ss_valmaxMut[2]	uint16_t	4915	0x20004790
ss_valminMut	uint16_t [3]	0x2000474c ...	0x2000474c
ss_valminMut[0]	uint16_t	2616	0x2000474c
ss_valminMut[1]	uint16_t	2957	0x2000474e
ss_valminMut[2]	uint16_t	4824	0x20004750
ss_valnoiseMut	uint16_t [3]	0x2000477c ...	0x2000477c
ss_valnoiseMut[0]	uint16_t	89	0x2000477c
ss_valnoiseMut[1]	uint16_t	70	0x2000477e
ss_valnoiseMut[2]	uint16_t	91	0x20004780
ss_valtouchMut	uint16_t [3]	0x2000475c ...	0x2000475c
ss_valtouchMut[0]	uint16_t	38	0x2000475c
ss_valtouchMut[1]	uint16_t	34	0x2000475e
ss_valtouchMut[2]	uint16_t	44	0x20004760

	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
	Actual Result	S49  S48 

S49

ss_valtouchSelf[0]	uint16_t	2392	0x20004784

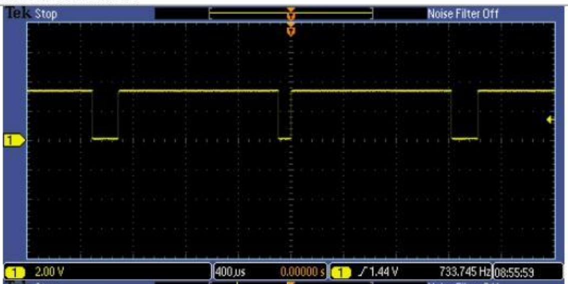
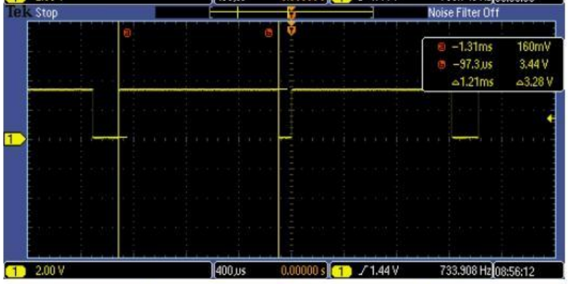
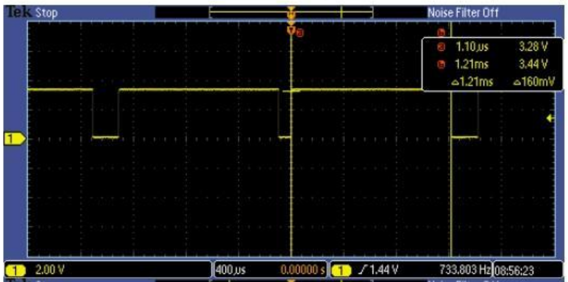
S48

ss_valnoiseSelf[1]	uint16_t	3833	0x2000475a

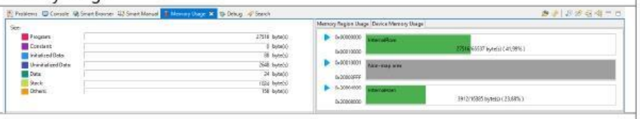
<table>
<tr><td></td><td></td><td>S18

ss_valtouchMut[0] uint16_t 346 0x2000475c
S10

ss_valtouchMut[1] uint16_t 514 0x2000475e
S2

ss_valtouchMut[2] uint16_t 557 0x20004760</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td></td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td>3</td><td>Actual Result</td><td>
<table>
<tr><td>RA parallel</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>15434</td><td>15283</td><td>151</td><td>2392</td><td>15,84</td></tr>
<tr><td>S48</td><td>15401</td><td>15275</td><td>126</td><td>3833</td><td>30,42</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>2705</td><td>2616</td><td>89</td><td>346</td><td>3,89</td></tr>
<tr><td>S10</td><td>3027</td><td>2957</td><td>70</td><td>514</td><td>7,34</td></tr>
<tr><td>S2</td><td>4915</td><td>4824</td><td>91</td><td>557</td><td>6,12</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>4</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check measurement time</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>Measurement time</td></tr>
<tr><td></td><td>Actual Result</td><td>


 </td></tr>
</table>

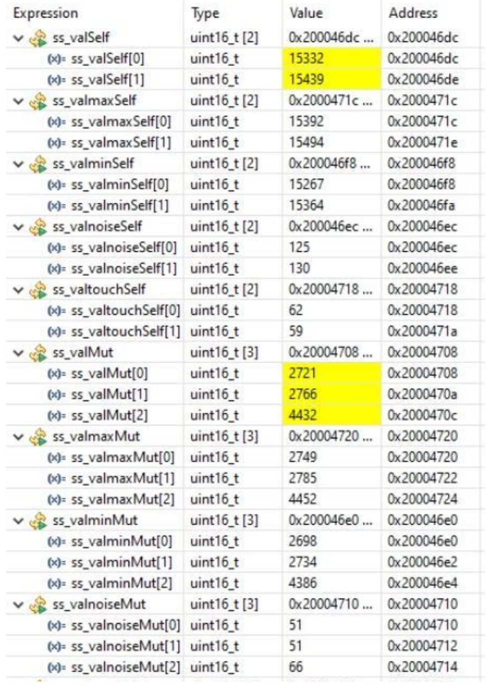
Measurement time (ms)	
Self	1,21
Mutual	1,21
Total	2,42
Global	2,73

	Issues	-
5	**Status**	PASS
	Details	Check memory usage
	Test Data	-
	Expected Result	Memory usage
	Actual Result	
	Issues	-

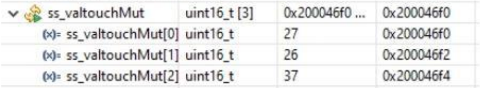
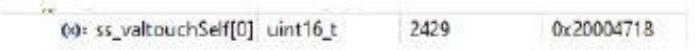
Key	SP00014-T9	**Status**	PASS	**Name**	RA tests configuring multi-frequency scan without leds
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-11				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:44	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

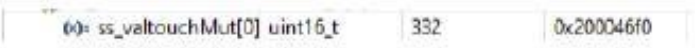
Test Script

1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
	Actual Result	

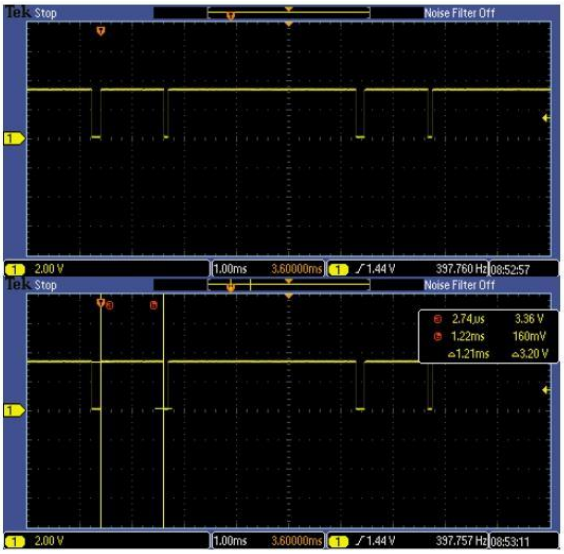


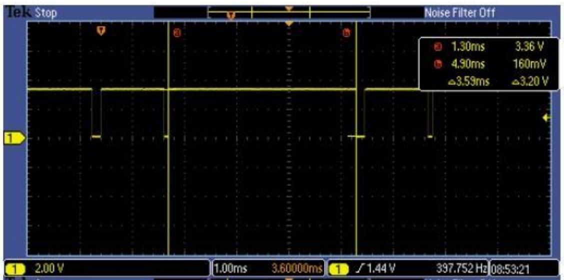
Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x200046dc ...	0x200046dc
(x)= ss_valSelf[0]	uint16_t	15332	0x200046dc
(x)= ss_valSelf[1]	uint16_t	15439	0x200046de
ss_valmaxSelf	uint16_t [2]	0x2000471c ...	0x2000471c
(x)= ss_valmaxSelf[0]	uint16_t	15392	0x2000471c
(x)= ss_valmaxSelf[1]	uint16_t	15494	0x2000471e
ss_valminSelf	uint16_t [2]	0x200046f8 ...	0x200046f8
(x)= ss_valminSelf[0]	uint16_t	15267	0x200046f8
(x)= ss_valminSelf[1]	uint16_t	15364	0x200046fa
ss_valnoiseSelf	uint16_t [2]	0x200046ec ...	0x200046ec
(x)= ss_valnoiseSelf[0]	uint16_t	125	0x200046ec
(x)= ss_valnoiseSelf[1]	uint16_t	130	0x200046ee
ss_valtouchSelf	uint16_t [2]	0x20004718 ...	0x20004718
(x)= ss_valtouchSelf[0]	uint16_t	62	0x20004718
(x)= ss_valtouchSelf[1]	uint16_t	59	0x2000471a
ss_valMut	uint16_t [3]	0x20004708 ...	0x20004708
(x)= ss_valMut[0]	uint16_t	2721	0x20004708
(x)= ss_valMut[1]	uint16_t	2766	0x2000470a
(x)= ss_valMut[2]	uint16_t	4432	0x2000470c
ss_valmaxMut	uint16_t [3]	0x20004720 ...	0x20004720
(x)= ss_valmaxMut[0]	uint16_t	2749	0x20004720
(x)= ss_valmaxMut[1]	uint16_t	2785	0x20004722
(x)= ss_valmaxMut[2]	uint16_t	4452	0x20004724
ss_valminMut	uint16_t [3]	0x200046e0 ...	0x200046e0
(x)= ss_valminMut[0]	uint16_t	2698	0x200046e0
(x)= ss_valminMut[1]	uint16_t	2734	0x200046e2
(x)= ss_valminMut[2]	uint16_t	4386	0x200046e4
ss_valnoiseMut	uint16_t [3]	0x20004710 ...	0x20004710
(x)= ss_valnoiseMut[0]	uint16_t	51	0x20004710
(x)= ss_valnoiseMut[1]	uint16_t	51	0x20004712
(x)= ss_valnoiseMut[2]	uint16_t	66	0x20004714

<table>
<tr><td></td><td></td><td>
ss_valtouchMut uint16_t [3] 0x200046f0 ... 0x200046f0
(x)= ss_valtouchMut[0] uint16_t 27 0x200046f0
(x)= ss_valtouchMut[1] uint16_t 26 0x200046f2
(x)= ss_valtouchMut[2] uint16_t 37 0x200046f4</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>2</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check touch value (max-reference difference while touching)</td></tr>
<tr><td></td><td>Test Data</td><td>ss_valtouchSelf, ss_valtouchMut</td></tr>
<tr><td></td><td>Expected Result</td><td>Maximum measured signal</td></tr>
<tr><td></td><td>Actual Result</td><td>S49

(x)= ss_valtouchSelf[0] uint16_t 2429 0x20004718
S48:

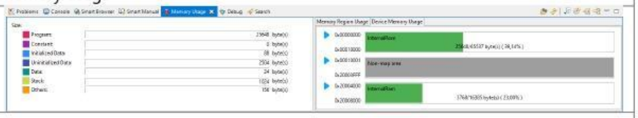
(x)= ss_valtouchSelf[1] uint16_t 3542 0x2000471a
S18

(x)= ss_valtouchMut[0] uint16_t 332 0x200046f0
S10

(x)= ss_valtouchMut[1] uint16_t 425 0x200046f2
S2

(x)= ss_valtouchMut[2] uint16_t 492 0x200046f4</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>3</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td></td><td>Actual Result</td><td>
<table>
<tr><td>RA</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>15392</td><td>15267</td><td>125</td><td>2429</td><td>19,43</td></tr>
<tr><td>S48</td><td>15494</td><td>15364</td><td>130</td><td>3542</td><td>27,25</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>2749</td><td>2698</td><td>51</td><td>332</td><td>6,51</td></tr>
<tr><td>S10</td><td>2785</td><td>2734</td><td>51</td><td>425</td><td>8,33</td></tr>
<tr><td>S2</td><td>4452</td><td>4386</td><td>66</td><td>492</td><td>7,45</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>4</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check measurement time</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>Measurement time</td></tr>
<tr><td></td><td>Actual Result</td><td></td></tr>
<tr><td></td><td></td><td>
 </td></tr>
</table>



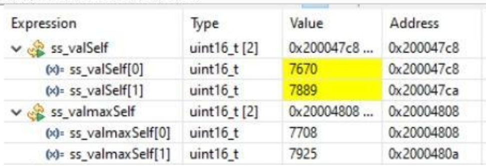
Measurement time (ms)	
Self	1,21
Mutual	3,59
Total	4,80
Global	5,07

	Issues	-
5	**Status**	PASS
	Details	Check memory usage
	Test Data	-
	Expected Result	Memory usage
	Actual Result	
	Issues	-

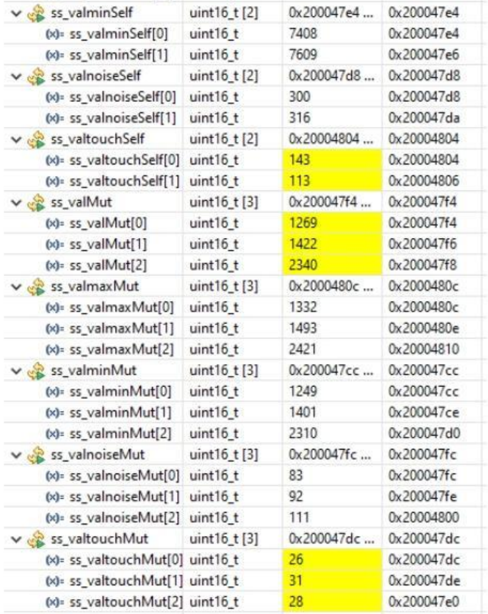
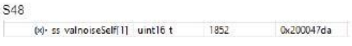
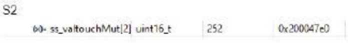
Key	SP00014-T8	**Status**	PASS	**Name**	RA tests configuring one frequency with parallel scan with leds

Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-11				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:44	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

Test Script

	Status	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
		

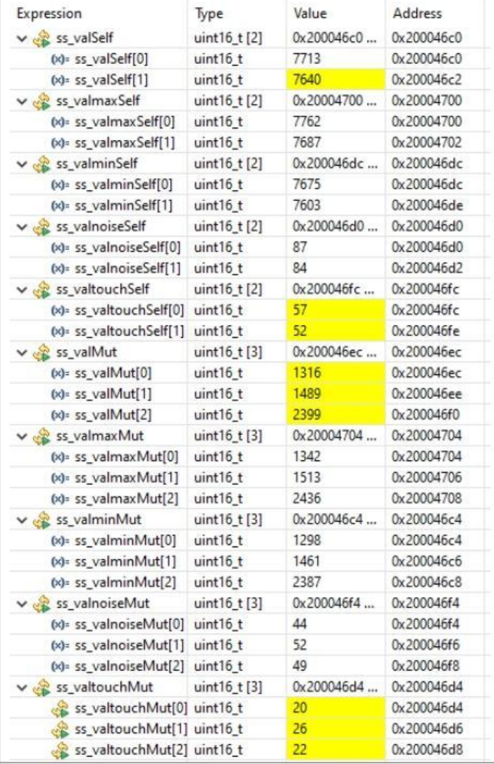
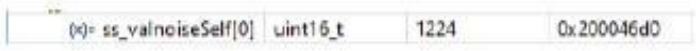
Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x200047c8 ...	0x200047c8
(x)= ss_valSelf[0]	uint16_t	7670	0x200047c8
(x)= ss_valSelf[1]	uint16_t	7889	0x200047ca
ss_valmaxSelf	uint16_t [2]	0x20004808 ...	0x20004808
(x)= ss_valmaxSelf[0]	uint16_t	7708	0x20004808
(x)= ss_valmaxSelf[1]	uint16_t	7925	0x2000480a

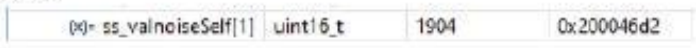
1	**Actual Result**	ss_valminSelf uint16_t [2] 0x200047e4 ... 0x200047e4 (x)= ss_valminSelf[0] uint16_t 7408 0x200047e4 (x)= ss_valminSelf[1] uint16_t 7609 0x200047e6 ss_valnoiseSelf uint16_t [2] 0x200047d8 ... 0x200047d8 (x)= ss_valnoiseSelf[0] uint16_t 300 0x200047d8 (x)= ss_valnoiseSelf[1] uint16_t 316 0x200047da ss_valtouchSelf uint16_t [2] 0x20004804 ... 0x20004804 (x)= ss_valtouchSelf[0] uint16_t 143 0x20004804 (x)= ss_valtouchSelf[1] uint16_t 113 0x20004806 ss_valMut uint16_t [3] 0x200047f4 ... 0x200047f4 (x)= ss_valMut[0] uint16_t 1269 0x200047f4 (x)= ss_valMut[1] uint16_t 1422 0x200047f6 (x)= ss_valMut[2] uint16_t 2340 0x200047f8 ss_valmaxMut uint16_t [3] 0x2000480c ... 0x2000480c (x)= ss_valmaxMut[0] uint16_t 1332 0x2000480c (x)= ss_valmaxMut[1] uint16_t 1493 0x2000480e (x)= ss_valmaxMut[2] uint16_t 2421 0x20004810 ss_valminMut uint16_t [3] 0x200047cc ... 0x200047cc (x)= ss_valminMut[0] uint16_t 1249 0x200047cc (x)= ss_valminMut[1] uint16_t 1401 0x200047ce (x)= ss_valminMut[2] uint16_t 2310 0x200047d0 ss_valnoiseMut uint16_t [3] 0x200047fc ... 0x200047fc (x)= ss_valnoiseMut[0] uint16_t 83 0x200047fc (x)= ss_valnoiseMut[1] uint16_t 92 0x200047fe (x)= ss_valnoiseMut[2] uint16_t 111 0x20004800 ss_valtouchMut uint16_t [3] 0x200047dc ... 0x200047dc (x)= ss_valtouchMut[0] uint16_t 26 0x200047dc (x)= ss_valtouchMut[1] uint16_t 31 0x200047de (x)= ss_valtouchMut[2] uint16_t 28 0x200047e0
	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
	Actual Result	S49 (x)= ss_valnoiseSelf[0] uint16_t 1411 0x200047d8 S48 (x)= ss valnoiseSelf[1] uint16 t 1852 0x200047da S18 (x)= ss_valtouchMut[0] uint16_t 176 0x200047dc S10 (x)= ss_valtouchMut[1] uint16_t 221 0x200047de S2 (x)= ss_valtouchMut[2] uint16_t 252 0x200047e0
	Issues	-

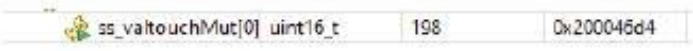
<table>
<tr><td>3</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td></td><td>Actual Result</td><td>
<table>
<tr><td>RA 1 freq</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>7708</td><td>7408</td><td>300</td><td>1411</td><td>4,70</td></tr>
<tr><td>S48</td><td>7925</td><td>7609</td><td>316</td><td>1852</td><td>5,86</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>1332</td><td>1249</td><td>83</td><td>176</td><td>2,12</td></tr>
<tr><td>S10</td><td>1493</td><td>1401</td><td>92</td><td>221</td><td>2,40</td></tr>
<tr><td>S2</td><td>2421</td><td>2310</td><td>111</td><td>252</td><td>2,27</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
</table>

<table>
<tr><td>Key</td><td>SP00014-T7</td><td>Status</td><td>PASS</td><td>Name</td><td>RA tests configuring one frequency with parallel scan without leds</td></tr>
<tr><td>Objective</td><td colspan="5">To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA</td></tr>
<tr><td>Precondition</td><td colspan="5">Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)</td></tr>
<tr><td>Coverage (issues)</td><td colspan="5">SP00014-11</td></tr>
<tr><td>Coverage (confluence pages)</td><td colspan="5"></td></tr>
<tr><td>Coverage (web links)</td><td colspan="5"></td></tr>
<tr><td>Actual end date</td><td>2021-12-27, 11:44</td><td>Estimated Time</td><td>00:00</td><td>Actual time</td><td>00:00</td></tr>
</table>

Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

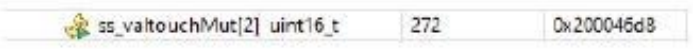
<table>
<tr><th colspan="3">Test Script</th></tr>
<tr><td>1</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check max, min, and noise values without touching</td></tr>
<tr><td></td><td>Test Data</td><td>ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf
ss_valmaxMut, ss_valminMut, ss_valnoiseMut</td></tr>
<tr><td></td><td>Expected Result</td><td>Max, min and noise values</td></tr>
<tr><td></td><td>Actual Result</td><td>

<table>
<tr><th>Expression</th><th>Type</th><th>Value</th><th>Address</th></tr>
<tr><td>ss_valSelf</td><td>uint16_t [2]</td><td>0x200046c0 ...</td><td>0x200046c0</td></tr>
<tr><td>(x)= ss_valSelf[0]</td><td>uint16_t</td><td>7713</td><td>0x200046c0</td></tr>
<tr><td>(x)= ss_valSelf[1]</td><td>uint16_t</td><td>7640</td><td>0x200046c2</td></tr>
<tr><td>ss_valmaxSelf</td><td>uint16_t [2]</td><td>0x20004700 ...</td><td>0x20004700</td></tr>
<tr><td>(x)= ss_valmaxSelf[0]</td><td>uint16_t</td><td>7762</td><td>0x20004700</td></tr>
<tr><td>(x)= ss_valmaxSelf[1]</td><td>uint16_t</td><td>7687</td><td>0x20004702</td></tr>
<tr><td>ss_valminSelf</td><td>uint16_t [2]</td><td>0x200046dc ...</td><td>0x200046dc</td></tr>
<tr><td>(x)= ss_valminSelf[0]</td><td>uint16_t</td><td>7675</td><td>0x200046dc</td></tr>
<tr><td>(x)= ss_valminSelf[1]</td><td>uint16_t</td><td>7603</td><td>0x200046de</td></tr>
<tr><td>ss_valnoiseSelf</td><td>uint16_t [2]</td><td>0x200046d0 ...</td><td>0x200046d0</td></tr>
<tr><td>(x)= ss_valnoiseSelf[0]</td><td>uint16_t</td><td>87</td><td>0x200046d0</td></tr>
<tr><td>(x)= ss_valnoiseSelf[1]</td><td>uint16_t</td><td>84</td><td>0x200046d2</td></tr>
<tr><td>ss_valtouchSelf</td><td>uint16_t [2]</td><td>0x200046fc ...</td><td>0x200046fc</td></tr>
<tr><td>(x)= ss_valtouchSelf[0]</td><td>uint16_t</td><td>57</td><td>0x200046fc</td></tr>
<tr><td>(x)= ss_valtouchSelf[1]</td><td>uint16_t</td><td>52</td><td>0x200046fe</td></tr>
<tr><td>ss_valMut</td><td>uint16_t [3]</td><td>0x200046ec ...</td><td>0x200046ec</td></tr>
<tr><td>(x)= ss_valMut[0]</td><td>uint16_t</td><td>1316</td><td>0x200046ec</td></tr>
<tr><td>(x)= ss_valMut[1]</td><td>uint16_t</td><td>1489</td><td>0x200046ee</td></tr>
<tr><td>(x)= ss_valMut[2]</td><td>uint16_t</td><td>2399</td><td>0x200046f0</td></tr>
<tr><td>ss_valmaxMut</td><td>uint16_t [3]</td><td>0x20004704 ...</td><td>0x20004704</td></tr>
<tr><td>(x)= ss_valmaxMut[0]</td><td>uint16_t</td><td>1342</td><td>0x20004704</td></tr>
<tr><td>(x)= ss_valmaxMut[1]</td><td>uint16_t</td><td>1513</td><td>0x20004706</td></tr>
<tr><td>(x)= ss_valmaxMut[2]</td><td>uint16_t</td><td>2436</td><td>0x20004708</td></tr>
<tr><td>ss_valminMut</td><td>uint16_t [3]</td><td>0x200046c4 ...</td><td>0x200046c4</td></tr>
<tr><td>(x)= ss_valminMut[0]</td><td>uint16_t</td><td>1298</td><td>0x200046c4</td></tr>
<tr><td>(x)= ss_valminMut[1]</td><td>uint16_t</td><td>1461</td><td>0x200046c6</td></tr>
<tr><td>(x)= ss_valminMut[2]</td><td>uint16_t</td><td>2387</td><td>0x200046c8</td></tr>
<tr><td>ss_valnoiseMut</td><td>uint16_t [3]</td><td>0x200046f4 ...</td><td>0x200046f4</td></tr>
<tr><td>(x)= ss_valnoiseMut[0]</td><td>uint16_t</td><td>44</td><td>0x200046f4</td></tr>
<tr><td>(x)= ss_valnoiseMut[1]</td><td>uint16_t</td><td>52</td><td>0x200046f6</td></tr>
<tr><td>(x)= ss_valnoiseMut[2]</td><td>uint16_t</td><td>49</td><td>0x200046f8</td></tr>
<tr><td>ss_valtouchMut</td><td>uint16_t [3]</td><td>0x200046d4 ...</td><td>0x200046d4</td></tr>
<tr><td>ss_valtouchMut[0]</td><td>uint16_t</td><td>20</td><td>0x200046d4</td></tr>
<tr><td>ss_valtouchMut[1]</td><td>uint16_t</td><td>26</td><td>0x200046d6</td></tr>
<tr><td>ss_valtouchMut[2]</td><td>uint16_t</td><td>22</td><td>0x200046d8</td></tr>
</table>
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>2</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Check touch value (max-reference difference while touching)</td></tr>
<tr><td></td><td>Test Data</td><td>ss_valtouchSelf, ss_valtouchMut</td></tr>
<tr><td></td><td>Expected Result</td><td>Maximum measured signal</td></tr>
<tr><td></td><td>Actual Result</td><td>
S49

(x)= ss_valnoiseSelf[0] | uint16_t | 1224 | 0x200046d0

S48

(x)= ss_valnoiseSelf[1] | uint16_t | 1904 | 0x200046d2

S18

ss_valtouchMut[0] | uint16_t | 198 | 0x200046d4

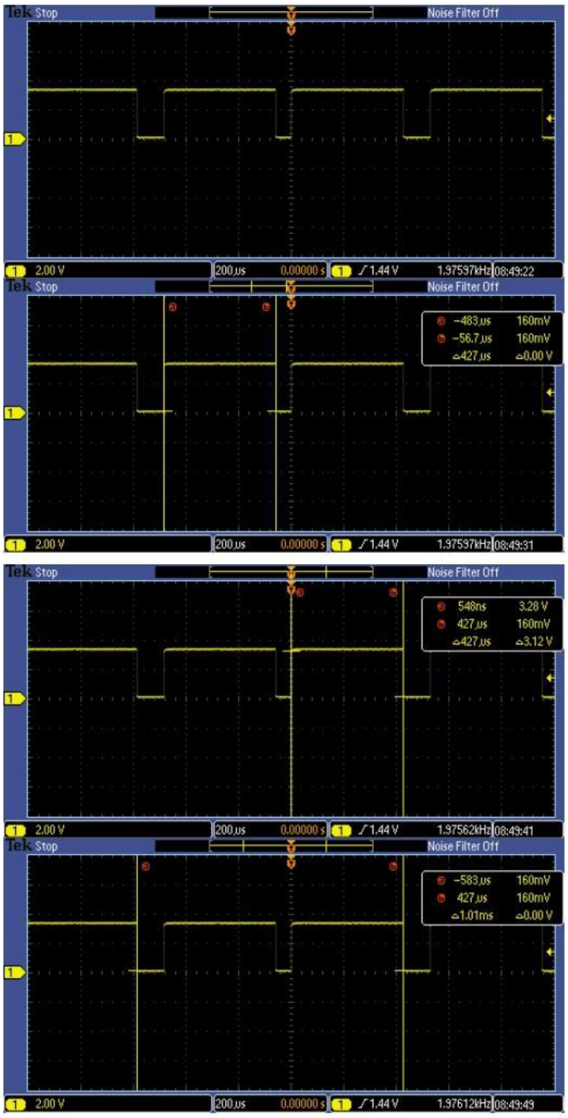
S10

ss_valtouchMut[1] | uint16_t | 241 | 0x200046d6

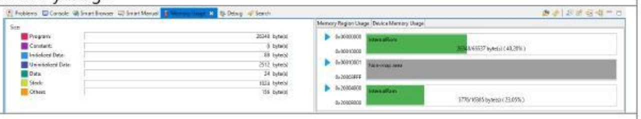
S2

ss_valtouchMut[2] | uint16_t | 272 | 0x200046d8
</td></tr>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td>3</td><td>Status</td><td>PASS</td></tr>
<tr><td></td><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td></td><td>Test Data</td><td>-</td></tr>
<tr><td></td><td>Expected Result</td><td>SNR</td></tr>
<tr><td></td><td>Actual Result</td><td>
<table>
<tr><th>RA 1 freq</th><th>Max NT</th><th>Min NT</th><th>Noise</th><th>Max T</th><th>SNR</th></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>7762</td><td>7675</td><td>87</td><td>1224</td><td>14,07</td></tr>
<tr><td>S48</td><td>7687</td><td>7603</td><td>84</td><td>1904</td><td>22,67</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>1342</td><td>1298</td><td>44</td><td>198</td><td>4,50</td></tr>
<tr><td>S10</td><td>1513</td><td>1461</td><td>52</td><td>241</td><td>4,63</td></tr>
</table>
</td></tr>
</table>

S2	2436	2387	49	272	5,55

	Issues	-
4	**Status**	PASS
	Details	Check measurement time
	Test Data	-
	Expected Result	Measurement time
	Actual Result	

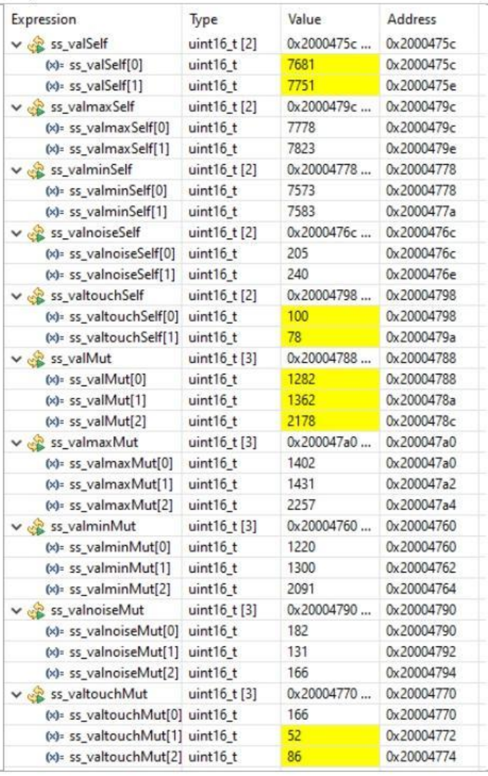


Measurement time (ms)	
Self	0,43
Mutual	0,43
Total	0,85
Global	1,01

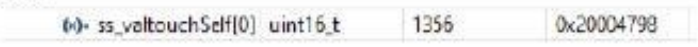
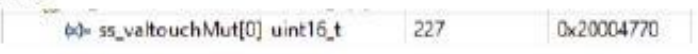
	Issues	-
	Status	PASS
	Details	Check memory usage
	Test Data	-
	Expected Result	Memory usage
5	**Actual Result**	
	Issues	-

Key	SP00014-T6	**Status**	PASS	**Name**	RA tests in former CTSU configuration (0.128ms for self scan time, 0.256ms for mutual scan time, with leds)
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:45	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

Test Script

1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
	Actual Result	

Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x2000475c ...	0x2000475c
ss_valSelf[0]	uint16_t	7681	0x2000475c
ss_valSelf[1]	uint16_t	7751	0x2000475e
ss_valmaxSelf	uint16_t [2]	0x2000479c ...	0x2000479c
ss_valmaxSelf[0]	uint16_t	7778	0x2000479c
ss_valmaxSelf[1]	uint16_t	7823	0x2000479e
ss_valminSelf	uint16_t [2]	0x20004778 ...	0x20004778
ss_valminSelf[0]	uint16_t	7573	0x20004778
ss_valminSelf[1]	uint16_t	7583	0x2000477a
ss_valnoiseSelf	uint16_t [2]	0x2000476c ...	0x2000476c
ss_valnoiseSelf[0]	uint16_t	205	0x2000476c
ss_valnoiseSelf[1]	uint16_t	240	0x2000476e
ss_valtouchSelf	uint16_t [2]	0x20004798 ...	0x20004798
ss_valtouchSelf[0]	uint16_t	100	0x20004798
ss_valtouchSelf[1]	uint16_t	78	0x2000479a
ss_valMut	uint16_t [3]	0x20004788 ...	0x20004788
ss_valMut[0]	uint16_t	1282	0x20004788
ss_valMut[1]	uint16_t	1362	0x2000478a
ss_valMut[2]	uint16_t	2178	0x2000478c
ss_valmaxMut	uint16_t [3]	0x200047a0 ...	0x200047a0
ss_valmaxMut[0]	uint16_t	1402	0x200047a0
ss_valmaxMut[1]	uint16_t	1431	0x200047a2
ss_valmaxMut[2]	uint16_t	2257	0x200047a4
ss_valminMut	uint16_t [3]	0x20004760 ...	0x20004760
ss_valminMut[0]	uint16_t	1220	0x20004760
ss_valminMut[1]	uint16_t	1300	0x20004762
ss_valminMut[2]	uint16_t	2091	0x20004764
ss_valnoiseMut	uint16_t [3]	0x20004790 ...	0x20004790
ss_valnoiseMut[0]	uint16_t	182	0x20004790
ss_valnoiseMut[1]	uint16_t	131	0x20004792
ss_valnoiseMut[2]	uint16_t	166	0x20004794
ss_valtouchMut	uint16_t [3]	0x20004770 ...	0x20004770
ss_valtouchMut[0]	uint16_t	166	0x20004770
ss_valtouchMut[1]	uint16_t	52	0x20004772
ss_valtouchMut[2]	uint16_t	86	0x20004774

	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
	Actual Result	S49  S48  S18 

S49

ss_valtouchSelf[0]	uint16_t	1356	0x20004798

S48

ss_valtouchSelf[1]	uint16_t	1833	0x2000479a

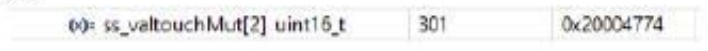
S18

ss_valtouchMut[0]	uint16_t	227	0x20004770

S10

(x)= ss_valtouchMut[1]	uint16_t	237	0x20004772

S2

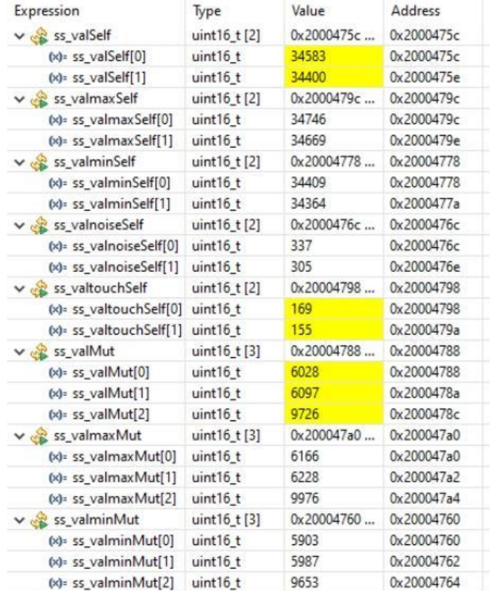


(x)= ss_valtouchMut[2]	uint16_t	301	0x20004774

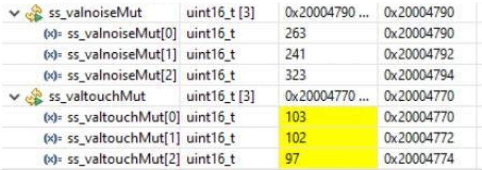
<table>
<tr><td></td><td>Issues</td><td>-</td></tr>
<tr><td rowspan="6">3</td><td>Status</td><td>PASS</td></tr>
<tr><td>Details</td><td>Calculate SNR (touch/noise value)</td></tr>
<tr><td>Test Data</td><td>-</td></tr>
<tr><td>Expected Result</td><td>SNR</td></tr>
<tr><td>Actual Result</td><td>
<table>
<tr><td>RA 1 freq</td><td>Max NT</td><td>Min NT</td><td>Noise</td><td>Max T</td><td>SNR</td></tr>
<tr><td>Self but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S49</td><td>7778</td><td>7573</td><td>205</td><td>1356</td><td>6,61</td></tr>
<tr><td>S48</td><td>7823</td><td>7583</td><td>240</td><td>1883</td><td>7,85</td></tr>
<tr><td>Mutual but</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>S18</td><td>1402</td><td>1220</td><td>182</td><td>227</td><td>1,25</td></tr>
<tr><td>S10</td><td>1431</td><td>1300</td><td>131</td><td>237</td><td>1,81</td></tr>
<tr><td>S2</td><td>2257</td><td>2091</td><td>166</td><td>301</td><td>1,81</td></tr>
</table>
</td></tr>
<tr><td>Issues</td><td>-</td></tr>
</table>

Key	SP00014-T4	**Status**	PASS	**Name**	RA tests in former CTSU configuration (0.576ms for self scan time, 1.152ms for mutual scan time, with leds)
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:45	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

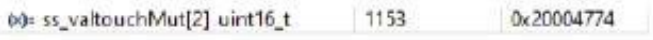
	Test Script	
1	**Status**	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values
	Actual Result	



Expression	Type	Value	Address
ss_valSelf	uint16_t [2]	0x2000475c ...	0x2000475c
(x)= ss_valSelf[0]	uint16_t	34583	0x2000475c
(x)= ss_valSelf[1]	uint16_t	34400	0x2000475e
ss_valmaxSelf	uint16_t [2]	0x2000479c ...	0x2000479c
(x)= ss_valmaxSelf[0]	uint16_t	34746	0x2000479c
(x)= ss_valmaxSelf[1]	uint16_t	34669	0x2000479e
ss_valminSelf	uint16_t [2]	0x20004778 ...	0x20004778
(x)= ss_valminSelf[0]	uint16_t	34409	0x20004778
(x)= ss_valminSelf[1]	uint16_t	34364	0x2000477a
ss_valnoiseSelf	uint16_t [2]	0x2000476c ...	0x2000476c
(x)= ss_valnoiseSelf[0]	uint16_t	337	0x2000476c
(x)= ss_valnoiseSelf[1]	uint16_t	305	0x2000476e
ss_valtouchSelf	uint16_t [2]	0x20004798 ...	0x20004798
(x)= ss_valtouchSelf[0]	uint16_t	169	0x20004798
(x)= ss_valtouchSelf[1]	uint16_t	155	0x2000479a
ss_valMut	uint16_t [3]	0x20004788 ...	0x20004788
(x)= ss_valMut[0]	uint16_t	6028	0x20004788
(x)= ss_valMut[1]	uint16_t	6097	0x2000478a
(x)= ss_valMut[2]	uint16_t	9726	0x2000478c
ss_valmaxMut	uint16_t [3]	0x200047a0 ...	0x200047a0
(x)= ss_valmaxMut[0]	uint16_t	6166	0x200047a0
(x)= ss_valmaxMut[1]	uint16_t	6228	0x200047a2
(x)= ss_valmaxMut[2]	uint16_t	9976	0x200047a4
ss_valminMut	uint16_t [3]	0x20004760 ...	0x20004760
(x)= ss_valminMut[0]	uint16_t	5903	0x20004760
(x)= ss_valminMut[1]	uint16_t	5987	0x20004762
(x)= ss_valminMut[2]	uint16_t	9653	0x20004764



⌄ ss_valnoiseMut	uint16_t [3]	0x20004790 ...	0x20004790
(x)= ss_valnoiseMut[0]	uint16_t	263	0x20004790
(x)= ss_valnoiseMut[1]	uint16_t	241	0x20004792
(x)= ss_valnoiseMut[2]	uint16_t	323	0x20004794
⌄ ss_valtouchMut	uint16_t [3]	0x20004770 ...	0x20004770
(x)= ss_valtouchMut[0]	uint16_t	103	0x20004770
(x)= ss_valtouchMut[1]	uint16_t	102	0x20004772
(x)= ss_valtouchMut[2]	uint16_t	97	0x20004774

	Issues	-
	Status	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
2	**Actual Result**	S49:  (x)= ss_valtouchSelf[0] uint16_t 4655 0x20004798 S48:  (x)= ss_valtouchSelf[1] uint16_t 8778 0x2000479a S18:  (x)= ss_valtouchMut[0] uint16_t 942 0x20004770 S10:  (x)= ss_valtouchMut[1] uint16_t 898 0x20004772 S2:  (x)= ss_valtouchMut[2] uint16_t 1153 0x20004774
	Issues	-
	Status	PASS
	Details	Calculate SNR (touch/noise value)
	Test Data	-
	Expected Result	SNR
3	**Actual Result**	

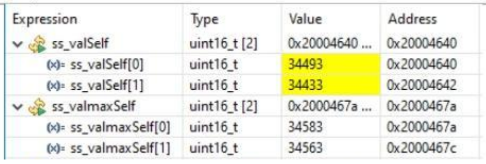
RA 1 freq	Max NT	Min NT	Noise	Max T	SNR
Self but					
S49	34746	34409	337	4655	13,81
S48	34669	34364	305	8778	28,78
Mutual but					
S18	6166	5903	263	942	3,58
S10	6228	5987	241	898	3,73
S2	9976	9653	323	1153	3,57

	Issues	-

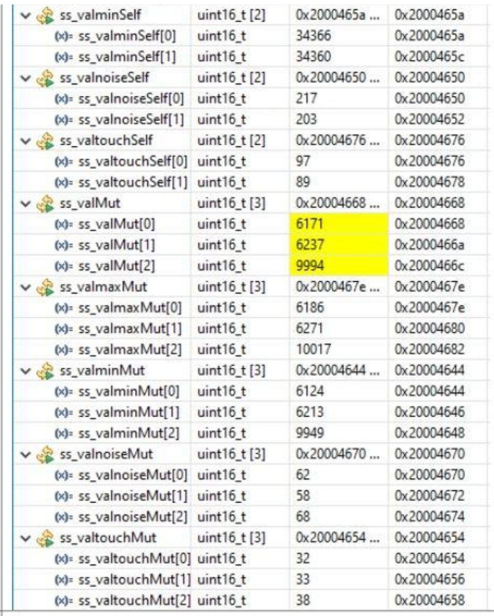
Key	SP00014-T2	**Status**	PASS	**Name**	RA tests in former CTSU configuration (0.576ms for self scan time, 1.152ms for mutual scan time, without leds)
Objective	To find measurement time, maximum measured signal, SNR and Memory usage using a reference finger for T4 RA				
Precondition	Config: 2 self buttons, 3 mutual buttons (1 TX, 3 RX)				
Coverage (issues)	SP00014-10				
Coverage (confluence pages)					
Coverage (web links)					
Actual end date	2021-12-27, 11:45	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Nira Tubert	**Environment**	e2 studio	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

Test Script

	Status	PASS
	Details	Check max, min, and noise values without touching
	Test Data	ss_valmaxSelf, ss_valminSelf, ss_valnoiseSelf ss_valmaxMut, ss_valminMut, ss_valnoiseMut
	Expected Result	Max, min and noise values

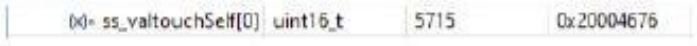


Expression	Type	Value	Address
⌄ ss_valSelf	uint16_t [2]	0x20004640 ...	0x20004640
(x)= ss_valSelf[0]	uint16_t	34493	0x20004640
(x)= ss_valSelf[1]	uint16_t	34433	0x20004642
⌄ ss_valmaxSelf	uint16_t [2]	0x2000467a ...	0x2000467a
(x)= ss_valmaxSelf[0]	uint16_t	34583	0x2000467a
(x)= ss_valmaxSelf[1]	uint16_t	34563	0x2000467c

1	**Actual Result**	

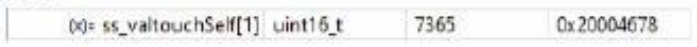
```
v  ss_valminSelf            uint16_t [2]   0x2000465a ...  0x2000465a
     (x)= ss_valminSelf[0]  uint16_t       34366           0x2000465a
     (x)= ss_valminSelf[1]  uint16_t       34360           0x2000465c
v  ss_valnoiseSelf          uint16_t [2]   0x20004650 ...  0x20004650
     (x)= ss_valnoiseSelf[0] uint16_t      217             0x20004650
     (x)= ss_valnoiseSelf[1] uint16_t      203             0x20004652
v  ss_valtouchSelf          uint16_t [2]   0x20004676 ...  0x20004676
     (x)= ss_valtouchSelf[0] uint16_t      97              0x20004676
     (x)= ss_valtouchSelf[1] uint16_t      89              0x20004678
v  ss_valMut                uint16_t [3]   0x20004668 ...  0x20004668
     (x)= ss_valMut[0]      uint16_t       6171            0x20004668
     (x)= ss_valMut[1]      uint16_t       6237            0x2000466a
     (x)= ss_valMut[2]      uint16_t       9994            0x2000466c
v  ss_valmaxMut             uint16_t [3]   0x2000467e ...  0x2000467e
     (x)= ss_valmaxMut[0]   uint16_t       6186            0x2000467e
     (x)= ss_valmaxMut[1]   uint16_t       6271            0x20004680
     (x)= ss_valmaxMut[2]   uint16_t       10017           0x20004682
v  ss_valminMut             uint16_t [3]   0x20004644 ...  0x20004644
     (x)= ss_valminMut[0]   uint16_t       6124            0x20004644
     (x)= ss_valminMut[1]   uint16_t       6213            0x20004646
     (x)= ss_valminMut[2]   uint16_t       9949            0x20004648
v  ss_valnoiseMut           uint16_t [3]   0x20004670 ...  0x20004670
     (x)= ss_valnoiseMut[0] uint16_t       62              0x20004670
     (x)= ss_valnoiseMut[1] uint16_t       58              0x20004672
     (x)= ss_valnoiseMut[2] uint16_t       68              0x20004674
v  ss_valtouchMut           uint16_t [3]   0x20004654 ...  0x20004654
     (x)= ss_valtouchMut[0] uint16_t       32              0x20004654
     (x)= ss_valtouchMut[1] uint16_t       33              0x20004656
     (x)= ss_valtouchMut[2] uint16_t       38              0x20004658
```

	Issues	-
2	**Status**	PASS
	Details	Check touch value (max-reference difference while touching)
	Test Data	ss_valtouchSelf, ss_valtouchMut
	Expected Result	Maximum measured signal
	Actual Result	S49



```
(x)= ss_valtouchSelf[0]  uint16_t  5715  0x20004676
```

S48



```
(x)= ss_valtouchSelf[1]  uint16_t  7365  0x20004678
```

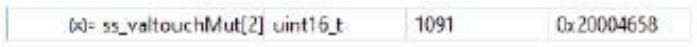
S18

```
(x)= ss_valtouchMut[0]  uint16_t  972  0x20004654
```

S10

```
(x)= ss_valtouchMut[1]  uint16_t  925  0x20004656
```

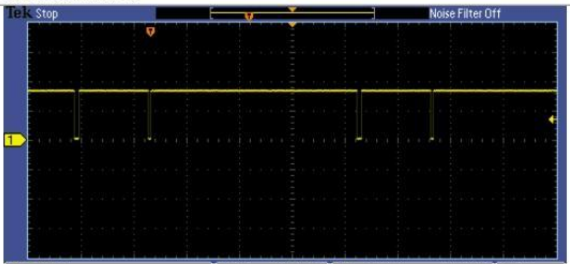
S2

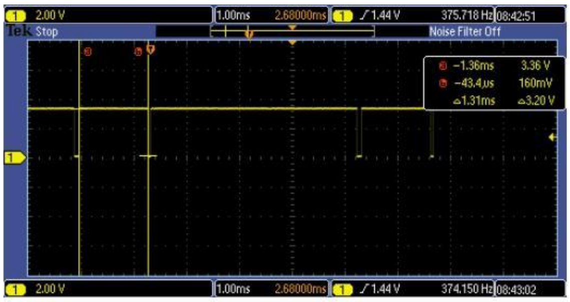


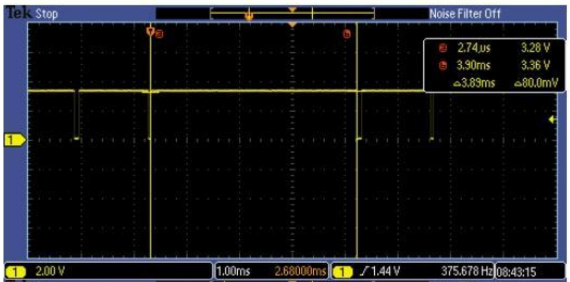
```
(x)= ss_valtouchMut[2]  uint16_t  1091  0x20004658
```

	Issues	-
3	**Status**	PASS
	Details	Calculate SNR (touch/noise value)
	Test Data	-
	Expected Result	SNR
	Actual Result	

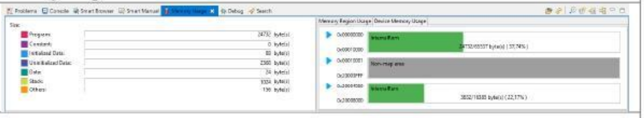
RA 1 freq	Max NT	Min NT	Noise	Max T	SNR
Self but					
S49	34583	34366	217	5715	26,34
S48	34563	34360	203	7365	36,28
Mutual but					
S18	6186	6124	62	972	15,68
S10	6271	6213	58	925	15,95
S2	10017	9949	68	1091	16,04

	Issues	-
4	**Status**	PASS
	Details	Check measurement time
	Test Data	-
	Expected Result	Measurement time
	Actual Result	

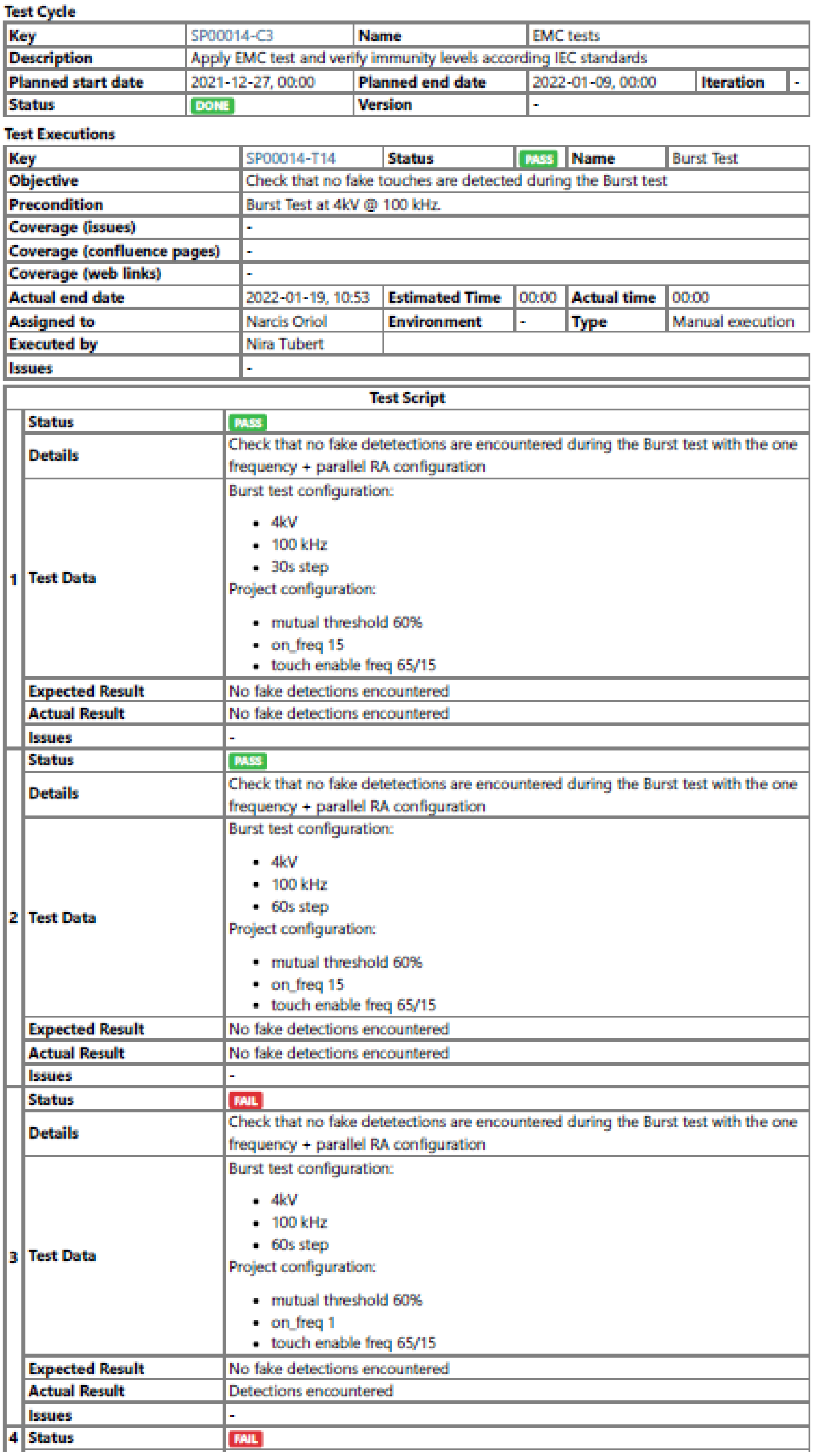




Measurement time (ms)	
Self	1,31
Mutual	3,89
Total	5,20
Global	5,33

	Issues	-
5	**Status**	PASS
	Details	Check memory usage
	Test Data	-
	Expected Result	Memory usage
	Actual Result	
	Issues	-

EMC test execution results:



Test Cycle

Key	SP00014-C3	**Name**	EMC tests		
Description	Apply EMC test and verify immunity levels according IEC standards				
Planned start date	2021-12-27, 00:00	**Planned end date**	2022-01-09, 00:00	**Iteration**	-
Status	DONE	**Version**	-		

Test Executions

Key	SP00014-T14	**Status**	PASS	**Name**	Burst Test
Objective	Check that no fake touches are detected during the Burst test				
Precondition	Burst Test at 4kV @ 100 kHz.				
Coverage (issues)	-				
Coverage (confluence pages)	-				
Coverage (web links)	-				
Actual end date	2022-01-19, 10:53	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Narcis Oriol	**Environment**	-	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

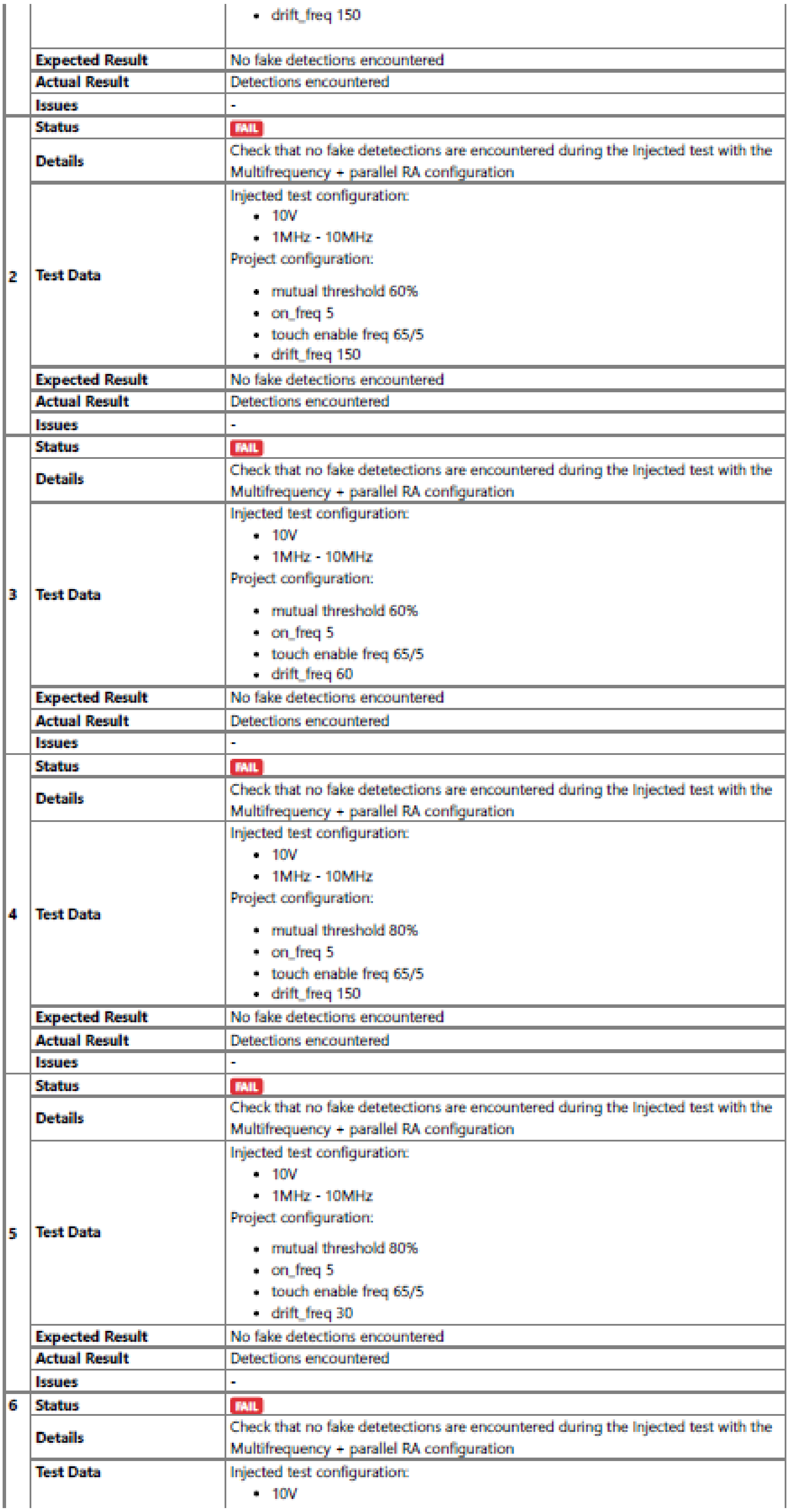
Test Script

1	**Status**	PASS
	Details	Check that no fake detetections are encountered during the Burst test with the one frequency + parallel RA configuration
	Test Data	Burst test configuration: • 4kV • 100 kHz • 30s step Project configuration: • mutual threshold 60% • on_freq 15 • touch enable freq 65/15
	Expected Result	No fake detections encountered
	Actual Result	No fake detections encountered
	Issues	-
2	**Status**	PASS
	Details	Check that no fake detetections are encountered during the Burst test with the one frequency + parallel RA configuration
	Test Data	Burst test configuration: • 4kV • 100 kHz • 60s step Project configuration: • mutual threshold 60% • on_freq 15 • touch enable freq 65/15
	Expected Result	No fake detections encountered
	Actual Result	No fake detections encountered
	Issues	-
3	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Burst test with the one frequency + parallel RA configuration
	Test Data	Burst test configuration: • 4kV • 100 kHz • 60s step Project configuration: • mutual threshold 60% • on_freq 1 • touch enable freq 65/15
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
4	**Status**	FAIL

	Details	Check that no fake detetections are encountered during the Burst test with the one frequency + parallel RA configuration
	Test Data	Burst test configuration: • 4kV • 100 kHz • 60s step Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/15
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
5	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Burst test with the Multifrequency + parallel RA configuration
	Test Data	Burst test configuration: • 4kV • 100 kHz • 60s step Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/5
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
6	**Status**	PASS
	Details	Check that no fake detetections are encountered during the Burst test with the Multifrequency + parallel RA configuration
	Test Data	Burst test configuration: • 4kV • 100 kHz • 60s step Project configuration: • mutual threshold 80% • on_freq 5 • touch enable freq 65/5
	Expected Result	No fake detections encountered
	Actual Result	No fake detections encountered
	Issues	-

Key	SP00014-T15	**Status**	PASS	**Name**	Injected Test
Objective	Check that no Fake detections are encountered during the Injected test				
Precondition	-				
Coverage (issues)	-				
Coverage (confluence pages)	-				
Coverage (web links)	-				
Actual end date	2022-01-21, 11:45	**Estimated Time**	00:00	**Actual time**	00:00
Assigned to	Narcis Oriol	**Environment**	-	**Type**	Manual execution
Executed by	Nira Tubert				
Issues	-				

	Test Script	
1	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the one frequency + parallel RA configuration
	Test Data	Injected test configuration: • 10V • 1MHz - 10MHz Project configuration: • mutual threshold 60% • on_freq 15 • touch enable freq 65/15



		• drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
2	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 10V • 1MHz - 10MHz Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/5 • drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
3	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 10V • 1MHz - 10MHz Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/5 • drift_freq 60
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
4	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 10V • 1MHz - 10MHz Project configuration: • mutual threshold 80% • on_freq 5 • touch enable freq 65/5 • drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
5	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 10V • 1MHz - 10MHz Project configuration: • mutual threshold 80% • on_freq 5 • touch enable freq 65/5 • drift_freq 30
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
6	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 10V

		• 1MHz - 10MHz Project configuration: • mutual threshold 80% • on_freq 15 • touch enable freq 65/5 • drift_freq 30
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
7	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the one frequency + parallel RA configuration
	Test Data	Injected test configuration: • 6V • 1MHz - 5MHz Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/15 • drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
8	**Status**	FAIL
	Details	Check that no fake detetections are encountered during the Injected test with the one frequency + parallel RA configuration
	Test Data	Injected test configuration: • 6V • 1MHz - 5MHz Project configuration: • mutual threshold 60% • on_freq 15 • touch enable freq 65/15 • drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	Detections encountered
	Issues	-
9	**Status**	PASS
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 6V • 1MHz - 5MHz Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/5 • drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	No fake detections encountered
	Issues	-
10	**Status**	PASS
	Details	Check that no fake detetections are encountered during the Injected test with the Multifrequency + parallel RA configuration
	Test Data	Injected test configuration: • 6V • 1MHz - 10MHz Project configuration: • mutual threshold 60% • on_freq 5 • touch enable freq 65/5 • drift_freq 150
	Expected Result	No fake detections encountered
	Actual Result	No fake detections encountered
	Issues	-

Glossary

API	Application Programming Interface
CFC	Current to Frequency Converter
CSTU	Capacitive Touch Sensing Unit Driver
EMC	Electromagnetic compatibility
EUT	Equipment Under Test
FSP	Flexible Software Package
HMI	Human-Machine Interface
IDE	Integrated Development Environment
PCB	Printed Circuit Board
QE	Quick and Effective Tool Solutions
RA2L1, RA2E1	
	RA: Renesas Advanced Family
	2: Renesas RA2 Series
	L: Ultra-Low power
	E: entry line
RX130	Renesas Xtreme 130 Product Group
SWD	Serial Wire Debug
SNR	Signal to Noise Ratio
TSn	Sensor pins for the CTSU (touch sensor)
Rx	Receiver TS
Tx	Transmitter TS