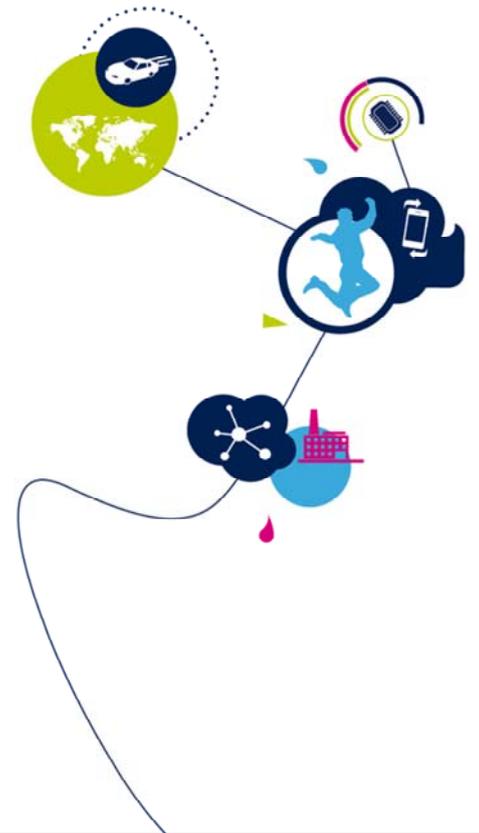


# STM32WB - TSC

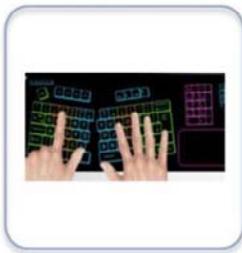
Touch Sensing Controller  
Revision 1.0



Hello, and welcome to this presentation of the STM32 Touch Sensing Controller (TSC) which enables the designer to simply add touch sensing functionality to any application.



- Provides a capacitive sensing technology
  - Surface charge transfer acquisition principle
  - Up to 21 capacitive sensing channels
  - Fully configurable
  - Reduced BOM



### Application benefits

- Pleasant and user-friendly interface
- Reliable touch buttons thanks to a sealed environment
- No need for mechanical parts

Over recent years, Touch Sensing has become quite common in many applications such as mobile phones, induction cooktops and ovens, coffee machines, etc. This type of interface is more flexible and reliable compared to standard push buttons because mechanical parts are no longer needed.

The Touch Sensing Controller (TSC) embedded in STM32WB devices offers a simple way to manage such interfaces. The TSC supports a robust charge transfer acquisition principle with up to 21 capacitive sensing channels. It is fully configurable and only a few external components are required to design a user-friendly interface.

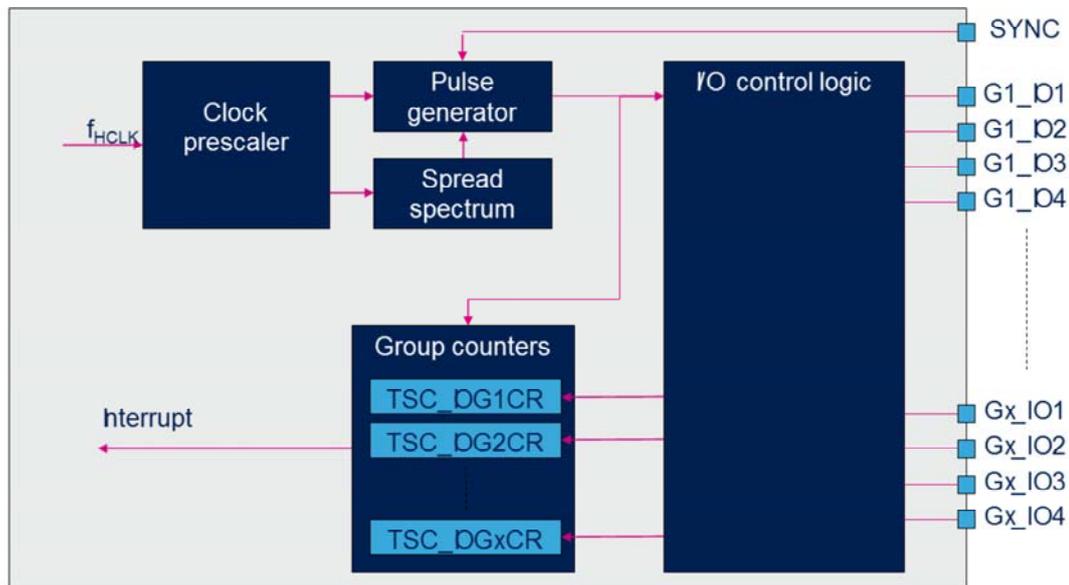
- Proven surface charge transfer acquisition principle available on several STM32 MCU series
- Supports up to 21 capacitive sensing channels split over 1 analog I/O groups
  - Number of channels and analog I/O groups depends on the MCU used
- Up to 7 capacitive sensing channels can be acquired in parallel offering a very good response time
- One sampling capacitor for up to 3 capacitive sensing channels to reduce system components
- Full hardware management of the charge transfer acquisition sequence with spread spectrum to improve system robustness in noisy environments
- Designed to operate with the free STM32Cube touch sensing library available in the corresponding STM32Cube package
  - Enhanced processing features for optimized sensitivity and immunity
  - Compatible with proximity, touchkey, linear and rotary touch sensors



The key features of the Touch Sensing Controller are:

- Proven charge transfer acquisition principle which is available on several STM32 MCU series (STM32F0, STM32F3, STM32L0 and STM32L4).
- Supports up to 21 capacitive sensing channels which are split over 7 analog I/O groups. The number of channels and I/O groups depends on the selected MCU.
- For optimum performance, up to 7 capacitive sensing channels can be acquired in parallel. This offers a very good response time.
- Only one sampling capacitor is needed to manage up to 3 capacitive sensing channels. This ensures a reduced BOM.
- The charge transfer acquisition is fully managed by hardware to reduce CPU overhead. A spread spectrum feature is available to improve system robustness in noisy environments.
- Finally, the Touch Sensing Controller is designed to

operate with the free STM32Cube touch sensing library available in the corresponding STM32Cube package. This library offers all the processing required to develop a robust capacitive sensing solution and supports proximity, touchkey, linear and rotary touch sensors.

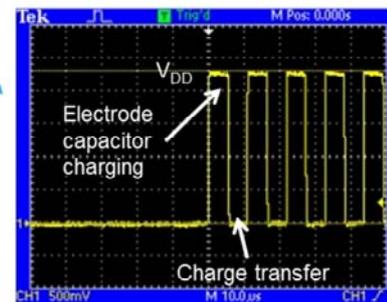
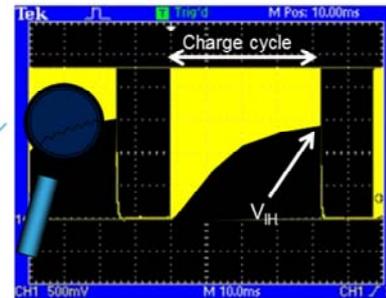


To offer sufficient clock granularity, the TSC is directly clocked using the AHB clock. This clock is used by the spread spectrum block while the clock feeding the pulse generator is reduced using a prescaler. GPIOs supporting touch sensing must be configured in an alternate mode in order to connect them to the Touch Sensing Controller. The SYNC input pin is used to synchronize the capacitive sensing acquisition with an external stimulus without the need for CPU interaction. One counter per analog I/O group is used to store the result of the acquisition. An interrupt can be generated upon the end of acquisition of all the enabled analog I/O groups or when an error is detected. This interrupt helps limit CPU overhead.

# Charge transfer acquisition overview

5

- Charge transfer uses electrical properties of the capacitor (charge  $Q$ )
- The sensor capacitor ( $C_X$ ) is charged to  $V_{DD}$ . Once it is fully charged, a portion of the capacitor's charge is transferred to a sampling capacitor ( $C_S$ ).
- Charge is transferred using analog switches directly embedded into the GPIO
- The charge transfer cycle is repeated  $N$  times until the voltage on the sampling capacitor reaches the  $V_{IH}$  threshold of the GPIO it is connected to
- The number  $N$  of transfer cycles required to reach the threshold represents the size of  $C_X$
- The number of transfer cycles decreases when the sensor is touched.

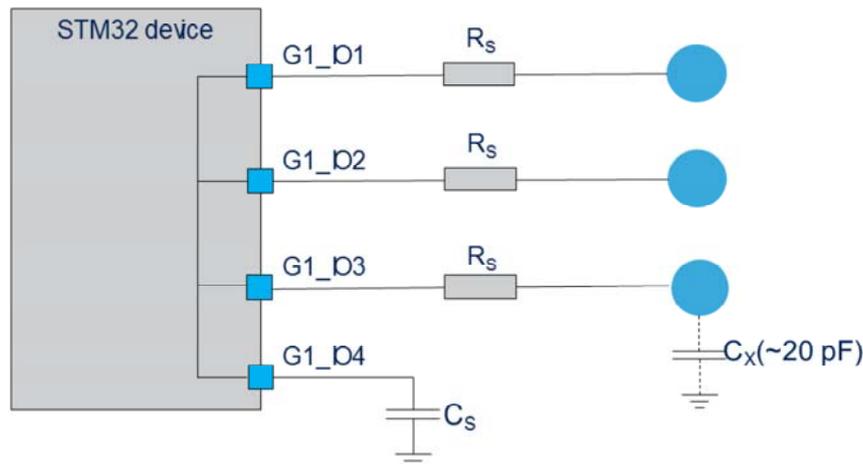


The charge transfer acquisition technique works using the electrical properties of the capacitor. It consists in charging the sensor capacitor ( $C_X$ ) to  $V_{DD}$ . Once this capacitor is fully charged, a part of the accumulated charge is transferred into a sampling capacitor ( $C_S$ ). The number of charges transferred to the sampling capacitor depends on the factor  $C_X/C_S$ . The charge transfer cycle is repeated  $N$  times until the voltage on the sampling capacitor reaches a threshold ( $V_{IH}$  in our case). The number  $N$  represents the size of  $C_X$ . When there is a touch, the sensor capacitor is increased and thus the amount of charge transferred from the sensor capacitor to the sampling capacitor is higher leading to a decrease in the number of charge transfer cycles. The charge transfer is performed through analog switches embedded in the GPIO.

# Charge transfer measuring circuit

6

- $R_S$  (typically 1 k $\Omega$ ) is used to improve ESD robustness
- $C_S$  sampling capacitor value depends on the required channel sensitivity
  - The higher the  $C_S$  value is, the higher the sensitivity, but the longer the acquisition time



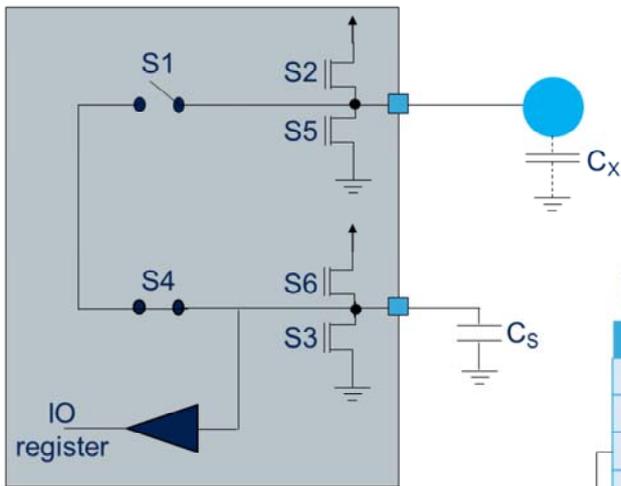
The measuring circuit is simple. Let's consider one analog I/O group composed of four I/Os. One of these I/Os is the sampling capacitor I/O. It is connected to an external capacitor ( $C_S$ ) commonly called the sampling capacitor. There is a single sampling capacitor per analog I/O group. The sampling capacitor value depends on the channel sensitivity. The higher the  $C_S$ , the higher the sensitivity and the longer the acquisition time.

The three other I/Os are dedicated to channels. Each of them is connected to a sensor electrode through a serial resistor ( $R_S$ ).  $R_S$  is used to improve the ESD robustness of the application. Within one analog I/O group, only one channel is acquired at a time. This means that if three channels are implemented, three consecutive acquisitions will be required to get the image of the three sensors. For optimum performance, the sensor capacitance should be as low as possible. We often consider this capacitor value to be in the tens of picofarads. A touch leads to an increase in the

sensor capacitance by a few picofarads, for example 5 picofarads.

# Charge transfer acquisition sequence

7



S4 is closed for the whole acquisition  
S5 & S6 are open for the whole acquisition

Step	S1	S2	S3	Description
1	Closed	Open	Closed	$C_S$ & $C_X$ discharge
2	Open	Open	Open	Dead time
3	Open	Closed	Open	Charge cycle ( $C_X$ charge)
4	Open	Open	Open	Dead time
5	Closed	Open	Open	Transfer cycle (charge transferred to $C_S$ )
6	Open	Open	Open	Dead time
7	Open	Open	Open	Read voltage on $C_S$

Repeat until  $V_{CS}$  is read as a logical '1'



The charge transfer acquisition sequence is composed of 7 steps.

First, the sampling capacitor and the sensor capacitor are discharged to get a stable starting point by closing the analog switch S1 and enabling S3. Between each major step, an intermediate step is inserted to avoid an acquisition artifact. This step, called dead time, consists of opening all active analog switches and disabling all active transistors. Next, the sensor capacitor ( $C_X$ ) is charged to  $V_{DD}$  by closing S2.

After the dead time, a portion of the charge accumulated in  $C_X$  is transferred into the sampling capacitor  $C_S$  by closing the analog switch S1.

Once the charge is transferred, the voltage on  $C_S$  ( $V_{CS}$ ) is read. If the voltage is lower than  $V_{IH}$ , a logical '0' is returned. If it is greater than  $V_{IH}$ , a logical '1' is read. If the returned logical value is '0', Steps 3 to 7 are repeated. After each charge transfer loop, a counter is incremented, representing

the capacitance of the sensor.

- Sampling capacitor I/O is configured as
  - Alternate output open-drain mode with Schmidt trigger hysteresis disabled
- Channel I/O is configured as
  - Alternate output push-pull mode



In order for a Touch Sensing GPIO to be controlled by the TSC:

- A sampling capacitor I/O must be configured in alternate output open-drain mode. In addition, to avoid artifacts, the Schmidt trigger hysteresis must be disabled.
- A channel I/O must be configured in alternate output push-pull mode.

## • Flexible acquisition modes to reduce CPU load

- Normal acquisition mode
  - Once the TSC peripheral is configured, the acquisition is launched by setting the START bit of the TSC\_CR register
- Synchronized acquisition mode
  - Once the TSC peripheral is configured, the acquisition is launched upon the detection of a falling or rising edge and a high level on the SYNC input pin
  - This mode is useful to limit the effect of noise in some applications (induction cooktop, ...)
- In both modes, the end of acquisition and/or max count error can be managed by polling or interrupt



To reduce the CPU load, two acquisition modes are supported:

- Normal acquisition mode where the acquisition starts by setting the START bit of the TSC\_CR register.
- Synchronized acquisition mode where the acquisition only starts upon the detection of a falling or rising edge and a high level on the SYNC input pin. This mode is useful to limit the effect of noise in some applications such as an induction cooktop.

In both modes, the end of acquisition and/or max count error can be managed either by polling or interrupt.

## Flexible and versatile GPIOs

- The touch sensing controller also provides a manual control for both the embedded analog switch and hysteresis of GPIOs belonging to analog I/O groups
  - This could be useful for implementing a different capacitive sensing acquisition principle or for other purposes (i.e. analog multiplexor).



life.augmented

In addition to managing the charge transfer capacitive acquisition principle, the TSC peripheral allows the designer to individually control the analog switch and Schmidt trigger hysteresis of I/Os belonging to analog I/O groups. This capability could be useful to implement a different capacitive sensing acquisition principle or for other purposes such as an analog multiplexer.

Interrupt event	Description
End of acquisition	Set when an acquisition is completed.
Max count error	Set when a max count error occurs. The management of the max count error avoids getting into an infinite acquisition in case of channel issues (C <sub>S</sub> in short-circuit, ...).



The TSC peripheral offers two interrupt sources:

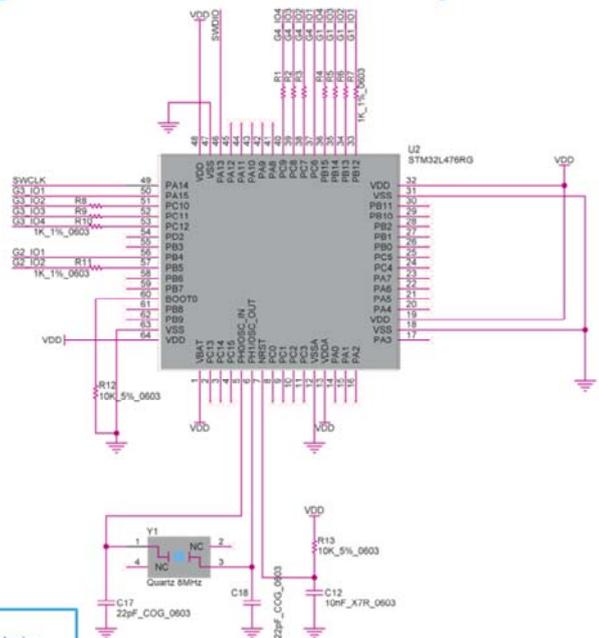
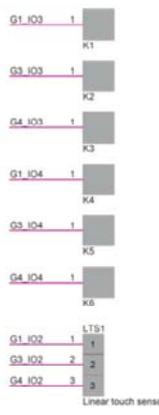
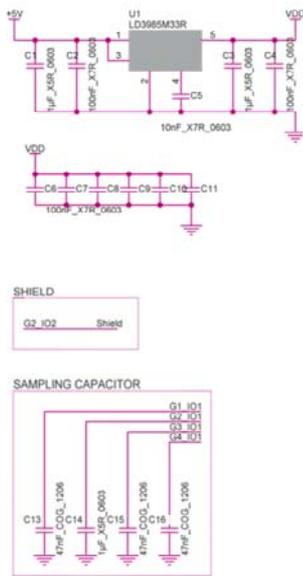
- End of acquisition which notifies the CPU when all the active channels are acquired.
- Max count error which is set when the acquisition fails on one or several channels. It is useful in preventing an infinite acquisition which can occur in the event of a hardware failure.

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Low-power run	Active.
Low-power sleep	Active. Peripheral interrupts cause the device to exit Low-power sleep mode.
Stop 0/Stop 1/Stop 2	Frozen. Peripheral registers content is kept.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby or Shutdown mode.
Shutdown	



The touch sensing controller is active in Run, Sleep, Low-power run and Low-power sleep modes. This means that charge transfer acquisition can only be performed in these modes. In all others modes (Stop 0, Stop 1, Stop 2, Standby and Shutdown), the touch sensing controller is not operational. In Stop modes, the peripheral is frozen but the registers content is kept. In Standby and Shutdown modes, the registers content is lost and the peripheral must be reinitialized.

# Application example



- Notes:
1. ESD Serial resistors and sampling capacitors must be placed as close as possible to MCU device
  2. Sampling capacitors must be COG type or better
  3. A dedicated low drop voltage regulator powering the touch controller is recommended



This example details a solution with 6 touchkeys and one linear touch sensor. It is important to note that a dedicated voltage regulator is used and that the solution also uses the active shield for optimum conducted noise robustness. A COG type capacitor is used for the sampling capacitors as they offer a good stability over the temperature range and no memory effect.

## Application tips and tricks

14

- A voltage regulator is recommended to minimize noise induced by power supply variations
- For optimum sensitivity, parasitic capacitance to ground must be minimized
  - Sensing tracks should be as short as possible
  - Thin sensing tracks are recommended
  - Serial resistors and sampling capacitors should be placed as close as possible to MCU
  - Group sensing tracks by bank (channels acquired simultaneously)
  - Use bypass capacitor to ensure low impedance (i.e. LED drive)
- Use of active shield and spread spectrum is recommended for optimum conducted noise performance
- Do not use conductive painting and ensure a stable mechanical assembly to avoid false or spurious touch detection



life.augmented

A voltage regulator used to power the touch controller is strongly recommended. It will minimize the measurement noise induced by power supply variations.

For optimum sensitivity, the parasitic capacitance to ground must be minimized. This implies short and thin sensing tracks. Serial resistors ( $R_s$ ) and sampling capacitors ( $C_s$ ) must be placed as close as possible to the MCU. The sensing tracks driving sensors which are acquired at the same time should be grouped together (bank) and kept separated from others banks. Finally, a bypass capacitor should be used in case of high impedance drive (i.e. LED driven through an open drain circuitry) to ensure a low impedance path.

For optimum conducted noise performance, we recommend to use an active shield around tracks and sensor pads combined with spread spectrum.

Finally, conductive paint must be avoided and a stable mechanical assembly is required to avoid false or spurious

touch detections.

# STM32Cube Touch sensing library features

15

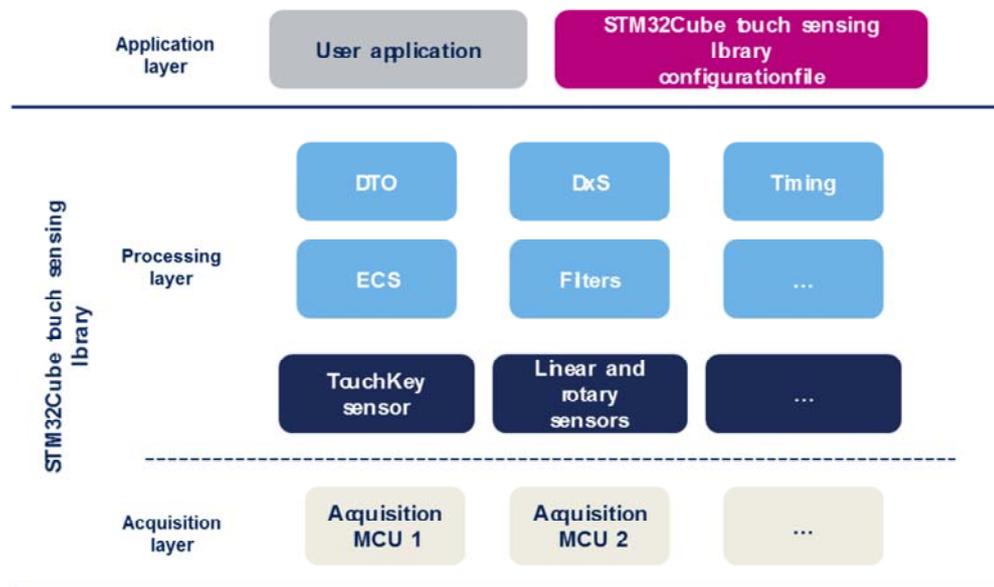
- Complete free C source code library supporting proximity, touchkey, linear and rotary touch sensors
- Multifunction capability combining capacitive sensing functions with traditional MCU features
- Enhanced processing features for optimized sensitivity and immunity
  - Calibration, environment control system (ECS), debounce filtering, detection exclusion system (DxS), ...
- Complete and simple API for status reporting and application configuration
- Compliant with MISRA and with all STM32 C compilers



As previously indicated, the TSC peripheral is designed to operate with the touch sensing library. This free C library supports proximity, touchkey, linear and rotary touch sensors. It allows combining capacitive sensing functions with traditional MCU features such as LCD drive, communication with a host device, ... This library offers all the processing required to get optimum sensitivity and to design a robust application. Some of the features include power-on calibration, environment control system (ECS), debounce filtering and a detection exclusion system (DxS). This library offers a simple API to configure the channels, sensors and to get the state of the sensors. It is MISRA compliant and it supports all STM32 C compilers.

# STM32Cube Touch sensing library overview

16



The STM32Cube touch sensing library is composed of several modules. The library relies on the corresponding STM32 series HAL and it is configured through a dedicated configuration file. Once included into your project, the STM32Cube touch sensing library is part of the overall application and each C function can be launched to get the appropriate behavior.

For further details on the STM32Cube touch sensing library, please refer to the corresponding user manual.

- Refer to these peripheral trainings related to this peripheral:
  - Reset and clock control (TSC clock control and reset)
  - Interrupts (TSC interrupt mapping)
  - General-purpose IOs (TSC channel and sampling capacitor I/O)



This is a list of peripherals related to the STM32 Touch Sensing Controller. Please refer to these peripheral trainings for more information if needed.

- For more details, please refer to following sources
  - AN4299 - Guidelines to improve conducted noise robustness on touch sensing applications
  - AN4310 - Sampling capacitor selection guide for MCU based touch sensing applications
  - AN4312 - Guidelines for designing touch sensing applications with surface sensors
  - AN4316 - Tuning a STMTouch-based application



For more details, please refer to application notes AN4299, AN4310, AN4312 and AN4316.

Thank you.