TN0075
Technical note

Touch sensing software library (TSSL) frequently asked and anticipated questions (FAAQs)

What is the touch sensing software library designed for?

This software transforms any 8-bit STM8 and 32-bit STM32 microcontroller into a capacitive touch keys controller.

This solution enables designers, comfortable with the use of standard microcontrollers, to create higher-end “look and feel” human interfaces by replacing conventional electro-mechanical switches with touch sensing controls.

Designers can combine touch sensing functions using multiple configurations (touch keys, wheels, sliders) with traditional MCU features (communication, LED control, beeper, LCD control, etc.).

The touch sensing software library is part of the application firmware.

Maturity, robustness, flexibility and performance make this solution simple to implement and time to market to develop any kind of applications such as mobile phones, cooking appliances and printers to name only a very few examples.

What is the basic principle used in the TSSL to detect a capacitive human touch?

Human touch is detected by monitoring the charge/discharge timing cycle of a RC network formed by a single resistor and the touch electrode capacitance.

Any variation in the RC timing due to the electrode capacity change is detected then filtered and eventually reported to a host system using dedicated I/Os or I^2C/SPI interface.

The bill of material is low as only one resistor is needed per touch channel to enable this function.

The RC acquisition method is based on a 1976 US Patent from now in the public domain.

What is the working principle of the electrodes?

Copper electrodes emit an electric field through a touch panel typically made of glass or plastic film acting as a dielectric. A finger touching the panel increases the capacitance value of the electrode by a few pF due to the coupling of the human body to the surrounding environment. The increase in capacitance is detected by the touch sensing library.

What are the recommendations for electrode design?

Copper electrode shapes are simple to create with a large efficiency margin for designing electrodes with different shapes, sizes and location. Low-cost, single-sided CEM-1 PCBs can be used, but many other types of substrates are adequate.

What are the recommendations for the panel that protects the PCB electrode?

The most important factors are panel thickness and its “epsilon” value (dielectric constant of the material). Adequate thickness and epsilon values are interdependent. Glass or plastic panels are typically used. For example, a plastic panel up to 10mm thick can be used. Also, a good contact between the panel and the PCB substrate (adhesive or mechanical) is essential for optimized performance. Good examples of PSA (pressure sensitive adhesive) glue are: 3M467 and 3M468.

Does ST provide the library source code to customers?

Yes, the library source code written in C is provided.

Is the touch sensing software library subject to a license agreement?

Yes, this is licensed royalty-free software.

When downloading the library, the customer must agree with the license agreement to install and use the software library. The provided software and related documentation are intended and supplied for use solely and exclusively in a system that incorporates ST microcontrollers (STM8 and STM32).

How many keys/wheels and sliders can the TSSL manage?

This is MCU dependent but, up to 24 single keys and 2 wheels or sliders can be controlled by the touch sensing software library.

What are the MCU hardware resources used by the library?

When using an STM8 platform, the TSSL only needs one 16-bit timer to measure the RC charge and discharge time and one standard 8-bit timer used as a time base for the post processing (calibration,filtering, etc.). One I/O is required for each channel and one I/O for each load output.

What is the memory size used by the library?

The reserved Flash memory footprint depends on the application key configuration:

- Keys only: approximately 2 Kbytes of Flash memory are needed.
- Keys + 2 wheels/sliders: approximately 3.9 Kbytes are needed.

What are the different software layers embedded in the library?

Three main software layers are implemented. Basically,

- The acquisition layer managing the RC measurement
- The post processing layer controlling the calibration, filtering and Environment Change System.
- Configuration, filtering parameters and channel status values are accessible by the user through the API interface.

Is there a specific software algorithm implemented for noise rejection?

The customer does not need to add any additional software layers.

Two digital filters are embedded in the TSSL making this software solution very robust and immune against noise.

- Multi-sampling: the RC charge/discharge acquisition is performed several times per channel to eliminate high frequency noise while giving an accurate measurement.
- The second feature used to help filter low frequency noise is a user programmable de-bounce filter. This digital filter mechanism requires consecutive detections over a number of measurements for a touch to be confirmed.

Is calibration needed during the manufacturing process?

No, the calibration process is already part of the post-processing layer. Keys are auto-calibrated at power-on or upon user request.

This feature adjusts the touch sensing system to the final assembly and environment conditions. So there is no need of final adjustments during the manufacturing process.

Is there a specific feature implemented to compensate the environment change over the application life time?

Yes. To avoid false detection or non-detection caused by variations in power supply, electrode capacitance, temperature, humidity, dirt and other environmental effects, the Environment Change System (ECS) available in the TSSL automatically compensates for signal drift over the time.

Are there low power modes available to reduce consumption?

Yes, the standard microcontroller low power modes remain available for the user.

Thus, by periodically adding some low power mode windows in the application firmware, the user can significantly reduce the average consumption while keeping active the human touch detection.

The average consumption is mainly dependent of the MCU target (I_{DD} run value) and the number of keys, as well as the application firmware profile.

What is the key response time?

Key response time depends on many parameters such as the number of channels, de-bounce filter value, multi-sampling value, user firmware task duration, dielectric thickness, electrode capacitance and R value.

The formula below can be used to estimate the worst key response time:

Maximum response time = [(DF+1) x t_S x NB_channels] + (DF x t_U)

Where:

t_S = approximately 1ms (standard RC acquisition time for 1 channel)

NB_channels = number of channels

DF = de-bounce filter value (user programmable)

t_U = user firmware execution duration

For example in a 4-key application:

DF = 2 (standard value)

t_S = 4 ms

t_U = 1 ms (example/application dependent)

Response time = 14 ms (in this configuration)

What documents and associated materials are available?

Here is the list of the main deliverables:

- Touch sensing software library source code
- Library user guide
- Technical/marketing presentation
- Application notes
- Development board with user manual

What are the main advantages of the TSSL over the most active direct solutions from competitors?

In comparison to standalone or application specific touch sensing ICs that competitors may propose, the open source touch sensing software solution combined with a well-known microcontroller architecture, offers better flexibility and robustness at the application level.

Customers are familiar with microcontrollers where they can simply configure, adapt or accommodate the library in regards to application needs.

At system level, maximum performance is achieved by controlling the sensitivity, robustness and noise immunity through specific embedded software functions (noise reduction, filtering) where parameters are user configurable.

The one key advantage, versus competitors offering similar software solutions, is the know-how embedded in this library to calibrate and compensate the signal drift from the keys caused by the environmental changes. Thus, in the manufacturing process or during the application lifetime, this “safe guard” as well as the state-of-the-art software algorithm ensure correct touch sensing functions over time.

Revision history

Table 1. Document revision history

Date	Revision	Changes
02-Feb-2009	1	Initial release.