



Freescale Enablement Solutions

Touch Sensing Software Suite

Enabling natural human-machine interface

Overview

To reduce time to market and improve application development success, Freescale Semiconductor is offering the Touch Sensing Software (TSS) Suite to enable any 8-bit S08 MCU as a touch sensor, providing cost-effective and flexible solutions for human machine interface. Tactile is an example of a natural human interface. Touch inputs are used on a wide range of applications, including mobile phone handsets, personal media players, audio/video systems, multimedia Internet devices, home appliances, home security, PCs and peripherals, medical devices, automotive and industrial equipment.

The TSS Suite simplifies user interface designs, enabling customers to develop an application within minutes using the included development board kit and demonstration software. The suite provides the means to develop user interfaces and avoid mechanical wear and tear, increase the life time of products, and give greater flexibility to designers to select from more than 300 MCU options in Freescale's 8-bit portfolio.

TSS is based on two main conversion algorithms for the capacitive measurement, advance touch logic (ATL) and capacitive turbo sensing (CTS). These algorithms are part of Freescale's extensive IP portfolio and patents. These modules provide the low-level conversion from analog to digital values using a simple I/O pin. This digitalized value is then

processed to avoid false detections or noise influence from the detection and calibration modules. Finally, the decoding layer enables the key detected to be included in a decoder control that treats the detection as a part of keypad, slider or rotary interface.

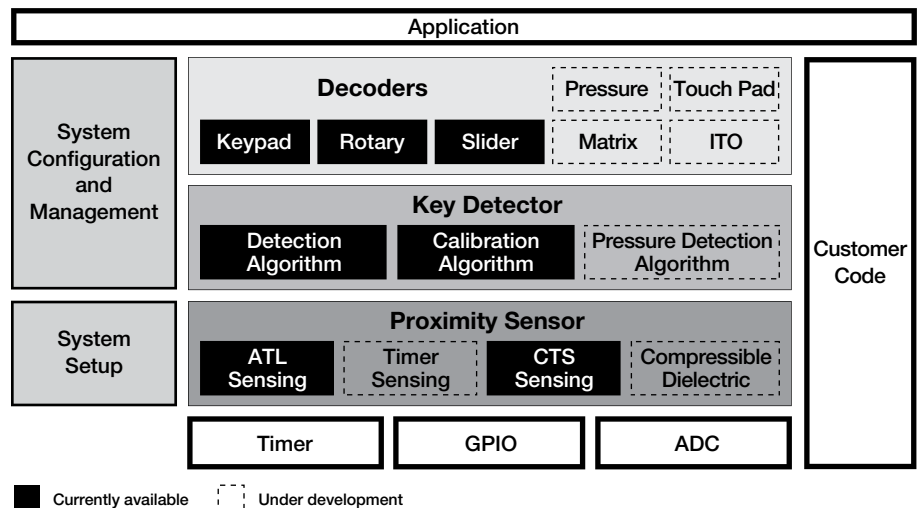
Advanced touch logic (ATL) is a software algorithm which use timer module and is based on common RC charging principle. Capacitive turbo sensing (CTS) is an IP protected solution that only requires the I/O pin and provides fast charging measurement technology in μ s range. Conversion time is guaranteed while the resolution is not reduced across typical hardware conditions.

TSS Design and Development Tools

The TSS library supports different kinds of hardware designs, PCB, capacitive film, flexpcb and more. For guidance on best practices for electrode and layout design, download the application note, *Designing Proximity Sensing Electrodes* (document AN3863) from www.freescale.com/touchsensing.

Full set of development tools includes touch sensing software evaluation board (TSSEVB), electrode graphing tool (EGT), library configuration GUI, demo applications and documentation.

Touch Sensing Software Suite Architecture

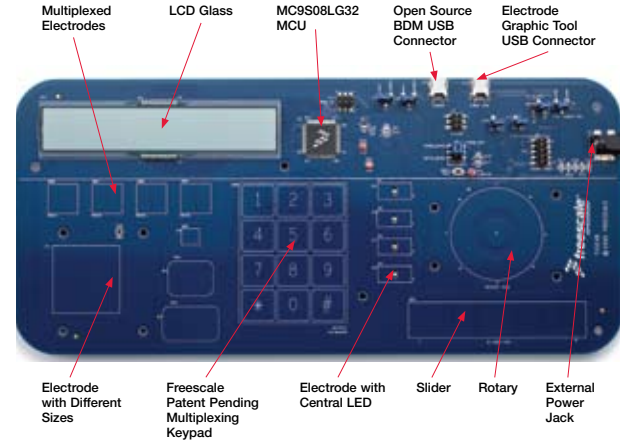


Touch Sensing Software Suite Free License Includes

- Full API set support
- Configurable rotary, slider and keypad decoders
- Smart auto-calibration mechanisms to prevent environmental hassles
- Noise rejection algorithms
- Optimized buffer structure enabling any arrangement of electrodes
- Ability to enable and disable keys on runtime
- Auto repeat, stuck-up key, gorilla hand and other typical HMI function capabilities
- Ability to co-exist with other application code
- PC GUI application for electrode characterization
- Support for Freescale S08 MCU families
- Demos and application examples
- CodeWarrior™ 6.3 compatibility

TSSEVB Key Features

- Freescale patented Multiplexed Keyboard (USPTO approval pending)
- Quick evaluation of rotary, slider, keypad and different sized electrodes
- Electrode with central back-mounted LED
- LCD glass, LEDs and buzzer
- Integrated S08LG32 MCU for library evaluation
- Serial to USB bridge for real-time software tuning using custom GUI
- Integrated open source BDM
- Power over USB
- Prepared for different dielectric types of testing



Memory Usage			
TSS SW Modules	Description	RAM	Flash
<ul style="list-style-type: none"> • System Configuration and Management • System Setup • Key Detector • ATL Capacitive Sensing • Timer • GPIO 	<ul style="list-style-type: none"> • Minimum memory usage of the TSS when no decoder is used and using the ATL Sensing method. 	105 bytes	3723 bytes
	<ul style="list-style-type: none"> • Memory usage of the TSS per electrode used in the application and using ATL Sensing method. 	12 bytes	41 bytes
<ul style="list-style-type: none"> • System Configuration and Management • System Setup • Key Detector • CTS Capacitive Sensing • Timer • GPIO 	<ul style="list-style-type: none"> • Minimum memory usage of the TSS when no decoder is used and using the CTS Sensing method. 	104 bytes	3905 bytes
	<ul style="list-style-type: none"> • Memory usage of the TSS per electrode used in the application and using CTS Sensing method. 	13 bytes	118 bytes
<ul style="list-style-type: none"> • Keypad Decoder 	<ul style="list-style-type: none"> • Memory usage for one Keypad Decoder on TSS. (Excludes the memory footprint of the library and the electrodes assigned to this decoder) 	29 bytes	1158 bytes
	<ul style="list-style-type: none"> • Memory usage per additional Keypad Decoder. This value must be added for each additional keypad decoder used in the application. 	29 bytes	42 bytes
<ul style="list-style-type: none"> • Slider Decoder* 	<ul style="list-style-type: none"> • Memory usage for one Slider Decoder on TSS. (Excludes the memory footprint of the library and the electrodes assigned to this decoder) 	9 bytes	1008 bytes
	<ul style="list-style-type: none"> • Memory usage per additional Slider Decoder. This value must be added for each additional slider decoder used in the application.. 	9 bytes	26 bytes
<ul style="list-style-type: none"> • Rotary Decoder* 	<ul style="list-style-type: none"> • Memory usage for one Rotary Decoder on TSS. (Excludes the memory footprint of the library and the electrodes assigned to this decoder) 	9 bytes	1189 bytes
	<ul style="list-style-type: none"> • Memory usage per additional Rotary Decoder. This value must be added for each additional rotary decoder used in the application. 	9 bytes	19 bytes

*NOTE: The Slider Decoder and Rotary Decoder share common modules within the TSS. Therefore, when you are using both of them into an application, you must only add the memory usage of one Rotary Decoder and the memory usage for an additional Slider.

Low Level Layer	Peripheral Usage
ATL	<ul style="list-style-type: none"> • One TPM or MTIM module • One GPIO per electrode
CTS	<ul style="list-style-type: none"> • One GPIO per electrode

Learn more: For information about our Touch Sensing Software Suite, visit www.freescale.com/touchsensing.