Xtrinsic Touch Sensing Software 2.0
Add touch. With flexible libraries.
Eduardo Viramontes
Agenda

- Freescale Touch Sensing Overview
- Capacitive touch sensing
- TSS features
- Touch sensing enablement
- Touch sensing hardware considerations
- Demo TSSEVB
- Questions
Freescale Touch Sensing Overview
Opaque Touch Panels Market

Opaque touch-sensing panels
- One of the fastest-growing markets for MCUs
- Capacitive sensing products are rapidly penetrating the overall touch sensing technology market
- MCU based market will grow from $.5 in 2011 to $1.5B in 2015
- Wide range of markets and applications, including mobile phone handsets, consumer electronics (such as personal media players and audio/video systems), home appliances, home security, PC peripherals, medical, automotive, and industrial equipment

Growth Drivers:
- Portable Devices Growth
- Low-cost Capacitive Film Material
- Penetration of touch sensing on common life
- User Interface Evolution - IPOD effect

Solution Example
MCU + TSS

HMI within these IMM target segments is rapidly migrating to touch sensing
# Xtrinsic Touch Sensing Software 2.0: Target Markets and Applications

## Target Markets
- Touch screens
- Front panels
- Medical devices interfaces
- Multi-media boards
- Front panels
- Instrumentation panels
- Building control panels
- HVAC controls
- Central stacks
- Factory Automation
- Medical

## Applications
- Touch screens
- Keyboards
- Touchpads
- Mouse/peripherals
- Gaming controls
- Central stacks
- Lighting controls
- Steering wheel sensors
- Driver info and entertainment
- Front panels
- Medical device interfaces
- Standard & Multi-Market Automotive
- Appliances
- Handhelds
<table>
<thead>
<tr>
<th>Capacitive Touch Sensing</th>
<th>MPR031</th>
<th>MPR083</th>
<th>MPR084</th>
<th>MPR121</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone Controllers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capacitive Touch Sensing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software added to MCUs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S08 MCUs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More than 650 Microcontrollers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit MCUs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ColdFire V1s, ColdFire + and Kinetis MCUs Families</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resistive Touch Screen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controllers on MPUs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ColdFire MCF5227x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i.MX233</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i.MX251</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i.MX255</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i.MX257</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i.MX258</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Freescale Capacitive Touch Solutions Roadmap

- **Touch Sensing Software Suite 1.0**
  - 1.8 – 5.5V, 64 electrodes, keypad, slider, rotary, matrix & touchpad, I²C, LIN, CAN, SPI, USB, and all S08 package options

- **Touch Sensing Software Suite 2.0**
  - ColdFire V1 support
  - IIR filter, KBI, IC Algorithms
  - Processor Expert Component

- **Touch Sensing Software Suite 2.5**
  - Codewarrior 10.0
  - TSI module for ColdFire+ and Kinetis support

---

- **2008**
  - MPR321 Multi touch
  - MPR123 Industrial
  - MPR122 GPIO output

- **2009**
  - Touch Sensing Software Rev. 3.0

- **2010**
  - Touch Sensing Software Rev. 4.0

- **2011**
  - Touch Sensing Software Rev. 2.0

- **2012**
  - Touch Sensing Software Rev. 1.0

---

- **Portfolio**
  - Concept Proposal
  - Under Design
  - In Production

---

*Not recommended for new designs.*
Capacitive touch sensing
Electrode acts like a capacitance from 10pF to 500pF (typical)

Measure the charge time using GPIOs, 1 Timer and 1 external pull-up
Proximity Sensing Technology

• The capacitive sensing technology works with parasitic capacitance
How to sense capacitance – Basic Algorithm

1. Start
2. Discharge Electrode (Configure the Pin as output)
3. Counter stop And reset
4. Configure pin as input
5. Counter start
6. Electrode Charged?
   - No
   - Yes
8. Timeout Event?
   - No
   - Yes
9. Counter stop
10. Store counter value
11. Return Status
12. End
KBI or PTI Interrupt method

1. Start
2. Discharge Electrode (Configure the Pin as output)
3. Counter stop And reset
4. Enable KBI
5. Configure pin as input
6. Counter start
7. MCU enter WAIT
8. 1
9. 1
10. Counter start
11. Stop Counter
12. KBI Wake Interrupt source
13. Time Out
14. KBI Interrupt
15. Time Out Interrupt
16. End
1. In the start of the capacitor charge uses the internal pull-up to accelerate the charge. So reduces the total conversion time

2. Near the threshold crossing disable the pull-up to have better accuracy

3. Uses the Bus clock to count the charge time. Does not need to use a timer.
Timer/CPU bus Frequency and Pull-Up Configuration

✓ Step-up response: \( V_c = V^* (1 - e^{-t/RC}) \)

✓ Bigger Pull-Up Resistor => greater sensitivity => less noise robustness
✓ Smaller Pull-Up Resistor => less sensitivity => more noise robustness

#define ATLSENSOR_PRESCALER 1

![Graph of Capacitance vs. Sample Time](image1)

![Graph of Sensitivity vs. Bus/Timer Freq](image2)

• Sensitivity resolution down to 33fF/count
Baseline Tracking and Debouncing

Baseline: instantaneous value filtered with a low pass filter

Delta: Instantaneous Value – Baseline

DEBOUNCE(If (Delta > Sensitivity)) => Touch Detected
TSS Features
Touch Sensing Software Suite Key Benefits

► Touch added to one chip solution to reduce the overall system cost and size
► Enables over 650 Freescale 8-bit S08 and 32-bit ColdFire V1 MCUs as a touch sensors, allowing scalability among Freescale MCU portfolio.
► Free license software with low cost development tools
► Enhances reliability by eliminating mechanical wear and tear
► Gives flexibility to designers by allowing any electrode array combination and configuration using only GPIOs.
► Easy to Use – TSS 2.0 simplifies user interface designs, enabling customers to develop an application within 10 minutes from scratch.
► Reduce time to market and development costs
► Increase lifetime of products
Xtrinsic Touch Sensing Software 2.0 Key Features

► Full API set support
► Support up to 64 electrodes
► No need of extra IC
► Configurable rotary, slider and keypad decoders with optimized buffer structure enabling any arrangement of electrodes.
► Smart auto-calibration mechanisms to prevent environmental hassles
► Noise rejection algorithms, new IIR filtering
► New KBI and IC functions for TSS ATL method
► 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 multiple communication protocols: I2C, LIN, CAN, SPI, USB and more
► Possibility to have other peripherals working at the same time: LCD, LEDs, buzzer, ADC, other sensors, and more.
New Capacitance Measurement Methods – Increased Resolution

The TSS 2.0 now offers several kinds of measurement methods and each of them has different sensitivity.

**Timer Input Capture Method**
- Sensitivity Resolution = 1 Bus/Timer cycle
- limited number of input capture pins

**KBI/PTI Method**
- Sensitivity Resolution = 1 Bus/Timer cycle
- limited number of input capture pins

**GPIO Method**
- Sensitivity Resolution = 5 Bus/Timer cycle
- limited number of input capture pins

**CTS**
- Sensitivity Resolution = 9 Bus/Timer cycle

New Methods
5x better resolution!
Capacitance Sensing Methodology

**Enhanced sensitivity and noise robustness:**

- Library scans every electrode several times integrating the total charging time.
- Trade-off: sample time vs. sensitivity/robustness.

![Diagram showing the charging process and sample time](image)

**Library Parameter:** **Nsamples:** Determines the number of samples to be scanned for a single measurement.
Capacitance Sensing Methodology / Interrupt

► Automatically detect interrupts during the measurement.
► A new measure is taken to replace the compromised one.
► Require application proper design.

Library Parameter: **Nsamples**: Determines the number of samples to be scanned for a single measurement.
TSS CPU Processing Time: TSS_Task()

- TSS_Task() start the scan of all electrodes. Giving a capacitance point.
- TSS_Task() should be called periodically by the application. Usual times are from 5ms to 200ms.
- The free time between TSS_Task() execution is available to the application.
TSS CPU Processing Time: TSS_TaskSeq()

- TSS_TaskSeq() start the scan of one electrode. Giving a capacitance point.
- TSS_TaskSeq() should be called periodically by the application. It should be called the number of active electrodes faster to provide the same response time as TSS_Task()
- The free time between TSS_TaskSeq() execution is available to the application.
- Depending of the application timing requirement may be easier to do time scheduling
Outstanding Noise Filtering

Two Powerful New Filters Available:
• IIR Filter
• Amplitude Noise Filter

Noisy Original Signal

Same signal with Filters On
Touch sensing enablement
Tools to Make Development Easy and Friendly

**Library Configuration GUIs**
- Processor Expert Bean
- Stand Alone GUI

**Real Time Visualization of the Electrode Capacitance**
- Allows you to easily tune and debug your application

Freescale, the Freescale logo, AliVcc, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, M8C, Platform in a Package, Processor Expert, CoreIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Complete and Friendly Documentation Set

► TSSEVB – Full evaluation and development (Resale Price: ~$99)
  • Four demo applications included
  • Rotary, slider, keypad, matrix, multiplexed, and back lighted
  • LCD glass and buzzer
  • Serial to USB converter and on board BDM
  • Powered over USB or external power supply
  • Different thickness overlays

► Any board for S08s and ColdFire V1s Flexis families combined with the TSSELECTRODEEVM

► Tower Sensor Pack (TWR-SENSOR-PAK)
Learn Quicker, Develop Faster – right out of the box
Freescale has a full set of development tools and technical documentation to help you speed time to market:

► Technical Documentation
  • Library
    ▪ TSS API Reference Manual
    ▪ TSS User Guide
    ▪ Touch Sensing Electrode Graphing Tool Users Guide
    ▪ TSS Fact Sheet
  • TSSEVB
    ▪ TSSEVB Users Guide
    ▪ TSSEVB Quick Start Guide
  • Demonstration applications
    ▪ Source code
    ▪ Demo application guide

► Application Notes
  • Designing Proximity Sensing Electrodes (AN3863)
  • How to implement a HMI using FSL touch-sensing suite (AN3934)

► Touch GUIs
  • Compilation time configuration GUI
  • Execution time electrode tuning (EGT) GUI
Touch Sensing Software Development Benefits

► Flexibility
  • Allows any electrode array combination on the same application
  • Allows configuration of any electrode GPIOs
  • Smart Software Architecture enables application compatibility
  • Allows reusability and scalability among Freescale MCU portfolios

► CPU Usage
  • Limits the execution time to avoid interfering with other running tasks

► Peripherals Usage
  • Assigns a minimum of peripherals needed for capacitive sensing
  • Allows the user to implement any desired feedback

► Scheduling
  • Gives control of CPU timing and allocation to user

► Software Quality
  • Uses third party for unit and system testing
  • MISRA-C and PC-Lint
TSS Layered Architecture

System Configuration & Management

Decoders
- Keypad
- Rotary
- Slider
- Dec X

Key Detector
- Detection Algorithm
- Calibration Algorithm

Capacitive Sensing
- Timer
- GPIO
- KBI
## TSS Resource usage

<table>
<thead>
<tr>
<th>TSS SW Module</th>
<th>Description</th>
<th>RAM</th>
<th>Flash</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimum memory usage of the TSS when no decoder is used and using the ATL Sensing method.</td>
<td>104 bytes</td>
<td>3905 bytes</td>
</tr>
<tr>
<td></td>
<td>Memory usage of the TSS per electrode used in the application and using ATL Sensing method.</td>
<td>12 bytes</td>
<td>41 bytes</td>
</tr>
<tr>
<td></td>
<td>Minimum memory usage of the TSS when no decoder is used and using the CTS Sensing method.</td>
<td>105 bytes</td>
<td>3723 bytes</td>
</tr>
<tr>
<td></td>
<td>Memory usage of the TSS per electrode used in the application and using CTS Sensing method.</td>
<td>13 bytes</td>
<td>118 bytes</td>
</tr>
<tr>
<td>Keypad Decoder</td>
<td>Memory usage for one Keypad Decoder on TSS. (Excludes the memory footprint of the library and the electrodes assigned to this decoder)</td>
<td>29 bytes</td>
<td>1158 bytes</td>
</tr>
<tr>
<td></td>
<td>Memory usage per additional Keypad Decoder. This value must be added for each additional keypad decoder used in the application.</td>
<td>29 bytes</td>
<td>42 bytes</td>
</tr>
<tr>
<td>Slider Decoder</td>
<td>Memory usage for one Slider Decoder on TSS. (Excludes the memory footprint of the library and the electrodes assigned to this decoder)</td>
<td>9 bytes</td>
<td>1008 bytes</td>
</tr>
<tr>
<td></td>
<td>Memory usage per additional Slider Decoder. This value must be added for each additional slider decoder used in the application.</td>
<td>9 bytes</td>
<td>26 bytes</td>
</tr>
<tr>
<td>Rotary Decoder</td>
<td>Memory usage for one Rotary Decoder on TSS. (Excludes the memory footprint of the library and the electrodes assigned to this decoder)</td>
<td>9 bytes</td>
<td>1189 bytes</td>
</tr>
<tr>
<td></td>
<td>Memory usage per additional Rotary Decoder. This value must be added for each additional rotary decoder used in the application.</td>
<td>9 bytes</td>
<td>19 bytes</td>
</tr>
</tbody>
</table>

*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.*
Touch sensing hardware considerations
Trace recommendations for the layout design of a touch sense solution.

► Trace Width — Keep traces width as thin as possible. 5-7 mil traces are recommended. The wider the more base capacitance.

► Clearance — Leave a minimum clearance of 10 mils. At the trace connection to the MCU the pitch is lower than 10 mils, use bottleneck mode. (see picture)

► Keep traces length as short as possible. As traces becomes longer the baseline capacitance will increase and also more susceptible to couple noise.

► Routing layers:

► Electrode traces should be routed in a layer different to the one containing the electrodes.

► Components and traces should not be placed right underneath the electrodes area. (Good results can still be obtained if the number of components behind the electrodes is minimized and limited to passive devices and running as few traces as possible)
Electrodes Pattern Design

► The shape and size of the electrode must be as similar as possible to a human finger or as the object the electrode will detect.

► Larger electrodes easily detect the presence and absence of a finger. However, in some cases, they can be more sensitive to electrical noise or undesirable surrounding objects.

► Always use an external layer to place the electrodes (Either top or bottom).
When the number of keys exceed the number of sensor channels available, multiplexing or logical combinations can be created.

The detection of a valid key is not directly the change in capacitance of a single electrode but of two or more electrodes at the same time.

When multiplexing a key, the contact area for each electrode is reduced at least to half.
Basic Slider pattern.

Slider pattern for micro-stepping implementation.
This simple topology also includes a center button, optional for some applications.
The arrow keyboard typically consists of five keys; a central key surrounded by an arrow key at each side.

The Keyboard multiplexed consists in arranging the capacitive electrodes on a keyboard so that each key is detected with a combination of two capacitive electrodes.
### Continuous Ground plane vs. X-hatch

<table>
<thead>
<tr>
<th>Continuous planes</th>
<th>X-hatch pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better EMI performance.</td>
<td>Not recommended for applications that radiate High energy (i.e. LED Backlights, Motor control)</td>
</tr>
<tr>
<td>Filters out noise almost any noise cached at the back side of the electrodes.</td>
<td>Reduces baseline capacitance significatively</td>
</tr>
<tr>
<td>Improves S/N ratio.</td>
<td>Improves touch signal detection</td>
</tr>
<tr>
<td>Adds larger baseline capacitance</td>
<td></td>
</tr>
<tr>
<td>Not recommended for small electrodes</td>
<td></td>
</tr>
</tbody>
</table>

---

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.
Electrode Shielding

- Ground behind the electrodes increase noise robustness but also reduces sensitivity. No ground plane, X-Hatch or full ground plane can be used dependable of the application requirements and EMC environment.

- Use a full ground plane for all board with the exception of the electrodes (see consideration above)

- Do not place any component or route any trace underneath the electrode's area.

- The Electrodes ideally should be in a reserved specific PBC area for them.
EMI Protection: Series registers

Avoid coupled noise in the electrodes to be injected in the MCU
Reduce sensitivity. But can improve signal to noise ratio.

Notes: Protections diodes cause a severe sensitivity drop
Mechanical Assembly

► It is important for the design to have the PCB with electrodes have a uniform contact with the overlay.
► Any discrepancies from unit to unit will provoke undesirable effects on the Touch Sensing Software.
Demo - TSSEVB
References:

Freescale.com/touchsensing

Freescale.com/support

twitter@freescale