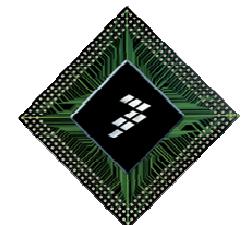


Convert your 8-bit MCU into a smart Touch Sensor

Microcontroller Solutions Group

Gabriel Sanchez
Applications Engineer

Mauricio Gomez
Marketing Manager



October, 2009

► Overview

- Touch Sensing Overview

► Proximity Detection Method

- How Proximity Detection Works

► Electrode Design

- Electrode Shapes and Sizes
- PCB Guidelines

► Application Use

- Software API
- Software Initialization



Touch Sensing – What and Why?

Touch sensing technology depends on capacitance sensing

► **What is capacitance sensing?**

- Capacitance sensing is a **technology that enables detection of touch by measuring capacitance**
- Capacitance sensors respond to a change in surrounding materials
- Used in many areas of consumer, industrial and automotive applications
- One of the most popular applications today is **touch sensing**

► **Why is touch sensing attractive?**

- Enables the elimination of mechanical buttons and sliders
- Enhances reliability by eliminating mechanical wear and tear
- Gives greater flexibility for product designers
- Reduces overall system costs
- **Systems are generally more intuitive for user**

Touch Sensing Market Overview

Touch sensing

(CAGR 2007 – 2011: 23.8%*)

- ▶ One of the fastest-growing markets for MCUs
- ▶ Capacitive sensing products are rapidly penetrating the overall touch sensing technology market
- ▶ Expected growth is over \$3 billion in 2011*
- ▶ Touch sensing in mobile phones alone is growing to 400 million units by 2011*

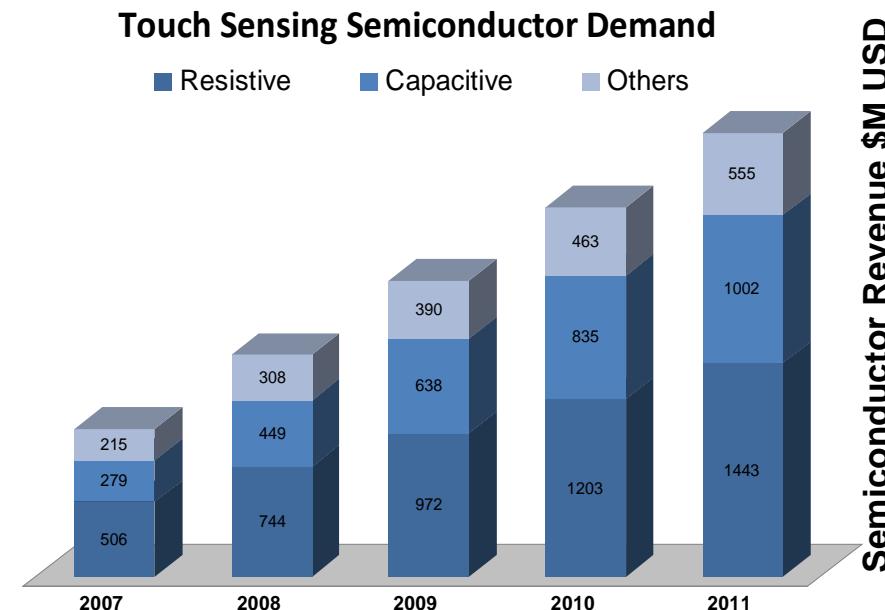
TAM: **\$10.7B**



Growth Drivers:

- Portable devices
- Low-cost touch screen/panel material
- Manufacturability
- User interface evolution - IPOD effect

Touch Sensing Semiconductor Demand



*Source: Strategy Analytics and iSuppli

Freescale TSS Target Markets and Applications

Touch-sensing Software Suite is a set licensed software libraries that enables any Freescale S08 MCU GPIO as capacitive touch sensor

Touch screens
Front panels
Medical devices interfaces



Medical

Instrumentation panels
Building control panels
HVAC controls



*Factory
Automation*

Multi-media boards
Front panels
Touch screens



Appliances

HVAC panels
Central stacks
Lighting controls
Steering wheel sensors
Driver info and entertainment



Automotive

Touch screens
Front panels
Medical device interfaces



Handhelds

Keyboards
Touch screens
Touchpads
Mouse/peripherals
Gaming controls



*Standard &
Multi-Market*

Freescale Touch Sensing Solutions Roadmap

Touch Sensing Software Suite

1.8 – 5.5V,
64 electrodes Keypad,
slider, rotary, matrix &
touch-pads, , I²C, LIN,
CAN , SPI, USB, and
S08 package options

MPR121

1.71 – 3.6V, 12
electrodes keypad, I²C,
3x3 mm 20QFN
Package

MPR031

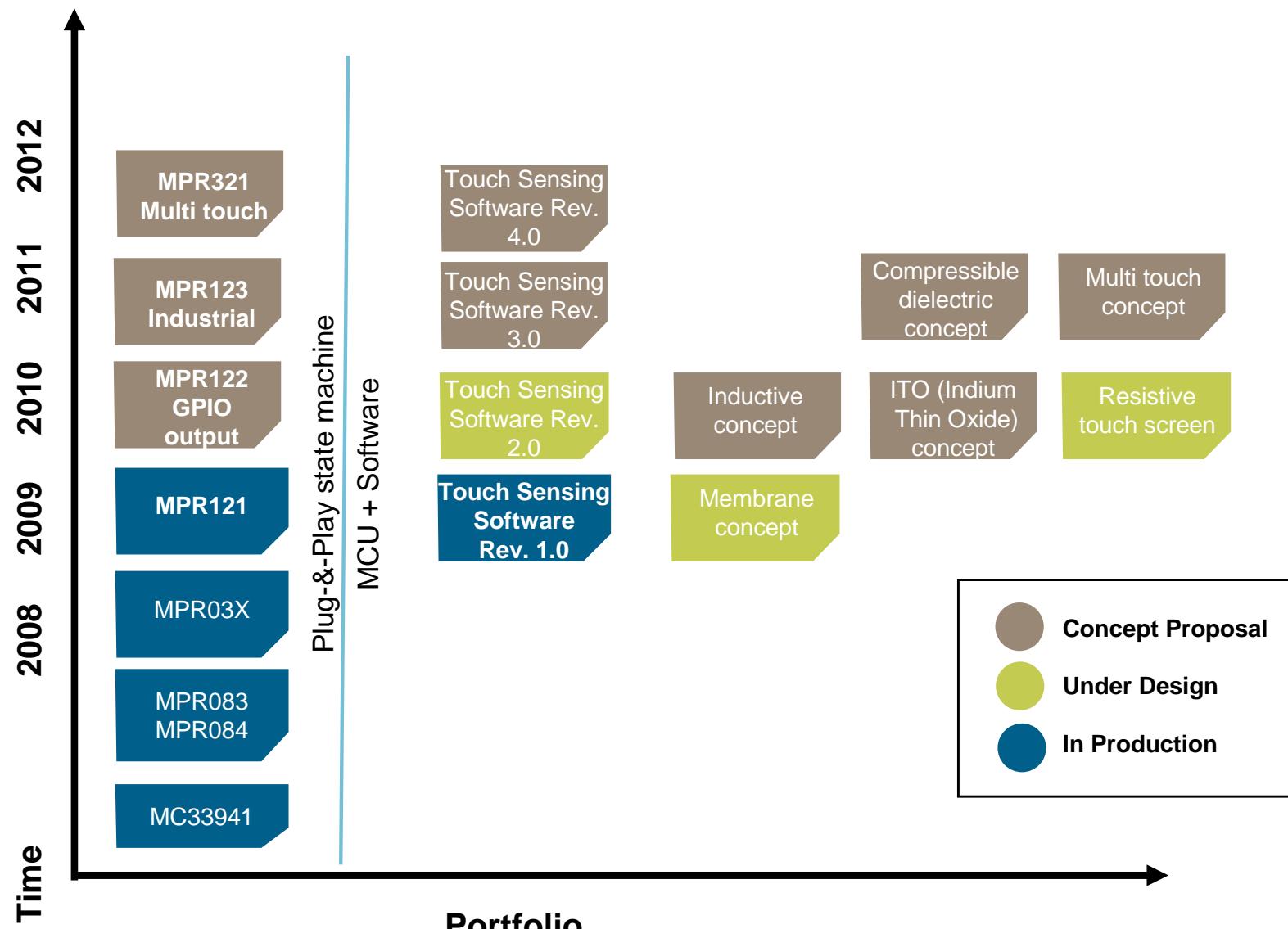
1.71 – 2.75V, 2 or 3
electrodes, keypad, I²C,
2x2 mm 8UDFN
Package

MPR083/084

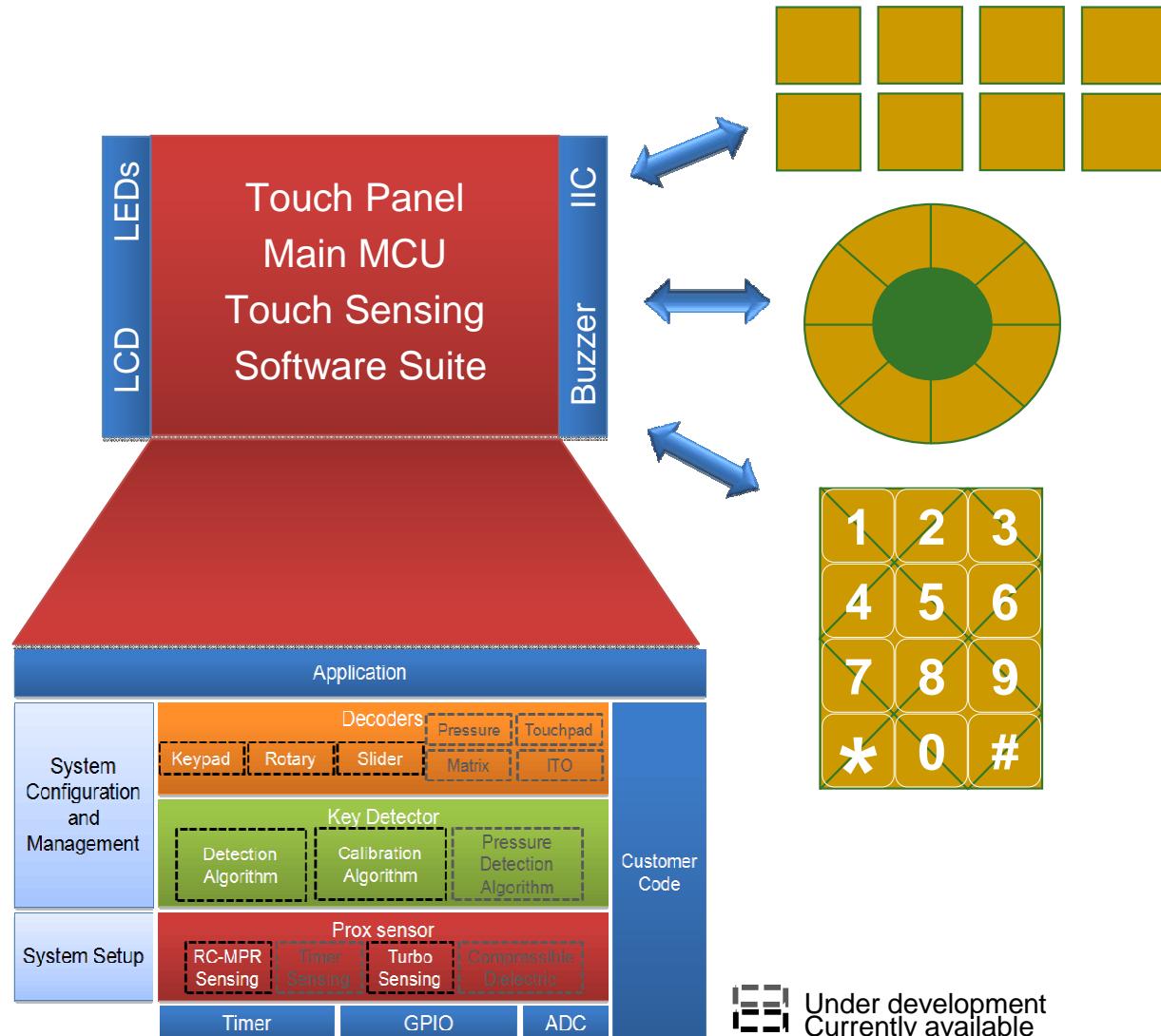
1.8 - 3.6V, 8 electrodes
rotary and keypad, I²C,
5x5 mm 16QFN

MC33941

Electric-field products



Touch Sensing Solution

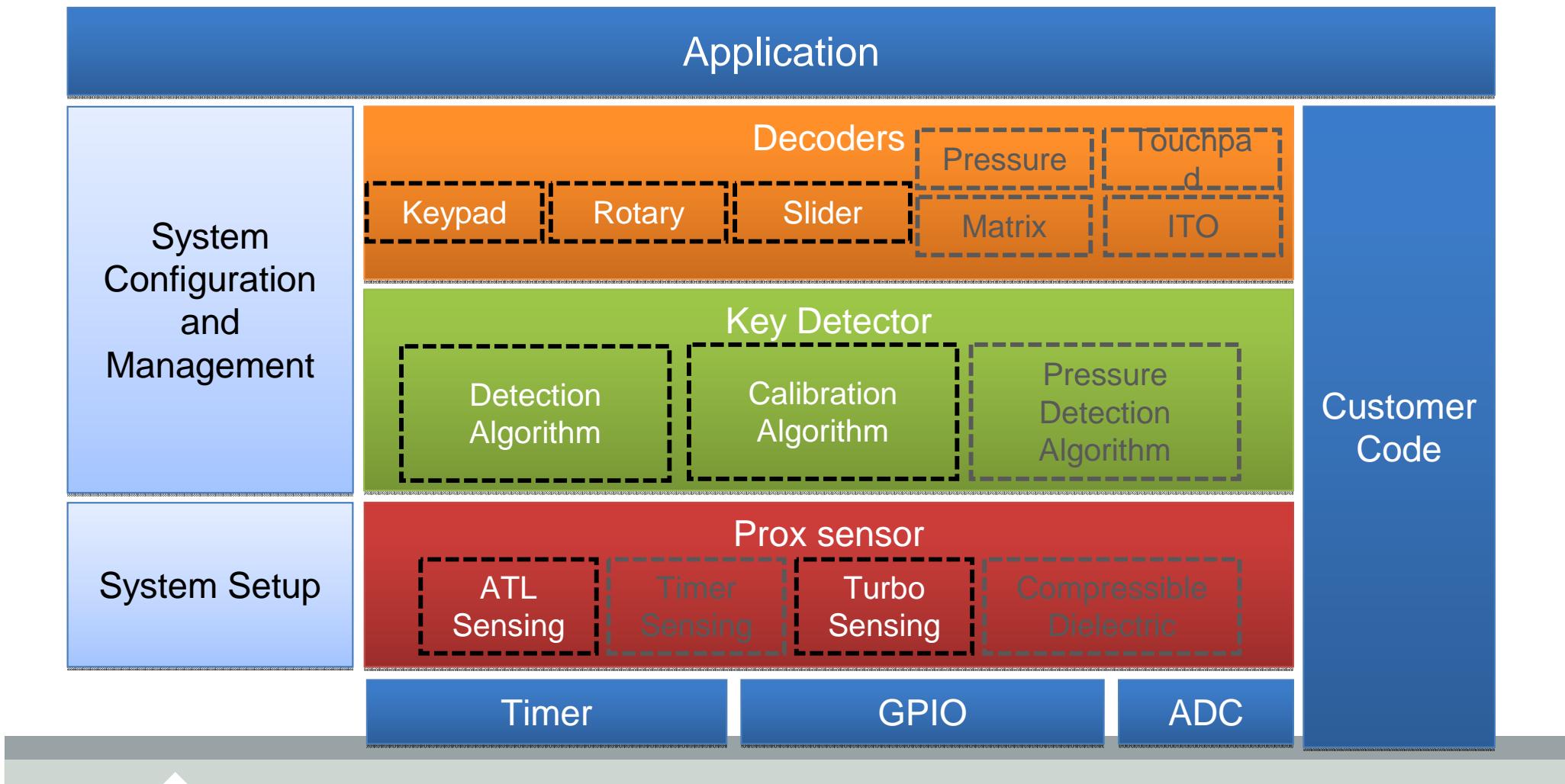


Features:

- ▶ Configurable Rotary, Slider, Keypad decoders
- ▶ Full API set support
- ▶ Smart auto-calibration mechanisms for environment hassles
- ▶ Noise rejection algorithms
- ▶ Optimized buffer structure enabling any arrange of electrodes
- ▶ Ability to enable and disable keys on runtime
- ▶ Auto repeat, stuck-up key, gorilla hand and other typical HMI functions .
- ▶ Capability to coexist with other application code.

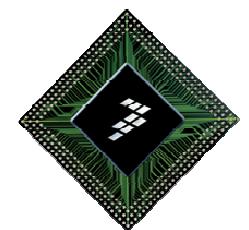
Touch Sensing Software Suite Architecture

- Under development
- Currently available



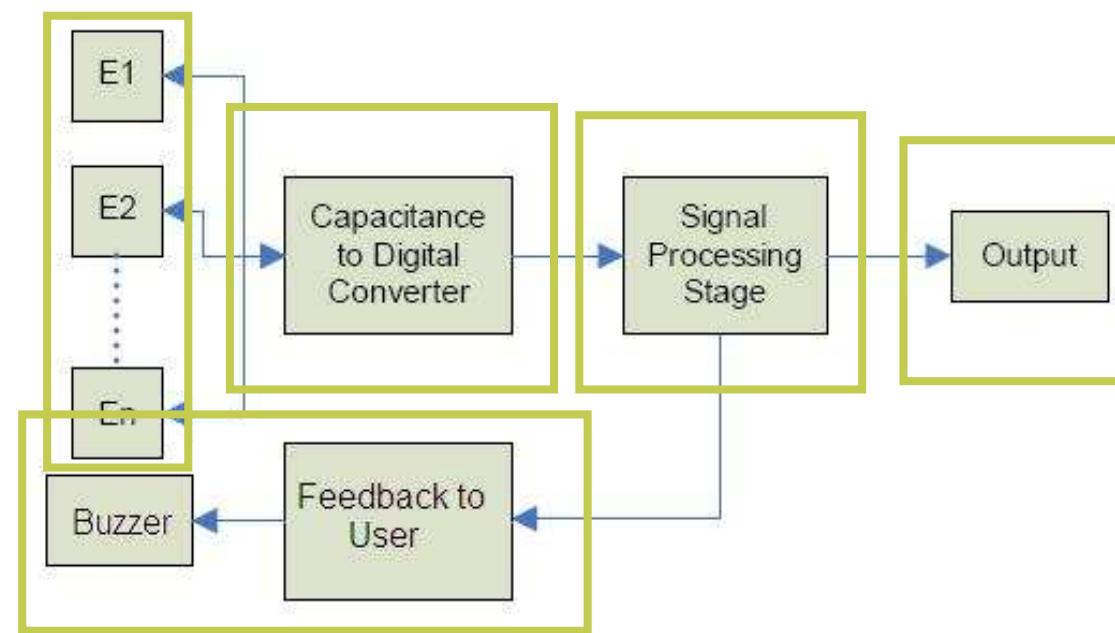


How does Touch Sensing technology work?



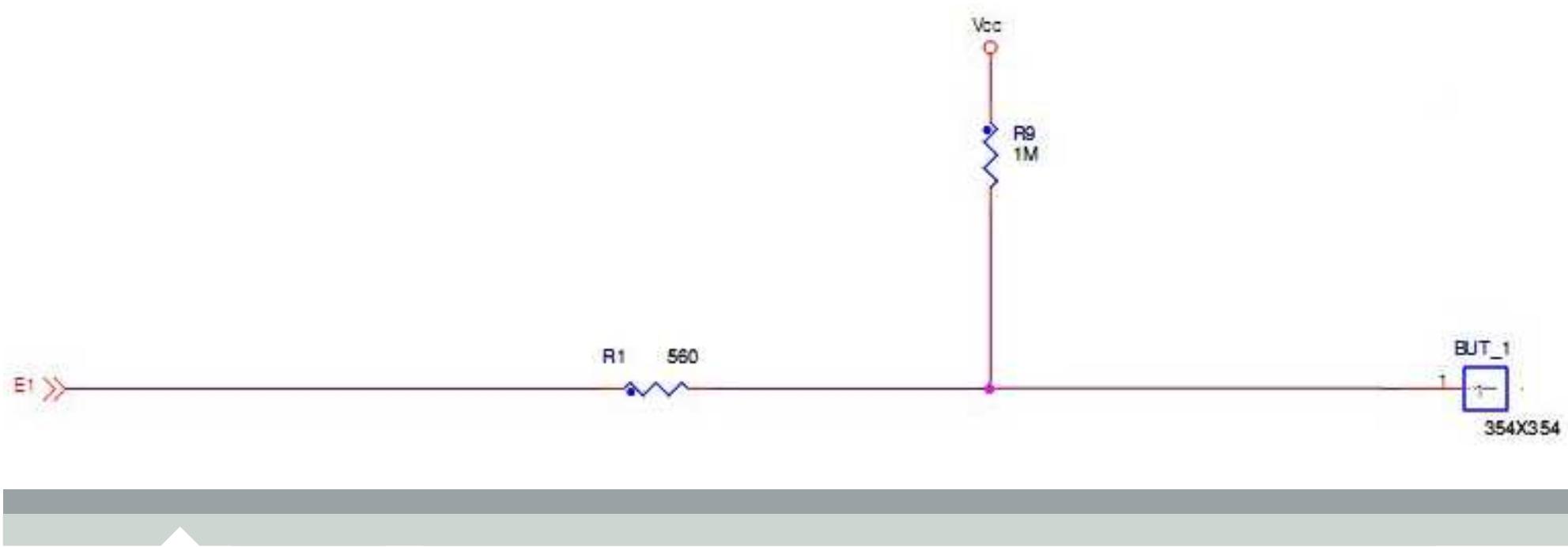
Touch Sensing System Description

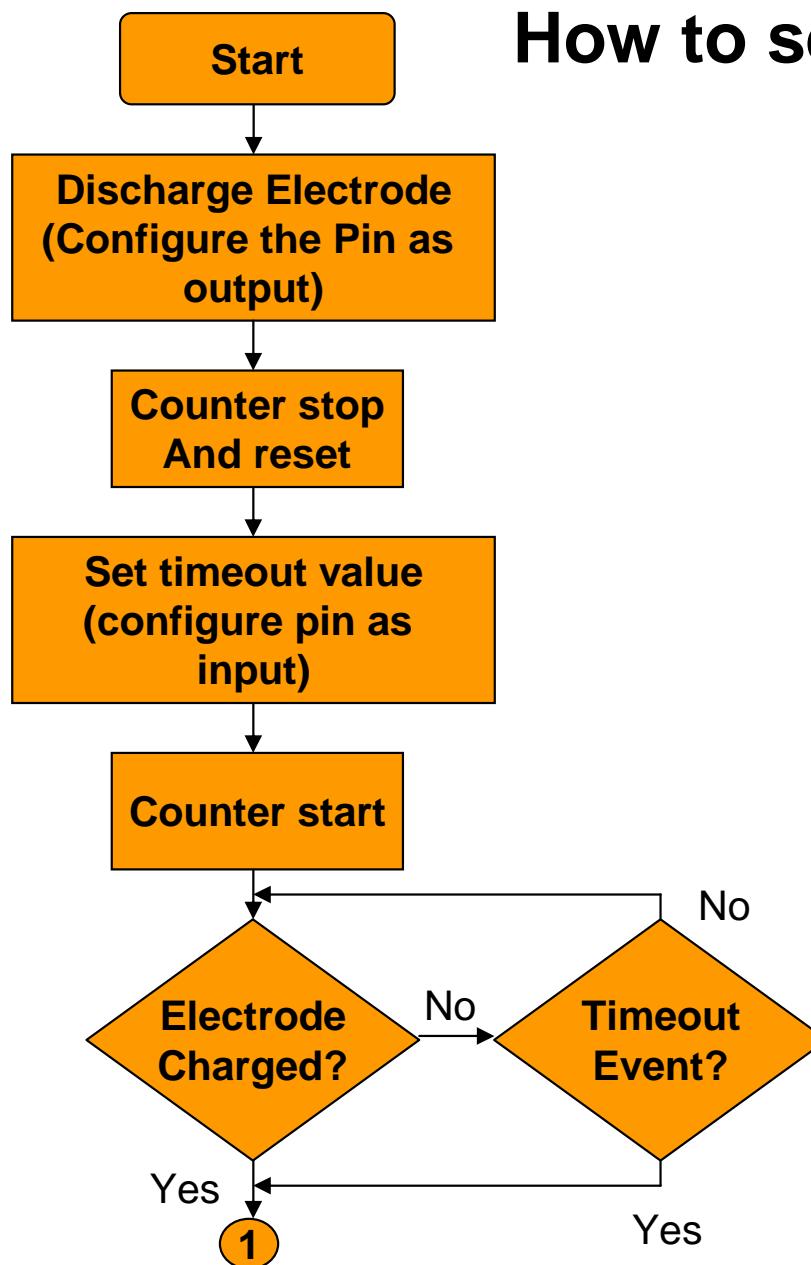
- **Electrodes:** Physical area that the user will use as Interface. It is usually made in PCB or in ITO (Indium Tin Oxide).
- **Capacitance to digital converter:** Measures capacitance on each electrode and produces a digital value as output.
- **Signal processing stage:** This stage translates measured capacitance to actual touch behavior, which is programmed here (rotary, keypad, slider...).
- **Output:** Indicate to the user that a touch was detected and to the application the actual programmed behavior.



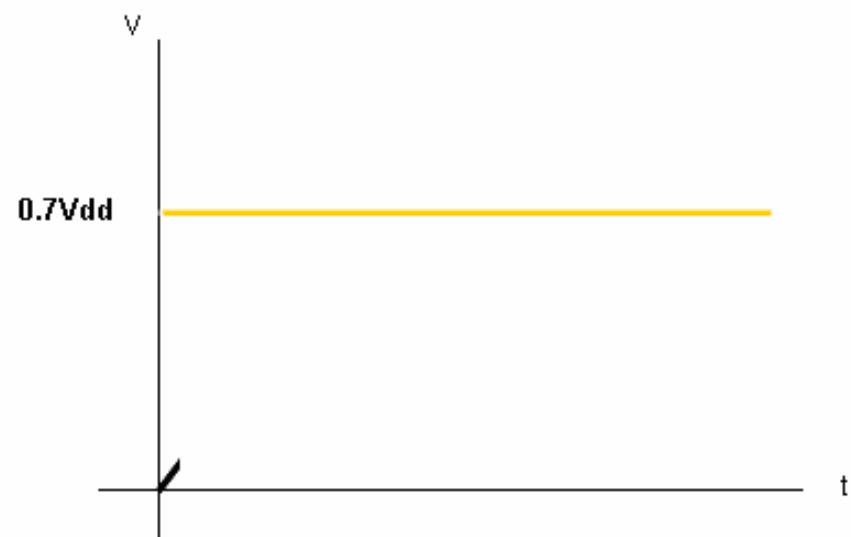
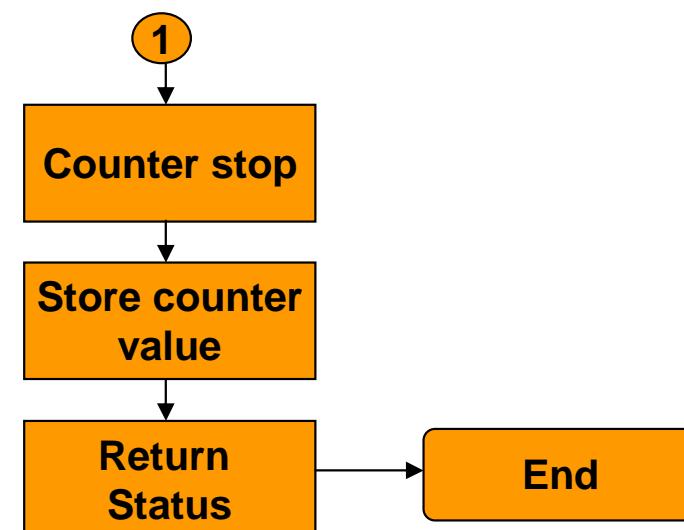
Capacitance Sensing Methodology

- ▶ Freescale's Touch Sensing Software uses an electrode's natural RC curve to determine the proximity of a user to an electrode.
- ▶ To do this, the Electrode pin is set to 0V and then checks the rise time of the electrode when pin is set as input.

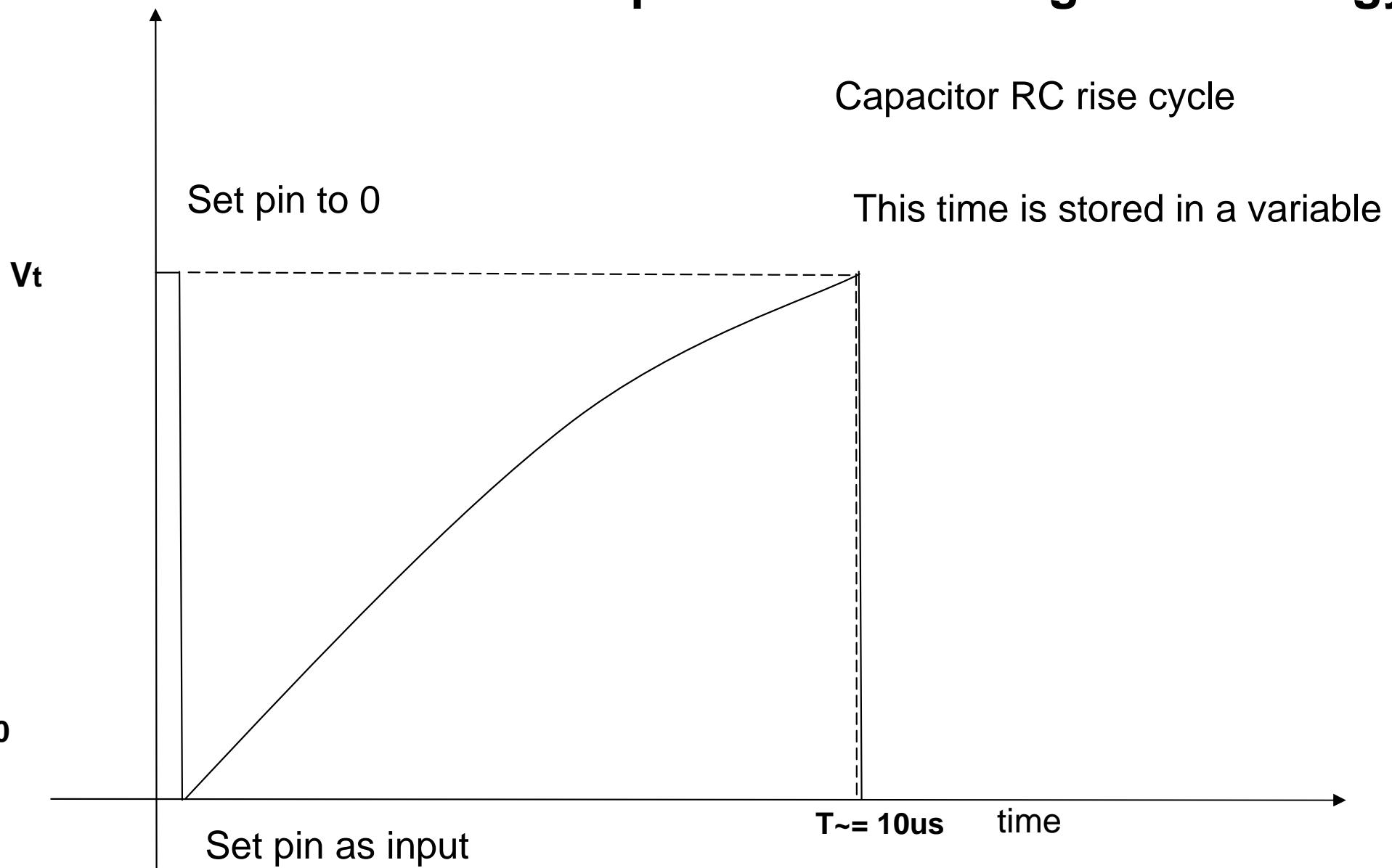




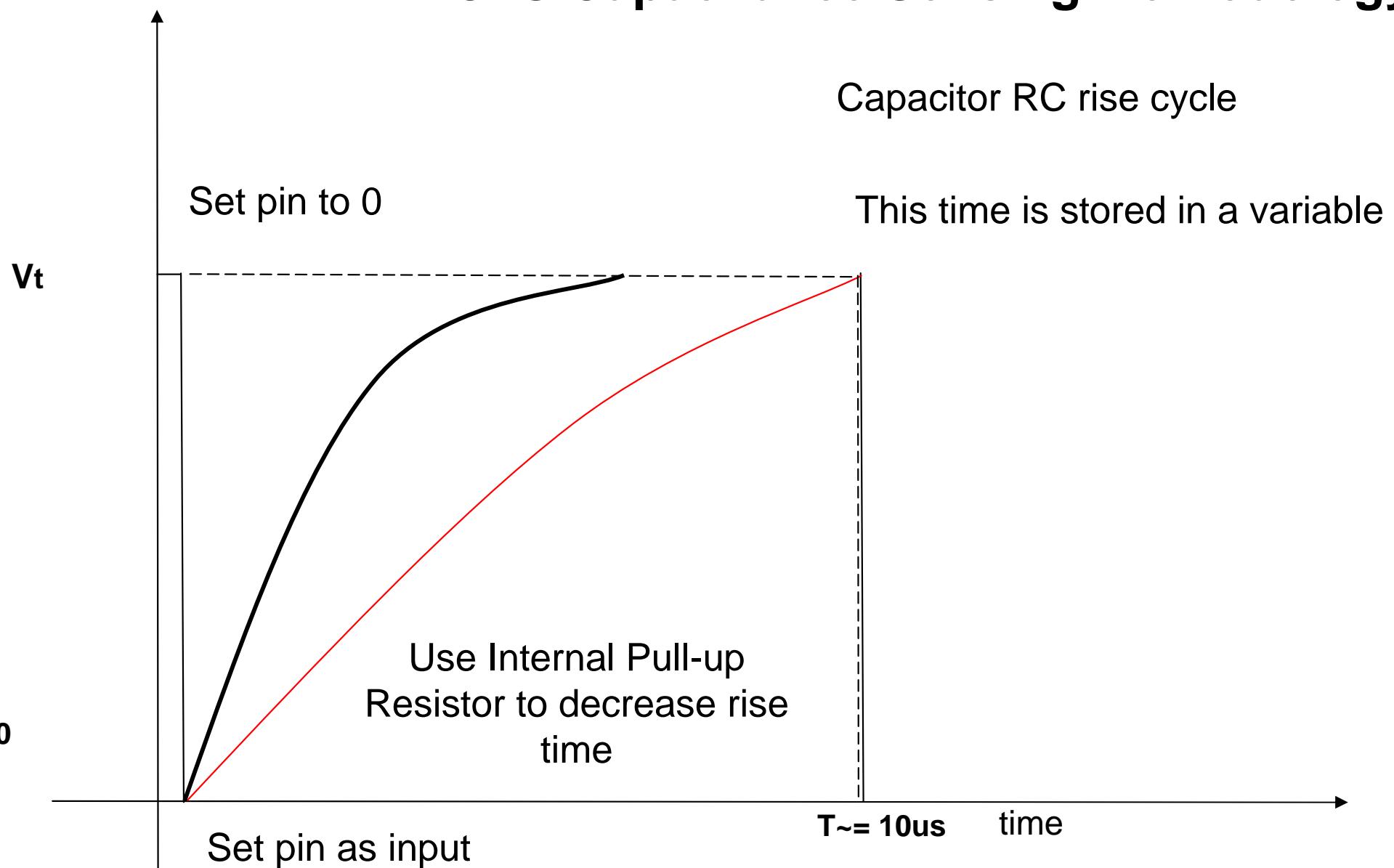
How to sense capacitance – Basic Algorithm



ATL Capacitance Sensing Methodology



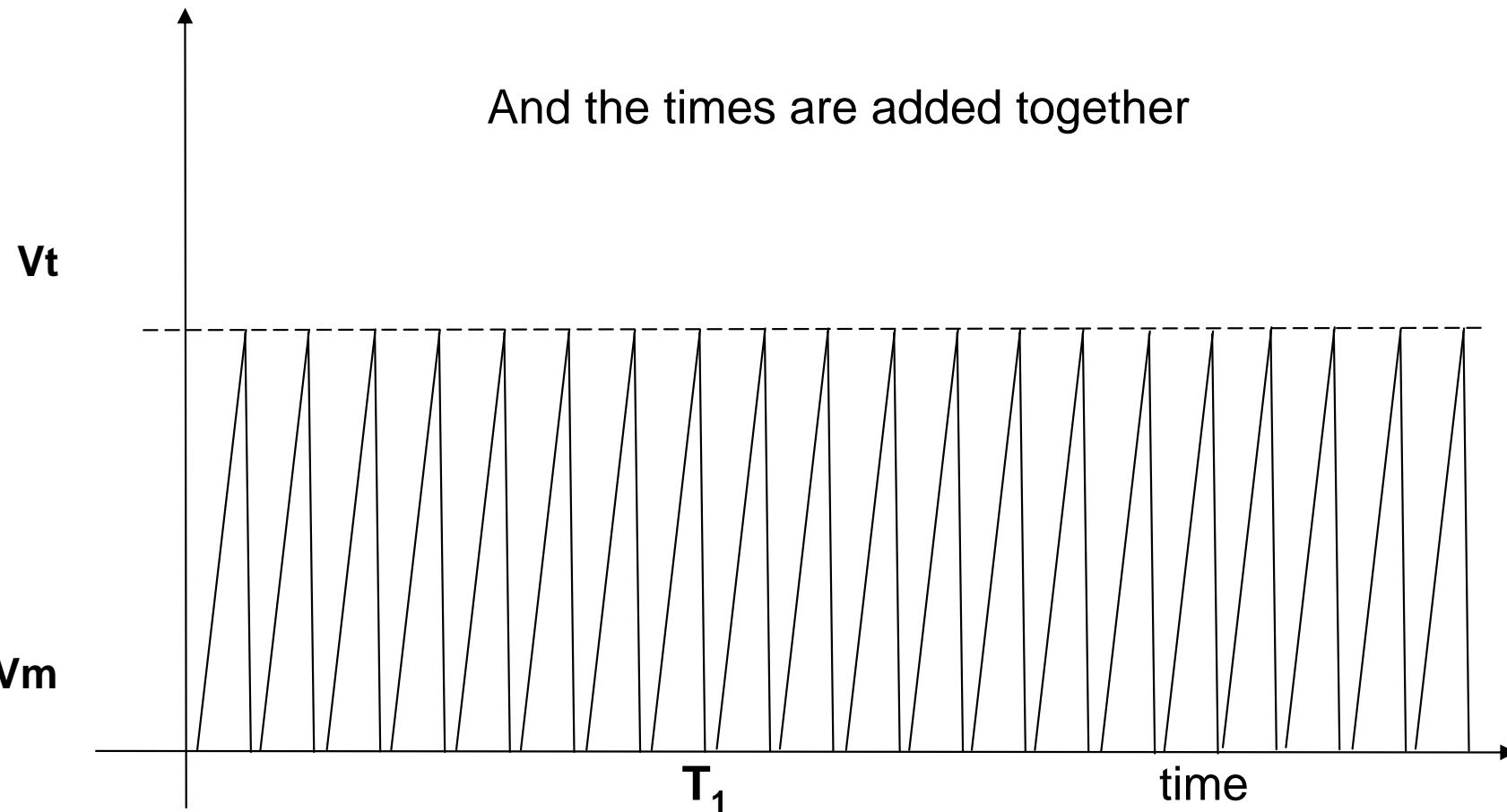
CTS Capacitance Sensing Methodology



Capacitance Sensing Methodology

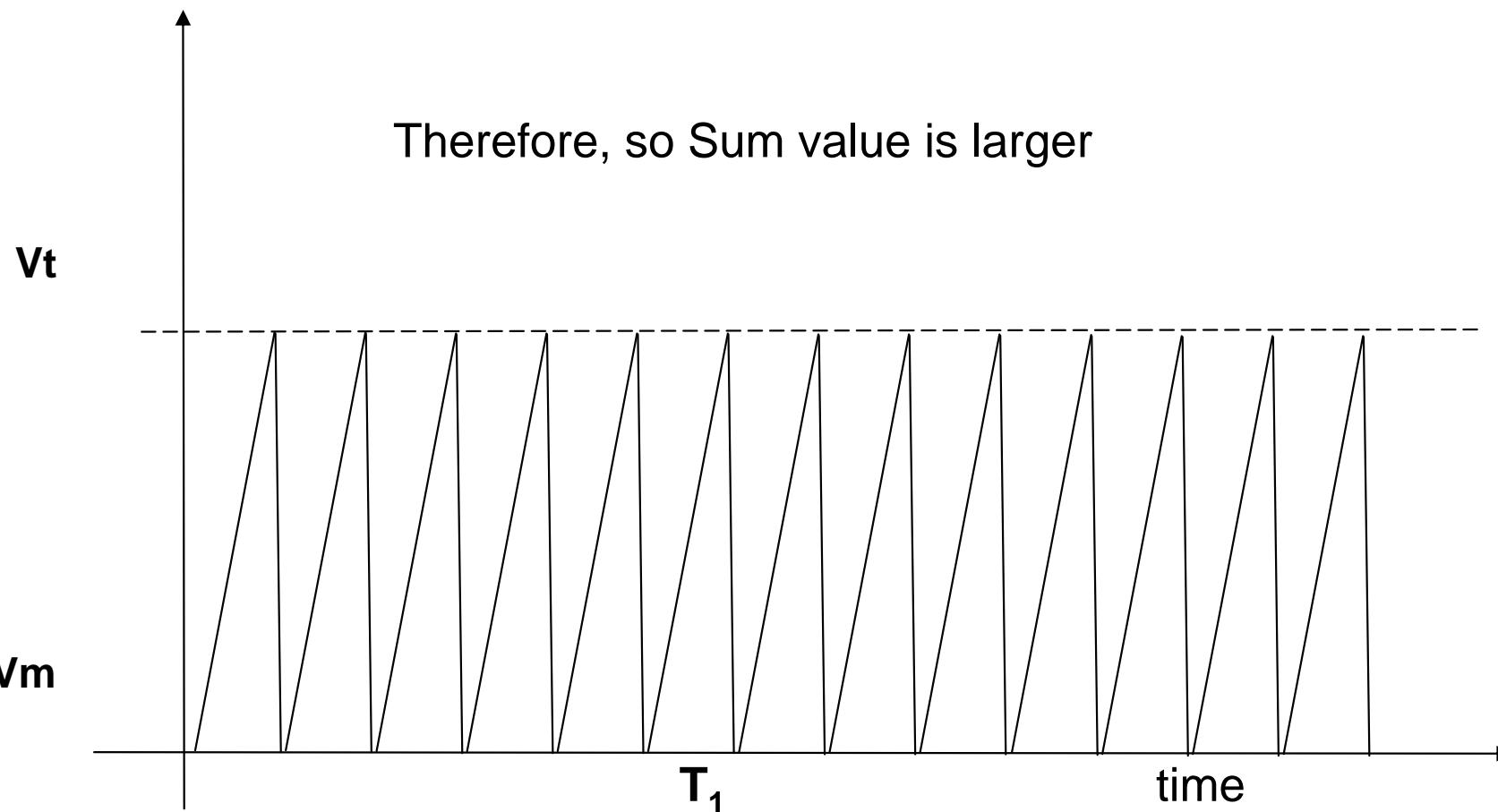
Repeat Process over and over

And the times are added together



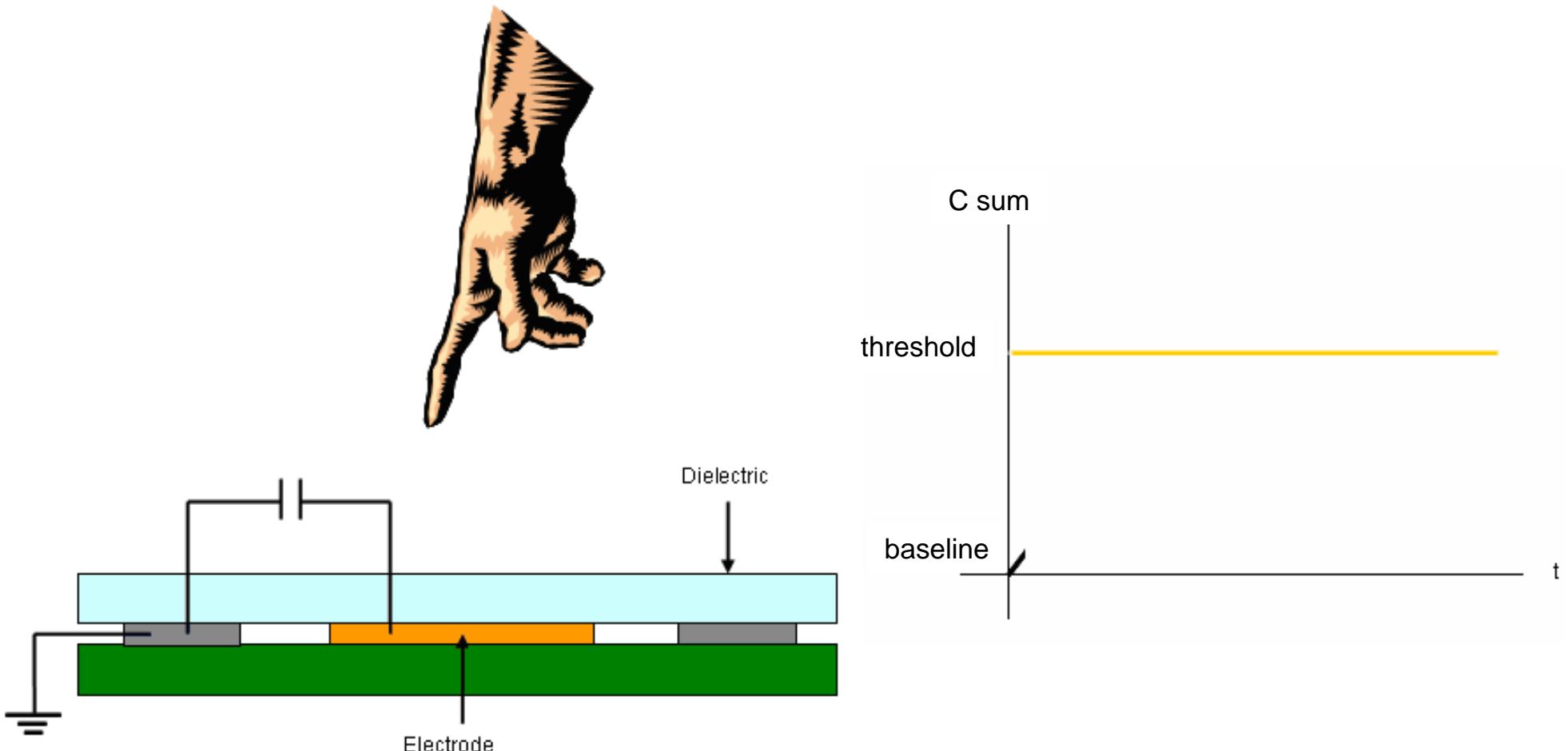
Capacitance Sensing Methodology

Whenever finger is set, rise time is larger

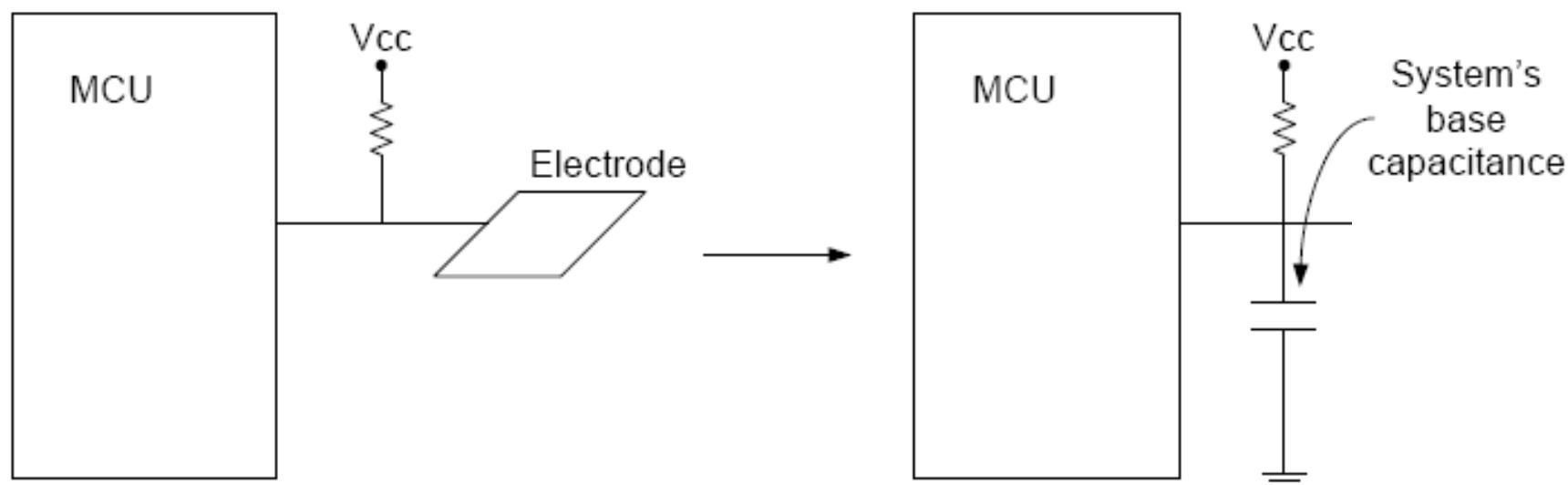


Proximity Sensing Technology

- The capacitive sensing technology works with parasitic capacitance

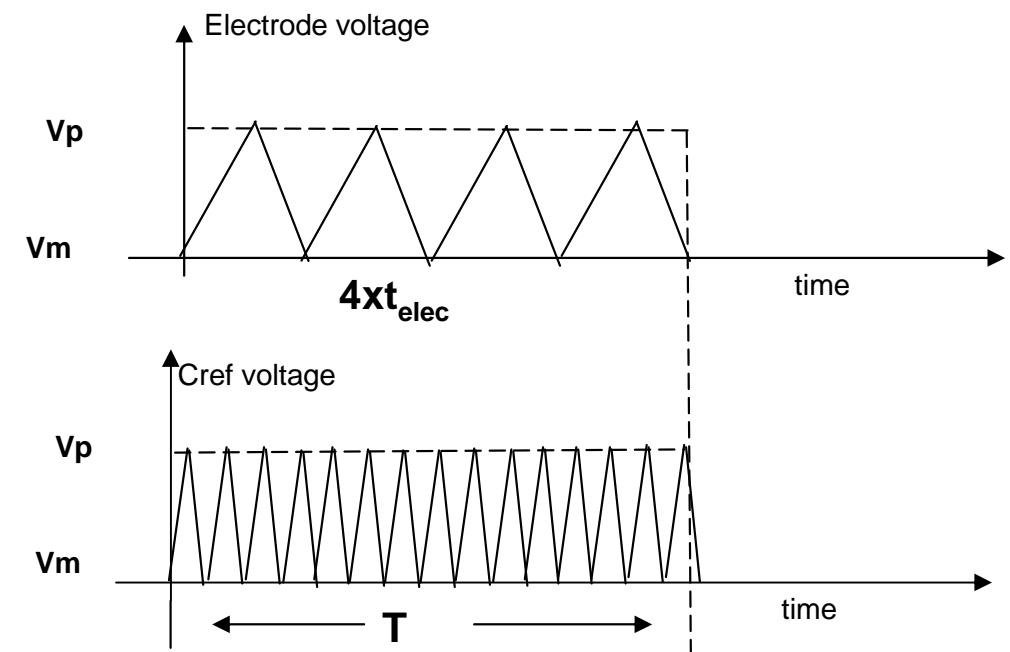
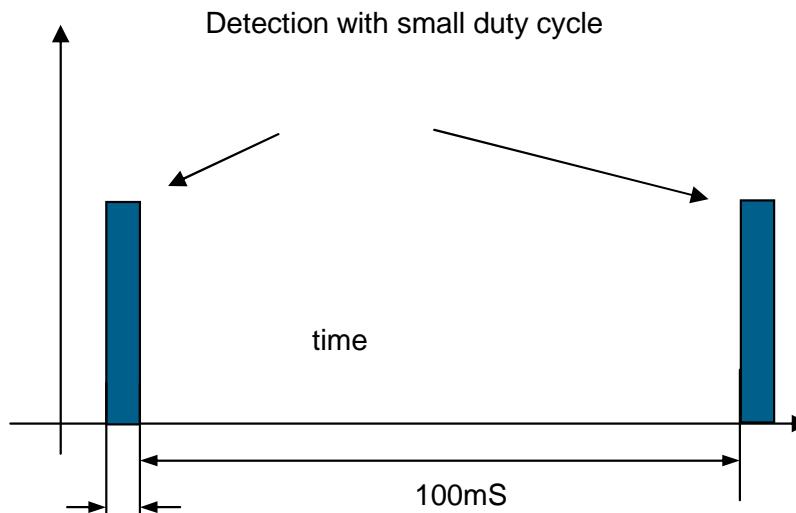


Electrodes



Capacitance Sensing Methodology

1. Every x ms all active electrodes have their capacitance measured.
2. Electrode is charged and discharged M times.
3. Time to charge is added up to obtain instant count.
4. Charge time moves as capacitance varies on electrode.
5. SW can use the variance in capacitance to detect user presence.



Touch Software Features

- ▶ Enable almost any Freescale MCU for proximity and touch sensing capability on your projects
 - Support the whole Freescale S08 family
- ▶ Allow the use and configuration of any available GPIO pin as an electrode input (up to 64 electrodes)
- ▶ Basic digital signal processing already implemented
 - Programmable sampling period
 - Adjustable touch threshold setting
 - Low pass digital filter
- ▶ Actual Touch behavior can be customized for each application

Key Benefits of SW vs IC specific

- **Flexibility**

- Allow for any electrode array combination on the same app.
- Allow the user to configure the electrode GPIOs (any FSL MCU) in any way.

- **CPU Usage**

- Restrict the execution time to avoid interfering with other running tasks.

- **Peripherals Usage**

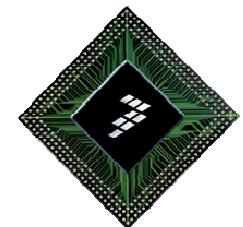
- Utilize the minimum peripherals needed for capacitive sensing
- Allow the user to implement any desired feedback.

- **Scheduling**

- Leave control of CPU timing and allocation to user

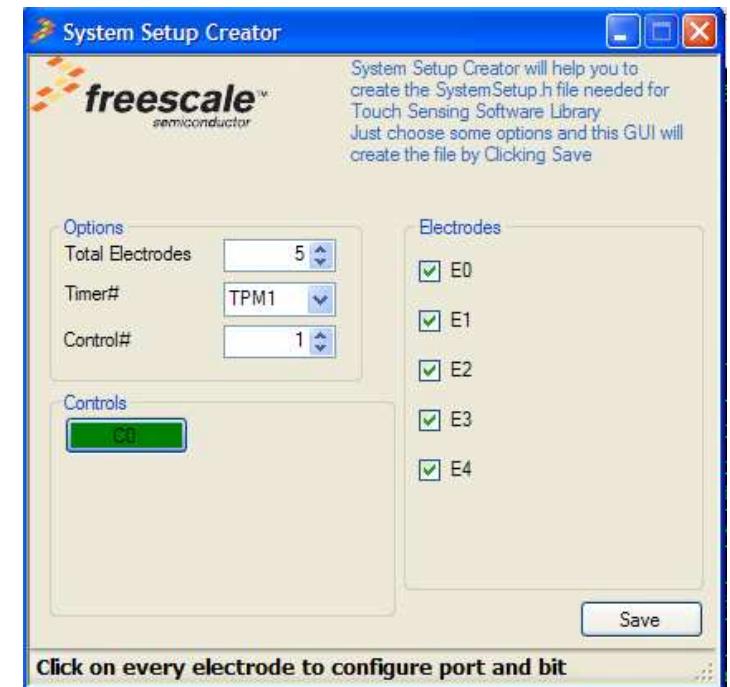
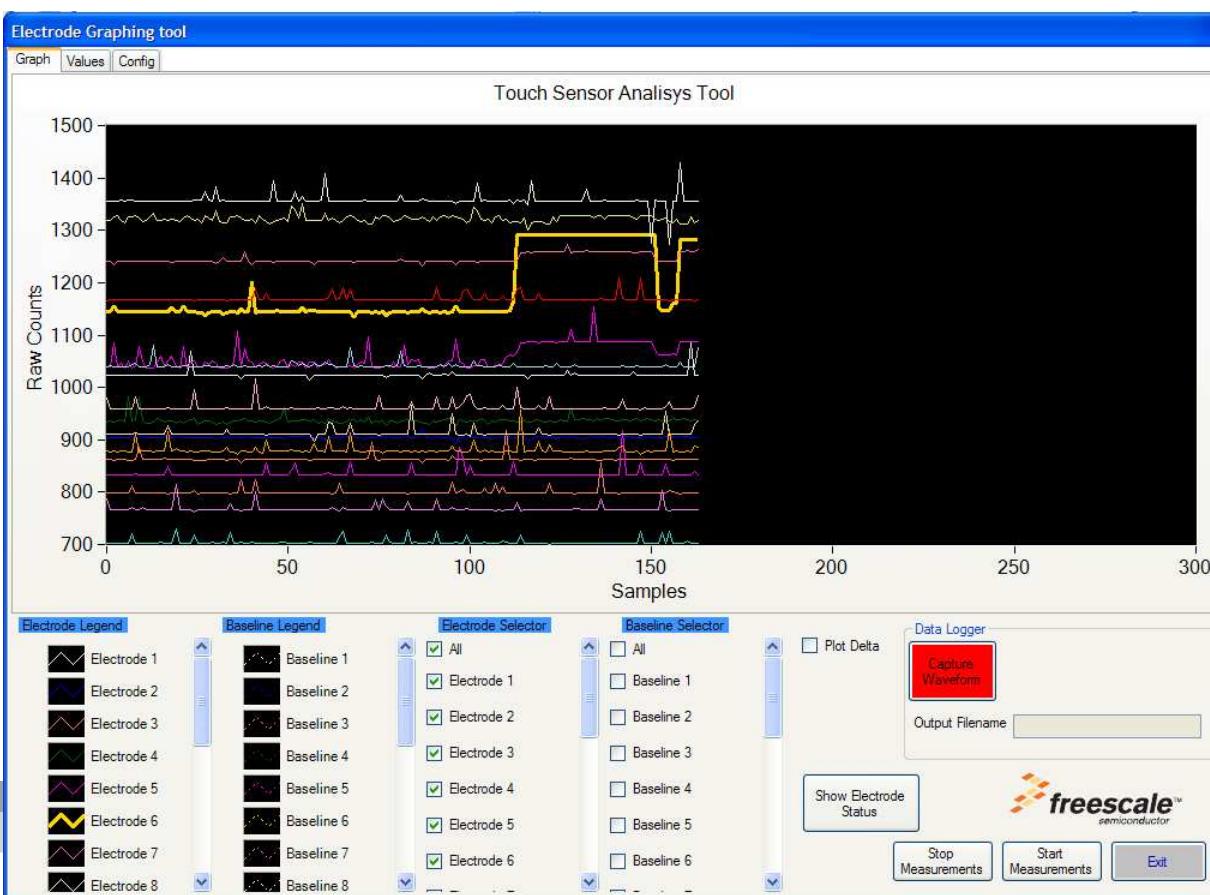


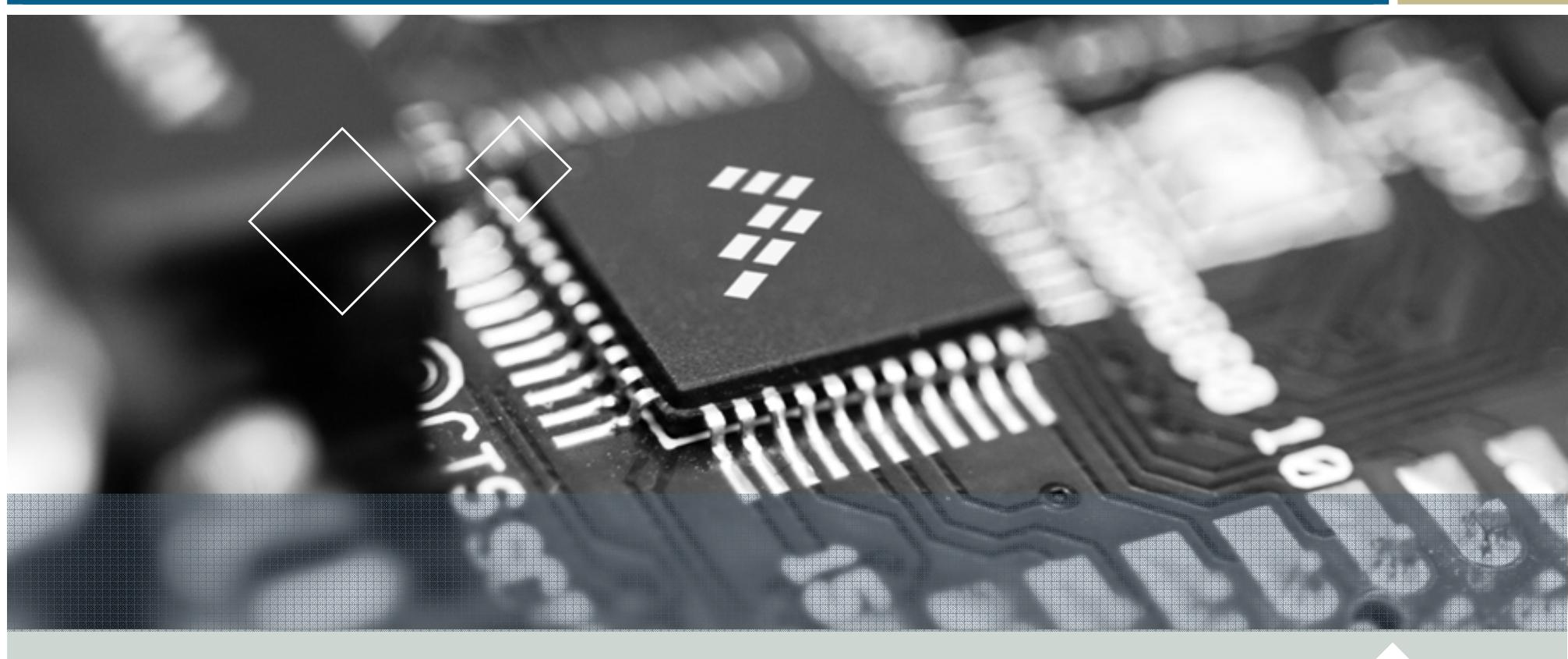
User GUIs



► Compilation time configuration GUI

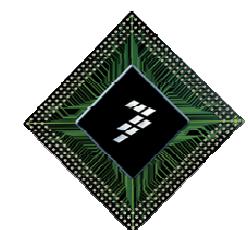
► Execution time Tuning GUIs





Electrode Design and Placement

What you need to know



Electrode Design and Placement

- One of the most important aspects of a touch sensing solution is the layout design. It is imperative to have a correct design of electrodes in order to have a reliable touch sensing solution.
- In the simplest way, proximity sensing is based on the parallel plate capacitor model in which the variation of the capacitance C is directly proportional to the area A of two parallel plates times the dielectric constant k of the object between them. The capacitance C is inversely proportional to the distance d between the plates.

Electrode Design and Placement

► Based on the model the following assumptions can be inferred:

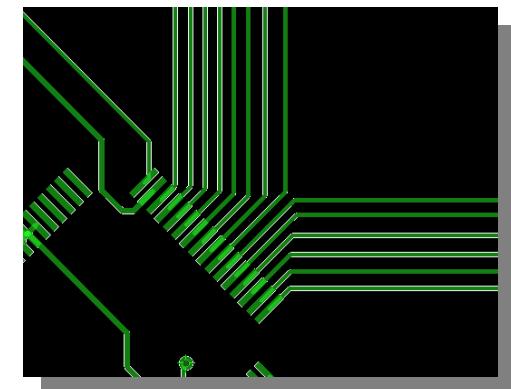
- The electrode forms one plate and the sensing objects the other plate.
- The capacitance and capacitance variance depend on the detected object size and electrode size.
- The thickness and material of the dielectric has a direct impact on the sensitivity.
- Any parasitic capacitance in the electrodes wiring or surrounding planes add a baseline capacitance to the system

Electrode Design and Placement

Traces

► Trace recommendations for the layout design of a touch sense solution.

- Width — Keep traces width as thin as possible.
5-7 mil traces are recommended.
- Clearance — Leave a minimum clearance of 10 mils.
At the sensor's end, pitch is lower than 10 mils,
use bottleneck mode connection.
- Routing layers — Only one side of the board must populated, the other side must contain only the electrodes. Proximity sensor's 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.



Electrode Design and Placement

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.



- Figure shows most common electrode patterns for single keys

Electrode Design and Placement

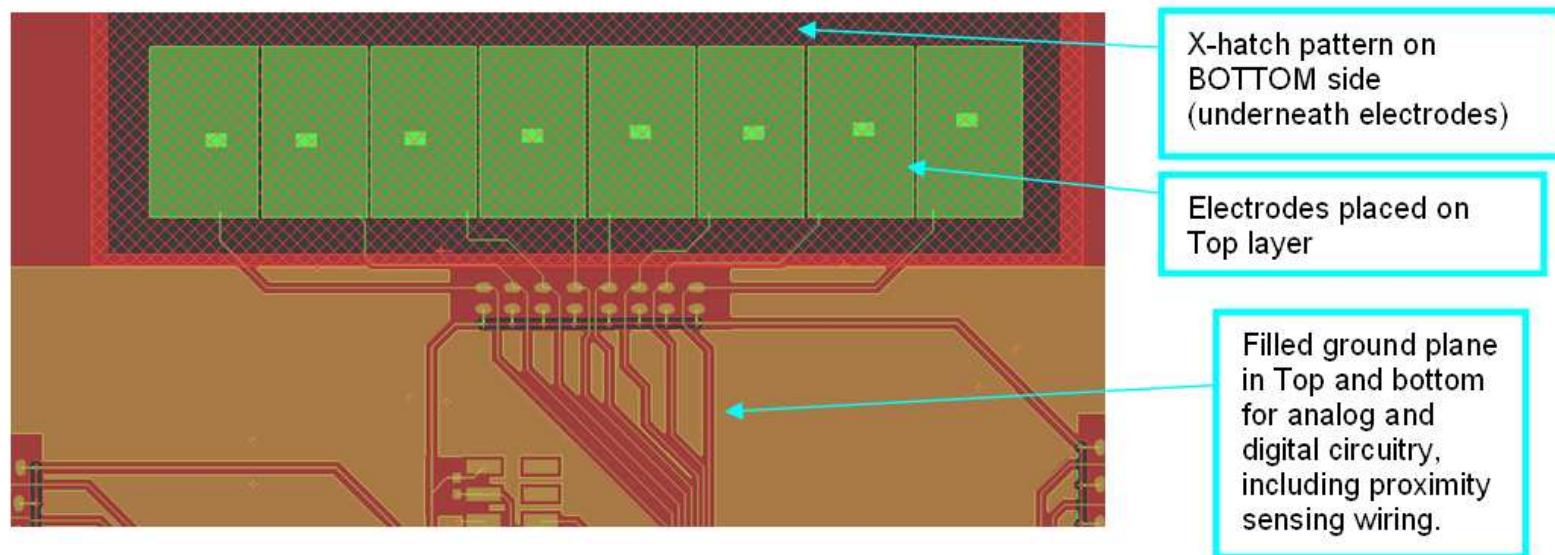
Placing Electrodes Recommendations

- ▶ Always use an external layer to place the electrodes.
- ▶ Do not place any component underneath the electrode's area.
- ▶ Height beyond the electrodes floor is not allowed.
- ▶ Choose surface mount components instead of through-hole parts
- ▶ Reverse mounting LEDs and displays.
- ▶ Two rooms must be used, one for the sensing circuitry and other for place electrodes.
- ▶ Use multi-boards solutions. One or more boards for the touch sensor circuitry and one board for wiring and the electrodes. The use of multi-boards allows to have boards with different technology such as FlexPCB, membranes, ITO, or any other.

Electrode Design and Placement

Planes

- ▶ Use ground planes around or behind electrodes only as needed
- ▶ Do not use filled ground planes underneath electrodes area
- ▶ Use X-hatch pattern underneath the electrodes area
- ▶ Use filled ground planes for analog and digital circuitry



Electrode Design and Placement

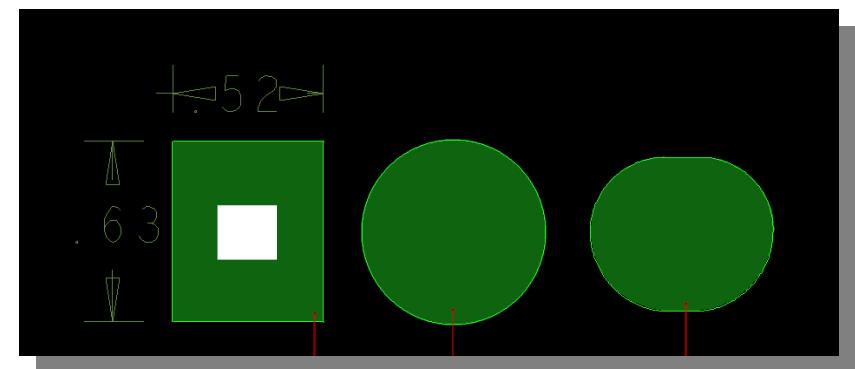
Electrode Patterns

► Electrodes with simple shapes as circles, rectangles, or ovals with a size similar to the object to be detected are a good starting point for designing proximity sensing applications. However, based on an electrode's function, its geometry can be modified to achieve better results or improve sensitivity.

► Single Keys

If LEDs are desired to be shown

- The shape of the electrode must be as similar as possible to the human fingers' (0.6 Inches x 0.6 inches is considered a good size).

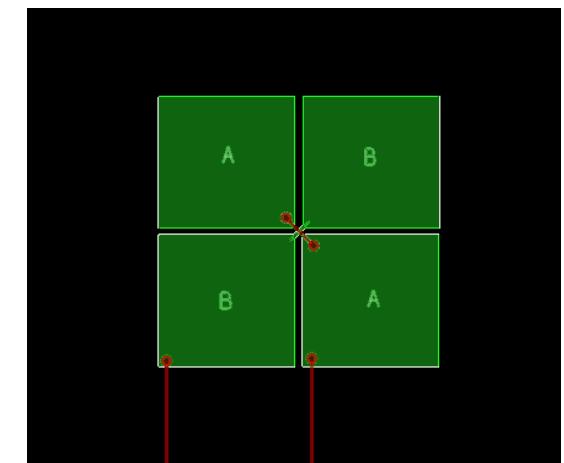


Reverse Mounting LEDs are recommended to be used

Electrode Patterns

► Multiplexed Keys

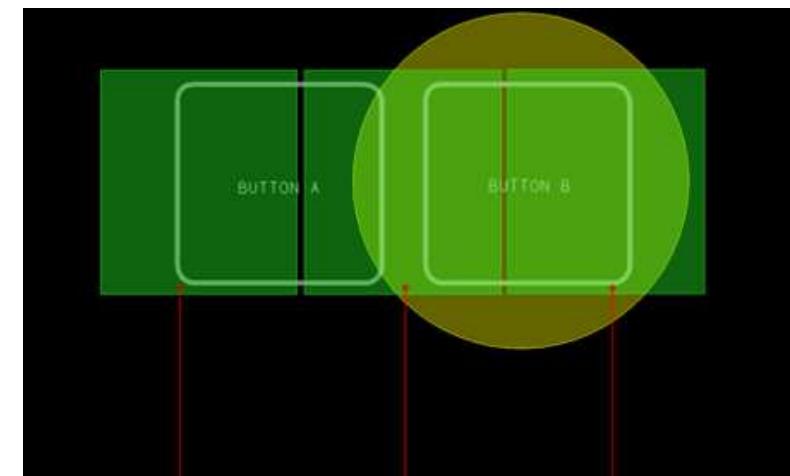
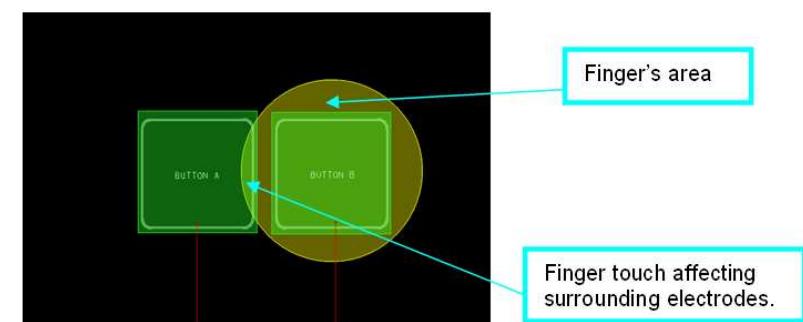
- When the number of keys exceed the number of sensor channels available, multiplexing or logical combinations can be created.
- The detection of a valid key is not directly the change in capacitance of a single electrode but of two or more electrodes at the same time.
- When multiplexing a key, the contact area for each electrode is reduced at least to half.
- Figure shows the recommended geometry for multiplexing a key.



Electrode Patterns

► Size-Restricted Applications

- In Size- Restricted applications the key area is smaller than a finger's transversal area or the keys are so close to each other that multiple keys are detected when a key is touched.
- Reducing the electrode size does not address the problem in most cases because electrode sensitivity is reduced.
- In such cases, it is recommended to multiplex keys in a different way by using some proximity sensing electrodes as common electrodes for two or more keys.

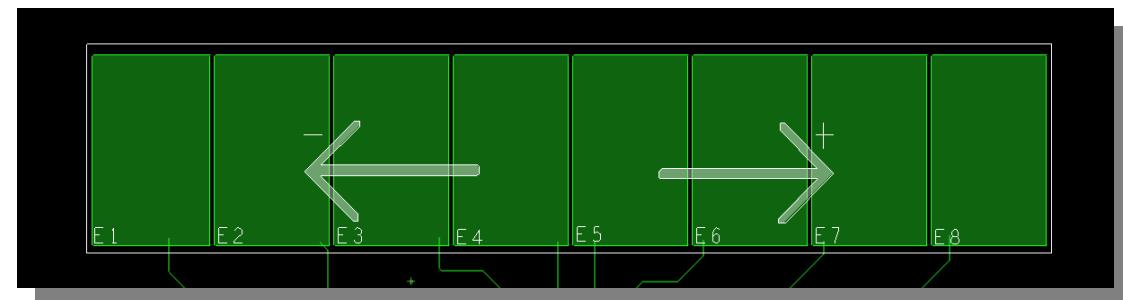


Electrode Design and Placement

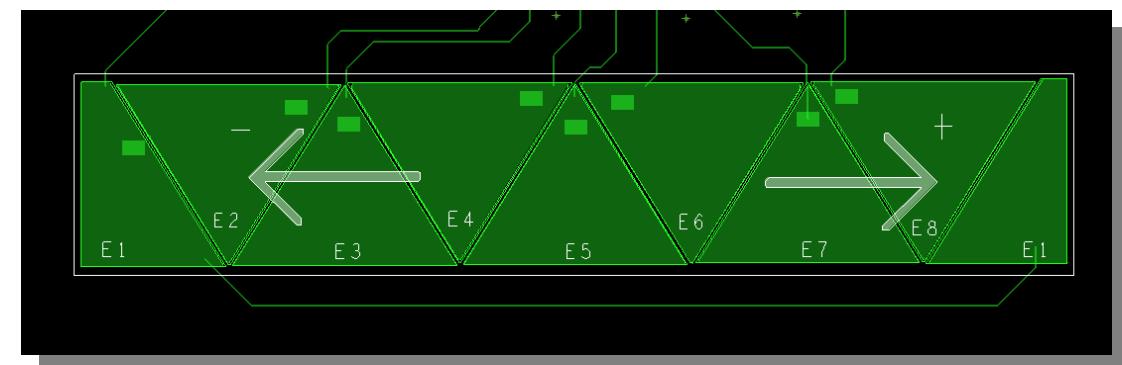
Electrode Patterns

► Sliders

- Basic Slider pattern.



- Slider pattern for micro-stepping implementation.

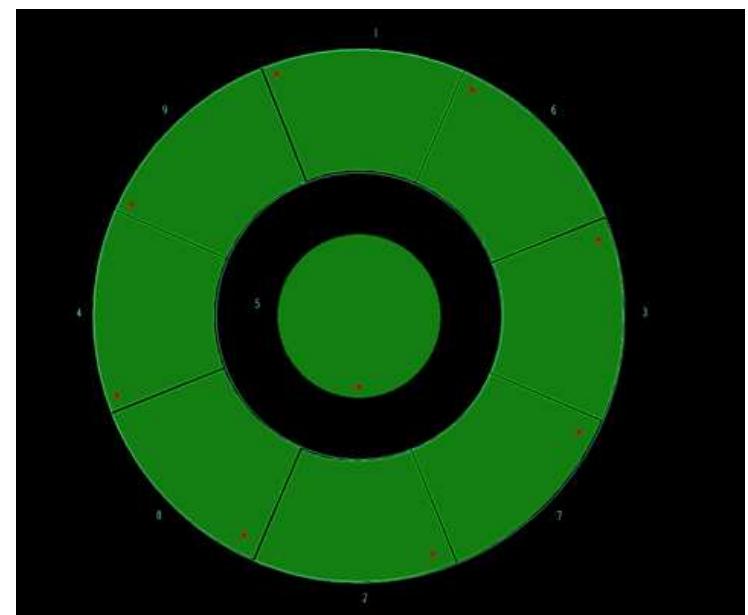


- When using this topology, ensure that the height of the electrodes is not that large so that finger movement in the perpendicular way could be detected as displacement along the slider.

Electrode Patterns

► Rotaries

- Figure 19 shows the recommended proximity electrodes array for a rotary. This simple topology also includes a center button, optional for some applications.

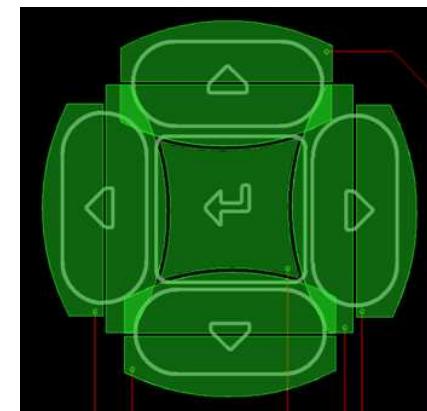


Electrode Design and Placement

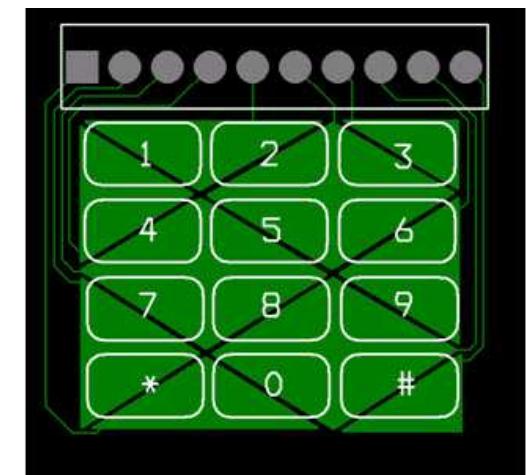
Electrode Patterns

► Keyboards

- The arrow keyboard typically consists of five keys; a central key surrounded by an arrow key at each side.

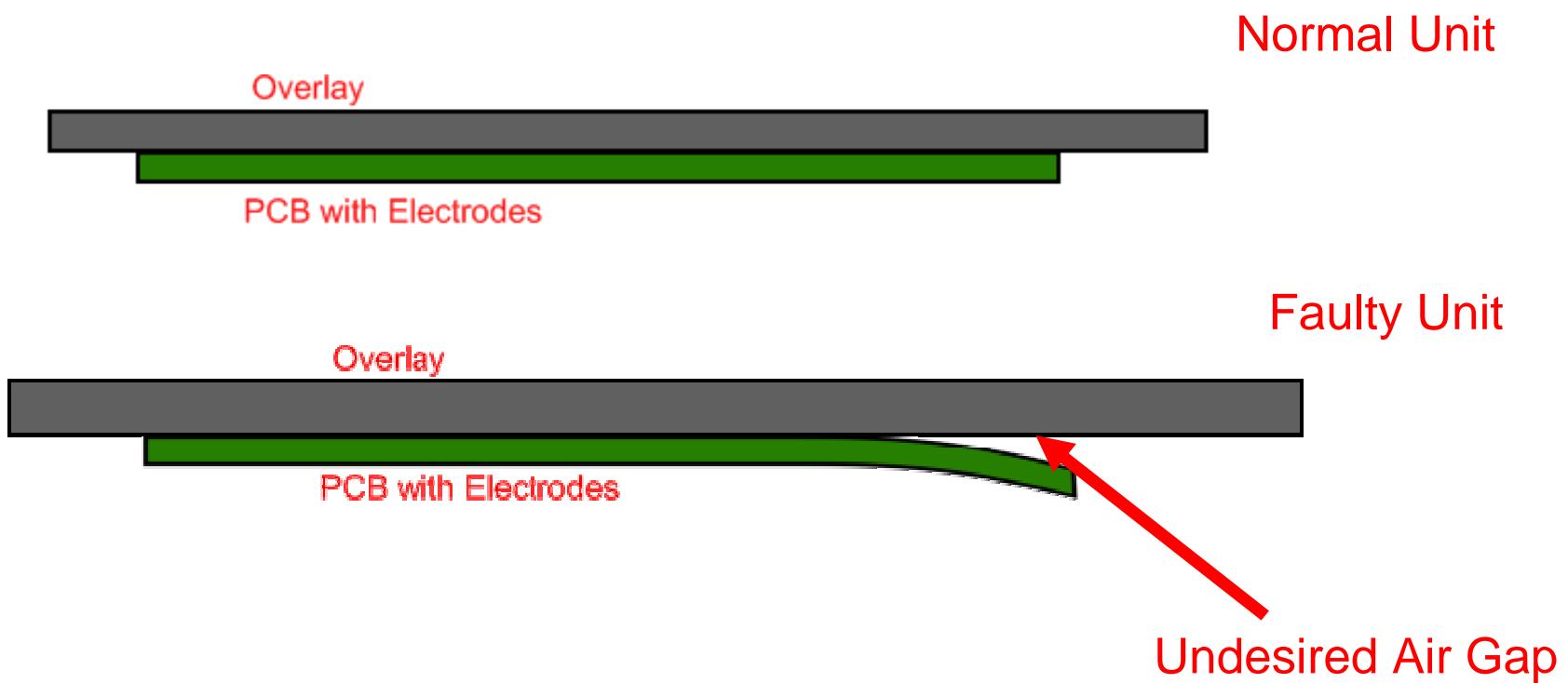


- The Keyboard multiplexed consists in arranging the capacitive electrodes on a keyboard so that each key is detected with a combination of two capacitive electrodes.



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.



AN3863

Designing Touch Sensing Electrodes

AN3516

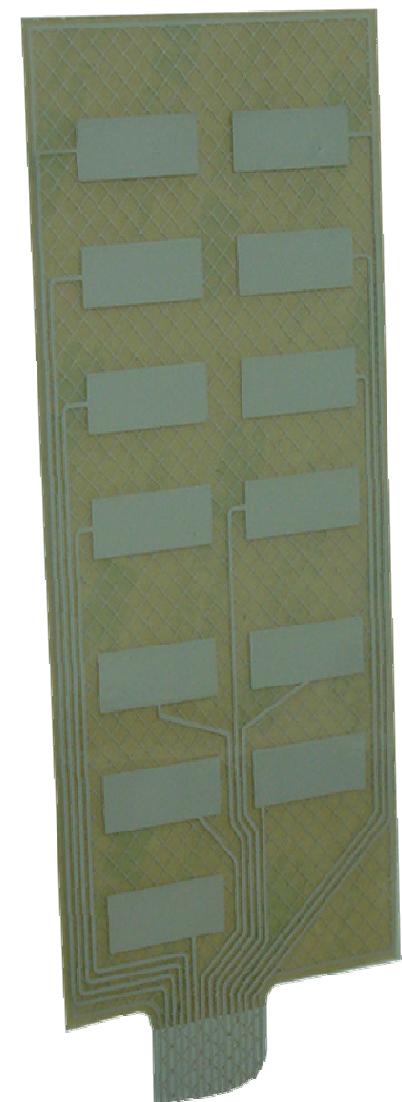
E-field Keyboard Designs

AN1985

Touch Panel Applications Using MC34940

Capacitive Film Electrodes

- ▶ Variance of Membrane Switches
- ▶ LED Backlight
- ▶ Some Surface Mount Components



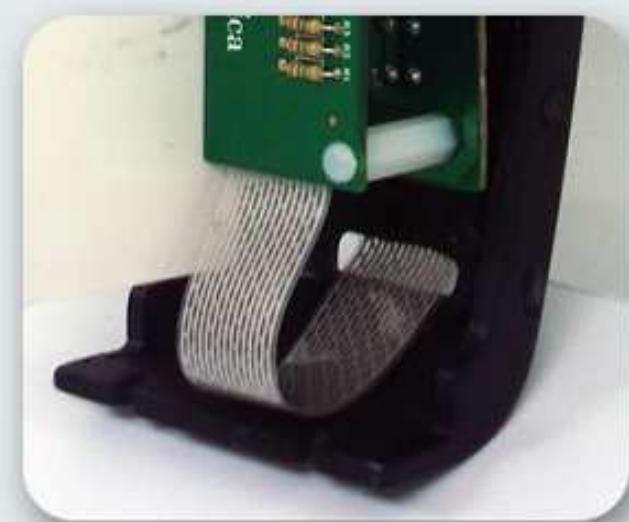
Application Example

- Customer Toaster
Oven
- Need for curved
overlay



Membrane

The membrane is placed on the panel, which contains an incision through which the membrane connector tail is inserted and connected to the Adapter board.



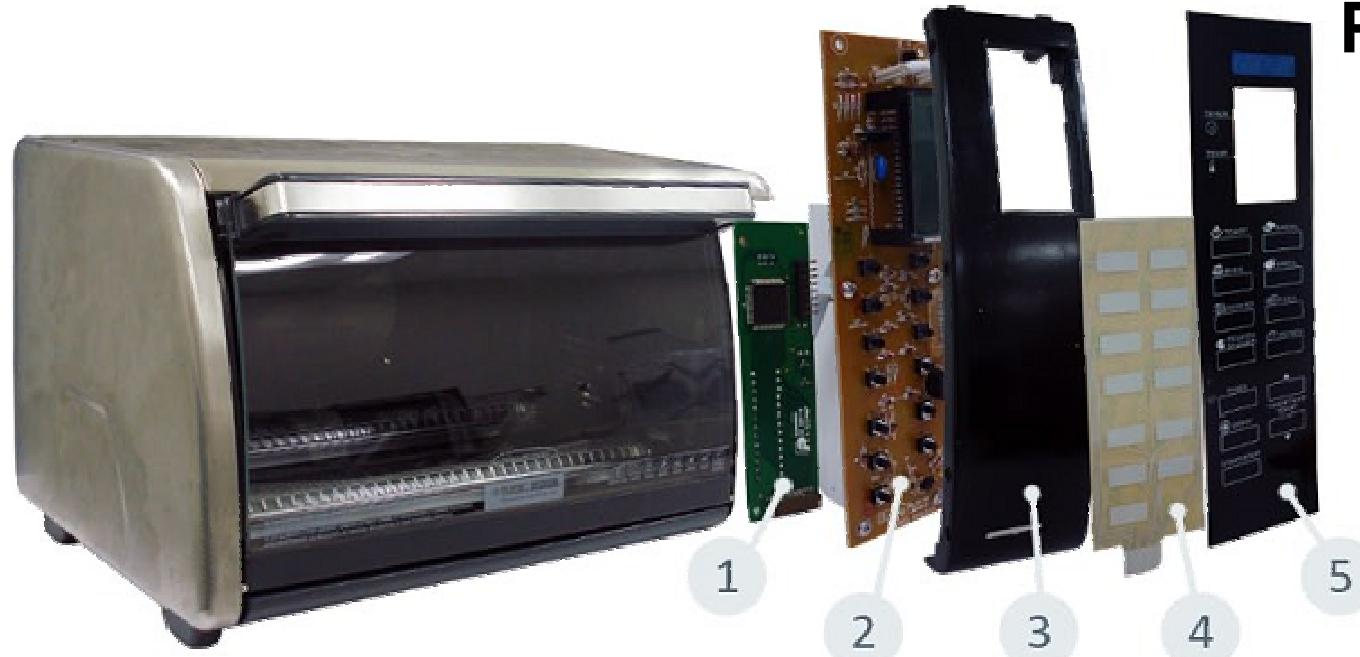
Assembly

The touch sensing membrane is placed on top of the oven side panel, and the tail is inserted through the incision. The overlay is then placed over the entire panel, covering the touch sensing membrane. On the reverse side, the adapter board is placed using two standoffs. The membrane and the oven board interface cable are both connected to the adapter board. The integrated sidepanel is then placed inside the oven and connected the same way as the original oven sidepanel.



Putting it all Together

Components



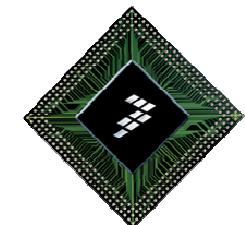
- 1. Adapter board:** Contains an 8-bit Freescale Microcontroller running the Freescale touch sensing library and simulating button presses on the Oven Board.
- 2. Oven Board:** The original oven board. It is not aware of the touch sensing application. A flat cable was added to send the required signals from the adapter board to emulate the buttons.
- 3. Panel:** A plastic panel on which the membrane is placed.
- 4. Membrane:** The capacitive touch sensing membrane contains 13 electrodes arranged the same way as the original oven keypad.
- 5. Overlay:** Contains the keypad design for the oven functions.

Assembled Oven





How to enable Touch Sensing Software



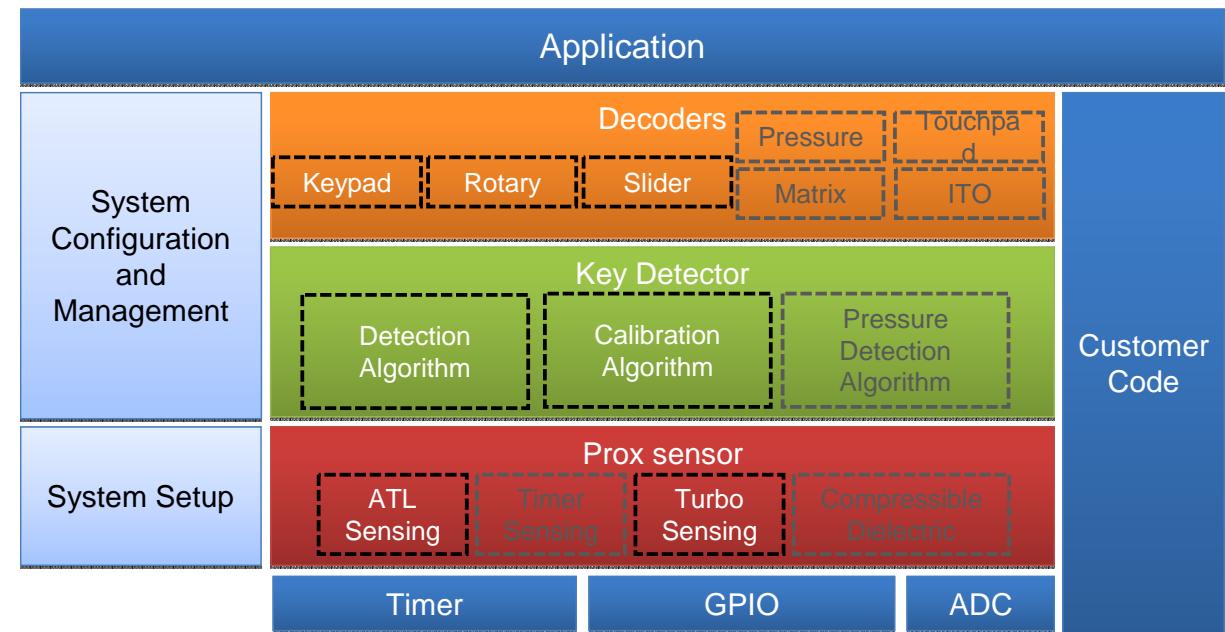
- ▶ GPIO module — Manages everything referred to the MCU's I/O lines. This module provides the interface with the electrodes and the system output, in this case a buzzer.
- ▶ Timer module — Manages all the hardware timer module functions used in the application. For example, timer clocks and prescaler configuration, counter start, and stop and reset. The timer's ISR is also located in this module.
- ▶ Sensing module — This is the core of the application and interfaces with the GPIO and timer modules to obtain capacitance measurements. This module processes the capacitance values and evaluates if an electrode has been touched.

► Low level Interface

- System Setup Module

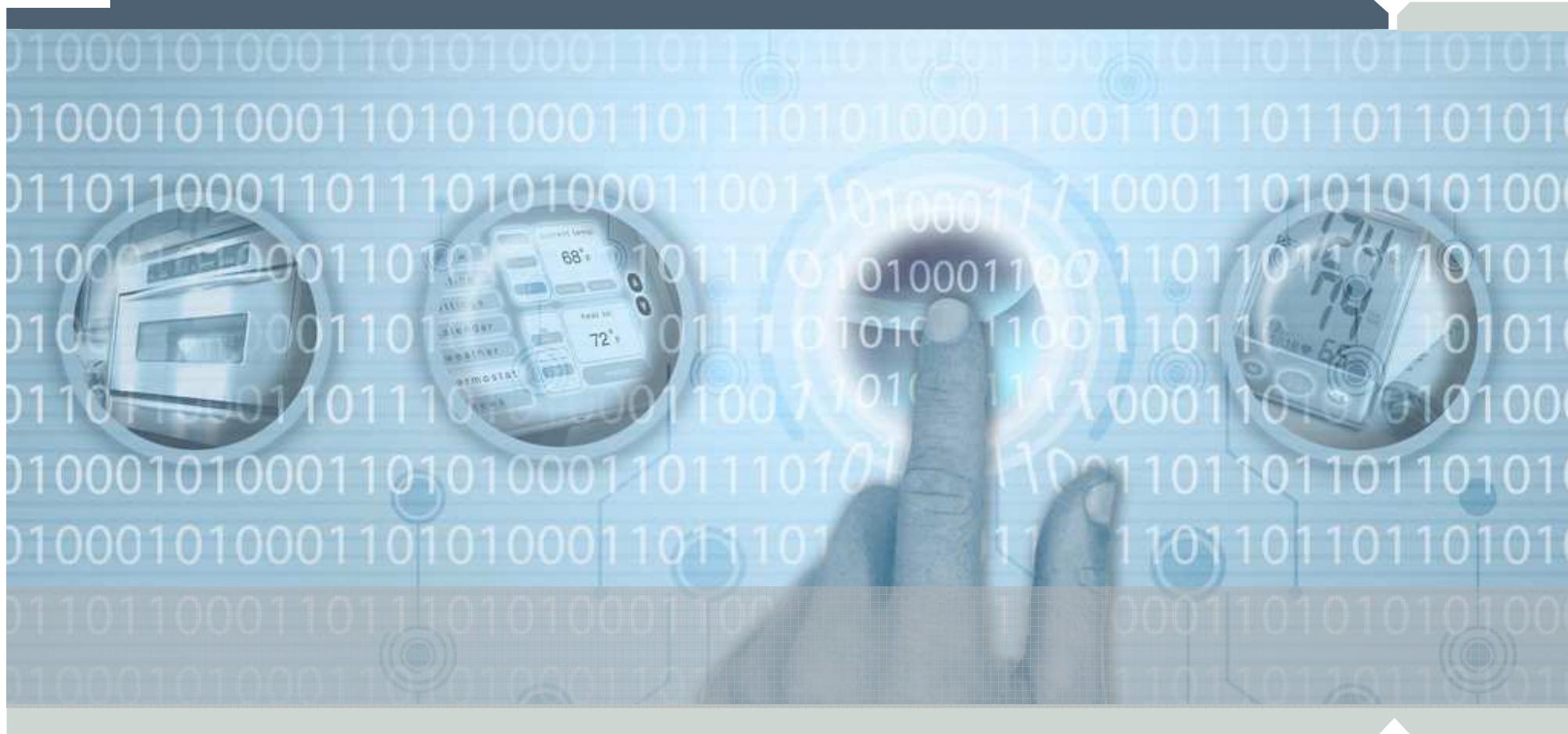
► Application Interface

- Initialization Function
- Task Function
- Register Configuration
- Keypad Decoder
- Rotary Decoder
- Slider Decoder

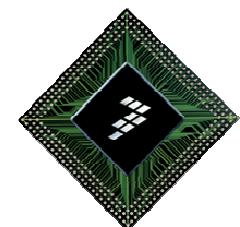


Low Level Interface

- ▶ The System Setup Module is found in the low level interface, this module must be configured by the user to determine which hardware peripherals will be used by the TSS.
- ▶ The System Setup Module is configured in the TSS_SystemSetup.h file
- ▶ The TSS_SystemSetup.h file must be edited before the application is compiled
- ▶ Parameters:
 - Capacitive Sensing Method
 - Drive Strength
 - Slew Rate
 - Number of electrodes
 - Port and Pin for each electrode
 - Number of controls
 - Control type
 - Number of electrodes per control
 - Structure name
 - Callback function name
 - MCU Timer module
 - Instant delta values

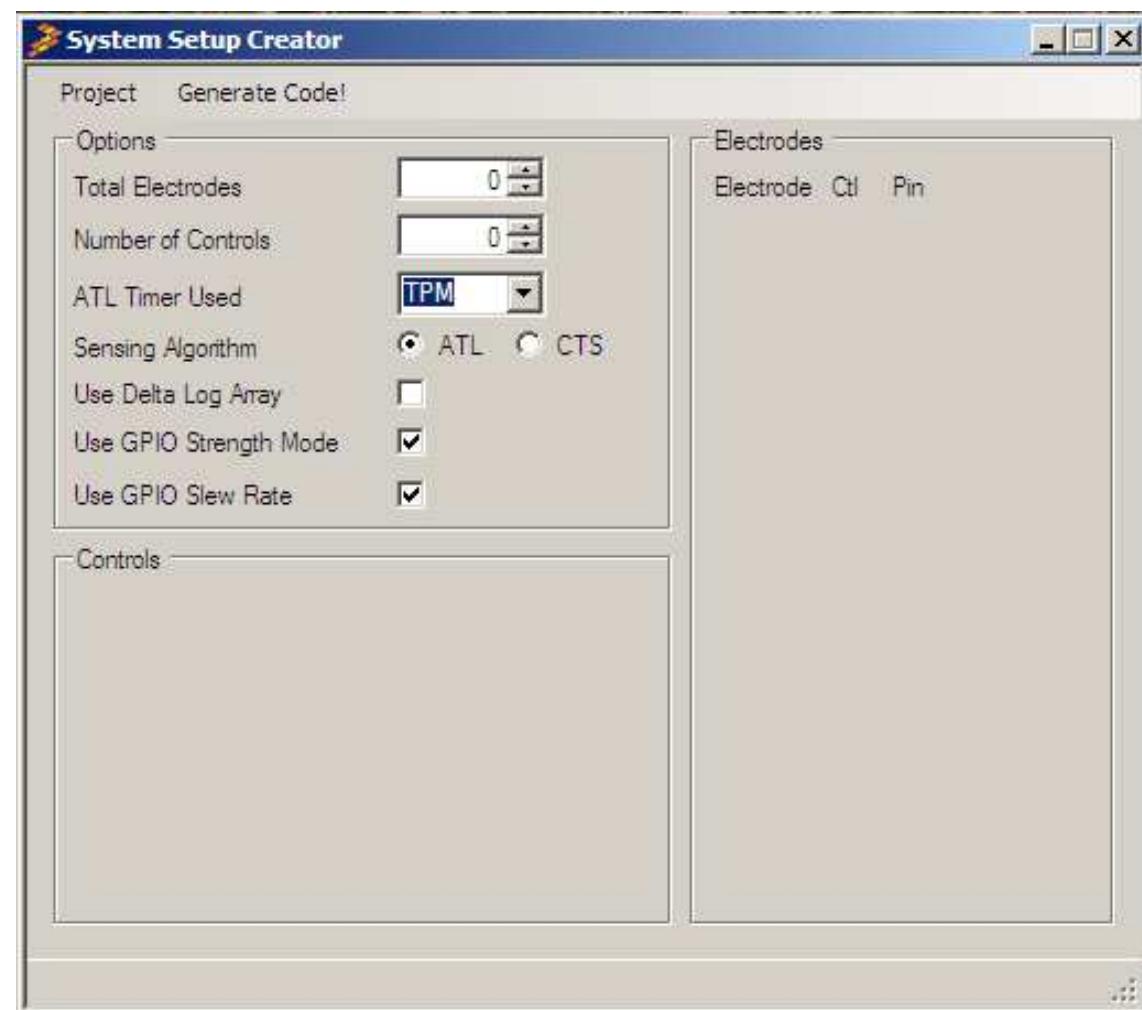


System Setup Creator



The System Setup Creator

- ▶ This program enables users to create the SystemSetup.h file used by the TSS library.
- ▶ Projects can be saved and accessed later by users.
- ▶ Enables users to modify all aspects of low level driver.



Configure Electrodes to use

► Set Total Electrodes to 1

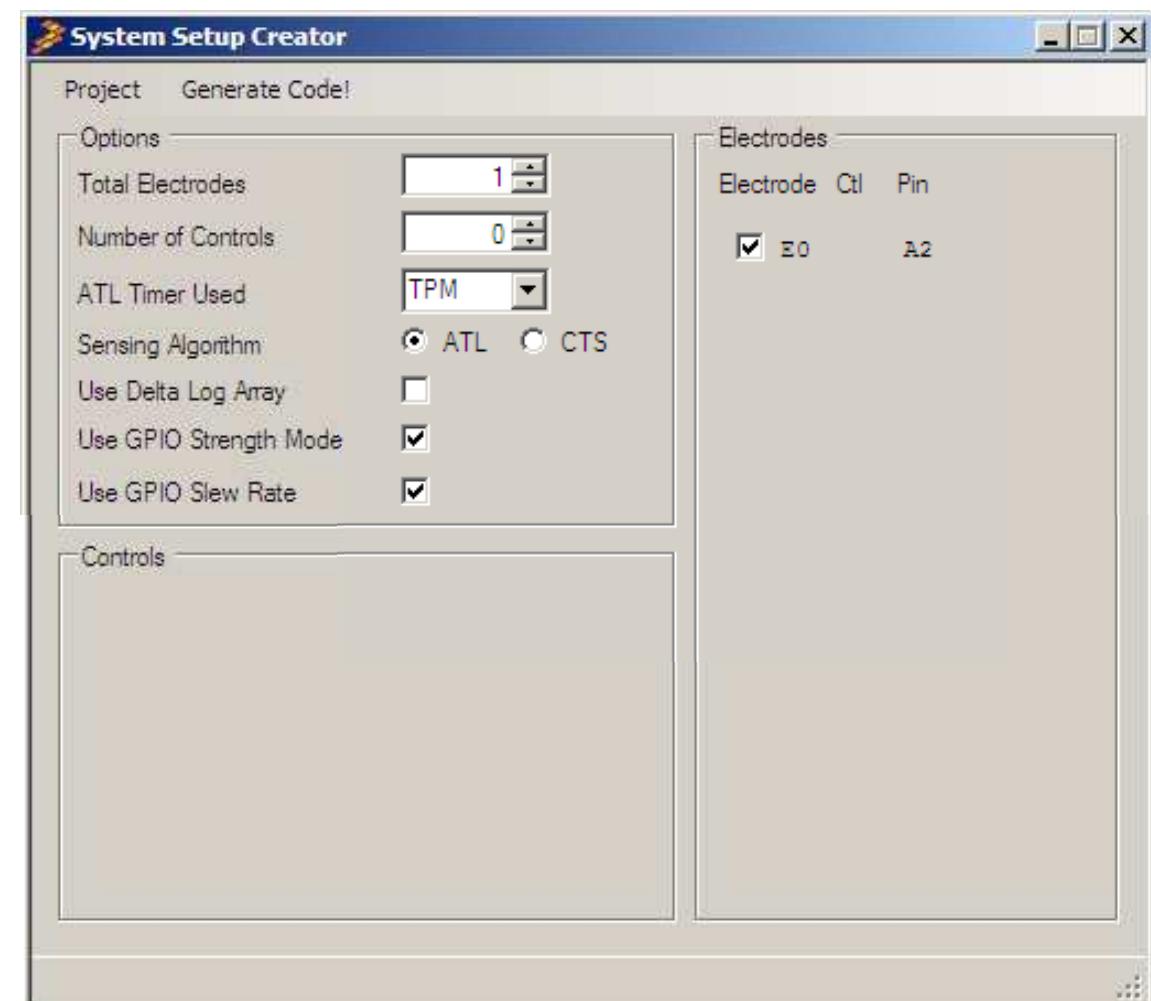
► Next Click on E0

► Select Port A, Bit 2

► Click OK

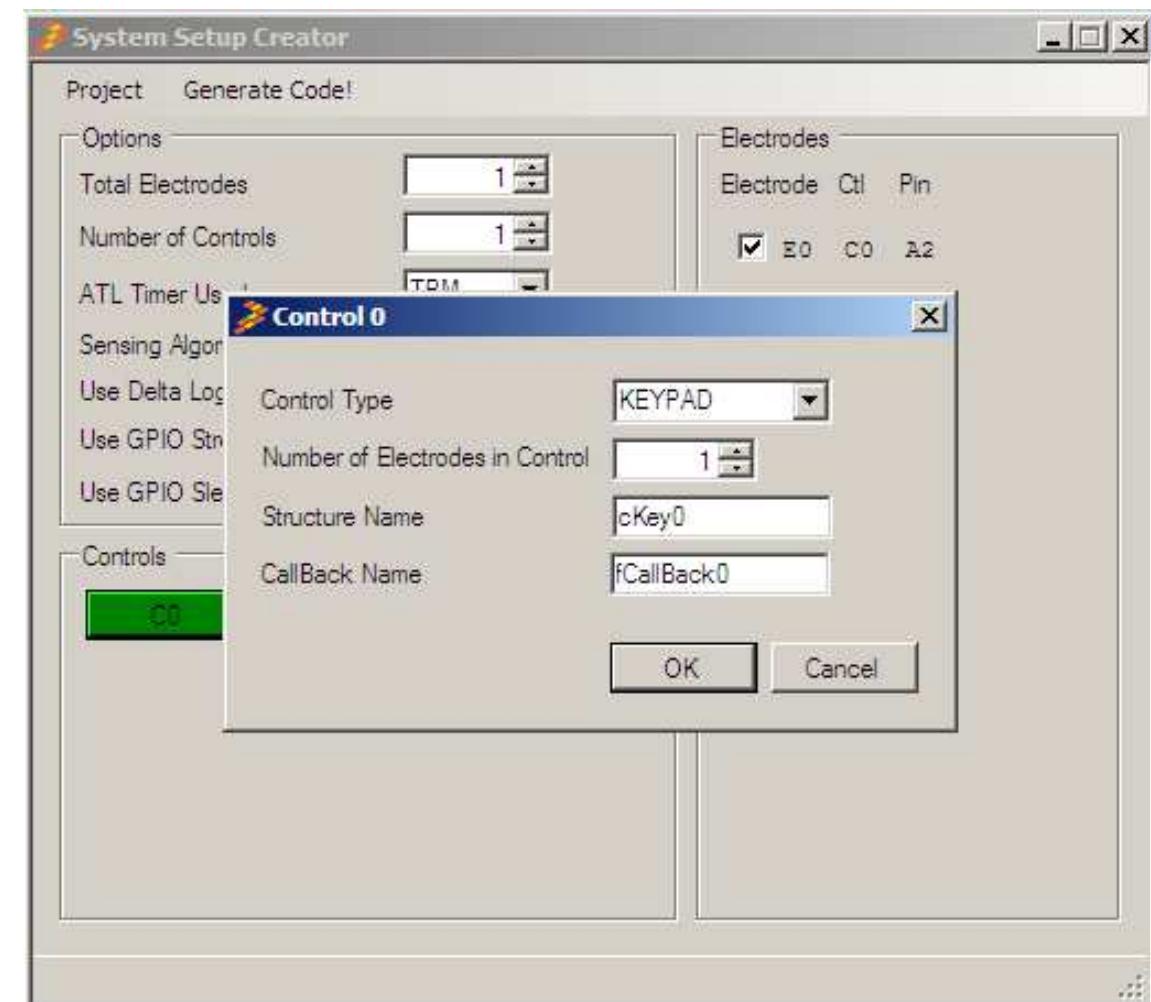
► Electrode Should now appear next to E0.

► Repeat as Needed for more electrodes



Configure Number of Controls

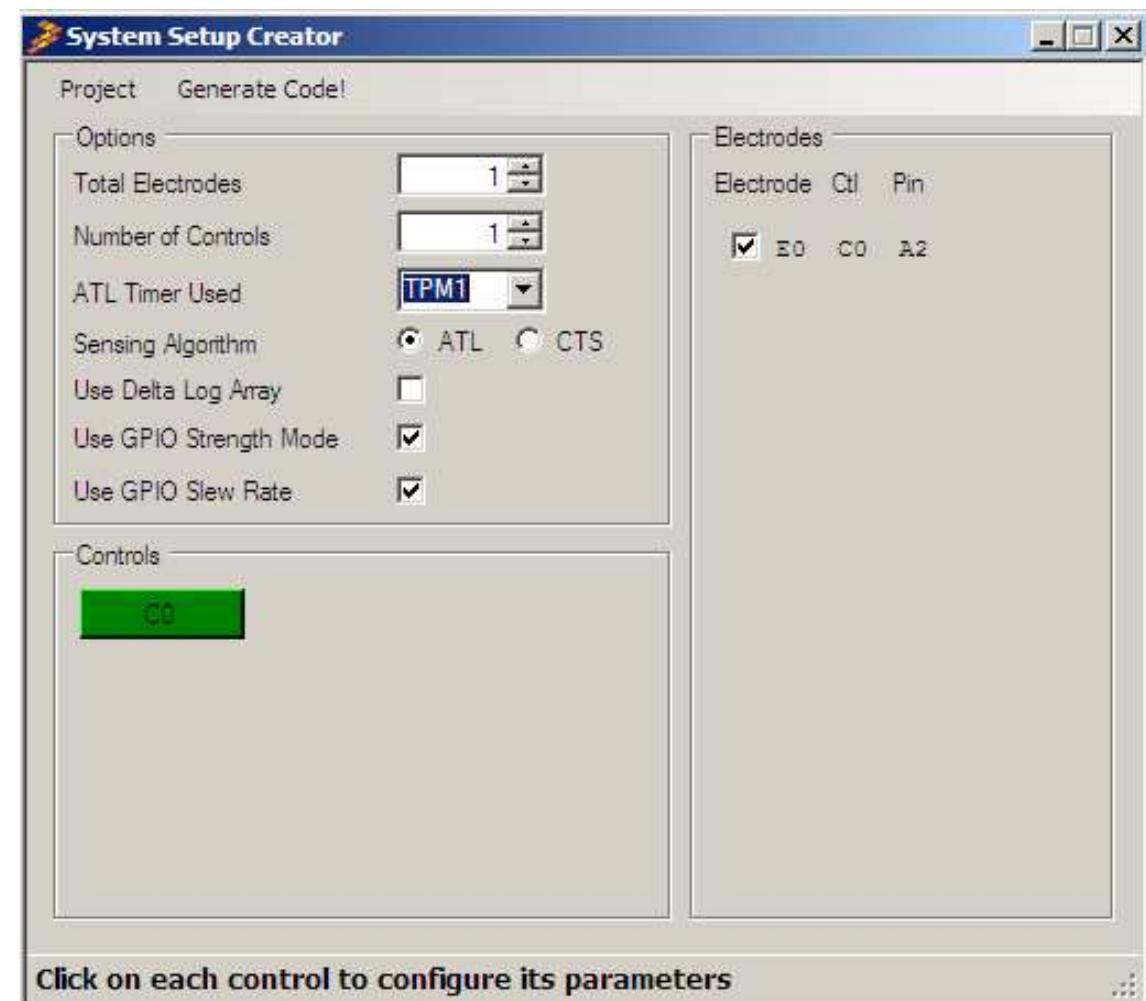
- ▶ Set Number of Controls to 1
- ▶ Click On C0
- ▶ Click OK



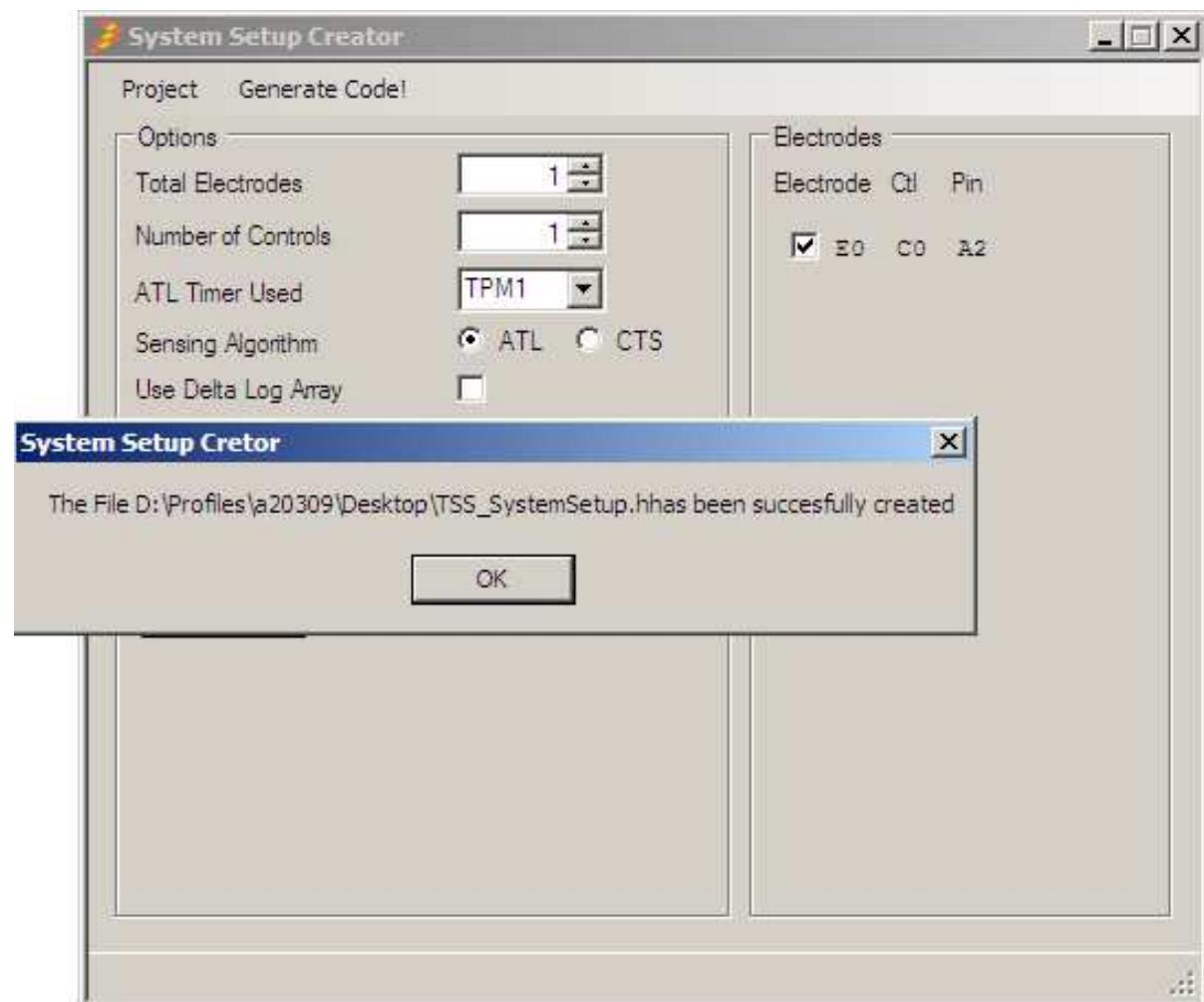
Sensing Algorithm

► We will use ATL Library

► We will use TPM1 as the Timer for the Library



- ▶ Click on Generate Code!
- ▶ Select Folder to Save to.
- ▶ Click OK



► TSS_SystemSetup.h file

```
#undef TSS_USE_CTS_LOW_LEVEL
#define TSS_USE_ATL_LOW_LEVEL      1

#define TSS_USE_DELTA_LOG          1
#define TSS_USE_GPIO_STRENGTH      0
#define TSS_USE_GPIO_SLEW_RATE     0

#define TSS_N_ELECTRODES    1      /* Number of electrodes present in the system */

/* Electrode's GPIOs configuration */

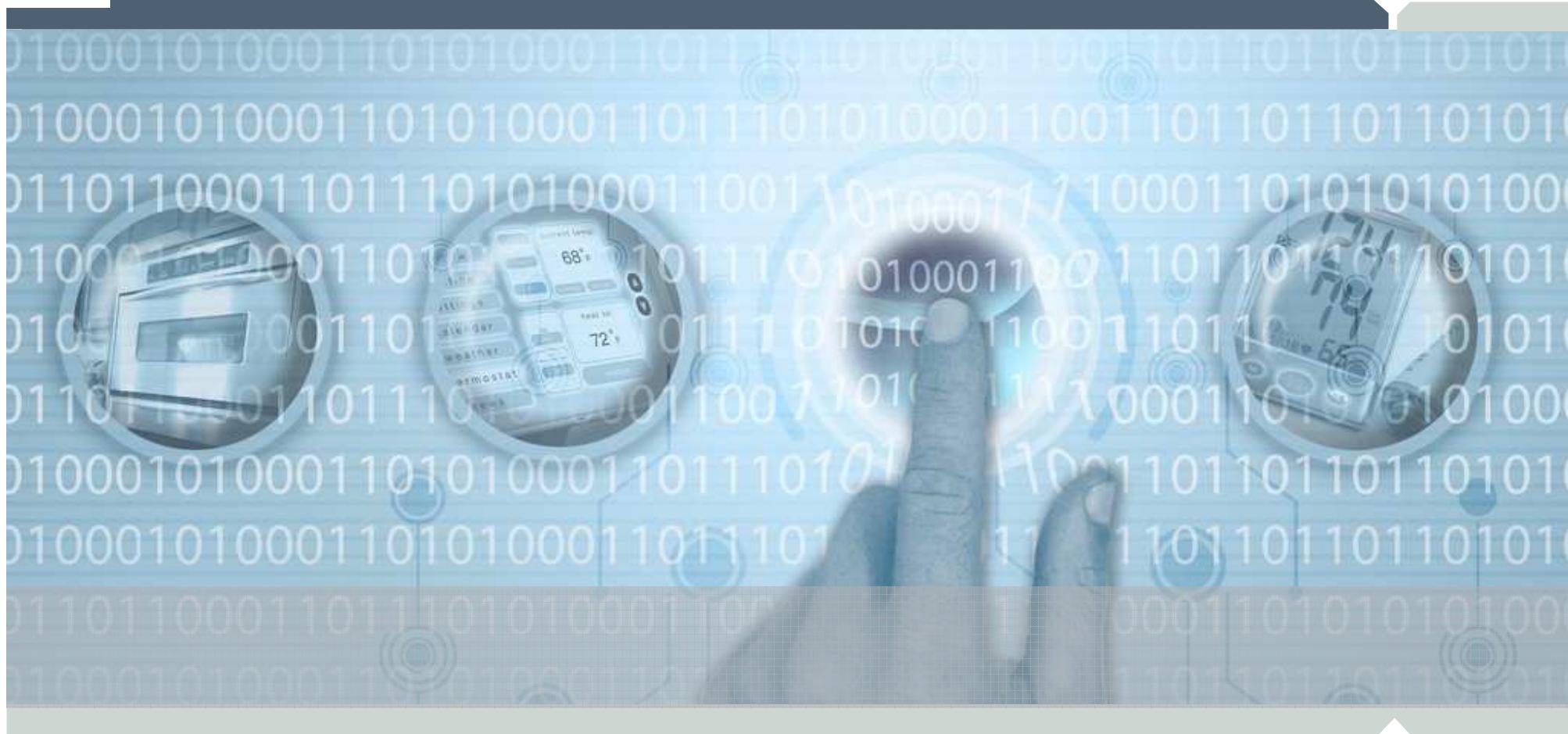
#define TSS_E0_P            A      /* Electrode port */
#define TSS_E0_B            2      /* Electrode bit */

/* Controls configuration */

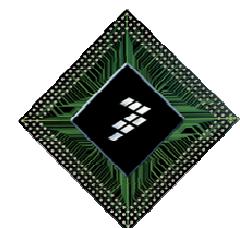
#define TSS_N_CONTROLS      1

#define TSS_C0_TYPE          TSS_CT_KEYPAD /* Control type */
#define TSS_C0_ELECTRODES    1      /* Number of electrodes in the control */
#define TSS_C0_STRUCTURE     cKey0   /* Name of the C&S struct to create */
#define TSS_C0_CALLBACK       fCallBack0 /* Identifier of the user's callback */

#define ATL_HW_TIMER         TPM1    /* Defines the TSS Timer */
```



Example API Initialization



Configuring the TSS application interface

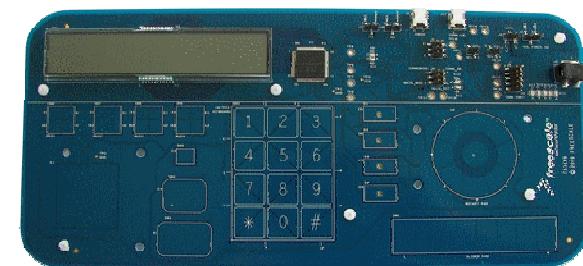
► Add the TSS Task Function to the Main Loop

```
void main(void) {  
  
    MCU_Init();                                /* Initializes MCU Peripherals */  
    (void)TSS_Init();      /* Initializes the Touch Sense Library */  
  
    /* Sets the Sensitivity value for each Electrode */  
    (void)TSS_SetSystemConfig(System_Sensitivity_Register,0x32);  
  
    /* Configure Touch Event */  
    (void)TSS_KeypadConfig(cKey0.ControlId,Keypad_Events_Register, TSS_KEYPAD_TOUCH_EVENT_EN_MASK);  
  
    /* Enables the control and enables the callback function */  
    (void)TSS_KeypadConfig(cKey0.ControlId,Keypad_ControlConfig_Register,  
                          TSS_KEYPAD_CONTROL_EN_MASK|TSS_KEYPAD_CALLBACK_EN_MASK);  
  
    /* Enables the TSS */  
    (void)TSS_SetSystemConfig(System_SystemConfig_Register,TSS_SYSTEM_EN_MASK);  
  
    EnableInterrupts;          /* Enable interrupts */  
    for(;;) {  
  
        TSS_Task();           /* TSS main function */  
  
        __RESET_WATCHDOG(); /* feeds the dog */  
    } /* loop forever */  
    /* please make sure that you never leave main */  
}
```

Touch Sensing Evaluation Board & Development Tools

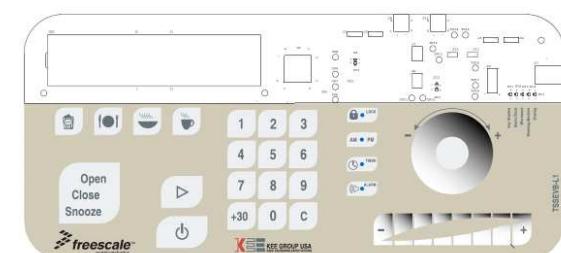
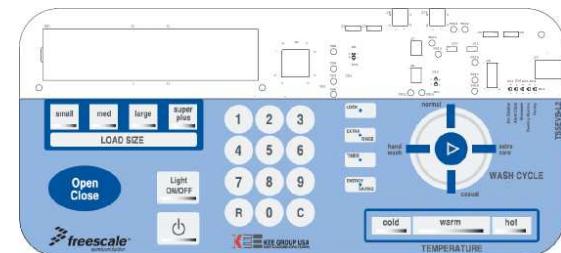
► TSSEVB – Full evaluation and development (*Resale Price: ~\$79*)

- Four demo applications included
- Rotary, slider, keypad, matrix, multiplexed, and backlit
- LCD glass and buzzer
- Serial to USB converter and on board BDM
- Powered over USB or external power supply
- Different thickness overlays

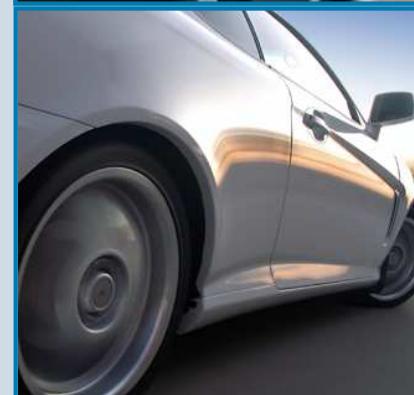


► Learn Quicker, Develop Faster – right out of the box

- In box DVD enhances the experience including:
 - Touch Sensing Software Suite
 - CodeWarrior 6.2 for microcontrollers
 - Access to complimentary GUIs
 - Out of the box walkthroughs – get up and running in minutes
 - Application examples and application notes



- Strong and broad portfolio in sensors and microcontroller solutions with more than 30 years of innovation
- Portfolio balance for consumer, industrial and automotive with our cost effective touch sensing solutions
- Higher integration from the smallest sensor in the industry (2 x 2 mm) to high end microcontrollers
- IP ownership beyond capacitive touch sensing





Q&A

For more info: www.freescale.com/touchsensing

