



## CAPACITIVE SENSING MADE EASY, Part 1: An Introduction to Different Capacitive Sensing Technologies

By Pushek Madaan and Priyadeep Kaur, Cypress Semiconductor Corp.

Capacitive sensing finds use in all kinds of industrial, automotive, medical, and consumer applications. The popularity of this technology, especially in human interface devices (HID), has grown rapidly due to its ability to reduce manufacturing cost, increase product lifespan by eliminating mechanical components, and enhance product look-and-feel. Track-pads and touch-screens constitute the major applications, but we are also seeing a growing use of this technology in implementing touch-buttons and sliders. Be it mobile phones, TV controls, automotive dashboards, remote controls, or industrial controls, capacitive sensing-based touch buttons and sliders are proving to be much more appealing than mechanical switches and rotary encoders, both in terms of looks and reliability. Not only do these sensors allow a striking user interface but also offer a highly durable and reliable solution, provided they are properly designed and calibrated. Herein lies the concern of the engineers and designers trying to implement solutions based on this technology.

Designing low-cost, responsive, and low-power capacitive sensors for reliable operation in noisy environments is considered to be a tough job by many engineers. What refutes this myth is that application development with such sensors can indeed be a matter of few days, including prototyping through validation for production requirements. This article discusses various methodologies, guidelines and know-how related to all development phases, from design to production of capacitive sensing-based keypads, all of which contribute to reducing time-to-market by a significant amount.

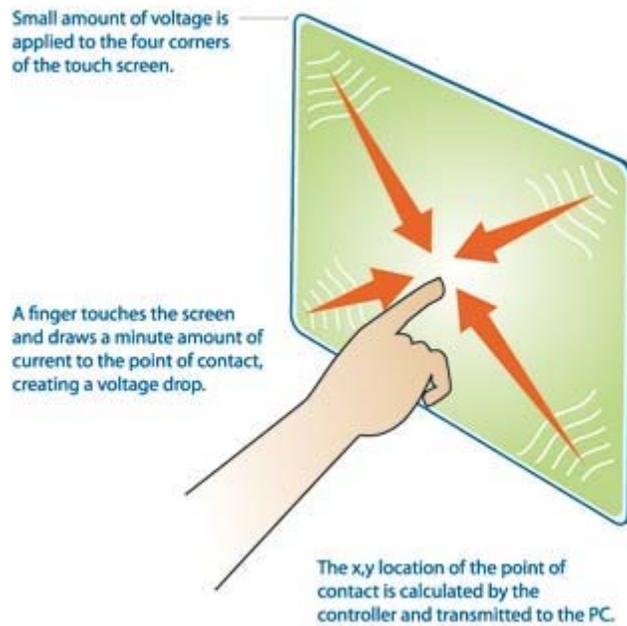
### **Capacitive Sensing, an Overview:**

A capacitive touch sensor is based on one of the following capacitance types:

1. Surface capacitance
2. Projected Capacitance: These are subdivided again into two types:
  - o Self capacitance
  - o Mutual capacitance

### **Surface Capacitance:**

In this technology, glass is uniformly coated with a conductive layer. During operation, a voltage signal is applied to all four corners of the panel, resulting in a uniform electrostatic field. When a human finger touches the panel, it forms a capacitance where one plate is the conductive layer and the other being the human finger. Depending upon the location of the finger touch, current drawn from the four corners will be different and thus the capacitance seen by those corners will also be different. This difference can be used to determine the exact location of the touch (see Figure 1).

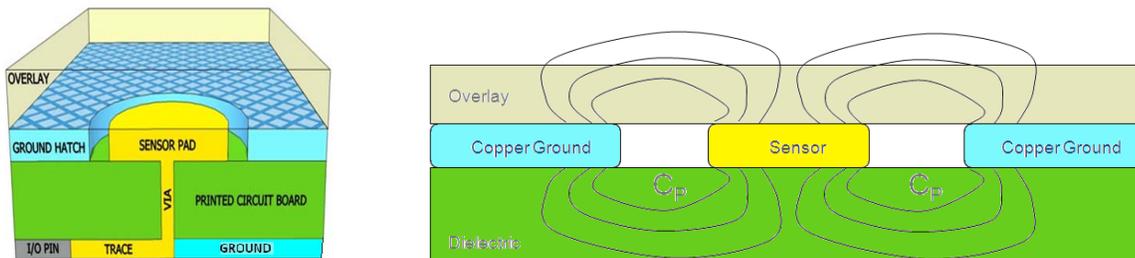


**Figure 1. Surface Capacitance**

This technique brings in all the advantages of capacitive touch technology as discussed above but is prone to false detection and requires special calibration during manufacturing.

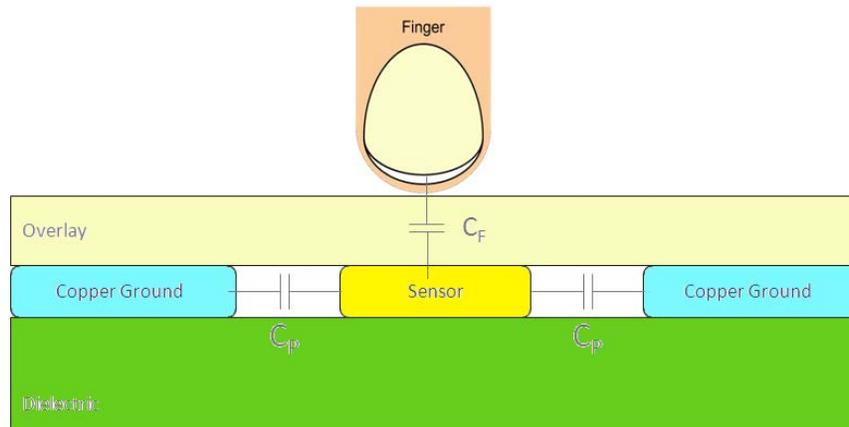
**Self Capacitance:**

Here, each capacitive sensor is a conductive pad laid on a PCB, surrounded by a ground pattern (see Figure 2).



**Figure 2. Sensor pad on a PCB**

This sensor forms a parasitic capacitance  $C_p$  with the surrounding ground pattern, and the electric field lines can be seen in the area above the sensor. When a conductor like a finger enters the area above this sensor, it alters the electric field lines and effectively adds a finger capacitance  $C_f$  to the sensor (see Figure 3).



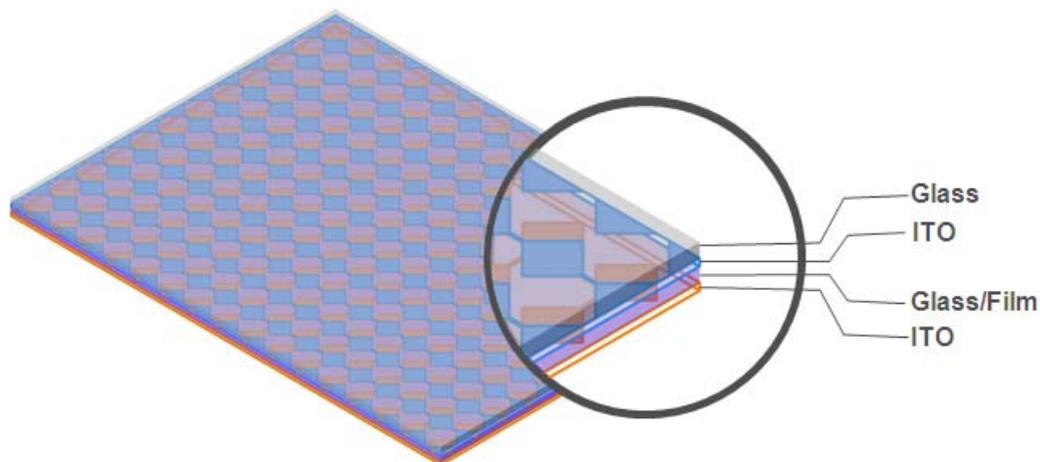
**Figure 3. Finger capacitance added to the sensor**

This results in an increase in capacitance of the sensor from  $C_p$  to  $C_p + C_f$ .

By continuously measuring the capacitance of the sensor(s) in the system and looking for a sudden change in capacitance, a microcontroller can determine when the finger was placed on the sensor. Here, the absolute value, or the parasitic capacitance of the sensor does not matter. The microcontroller just looks for a sudden change in capacitance and if this change is above a particular threshold, a finger presence is reported.

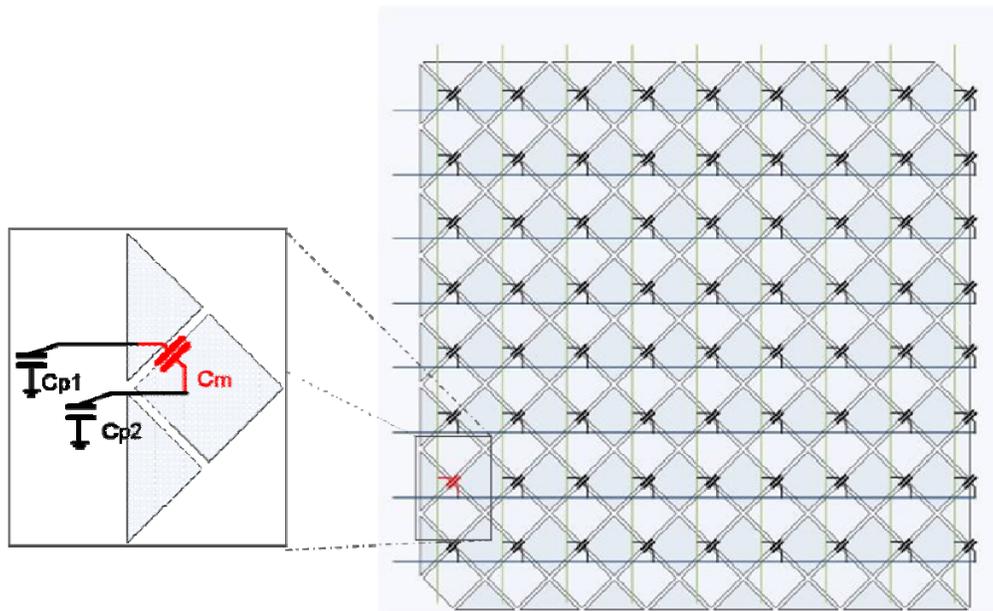
***Mutual Capacitance:***

This is the latest addition to the catalogue of capacitive sensing techniques. Such capacitive panels have two conductive layers stacked together with a very thin separation (see Figure 4).



**Figure 4. Dual layer ITO - Mutual Cap**

In this technology, when a finger touches the panel, the mutual capacitance between the row and column is reduced. This reduction in capacitance is used to identify the presence of a finger.



**Figure 5. Mutual capacitance between Rx-Tx**

As every intersection has its own mutual capacitance and can be independently tracked (see Figure 5), this method provides a distinct advantage for detecting multiple fingers.

Which method is right for your application:

**Table 1. Sensing technologies comparison**

Method	Linearity	Accuracy	Size Scalability	Optical Clarity	Damage Resistant	Multitouch
Infrared	★★★★★	★★★	★★★★★	★★★★★	★★★	Yes (expensive)
Surface Acoustic Wave (SAW)	★★★★	★★★★	★★	★★★	★★★★★	No
Surface Capacitance	★★	★★	★★	★★★★★	★★★★★	No
Resistive	★★★★	★★★★	★★★★	★★	★	Yes (expensive)
Projected Capacitance	★★★★★	★★★★	★★★	★★★★★	★★★★★	Yes

All the technologies mentioned in the table 1 have their merits and demerits for touchscreen-based applications. However, when it comes to implementing buttons or slider for a front panel where two buttons are separated by a significant distance, then the only economical solution is projected capacitance. This is because it works with PCB traces; all other technologies require that a special panel has to be designed.

For this reason, this article concentrates on the use of self capacitance.

### Measuring the capacitance change:

Measuring a capacitance is straightforward. Traditional methods are based on measuring the charging time of the capacitance or the resonating frequency of an RC circuit (as is done in LCR meters). However, these methods cannot be used directly for touch sensing because the change in capacitance of such a sensor would generally be in the range of a few tenths of a pF. If we try to measure the charging time/change in the resonating frequency of such a capacitor using a constant current, there would be three concerns:

1. Measuring such a small capacitance would require a high frequency clock and/or an accurate low current value for an optimum measurement.
2. While measuring charging time, a greater part of the time would be spent in measuring the self (parasitic) capacitance of the sensor (which is generally in the order of tens of pF) and the required measurement of the change would constitute only a fraction of the measurement time, thus leading to excessive use of controller time and power.
3. The effect of noise on such a system would be high.

So, how do we ensure less noise with a lower measurement time? The answer is to incorporate an integrating effect. This is similar to averaging the effect of noise on an ADC input (see Figure 6).

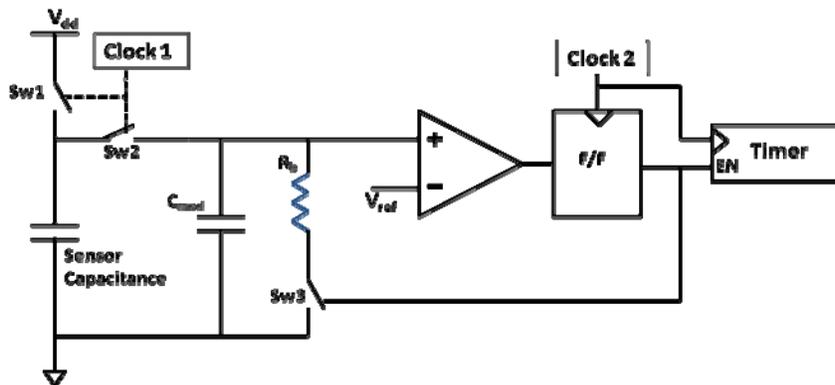


Figure 6. Sigma delta circuit for touch sensing

Here, the sensor capacitance is continuously switched between two voltage levels using two non-overlapping clocks driving Sw1 and Sw2 respectively. This emulates a resistance of  $R=1/fC_{\text{sensor}}$  where  $f$  is the switching frequency of clock1 (see Figure 7).

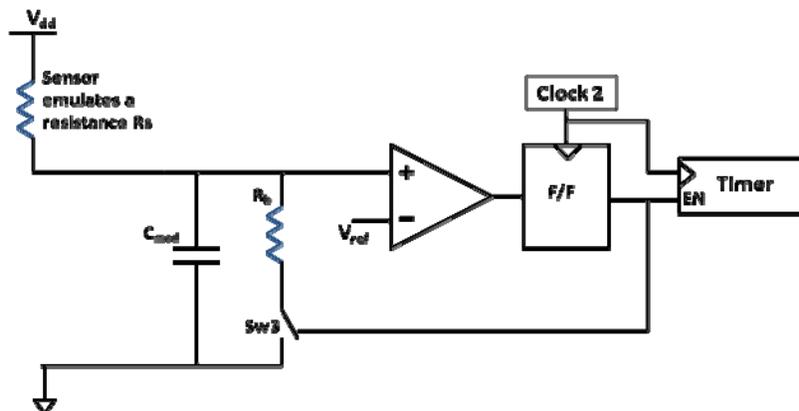


Figure 7. Sigma delta circuit with sensor capacitance replaced by equivalent resistance

The output of the flip flop in the above circuit is a density modulated bit stream, the duty cycle of which increases with a decrease in resistance. This bit stream goes to the enable signal of a timer, which measures the amount of time for which the F/F output is high during a given time T (decided by the timer resolution and clock frequency).

Bringing a finger close to the sensor increases  $C_{\text{sensor}}$  and hence decreases  $R_s$ . This leads to an increase in the average duty cycle and hence the timer output value. Note that any noise on the input of the comparator is now integrated over n cycles of clock (here n is the timer resolution) and so the effect of noise is lower in such implementations.

### **Reliability considerations:**

It is important to quantify various signal and noise parameters associated with any system in order to evaluate its performance and reliability. While reporting a finger press, it becomes crucial that only a valid finger press should be reported i.e. a noisy environment should not cause spurious detections.

Any unwanted change in the signal, whether from devices operating in its vicinity or environmental changes, can be termed as noise. When measuring a change in capacitance of the order of femto Farads, a small amount of noise can also lead to spurious detections. Before going into the techniques of how to build a system where the effect of noise is minimal, it is important to understand the possible sources of noise.

### **Different types of noise:**

A capacitive sensing system is mainly susceptible to noise generated from the following three sources:

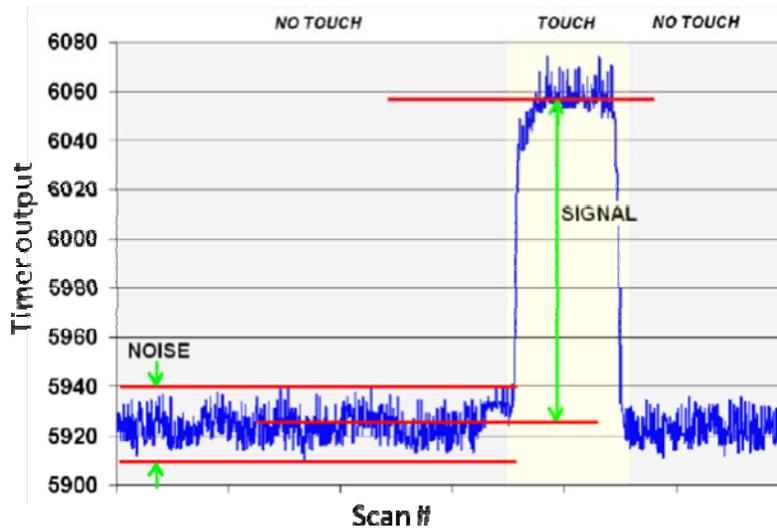
1. **Radiated noise** – Any operating circuitry radiates energy that can potentially create problems with the operation of other circuits in its vicinity. Capacitive sensing buttons constitute only the user interface part of any system, and generally there is much circuitry sitting behind the user interface. This nearby circuitry can also radiate noise if not properly designed. Sources of noise can be the LCD (Liquid Crystal Display), switching power supplies, mobile phone, Wi-Fi radio, etc.
2. **Conducted noise** – A noisy power supply is the most common source of conducted noise. The increasing demand of low cost implementations forces developers to use less expensive supplies which in turn generate more noise. This can adversely affect the operation of the sensor. A human body touching the sensor can also couple a 50/60Hz common mode noise into the system.
3. **Environmental changes** – Changes in environmental parameters such as humidity, temperature, and device aging also change the capacitance of the sensor. Such unwanted changes can also be termed as noise.

So how do we ensure that noise will not cause spurious detections? If we define a signal-to-noise ratio for a particular system and ensure that the noise in the system is much less than the required signal, we can ensure spurious free detections. Here is how we define signal and noise in such a system:

**Signal:** Signal can be defined as the change introduced by the finger in the timer output value.

**Noise:** When a finger is not present, the timer output will vary by a small amount for each scan because of various noise sources. The consolidated effect of such noise can thus be measured by monitoring the peak-to-peak variance in the timer output value for a number of scans.

Figure 8 illustrates SNR:



**Figure 8. Signal and noise for the demonstration of SNR**

It might look a little tricky, but maintaining a good SNR is not very tough. There are multiple techniques for achieving a good SNR, which are as follows:

1. **Tuning (Auto or Manual)** – Calibrate the device during the design phase to ensure that it exhibits a minimum of 5:1 SNR for fail-safe operation. Using some software overhead, this manual tuning effort can be switched to auto-tuning wherein the device calibrates itself in the field to ensure that it achieves the minimum SNR required. Cypress' SmartSense solution is an example of such an innovative technique.
2. **Auto correction** – Gradual changes in capacitance because of temperature, humidity, or component aging can be compensated by monitoring the counts obtained (digital representation of capacitance) in firmware and updating the reference signal with the gradual change observed. Note that in the case of auto-calibrating solutions, it is also possible to recalibrate the system if the gradual change in counts exceeds a particular threshold.
3. **Layout** – A proper schematic and PCB layout is address all the problems mentioned above.
4. **Filters** – Software filters used to process the digital counts obtained can also be used to improve the SNR. The use of filters increases response time but improves SNR dramatically. Depending upon system requirements like response time and power consumption, the use of software filters may be feasible.

In the next part of this article, we will explore all of these design techniques for improving SNR along with sensor patterning (i.e., buttons/wheels/touch-pads) for different types of applications.



Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709  
Phone: 408-943-2600  
Fax: 408-943-4730  
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™, Programmable System-on-Chip™, and PSoC Express™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and/or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.