DESIGNING A CAPACITIVE SENSING SYSTEM FOR A SPECIFIC APPLICATION

By Shruti H Applications Engineer, and Subbarao Lanka, Applications Engineer Senior, Cypress Semiconductor Corp.

Capacitive sensing buttons are emerging as a popular interface alternative to mechanical switches in many consumer electronics and white goods applications. However, designing a capacitive sensing interface comes with its own challenges that introduce new product development, production, and quality control concerns. For example, the parasitic capacitance ($C_P$) of capacitive sensing buttons can vary from board to board and also can vary due to environmental changes like change in temperature and humidity. Noise also varies from system to system. Another common issue with user interface (UI) design is design portability. If the UI design of a TV front panel changes, for example, the design will require sensor retuning to accommodate changes in layout, sensor dimensions, etc. The laborious process of tuning adds cost to a system in terms of manpower and time, as well as delays the product’s time to market. This article focuses on different hurdles faced when designing a capacitive sensing interface and methods developers can use to overcome typical problems to make a design robust and portable.

**Capacitive Sensing Basics**

Figure 1 shows how a capacitive sensor board looks.

![Figure 1: Cross sectional view of capacitive sensing board](image)

The sensor capacitance with respect to circuit ground in the absence of a finger is as shown in Figure 2 (a). This capacitance is called parasitic capacitance ($C_P$). When a finger touches the sensor (or comes near the sensor), it introduces another capacitance called finger capacitance ($C_F$) in parallel to the $C_P$. In the presence of a finger, the total sensor capacitance ($C_X$) is given by equation 1.

$$C_X = C_P + C_F$$ – Equation 1
The change in capacitance introduced by $C_F$ is used to detect a finger press.

**Capacitance Measuring System**

An electronic system measures the sensor capacitance by converting the capacitance into a digital value. Figure 3 shows the block diagram of a capacitive sensing preprocessing circuit. (Note: There can be numerous different methods to measure capacitance).
This system uses a switched-capacitor circuit on the front end of the system to convert the sensor capacitance to an equivalent resistor, as shown in Equation 2 and Figure 4. A Sigma-Delta modulator converts the current measured through the resistor into a digital count. When a finger is on the sensor, the capacitance increases and the equivalent resistance decreases. This causes an increase in the current through the resistor, resulting in an increase in the digital count.

This method requires one external component, CMOD. This is an Integration capacitor that gets charged through a constant current source (IDAC) and discharged through the equivalent resistor.

$$R_{EQ} = \frac{1}{F_S C_X} \quad \text{Equation 2}$$

Where $F_S$ is the switching frequency of the switched capacitor block.

The current is measured with the help of a comparator whose output bit stream (shown in Figure 3) is fed to a counter for a fixed amount of time. This counter value (the digital count) gives us an indication of the magnitude of $C_X$. Let us measure the counter value as raw counts. The amount of time for which the counter counts determines the response time.

The fixed time duration of the counter is called as scan time. Raw counts are compared with a reference level to decide the sensor ON and OFF status; this reference level is called baseline. When there is a finger touch, the raw count increases. In order for the sensor to be ON, this increase in the raw count must be greater than a threshold called the finger threshold. If the change in raw counts is below this threshold, then it is considered to be noise and that threshold is called the noise threshold (see Figure 5).
Now that we have understood the basics of capacitive sensing, let's discuss why capacitive sensing is needed and what challenges are faced in implementing them.
Comparison between capacitive sensing and mechanical button user interface

Figure 6. Mechanical buttons and Capacitive sensing buttons on TV front panel

Capacitive sensing brings elegance, easy-to-use touch sensing functionality to a user interface. Capacitive touch sensor shave already replaced billions of mechanical buttons. Capacitive sensing not only gives sleek look to front panels but also eliminates wear and tear problems associated with mechanical buttons. In TV/monitor applications, capacitive sensing is being widely adopted because it eliminates tooling costs and adds to the aesthetics.

The table below lists some of the problems associated with mechanical buttons and the advantages of capacitive sensing buttons over mechanical switches.

<table>
<thead>
<tr>
<th>Mechanical buttons</th>
<th>Capacitive sensing buttons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can wear out and get stuck</td>
<td>Highly reliable for industrial designs</td>
</tr>
<tr>
<td>Collect dirt</td>
<td>Easy to clean surface</td>
</tr>
<tr>
<td>Have bulky aesthetics</td>
<td>Sleek, thinner designs. Enables advanced UI features like proximity</td>
</tr>
<tr>
<td>High tooling cost</td>
<td>No tooling cost</td>
</tr>
</tbody>
</table>
Design flow – Capacitive sensing user interface

Figure 7 below provides the typical design flow for implementing capacitive sensing.

Figure 7. Capacitive sensing interface design flow

Firmware (F/W) Development, Tuning, and Production Fine Tuning are the critical phases in the life cycle of a capacitive sensing User Interface design.

1. F/W Development:

   In a broader sense, firmware implements the functionality required for the specific application; i.e., number of buttons, additional features like PWM, ADC, DAC, etc. From a capacitive sensing standpoint, the firmware does the job of scanning sensors (i.e. measuring the sensor capacitance) as well as other associated functions like processing feedback based on the sensor ON/OFF status. For systems implementing only capacitive sensing, devices with configurable options are available. Registers are configured through serial communication protocols (like I2C) for specific sensor functions and no firmware development is required. Implementing capacitive sensing using a programmable device provides flexibility to meet varying user needs as well as perform sensor scanning and processing.

2. Tuning:

   Tuning is the process of determining the optimum values for set of capacitive sensing parameters for robust and reliable performance under various environmental conditions and for different mechanical constructions of the interface. This demands a thorough understanding of how a capacitive sensing system behaves under various conditions.

   The key things to be considered while tuning are
a) SNR (Signal to Noise Ratio) of sensor
b) Sensor scan time
c) Finger threshold settings

SNR of sensor:

![Signal and Noise Graph](image)

**Figure 8. Signal and Noise**

One of the main goals of tuning a capacitive sensing system is to reliably discriminate between TOUCH and NO TOUCH sensor states. In an SNR calculation, the signal is the change in the sensor response when a finger is placed on the sensor. Noise is the peak-to-peak variation in sensor response when a finger is not present. For reliable capacitive sensing performance, signal strength needs to be significantly larger than noise; the general recommendation is that the signal should be at least five times the noise for a minimum recommended SNR of 5:1.

Sensor scan time:

Sensor Scan time is the amount of time the counter counts, as described in the capacitance measuring system section above. Shorter sensor scan times lead to lower SNR. Higher scan times lead to delayed response time and higher power consumption. Thus, based on the sensor parasitic capacitance \(C_P\), the sensor scan time needs to be optimized for SNR, response time, and power consumption.

Finger Threshold setting:

The Finger Threshold is set to indicate a finger touch. This Finger threshold should be set carefully to avoid false triggering because of noise and atmospheric changes. The general recommendation is that the finger threshold should be set to 75% of the signal strength for reliable touch detection as shown in Figure 5.
As can be seen from Figure 9, tuning is a time consuming, laborious, and repetitive process that has to be repeated whenever the PCB or overlay is changed during development.

3. Production Fine Tuning:

Capacitive sensing performance depends on the physical properties of the capacitive sensors and environmental conditions. The parasitic capacitance for a sensor varies when there is vendor change, process variation, or variation in environment such as humidity or temperature. This requires fine tuning through statistical analysis on samples during production in order to minimize yield loss due to failure. As we can see, there are many steps and issues which one needs to address before releasing a design for mass production.

While designing a capacitive sensing system for a particular application like a TV or monitor, we encounter typical challenges like PCB vendor change and noise affecting the capacitive sensing performance which often lead to retuning. Some methods to deal with such challenges and reduce the tuning effort are:

a) Auto-tuning
b) Layout considerations
Auto-tuning

An innovative method is now available in the industry where the device dynamically tunes itself (monitors and sets parameters automatically) by monitoring changes in noise and the environmental conditions of the system. This method also enables the device to initialize all the capacitive sensing related parameters at power up based on the environmental conditions and the mechanical design of the system.

**Baselining:**

Common environmental changes that affect the capacitive sensing measurement are temperature and humidity drift. Temperature/Humidity drift causes changes in capacitive measuring circuit components/parameters and also affects $C_p$ and $C_f$, causing the raw counts to change. A typical variation of raw count with temperature is shown in Figure 10.

![Raw count variation with temperature](figure)

If a constant reference is used to detect a button touch, the temperature/humidity drift may result in a false button press or a missed button press.

Baseline compensation as part of an autotuning sequence adjusts the reference level (baseline) and the noise thresholds automatically so that low frequency noise is kept below the threshold levels to avoid false triggers.

**Optimal parameter settings:**

Based on a sensor’s physical properties and environmental noise, sensor parameters should be assigned with optimal values at power up to make the capacitive sensing system work reliably.

As mentioned before, it is critical to maintain the SNR above 5:1 in order for the sensors to work consistently. The first thing that an auto tuning algorithm needs to do is compute and optimize values of specific device parameters such that the SNR is maintained above 5:1. Optimal scan speed is set in order not to increase power consumption and response time. Threshold levels for noise and finger response are optimized and set at power up.

Let us understand by taking most common scenarios how auto tuning is beneficial.

**PCB process variation:**

A single board manufacturer may have different manufacturing sites which leads to slight variations in the manufacturing process that changes the parasitic capacitance of sensors from board to board. This leads to retuning if the yield loss because of this change is too high. For example, in Figure 11, one of the sensors of Board #4 does not respond to finger press since the sensor has a high $C_p$ due to a PCB process variation.
PCB Vendor change:

OEMs typically have many PCB/FPC manufacturing sources qualified in order to protect against manufacturing cost increases and capacity shortage. Each board manufacturer may use different PCB materials which may have different parasitic capacitance on sensors, resulting in lower yield. For example, if tuning is finalized for a board manufactured from a vendor X, the same tuning parameters may not be fully applicable to a board manufactured from vendor Y, resulting in lower yield.

Figure 11. Performance of boards subjected to process variations

In Figure 11, the performance of boards subjected to process variations is shown. The figure illustrates how different PCB vendors can affect the performance of capacitive sensors. The graph shows the response of the sensors on different boards when a finger is pressed on them. The performance varies depending on the PCB vendor, indicating the need for auto-tuning algorithms to adjust for these variations.

An auto tuning algorithm at power up that automatically determines the parasitic capacitance of the sensors eliminates both of the above mentioned problems.

Overlay dielectric / thickness change:

In Figure 12, the sensors on Board #3 and Board #4 do not respond to a finger press since the sensor has high $C_P$ due to the PCB manufacturer being different.

An auto tuning algorithm at power up that automatically determines the parasitic capacitance of the sensors eliminates both of the above mentioned problems.

Figure 12. Performance of boards from different vendors

The figure shows the performance of boards from different vendors, illustrating how changes in the overlay dielectric or thickness can affect the sensor response. The graph displays the sensor response on different boards, highlighting the need for auto-tuning algorithms to adjust for these changes.
Sensitivity is directly proportional to the dielectric constant of the overlay material and inversely proportional to overlay thickness. This means that if a design tuned for 2mm glass (dielectric constant ($\varepsilon$) 8.0) is changed to a 2mm plastic (dielectric constant($\varepsilon$) 2.8), the design need to be retuned to work effectively. Similarly, if a 3mm plastic overlay is changed to a 3.5mm plastic overlay, the tuning has to be done all over again.

Figure 13. Performance of boards with different overlay thickness

An auto tuning algorithm adjusts the sensitivity when there is a change in overlay thickness such that the sensors work as expected even when there is overlay thickness variation.

The capability to do all of the auto tuning functions at run time as well as set the parameters automatically to provide a stable SNR of greater than 5:1 is the key to a robust auto tuning technique.

Design portability is another common scenario in TV/monitor applications which requires considerable reworking in terms of tuning when porting a design across different applications having similar requirements.

In TV/monitor applications we usually come across requirements where a single design needs to be used for different models of TV/monitors with varying screen size. Hence, the placement of the touch control panel may vary, which in turn may lead to change in button position. When there is change in button position, there is a possibility that the trace length will vary as well. Two different models can also have two different button size requirements. In addition, different models of TV/monitors may have different requirements for overlay thickness. All of these differences lead to variations in parasitic capacitance.

With the capability of auto tuning to handle changes in parasitic capacitance and overlay thickness, it is possible to achieve seamless porting of the design.
Noise determination - Noise immunity level improvement:

Noise is random fluctuation in an electrical signal. Noise generated by electronic devices varies greatly, as it can be produced by several different effects and sources. Noise can also be described as a summation of unwanted or disturbing energy from natural and man-made sources.

Multiple tuning efforts to compensate for noise are needed, especially if noise levels are in between noise threshold and finger thresholds or if the noise level keeps varying in a system. Auto-tuning must determine noise levels in the system dynamically and adjust the noise and finger threshold accordingly.

Layout

Another factor which needs to be taken into account is layout. Optimizing layout is critical to ensure reduced tuning efforts based on interactions with multiple system level noise conditions.

Another method for avoiding noise entering a capacitive sensing system is at the hardware level. Let us understand what types of noises affect capacitive sensing system in a TV/monitor application and how they can be avoided during designing the layout.

Capacitive systems in TV/Monitors are subjected to the following types of noises caused due to the peripheral circuits in the system.

a) Switching power supply noises or conducted interference
b) LCD inverter noise
c) Radiated interference

Switching power supply noises or conducted interference:

Conducted noise is the noise current often generated by high frequency switching circuits entering the system through the power and communication lines. The following guidelines will help in preventing conducted noise from entering a capacitive sensing system:

- Provide GND and VDD planes that reduce current loops.
If the capacitive sensing controller PCB is connected to the power supply by a cable, minimize the cable length and consider using a shielded cable.

Place a ferrite bead around power supply or communication lines to help reduce high frequency noise.

**LCD Inverter noise:**

In TV/Monitor applications, care must be taken care to prevent noise from LCD inverters from upsetting the capacitive sensing system. LCD inverters induce a lot of noise into a capacitive sensing system and reduce SNR drastically.

A simple technique to minimize LCD inverter noise is to partition the system with noise sources from capacitive sensing inputs, as demonstrated in Figure 15. Due to the practical limitations of product size, the noise source and the capacitive sensing circuitry may only be separated by a few inches. This small separation, however, can provide the extra margin required for good sensor performance compared to with close proximity between noise sources and capacitive sensing circuitry.

![Figure 15. Separating noise source from capacitive sensing interface](image)

**Radiated Interference:**

Radiated electrical energy can influence system measurements and potentially influence the operation of the capacitive sensing processor core. This interference enters the capacitive sensing chip at the PCB level, through the sensor traces, and via other digital and analog inputs.

The following methods can be used to suppress the radiated interference on capacitive sensing system:

- In general, providing a ground plane on the PCB helps to reduce the RF noise picked up by the capacitive sensing controller.

- Every capacitive sensing controller pin has some parasitic capacitance, $C_p$, associated with it. Adding an external resistor forms a low pass RC filter that can dampen RF noise amplitude. Series resistors should be placed within 10 mm of the capacitive sensing controller pins.
Long traces can pick up more noise than short traces. Long traces also add to $C_p$. Hence trace length should be kept to a minimum.

In the article, we have spoken about how a capacitive sensing system works, why capacitive sensors are fast replacing mechanical buttons, and what it takes to simplify capacitive sensing system design. Some of the common challenges encountered in design a capacitive sensing user interface and techniques to overcome these problems have been described. Using the auto tuning techniques discussed in the article, we can build a robust capacitive sensing system while significantly easing design complexity. Auto tuning itself has various characteristics; the two main criteria being compensation during run time (for noise and environment) and automatic parameter setting which are critical in easing design time and thus reducing the time to market of capacitive sensing system.