Implement glove touch in capacitive touch user interfaces

Joshan Abraham, Vibheesh Bharathan, Cypress Semiconductor - June 08, 2015

Capacitive touch-sensing user interfaces are replacing mechanical buttons in products across consumer, medical and industrial segments. However, with the advent of touch-sensing user interfaces, end-users are demanding advanced features such as haptics support and glove touch to mimic mechanical button usage, as well as additional features such as stylus support and proximity sensing. These features improve the overall user experience of the product and offer manufacturers the opportunity to differentiate themselves. This article focuses on one of these features - glove touch, which is increasingly finding use in consumer, industrial and medical spaces. For example, the touch interface on a wearable smart band should work properly even when the user has gloves on because of cold weather conditions, or medical laboratory equipment should work properly even when touched with latex gloves.

However, implementing glove touch on capacitive touch interfaces is not easy, and most implementations tend to offer unreliable and inconsistent performance. This article focuses on the challenges in implementing glove touch on capacitive touch buttons and how these challenges can be overcome to design a robust and reliable touch-sensing interface with glove touch capability.

There are two primary challenges to implementing reliable glove touch, they are:

- Detecting low signals produced by a gloved hand
- Ignoring false touches from a finger hovering above the sensors

Why glove touch produces a low signal

Capacitive touch sensing works on the principle that a finger introduces a change to the capacitance of a sensor when the finger touches the overlay covering the sensor. This change in capacitance is measured and converted to the digital domain (A to D conversion) by a touch-sensing controller. When the measured value exceeds a pre-defined threshold, a touch is registered.

The change in digitized capacitance due to a finger touch is known as signal and the unintentional change in digitized capacitance without a finger touch is known as noise. A signal-to-noise ratio (SNR) of 5:1 is recommended for a reliable touch-sensing system. Figure 1 shows how capacitance is measured in a touch-sensing system.

In simple terms, capacitance introduced by a finger can be viewed as a parallel plate capacitor, where the finger and the sensor are the two conductive plates and the overlay is the dielectric medium between the plates. The finger-introduced change in capacitance is proportional to factors such as the size of the sensor and the finger (i.e. area of plates) and the dielectric constant of the overlay material; and inversely proportional to the thickness of the overlay on top of the sensor (i.e.
the distance between the plates). A thicker overlay increases the distance of separation between plates, thus producing a smaller change in capacitance. This leads to a lower signal-to-noise ratio.

Wearing a glove on a finger adds a new overlay proportional to the glove thickness on top of the existing overlay increasing the overall overlay thickness. This decreases the strength of the signal below the pre-defined threshold and a touch with a gloved hand is typically not detected. This is the reason why most users have to remove their gloves to effectively touch a button on a capacitive touch-sensing user interface.

**Unwanted hover and false touches**

A touch sensor can be tuned to work with thicker overlays by increasing its sensitivity. Similarly, a touch sensor can be tuned to detect a touch, even when touched by a gloved hand. Increasing the sensitivity of a sensor means that it requires a smaller change in capacitance to detect a touch.

However, the problem here is that it produces a condition known as “unwanted hover”, where a bare finger in close proximity to the sensor (hovering above the sensor) produces an equivalent capacitance change as introduced by a glove touch. An erroneous touch could be registered as a glove touch even though the finger neither touched the sensor nor did it have a glove on. This condition is mostly undesirable and can adversely affect the user experience of the product. Figure 2 indicates the signal produced by a glove touch, finger touch and a hovering finger.
A designer hence faces the following problem: A system tuned for regular touch-sensing doesn’t pick up touches from a gloved hand and a system tuned for glove touch produces false touches due to “unwanted hover”.

An easy, non-elegant solution would be for the design to add a user-triggered interrupt or physical switch to indicate if they are wearing a glove or not. This diminishes the user experience, especially in consumer products that need to have “one action less” and in medical products which need to work the same in all conditions.

**Improving signal strength**

There are three important design parameters that need to be considered to improve the signal strength of glove touches:

**Sensitivity**
Sensitivity is a measure of the ability of a capacitive touch-sensing circuit to produce a signal - a more sensitive circuit produces a larger signal. Sensitivity is typically measured in counts per capacitance. In the context of capacitive touch-sensing, the magnitude of change introduced by a touch is in the order of 100s of femto-farads (fF). A touch by a gloved finger typically introduces a capacitance of 100fF. A circuit with a sensitivity of 500 counts/pF can produce 50 counts of signal for a 100fF touch, while a circuit with sensitivity of 50 counts / pF can only produce 5 counts of signal for the same touch. Therefore, a circuit with a higher sensitivity can detect a glove touch more reliably. **Parasitic**

**Capacitance**
Parasitic capacitance is the intrinsic capacitance of the sensor, which is introduced due to its proximity to other conductive objects. The change in capacitance due to a touch interaction, the signal, is perceived relative to the parasitic capacitance of the sensor. Higher the ratio between the change in capacitance and the parasitic capacitance, the higher the sensitivity the sensor can be tuned for. **Avoiding Challenges**
For example, a capacitance touch-sensing circuit that uses a 12-bit A-to-D converter can have a maximum output of 4096. A sensor having a parasitic capacitance of 16pF can be tuned to achieve a maximum sensitivity of 256 counts/pF, beyond which the A-to-D converter would get saturated. However by decreasing the parasitic capacitance to 8pF, the sensor can be tuned to a maximum sensitivity of 512 counts/pF. A change of 100fF capacitance due to a touch, would produce a signal of approximately 25 counts in the first case and 50 counts in the second case.

The parasitic capacitance of a sensor is dominated by design specifications of sensor stack up and layout, such as trace thickness, distance between traces and distance between PCB layers. A careful sensor layout design and sensor stack-up is required to maintain low parasitic capacitance.

To improve performance and to provide flexibility to the designer, some touch-sensing controllers integrate the following two features to reduce the impact of excess parasitic capacitance on sensitivity:

- Faux differential measurement capability.
- Support for shield electrodes.

Faux differential measurement capability: A typical touch-sensing controller measures capacitance from 0 to a maximum measurable value (e.g. from 0pF to 8pF). A touch-sensing controller that can implement faux-differential measurement (i.e. faux differential A-to-D conversion) can be set to measure a specific range of capacitance (e.g. from 8pF to 16pF) and achieve a higher sensitivity. Using this method, a capacitance touch-sensing circuit with a 12-bit A-to-D converter, can be tuned to achieve a sensitivity of 512 counts/pF, even with a sensor having a parasitic capacitance of 16pF.

Support for shield electrodes: Shielding the sensor from other conductive objects around the sensors will minimize the extra capacitance added and therefore minimize the parasitic capacitance of the sensor.

Controllers supporting both the faux differential measurement capability and shield electrodes, typically double the sensitivity and subsequently better glove touch performance.

**Noise**

Wherever there is a signal, there is also noise. The signal is the change in capacitance that results in a meaningful change in output. Noise, on the other hand, is any disturbance that does not change the capacitance but does change the output. Noise that exceeds a threshold could potentially produce false touches. In general, a signal-to-noise ratio of 5:1 is required for a reliable touch-sensing system. This means that along with having a high sensitivity, the controller must maintain low noise. In other words, a controller tuned to 500 counts / pF sensitivity must limit the noise below 10 counts to maintain a 5:1 SNR for a touch of 100 fF.

More often than not, noise is introduced into a system through conduction effects (such as power supply switching noise or electrical fast transient (EFT) currents) or through coupling effects (such as radiated noise from a cell phone or crosstalk between signal traces). In general, capacitive sensors and the controller must be isolated from noise sources, such as a switching power supply. Careful system and PCB design are key to maintaining the isolation and avoiding noise from entering the capacitive sensing system.

**Avoid “unwanted hover” and false touches**

This section details design methods to reduce "unwanted hover" in touch-sensing systems that
support glove touch.

*Use dedicated thresholds*

The amplitude of a glove touch signal is significantly smaller than that of a regular finger touch. Using dedicated thresholds, along with firmware design logic, can help to detect and differentiate between finger touch and glove touch and thereby improve the hover rejection performance.

Two dedicated thresholds can be set for finger touch and glove touch signals (FThreshold and GThreshold in Figure 3 below). Typically, these thresholds are set at 80% of a typical finger touch or glove touch signal.

![Diagram of glove, finger, and hovering finger signals](Figure 3-Dedicated thresholds for glove and finger touch signals)

When a user first touches the sensor, the firmware identifies if the signal is above the finger threshold or only above the glove threshold. If the signal is above the finger threshold, it assumes that the user does not have gloves on and it discards all signals below the finger threshold for a predefined amount of time (e.g. 30 sec) from the detection of the last touch. This ensures that a hovering finger is not detected as a false glove touch. The firmware decision tree is shown in Figure 4.

It is reasonable to assume that it would take the user at least 30 seconds to put on a glove and re-touch the sensor. However, if the first touch produces a signal that crosses the glove threshold but not the finger threshold, the system assumes that the user has gloves on and will continue to detect glove touches. While in this mode if a user removes the glove and touches the sensor, the signal will cross the finger threshold and the system will immediately move into a mode that will only detect finger touches.
A typical touch-sensing user interface panel consists of multiple sensors. It is possible to improve the firmware decision logic of the panel such that it looks for signals on all sensors, and if a finger touch is detected on any of the sensors, all of the sensors can be made to reject glove touches for predefined amounts of time.

The primary disadvantage of this method is that if the first signal detected is that of a hovering finger, it may cause a false touch.

**Use touchscreen inputs**

Some products, such as mobile phones, printers or high-end home appliances, have an independently controlled touchscreen as well as touch buttons on their user interface (UI) panels as shown in Figure 5. In such systems, intercommunication between the respective controllers can be helpful to efficiently manage glove and finger touches.
Touchscreen controllers can efficiently differentiate between a hovering finger and glove touch due to the nature of a touchscreen's sensor construction. A gloved finger, which covers a larger area than a bare finger, produces low amplitude signals on a larger number of neighboring sensing nodes, whereas a hovering finger produces low amplitude signal only on fewer neighboring sensing nodes as shown in the Figure 6 below. The touchscreen controller uses the difference in the signal pattern to identify if a user is wearing gloves or not.

**Tying it together**

If a glove touch is detected by the touchscreen, the information is conveyed to the capacitive button controller that controls the buttons. Generally, the host controller has communication interfaces between both controllers, and can manage the information exchange between them; avoiding the need for additional interfaces.

This method is obviously not suitable for systems without a touchscreen. Additionally, this method assumes that the first touch happens on the touchscreen and not on the buttons. Otherwise a false touch could be registered for the first touch on the buttons, similar to the false touches when using the “dedicated threshold” method.

**Use split-sensor design**

In general, capacitive buttons are constructed using a single sensor that can detect the present or
absence of a conductive object. A split sensor design is an innovative-patented solution that overcomes the disadvantages of the previously described methods.

The image on the left of Figure 7 shows a typical capacitive sensor layout with an optional hole at the center for LED back lighting. The image on the right shows the split sensor design, in which the button touch area is split into two dedicated sensors (inner and outer).

![Typical Sensor and Split-sensor Designs](image)

The basic principle behind this method is that different touches will produce unique signal patterns on the inner and outer sensors. These unique signal patterns can be interpreted in firmware to differentiate between a finger and glove touch. Figure 8 shows the typical signal profiles on both the sensors. The three use cases are:

- A glove touch overlaps with both the inner and outer sensors, and produces low amplitude signals on both the inner and outer sensors.
- A finger touch overlaps with both the inner and outer sensors, and produces high amplitude signals on both the inner and outer sensors. A hovering finger, which has a convex shape and is smaller than a glove, produces a higher signal on the inner sensor and a comparatively lower signal on the outer sensor.
The firmware decision tree for the split-sensor design is shown in Figure 9.
The split sensor design is the most reliable method to implement glove touch on your touch-sensing user interfaces.

**Conclusion**

As glove touch becomes a common feature in applications, end-product users expect the performance to be consistently reliable. This article highlighted key design aspects to consider and methods to use to implement reliable glove touch. The most important step in ensuring your user interface panels "just works" is to choose the right touch-sensing solution. You need a proven solution that offers a consistently high SNR, high sensitivity, allows excess parasitic capacitance compensation such as faux differential measurement capability and shield electrodes, and is supported by detailed documentation on design and layout guidelines.

For example, the Cypress’ PSoC® 4 portfolio supports an SNR>100:1, faux differential measurement capability using an integrated constant current source (IDAC) and shield electrodes. It is suited for implementing advanced features such as reliable glove touch detection.

**Resources**

Cypress' [CapSense Design Guide](#) provides design best practices and guidelines to maintain a low parasitic capacitance for the sensors in your design.

Cypress' [Getting Started with CapSense](#) details best practices and design guidelines for ensuring minimal noise enters the system.