Efficient tuning of capacitive-sensing designs

Robert Jania, Cypress Semiconductor

A fundamental aspect of any capacitive sensing design is the calibration of the sensor. Calibration eliminates the sensor providing false actives in the system and as well as the sensor failing when it is supposed to be active. Until recently, individual sensor also had to be accompanied by discrete components comprised of resistors and capacitors in order to adjust the sensitivity of each particular sensor.

With this method, engineers had to solder down resistors and capacitors to the board, test out the circuit with these values, and then determine if the values are acceptable or if new values are needed. If new values are needed, the process begins anew with re-soldering of new resistors and capacitors of different values down and testing of the system again.

In some cases, it might be necessary to use two capacitors in parallel to achieve the desired value to obtain the desired key sensitivity balance. Designers also have to take into account the fact that long trace lengths and ground plans add to sensor capacitance, resulting in a poor signal to noise ratio (SNR) and lowered performance. Adding series resistors and load capacitors could sometimes address this issue.

Such manual “guess-and-check” calibration is very time consuming and quickly erodes the cost-effectiveness of using capacitive sensors. Fortunately, imbalances in sensors from sensor to sensor on a board can typically be compensated by adjusting the discrete capacitors on a per-sensor basis.

Calibrating sensors through firmware

Programmable devices have been gaining popularity over the past few years, enabling firmware-based calibration of sensors as an alternative to the hardware guess-and-check method. The ability to dynamically program calibration enables engineer to adjust the sensitivity of a button by changing values inside the device’s firmware. This approach still entails some guess work but in a much less time-consuming fashion. Parameters can be programmed into the sensor, either by adjust firmware values and recompiling code for downloading, or by using an adjustable test-bench. A process that used to take days can be completed in hours.

When running capacitive sensing in a programmable IC, the firmware in the application program can scan each sensor, measuring the capacitance. This value, often referred to as the raw count value, is stored in the device. The device also tracks a baseline for the raw counts. The baseline value of each sensor is an average raw count level computed periodically by the device.

The use of an updated baseline is important for ensuring that a system is able to adapt to drift in the system over time due to temperature and other environmental effects. The current ON/OFF state of the sensor is determined using the raw counts

and baseline delta. This allows the performance of the system to remain constant even though the baseline may drift over time. Figure 1 shows the transfer function between difference counts and button state that is implemented in firmware.

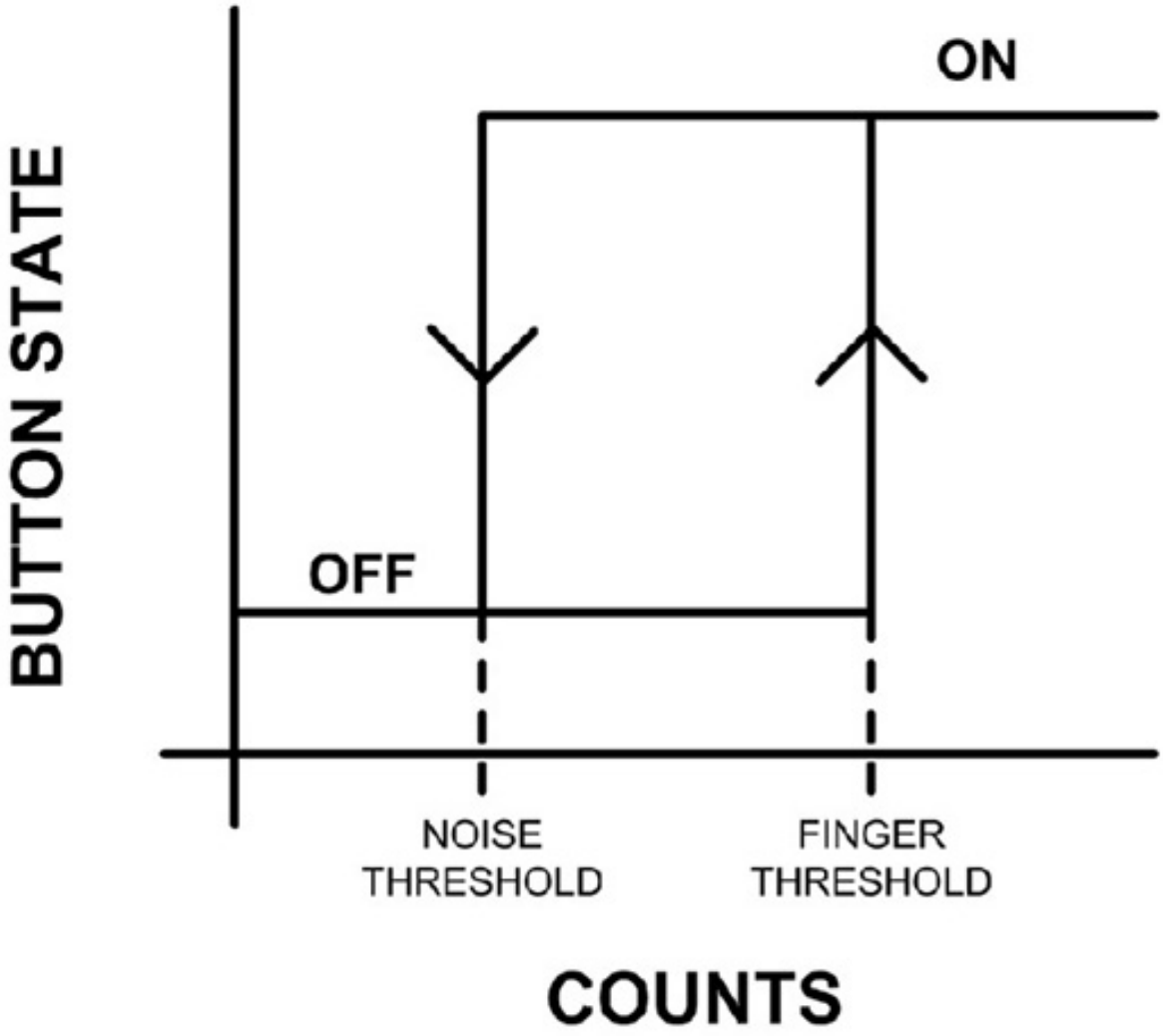

Figure 1. Transfer function between difference counts and button state

The hysteresis in this transfer function provides clean transitions between ON and OFF states even though the counts are noisy. This provides a debounce function for the sensors. Setting of the threshold levels determines the overall performance of the system and can reflect the tolerance of the system for false actives and sensing failures. With very thick overlays, the signal-to-noise ratio is low. Setting the threshold levels in this kind of system, however, can be challenging.

Pre-defined firmware development tools can facilitate the design of capacitive sensors by providing device interconnects, I/O drive modes, and APIs for both capacitive sensor operation and calibration. Reference code provides not only a starting point for basic functionality but can be used as a learning tool leading to customization and optimization for a particular project. Developers can use standard debugging tools instrumented to reduce the guess-and-check work of calibration by providing visual feedback in real-time.

Calibrating sensors through software

More recently, software has become a feasible way to calibrate capacitive sensors.

These programs integrate the math behind the sensing method, essentially hiding it from the user. These tools provide high- and low-level application programming interfaces (APIs) that work together with the data being collected and stored in the IC, removing several steps when compared to sensor calibration through firmware. APIs include scanning the sensors, setting the resolution, setting clock signals and timing, controlling logical outputs, and sleeping/waking the device.

These software tools control what otherwise would have been written in a main.c file (or its equivalent), thus reducing the amount of code that needs to be written. Rather than controlling the sensors directly, developers use a visual interface to watch the status of sensors. Calibration changes can be made and the results seen immediately. The software takes care of all the APIs in the background to wake the device, scan the sensors, and put the device back to sleep. It also takes care of waking the device back up again at the proper interval (see Figure 2).

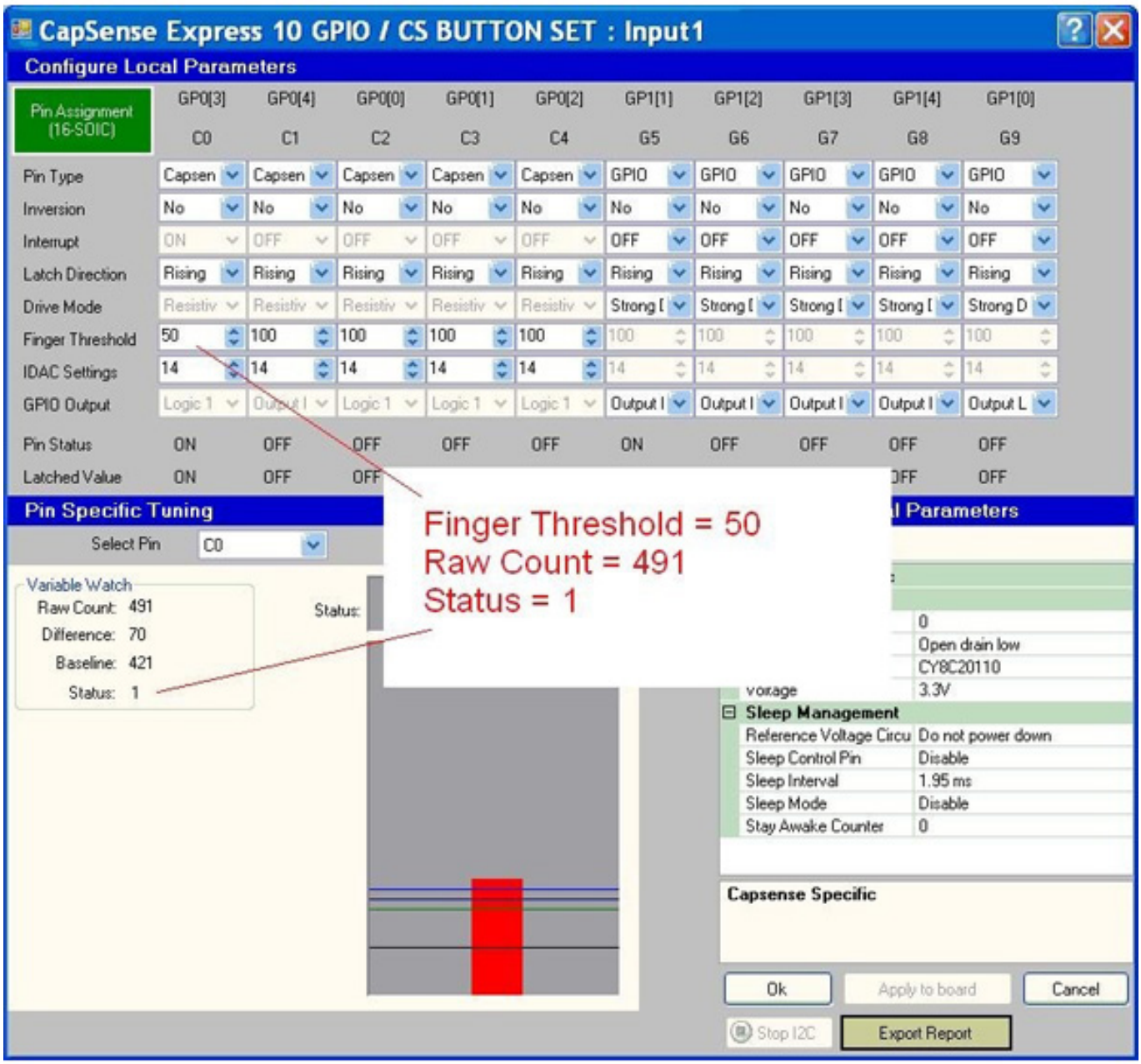

Figure 2. Calibrating a capacitive sensor through software

Visual software design tools will help continue to usher in this new era of capacitive sensing input devices. OEMs will be able to benefit from using capacitive sensors for button replacement, especially with higher-end equipment, leading to costs savings from a PCB perspective, the need for fewer components, and higher reliability on the long-term through elimination of mechanical button failure. With new development tools, the time to implement a capacitive sensing input can be just as fast, if not faster, than working with mechanical parts.

Source URL (retrieved on *03/12/2014 - 12:15am*):
http://www.ecnmag.com/articles/2012/06/efficient-tuning-capacitive-sensing-designs?qt-most_popular=0