Application Report
SLAA574–January 2013

Capacitive Touch Sensing, MSP430™ Button Gate Time Optimization and Tuning Guide

MCU Touch Solutions

ABSTRACT

MSP430™ microcontroller based capacitive touch buttons can offer increased performance when properly optimized and tuned for their specific application. Performance benefits that result from button optimization can include, but are not limited to, decreased power consumption, improved response time, and the ability to grow a design to include more buttons. This application report provides the reader with a starting point for button design at the system and software level. It answers questions such as the following: What measurement method (RO or fRO) should I use? How long do I need to scan each button for? What kind of power performance can I expect? A how-to guide for button gate time selection and tuning is also included in this report.

This report assumes the reader has a basic understanding of the TI Capacitive Touch Sense Library (CAPSENSELIBRARY). It may be helpful to reference the *TI Capacitive Touch Sense Library Programmer's Guide* (SLAA490) with this document. For information on how to optimize and tune capacitive sliders and wheels, see the *MSP430™ Capacitive Touch Slider and Wheel Tuning User Guide* (SLAA575).

Code examples and other associated files can be downloaded from http://www.ti.com/lit/zip/slaa574.

Contents

List of Figures

List of Tables

1 Introduction

Oscillator-based capacitive sensing with a microcontroller requires that each touch sensor (electrode) in a given system be scanned for a period of time. This time is referred to as the electrode's gate time. The gate time of an electrode affects several performance characteristics of the capacitive touch system, such as electrode sensitivity, noise immunity, response time, and power consumption.

1.1 Purpose

This application report provides an overview of how capacitive touch button gate time affects the performance of that button and the capacitive touch system as a whole. It demonstrates what kind of performance can generally be expected from different gate times in different applications. In addition, the differences between the RO and fRO measurement methods as well as basic instructions on how to set gate times in the *TI Capacitive Touch Software Library* (CAPSENSELIBRARY) are discussed along the way. After reading this document, the reader will be able to estimate what kind of capacitive touch button gate times are achievable for a given system, as well as what kind of average power consumption can be expected from that system.

1.2 Scope of Application Report

This report only discusses gate time as it relates to the design of buttons; it does not examine implications on sliders, wheels, or proximity sensors. Sliders and wheels impose additional restrictions and design considerations upon sensor gate times, as the gate time will often be based upon the required positional resolution of the slider or wheel. Likewise, proximity sensor scan times are dependent upon the desired proximity distance that the designer would like for the sensor, and are often quite large when compared with buttons. Buttons only require the determination of a touch versus a no-touch condition. Therefore, the gate time for a button should be as small as possible while still providing enough information to reliably and repeatedly determine the difference between a touch and a no touch in the operating conditions that are specified for the system.

1.3 Capacitive Sensor Performance Metrics

Several key metrics define the performance of a capacitive touch sensor. From a hardware and firmware perspective, designing a capacitive touch sensor involves sacrificing one performance metric for another. The effect of sensor gate time on each metric is described in this section.

Any reference to "the system" in this document is a reference to the entire capacitive touch system, including the capacitive touch electrodes, electrode connections to the microcontroller (MCU), any required discrete components, any overlay material, and the MCU itself.

1.3.1 Power Consumption

The average current draw of a given system is directly related to the gate time of the sensors in that system. Reducing the gate time of a sensor allows the system to be in ultra-low-power sleep modes more often. Other factors that contribute to power consumption include but are not limited to the number of sensors in the system, the scan rate of the system, operating voltage, and operating frequency.

1.3.2 Response Time

Reducing the gate time of a sensor reduces the active duty cycle of the system. This can allow for a decrease in average power consumption, an increase in the number of keys that can be scanned, or an increase in the scan rate of the system (a decrease in the response time). Note that increasing the number of keys or the scan rate increases the average power consumption of the system.

1.3.3 Measurement Resolution

A capacitive measurement's resolution is defined as the number of measurement counts per unit of capacitance. Resolution is equivalent to sensitivity, or the ability to resolve large versus small changes in capacitance. As the gate time of a sensor is reduced, the resolution is also reduced. For buttons, the resolution need only be great enough to satisfy the safety margin desired. High resolution is not as important for a button as it is for a slider or wheel, where positional information must be extracted.

1.3.4 Signal-to-Noise Ratio (SNR)

A given capacitive sensor's signal-to-noise ratio is defined as the worst case signal strength when the sensor is in detect over the worst case signal noise when the sensor is not in detect. The worst case signal strength is equivalent to the sensor's threshold. The worst case signal noise is defined as the maximum measurement count during no interaction minus the minimum measurement count with no interaction over a 100 measurement sample. It is important to note that the SNR observed in lab testing may not match the SNR observed in an application setting. SNR should always be measured in the actual application while subjected to whatever noise sources the application may have.

$$SNR = \frac{sensor_{threshold}}{(no\_interaction_{max} - no\_interaction_{min})_{ave100}}$$

$$SNR_{db} = 20 \log\left(\frac{sensor_{threshold}}{(no\_interaction_{max} - no\_interaction_{min})_{ave100}}\right) \quad (1)$$

To a point, signal-to-noise remains relatively constant as gate time is reduced. For example, if the gate time of a sensor is halved, the signal accumulation within the gate time is halved but the noise accumulation is also halved. SNR is affected much more by system mechanicals such as overlay size, electrode size, and ground loading than it is by sensor gate time. Eventually, counts of noise in a measurement are reduced down to ±1 count. When the system reaches this point, further reductions in scan time reduce SNR.

Ultimately, it is up to the designer to determine what an acceptable SNR is for a given system. For most applications, to ensure robust touch operation, the SNR should be at least 5:1. An SNR of 15:1 is recommended and is generally very attainable.

1.3.5 Noise Immunity (Conducted and Fast Transient)

Longer gate times enhance immunity to noise events due to the natural averaging that takes place when many oscillations are counted. As the gate time of a sensor is decreased, each oscillation plays a greater role in each measurement. Effects from an electrical fast-transient (EFT) burst as well as sensor movement and clock jitter may be increased.

1.3.6 Electrostatic Discharge (ESD) Immunity

Capacitive sensors should be covered with a non-conductive overlay material. This material serves as a dielectric between the sensor and the user's body, and also provides ESD protection for the device. Typical overlays are 1.5 mm to 3 mm thick. Thicker overlays provide greater ESD protection but reduce sensitivity and SNR and, therefore, require longer gating times or larger electrode sizes.

2 Capacitive Measurement Methods

Capacitive touch systems in general operate on the principal that the introduction of a human finger to an electrode adds a parallel capacitance to earth ground (in parallel with the single ended self-capacitance of the electrode). The electrode is also influenced by the parasitic capacitances resulting from the internal GPIO pin of the MCU and from the capacitance between the electrode's trace and its signal ground. These parasitic capacitances are in series with the free-space coupling to earth ground, and the combination is in parallel to the electrode capacitance. See the *Sensor Design Guide* (SLAA576) for more information regarding parasitic capacitances and their effects.

An electrode is typically a plane of conductive material that the user interacts with through a nonconductive overlay. Figure 1 shows the equivalent circuit model for a typical self-capacitance system.

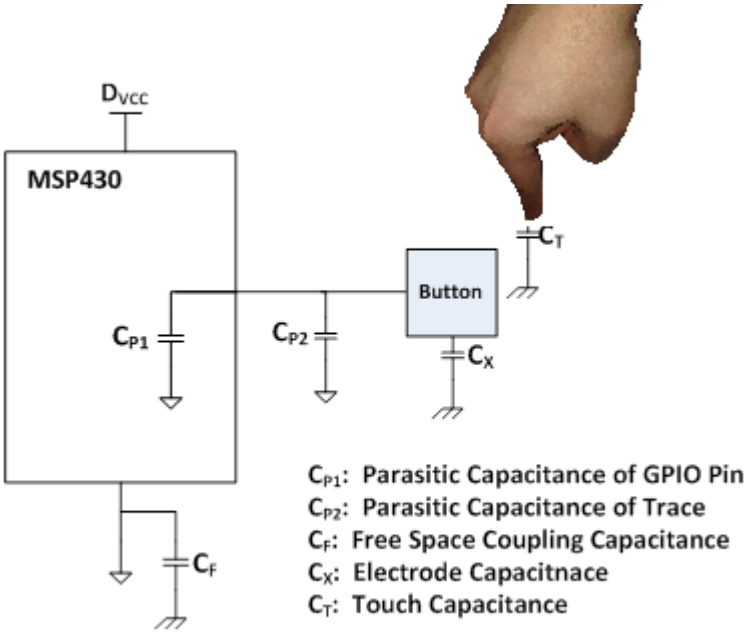

Figure 1. Capacitive Circuit Model – MCU Independent

Texas Instruments microcontrollers (MCUs) achieve capacitive touch sensing by establishing an oscillation on a conductive electrode and measuring the frequency of that oscillation. That electrode has a certain capacitance with respect to earth ground. An RC timing circuit is set up with a resistor and the capacitance inherent to the electrode. The MCU establishes a hysteresis between two potentials, and the circuit charges and discharges at a frequency determined by the resistance, capacitance, and hysteresis settings. This is known as the relaxation oscillator configuration.

If the capacitance increases due to a touch, the oscillation frequency decreases and the MCU sees this change. For most touch applications, it is not the actual capacitance of the electrode that is of interest, but simply the change in capacitance due to a human interaction with the electrode. Figure 2 shows an equivalent circuit of the relaxation oscillator.

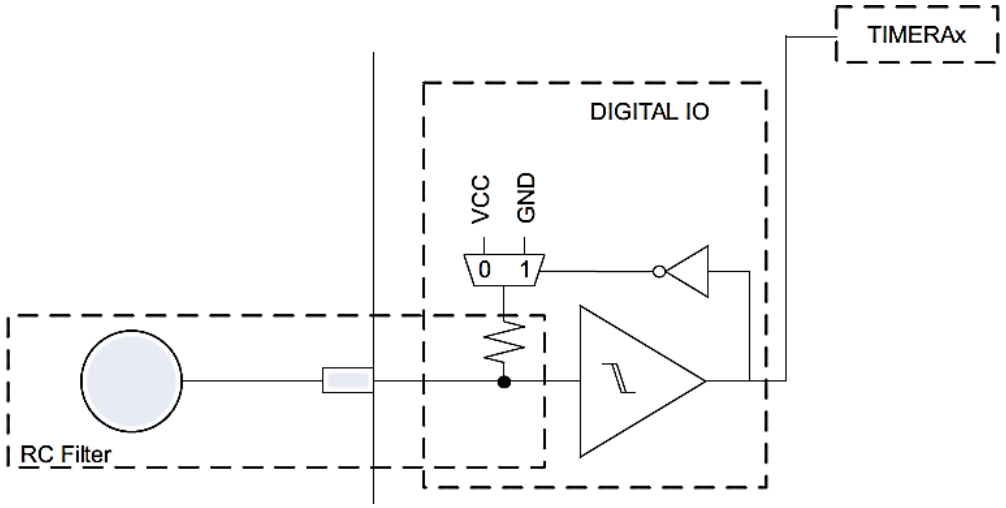

Figure 2. MSP430G2xx Pin Oscillator Schematic

While the relaxation oscillator method is not specific to MSP430 MCUs, this application note discusses only the MSP430 implementations. MSP430 MCUs can use either the built-in PinOsc touch peripheral (on MSP430G2xx2, MSP430G2xx3, MSP430FR58xx, and MSP430FR59xx as shown in Figure 2) or a comparator (Comp_A or Comp_B, which require discrete resistors) to create a relaxation oscillator.

TI supports two measurement methods based upon a relaxation oscillator: the **RO** method (fixed gate time and variable electrode oscillation counts) and the **fRO** method (fixed electrode oscillation counts and variable gate time). Which method is used depends upon the application and the microcontroller selected.

2.1 RO (Standard Relaxation Oscillator) Method

The RO method measures electrode capacitance by using a timer to establish a fixed window of time during which the electrode oscillates (through its relaxation oscillator – see Section 2). A second timer counts the number of oscillations that occur within that fixed gate time.

If a human interacts with the sensor's electrode, the increase in capacitance causes the oscillation frequency to decrease. This results in a lower oscillation count in a fixed time window. Therefore, a touch creates a decrease in counts in a given measurement. Figure 3 shows the principles of the RO method.

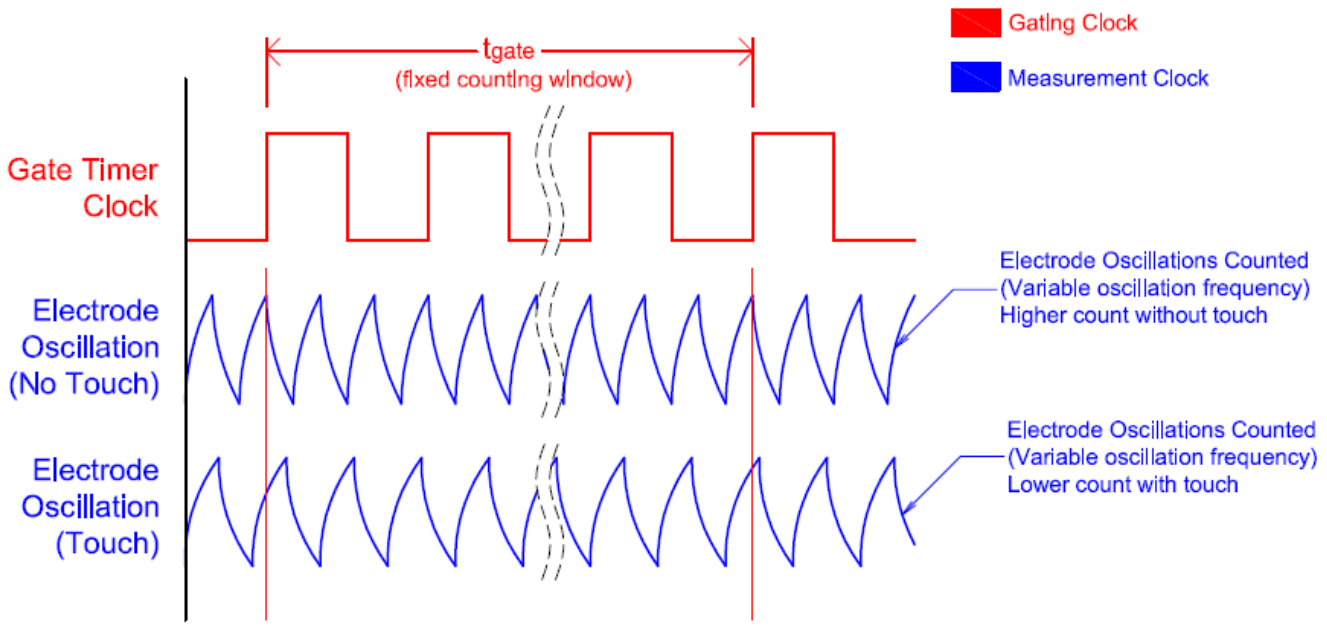

Figure 3. RO Measurement Timing Diagram

2.1.1 Required Peripherals for the RO Method

In addition to the relaxation oscillator peripherals, the RO method requires two timers (a gate timer and an oscillation counter). However, the MSP430 watchdog timer can be used as the gate timer. The watchdog timer can only be configured for certain time intervals (64, 512, 8192, and 32768 cycles). Using the watchdog timer as the gate timer limits the number of selectable gate times but frees a timer for application code management or allows the use of an MCU that only has one standard timer. The Timer_A peripheral is typically used for counting oscillations, and it is internally routed to the internal pin oscillator peripheral on MSP430G2xx2, MSP430G2xx3, MSP430FR58xx, and MSP430FR59xx devices.

2.1.2 Benefits and Drawbacks of the RO Method

Power consumption performance for the RO method is typically on par with that of the fRO method. See Section 5.2 for a more detailed power analysis. RO can have a power consumption advantage if ACLK is used to clock the gate timer instead of SMCLK, as ACLK implementations enter LPM3 during the electrode gate time rather than LPM0 for SMCLK implementations.

Obtainable response times are longer with the RO method than the fRO method for measurements of the same resolution. For example, a typical system using the fRO method with a measurement clock of 12 MHz typically requires one-fifth of the gate time of an RO method implementation for the same resolution measurement. The fixed gating time used in the RO method may be ideal for certain applications in which the capacitive measurement must be a specific length for application scheduling reasons. This makes the RO method less attractive for systems that have a large number of buttons that need to be scanned very quickly.

Because the number of electrode oscillations is the variable of interest, and the base oscillation frequency of an electrode is not adjustable, the gating time (fixed) generally needs to be on the order of 256 µs to 4 ms to allow enough resolution to discern between a touch and no touch condition. The gate timer does not need to be high resolution. Using a slower clock for the gate timer results in lower power consumption during electrode scan. For example, a 1-MHz DCO and SMCLK consumes less power during the electrode scan than an 8-MHz DCO and SMCLK.

The RO method offers some natural immunity to EFT events, as the accumulation of many cycles over the gate time allows for natural averaging that filters out short aperiodic noise. How much averaging occurs is dependent upon how large the gate time is.

2.1.3 TI Capacitive Touch Library RO Method Configuration Parameters

The TI Capacitive Touch library provides hardware abstraction layers for different MSP430 capacitive touch methods. HAL selection and gate time selection are performed at the sensor level in the structure.c file. Figure 4 shows a sample Sensor structure.

```
const struct Sensor buttonEight =
{
  .halDefinition = RO_PINOSC_TA0_WDTp,
  .numElements = 1,
  .baseOffset = 1,
  // Pointer to elements
  .arrayPtr[0] = &button8,
  // Timer Information
  .measGateSource= GATE_WDT_SMCLK,          //  0->SMCLK, 1-> ACLK
  .accumulationCycles= WDTp_GATE_512        //512
};
```

Figure 4. TI Capacitive Touch Library structure.c Sensor Parameters for RO Method

The .halDefinition parameter specifies the RO method as well as the two timers that are used for the measurement. Here, TimerA0 is used to count the electrode oscillations, and the watchdog timer (WDTp) is used to set the gate time for the measurement. The *.measGateSource* parameter specifies the gate time clock when using the RO method. The gate time itself is set with the *.accumulationCycles* parameter.

If the gate timer is the watchdog timer, only the available fixed intervals may be selected. This may require manipulation of the source clock frequency to achieve the desired gate time. A configuration table can be seen in Appendix A.

2.2 fRO (Fast Relaxation Oscillator) Method

The fRO method measures electrode capacitance by using a timer to count a fixed number of electrode oscillations. A second timer operates from a fixed clock at a higher frequency (typically between 8 MHz and 25 MHz) and is used to measure the amount of time it takes the first timer to count the number of electrode oscillations. In this way, fRO is the inverse of RO. The high measurement clock frequency allows gate times to be shorter than the RO method while still providing the same resolution measurement.

If a human interacts with an electrode, the increase in capacitance causes the oscillation frequency to decrease. This causes the amount of time it takes to count the fixed number of oscillations to increase. Therefore, in fRO, a touch creates an increase in counts (the opposite effect of a touch in the standard RO method). Figure 5 shows an example of this concept.

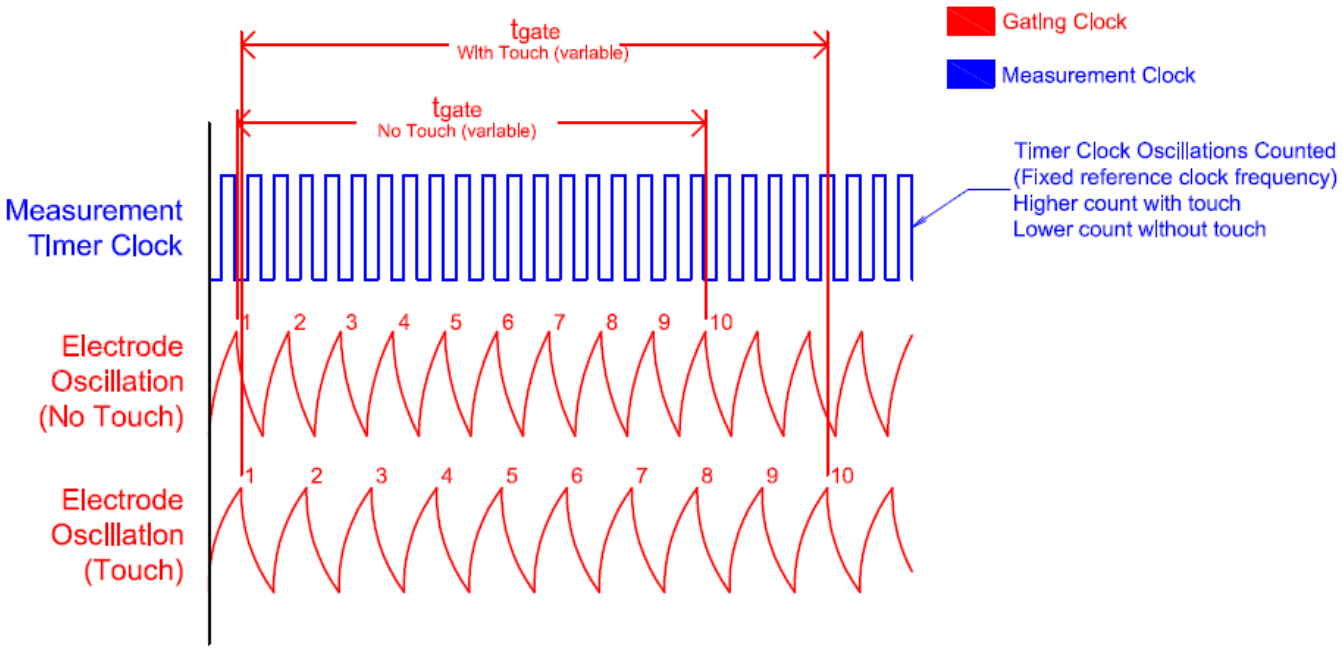

Figure 5. fRO Measurement Timing Diagram

The fast RO method gets its name from the fact that gating times (t_{gate}) can be much less than with the standard RO method. If the measurement timer clock (typically SMCLK) is run at a higher frequency (on the order of 12 MHz to 16 MHz, or approximately ten times the electrode oscillation frequency) the additional clock resolution can allow for gate times to be an order of magnitude shorter. With the fRO method, the gate time is selected by setting a fixed number of electrode oscillations, typically 200 to 1200 cycles. As the measurement clock resolution improves, fewer electrode oscillations are required to obtain the same amount of information.

2.2.1 Required Peripherals for the fRO Method

In addition to the relaxation oscillator peripherals, the fRO method requires two timers, neither of which can be the watchdog timer.

2.2.2 Benefits and Drawbacks of the fRO Method

Power consumption performance for the fRO method is typically on par with that of the RO method. See Section 5.2 for a more detailed power analysis. To obtain short gate times, the measurement timer clock must be run at a higher clock rate. This increases power consumption of the device during gate time and processing time. Depending upon the device, a higher operating voltage may be required to achieve the faster clock speeds. The DCO is required to obtain a faster clock (≥ 8 MHz) without the use of an external clock source. This means that fRO method implementations will need to be in LPM0 during the electrode gate time, and do not have the option of being in LPM3 as some RO method configurations sourcing their gate timer with ACLK may allow.

The fRO method can provide up to five times shorter gate times compared to RO. Quick measurements have several advantages. If a system contains a large number of electrodes (in a remote control keypad, for example) the fRO method allows more keys to be scanned in a period of time. This enables a faster system response time and offers an improved human-machine interface (HMI). In addition, some systems require that the keys be scanned only during a specific time (due to noise events or other system processes). The fRO method's short gate times allows key scanning to fit into a smaller duty cycle to accommodate this.

Some immunity to EFT and aperiodic noise is sacrificed with the fRO method. Shorter measurement times inherently remove some of the natural averaging associated with a longer measurement.

2.2.3 TI Capacitive Touch Library fRO Configuration Parameters

HAL selection and gate time selection are performed at the sensor level in the *structure.c* file. Figure 6 shows a sample Sensor structure.

```
const struct Sensor buttonEight =
{
  .halDefinition = fRO_PINOSC_TA0_TA1,
  .numElements = 1,
  .baseOffset = 0,
  // Pointer to elements
  .arrayPtr[0] = &button8,  // pointer to 8mm
  // Timer Information
  .measGateSource= TIMER_SMCLK,     //  0->SMCLK, 1-> ACLK
  .sourceScale = TIMER_SOURCE_DIV_0,
  .accumulationCycles= 600
};
```

Figure 6. TI Capacitive Touch Library structure.c Sensor Parameters for fRO Method

The .measGateSource parameter has a different meaning with the fRO method than it does with the RO method. The measurement timer clock source is specified in .measGateSource, as opposed to the gate timer clock with the RO method. The number of electrode oscillations to count is specified in .accumulationCycles (this effectively sets the gate time). The .halDefinition parameter specifies the fRO method, PinOsc hardware (MSP430G2xx2 and MSP430G2xx3), and the timers used for the measurement: Timer_A0 counts the electrode oscillations (gate timer for fRO), and Timer_A1 is the measurement timer.

2.3 Selecting a Measurement Method

Determining whether the RO method or fRO method is right for a given system requires an analysis of the goals for the system as well as the system's environment. For example, if the goal for a system is scanning a large array of keys while maintaining an HMI specification for response time, than fRO is more than likely the correct path for the design. Conversely, if the goal for the system is to be a highly robust and highly noise-immune system for an automotive application, the RO method is a better choice. Because the RO method requires only one Timer_A peripheral, it also presents a value proposition as it can be used with devices that only have one Timer_A (MSP430G2xx2 devices, for example).

3 Button Performance vs Gate Time

Following the selection of RO or fRO as the measurement method, the next design step is to set the gate time. As stated in Section 2, the RO method has a constant gate time which is specified by the designer of the system. The fRO method has a variable gate time due to the fixed parameter being the number of electrode oscillations, but the variance in gate time is quite small when compared to the overall gate time (0% to 10% of the baseline time). Again, RO method designs require longer gate times than fRO method designs.

The following sections serve as a starting point for gate time selection. They show what can kind of performance can reasonably be expected from a given gate time, overlay material, electrode size, and measurement method.

3.1 The Basics: Effect of Gate Time on Resolution and SNR

As previously stated, it is beneficial for power consumption and response time to minimize the gate time as much as possible. The effect of **reducing the gate time** (for either RO or fRO) is a **reduction in the resolution of the measurement**. For example, a given electrode scanned for 2.048 ms has a measurement with four times the resolution of a measurement of the same electrode scanned for 0.512 ms. At the same time, when scanned for 2.048 ms, the measurement exhibits four times as many counts of noise compared to a 0.512 ms measurement. Because buttons need only two states (touch or no touch), a sensor only needs enough resolution to provide the desired safety margin between the following three points: baseline count (no touch), threshold, and touch count.

How long the gate time needs to be to obtain that desired safety margin depends upon several factors: the size of the electrode, the thickness and dielectric properties of the overlay, how big the surrounding grounded conductors are (if any), and the resistance and hysteresis settings of the system.

If all of these factors are held constant, the deltas observed due to a touch will vary with gate time (see Figure 7). The delta of a measurement is the deviation in the direction of interest (direction of a touch) for that measurement from the long-term baseline average (a no touch condition). Therefore, even though measurement counts actually decrease due a touch in the RO method, the delta is always a positive number, indicating change from the baseline in the direction of a touch. Figure 7 shows deltas due to a touch in the time domain, with each touch representing a different gate time.

Figure 7 shows the reduction in resolution as gate time is reduced. It also shows that as gate times are reduced the noise count in the measurement is also reduced, as there is less time for noise counts to accumulate. How long the gate time needs to be for a given sensor is now determined by analyzing three factors: the baseline-to-touch delta, the baseline-to-threshold margin, and the threshold-to-touch margin. Figure 8 shows a sample touch for a given system configuration and identifies these factors.

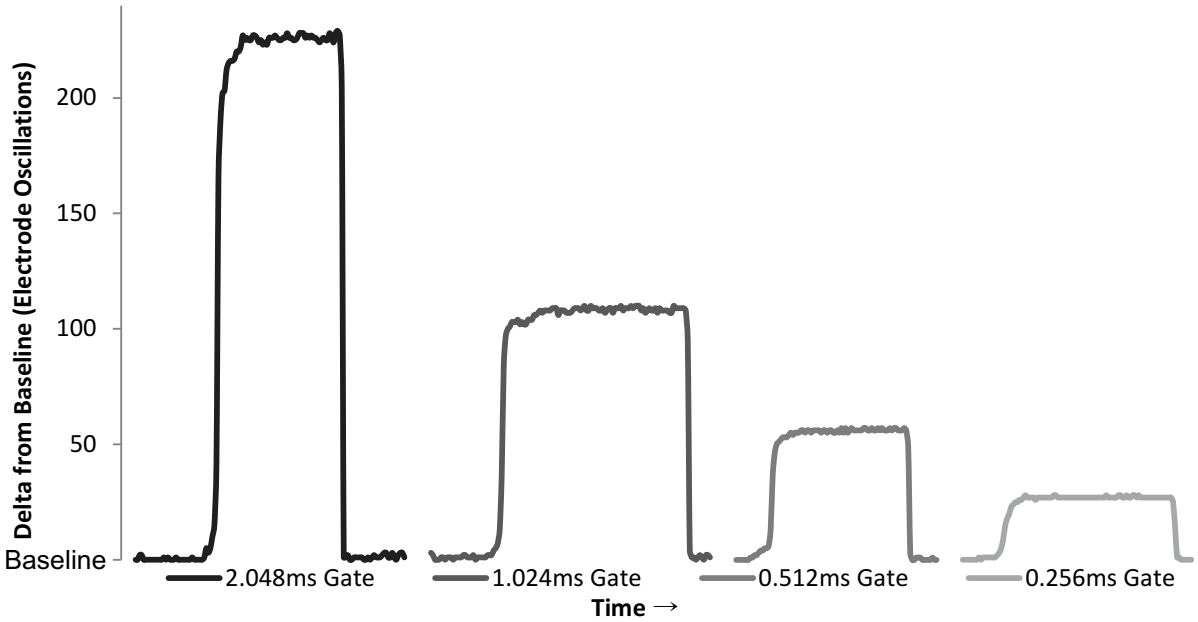

Figure 7. Button Touch Deltas for Four Gate Times (8mm Electrode, RO Method, MSP430G2553)

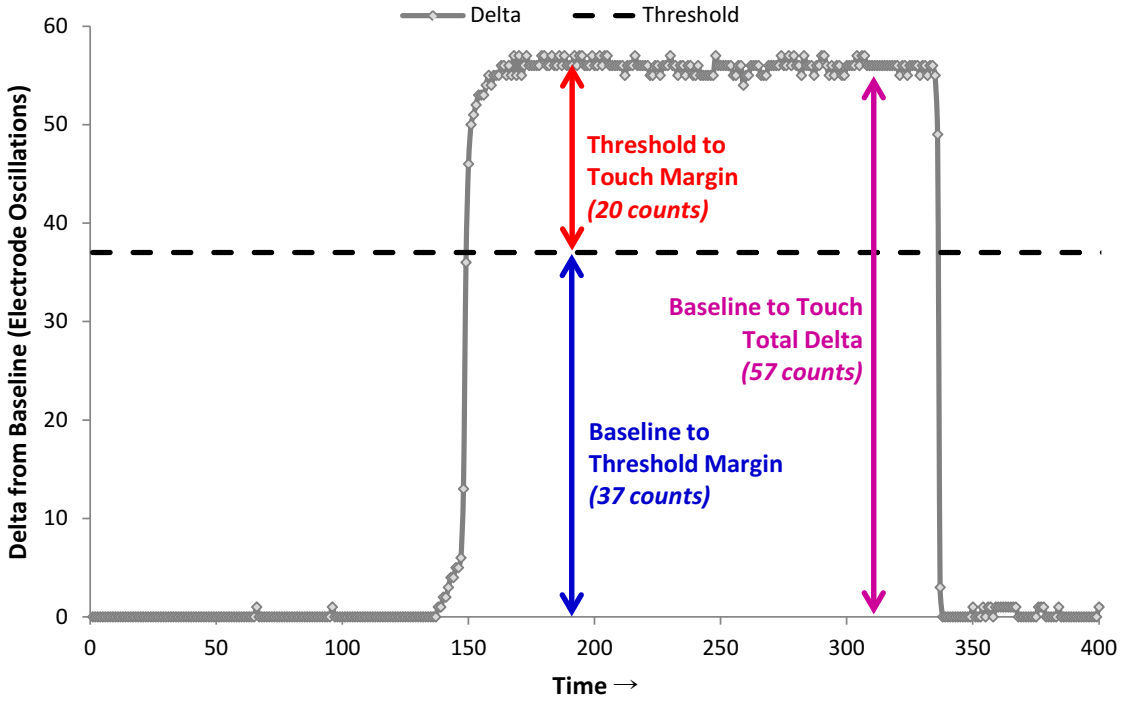

Figure 8. Analyzing a Touch – Safety Margin and Total Delta (0.512-ms Gate Time, 8-mm Electrode, RO Method, MSP430G2553)

3.1.1 Baseline-to-Touch Delta

The baseline-to-touch delta, or simply the "touch delta," is the difference in counts from the baseline due to a touch on the button. When calculating the SNR of a system, this metric represents the signal. The noise component of SNR is typically calculated by analyzing the variation in no-touch measurements over time.

The gate time of the sensor needs to be long enough to provide a touch delta large enough to place a threshold point. The threshold point is the crossover point at which a touch is declared. Setting the threshold higher or lower within the touch delta creates different "feels" for the button. The threshold can be set low to allow light touches to trigger detection, or it can be set high to require a firm touch to trigger detection. While the touch delta is considered to be the signal, using this information for the signal-to-noise analysis on its own can be misleading. This is why the baseline-to-threshold margin and threshold-to-touch margin need to be considered in gate time selection.

3.1.2 Baseline-to-Threshold Margin

The baseline-to-threshold margin, or simply the "threshold," is the delta from the baseline at which a given element is declared in detect. The threshold is the parameter that determines the "feel" of the element, and is designer-selected. The threshold for a given element is set with the .threshold parameter in the structure.c file (see Figure 9). The threshold for a given element should be set somewhere between zero and the baseline-to-touch delta. For a given system, each element usually has a separate threshold to achieve the desired feel.

"Feel" refers to how much interaction is required with the element to trigger a touch on that element. Low thresholds (low percentage of the baseline-to-touch delta) allow small interactions to trigger a touch. Where a threshold needs to be set depends on the intended application. The delta achieved to a user interaction varies with how hard the user is touching the panel, as a finger flattens with increased pressure, which increases surface area. Touch delta is also smaller for users with smaller fingers, such as children. Users wearing gloves add a dielectric layer between the user and the overlay, further decreasing the delta due to a touch. For information on how to select a threshold, see Section 4.

```
const struct Element buttonElement =
{
    .inputBits = BIT1,
    .inputPxselRegister = (unsigned char *)&P2SEL,
    .inputPxsel2Register = (unsigned char *)&P2SEL2,
    .threshold = 0
};
```

Figure 9. TI Capacitive Touch Library structure.c Element Parameters

The baseline-to-threshold margin is also the safety margin that protects against false detection. When designing for protection against false detections, the relationship between counts of noise and the threshold becomes important, not just SNR. It is important to consider the operating conditions of the device. If a device is subject to a noisy environment, it is ideal to increase gate time to provide greater measurement resolution, in turn increasing the margin between the baseline and the threshold.

3.1.3 Threshold-to-Touch Margin

The threshold-to-touch margin is the number of counts between the threshold and the touch delta. This margin should be large enough that the system does not bounce in and out of detect during a touch. The gate time should be set to provide enough resolution between the threshold and the touch delta to interpret small movements on the button's electrode as well as variations in how hard the touch is.

The same factors that are described in Section 3.1.2 affect the threshold-to-touch margin. When selecting a threshold, baseline-to-threshold margin and the threshold to touch margin are considered.

3.2 Case Study: Gate Time Feasibility

There is no absolute rule for predicting how small a gate time can be for a given system configuration. Every system has its own unique set of challenges, such as nearby ground loading or a thick overlay material. Still, it is possible to estimate what a reasonable gate time would be for a standard button configuration. This case study demonstrates the performance of a 6mm x 6mm, an 8mm x 8mm, and a 10mm x 10mm electrode when connected to an MSP430G2553 and an MSP430FR5969. The RO and fRO methods are compared. To test the performance of the three buttons, the PCB in Figure 10 is used. This PCB is a plug-in BoosterPack™ kit for the MSP430™ LaunchPad™ development kit (MSP-EXP430G2, MSP-EXP430FR5969). Two BoosterPacks are used, one with a 1.5-mm Lexan overlay and the other with a 2.5-mm overlay.

The three test buttons have a 25% ground fill on the lower layer (two-layer 0.0625-inch thick FR4 is used). The partial ground fill provides some stability from outside noise, while trading off a small amount of sensitivity and resolution. The PCB also includes a proximity detection electrode, which is not used for this study. The proximity electrode is held at ground while the other buttons are measured.

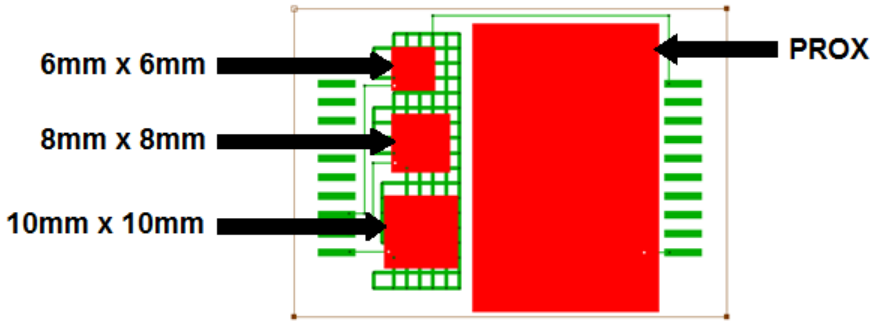

Figure 10. Case Study PCB Layout

The following tables show the touch deltas for a given configuration from the case study. Table 1 shows the recommendation color key. If an electrode size and gate time combination results in a "green" cell, the configuration is likely feasible in a variety of configurations. If the configuration is a yellow cell, it may be feasible, but will likely depend on the system. If the configuration is red, it is not recommended. The red, yellow, and green recommendations are based upon having a minimum 15:1 SNR and 30 counts of resolution. While it is possible to design buttons that provide deltas of less than 30 counts (down to a 5:1 SNR), having at least 30 counts will provide a more robust solution that also allows more flexibility in tuning due to the increased measurement resolution.

The data is here to serve as a starting point for a design. Every system requires careful gate time selection. In any system, each electrode has a unique base capacitance, and some buttons may require different gate times than others, even if they are the same size.

Table 1. Case Study Color Key

Typically Acceptable (Touch Δ ≥ 45+)	Possible (depends on system) (45 >Touch Δ > 30)	Not Recommended (30 ≥ Touch Δ)

Table 2 and Table 3 show the typical touch deltas for the MSP430G2553 using the RO method with a 1.5-mm overlay and a 2.5-mm overlay, respectively. For this scenario, 1-ms gating times are realistic for most systems, with 512 µs being possible in some situations.

Table 2. Typical Touch Deltas: MSP430G2553, RO Method, 1.5-mm Overlay

Button Electrode Size	2.048-ms Gate	1.024-ms Gate	0.512-ms Gate	0.256-ms Gate
6mm x 6mm	162	83	40	20
8mm x 8mm	224	113	56	28
10mm x 10mm	333	164	83	41

Table 3. Typical Touch Deltas: MSP430G2553, RO Method, 2.5-mm Overlay

Button Electrode Size	2.048-ms Gate	1.024-ms Gate	0.512-ms Gate	0.256-ms Gate
6mm x 6mm	99	48	23	12
8mm x 8mm	143	70	35	18
10mm x 10mm	211	104	52	26

Table 4 and Table 5 show the typical touch deltas for the MSP430G2553 using the fRO method with a 1.5-mm overlay and a 2.5-mm overlay, respectively. For this scenario, 180-µs gating times are realistic for most systems, with 90 µs being possible in some situations.

Table 4. Typical Touch Deltas: MSP430G2553, fRO Method at 12 MHz, 1.5-mm Overlay

Button Electrode Size	≈0.35-ms Gate (800 Cycles)	≈0.18-ms Gate (400 Cycles)	≈0.09-ms Gate (200 Cycles)	≈0.05-ms Gate (100 Cycles)
6mm x 6mm	130	63	29	13
8mm x 8mm	226	114	56	28
10mm x 10mm	369	184	95	48

Table 5. Typical Touch Deltas: MSP430G2553, fRO Method at 12 MHz, 2.5-mm Overlay

Button Electrode Size	≈0.35-ms Gate (800 Cycles)	≈0.18-ms Gate (400 Cycles)	≈0.09-ms Gate (200 Cycles)	≈0.05-ms Gate (100 Cycles)
6mm x 6mm	84	39	17	6
8mm x 8mm	149	77	38	20

Table 5. Typical Touch Deltas: MSP430G2553, fRO Method at 12 MHz, 2.5-mm Overlay (continued)

Button Electrode Size	≈0.35-ms Gate (800 Cycles)	≈0.18-ms Gate (400 Cycles)	≈0.09-ms Gate (200 Cycles)	≈0.05-ms Gate (100 Cycles)
10mm x 10mm	258	121	64	32

The internal capacitive sensing peripheral on the MSP430FR58xx and FR59xx devices oscillates at a higher frequency than the MSP430G2xxx devices. This configuration enables a higher resolution measurement when using the RO method, allowing the RO method gate times to be smaller. Table 6 shows these advantages.

Table 6. Typical Touch Deltas: MSP430FR5969, RO Method, 1.5-mm Overlay

Button Electrode Size	1.024-ms Gate	0.512-ms Gate	0.256-ms Gate	0.128-ms Gate
6mm x 6mm	87	43	21	11
8mm x 8mm	148	73	35	17
10mm x 10mm	221	110	53	26

3.3 Effect of System Geometry on Touch Deltas and SNR

The geometry of the system plays a key role in the performance of a button. Smaller button electrodes (<40mm^2) require longer gate times than larger electrodes (>64mm^2). Thicker overlays also require longer gating times. Figure 11 shows the effect of electrode size and overlay thickness on typical touch deltas.

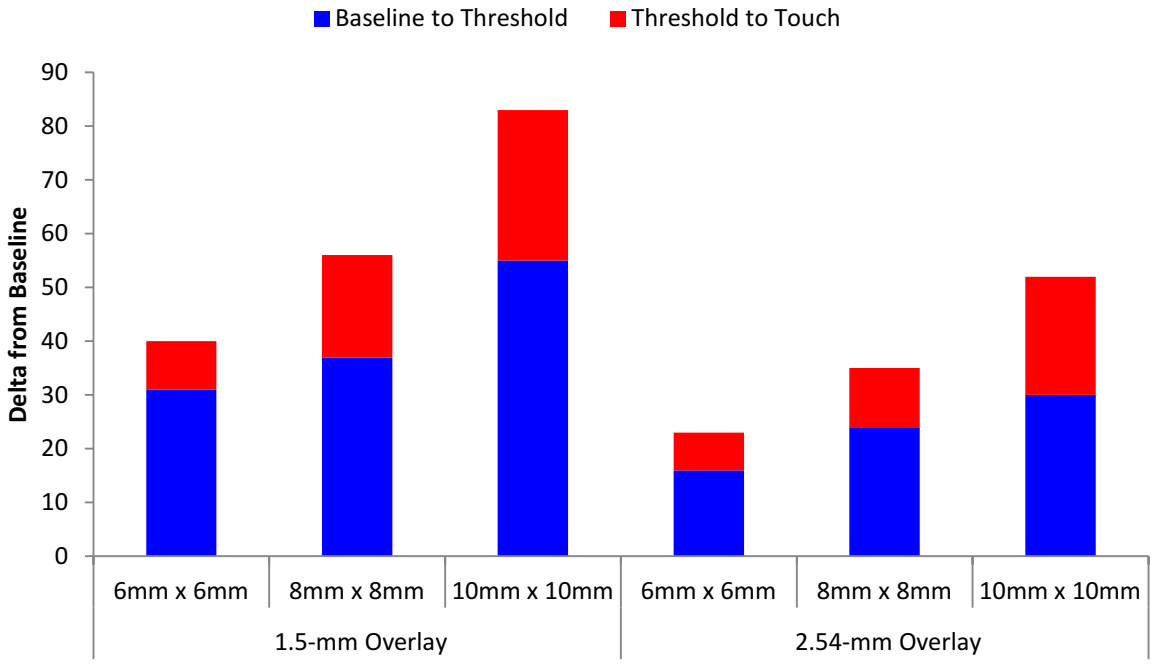

Figure 11. Effect of Electrode Size and Overlay Thickness on Touch Deltas (MSP430G2553, RO Method, 8 MHz, 3.3 V, 0.512-ms Gate Time)

4 Button Tuning and Gate Time Selection How-To

Recall from Section 3.1 that buttons have two parameters that are user-configurable with regard to tuning: the button's gate time and the button's threshold. This section serves as a step-by-step guide to selecting these two parameters for a given system. To assist the tuning process, TI's MSP-TouchPro GUI tool (SLAC552) is used to display real-time measurements on a PC.

To demonstrate button tuning and gate time selection, the PCB used in the gate time case study in Section 3.2 is used. This PCB features an 8mm x 8mm electrode that is used to demonstrate the tuning and gate time selection process for a button. The PCB is connected as a BoosterPack to an MSP-EXP430G2 LaunchPad (revision 1.3). Figure 12 shows the hardware configuration.

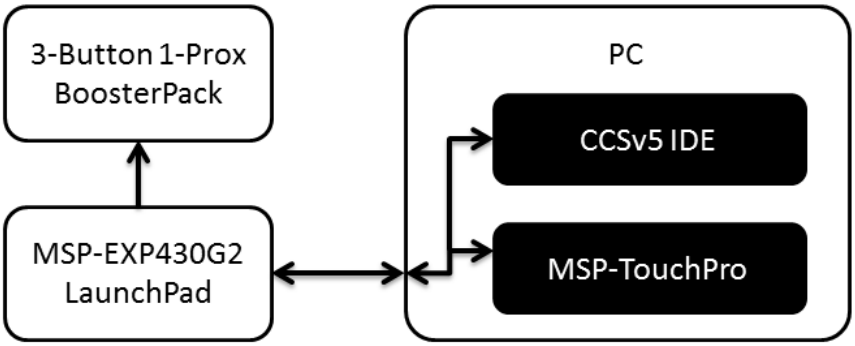

Figure 12. Tuning Example Hardware Setup

The collateral for this how-to guide can be found in the document folder. Sample projects for Code Composer Studio™ IDE v5.2 (\Examples\CCSv5\TouchPro_ButtonTuningSample_RO, and \Examples\CCSv5\TouchPro_ButtonTuningSample_fRO) allow this example to be replicated for another system. The MSP-TouchPro GUI tool can be downloaded from www.ti.com.

The sample CCSv5 projects include the capacitive touch library (CTS_HAL.h, CTS_HAL.c, CTS_Layer.h, CTS_Layer.c), the capacitive touch structure files (structure.h, structure.c), UART protocol transmit functions and a bit-bang UART function (MSP_TouchPro_Utility.h, MSP_TouchPro_Utility.c), and a sample application (main.c). The sample projects are intended for an MSP430G2553-PDIP20 device but can be ported to another Value Line device (MSP430G2xx2 or G2xx3). One project demonstrates the RO method, and another project demonstrates the fRO method.

The MSP-TouchPro GUI tool allows real-time viewing of capacitive touch measurements. This allows the designer of a capacitive touch system to quickly visualize important system characteristics such as measurement resolution and system noise. In addition, the real-time view is particularly helpful in observing the range of possible button touch deltas due to different user interactions such as large versus small fingers, fingers not completely on the button, and light versus firm touches. See the *MSP-TouchPro GUI User's Guide* (SLAU483) for detailed information on how to use the GUI.

The following sections describe the steps that are required for button tuning and gate time selection for the example system described above. Figure 13 shows a flowchart for the process.

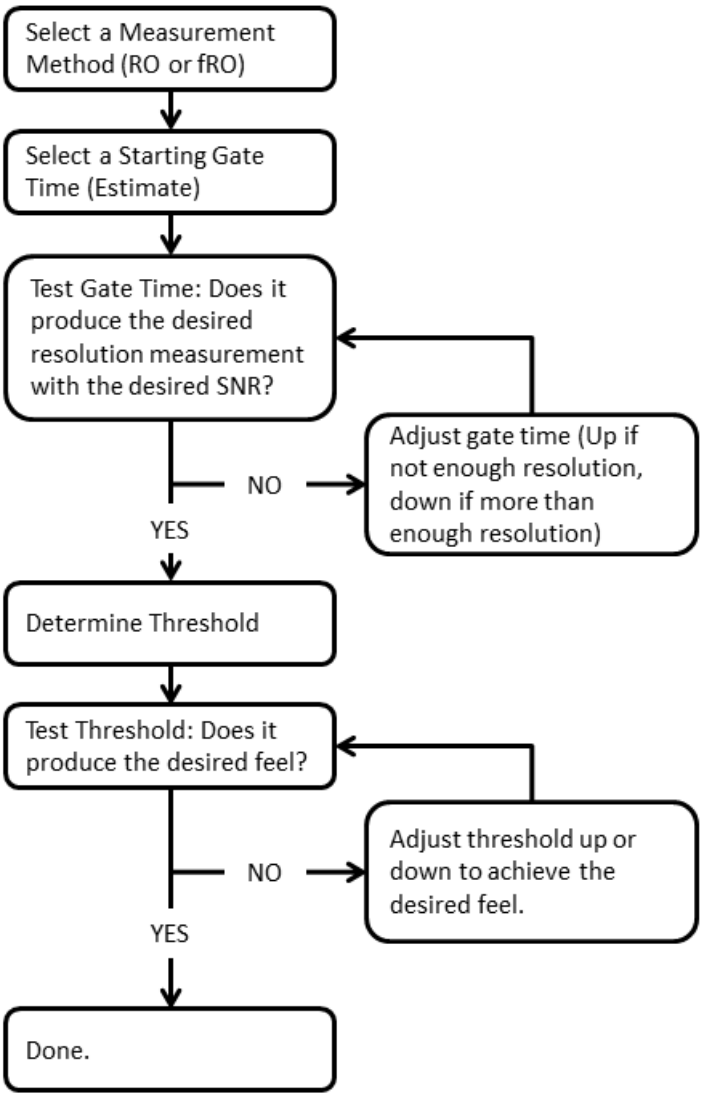

Figure 13. Button Gate Time Selection and Tuning Step-by-Step Flowchart

4.1 Step 1: Select a Measurement Method (RO or fRO)

Step 1 is the selection of a measurement method. For this example, the RO method is used as the sample system only has one button and does not require the fast gate times that the fRO method offers. The RO method also provides slightly greater immunity to EFT events and allows the use of just one Timer_A instance in addition to the watchdog timer.

In the sample CCSv5 project for the RO method, the *structure.c* file is configured to use the RO method with Timer_A0 as the oscillation counter and the watchdog timer as the gate timer. This is the red box in Figure 14.

```
#include "structure.h"

const struct Element buttonElement =
{
    .inputBits = BIT1,
    .inputPxselRegister = (unsigned char *)&P2SEL,
    .inputPxsel2Register = (unsigned char *)&P2SEL2,
    .threshold = 35

};

const struct Sensor buttonSensor =
{
    .halDefinition = RO_PINOSC_TA0_WDTp,
    .numElements = 1,
    .baseOffset = 0,
    // Pointer to elements
    .arrayPtr[0] = &buttonElement,

    .measGateSource = GATE_WDT_SMCLK,
    .accumulationCycles = WDTp_GATE_8192
};
```

Method & Timers

Gate Clock

Gate Time

Figure 14. How-To structure.c File

4.2 Step 2: Set an Initial Gate Time

Determining the appropriate gate time begins with an educated guess. Based upon the button's size, the thickness of the overlay, and the device being used, the case study in Section 3.2 can be used to determine a scan time that is close to what is required for the button. Generally, 1 ms is a good starting point for a button when using the RO method. The gate time is programmed by selecting a gate clock (SMCLK or ACLK, orange box in Figure 14) and setting the number of clock cycles to be accumulated (blue box in Figure 14). For the watchdog timer implementation, only fixed intervals are selectable. Review the structure.h file for available intervals. To obtain gate times other than what the fixed watchdog timer intervals allow, the gate clock source or its divider can be changed. Appendix A contains tables that show which combinations of watchdog timer interval and gate clock speed provide which gate times. For this example, 1.024 ms was selected as the starting gate time. This time was set by utilizing an 8-MHz SMCLK with 8192 accumulation cycles.

NOTE: For the RO method, having a high gate clock frequency increases power consumption during electrode scan. If possible, combinations with lower clock frequencies and higher accumulation cycles are preferred.

4.3 Step 3: Test the Button's Gate Time

After an initial gate time has been selected, the next step involves observing the performance of the button with that configuration. Based upon the performance, the gate time may need to be increased (if not enough resolution is provided) or it may be able to be decreased (if more than enough resolution is provided). Because the button has not yet been tuned, the threshold in the structure.c file should be set to zero. Setting the threshold to zero disables the baseline tracking feature of the library.

4.3.1 Configuring Firmware to Communicate with the MSP-TouchPro GUI Tool

The MSP_TouchPro_Utility.h and MSP_TouchPro_Utility.c files are used to interface an MSP430 with the MSP-TouchPro *GUI*. The files offer three APIs for communicating with TouchPro. The UART_TXbyte() function implements a bit-bang UART on the GPIO pin specified in the definitions UART_PORT and UART_PIN in the MSP_TouchPro_Utility.h file. The ASSIGNED_UART_BIT_LENGTH definition sets the bit length for a given MCLK and UART baud rate. Figure 15 shows these definitions.

```
#define ASSIGNED_UART_BIT_LENGTH    UART_BIT_LEN_8MHZ_9600B     // Assigned bit length (choose from above)
#define UART_PORT                   P1OUT                       // UART TXD port    (PUART_PORT.x)
#define UART_PIN                    BIT1                        // UART TXD pin     (Px.UART_PIN)

void UART_TXbyte(uint8_t byteToSend);
void UART_TXpacket_RawData(uint16_t* data, uint8_t channelCnt, uint8_t startChannel);
void UART_TXpacket_Meas_Base_Thresh(uint16_t* currMeas, uint16_t* currBaseline, uint16_t* currThresh);
```

Figure 15. MSP_TouchPro_Utility.h Definitions and Declarations

The UART_TXpacket_RawData() function accepts a pointer to an array of unsigned integers, a channel count to send, and a starting channel (starting index within the data array). It transmits a properly formatted TouchPro data packet via the UART_TXbyte() function. The UART_TXpacket_Meas_Base_Thresh() function transmits a 3-channel packet for tuning. It needs to be passed the measurement, baseline, and absolute threshold for a given sample. This is the function that will be used to observe button performance. The sample *main.c* file contains a #define CHARACTERIZE at the top of the file. Defining CHARACTERIZE configures the sample application code to repeatedly measure the electrode and transmit a packet to TouchPro for viewing. The "Multi Channel" mode in TouchPro is used.

4.3.2 Observing the Button's Performance

After the system is programmed to measure the electrode and transmit the sample data to *MSP-TouchPro*, the button's real-time measurement data can be observed. A touch on the button should produce a reduction in the measurement value (due to the RO method) (see Figure 16). Channel 1 represents the measurement, Channel 2 represents the baseline count, and Channel 3 represents the threshold (absolute, or the base count minus the programmed threshold). Looking at the plot, the baseline-to-touch delta can be measured. In the example, the baseline count is approximately 2238 oscillations. During a touch, the measurement is approximately 2135 counts. This gives a baseline-to-touch delta of 103. This is fairly close to the results seen in the gate time study (see Table 2).

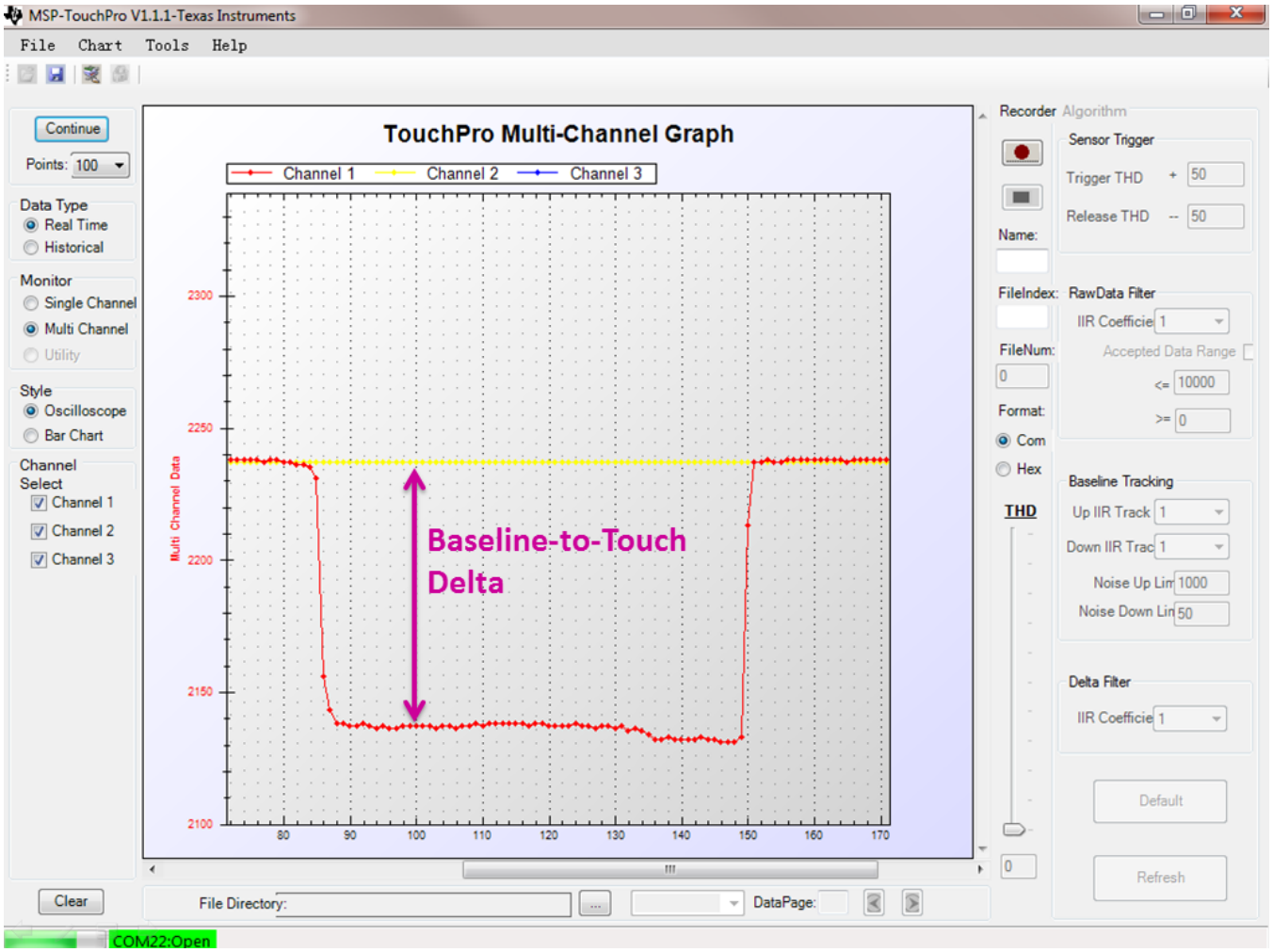

Figure 16. MSP-TouchPro Gate Time Testing

4.4 Step 4: Change Gate Time if Necessary

When the results from Step 3 are completed, the gate time may need to be increased or decreased depending upon whether or note the desired resolution was achieved.

For a button, 100 points of resolution between a touch and no touch measurement is more information than is needed to determine detect. While 1 ms was a good starting point, it offers more information than is required and therefore, for this example, the next step would be to reduce the gate time to 512 µs (a reduction by a factor of two). This provides a measurement with half the resolution (approximately 50 counts of delta due to a touch).

4.5 Step 5: Select a Threshold

When a gate time that provides the desired resolution for the button is identified, the next step is to determine the threshold for the button (baseline-to-threshold margin). The threshold is the artistic parameter in capacitive touch. It sets the level of interaction with the electrode at which a given button is declared to be in detect.

Where the threshold needs to be for a button is dependent upon the desired "feel" of the button. "Feel" refers to how hard or soft the touch needs to be to be declared a touch by the MCU. In addition to the firmness of the touch, the positioning of the finger on the button and the size of the user's finger need to be considered when selecting a threshold. Figure 17 shows touch deltas due to different touch angles and touch firmness.

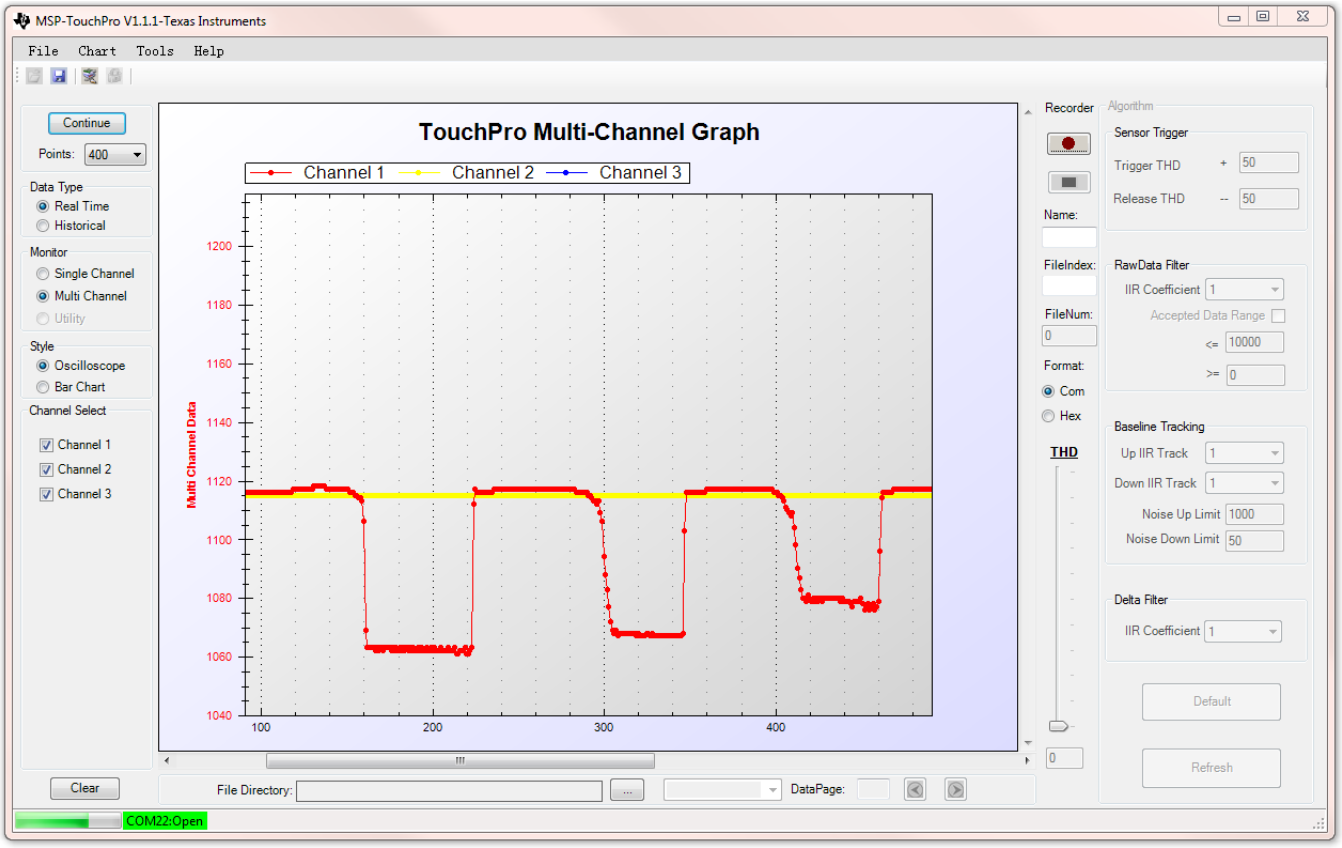

Figure 17. MSP-TouchPro Threshold Setting

4.6 Step 6: Test the Threshold

When a threshold is selected, it is programmed in the structure.c file (see Section 3.1.2 for more information). There are two ways to test the threshold: using the MSP-TouchPro GUI, or by blinking an LED. To test the threshold with the MSP-TouchPro in the sample project, simply leave the main.c application code the same and re-run the application. Now, when the data stream in MSP-TouchPro is viewed, the threshold will appear as Channel 3 (blue). To test the threshold with an LED, comment out the #define CHARACTERIZE line at the top of main.c. This sets the application code to run capacitive touch as if it were an actual application, where a finger on the button causes the LaunchPad LED on P1.0 to light.

4.7 Step 7: Done!

If the button's feel matches the desired feel for the system, then tuning is complete. If it does not, repeat steps 5 and 6 until the desired feel is obtained.

5 Gate Time vs Power Consumption

Maximizing capacitive touch system power performance requires minimizing the active duty cycle and maximizing the sleep duty cycle of the system. Section 3 describes how to determine a gate time for a given system and provides a starting point for doing so. This section elaborates on the power consumption benefits of reducing the gate time for a sensor.

To explore the effects of button gate time on power consumption further, download the MSP430™ Capacitive Touch Power Designer GUI (SLAC551). This tool enables a designer to estimate the power consumption of a given capacitive touch system.

5.1 Case Study: Gate Time Reduction

The effect of gate time reduction on power consumption can best be visualized through an example case study. The example system will be a capacitive touch system with three 8mm x 8mm square buttons. The selected MCU is an MSP430G2553, running at 3.3 V and 1 MHz. There are two test cases. First, the system is configured with a 2.048-ms gate time per button. This gate time offers more than enough resolution to determine whether or not a button is touched. Second, the system is configured with a 0.512-ms gate time per button. This gate time offers one-quarter the resolution of the first case but still provides enough information to determine whether or not a button is touched. Figure 18 shows the sample duty cycles.

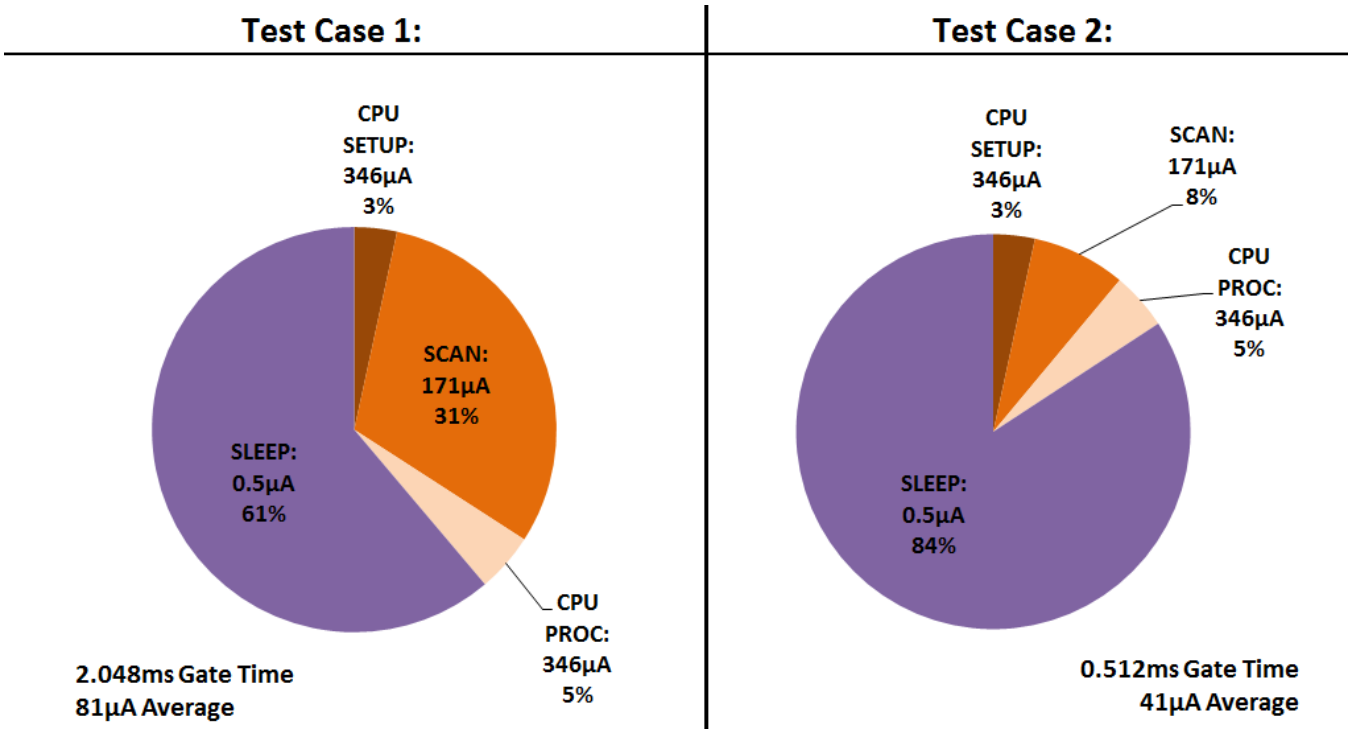

Figure 18. MCU Duty Cycle for Three Buttons at a 2.048-ms Gate Time and a 0.512-ms Gate Time (MSP430G2553, 3.3 V, RO Method, 1 MHz)

Figure 18 shows that reducing the sensor's gate time from 2.048 ms to 0.512 ms (a reduction factor of 4) decreases the average current draw of the system from 81 µA to 41 µA (a reduction factor of almost 2). The only sacrifice is extra measurement resolution, which was not required for the simple functionality of a button.

NOTE: There is some marginal CPU activity (not shown) in the scan portion of Figure 18. This activity re-configures the MCU to measure the next button.

5.2 Power Performance: RO vs fRO

Average power consumption is very similar between the RO method and the fRO method. To properly compare the two, each method is configured with gate times that provide the same resolution measurement (same counts of delta due to the same touch). With all other factors being equal, fRO can provide slightly lower average current draw than the RO method while returning the same resolution measurement. The smaller gate times associated with the fRO method offset the increased LPM0 and active mode currents associated with the higher DCO frequency required for the measurement. Figure 19 shows the differences in average current draw between RO and fRO for 10-Hz, 20-Hz, and 50-Hz scan rate at two resolutions.

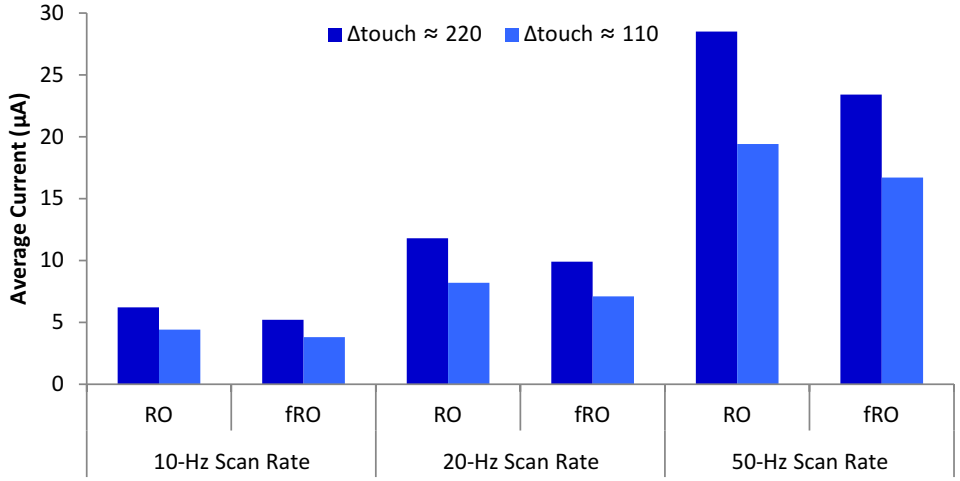

Figure 19. RO vs fRO Average Current (MSP430G2553, 3.3 V, fRO Measurement Clock = 8 MHz, One Button)

5.3 Power Performance: fRO at Different Measurement Clock Speeds

As discussed in Section 2.2, the fRO method uses a higher-frequency measurement clock, on the order of 8 MHz to 25 MHz, which is typically sourced from the digitally controlled oscillator (DCO). Running the measurement clock at a higher frequency increases the resolution of the measurement which in turn allows the gate time to be reduced for a given button. CPU processing times and CPU setup times also decrease proportionally with increased clock frequencies. Therefore, the key scan current draw occurs for less time. However, running the measurement clock at a higher frequency draws more current for that time and may also require a higher operating voltage. Figure 20 shows power performance for fRO across frequency and voltage.

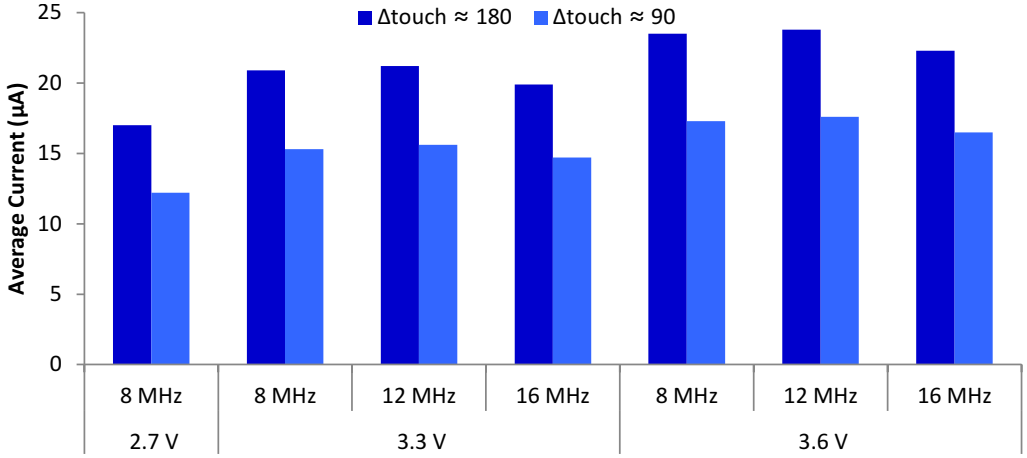

Figure 20. fRO Average Current for Various Measurement Clock Rates (MSP430G2553, One Button, 50 Hz)

Figure 20 shows that an increase of the measurement clock frequency with a decrease of the gate time (measurement resolution held constant) improves response time while holding average current draw relatively constant.

5.4 Power Performance: Scan Rate

An increase of the scan rate of the capacitive touch system increases the average power consumption in a linear fashion. Figure 21, which was generated with the MSP430 Capacitive Touch Power Designer GUI, demonstrates the relationship between scan rate and average current. In addition, it shows that operating at higher voltages can cause a significant increase in power consumption, especially as scan rate is increased.

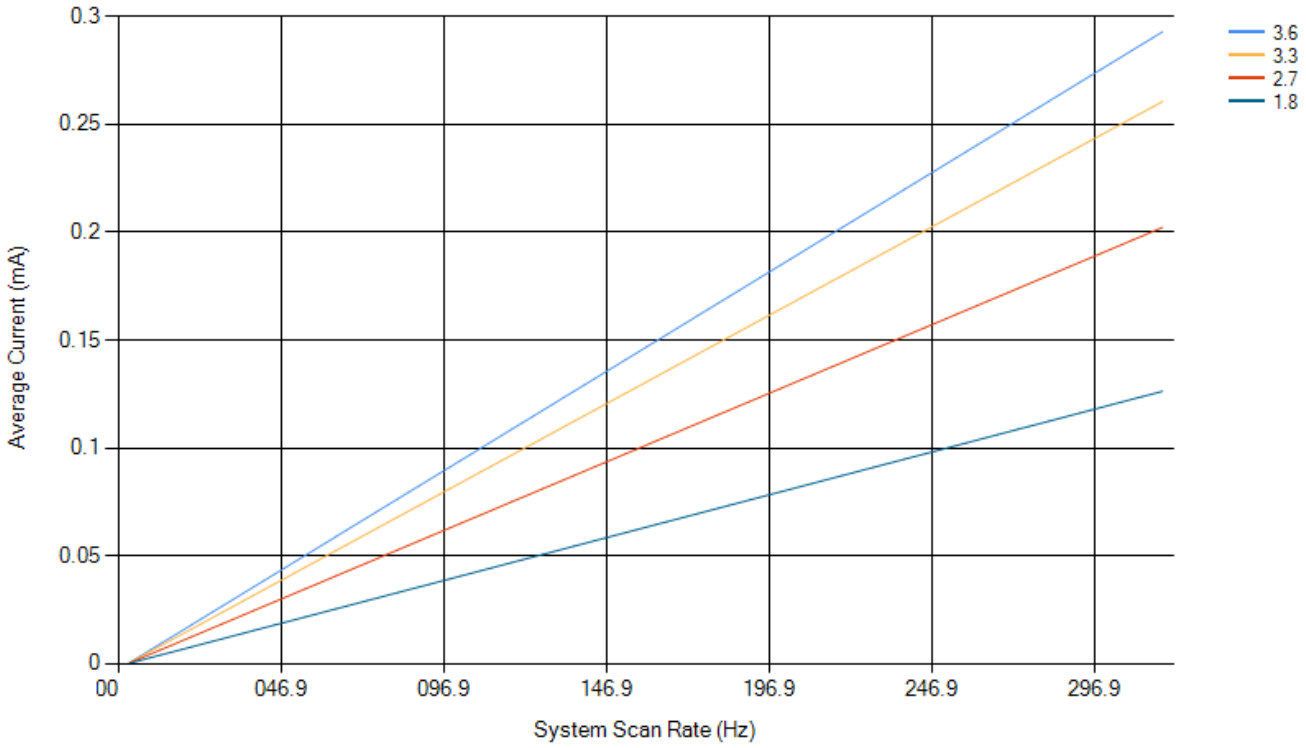

Figure 21. Power Performance: Scan Rate vs Average Current (MSP430G2553, RO Method, 1 MHz, One Button, 0.512-ms Gate Time)

6 What to Expect: A Capacitive Touch Button Quick Reference Sheet

This section is a summary and quick reference for MSP430 relaxation oscillator capacitive touch button design.

Which measurement method should I use (RO or fRO)?

RO: Provides a more robust measurement due to natural averaging over a longer gate time. Provides a value proposition as only one additional timer is required if the watchdog timer is used (two timers total, Timer_A and WDT). RO is ideal for most designs that do not require the fast response times of fRO.

fRO: Can provide gate times that are five times less than RO, which allows more buttons to be scanned in less time. fRO is ideal for designs with high button count or fast response times. Power consumption is slightly lower when compared with RO.

What kind of gate times can I expect to need for buttons?

RO: Gating times typically need to be between 0.25 ms and 4 ms, depending upon the electrode size, overlay thickness, nearby grounding, and desired button feel. Larger buttons decrease the required gate time, as do thinner overlays and reduced ground pours. 1 ms is a good starting point for most designs, with time less than 0.25 ms sometimes being achievable.

fRO: Gating times typically need to be between 0.06 ms and 0.7 ms, depending upon the electrode size, overlay thickness, nearby grounding, and desired button feel. Larger buttons decrease the required gate time, as do thinner overlays and reduced ground pours. In addition, the measurement clock frequency is directly related to the required gate time.

What will my power consumption be?

Average power consumption per electrode, for both RO and fRO methods, is typically in the tens of microamps on average. Every system is different. Power consumption varies greatly with gate time, which varies with electrode size, overlay thickness, nearby grounding, and desired button feel. These factors contribute most to the power performance that is achievable.

Appendix A RO Method Watchdog Timer (WDTp) Configurations

A.1 MSP430G2xx3 With 1-MHz DCO Frequency

Table 7. Oscillator Frequencies

	DCO	VLO
Hz	1000000	12000
ms	0.001	0.083333
Options:	1, 8, 12, 16 MHz	0.012 MHz

Table 8. Clock Divider Options

DIV	SMCLK (DCO) (Hz)	ACLK (VLO) (Hz)
1	1000000	12000
2	500000	6000
4	250000	3000
8	125000	1500

Table 9. SMCLK (DCO) Timing Periods (ms)

	SMCLK (Hz)	WDT+ Accumulation Cycles			
		64	512	8192	32768
SMCLK Dividers	1000000	0.064	0.512	8.192	32.768
	500000	0.128	1.024	16.384	65.536
	250000	0.256	2.048	32.768	131.072
	125000	0.512	4.096	65.536	262.144

Table 10. ACLK (VLO) Timing Periods (ms)

	ACLK (Hz)	WDT+ Accumulation Cycles			
		64	512	8192	32768
ACLK Dividers	12000	5.333	42.7	682.7	2730.7
	6000	10.667	85.3	1365.3	5461.3
	3000	21.333	170.7	2730.7	10922.7
	1500	42.667	341.3	5461.3	21845.3

A.2 MSP430G2xx3 With 8-MHz DCO Frequency

Table 11. Oscillator Frequencies

	DCO	VLO
Hz	8000000	12000
ms	0.000125	0.083333
Options:	1, 8, 12, 16 MHz	0.012 MHz

Table 12. Clock Divider Options

DIV	SMCLK (DCO) (Hz)	ACLK (VLO) (Hz)
1	8000000	12000
2	4000000	6000
4	2000000	3000
8	1000000	1500

Table 13. SMCLK (DCO) Timing Periods (ms)

	SMCLK (Hz)	WDT+ Accumulation Cycles			
		64	512	8192	32768
SMCLK Dividers	8000000	0.008	0.064	1.024	4.096
	4000000	0.016	0.128	2.048	8.192
	2000000	0.032	0.256	4.096	16.384
	1000000	0.064	0.512	8.192	32.768

Table 14. ACLK (VLO) Timing Periods (ms)

	ACLK (Hz)	WDT+ Accumulation Cycles			
		64	512	8192	32768
ACLK Dividers	12000	5.333	42.7	682.7	2730.7
	6000	10.667	85.3	1365.3	5461.3
	3000	21.333	170.7	2730.7	10922.7
	1500	42.667	341.3	5461.3	21845.3

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265