

1- μ A Capacitive Grip Detection Based on MSP430™ Microcontrollers

Holly Gu and Tyler Berryhill

ABSTRACT

This application report discusses how to design a simple and ultra low-power grip detector BoosterPack for the MSP430 LaunchPad™. The PINOSC feature in every msp430G2xx2 and msp430G2xx3 device makes it possible to eliminate all the external components when implementing a capacitive touch design. All the components used in the grip detector, such as capacitive sensors, LEDs are all integrated in a BoosterPack, which is completely compatible with the MSP430 LaunchPad interface. And this grip detector can also communicate with the TouchPro graphical user interface (GUI) to make the test and debug process much faster and easier. Measurement results show that the average current consumed by this grip detector working in idle mode is nearly 1 μ A, which proves that the MSP430 value line series of devices have excellent low-power characteristics and are suitable for battery powered applications.

Project collateral, example source code and additional information can be downloaded from www.ti.com/lit/zip/sl原因515.

Contents

1	System Overview	2
2	Capacitive Measurement Methods	3
3	Hardware Design	4
4	Firmware Implementation	6
5	Power and Performance	10
6	Conclusion	13
7	References	13
Appendix A	Schematic and PCB Layout of the Grip Detection BoosterPack	14

List of Figures

1	Demo of the Grip Detector.....	2
2	Schematic of PINOSC in MSP430G2xx Devices	3
3	RO Capacitive Measurement Method	4
4	Top and Bottom View of the Grip Detection BoosterPack.....	4
5	Cross-Sectional View of the Grip Detector	5
6	Firmware Design Flow.....	9
7	Current Profile in Idle Mode	10
8	Screenshots of the Current Profile in Idle Mode	11
9	Schematic of the Grip Detection BoosterPack.....	14
10	PCB Layout of the Grip Detection BoosterPack.....	14

List of Tables

1	Average Current Measured in Different Working Stages	11
2	Code Size in Different Usage Conditions	12
3	Threshold and Delta Counts When Touched and Non_touched	12

LaunchPad, Code Composer Studio are trademarks of Texas Instruments.
 All other trademarks are the property of their respective owners.

1 System Overview

Capacitive touch is increasingly popular and widely used in products such as mobile devices and white goods. You can create various new touch controls, such as proximity sensors, buttons, wheels, and slides. Another new capacitive touch application is the grip detector described in this application report. The grip detector includes two grip sensors, two buttons and two LEDs, as shown in [Figure 1](#). The printed circuit board (PCB) is protected by an ABS plastic enclosure and neither visible nor accessible to you.



Figure 1. Demo of the Grip Detector

Most of the time, this grip detector stays in low power mode to save power. The two buttons remain ineffective until a valid grip event is detected. When a grip is present, if the left button (L_BUTTON) is pressed, the LED 1 will blink. The same is true of the right button (R_BUTTON) and LED 2. This grip detector might be used in a handheld device that must remain in an ultra-low power state until the user grips it.

Each of the touch sensors in this design, including the grip sensors and buttons, are based on capacitive methods. They generate an electric field that couples with a nearby ground to measure capacitance. Any object that disrupts the field can be detected by way of detecting the change in capacitance. The intrinsic capacitance between the sensor and ground, while considering the influence of the environment, is treated as the baseline. The introduction of a grip event will bring a parallel capacitance to the earth ground, which will change the measurement results relative to the baseline. The principle for capacitance measuring is to try to let the capacitive change to be as dominate as possible.

This application report describes a whole process for creating and designing a capacitive touch grip detector. It starts with the introduction of the capacitive measurement methods (see [Section 2](#)), working through the hardware design (see [Section 3](#)), firmware implementation (see [Section 4](#)) and finally shows the performance of the whole design. Besides this application report, Texas Instruments has provided various capacitive touch solutions, as well as abundant design materials and tools, such as the:

- Capacitive Touch Software Library (<http://www.ti.com/tool/capsenselibrary>)
- Capacitive Touch BoosterPack (<http://www.ti.com/tool/430boost-sense1>)
- TouchPro GUI (<http://www.ti.com/tool/msptouchprogui>)

All of these help customers develop their products faster and easier.

2 Capacitive Measurement Methods

MSP430 devices achieve capacitive touch sensing by establishing an oscillation on a conductive electrode and measuring the frequency of that oscillation. The relaxation oscillator can be created either by using the built-in pin oscillator (PINOSC) touch peripheral or by using the on-chip comparator together with some external resistors.

TI mainly supports two measurement methods based upon a relaxation oscillator: the RO method (measures the number of electrode oscillations within a fixed gate time) and the fRO method (measures the number of SMCLK oscillations during a variable gate time determined by the electrode oscillation). The choice of method depends upon the application and the microcontroller selected. In the grip detector design, the built-in PINOSC oscillator is used to create the oscillation and RO is selected as the measurement method.

2.1 Principle of PINOSC

The PINOSC implementation of the relaxation oscillator replaces the comparator and reference circuitry with a Schmitt trigger, the input of a digital IO, and the output of an internal inverter. The feedback path to the RC filter is accomplished with the inverter controlling the pullup/down polarity. The schematic of PINOSC is illustrated in Figure 2. The PINOSC is a built-in feature in all the MSP430G2xx2 and MSP430G2xx3 devices, which allows each capacitive sensor to be directly connected to a pin without any external components. The input of the PINOSC is the irregular oscillation and the output is a regular square wave, which is directly delivered to, and can be identified by, an MSP430 timer module.

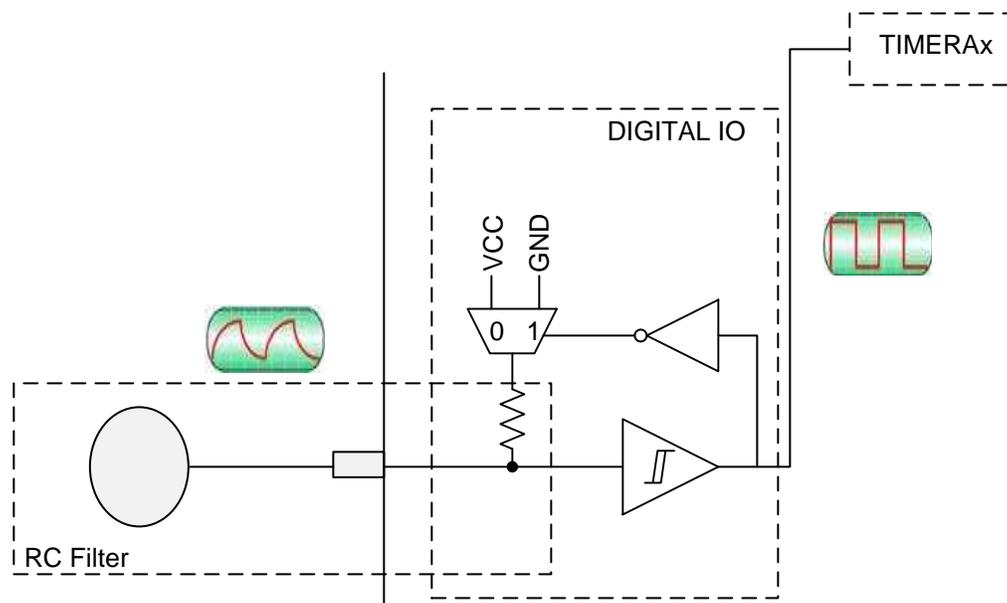


Figure 2. Schematic of PINOSC in MSP430G2xx Devices

2.2 Relaxation Oscillator

The RO method counts the number of relaxation oscillator cycles within a fixed period of time (gate time), as shown in Figure 3. In this specific design, TimeBase 1 is connected to the output of PINOSC circuit demonstrated in Figure 2, and TimeBase 2 is connected to an internal MSP430 system clock such as the DCO or VLO. The gate time is the number of the system clock oscillations.

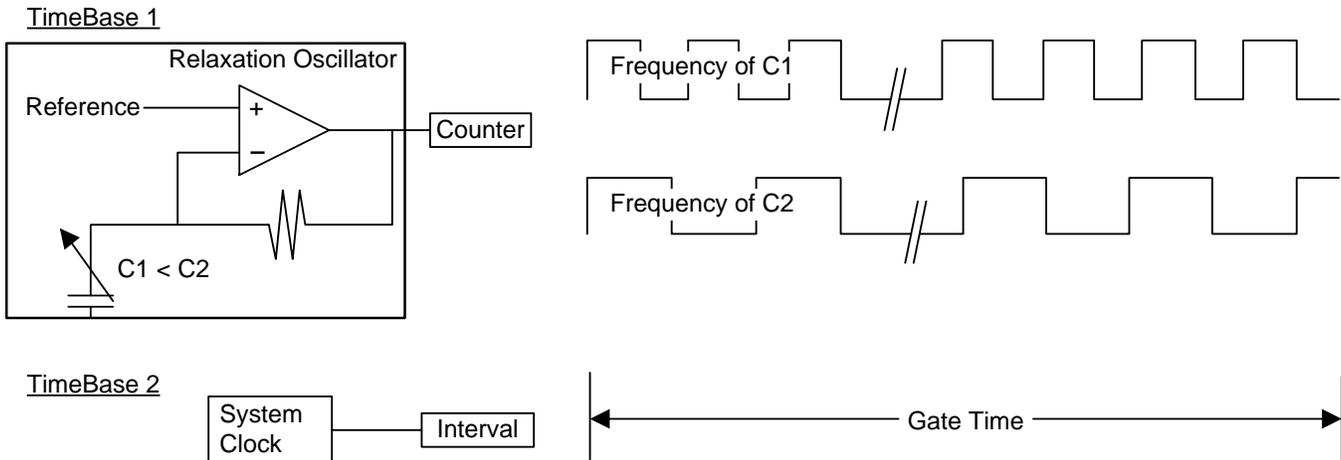


Figure 3. RO Capacitive Measurement Method

The frequency of the relaxation oscillator is the function of the capacitance to be measured. If the capacitance increases (from C1 to C2) due to a touch event, the oscillation frequency decreases, and so do the relaxation oscillator cycles counted within the gate time. The change in counts is obtained by the internal TIMER and can be interpreted by the MSP430. For most touch applications, it is not the actual capacitance that is of interest, but the change in capacitance due to a human interaction.

For more information of measurement methods, see the *Capacitive Touch Software Library Programmer's Guide (SLAA490)*.

3 Hardware Design

The grip detector is implemented as a BoosterPack to the MSP430 Launchpad. The LaunchPad is a simple development tool, the functionality of which is easily expanded via its BoosterPack interface. [Figure 4](#) illustrates the top and bottom view of the grip detection BoosterPack. Two grip sensors and two buttons are placed on the top side to detect capacitive touch events. Meanwhile, two LEDs are added on the same side to provide visual feedback to you that a grip event happened, and which button was pressed. A universal asynchronous receiver/transmitter (UART) interface is also reserved for future expansion.



Figure 4. Top and Bottom View of the Grip Detection BoosterPack

The BoosterPack is connected to the MSP430 LaunchPad via two female connectors soldered on the bottom side. Beneath the connector is a battery pack. This means that the grip detector can either be powered via the USB cable attached to the LaunchPad, or by battery.

Four screws are applied to fasten the whole BoosterPack to the enclosure. The schematic and PCB layout of the BoosterPack are shown in [Appendix A](#). The rest of this section describes the BoosterPack design in detail.

3.1 Mechanical

The main principle for capacitive sensing is to make the capacitance change induced by touch to be as dominant as possible. From the perspective of mechanical design, there may be some factors that exert influence on the overall sensitivity of the system. For example, the sensor placement, the thickness and material of the PCB as well as that of the enclosure, the gaps between the capacitive sensor and the area where the touch takes place and so on.

As shown in [Figure 4](#), the MSP430 LaunchPad and the grip detection BoosterPack are fastened into a plastic enclosure. The overall length, width and height of the enclosure are 5.63 inch, 3.25 inch and 1.15 inch, respectively, the relative dielectric of ABS is 2.4 to 4.1 while the thickness of its top cover is 110 mil. The 64 mil thick 2 layer BoosterPack PCB is made of FR4 material, the dielectric of which is around 4.8. These parameters exert a big influence on the touch sensitivity, but are a function of the system requirements and are not easily changed. What you can do is try to minimize the distance between the sensors and the area of interaction to achieve the best sensitivity. Following the principle in this application, the PCB board is designed intentionally to fit the size of the enclosure and the two grip sensors are placed as close as possible to the edge of PCB board to reduce the distance. Because of the thickness of the enclosure and how the Launchpad and BoosterPack are put into it, big gaps, nearly 480 mil exist between the two capacitive buttons and the inner side of the enclosure's top cover. This negatively influenced the sensitivity of the buttons when touched, which is a great example of the importance of improving the sensitivity from mechanical level. The cross-sectional view of the grip detector is shown in [Figure 5](#).

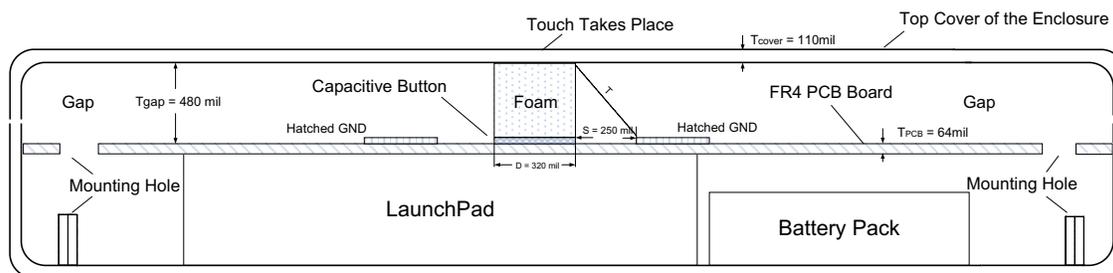


Figure 5. Cross-Sectional View of the Grip Detector

The gap must be filled or bridged with non-conductive filler (typically adhesive) or conductive extension. When the gap is big, in excess of 70 mil, a conductive extension, either foam or metal should be used. In this specific situation, foam is preferred because it is malleable to the shape of the surfaces and can prevent the formation of gaps. As shown in [Figure 5](#), the area created by the foam in contact with the button is now the area that influences the capacitance. And, the sensitivity of the two buttons can be significantly improved.

3.2 PCB Layout

Besides the mechanical aspects of the design, the PCB layout also has significant influence on the touch sensitivity. As previously mentioned, two grip sensors, with the width of 80 mil and the length of 4335 mil each, are placed as close as possible to the edge of the PCB board. The spacing between the two grip sensors and the edge of the PCB is 6 mils, which is the minimum distance available by PCB manufacturers. Even so, there are still some small gaps between the two grip sensors and the enclosure, which should be taken into consideration when designing firmware and setting capacitive touch parameters.

The size of the button is typically limited to the size of the interaction area. A small sensor area will not be able to maximize the touch performance, while a sensor area that extends outside of the overlap of the interaction area is essentially wasted, and doesn't contribute to the changing capacitance desired. In this application, the diameter of each button is 320 mil, which is nearly the size of a human finger.

The PCB boards with larger ground pour area will have better noise immunity, but this can also increase in baseline capacitance, decreasing sensitivity of the electrodes. Therefore, a 25% hatched ground is used instead of solid fill near the two capacitance buttons to reduce the area and, consequently, the parasitic capacitance while maintaining a certain noise isolation capability. Attention should be paid that enough spacing be allowed between the area where the button is pressed and the hatched ground, so that the e-field can propagate up and through the top cover of the enclosure.

A minimum spacing of one-half the laminate thickness has been found to provide sufficient sensitivity, which means that in this application the equivalent spacing T (drawn in dotted line in [Figure 5](#)) should be at least twice as much as T_{cover} (thickness of the top cover of the enclosure). So, $T = 406$ mil and $T_{cover} = 110$ mil can completely satisfy this spacing requirement.

4 Firmware Implementation

After introducing the hardware design of this grip detector, this section describes the software implementation. MSP430 is about ultra-low-power performance, so finding the best possible combination of performance and power consumption is a key component of this application. It should be taken into consideration in the entire firmware design flow.

4.1 Capacitive Touch Library

For quickly getting an MSP430-based capacitive touch solution up and running, TI offers the Capacitive Touch Library (CAPSENSELIBRARY), which enables several different capacitive touch sensing algorithms on the MSP430 platform. To use the library, it is helpful to have a basic understanding of the measurement methods, how peripheral resources are used, and the API function calls. The measurement methods and how peripheral resources are used were already introduced in [Section 2](#). The rest of this section explains in detail how the API function calls are used and how to configure sensors to make the firmware run.

For more detailed information, see the *Capacitive Touch Software Library Programmer's Guide* ([SLAA490](#)).

4.1.1 API Function Calls

The software library provides several customizable layers of abstractions, which is discussed in *Getting Started With the Capacitive Touch Software Library* ([SLAA491](#)). The Hardware Abstraction Layer (CTS_HAL) has functions for implementing sensing techniques using different peripheral sets, while the Capacitive Touch Layer (CTS_Layer) contains high level abstract functions for implementing sensor structures. The higher levels of abstraction provide standard controls for faster and easier development, while the lower levels allow for customization and unique controls.

Important API calls used in this application include *TI_CAPT_Init_Baseline()*, *TI_CAPT_Update_Baseline()* and *TI_CAPT_Custom()*, which belong to the CTS_Layer. The baseline initialization function determines the noise floor of every sensor element and *TI_CAPT_Update_Baseline* is used at power up to establish a baseline, based upon multiple measurements. The *TI_CAPT_Custom()* function contains a baseline tracking algorithm and measures the change in capacitance relative to the baseline for each element within the sensor. All three functions will call the *TI_CAPT_Raw()* function, which measures the appropriate sensor and provides the raw capacitance measurement to the application layer.

The CTS_HAL has functions for implementing sensing techniques using different peripheral sets. In this application, RO_PINOSC_TA0_WDTp belonging to this layer is used. This means the relaxation oscillator is implemented with PINOSC, TIMER_A0 is used to measure frequency of the oscillator, and the WDTp is used to set the gate time. WDTp interval represents the measurement window while the number of counts that have accumulated during the measurement window represents the capacitance of the element.

4.1.2 Sensor Definition

There are two main files that serve as the means to configure the software library: *structure.c* and *structure.h*. *structure.c* includes definitions of the elements and the sensors (groups of elements). The *structure.h* file makes the definition in *structure.c* visible to the other portions of the library and also uses precompiler definitions to enable functions and limit code size.

Two grip sensors and two buttons are defined as two sensors, which is shown in the following code example.

```
// Grip Sensors Definition
const struct Sensor Grip =
{
    .halDefinition = RO_PINOSC_TAO_WDTp,
    .numElements = 2,
    .baseOffset = 0,
    .arrayPtr[0] = &Left,
    .arrayPtr[1] = &Right,
    .measGateSource = GATE_WDT_SMCLK,
    .accumulationCycles= WDTp_GATE_8192
};
// Buttons Definition
const struct Sensor Buttons =
{
    .halDefinition = RO_PINOSC_TAO_WDTp,
    .numElements = 2,
    .baseOffset = 2,
    .arrayPtr[0] = &But1,
    .arrayPtr[1] = &But2,
    .measGateSource = GATE_WDT_SMCLK,
    .accumulationCycles = WDTp_GATE_8192
};
```

In the above code, `halDefinition` identifies which measurement implementation is being used for the sensor, while `accumulationCycles` is used to define the timer interval. Using the button sensor as an example, each element in this sensor is measured with the RO method for a period of 8192/SMCLK. As the frequency of the SMCLK is 8 MHz, this means that the scan time of each element is 1.024 ms.

Sensors are not difficult to define, but the parameters need to be set correctly. This should take both sensitivity and power consumption into consideration and are analyzed in detail in [Section 4.2](#).

4.2 Low-Power Considerations

The grip detector is suitable for battery-powered applications. Therefore, power consumption should be paid great attention during the whole design process. Giving sensors proper parameters and ensuring that the design stays in low power mode as long as possible are both effective ways to reduce power consumption.

4.2.1 MSP430 Low-Power Features

The ultra-low-power nature of the MSP430 includes characteristics like multiple low-power operating modes, wide power supply range, zero-power always-on brownout (BOR), extremely low pin leakage, and so on. The MSP430G2xx series is no exception; it can work in five low power modes. For detailed power statistics [7] (LPM0 ~ LPM4), see the device-specific data sheet.

A 16-MHz digitally controlled oscillator (DCO), as well as a very-low-power oscillator (VLO) is integrated. The DCO is a high-precision oscillator that can support wide range of frequency settings. The frequencies of 16 MHz, 12 MHz, 8 MHz and 1 MHz are calibrated and the calibration data are stored in the information Flash. The VLO is a typically about 12 KHz oscillator that is ideal for applications where power consumption is the highest priority. It is worth stressing that, with a 8 MHz DCO clock, it takes only 1 μ s for the micro to wake up from LPM3, which enables the MCU change working status between different power modes flexibly.

4.2.2 Different Operation Modes

The MSP430's various low power modes provide the possibility to create different operation modes for the grip detector. Two operation modes are implemented: active mode and idle mode.

In idle mode, the micro mainly stays in LPM3 with a clock source of nearly 12 KHz VLO. The MCU wakes up nearly every 999 ms (as defined by IDLE_DELAY) time to detect if any grip event happens. Buttons will not be scanned in idle mode. The purpose of idle mode is to conserve as much power as possible by running in a low frequency clock and reducing the activity of capacitive touch measurements.

When a grip event is detected, the MCU will exit idle mode and move into active mode. During active mode, besides the two grip sensors, two capacitive buttons will also be measured and the capacitive change will be calculated. Every ACTIVE_DELAY time interval, the micro will scan four elements (two grip sensors and two buttons) at a time in active mode. When not executing the scan, the MCU will stay in LPM3 to save power. If you grip the enclosure and click the button, the corresponding LED will blink. If the grip gestures are no longer present, the buttons will lose their control of the LEDs, the micro will exit active mode and enter idle mode again. Additionally, when measuring the capacitance in the active mode, the measurement window will be created by a WDTp sourced by 8 MHz SMCLK, the number of relaxation oscillators will be counted by TIMER_A0 during this period of time. In order to save more power, the micro will enter LPM0 until interrupt wakes it up to read the results from corresponding TIMER_A0 registers.

4.2.3 Balance Power Consumption and Sensitivity

There is a tradeoff between low-power consumption and the sensitivity of the grip detection. On one hand, it is desirable to reduce the scan time of each sensor as much as possible to allow the MCU to stay longer in LPM3 mode. However, a certain degree of sensitivity is required and this justifies sacrificing some low-power performance.

Scan time is mainly determined by the gate time, which is decided by measGateSource and accumulationCycles in sensor definition. Increasing the frequency of SMCLK and decreasing the WDTp_GATE_NUMBER, for example, changes it from WDTp_GATE_8192 to WDTp_GATE_512 and reduces the gate time as well as the power consumption. However, it reduces sensitivity because the number of relaxation oscillator counts are also decreased. Although great attention was paid to the hardware design, small gaps still exist between the grip sensors in the PCB board and the edge of enclosure where the grip gesture takes place. This illustrates that a proper gate time should be configured to have balance between sensitivity and power consumption. For more detailed information on how to set up a proper gate time, see the *Capacitive Touch Sensing, MSP430™ Button Gate Time Optimization and Tuning Guide* ([SLAA574](#)).

4.3 Firmware Design Flow

The MSP430 TouchPro GUI is a PC-based tool that can be used to verify capacitive touch designs. The tool receives and visualizes capacitive touch sensor data sent from the MCU via a universal asynchronous receiver/transmitter (UART). Interfaces are reserved in the firmware to communication with the GUI to help you quickly and easily evaluate and diagnose your designs.

An ELEMENT_CHARACTERIZATION_MODE is designed in the firmware. If this mode is enabled, you can observe and set parameters, such as measGateSource, accumulationCycles, and threshold. This mode is recommended to be used at the very beginning of the design. Once parameters are set, this mode is no longer needed, and the real application can run. The flowchart that identifies how the firmware works is demonstrated in [Figure 6](#).

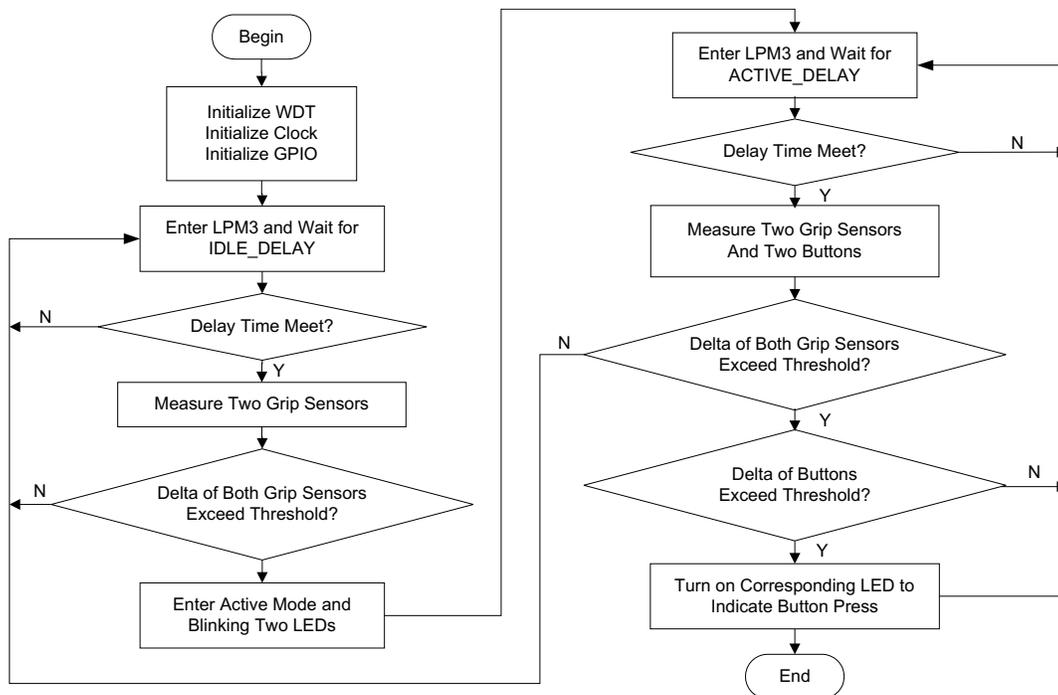


Figure 6. Firmware Design Flow

When the system is powered on, some modules like WDTp, GPIO and Clock will be initialized. Then, it enters idle mode to save power. During idle mode, the device does not respond to button presses, and every IDLE_DELAY time the MCU wakes up to see if any grip is detected.

To cause a grip condition, both grip sensors must return a value greater than their threshold to avoid false positives. When a grip is triggered, the micro enters active mode and the two LEDs blink three times. Then firmware begins to measure both grips and sensors as well as both buttons. If you press one of the capacitive touch buttons, the corresponding LED will light up. If both grip sensors are no longer above their threshold while in active mode, the device automatically returns to idle mode to save power. The whole process is exactly as mentioned in [Section 4.2.2](#).

5 Power and Performance

This section discusses the power consumption as well as performance of the grip detector designed in this application report. The current consumed in various operation modes during different working stages are measured and the average current of the idle mode is calculated. Further, the code size in various situations, the noise immunity and the sensitivity of the capacitive buttons with and without foam is elaborated in the performance section.

5.1 Overall Power Consumption

As mentioned previously, in the idle mode, the MCU is sourced by the VLO clock, which operates at around 11 KHz. This means if you set `IDLE_DELAY` as 10989, it will take nearly 999 ms for the MCU to wake up from LPM3 each time and measure the two grip sensors, the time consumed in this stage is shown as T_6 in [Figure 7](#). When it wakes up from LPM3, the micro will do some preparation before measuring, as T_1 demonstrated, and then it enters LPM0 to save power while waiting for the results. Once the results are available, interrupt wakes the micro from LPM0, read out the result, save corresponding registers, and prepare for measuring another grip sensor, as T_2 shows. After the two sensors are both measured, the results are saved and the capacitive change will be calculated, as T_5 shows. Then, the MCU enters LPM3 again.

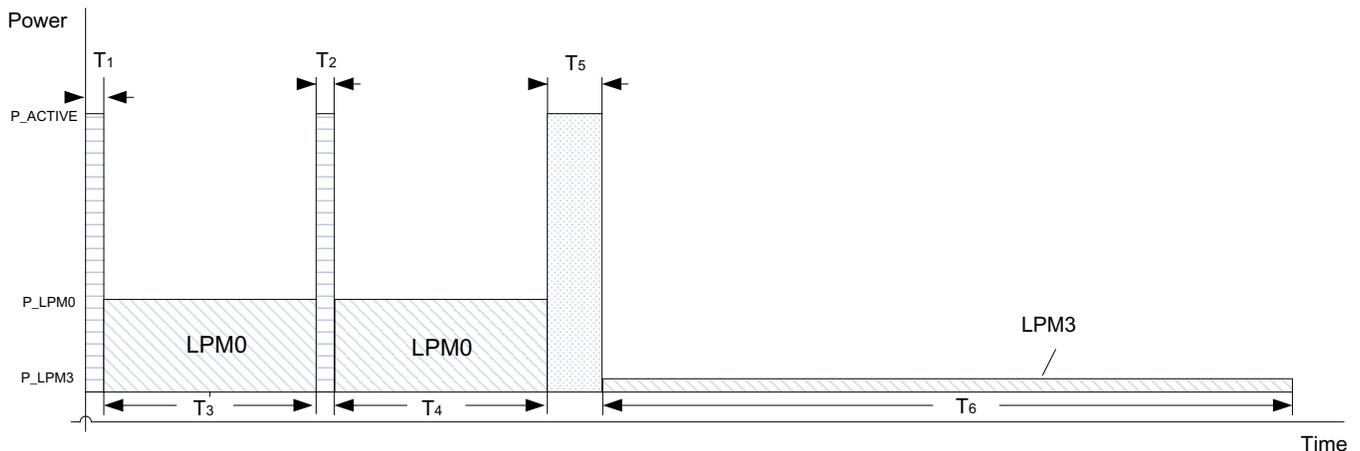


Figure 7. Current Profile in Idle Mode

The measurement gate is `WDTp` and the clock source is the 8 MHz `SMCLK`. Since `accumulationCycles` is configured as `WDTp_GATE_8192` in the sensor definition, it will take 1.024 ms, as T_3 and T_4 in [Figure 7](#), for scanning each element in this grip sensor; this time periods occupy 0.2% duty cycle in the whole idle mode working process for both sensors.

As mentioned before, the gate requires 8192 clock cycles with a clock frequency of 8 MHz. Therefore, the total measurement time for each button is 1.024 ms, which is exactly the same as the grip sensor. In active mode, if you set `ACTIVE_DELAY` as 1100, the two grip sensors and two buttons are measured at every turn every nearly 100 ms. This means this time period in which lots of current is consumed occupies a duty cycle of approximately 3.9%.

The current profile in idle mode is also measured using a logic analyzer, as illustrated in [Figure 8](#). The above one demonstrates the processing when the MCU wakes up from LPM3 every nearly 999 ms to measure the two grip sensors while the below picture extend the time stamp when the two grip sensors are measured, which is exactly the same as what was described in [Figure 7](#).

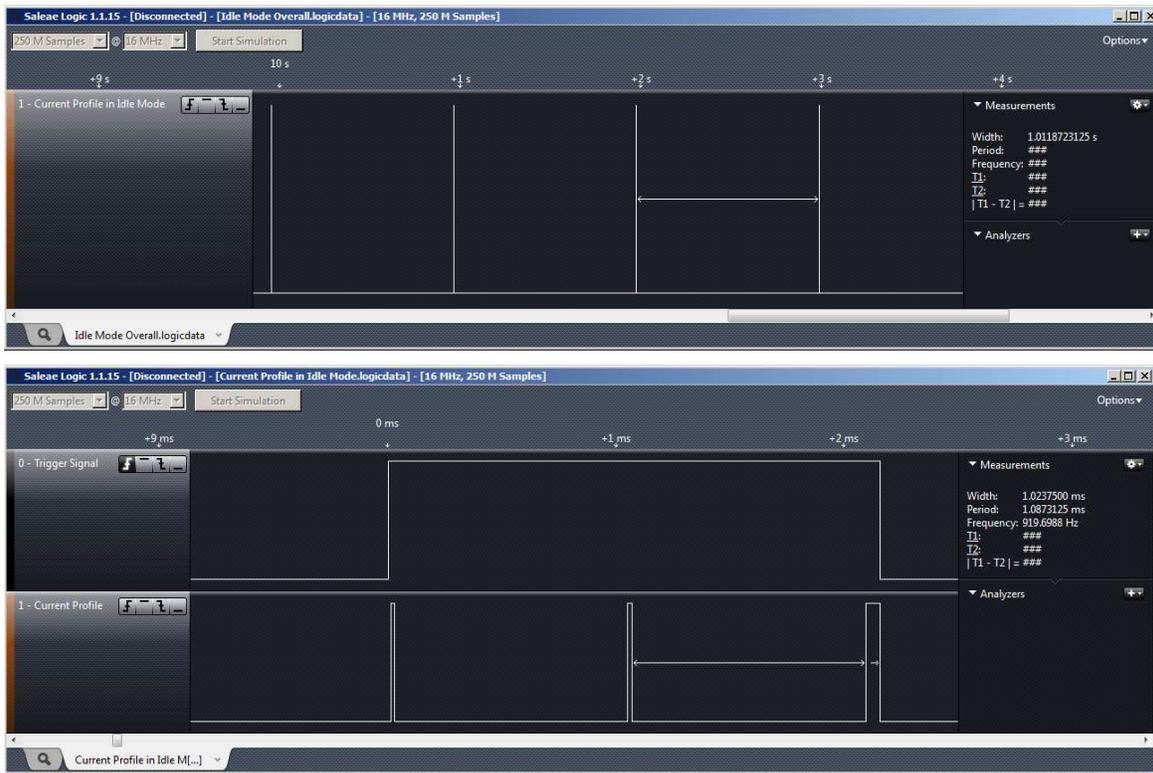


Figure 8. Screenshots of the Current Profile in Idle Mode

Table 1 lists the average current measured in different working stage in idle mode. The MSP430G2553 device is used. All the experiments are run and the measurement results are obtained at room temperature with the power supply of 3.3 V. The current in T6 is not only consumed by the low power mode itself but also the TIMER_A who is enabled and waiting to wake the micro up when IDLE_DELAY is satisfied.

Table 1. Average Current Measured in Different Working Stages

Time	Duration	Clock Source	Average I	Description
T1	12 μ s	8 MHz SMCLK	683.7 μ A	I_ACTIVE
T2	15 μ s	8 MHz SMCLK	683.7 μ A	I_ACTIVE
T3	1.024 ms	8 MHz SMCLK	341.4 μ A	I_LPM0
T4	1.024 ms	8 MHz SMCLK	341.4 μ A	I_LPM0
T5	64 μ s	8 MHz SMCLK	683.7 μ A	I_ACTIVE
T6	999 ms	11 KHz VLO Clock	0.54 μ A	I_LPM3

From Table 1, it is easy to calculate the average current consumed by this grip detector working in idle mode, which is about 1.3 μ A. When it comes to the active mode, it is not easy to measure the average current because it depends on how often the grip event happens and how many times the capacitive buttons are clicked.

5.2 Performance

5.2.1 Code Size

The amount of code space consumed by the library is largely a function of the number of elements, the number of sensors, the measurement method, and the level of abstraction. As mentioned in [Section 4.1.1](#), high-level API function calls were used directly from the library, and the code size of the real functional code is 2304 bytes while that of the code used for characterization is 2312 bytes.

Table 2. Code Size in Different Usage Conditions

Code Size (Bytes)	CHARACTERIZATION_MODE		NON_CHARACTERIZATION_MODE	
	With GUI	Without GUI	With GUI	Without GUI
FLASH	2324	2132	2496	2304
RAM	112	112	112	112

NOTE: Code Composer Studio™ v5.4.0 with Compiler v4.1.5 was used to create and the example projects.

[Table 2](#) summarizes the code size in different usage conditions. It costs an extra 192 bytes to integrate the GUI interface into the firmware; this speeds up the develop process at the cost of some Flash size. You can choose the usage conditions according to the MSP430 device they used and at which develop stage they are in.

5.2.2 Noise Immunity

The signal is defined as the change in capacitance, resulting in a meaningful change in counts. Noise, on the other hand, is any disturbance that does not change the capacitance itself but does change the counts. Most often, these disturbances are the result of power supply switching noise, electrostatic discharge (ESD), electrically fast transients (EFTs), radiated noise or some other type of electrical noise that couples into the system.

Table 3. Threshold and Delta Counts When Touched and Non_touched

	L_GRIP	R_GRIP	L_BUTTON	R_BUTTON
TOUCHED (MIN)	54	78	18	20
NON_TOUCHED	0/1	0/1	0/1	0/1
THRESHOLD	27	39	9	10

[Table 3](#) shows the threshold and number of counts of the four electrodes when they are touched and not touched. The delta counts of the left and right buttons are measured when foam is used to fill the gap.

Experimentation results show that the maximum noise level on the two buttons when not been touched was 0 or 1 count, and the delta counts reflecting the capacitive change are relatively large than system noise level. Taking pre-setting threshold into consideration, test results show a fairly good signal-to-noise ratio.

5.2.3 Effect of the Foam

As is mentioned in [Section 3.1](#), at the mechanical level, foam is used to fill the gap between the two capacitive buttons and the top cover of the enclosure to improve the sensitivity of the two capacitive buttons. Performance of the two capacitive buttons with and without foam is measured and illustrated in [Table 4](#).

Table 4. Performance With and Without Foam

	With Foam		Without Foam	
	L_BUTTON	R_BUTTON	L_BUTTON	R_BUTTON
TOUCHED	18	20	3	4
NON_TOUCHED	0/1	0/1	0/1	0/1

The typical delta values for the L_BUTTON and R_BUTTON without foam is 3 and 4, respectively. But when foam is added, the corresponding delta counts are increased to 18 and 20. This means the touch effects of the two buttons are more easily affected by the noise without foam. Therefore, improved sensitivity from the level of hardware is very important and useful, because it will cost more power and effort from a firmware perspective, if the same effects are expected.

Parameters like the threshold may be changed if a different enclosure or foam materials are used, or if the environment is significantly different. In these situations, ELEMENT_CHARACTERIZATION_MODE is helpful to pre-set new parameters, and attention should always be paid to both hardware and firmware design.

6 Conclusion

This application report provides general guide lines for designing and developing a capacitive grip detector. With the PINOSC features integrated in MSP430G2xx2 and MSP430G2xx3 devices, no extra capacitance or resistor is needed to generate relaxation oscillator. Hardware and firmware implementations are discussed in details to demonstrate how to minimize the average power consumption while maintaining a certain degree of sensitivity. Measurement results show that the average current consumed by this grip detector in idle mode can achieve as low as about 1.3 μ A.

It is also worth highlighting that you can make use of the capacitive touch software library and TouchPro GUI provided by TI, as well as all the schematic, PCB and Gerber files released with this report. Together, these tools make designs much faster and easier.

7 References

1. *PCB-Based Capacitive Touch Sensing With MSP430 User's Guide* ([SLAA363](#))
2. *Capacitive Touch Software Library Programmer's Guide* ([SLAA490](#))
3. *Capacitive Touch Hardware Design Guide* ([SLAA576](#))
4. *Capacitive Touch Sensing, MSP430™ Button Gate Time Optimization and Tuning Guide* ([SLAA574](#))
5. *Getting Started With the Capacitive Touch Software Library* ([SLAA491](#))
6. *MSP430x2xx Family User's Guide* ([SLAU144](#))
7. *MSP430G2x53/MSP430G2x13 Mixed Signal Microcontroller Data Sheet* ([SLAS735](#))
8. SL-53P SL Series Slender Profile ABS Electronics Enclosures: <http://www.polycase.com/sl-53p>

Appendix A Schematic and PCB Layout of the Grip Detection BoosterPack

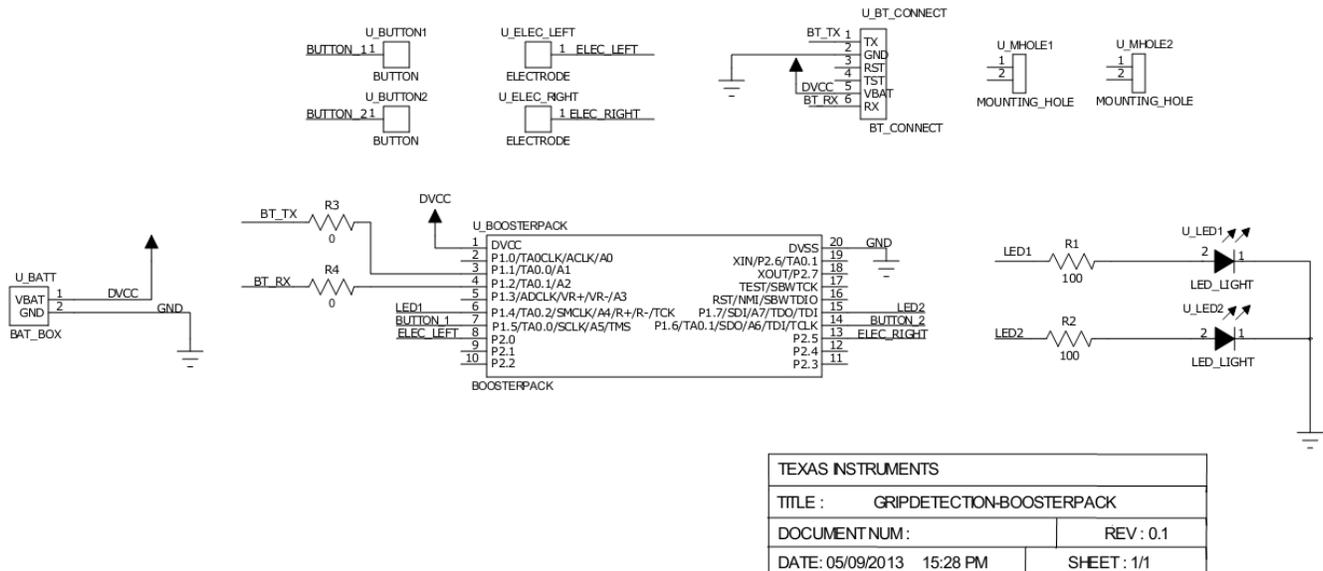


Figure 9. Schematic of the Grip Detection BoosterPack

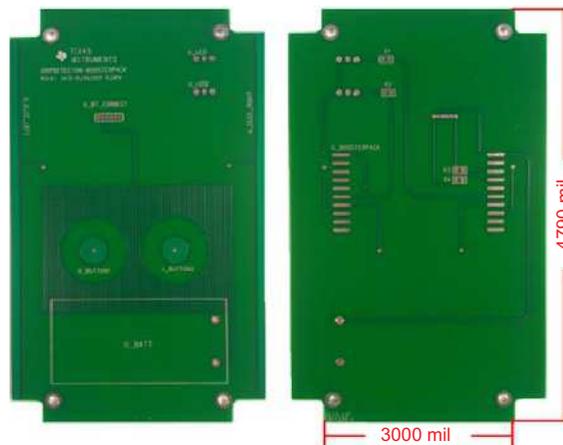


Figure 10. PCB Layout of the Grip Detection BoosterPack

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

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com