TI TechNotes

Tamper Detection Using Low-Memory MSP430™ MCUs

TEXAS INSTRUMENTS

Introduction

Security threats come in many forms and vary widely in their goals. They may include attempting to trick a sensor into collecting incorrect data such as for electronic meters or trying to open a secure location to gain access to proprietary hardware or information. This tampering can take many forms including applying heat or magnetic fields from the outside of a system to alter measurements or physically opening an enclosed container.

Properly enabled physical tamper detection can trigger defensive measures when an attacker is attempting to obtain or modify secure assets for unintended purposes. Some defensive measures after tamper detection include sounding an alarm, deleting sensitive data, or making a unit inoperable until a trusted source can repair it.

This document focuses on detecting when a secured enclosure is opened. The described system provides tamper detection and evidence, and typically is used as the first line of defense in system applications. Use cases could include detecting if a product (for example, meters, thermostats, or e-locks) was opened, and tamper evidence could be used for warranty voidance or further investigation. More information on security threats, physical attacks, and typical measures can be found in System-Level Tamper Protection Using MSP MCUs.

Open/close tamper detection has been implemented using different strategies in the past. Buttons may be placed along the edge of a box opening that are held pressed when the box is closed. However, due to the mechanical nature of the buttons, they may become stuck and fail to trigger when a box is opened.

Inductive sensors placed along the hinge next to a magnet or metal object, can measure a changing inductance when the box is opened, triggering detection. Externally applying a magnetic field may be able to fool the sensor when the box is opened to avoid detection.

The method outlined in this document implements open/close tamper detection by outputting a clock signal from a microcontroller (MCU) pin, reading the outputted signal into a separate input pin of the MCU, and counting the number of edges on the incoming signal over a consistent period of time. A real-time clock (RTC) can be used to set the time interval over which the number of edges is counted. If the signal is disconnected for any extended period of time, tampering has occurred, and the count at the end of a time interval will be incorrect. The value on a separate pin can then be toggled to signal the host processor to take defensive action. Because code does not always execute at the same speed, there will be a small variation to the actual counted value over the time interval that is unrelated to tampering. Therefore, a tolerance should be added when the value is checked to avoid indicating a false tampering event. The RTC and outputted signal should be clocked from the same source so clock inconsistencies will affect both in the same way and reduce the amount of tolerance needed.

The MSP430FR2000 MCU contains an RTC that can be sourced by an auxiliary clock (ACLK) and is able to output the ACLK on a pin. Additionally, the MCU has GPIOs available to count the edges of the clock signal and raise an alert signal to a host processor. The MCU is also able to overwrite persistent variables in FRAM to delete sensitive data in response to a tampering event. This project has been optimized to minimize the likelihood of missing a tampering event or triggering a false warning for the method described as well as fit within the 0.5 KB of memory in the MCU. To get started, download project files and a code example demonstrating this functionality.

More information about MSP security features can be found in Understanding MSP430 MCU Security Features Overview and MSP Code Protection Features.

Implementation

This application uses the MSP430FR2000 MCU with the MSP-TS430PW20 target development board. The MCU firmware implements the tamper detection strategy described in the introduction. It is possible to customize the clock source and count time interval to better suit the system according to the designer's needs.

Pin P1.1 outputs the ACLK signal and pin P1.0 counts the number of edges. Pin P2.0 is initially set to 0 when the MCU is powered on and, if a tamper event is detected, pin P2.0 is raised from 0 to 1. The value of P2.0 is set to 1 only during the time period after an interval in which the count was incorrect. If the count is correct in a following time period after an incorrect interval count, the value of P2.0 will toggle back from 1 to 0. External oscillator circuitry may be added to P2.6 and P2.7 to improve clock accuracy. Figure 1 shows an example of how the tamper detection hardware could be implemented in a system. The ACLK signal runs to the edge of the hatch and crosses the boundary with interlaced copper fingers or a similar mechanism. When the box is opened, the copper separates, and the ACLK signal disconnects, triggering detection. Care should be taken when designing the system to ensure the external oscillator and ACLK lines are not easily accessible from the outside of the box.

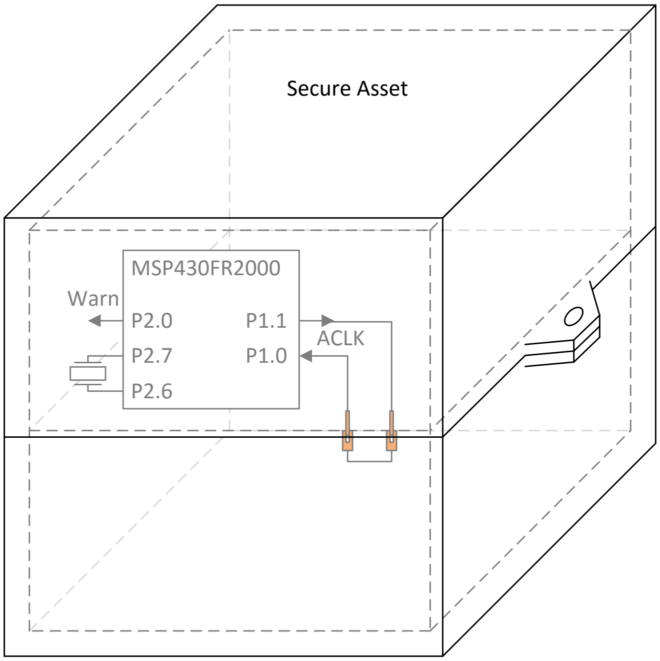
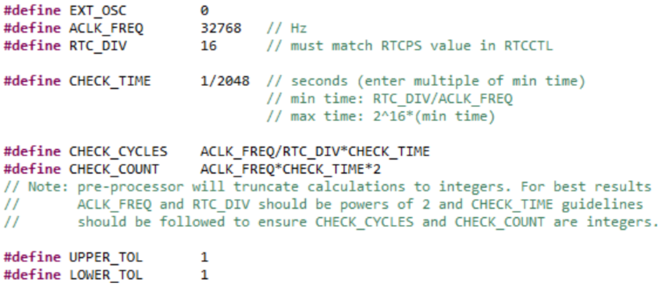


Figure 1. Hardware Connection Diagram

The MCU RTC is timed by ACLK which is clocked by reference oscillator (REFO) (32.768 kHz) by default. Additionally, an external crystal oscillator can be used by configuring the macros at the top of the source code. If an external oscillator is used, EXT_OSC should be changed to 1 and ACLK_FREQ should equal the external oscillator frequency.

There are also macros defining the time period (in seconds) for which the count is checked. The time interval should be entered as a multiple of a minimum fraction of whole numbers next to CHECK_TIME (for example, 1/2048, 1/8, 3/16, 1, or 75/32). The minimum fraction value is dependent on the RTC prescaler defined in RTC_DIV and RTCCTL. RTC_DIV and the RTC prescaler value should always match each other. RTC_DIV is set at the top of the code in a macro, and the RTC prescaler is set in RTCCTL in the main program. Figure 2 shows where the macros are located at the top of the code, and Figure 3 shows how to set RTCCTL in the main program.



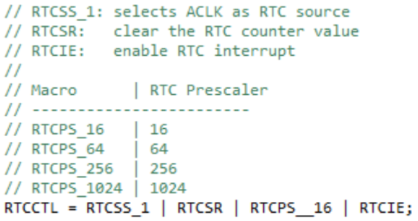
```
#define EXT_OSC         0
#define ACLK_FREQ       32768   // Hz
#define RTC_DIV         16      // must match RTCPS value in RTCCTL

#define CHECK_TIME      1/2048  // seconds (enter multiple of min time)
                                // min time: RTC_DIV/ACLK_FREQ
                                // max time: 2^16*(min time)

#define CHECK_CYCLES    ACLK_FREQ/RTC_DIV*CHECK_TIME
#define CHECK_COUNT     ACLK_FREQ*CHECK_TIME*2
// Note: pre-processor will truncate calculations to integers. For best results
//       ACLK_FREQ and RTC_DIV should be powers of 2 and CHECK_TIME guidelines
//       should be followed to ensure CHECK_CYCLES and CHECK_COUNT are integers.

#define UPPER_TOL       1
#define LOWER_TOL       1
```

Figure 2. Macro Configuration



```
// RTCSS_1: selects ACLK as RTC source
// RTCSR:   clear the RTC counter value
// RTCIE:   enable RTC interrupt
//
// Macro       | RTC Prescaler
// -------------------------
// RTCPS_16    | 16
// RTCPS_64    | 64
// RTCPS_256   | 256
// RTCPS_1024  | 1024
RTCCTL = RTCSS_1 | RTCSR | RTCPS__16 | RTCIE;
```

Figure 3. RTCCTL Configuration

CHECK_CYCLES is the calculated number of clock cycles the RTC should count down to match the time period set in CHECK_TIME. CHECK_COUNT is the ideal value that the count for each time interval should match and depends on ACLK_FREQ and CHECK_TIME.

Table 1 summarizes the macro values for each RTC prescaler choice. System designers interested in responding quickly to a tamper event can choose to set the macros to correspond to the values in the row for RTCPS__16.

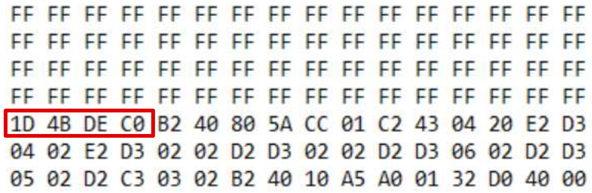
Table 1. Standard Macro Values for Different RTC Prescalers

RTCCTL	RTC_DIV	CHECK_TIME	
		Minimum Time Interval Fraction	Maximum Time Interval
RTCPS__1024	1024	1/32 s (31.25 ms)	2048 s (≈34 minutes)
RTCPS__256	256	1/128 s (7.81 ms)	512 s (≈8.5 minutes)
RTCPS__64	64	1/512 s (1.95 ms)	128 s (≈2 minutes)
RTCPS__16	16	1/2048 s (0.49 ms)	32 s

The value of UPPER_TOL and LOWER_TOL sets how much the count may differ from the ideal value and can also be adjusted, as needed, to avoid triggering false tamper warnings.

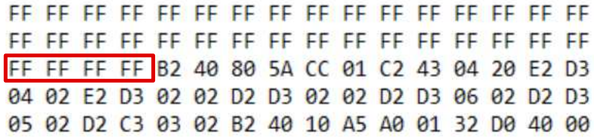
The system clock is set to 16 MHz so that the firmware can respond to the RTC and port interrupts as quickly as possible.

When the firmware is loaded onto the MCU, the persistent variable called secureData is loaded with value 0xC0DE4B1D into FRAM to act as sensitive data. The first time a tamper warning occurs, the firmware erases the stored value, and the value remains erased even after device reset until the firmware is reloaded onto the MCU. Figure 4 and Figure 5 show the MCU memory before and after the erase occurs.



```
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1D 4B DE C0 B2 40 80 5A CC 01 C2 43 04 20 E2 D3
04 02 E2 D3 02 02 D2 D3 02 02 D2 D3 06 02 D2 D3
05 02 D2 C3 03 02 B2 40 10 A5 A0 01 32 D0 40 00
```

Figure 4. Protected Data Stored in FRAM



```
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF B2 40 80 5A CC 01 C2 43 04 20 E2 D3
04 02 E2 D3 02 02 D2 D3 02 02 D2 D3 06 02 D2 D3
05 02 D2 C3 03 02 B2 40 10 A5 A0 01 32 D0 40 00
```

Figure 5. Protected Data Erased From FRAM

Performance

The feature uses an MSP430FR2000 MCU. To run the demo, connect the hardware as previously described, load the code into the device, allow the device to run, and end the debug session. For demonstration purposes, the Warn signal was connected to an LED so that it would light up when a tamper event is detected.

For the host processor to respond quickly to a tamper event, the RTC prescaler was set to 16 and the time interval was set to 1/2048 s. During testing, it was discovered that the count variable was very consistent except for the first time interval. This is reasonable behavior because during initialization the RTC and GPIO input interrupts must be enabled in different commands, making the count differ from subsequent time intervals after both are enabled. Therefore, a flag variable was created to ignore the first time interval. The subsequent time intervals had the following range:

CHECK_COUNT – 1 ≤ count ≤ CHECK_COUNT + 1

UPPER_TOL and LOWER_TOL were set to 1 and the above inequality was implemented in the firmware count check. Because the count signal is sourced by a clock with a frequency of 32.768 kHz and the tolerance allows for a discrepancy of up to 3 counts, the firmware is able to detect if the signal is disconnected for more than 46 µs and alert the host processor with a delay of no more than 490 µs. The maximum delay to the host processor will match the CHECK_TIME interval.

Tamper detection can be further improved by outputting randomly generated sequence on P1.1, instead of the ACLK. Additionally, the outputted signal could be read on more pins than P1.0 to allow for more crossings along the box hatch. Testing should be performed by system designers to ensure that the interrupts can respond quickly enough to changing input on multiple pins and no false tampering events are triggered. Using the RTC, timestamps could be generated and written to FRAM each time a tampering event is detected to indicate when they occurred.

Device Recommendations

The device used in this example is part of the MSP430 Value Line Sensing portfolio of low-cost MCUs, designed for sensing and measurement applications. This example can be used with the devices shown in Table 2 with minimal code changes. For more information on the entire Value Line Sensing MCU portfolio, visit www.ti.com/MSP430ValueLine.

Table 2. Device Recommendations

Part Number	Key Features
MSP430FR2000	0.5KB FRAM, 0.5KB RAM, eComp
MSP430FR2100	1KB FRAM, 0.5KB RAM, 10-bit ADC, eComp
MSP430FR2110	2KB FRAM, 1KB RAM, 10-bit ADC, eComp
MSP430FR2111	3.75KB FRAM, 1KB RAM, 10-bit ADC, eComp

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI’s provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI’s standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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