

Add Housekeeping Functions to Your MSP430™ MCU: ADC Wake and Transmit on Threshold



Many applications, such as battery monitors and thermostats, require sampling analog signals periodically so action can be taken based on the behavior of those signals. Analog-to-digital converters (ADCs) can be triggered with precise timers to provide a solution to this requirement. This consumes a lot of microcontroller (MCU) resources and also causes higher power consumption as the timer needs to be active with the ADC. Alternatively, many ADCs allow signals to be monitored continuously, but doing so can also require high power consumption. Operating the ADC integrated in the MCU independently of the CPU allows the CPU and all clocks other than the one being used by the ADC to be disabled to save power. The ADC also includes logic to determine conditions that require the CPU and the rest of the MCU to wake up. This is achieved by the ADC generating an interrupt to wake up the CPU when a configurable threshold is crossed. In addition, interrupts can be used to also send out ADC conversion data serially without having to power on the CPU.

Implementation

This implementation uses a UART to set an ADC threshold value, which when crossed will result in waking up the CPU, setting a GPIO with LED, and transmitting ADC data back over the UART. A simple graphical user interface (GUI) was developed and supports displaying the ADC data as well as configuration the threshold over UART. The MSP430FR2433 waits in a low power mode while still able to continuously monitor the analog signal using the internal ADC.

For clocking in this solution, ACLK is sourced from internal trimmed low-frequency oscillator (REFO) which frequency is 32768Hz. REFO is fed into the FLL for the DCO, which goes through a software trim routine to set SMCLK and MCLK both to 1 MHz. For more info on DCO Software Trim, refer to [section 3.2.11.2 of the MSP430FR2xx Family User's Guide](#). ACLK is used as the source clock of 10-bit SAR ADC and Timer_A. The source clock for the universal serial communication interface (eUSCI) UART is the 1MHz SMCLK.

ADC samples are taken continuously from P1.3 (A3) with each conversion taking place immediately after the preceding one has finished, with a periodicity of 2.7307 kHz. After configuring all necessary peripherals, the device goes into low-power mode 0 (LPM0). The ADC continually monitors the analog input to the MCU in this mode, while the CPU and all clocks other than ACLK and SMCLK are tuned off to reduce power consumption. When the set voltage threshold is reached by the analog input to the ADC, P1.0 is set to indicate that the threshold has been reached, and a timer is started to send out the ADC conversion results over UART at a defined interval.

Figure 1 shows the block diagram for this implementation.

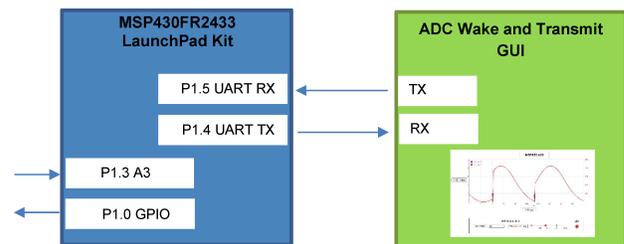


Figure 1. Implementation Overview

The MSP430FR2433 LaunchPad™ Development Kit should be used with this example project. back-channel UART interface on eZ-FET of the LaunchPad kit is used for UART communication with the GUI. The COM channel number information can be found in the PC device management under the control panel.

Figure 2 shows the MSP430FR2433 LaunchPad kit including eZ-FET, ADC analog input by P1.3, and an LED driven by P1.0

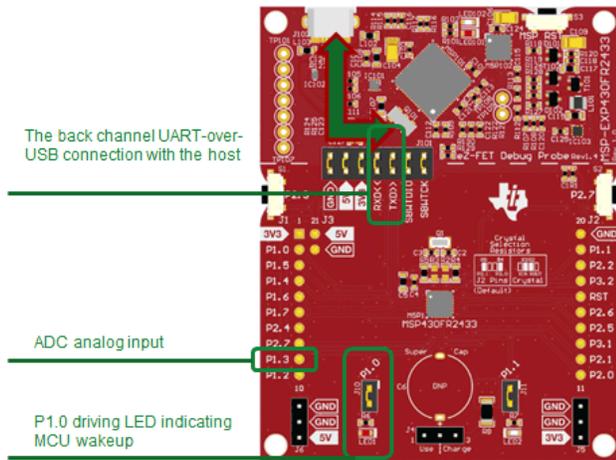


Figure 2. MSP430FR2433 LaunchPad Development Kit

The GUI is used to configure the ADC threshold value to wake the CPU and displays the ADC conversion data. The eUSCI_A0 peripheral was used in UART mode to enable commands to be received on P1.5/UCA0RXD and transmitted on P1.4/UCA0TXD. The eZ-FET inside of LaunchPad was used for evaluation. A baud rate of 9600 must be selected with one stop bit and no parity. Two hexadecimal inputs must be sent to the MCU individually. The first specifies the high byte of the ADC threshold value, and the second specifies the low byte. The maximum value that can be represented by the ADC is 0x03FF, and the minimum is 0x0000. All ADC values are represented by unsigned 16-bit integers. The ADC window comparator threshold value is stored in FRAM, so it is retained on reset of the device. When the ADC threshold value is reached, ADC conversion values are sent over UART with a period that is set by default to 1 second. To change this, modify the value of the TA0CCR0 register in the example source code. Keep in mind that the timer is using a clock of 32768 Hz. So, for example, a TB0CCR0 value of 32768 results in a timer period of 1 second. Because the ADC produces a 10-bit unsigned integer, the data is sent over UART in two bytes, with the high byte first. These UART transmissions occurred after a threshold of 0x01FF was reached.

Performance

The operation of the demo can be run as described in the implementation section regarding the use of UART to control the threshold value and receive ADC conversion values. P1.3 (A3) is used for the analog input to the ADC, and a voltage can be applied to this pin to trigger the high threshold value of the ADC.

Figure 3 shows the GUI interface with a real-time refreshing scalar graph to display the status of the

MSP430FR2433 MCU and the conversion value of the ADC. Below the scalar graph, there is a text box on the left to display the ADC conversion value in real time. This value is synchronized with the value displayed in the scalar graph. In the middle is a slider to set the ADC comparison threshold. When the MSP430FR2433 receives the threshold, the updated threshold is stored in FRAM. On the right is a LED widget that can display the status of the LED in real time.

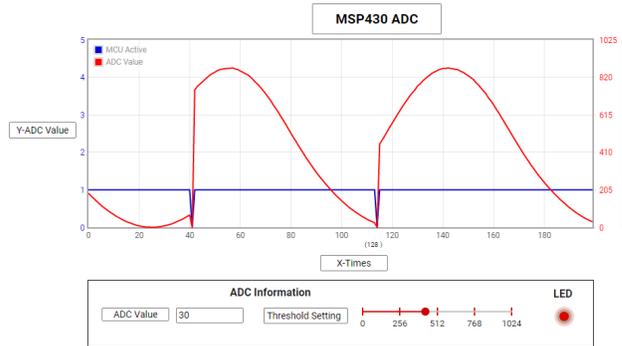


Figure 3. ADC Wake and Transmit Example GUI

To Get Started

1. Watch our [training video “ADC Wake and Transmit on Threshold with a Housekeeping MCU”](#), where we walk through how to use a GUI to adjust the ADC threshold.
2. [Order a MSP430FR2433 LaunchPad kit](#) to evaluate the ADC Wake and Transmit GUI and example code.
3. Download and test this example with the [ADC Wake and Transmit example GUI](#), where you can change the ADC threshold.
4. Evaluate the [ADC Wake and Transmit example code](#) for the MSP430FR2433 LaunchPad kit.

Device Recommendations

The device used in this example is part of the MSP430 housekeeping portfolio. This example can be used with the devices shown in [Table 1](#) with minimal code changes.

Table 1. Device Recommendations

Part Number	Key Features
MSP430FR2433	16KB FRAM, 4KB SRAM, 10-bit ADC, UART/SPI/I2C, Timer
MSP430FR2422	8KB FRAM, 2KB SRAM, 10-bit ADC, UART/SPI/I2C, Timer

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2020, Texas Instruments Incorporated