Application Brief

Battery Charging Using the MSP430FR2355 MCU

Battery chargers require a combination of PWMs, ADCs, a real-time clock, and some GPIOs. The PWMs in this case are used to control up to two different charging voltages and currents (only 2 PWMs are required if only one battery will be charged at a time). The ADCs are used to sense 4 cell voltages and an integrated thermistor. The real-time clock can be used to keep track of how much time it takes to charge the battery (if the charging time exceeds some threshold, a fault should be indicated). The GPIOs are used to trigger an interrupt when over current is detected and to drive various controls and indicators implemented in the charger.

Implementation

NOTE: The implementation described includes the MCU functions but does not include the external analog circuitry that the user can define.

The firmware for the LaunchPad™ development kit takes the user-defined values, converts them into proportional PWM duty cycle waveforms, and outputs these signals to the respective PWMs. The UART communication is used to send ADC values from the LaunchPad kit to the GUI.

For clocking in this solution, ACLK is sourced from internal trimmed low-frequency oscillator (REFO) operating at a frequency of 32768 Hz. 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 information on DCO software trim, refer to section 3.2.11.2 of the *MSP430FR4xx and MSP430FR2xx Family User's Guide*. The source clock for the Timer_A0 and Timer_A1 modules is SMCLK.

Timer_B3 is configured to output varying duty cycle PWM waveforms on P6.0, P6.1, P6.2, and P6.3. These pins are tied to TB3.1, TB3.2, TB3.3, and TB3.4 respectively by selecting the secondary I/O functions for those pins. The capture/compare registers for Timer_B3 are configured so that the duty cycle is adjustable by adjusting TB3CCR1, TB3CCR2, TB3CCR3, and TB3CCR4, respectively.

The ADC is configured to capture the analog signals on P1.1, P1.4, P1.5, P1.6, P1.7, P5.0, P5.1, P5.2, and P5.3. These pins are tied to ADCINCH_1, ADCINCH_4, ADCINCH_5, ADCINCH_6, ADCINCH_7, ADCINCH_8, ADCINCH_9, ADCINCH_10, and ADCINCH_11, respectively. During acquisition, each channel is selected by the ADCMCTL0 register followed by sampling and conversion. After each conversion, code execution resumes and the subsequent channel is selected until all 9 channels have been sampled.

Three GPIOs are configured as interrupts. P3.0 and P4.0 are triggered with a high-to-low transition on the relevant header pin. P4.1 is also configured to trigger with a high to low transition when the S1 button on the LaunchPad kit is pressed.

Ten GPIOs are configured for control and LED output on P1.0, P6.6, P2.0, P2.1, P2.2, P2.4, P2.5, P3.1, P3.2, and P3.3. P1.0 and P6.6 are configured to output to the red and green LEDs on the LaunchPad kit. The P4.1 interrupt toggles the LED connected to P6.6. P6.6 is also turned off when the ADC result on P1.1 is less than 0x7FF. P1.0 toggles the RED LED on the LaunchPad kit once per second when the real-time clock triggers an interrupt. All other pins are accessible through the LaunchPad kit headers.

Pins P1.2 and P1.3 are configured for optional I^2C communication (SCL and SDA) with a battery management solution through the eUSCI_B0 peripheral.

The eUSCI_A1 peripheral was used in UART mode to enable commands to be received on P2.5/ UCA1RXD and transmitted on P2.6/UCA1TXD. The eZ-FET inside of the LaunchPad kit was used for evaluation. A baud rate of 9600 must be selected with one stop bit and no parity. The LaunchPad kit will send five 16-bit integers of data containing the ADC values in hex for the thermistor and four voltage cells. If the __ENABLE_GUI__ define is unused, then the MCU will not output these ADC values over the UART.

Although the MSP430FR2355 LaunchPad development kit is used with this example project, it can be used with other MSP430™ microcontrollers with proper code migration. Backchannel UART interface on eZ-FET of the LaunchPad kit is used for UART communication with the GUI. Figure 1 shows the system block diagram for the charger.

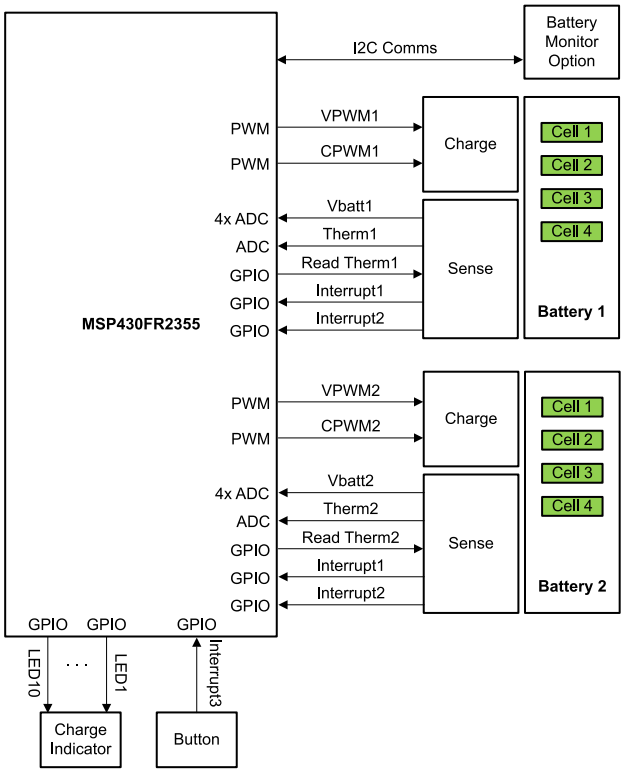

Figure 1. System Block Diagram

As can be seen from this block diagram, the voltage is sensed across 4 different cells in each battery along with a thermal sensor. Two different interrupts can also be triggered by the sensing circuit when a fault condition is sensed.

Figure 2 shows the charging circuit from Figure 1.

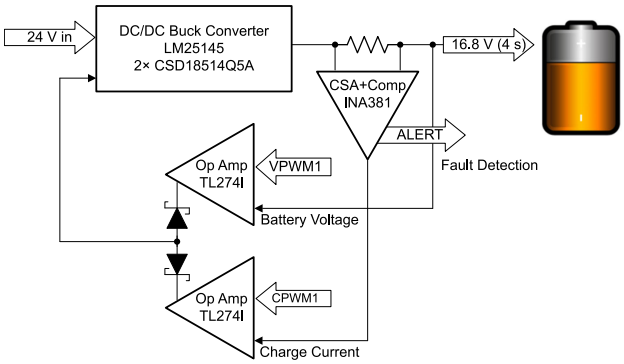

Figure 2. Charging Circuit

The feedback to the DC/DC converter is controlled by two loops, one voltage loop and one current loop. Only one of the loops is in control of the power supply at a given time. The current and voltage levels are set by the PWM outputs from the MSP430 MCU. Duty cycles between 0% and 100% are filtered to produce an analog voltage reference. Diode D1 combines the outputs of the two amplifiers with a logical OR. The voltage that is lowest is fed into an inverting amplifier that makes the error signal polarity correct for the DC/DC controller integrated error amplifier. More details on this circuit can be found in Figure 7 of Wide-Vin Battery-Charger Using SMBus MSP430 MCUs and bq Fuel Gauges.

The output of 4 ADCs, 4 PWMs, and 2 comparators can be seen in the GUI (see Figure 3). Figure 4 shows the GUI interface to configure various analog components on the MSP430FR2355. Pin P5.3 should be connected to P3.5 and pin P5.2 should be connected to P3.1 on the LaunchPad kit to see the outputs of PGA2 and PGA3 on ADC2 and ADC3. GPIO functions are not currently implemented.

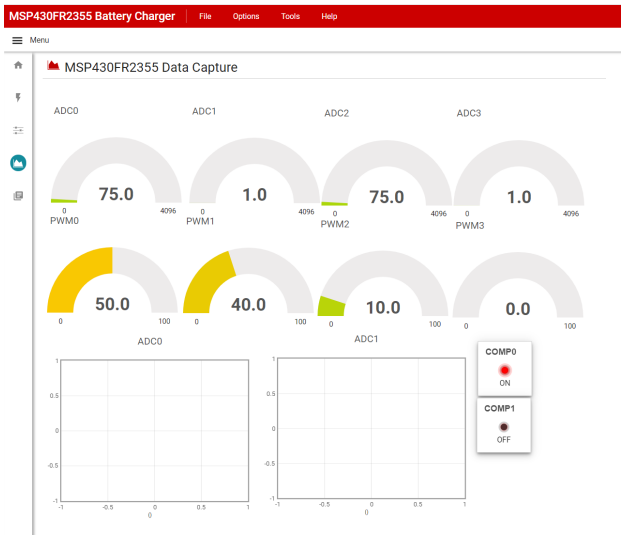

Figure 3. Battery Charger GUI

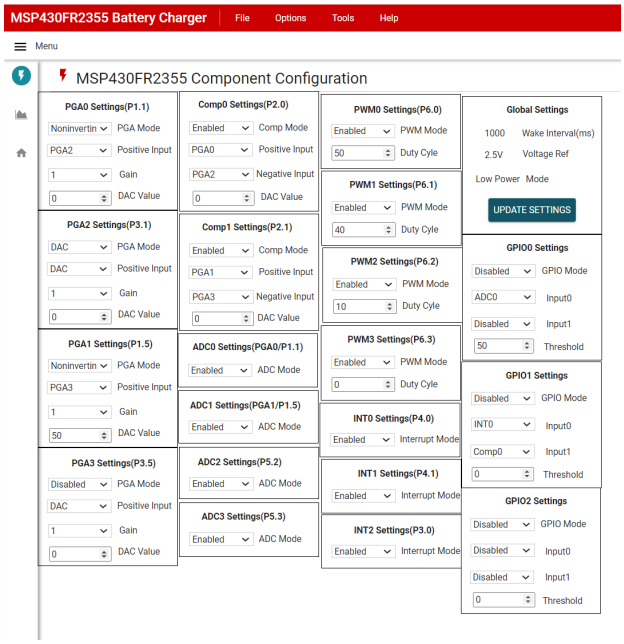

Figure 4. MSP430FR2355 Component Configuration

To Get Started

1. Order an MSP430FR2355 LaunchPad kit to evaluate the battery charger example code.
2. Download and test this example with the UART Battery Charger example GUI, where you can monitor the voltage levels of up to 4 different cells.
3. Evaluate the UART Battery Charger example code for the MSP430FR2355 LaunchPad kit.

Device Recommendations

Part Number	Key Features
MSP430FR2355	32KB FRAM, 4KB SRAM, 12-bit ADC, UART/SPI/I2C, Timer

Note: MSP430FR23x and MSP430FR215x devices are also recommended.

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 (https: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