Analog Input to PWM Output Using the MSP430™ MCU Enhanced Comparator

Introduction

An analog-to-digital converter (ADC) is a system that converts an analog signal, such as analog voltage or current to a digital number proportional to the magnitude of the voltage or current. In some applications such as lighting or DC electric motor control, the output will be a pulse width modulation (PWM) signal. The MSP430FR2000 microcontroller (MCU) does not include an ADC module. In this example, the enhanced comparator (eCOMP) is used to implement a 6-bit ADC function.

The implementation presented here demonstrates how to use the MSP430™ on-chip analog voltage comparator with an internal reference digital-to-analog converter (DAC) to measure the voltage then output PWM signals. This has been optimized for lowest code size, fitting in MSP430FR2000 MCU, which contains 512 bytes of main memory. To get started, download project files and a code example demonstrating this functionality.

Implementation

The eCOMP module compares the analog voltages at the positive (V+) and negative (V–) input terminals. In this solution the external voltage is connected to the positive input terminal and the eCOMP's built-in 6-bit DAC is connected to the negative terminal.

A GPIO (P1.3) interrupt is used for mechanical button input detection. This is used to trigger an ADC conversion and output a PWM signal from the MSP430FR2000 device output pin. A capacitor is connected to P1.3 for debouncing. The MSP430FR2000 was used with the MSP-TS430PW20 target development board and connected with wires to the mechanical buttons and voltage input as shown in Figure 1.

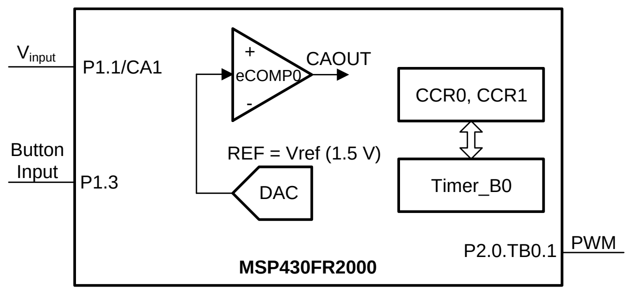

Figure 1. eCOMP ADC And PWM Output Block Diagram

This demo uses the on-chip 1.5-V VREF as the reference voltage for the DAC. The input voltage is measured by finding the correct setting for CPDACDATA. The buffer data is initialized as 0x20 which is 32 of 64 steps. If the result of the compare is positive then the buffer data needs to be increased and compared again. If the result is negative then the buffer data needs to be decreased and compared again. With this iterative process the correct buffer data value should be found after several iterations. The buffer data value is then used to generate different PWM signals to indicate the input voltage. Timer_B0 is used to generate the PWM output on P2.0, and the PWM frequency and duty cycle are set by TB0CCR0 and TB0CCR1.

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. Note that the MSP-TS430PW20 target board already includes the correct connections for the MSP-FET programmer and debugger. At startup, connect V_{input} to the external voltage that needs to be measured. Pressing the button will start the measurement and output the PWM.

The 6-bit DAC can be set to 64 levels, which gives an effective resolution for this ADC of 6 bits. In the example code, the expected PWM frequency can be translated from V_{input}.

$$f_{pwm} = 1 / (\text{Buffer Data} \times 128) \quad (1)$$

The duty cycle is always 50%. The frequency and duty cycle could be changed based on application requirements. Table 1 lists some PWM frequency examples with the V_{input}.

Table 1. Voltage to PWM

V_{input} (V)	Buffer Data	Frequency (Hz)
0	1	8000
0.5	23	340
1	43	181
1.5	63	124

Performance

Figure 2 shows an example of the analog-to-PWM function. The measured response time from a button trigger to output PWM waveforms is 5.95 ms. Because the algorithm will take the same number of cycles to find the correct Buffer data value, the response time will remain constant. In this example the V_{input} is connected to a 1.09-V power supply. After the ADC has been implemented using the eCOMP, the buffer data value is 47. The minimum resolution is 1.5 V / 64 = 0.023 V. The measured value of 47 equates to (1.5 V × 47) / 64 = 1.10 V, and 1.10 V – 1.09 V = 0.01 V, which is less than the minimum resolution.

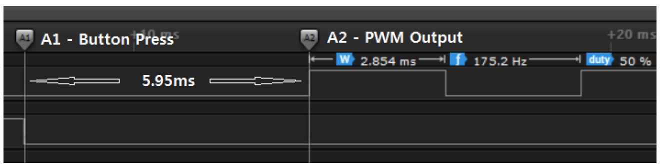

Figure 2. Analog-to-PWM Demo

The code example uses around 380 bytes of main memory. More space can therefore be allocated for additional application code like GPIO control or a lookup table for PWM output. Further increase in code development will require migrating to MSP430 MCUs with larger memory footprints.

Because SMCLK is required as the timer clock source, low-power mode 0 (LPM0) can be accessed during inactivity. If the application needs lower power consumption, LPM3 can be used which would require changing the Timer_B clock source.

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

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