Author: Troy Pandhumsoporn

MSP430G2xxx Analog Comparator

A comparator is essentially a non-feedback device and high-gain differential amplifier that compares two analog signals and outputs a one bit digital signal (logic 1 or 0) depending on which input is higher. When the non-inversing input terminal is greater than the inverting input terminal, the output goes high; otherwise, in the converse case, the output goes low. With this description, the comparator acts similar to a 1-bit ADC. Applications using on-chip comparator's involves supporting precision slope analog-to-digital conversions, battery-voltage supervision, and the monitoring of an external analog signal. The comparison voltage can be a voltage reference, or it can be setup to connect to an external analog input signal. As part of the analog peripherals included in the MSP430G2xxx Value Line Series microcontrollers, the Launchpad's 2553 and 2453 microcontrollers share the same on-chip comparator, Comp_A+. Comparator_A+ is an eight channel device with internal voltage reference for simple measurements or voltage threshold detection.

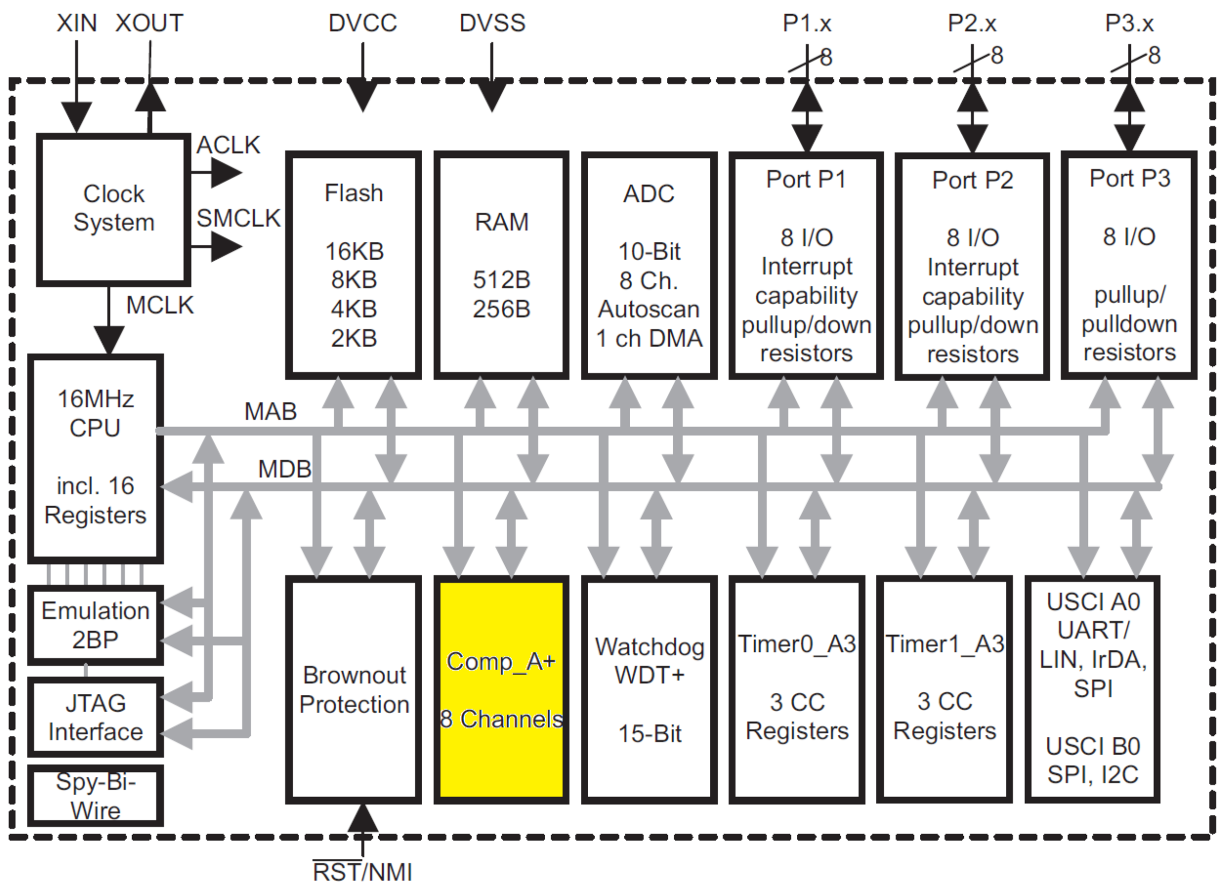


Figure 1: MSP430G2553 System Level Flowchart (Comp_A+)

Comparator_A+ Characteristics

- Inverting/Non-Inverting input terminal multiplexer
- Programmable RC-filter for the output comparator
- Comparator output supplied to Timer_A capture input

- Programmable port input buffer
- Interrupt capability
- User-Defined Reference voltage generator
- Comparator and reference generator deactivation
- Input Multiplexer

Comparator_A+ User Reference/Datasheet Specifications

MODULE	REGISTER DESCRIPTION	REGISTER NAME	OFFSET
Comparator_A+	Comparator_A+ port disable	CAPD	05Bh
	Comparator_A+ control 2	CACTL2	05Ah
	Comparator_A+ control 1	CACTL1	059h

Figure 2: Comparator Register List. These registers need to be referenced in your code in configuring the analog comparator; the CAPD register plays a role in switching off unneeded ports, reducing the contribution to circuit behavior.

Comparator_A+

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	TYP	MAX	UNIT
$I_{(DD)}$ (1)		CAON = 1, CARSEL = 0, CAREF = 0	3 V		45		μA
$I_{(Refladder/RefDiode)}$		CAON = 1, CARSEL = 0, CAREF = 1, 2, or 3, No load at CA0 and CA1	3 V		45		μA
$V_{(IC)}$	Common–mode input voltage	CAON = 1	3 V	0		V_{CC}-1	V
$V_{(Ref025)}$	(Voltage at 0.25 V_{CC} node) / V_{CC}	PCA0 = 1, CARSEL = 1, CAREF = 1, No load at CA0 and CA1	3 V		0.24		
$V_{(Ref050)}$	(Voltage at 0.5 V_{CC} node) / V_{CC}	PCA0 = 1, CARSEL = 1, CAREF = 2, No load at CA0 and CA1	3 V		0.48		
$V_{(RefVT)}$	See Figure 21 and Figure 22	PCA0 = 1, CARSEL = 1, CAREF = 3, No load at CA0 and CA1, TA = 85°C	3 V		490		mV
$V_{(offset)}$	Offset voltage (2)		3 V		±10		mV
V_{hys}	Input hysteresis	CAON = 1	3 V		0.7		mV
$t_{(response)}$	Response time (low-high and high-low)	T_A = 25°C, Overdrive 10 mV, Without filter: CAF = 0	3 V		120		ns
		T_A = 25°C, Overdrive 10 mV, With filter: CAF = 1			1.5		μs

(1) The leakage current for the Comparator_A+ terminals is identical to $I_{lkg(Px.y)}$ specification.
(2) The input offset voltage can be cancelled by using the CAEX bit to invert the Comparator_A+ inputs on successive measurements. The two successive measurements are then summed together.

Figure 3: Comparator_A+ 's specifications from the msp430g2xxx datasheets--screen-captured for easier reference.

21.3.1 CACTL1, Comparator_A+ Control Register 1

7	6	5	4	3	2	1	0
CAEX	CARSEL	CAREFx		CAON	CAIES	CAIE	CAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

- CAEX, or Bit 7, represents the bit that exchanges the comparator inputs (V- and V+) and inverts the comparator output (to compensate for the change of the inputs). This is useful in comparing the similarity of close values, but in unneeded for basic functionality
- CARSEL and CAREF (Bits 6, 5-4) are responsible for the internal comparator's voltage references. The CAREF bits choose the reference value with the CARSEL bit decides what input is connected to the reference. Three references are offered in the comparator module. Two of the references are a fraction of the supply voltage that power the MSP430G2xxx: 0.5 Vcc and 0.25 Vcc (1% accuracy). These values are respectively 1.8 V and 0.9 V for the USB powered Launchpad. The third reference is the MOSFET transistor's forward voltage, though it is usually not used due to its lower accuracy and temperature dependent characteristic.
- CAON, Bit 3, turns on/off the comparator; the associated circuitry is activated or deactivated independently
- CAIFG, CAIE, and CAIES bits (0-2) serve as the interrupt control bits of the comparator. Bit 2 decides whether an interrupt is toggled on the rising edge (low to high) or a falling edge (high-to-low). Bit 1 switches the interrupt on. Bit 0 serves as the interrupt flag, which is cleared automatically when the interrupt service routine is called upon.

21.3.2 CACTL2, Comparator_A+, Control Register

7	6	5	4	3	2	1	0
CASHORT	P2CA4	P2CA3	P2CA2	P2CA1	P2CA0	CAF	CAOUT
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

- The CASHORT bit 7 shown shorts two comparator inputs (V+ and V-) together. While this is an odd case, the comparator possibly acts as a high input impedance buffer if needed; though, the basic use of Comparator_A+ prefers neglecting this function.
- P2CAx bits (6-2) select inputs pins that connect to comparator.
- CAF bit (1) controls the whether the comparator output will be filtered or not by the 1st order, internal low-pass RC filter. This should be enabled if the signals used are gradually varying or the signal needs its rapid oscillations smoothed out, which happens when the comparators inputs are close together.
- CAOUT bit (0) is the comparator output value; this value cannot be controlled or written to—instead it gives a test check for the user to write code to read the bit to confirm the current comparator output

21.3.3 CAPD, Comparator_A+, Port Disable Register

7	6	5	4	3	2	1	0
CAPD7	CAPD6	CAPD5	CAPD4	CAPD3	CAPD2	CAPD1	CAPD0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

- The CAPD (Comparator_A+ port disable) register is responsible for manually controlling the ports disconnection. Each bit corresponding to input pins CA0-CA7 can individually disable the digital circuitry for the GPIO ports from the pins being used for analog signals related with Comparator_A+. If an analog signal near the digital circuit's transition voltage is applied to the pin, the digital section can oscillate between the high and low thresholds fairly rapidly, causing circuit degradation in the chip. The digital section of this register should be disconnected when analog signals are used. To use the comparator on multiple signals, one pin can be connected to the comparator at a time; however, the input connection can be switched using software.

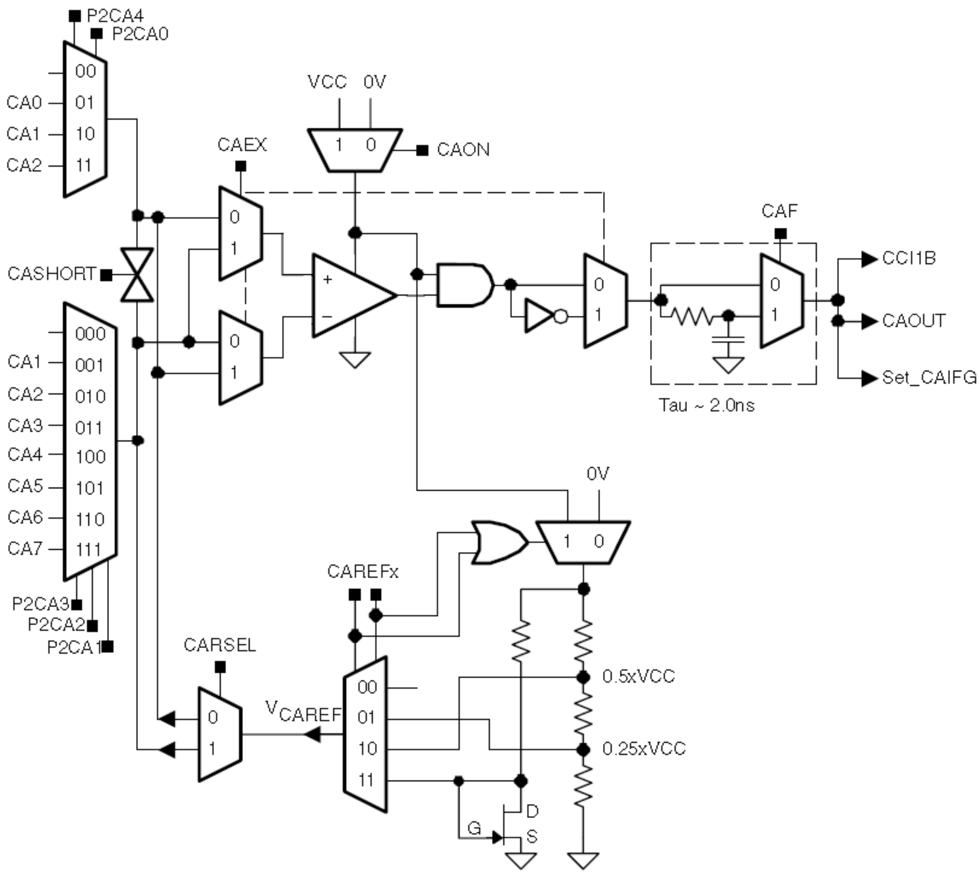


Figure 4: Comparator_A+ Block Diagram

The most common application for the analog voltage comparator remains a simple threshold detector, often seen as usually the critical low battery indicator in portable devices (such as battery operated laptops and cell phones). An emulation exercise is provided near the end of the document.

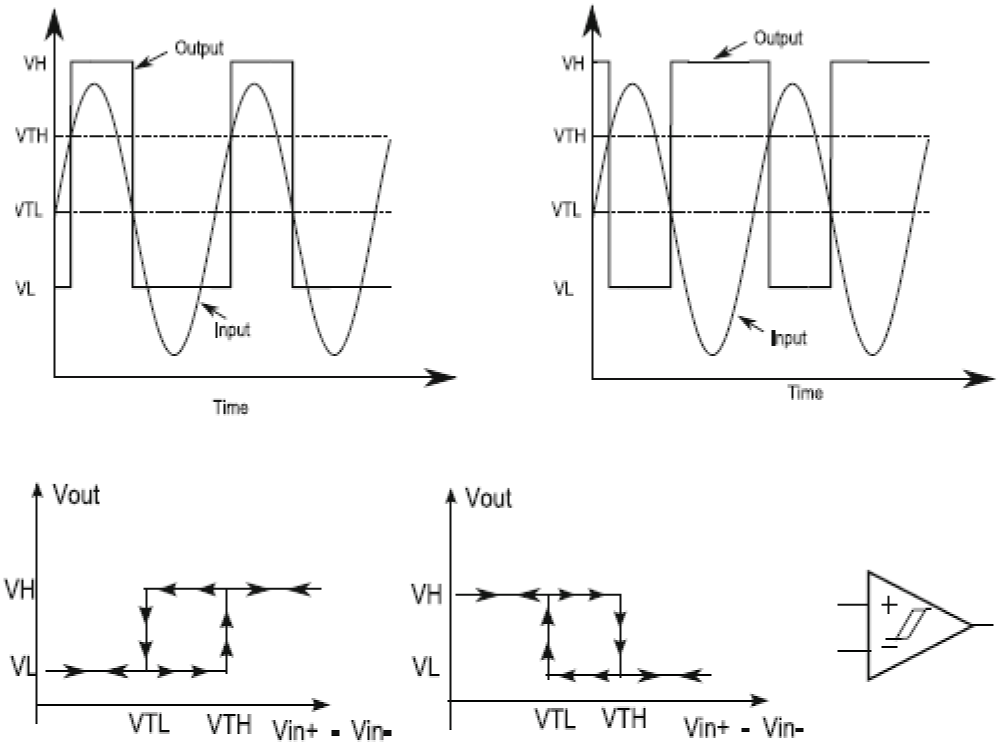


Figure 5: Hysteresis threshold comparator (non-inverting, left; inverting, right)

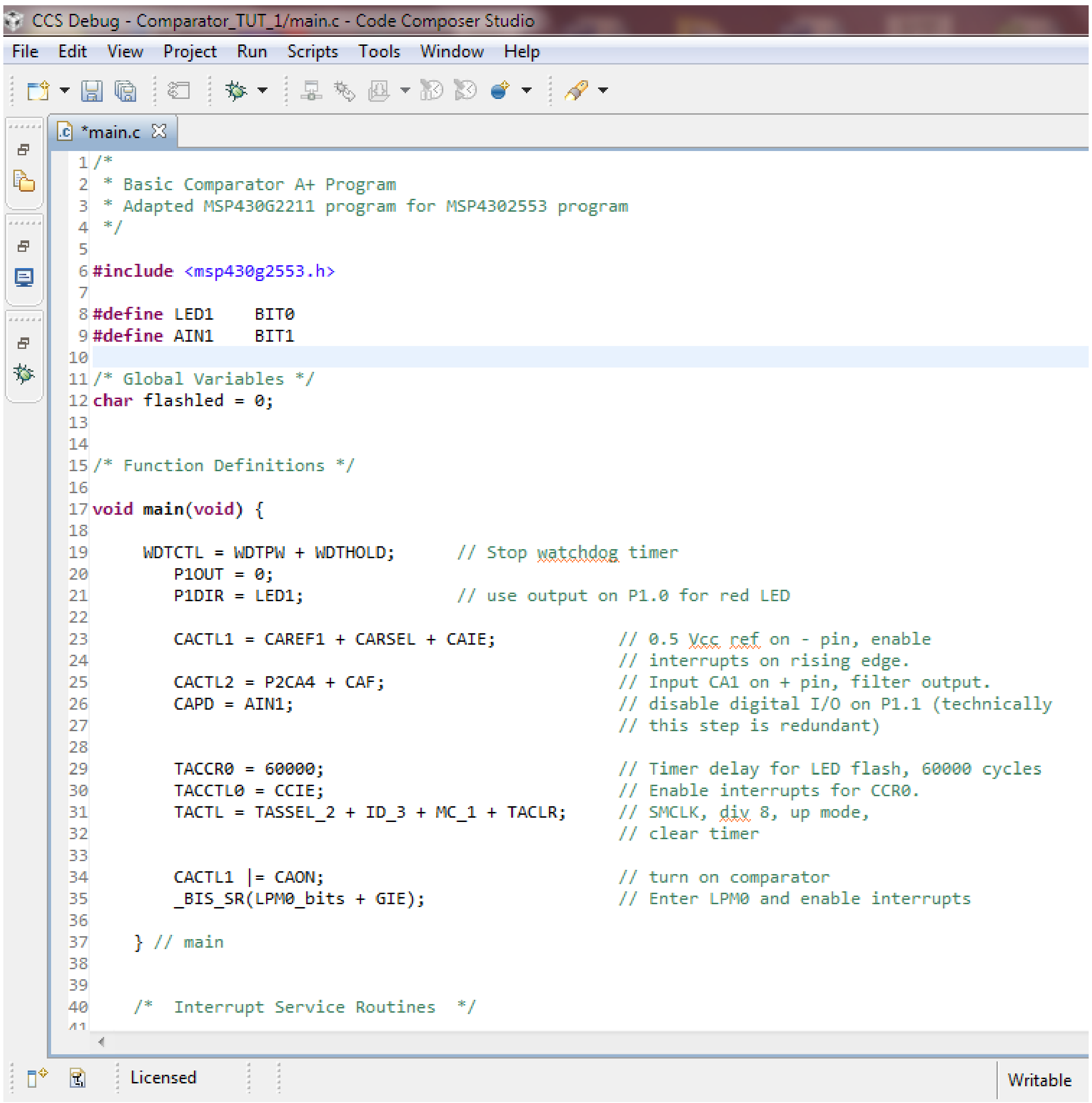
Despite the comparator operating with usually small absolute differences between the two inputs, it still experiences stability and oscillation problems; this event also occurs when the comparator processes gradually varying signals with small amounts of noise. Although noisy signals are to be expected, the comparator operation can be improved using hysteresis and proper filtering to limit the oscillation. Hysteresis comparator circuits involve the transition between states of different threshold variables V_{IH} and V_{IL}. When the output of the non-inverting comparator stays low, V_L, it will remain there until the input difference “V_{in+} - V_{in-}” value increases and approaches the high threshold V_{TH}—the output changes state to the high value of V_H. The next instance for changing the threshold to the low state V_L is $V_{TL} < V_{TH}$. The hysteresis variable is given by V_{TH}-V_{TL}

Author: Troy Pandhumsoporn

Comparator Sample Project Exercise:

Basic Comparator Program (Launchpad)

The program measures an analog voltage, and flashes an LED if the voltage is higher than a specific reference. For this setup using the running USB power, the value is ½ Vcc = 1.8 V. The program uses the "Timer0_A0" module to flash the LED, while Comparator_A+ calls an interrupt to switch the LED to flash off and on.




```c
/*
 * Basic Comparator A+ Program
 * Adapted MSP430G2211 program for MSP4302553 program
 */

#include <msp430g2553.h>

#define LED1    BIT0
#define AIN1    BIT1

/* Global Variables */
char flashled = 0;


/* Function Definitions */

void main(void) {

     WDTCTL = WDTPW + WDTHOLD;      // Stop watchdog timer
        P1OUT = 0;
        P1DIR = LED1;               // use output on P1.0 for red LED

        CACTL1 = CAREF1 + CARSEL + CAIE;            // 0.5 Vcc ref on - pin, enable
                                                    // interrupts on rising edge.
        CACTL2 = P2CA4 + CAF;                       // Input CA1 on + pin, filter output.
        CAPD = AIN1;                                // disable digital I/O on P1.1 (technically
                                                    // this step is redundant)

        TACCR0 = 60000;                             // Timer delay for LED flash, 60000 cycles
        TACCTL0 = CCIE;                             // Enable interrupts for CCR0.
        TACTL = TASSEL_2 + ID_3 + MC_1 + TACLR;     // SMCLK, div 8, up mode,
                                                    // clear timer

        CACTL1 |= CAON;                             // turn on comparator
        _BIS_SR(LPM0_bits + GIE);                   // Enter LPM0 and enable interrupts

    } // main


    /*  Interrupt Service Routines  */
```

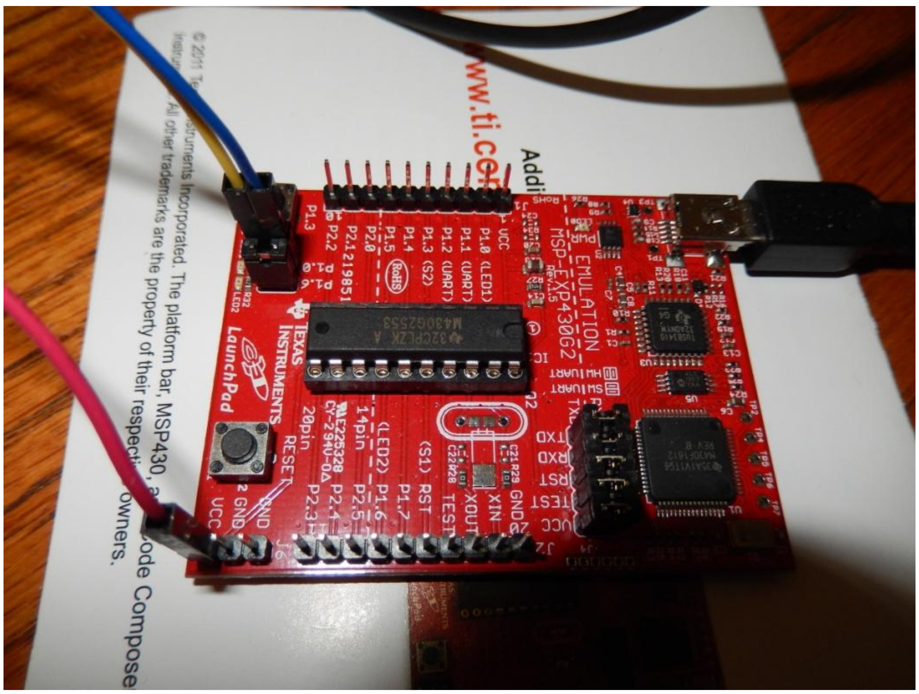
```
    /*  Interrupt Service Routines  */

// http://processors.wiki.ti.com/index.php/MSP430_FAQ#How_to_assign_the_correct_Timer_A_interrupt_vector.3F

    #pragma vector = TIMER0_A0_VECTOR
    __interrupt void CCR0_ISR(void) {
        P1OUT ^= flashled;              // if flashled is zero, keep LED off
                                    // if flashled is LED1, toggle LED
    } // CCR0_ISR

    #pragma vector = COMPARATORA_VECTOR
    __interrupt void COMPA_ISR(void) {
        if ((CACTL2 & CAOUT)==0x01) {
            CACTL1 |= CAIES;        // value high, so watch for falling edge
            flashled = LED1;            // let LED flash
        }
        else {
            CACTL1 &= ~CAIES;       // value low, so watch for rising edge
            flashled = 0;               // turn LED off
            P1OUT = 0;
        }
    } // COMPA_ISR
```

The program can be tested by using a potentiometer (I found using the 100K pot was the best for distinct results) between Vcc and ground.



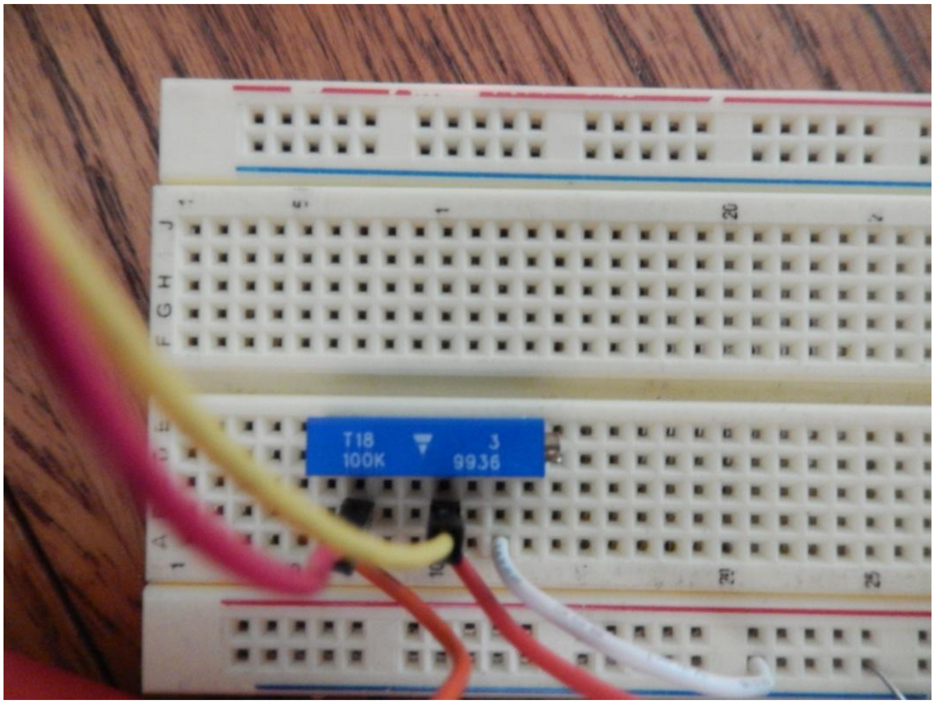


Figure 5: Hands-On Analog Comparator Testing

1. Remove the J5 jumper from the Launchpad connecting P1.0 and LED1, since an external potentiometer will have to be connected to the two pins.
2. Connect the Vcc pin (the one at the leftmost third partition) to one end of the potentiometer.
3. Connect the two jumpers from the potentiometer's wiper to the male header pins of P1.0 and LED1.
4. Ground the other end of the potentiometer. Add three more jumpers at the top end, wiper, and ground pin for multi-meter readings for monitoring the voltage variation of decreasing and increasing beyond the threshold 1.8V.

Author: Troy Pandhumsoporn

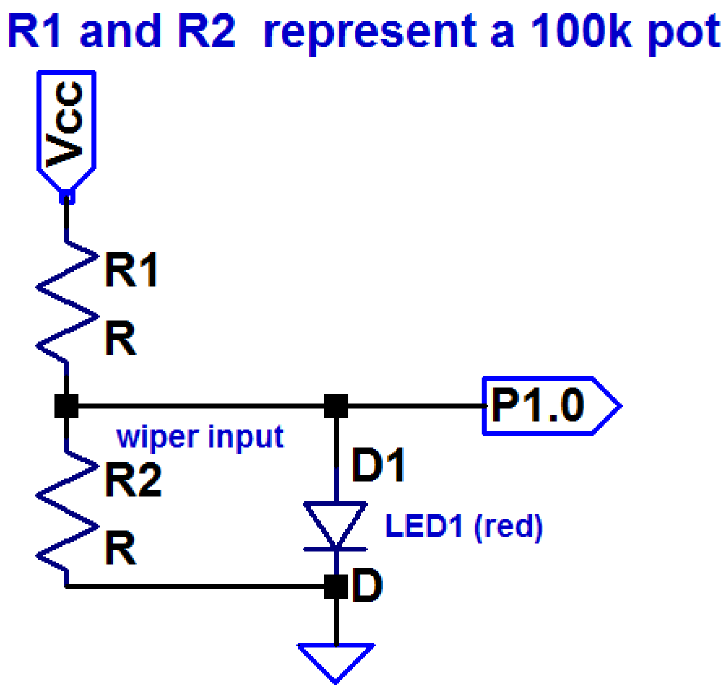


Figure 6: Schematic (LTSpice) view of the test circuit

Observations

- At 106.9K, LED1 is at minimum brightness at 1.928 V
- At 4.70K, LED1 starts to get even brighter at 1.649 V
- At 3.54k, LED1 reaches a respectable brightness level (comfortable)
- At, 0 ohms LED1 is extremely (blindingly) bright (might burn out the LED)
- At 41.3k for the 100k trimpot, 3.67V, threshold of LED1 turning on
- ~0 ohms, less than 1 V, LED1 at full brightness

Oscilloscope Screenshots (Comparator Visualization)

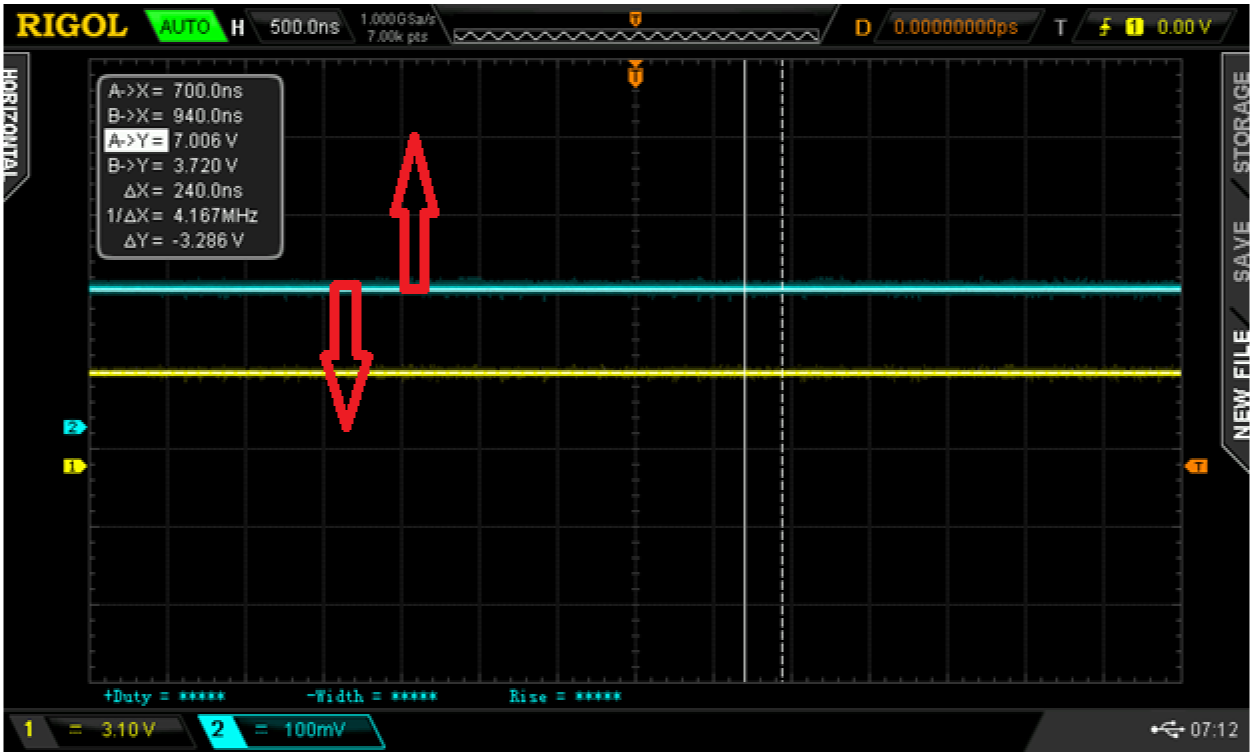


Figure 7: While keeping the same reference levels, the blue dc trace will move down or up depending respectively on the increase or decrease of the potentiometer's resistance. (41.3kΩ ref)

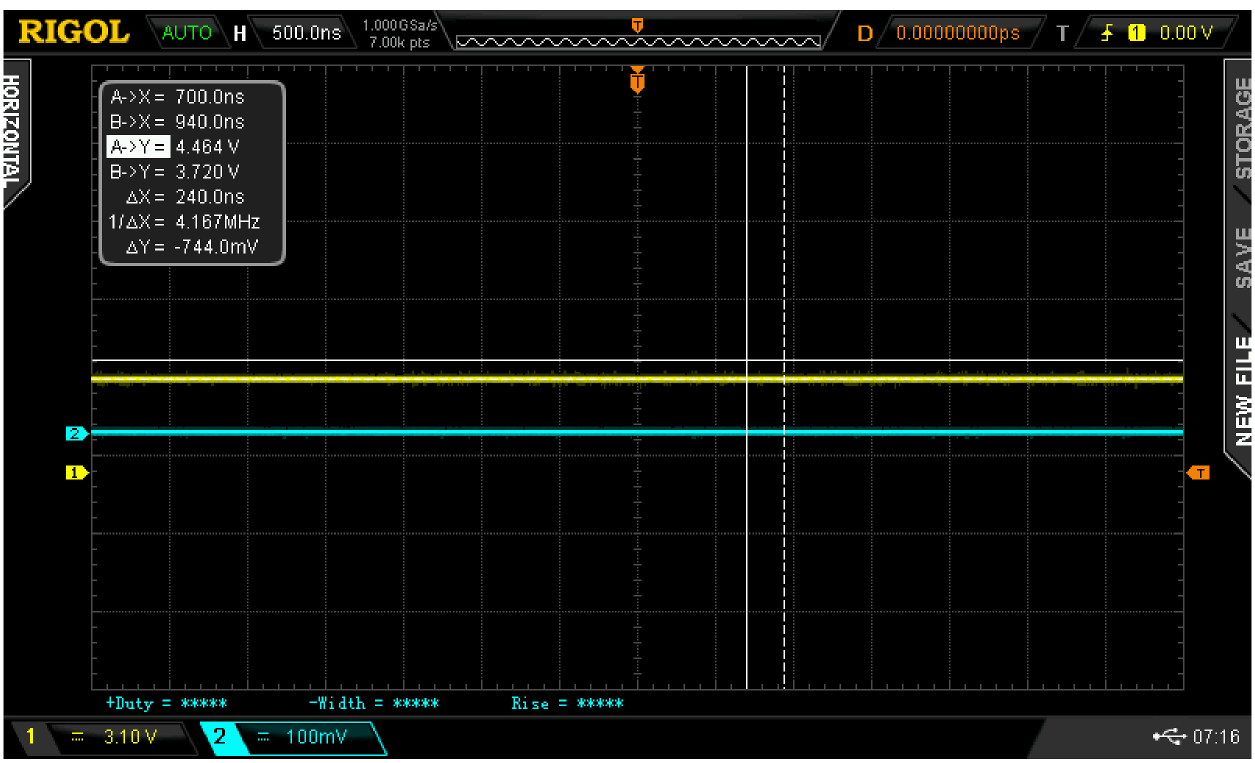


Figure 8: At a maximum resistance of 106.9kΩ, the msp430 analog comparator is below the yellow reference line signal (1.8 Vcc) and LED1 stops blinking.

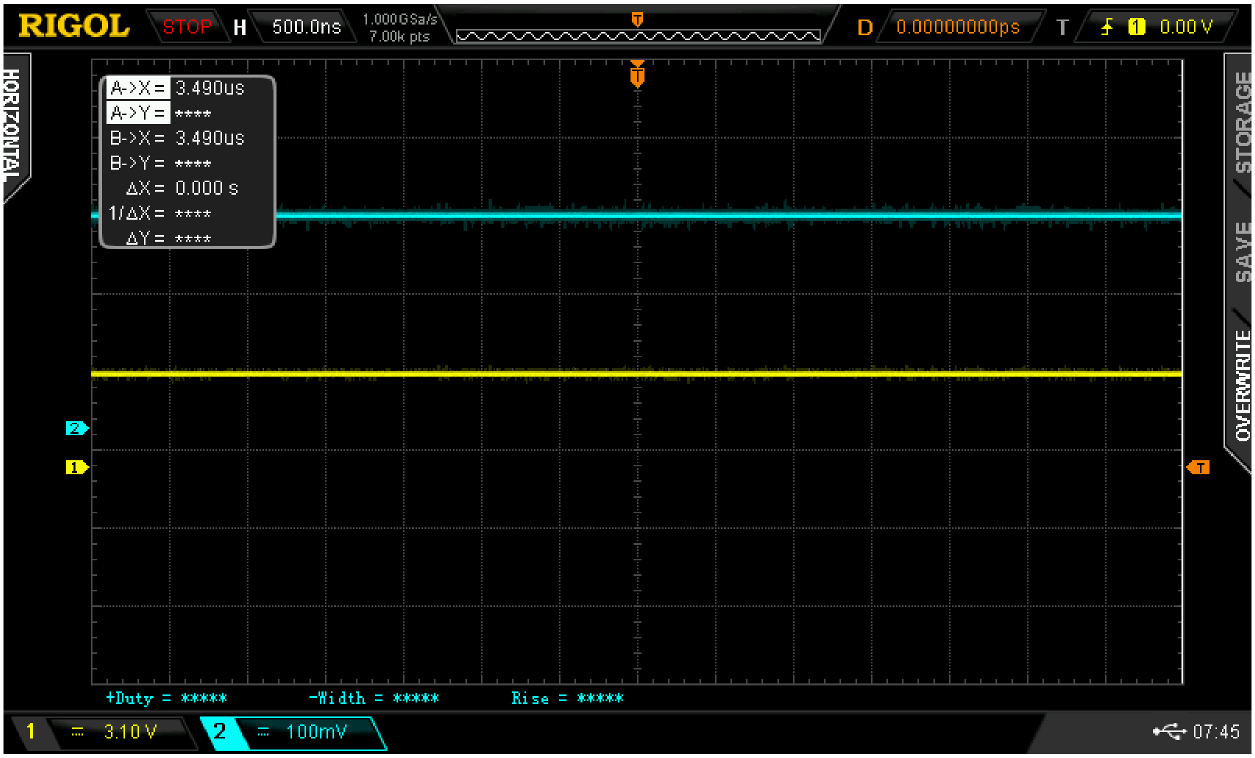
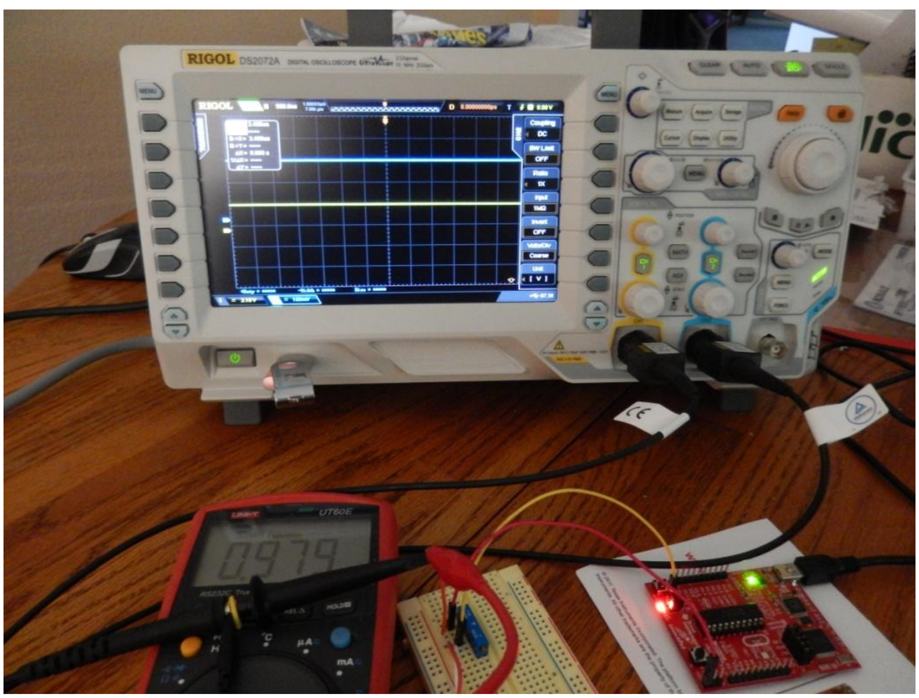


Figure 9: At a minimum resistance of 620 Ω, the msp430 analog comparator shifts the signal above the yellow reference line signal (1.8 Vcc) and LED1 blinks (per second, max brightness).



*** Since the MSP430G2xxx only has one analog comparator (other microcontrollers like the PIC18F4550 have two), the design only saves on chip area if the application such as voltage detection, sensor meters (temperature, pressure, water level…), level shifting requires one comparator. For applications requiring more than one comparator (window comparator), engineers should use a LM393 integrated circuit or similar external TI, LT, AD, Maxim chip to save the hassle of writing embedded code.*

** Threshold Detection (Battery Indicator Emulation) Exercise #2

Rewrite the program to set the LED turn on/off continuously for high/low values of CAOUT. Timer A should be used to periodically adjust the reference voltage between ½ Vcc and ¼ Vcc. When the CompA+ output is compared to ½ Vcc, the red LED located on P1.0 should turn on if the analog signal is found to be higher than the reference. When the CompA+ output is compared to the ¼ Vcc, the green LED on P1.6 should turn on if the analog signal is above that reference. Altogether, the program tests two different comparator references emulating two battery indicator modes (very low battery power remaining (inevitable turn off), low battery power (warning)). To test the program, you can adjust the voltage by turning the knob of the potentiometer. The green LED light turns on when above the ¼ Vcc reference, and the red LED light turns on when above the ½ Vcc.

main.c | main.c | main.c

```
 * Modified Excercise Comparator A+ Program
 * Adapted MSP430G2211 program for MSP4302553 program
 */

#include <msp430g2553.h>

#define LED1    BIT0
#define LED2    BIT6

#define AIN1    BIT1
#define AIN2    BIT3

/* Global Variables */
char flashled = 0;
char flashled2 = 0;

/* Function Definitions */

void main(void) {

    WDTCTL = WDTPW + WDTHOLD;       // Stop watchdog timer
        P1OUT = 0;
        P1DIR |= (LED1 + LED2);             // use output on P1.0 for red LED and P1.6 for green LED

        CACTL1 = CAREF1 + CARSEL + CAIE;            // 0.5 Vcc ref on - pin, enable
                                                    // interrupts on rising edge.
        CCTL0 = CAIE;
        CACTL2 = P2CA4 + CAF;                       // Input CA1 on + pin, filter output.
```

main.c main.c main.c

```
	CACTL2 = P2CA4 + CAF;			// Input CA1 on + pin, filter output.
	CAPD = AIN1;				// disable digital I/O on P1.1 (technically
						// this step is redundant)

	TACCR0 = 60000;				// Timer delay for LED flash, 60000 cycles
	TACCTL0 = CCIE;				// Enable interrupts for CCR0.
	TACTL = TASSEL_2 + ID_3 + MC_1 + TACLR;	// SMCLK, div 8, up mode,
						// clear timer

	CACTL1 |= CAON;				// turn on comparator
	_BIS_SR(LPM0_bits + GIE);		// Enter LPM0 and enable interrupts

} // main


/*  Interrupt Service Routines  */

// http://processors.wiki.ti.com/index.php/MSP430_FAQ#How_to_assign_the_correct_Timer_A_interrupt_vector.3F

#pragma vector = TIMER0_A0_VECTOR
__interrupt void Timer_A(void)
{
	while(1)
	{
		_BIC_SR_IRQ(LPM0_bits);		// Clear LPM0 bits from 0(SR)
		CASTL1 = CAREF0 + CAON;		// 0.25*Vcc, Comparator on
		_BIS_SR(LPM0_bits);		// Enter LPM0
		i = 16384;			// delay
```

```
			while(i>0) {
				i--;
			}
		_BIC_SR_IRQ(LPM0_bits);		// Clear LPM0 bits from 0(SR)
		CACTL1 = CAREF1 + CAON;		// 0.5*Vcc, Comparator on
		_BIS_SR(LPM0_bits);		// Enter LPM0
	}
} // Timer_A

#pragma vector = COMPARATORA_VECTOR
__interrupt void COMPA_ISR(void) {
	if ((CACTL2 & CAOUT)==0x01) {
		CACTL1 |= CAIES;		// value high, so watch for falling edge
		flashled = LED1;			// let red LED flash
	}
	else if((CACTL2 & COUT)==0x40){
		CACTL1 |= CAIES;		// value high, so watch for falling edge
		flashled2 = LED2;			// let green LED flash
	}
	else {
		CACTL1 &= ~CAIES;		// value low, so watch for rising edge
		flashled = 0;				// turn LED off
		flashled2 = 0;
		P1OUT = 0;				// turn ports off.
	}
} // COMPA_ISR
```

Analog-Based Exercises

Make an Op Amp out of a Comparator

For signal conditioning sensor interfaces, the circuit requires some gain to match the sensor's full range to the ADC's full range. Although this task is normally done with an operational amplifier, in some cases, an additional component can exceed the budget for cost-efficient applications. An on-chip comparator can be used as an operational amplifier gain stage for slow sensor signals.

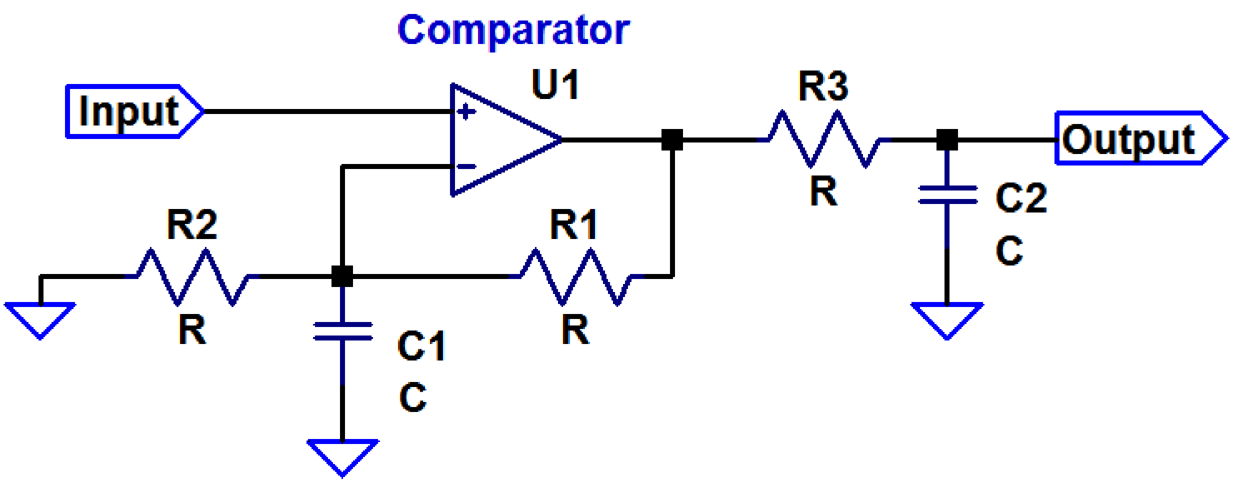


Figure 10: Non-Inverting Amplifier

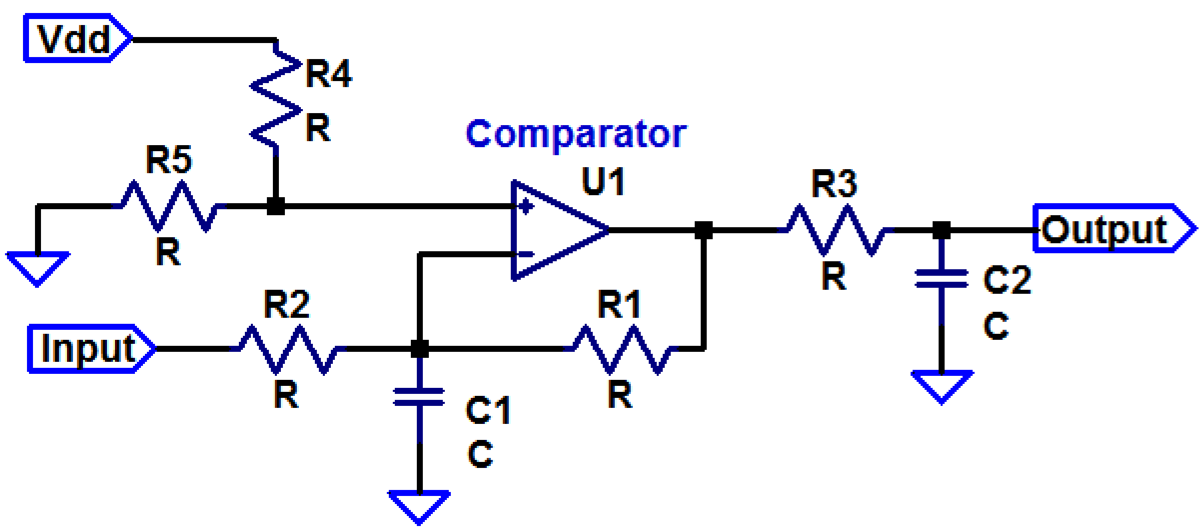


Figure 11: Inverting Amplifier

Hardware/Software Based Delta-Sigma ADC

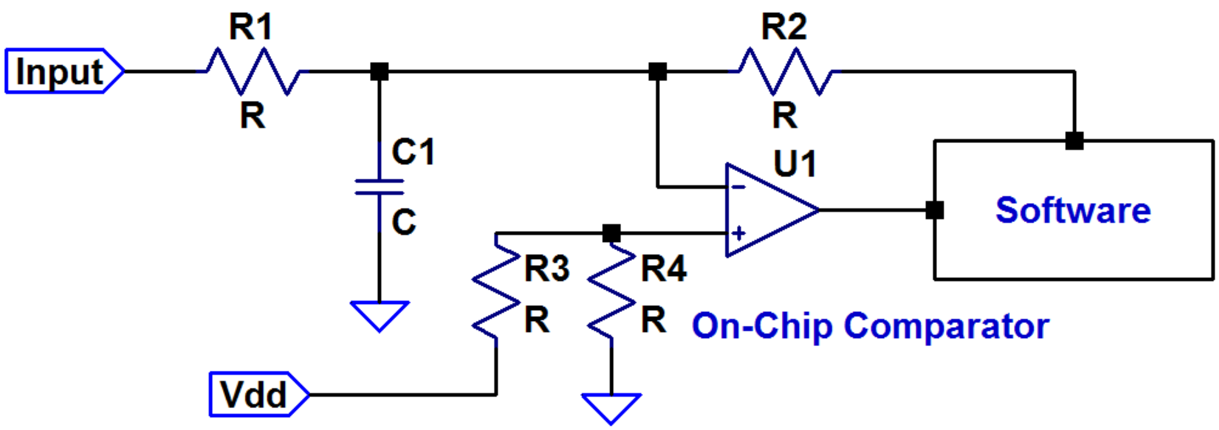


Figure 12: MSP430 Delta-Sigma Modulator (Requires translating PIC assembly code to C)

Works Cited

[1] Zack Albus et al. 2013, Senior Project Guide to Texas Instruments Components (1st) [Online] Available: http://cnx.org/content/m44319/latest/

[2] Anonymous. Revised 2013, MSP430x2xx Family User's Guide [Online] Available: http://www.ti.com/lit/ug/slau144j/slau144j.pdf

[3] John Davies. "Mixed-Signal Systems: Analog Input and Output" in *MSP430 Microcontroller Basics*, 1st ed. Oxford, UK: Newnes, 2008, ch.9, sec.1, pp.371-392

[4] Jerry Luecke. "Signal Conditioning" in *Analog and Digital Circuits for Electronic Control System Applications Using the TI MSP430 Microcontroller*, 1st ed. Oxford, UK: Newnes, 2005, ch.4, pp. 57-.

[5] Anonymous, 2003. "Microchip PICmicro® Comparator Tips 'n Tricks" 1st ed. [Online] Available: http://ww1.microchip.com/downloads/en/devicedoc/41215a.pdf

[6] Beretta, 2010. "Tutorial 12: Making Comparisons: Scientific Instruments using the TI MSP430" [Online] Available: http://mspsci.blogspot.com/2010/12/tutorial-12-making-comparisons.html

[7] Manuel Jiménez et al. "Introduction to Embedded Systems: Using Microcontrollers and the MSP430" [Online] Available: http://www.ernstchan.com/b/src/1384309154506238.pdf

[8] Murugavel Raju, 2001, "Ultrasonic Distance Measurement with the MSP430" [Online] Available: ***http://www.ti.com/lit/an/slaa136a/slaa136a.pdf***