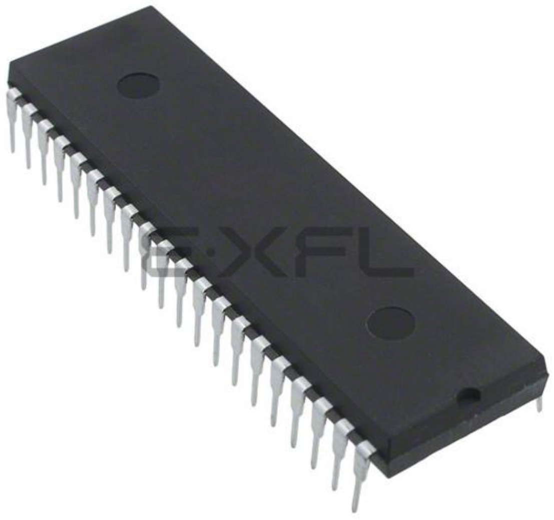
Microchip Technology - PIC16C65B-04/P Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c65b-04-p

TIP #16 Optimizing Destinations

- Destination bit determines W for F for result
- Look at data movement and restructure

Example 16-1

Example: A + B $\rightarrow$ A

```
MOVF        A,W
ADDWF       B,W
MOVWF       A

3 instructions
```

```
MOVF     B,W
ADDWF    A,F

2 instructions
```

Careful use of the destination bits in instructions can save program memory. Here, register A and register B are summed and the result is put into the A register. A destination option is available for logic and arithmetic operations. In the first example, the result of the ADDWF instruction is placed in the working register. A MOVWF instruction is used to move the result from the working register to register A. In the second example, the ADDWF instruction uses the destination bit to place the result into the A register, saving an instruction.

TIP #17 Conditional Bit Set/Clear

- To move single bit of data from REGA to REGB
- Precondition REGB bit
- Test REGA bit and fix REGB if necessary

Example 17-1

```
BTFSS  REGA,2
BCF    REGB,5
BTFSC  REGA,2
BSF    REGB,5

4 instructions
```

```
BCF    REGB,5
BTFSC  REGA,2
BSF    REGB,5

3 instructions
```

One technique for moving one bit from the REGA register to REGB is to perform bit tests. In the first example, the bit in REGA is tested using a BTFSS instruction. If the bit is clear, the BCF instruction is executed and clears the REGB bit, and if the bit is set, the instruction is skipped.The second bit test determines if the bit is set, and if so, will execute the BSF and set the REGB bit, otherwise the instruction is skipped. This sequence requires four instructions.

A more efficient technique is to assume the bit in REGA is clear, and clear the REGB bit, and test if the REGA bit is clear. If so, the assumption was correct and the BSF instruction is skipped, otherwise the REGB bit is set. The sequence in the second example uses three instructions because one bit test was not needed.

One important point is that the second example will create a two-cycle glitch if REGB is a port outputting a high. This is caused by the BCF and BTFSC instructions that will be executed regardless of the bit value in REGA.

CHAPTER 3
PIC® Microcontroller CCP and ECCP Tips ‘n Tricks

Table Of Contents

TIPS ‘N TRICKS INTRODUCTION

Microchip continues to provide innovative products that are smaller, faster, easier-to-use and more reliable. PIC® microcontrollers (MCUs) are used in a wide range of everyday products, from washing machines, garage door openers and television remotes to industrial, automotive and medical products.

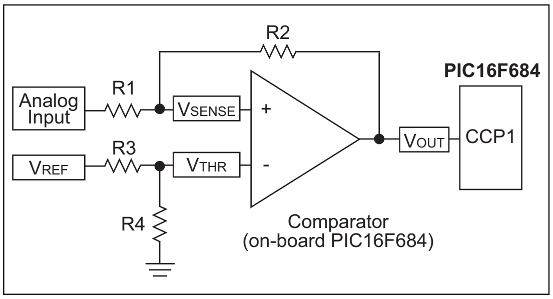
The Capture, Compare and PWM (CCP) modules that are found on many of Microchip’s microcontrollers are used primarily for the measurement and control of time-based pulse signals. The Enhanced CCP (ECCP), available on some of Microchip’s devices, differs from the regular CCP module in that it provides enhanced PWM functionality – namely, full-bridge and half-bridge support, programmable dead-band delay and enhanced PWM auto-shutdown. The ECCP and CCP modules are capable of performing a wide variety of tasks. This document will describe some of the basic guidelines to follow when using these modules in each mode, as well as give suggestions for practical applications.

TIP #6 Measuring the Period of an Analog Signal

Microcontrollers with on-board Analog Comparator module(s), in addition to a CCP (or ECCP) module, can easily be configured to measure the period of an analog signal.

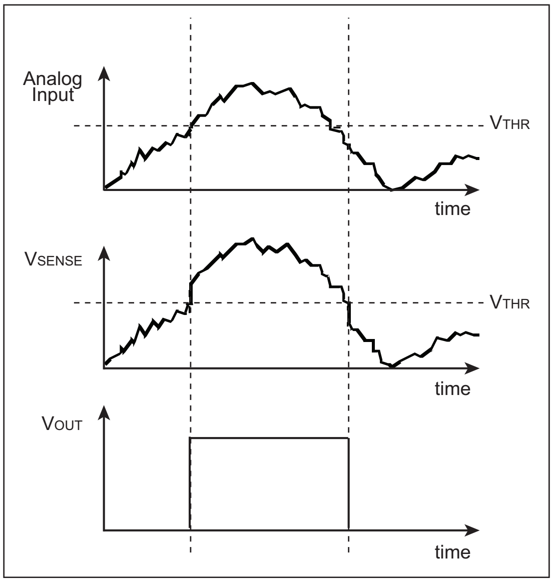
Figure 6-1 shows an example circuit using the peripherals of the PIC16F684.

Figure 6-1: Circuit



R3 and R4 set the threshold voltage for the comparator. When the analog input reaches the threshold voltage, VOUT will toggle from low to high. R1 and R2 provide hysteresis to insure that small changes in the analog input won’t cause jitter in the circuit. Figure 6-2 demonstrates the effect of hysteresis on the input. Look specifically at what VSENSE does when the analog input reaches the threshold voltage.

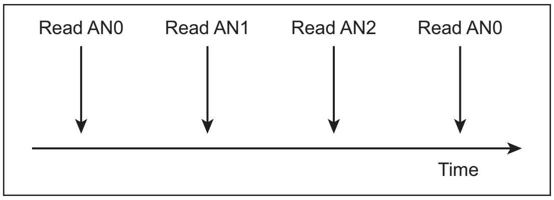
Figure 6-2: Signal Comparison



The CCP module, configured in Capture mode, can time the length between the rising edges of the comparator output (VOUT.) This is the period of the analog input, provided the analog signal reaches VTHR during every period.

TIP #11 Sequential ADC Reader

Figure 11-1: Timeline



Trigger Special Event mode (a sub-mode in Compare mode) generates a periodic interrupt in addition to automatically starting an A/D conversion when Timer1 matches CCPRxL and CCPRxH. The following example problem demonstrates how to sequentially read the A/D channels at a periodic interval.

Example

Given the PIC16F684 running on its 8 MHz internal oscillator, configure the microcontroller to sequentially read analog pins AN0, AN1 and AN2 at 30 ms intervals.

Step #1: Determine Timer1 Prescaler

a) Timer1 overflows at: Tosc*4*65536* prescaler.

b) For a prescaler of 1:1, the Timer1 overflow occurs in 32.8 ms.

c) This is greater than 30 ms, so a prescaler of 1 is adequate.

Step #2: Calculate CCPR1 (CCPR1L and CCPR1H)

a) CCPR1 = Interval Time/(Tosc*4*prescaler) = 0.030/(125 ns*4*1) = 6000 = 0xEA60

b) Therefore, CCPR1L = 0x60, and CCPR1H = 0xEA

Step #3: Configuring CCP1CON

The ECCP module should be configured in Trigger Special Event mode. This mode generates an interrupt when Timer1 equals the value specified in CCPR1. Timer1 is automatically cleared and the GO bit in ADCON0 is automatically set. For this mode, CCP1CON = ‘`b00001011`’.

Step #4: Add Interrupt Service Routine Logic

When the ECCP interrupt is generated, select the next A/D pin for reading by altering the ADCON0 register.

TIP #13 Deciding on PWM Frequency

In general, PWM frequency is application dependent although two general rules-of-thumb hold regarding frequency in all applications. They are:

1. As frequency increases, so does current requirement due to switching losses.
2. Capacitance and inductance of the load tend to limit the frequency response of a circuit.

In low-power applications, it is a good idea to use the minimum frequency possible to accomplish a task in order to limit switching losses. In circuits where capacitance and/or inductance are a factor, the PWM frequency should be chosen based on an analysis of the circuit.

Motor Control

PWM is used extensively in motor control due to the efficiency of switched drive systems as opposed to linear drives. An important consideration when choosing PWM frequency for a motor control application is the responsiveness of the motor to changes in PWM duty cycle. A motor will have a faster response to changes in duty cycle at higher frequencies. Another important consideration is the sound generated by the motor. Brushed DC motors will make an annoying whine when driven at frequencies within the audible frequency range (20 Hz-4 kHz.) In order to eliminate this whine, drive brushed DC motors at frequencies greater than 4 kHz. (Humans can hear frequencies at upwards of 20 kHz, however, the mechanics of the motor winding will typically attenuate motor whine above 4 kHz).

LED and Light Bulbs

PWM is also used in LED and light dimmer applications. Flicker may be noticeable with rates below 50 Hz. Therefore, it is generally a good rule to pulse-width modulate LEDs and light bulbs at 100 Hz or higher.

TIP #14 Unidirectional Brushed DC Motor Control Using CCP

Figure 14-1: Brushed DC (BDC) Motor Control Circuit

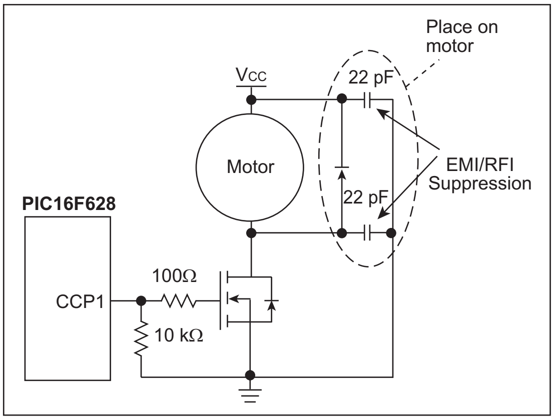


Figure 14-1 shows a unidirectional speed controller circuit for a brushed DC motor. Motor speed is proportional to the duty cycle of the PWM output on the CCP1 pin. The following steps show how to configure the PIC16F628 to generate a 20 kHz PWM with 50% duty cycle. The microcontroller is running on a 20 MHz crystal.

Step #1: Choose Timer2 Prescaler

a) F_{PWM} = Fosc/((PR2+1)*4*prescaler) = 19531 Hz for PR2 = 255 and prescaler of 1

b) This frequency is lower than 20 kHz, therefore a prescaler of 1 is adequate.

Step #2: Calculate PR2

PR2 = Fosc/(F_{PWM}*4*prescaler) – 1 = 249

Step #3: Determine CCPR1L and CCP1CON<5:4>

a) CCPR1L:CCP1CON<5:4> = DutyCycle*0x3FF = 0x1FF

b) CCPR1L = 0x1FF >> 2 = 0x7F, CCP1CON<5:4> = 3

Step #4: Configure CCP1CON

The CCP module is configured in PWM mode with the Least Significant bits of the duty cycle set, therefore, CCP1CON = ‘`b001111000`’.

COMBINATION CAPTURE AND COMPARE TIPS

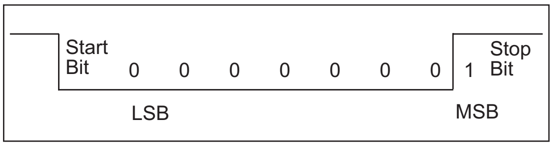
The CCP and ECCP modules can be configured on the fly. Therefore, these modules can perform different functions in the same application provided these functions operate exclusively (not at the same time). This section will provide examples of using a CCP module in different modes in the same application.

TIP #20 RS-232 Auto-baud

RS-232 serial communication has a variety of baud rates to choose from. Multiple transmission rates require software which detects the transmission rate and adjusts the receive and transmit routines accordingly. Auto-baud is used in applications where multiple transmission rates can occur. The CCP module can be configured in Capture mode to detect the baud rate and then be configured in Compare mode to generate or receive RS-232 transmissions.

In order for auto-baud to work, a known calibration character must be transmitted initially from one device to another. One possible calibration character is show in Figure 20-1. Timing this known character provides the device with the baud rate for all subsequent communications.

Figure 20-1: RS-232 Calibration Character



Auto-baud Routine Implementation:

1. Configure CCP module to capture the falling edge (beginning of Start bit).
2. When the falling edge is detected, store the CCPR1 value.
3. Configure the CCP module to capture the rising edge.
4. Once the rising edge is detected, store the CCPR1 value.
5. Subtract the value stored in step 2 from the value in step 4. This is the time for 8 bits.
6. Shift the value calculated in step 5 right 3 times to divide by 8. This result is the period of a bit (T_B).
7. Shift value calculated in step 6 right by 1. This result is half the period of a bit.

The following code segments show the process for transmitting and receiving data in the normal program flow. This same functionality can be accomplished using the CCP module by configuring the module in Compare mode and generating a CCP interrupt every bit period. When this method is used, one bit is either sent or received when the CCP interrupt occurs.

Note: Refer to Application Note AN712 “*RS-232 Auto-baud for the PIC16C5X Devices*” for more details on auto-baud.

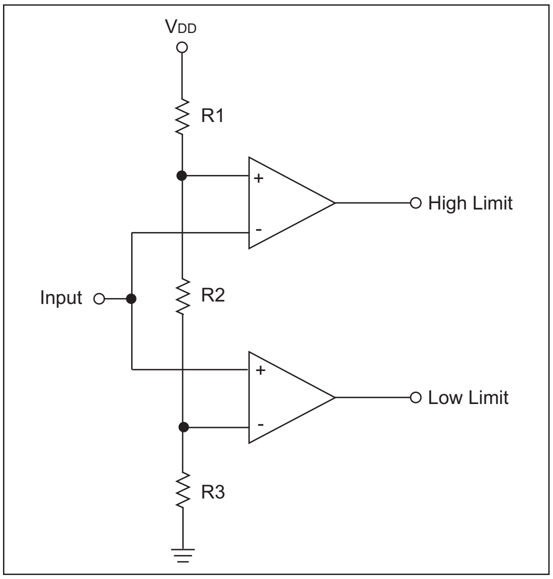
NOTES:

TIP #5 Window Comparison

When monitoring an external sensor, it is often convenient to be able to determine when the signal has moved outside a pre-established safe operating range of values or window of operation. This windowing provides the circuit with an alarm when the signal moves above or below safety limits, ignoring minor fluctuations inside the safe operating range.

To implement a window comparator, two voltage comparators and 3 resistors are required (see Figure 5-1).

Figure 5-1: Window Comparator



Resistors R1, R2 and R3 form a voltage divider which generates the high and low threshold voltages. The outputs HIGH LIMIT and LOW LIMIT are both active high, generating a logic one on the HIGH LIMIT output when the input voltage rises above the high threshold, and a logic one on the LOW LIMIT output when the input voltage falls below the low threshold.

To calculate values for R1, R2 and R3, find values that satisfy Equation 5-1 and Equation 5-2.

Note: A continuous current will flow through R1, R2 and R3. To limit the power dissipation in the resistors, the total resistance of R1, R2 and R3 should be at least 1k. The total resistance of R1, R2 and R3 should also be kept less than 1 M to prevent offset voltages due to the input bias currents of the comparator.

Equation 5-1

$$V_{TH-HI} = \frac{V_{DD} * (R_3 + R_2)}{R1 + R2 + R3}$$

Equation 5-2

$$V_{TH-LO} = \frac{V_{DD} * R_3}{R1 + R2 + R3}$$

Example:

- V_{DD} = 5.0V, V_{TH} = 2.5V, V_{TL} = 2.0V
- R1 = 12k, R2 = 2.7k, R3 = 10k
- V_{TH} (actual) = 2.57V, V_{TL} (actual) = 2.02V

Adding Hysteresis:

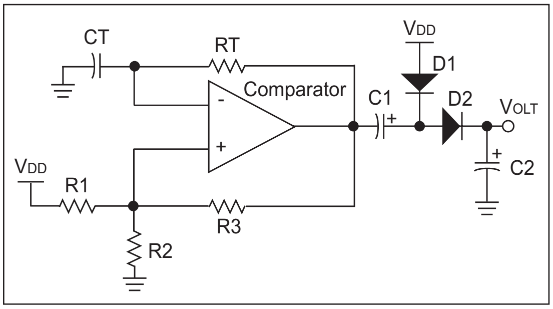
To add hysteresis to the HIGH LIMIT comparator, follow the procedure outlined in Tip #3. Use the series combination of R2 and R3 as the resistor R2 in Tip #3.

To add hysteresis to the LOW LIMIT comparator, choose a suitable value for Req, 1k to 10 k , and place it between the circuit input and the non-inverting input of the LOW LIMIT comparator. Then calculate the needed feedback resistor using Equation 3-4 and Equation 3-5.

TIP #10 Capacitive Voltage Doubler

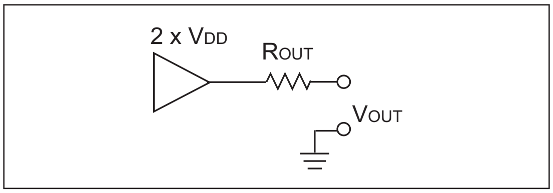
This tip takes the multi-vibrator described in Tip #8 and builds a capacitive voltage doubler around it (see Figure 10-1). The circuit works by alternately charging capacitor C1 through diode D1, and then charge balancing the energy in C1 with C2 through diode D2. At the start of the cycle, the output of the multi-vibrator is low and charge current flows from V_{DD} through D1 and into C1. When the output of the multi-vibrator goes high, D1 is reverse biased and the charge current stops. The voltage across C1 is added to the output voltage of the multi-vibrator, creating a voltage at the positive terminal of C1 which is 2 x V_{DD}. This voltage forward biases D2 and the charge in C1 is shared with C2. When the output of the multi-vibrator goes low again, the cycle starts over.

Figure 10-1: Capacitive Voltage Doubler



Note: The output voltage of a capacitive double is unregulated and will sag with increasing load current. Typically, the output is modeled as a voltage source with a series resistance (see Figure 10-2).

Figure 10-2: Equivalent Output Model



To design a voltage doubler, first determine the maximum tolerable output resistance based on the required output current and the minimum tolerable output voltage. Remember that the output current will be limited to one half of the output capability of the comparator. Then choose a transfer capacitance and switching frequency using Equation 10-1.

Equation 10-1

$$R_{OUT} = \frac{1}{F_{SWITCH} * C1}$$

Note: R_{OUT} will be slightly higher due to the dynamic resistance of the diodes. The equivalent series resistance or ESR, of the capacitors and the output resistance of the comparator. See the TC7660 data sheet for a more complete description.

Once the switching frequency is determined, design a square-wave multi-vibrator as described in Tip #8.

Finally, select diodes D1 and D2 for their current rating and set C2 equal to C1.

Example:

From Tip #8, the values are modified for a F_{OSC} of 4.8 kHz.

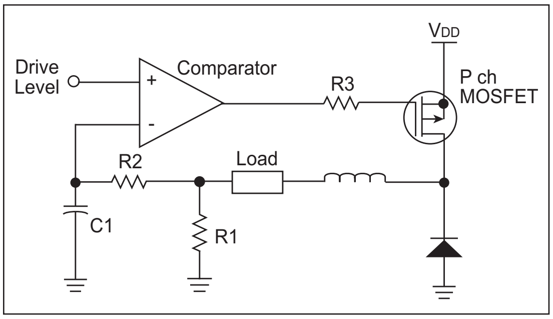
- C1 and C2 = 10 μF
- R_{OUT} = 21

TIP #13 PWM High-Current Driver

This tip combines a comparator with a MOSFET transistor and an inductor to create a switch mode high-current driver circuit. (See Figure 13-1).

The operation of the circuit begins with the MOSFET off and no current flowing in the inductor and load. With the sense voltage across R1 equal to zero and a DC voltage present at the drive level input, the output of the comparator goes low. The low output turns on the MOSFET and a ramping current builds through the MOSFET, inductor, load and R1.

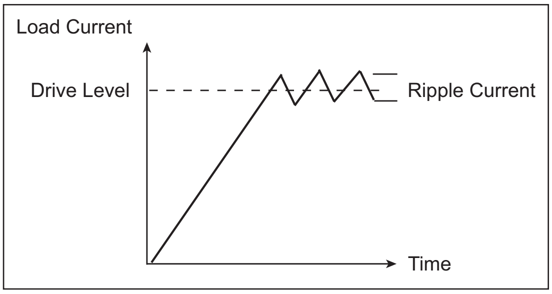
Figure 13-1: High Current Driver



When the current ramps high enough to generate a voltage across R1 equal to the drive level, the comparator output goes high turning off the MOSFET. The voltage at the junction of the MOSFET and the inductor then drops until D1 forward biases. The current continues ramping down from its peak level toward zero. When the voltage across the sense resistor R1 drops below the drive level, the comparator output goes low, the MOSFET turns on, and the cycle starts over.

R2 and C1 form a time delay network that limits the switching speed of the driver and causes it to slightly overshoot and undershoot the drive level when operating. The limit is necessary to keep the switching speed low, so the MOSFET switches efficiently. If R2 and C1 were not present, the system would run at a speed set by the comparator propagation delay and the switching speed of the MOSFET. At that speed, the switching time of the MOSFET would be a significant portion of the switching time and the switching efficiency of the MOSFET would be too low.

Figure 13-1: Current Through the Load



To design a PWM high current driver, first determine a switching speed (F_{SWX}) that is appropriate for the system. Next, choose a MOSFET and D1 capable of handling the load current requirements. Then choose values for R2 and C1 using Equation 13-1.

Equation 13-1

$$F_{SWX} = \frac{2}{R2 * C1}$$

Next determine the maximum ripple current that the load will tolerate, and calculate the required inductance value for L1 using Equation 13-2.

Equation 13-2

$$L = \frac{V_{DD} - V_{LOAD}}{I_{RIPPLE} * F_{SWX} * 2}$$

Finally, choose a value for R1 that will produce a feedback ripple voltage of 100 mV for the maximum ripple current I_{RIPPLE}.

Example:

- F_{SWX} = 10 kHz, R2 = 22k, C1 = .01 µF
- I_{RIPPLE} = 100 mA, V_{DD} = 12V, V_L = 3.5V
- L = 4.25 mH

TIP #4 Drive Software

Pulse-Width Modulation (PWM) Algorithms

Pulse-Width Modulation is critical to modern digital motor controls. By adjusting the pulse width, the speed of a motor can be efficiently controlled without larger linear power stages. Some PIC devices and all dsPIC DSCs have hardware PWM modules on them. These modules are built into the Capture/Compare/PWM (CCP) peripheral. CCP peripherals are intended for a single PWM output, while the Enhanced CCP (ECCP) is designed to produce the complete H-Bridge output for bidirectional Brushed DC motor control. If cost is a critical design point, a PIC device with a CCP module may not be available, so software generated PWM is a good alternative.

The following algorithms are designed to efficiently produce an 8-bit PWM output on the Mid-Range family of PIC microcontrollers. These algorithms are implemented as macros. If you want these macros to be a subroutine in your program, simply remove the macro statements and replace them with a label and a return statement.

Example 4-1: 1 Output 8-Bit PWM

```
pwm_counter equ xxx ;variable
pwm         equ xxx ;variable

set_pwm macro A     ;sets the pwm
                    ;setpoint to the
                    ;value A
 MOVLW A
 MOVWF pwm
 endm

update_PWM macro    ;performs one update
                    ;of the PWM signal
                    ;place the PWM output
                    ;pin at bit 0 or 7 of
                    ;the port
 MOVF pwm_counter,w
 SUBWF pwm, w       ;if the output
                    ;is on bit 0
 RLF        PORTC,f ;replace PORTC with
                    ;the correct port if
                    ;the output is on bit
                    ;7 of the port
                    ;replace the rlf with
                    ;rrf incf
                    ;pwm_counter,f
```

Example 4-2: 8 Output 8-Bit PWM

```
pwm_counter equ xxx     ;variable
pwm0        equ xxx     ;
pwm1        equ pwm0+1
pwm2        equ pwm1+1
pwm3        equ pwm2+1
pwm4        equ pwm3+1
pwm5        equ pwm4+1
pwm6        equ pwm5+1
pwm7        equ pwm6+1
output      equ pwm7+1
set_pwm macro A,b       ;sets pwm b with
                        ;the value A
 MOVLW pwm0
 ADDLW b
 MOVWF fsr
 MOVLW a
 MOVWF indf
 endm

update_PWM macro        ;peforms one
                        ;update of all 8
                        ;PWM signals
                        ;all PWM signals
                        ;must be on the
                        ;same port
 MOVF       pwm_counter,w
 SUBWF      pwm0,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm1,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm2,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm3,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm4,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm5,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm6,w
 RLF        output,f
 MOVF       pwm_counter,w
 SUBWF      pwm7,w
 RLF        output,w
 MOVWF      PORTC
 INCF       pwm_counter,f
endm
```

TIP #6 Current Sensing

The torque of an electric motor can be monitored and controlled by keeping track of the current flowing through the motor. Torque is directly proportional to the current. Current can be sensed by measuring the voltage drop through a known value resistor or by measuring the magnetic field strength of a known value inductor. Current is generally sensed at one of two places, the supply side of the drive circuit (high side current sense) or the sink side of the drive circuit (low side current sense). Low side sensing is much simpler but the motor will no longer be grounded, causing a safety issue in some applications. High side current sensing generally requires a differential amplifier with a common mode voltage range within the voltage of the supply.

Figure 6-1: Resistive High Side Current Sensing

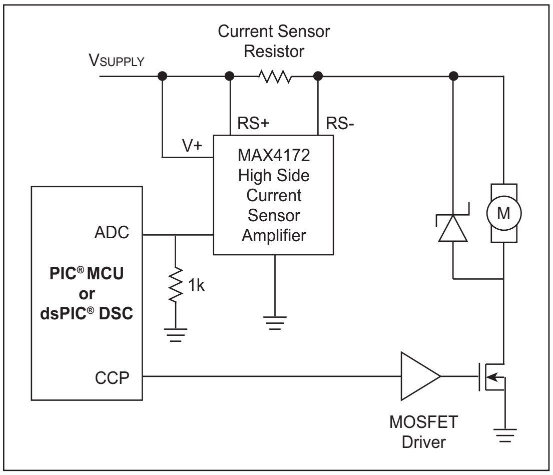
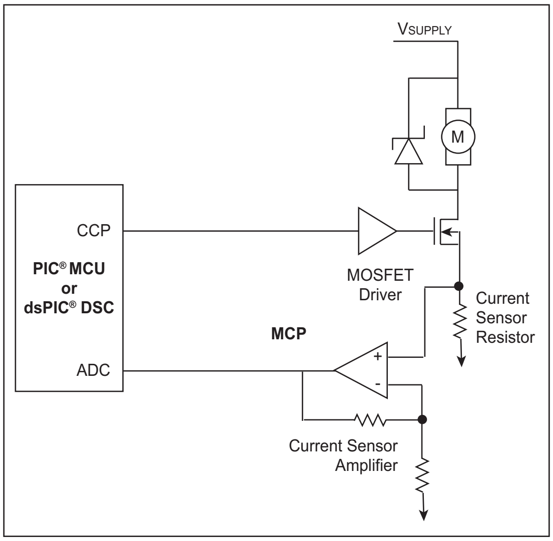
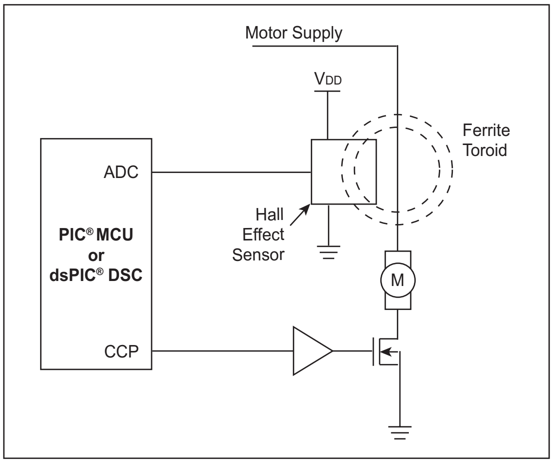


Figure 6-2: Resistive Low Side Current Sensing



Current measurement can also be accomplished using a Hall effect sensor to measure the magnetic field surrounding a current carrying wire. Naturally, this Hall effect sensor can be located on the high side or the low side of the load. The actual location of the sensor does not matter because the sensor does not rely upon the voltage on the wire. This is a non-intrusive method that can be used to measure motor current.

Figure 6-3: Magnetic Current Sensing



TIP #1 Typical Ordering Considerations and Procedures for Custom Liquid Displays

1. Consider what useful information needs to be displayed on the custom LCD and the combination of alphanumeric and custom icons that will be necessary.
2. Understand the environment in which the LCD will be required to operate. Operating voltage and temperature can heavily influence the contrast of the LCD and potentially limit the type of LCD that can be used.
3. Determine the number of segments necessary to achieve the desired display on the LCD and reference the PIC Microcontroller LCD matrix for the appropriate LCD PIC microcontroller.
4. Create a sketch/mechanical print and written description of the custom LCD and understand the pinout of the LCD. (Pinout definition is best left to the glass manufacturer due to the constraints of routing the common and segment electrodes in two dimensions.)
5. Send the proposed LCD sketch and description for a written quotation to at least 3 vendors to determine pricing, scheduling and quality concerns.
   a) Take into account total NRE cost, price per unit, as well as any setup fees.
   b) Allow a minimum of two weeks for formal mechanical drawings and pin assignments and revised counter drawings.
6. Request a minimal initial prototype LCD build to ensure proper LCD development and ensure proper functionality within the target application.
   a) Allow typically 4-6 weeks for initial LCD prototype delivery upon final approval of mechanical drawings and pin assignments.
7. Upon receipt of prototype LCD, confirm functionality before giving final approval and beginning production of LCD.

Note: Be sure to maintain good records by keeping copies of all materials transferred between both parties, such as initial sketches, drawings, pinouts, etc.

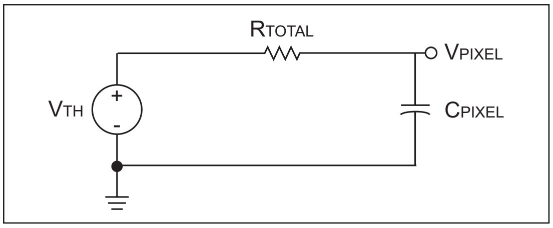
TIP #2 LCD PIC® MCU Segment/ Pixel Table

Table 2-1: Segment Matrix Table

Multiplex Commons	Maximum Number of Segments/Pixels					Bias
	PIC16F913/ 916	PIC16F914/ 917	PIC16F946	PIC18F6X90 (PIC18F6XJ90)	PIC18F8X90 (PIC18F8XJ90)	
Static (COM0)	15	24	42	32/ (33)	48	Static
1/2 (COM1: COM0)	30	48	84	64/ (66)	96	1/2 or 1/3
1/3 (COM2: COM0)	45	72	126	96/ (99)	144	1/2 or 1/3
1/4 (COM3: COM0)	60	96	168	128/ (132)	192	1/3

This Segment Matrix table shows that Microchip’s 80-pin LCD devices can drive up to 4 commons and 48 segments (192 pixels), 64-pin devices can drive up to 33 segments (132 pixels), 40/44 pin devices can drive up to 24 segments (96 pixels) and 28-pin devices can drive 15 segments (60 segments).

Figure 3-4: Voltage Change Across Pixel



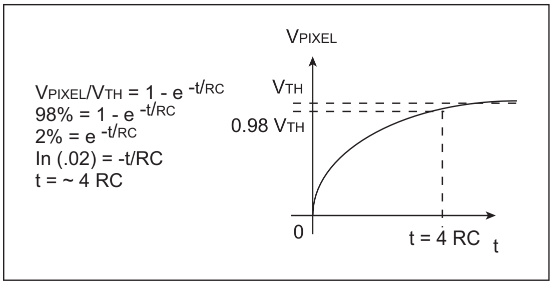
The step response of the voltage across a pixel is subject to the following equation:

Equation 3-1

$$V_{PIXEL} = V_{TH} (1 - e^{-t/RC})$$

By manipulating the equation, we can see that it will take a time equal to 4 time constants for the pixel voltage to reach 98% of the bias voltage.

Figure 3-5: Step Response Diagram



Now we need to estimate the capacitance. Capacitance is proportional to the area of a pixel. We can measure the area of a pixel and estimate the capacitance as shown. Obviously, a bigger display, such as a digital wall clock, will have bigger pixels and higher capacitance.

Equation 3-2

$$C_{PIXEL} = 1500 \text{ pF/cm}^2$$
$$AREA_{PIXEL} = 1 \text{ mm} * 3 \text{ mm} = .03 \text{ cm}^2$$
$$C_{PIXEL} = 45 \text{ pF}$$

We want the time constant to be much smaller than the period of the LCD waveform, so that rounding of the LCD waveform will be minimized. If we want the RC to be equal to 100 µS, then the total resistance can be calculated as shown:

Equation 3-3

$$R_{TOTAL} = 100\ \mu\text{S}/45 \text{ pF} = 2.22 \text{ m}\Omega$$
$$R_{TH} = 2.2\text{M} - 5.1\text{K} = 2.2\text{M}$$

The resistance of the switching circuits within the LCD module is very small compared to this resistance, so the Thevenin resistance of the resistor ladder at $V_{LCD}2$ and $V_{LCD}1$ can be treated the same as R_{TOTAL}. We can then calculate the value for R that will give us the correct Thevenin resistance.

Equation 3-4

$$R = 3\ R_{TH}/2 = 3.3\text{M}$$

Now we can calculate the current through the resistor ladder if we used 3.3 mΩ resistors.

Equation 3-5

$$R_{LADDER} = 9.9\text{M},$$
$$I_{LADDER} = 5\text{V}/9.9\text{M} = 0.5\ \mu\text{A}$$

Use this process to estimate maximum resistor sizes for your resistor ladder and you will drastically reduce power consumption for your LCD application. Don't forget to observe the display over the operating conditions of your product (such as temperature, voltage and even, humidity) to ensure that contrast and display quality is good.

Application Note References

- AN220, “*Watt-Hour Meter Using PIC16C923 and CS5460*” (DS00220)
- AN582, “*Low-Power Real-Time Clock*” (DS00582)
- AN587, “*Interfacing PIC® MCUs to an LCD Module*” (DS00587)
- AN649, “*Yet Another Clock Featuring the PIC16C924*” (DS00649)
- AN658, “*LCD Fundamentals Using PIC16C92X Microcontrollers*” (DS00658)
- TB084, “*Contrast Control Circuits for the PIC16F91X*” (DS91084)

Application notes can be found on the Microchip web site at www.microchip.com.

TIP #4 Powering 3.3V Systems From 5V Using Switching Regulators

A buck switching regulator, shown in Figure 4-1, is an inductor-based converter used to step-down an input voltage source to a lower magnitude output voltage. The regulation of the output is achieved by controlling the ON time of MOSFET Q1. Since the MOSFET is either in a lower or high resistive state (ON or OFF, respectively), a high source voltage can be converted to a lower output voltage very efficiently.

The relationship between the input and output voltage can be established by balancing the volt-time of the inductor during both states of Q1.

Equation 4-1

$$(V_S - V_O) * t_{on} = V_o * (T - t_{on})$$
$$\text{Where: } T \equiv t_{on}/\text{Duty\_Cycle}$$

It therefore follows that for MOSFET Q1:

Equation 4-2

$$\text{Duty\_Cycle}_{Q1} = V_O/V_S$$

When choosing an inductor value, a good starting point is to select a value to produce a maximum peak-to-peak ripple current in the inductor equal to ten percent of the maximum load current.

Equation 4-3

$$V = L * (di/dt)$$
$$L = (V_S - V_O) * (t_{on}/I_o * 0.10)$$

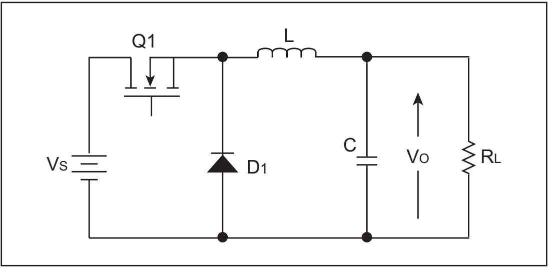
When choosing an output capacitor value, a good starting point is to set the LC filter characteristic impedance equal to the load resistance. This produces an acceptable voltage overshoot when operating at full load and having the load abruptly removed.

Equation 4-4

$$Z_o \equiv \sqrt{L/C}$$
$$C = L/R^2 = (I_O^2 * L)/V_O^2$$

When choosing a diode for D1, choose a device with a sufficient current rating to handle the inductor current during the discharge part of the pulse cycle (I_L).

Figure 4-1: Buck Regulator



Digital Interfacing

When interfacing two devices that operate at different voltages, it is imperative to know the output and input thresholds of both devices. Once these values are known, a technique can be selected for interfacing the devices based on the other requirements of your application. Table 4-1 contains the output and input thresholds that will be used throughout this document. When designing an interface, make sure to reference your manufacturers data sheet for the actual threshold levels.

Table 4-1: Input/Output Thresholds

	V_{OH} min	V_{OL} max	V_{IH} min	V_{IL} max
5V TTL	2.4V	0.5V	2.0V	0.8V
3.3V LVTTL	2.4V	0.4V	2.0V	0.8V
5V CMOS	4.7V (Vcc-0.3V)	0.5V	3.5V (0.7xVcc)	1.5V (0.3xVcc)
3.3V LVCMOS	3.0V (Vcc-0.3V)	0.5V	2.3V (0.7xVcc)	1.0V (0.3xVcc)

TIP #5 3.3V → 5V Direct Connect

The simplest and most desired way to connect a 3.3V output to a 5V input is by a direct connection. This can be done only if the following 2 requirements are met:

- The V_{OH} of the 3.3V output is greater than the V_{IH} of the 5V input
- The V_{OL} of the 3.3V output is less than the V_{IL} of the 5V input

An example of when this technique can be used is interfacing a 3.3V LVCMOS output to a 5V TTL input. From the values given in Table 4-1, it can clearly be seen that both of these requirements are met.

3.3V LVCMOS V_{OH} of 3.0 volts is greater than 5V TTL V_{IH} of 2.0 volts, and

3.3V LVCMOS V_{OL} of 0.5 volts is less than 5V TTL V_{IL} of 0.8 volts.

When both of these requirements are not met, some additional circuitry will be needed to interface the two parts. See Tips 6, 7, 8 and 13 for possible solutions.

TIP #6 3.3V → 5V Using a MOSFET Translator

In order to drive any 5V input that has a higher V_{IH} than the V_{OH} of a 3.3V CMOS part, some additional circuitry is needed. A low-cost two component solution is shown in Figure 6-1.

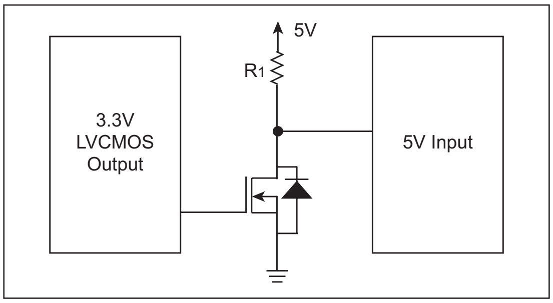
When selecting the value for R1, there are two parameters that need to be considered; the switching speed of the input and the current consumption through R1. When switching the input from a ‘0’ to a ‘1’, you will have to account for the time the input takes to rise because of the RC time constant formed by R1, and the input capacitance of the 5V input plus any stray capacitance on the board. The speed at which you can switch the input is given by the following equation:

Equation 6-1

$$T_{SW} = 3 \times R_1 \times (C_{IN} + C_S)$$

Since the input and stray capacitance of the board are fixed, the only way to speed up the switching of the input is to lower the resistance of R1. The trade-off of lowering the resistance of R1 to get faster switching times is the increase in current draw when the 5V input remains low. The switching to a ‘0’ will typically be much faster than switching to a ‘1’ because the ON resistance of the N-channel MOSFET will be much smaller than R1. Also, when selecting the N-channel FET, select a FET that has a lower V_{GS} threshold voltage than the V_{OH} of 3.3V output.

Figure 6-1: MOSFET Translator



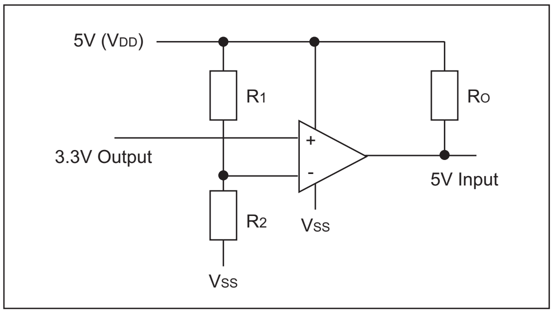
TIP #8 3.3V → 5V Using a Voltage Comparator

The basic operation of the comparator is as follows:

- When the voltage at the inverting (-) input is greater than that at the non-inverting (+) input, the output of the comparator swings to V_{SS}.
- When the voltage at the non-inverting (+) input is greater than that at the non-inverting (-) input, the output of the comparator is in a high state.

To preserve the polarity of the 3.3V output, the 3.3V output must be connected to the non-inverting input of the comparator. The inverting input of the comparator is connected to a reference voltage determined by R1 and R2, as shown in Figure 8-1.

Figure 8-1: Comparator Translator



Calculating R1 and R2

The ratio of R1 and R2 depends on the logic levels of the input signal. The inverting input should be set to a voltage halfway between V_{OL} and V_{OH} for the 3.3V output. For an LVCMOS output, this voltage is:

Equation 8-1:

$$1.75V = \frac{(3.0V + .5V)}{2}$$

Given that R1 and R2 are related by the logic levels:

Equation 8-2:

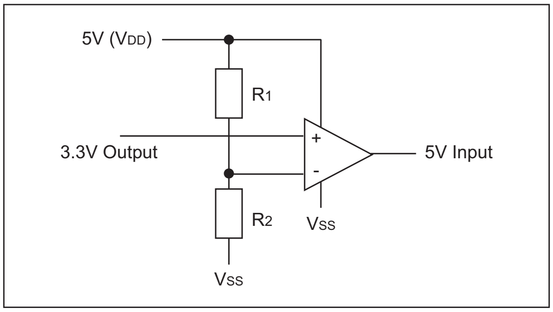
$$R_1 = R_2 \left(\frac{5V}{1.75V} - 1 \right)$$

assuming a value of 1K for R2, R1 is 1.8K.

An op amp wired up as a comparator can be used to convert a 3.3V input signal to a 5V output signal. This is done using the property of the comparator that forces the output to swing high (V_{DD}) or low (V_{SS}), depending on the magnitude of difference in voltage between its 'inverting' input and 'non-inverting' input.

Note: For the op amp to work properly when powered by 5V, the output must be capable of rail-to-rail drive.

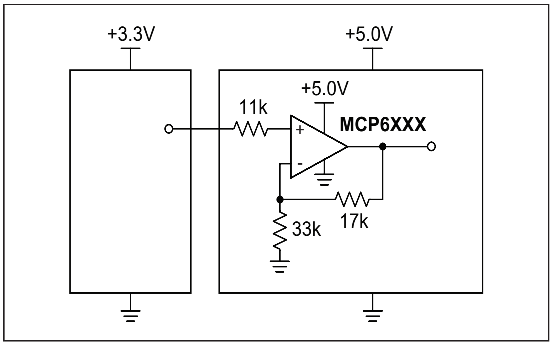
Figure 8-2: Op Amp as a Comparator



TIP #14 3.3V → 5V Analog Gain Block

To scale analog voltage up when going from 3.3V supply to 5V supply. The 33 kΩ and 17 kΩ set the op amp gain so that the full scale range is used in both sides. The 11 kΩ resistor limits current back to the 3.3V circuitry.

Figure 14-1: Analog Gain Block

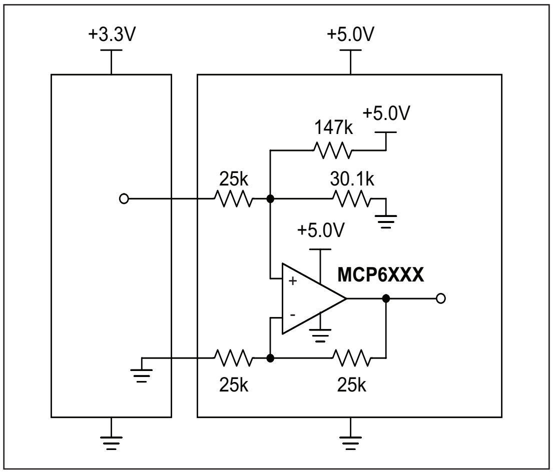


TIP #15 3.3V → 5V Analog Offset Block

Offsetting an analog voltage for translation between 3.3V and 5V.

Shift an analog voltage from 3.3V supply to 5V supply. The 147 kΩ and 30.1 kΩ resistors on the top right and the +5V supply voltage are equivalent to a 0.85V voltage source in series with a 25 kΩ resistor. This equivalent 25 kΩ resistance, the three 25 kΩ resistors, and the op amp form a difference amplifier with a gain of 1 V/V. The 0.85V equivalent voltage source shifts any signal seen at the input up by the same amount; signals centered at 3.3V/2 = 1.65V will also be centered at 5.0V/2 = 2.50V. The top left resistor limits current from the 5V circuitry.

Figure 15-1: Analog Offset Block



TIP #16 5V → 3.3V Active Analog Attenuator

Reducing a signal's amplitude from a 5V to 3.3V system using an op amp.

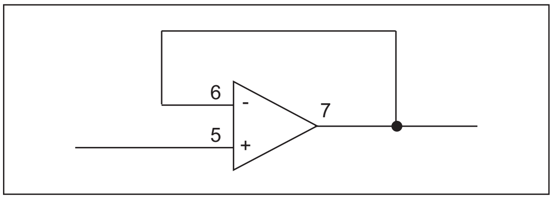
The simplest method of converting a 5V analog signal to a 3.3V analog signal is to use a resistor divider with a ratio R1:R2 of 1.7:3.3. However, there are a few problems with this.

1. The attenuator may be feeding a capacitive load, creating an unintentional low pass filter.
2. The attenuator circuit may need to drive a low-impedance load from a high-impedance source.

Under either of these conditions, an op amp becomes necessary to buffer the signals.

The op amp circuit necessary is a unity gain follower (see Figure 16-1).

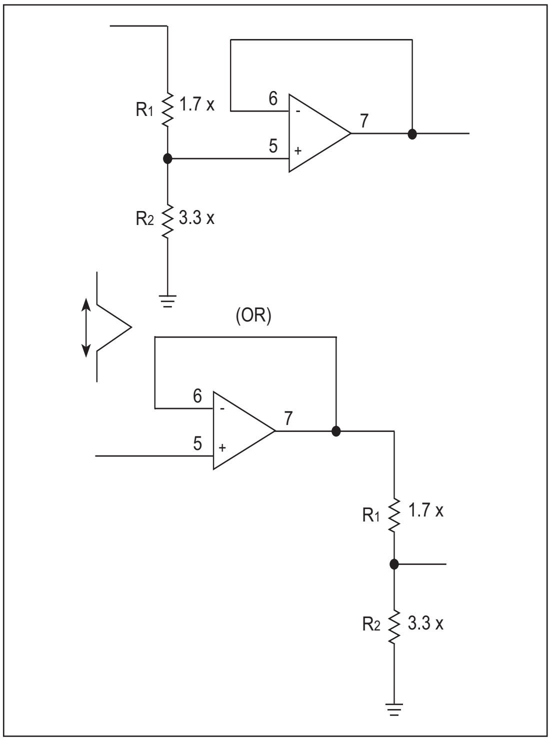
Figure 16-1: Unity Gain



This circuit will output the same voltage that is applied to the input.

To convert the 5V signal down to a 3V signal, we simply add the resistor attenuator.

Figure 16-2: Op Amp Attenuators



If the resistor divider is before the unity gain follower, then the lowest possible impedance is provided for the 3.3V circuits. Also, the op amp can be powered from 3.3V, saving some power. If the X is made very large, then power consumed by the 5V side can be minimized.

If the attenuator is added after the unity gain follower, then the highest possible impedance is presented to the 5V source. The op amp must be powered from 5V and the impedance at the 3V side will depend upon the value of R1||R2.