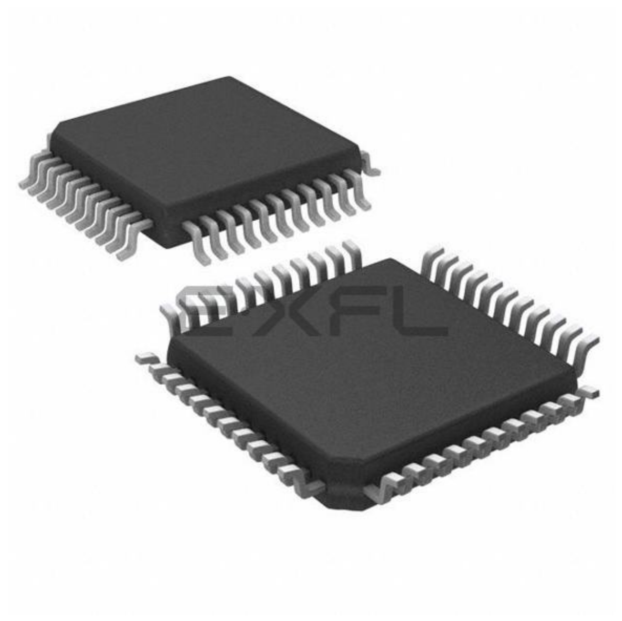


What is "Embedded - Microcontrollers"?

"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	20MHz
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	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c65b-20-pq

TIPS 'N TRICKS WITH SOFTWARE

To reduce costs, designers need to make the most of the available program memory in MCUs. Program memory is typically a large portion of the MCU cost. Optimizing the code helps to avoid buying more memory than needed. Here are some ideas that can help reduce code size.

TIP #15 Delay Techniques

- Use GOTO "next instruction" instead of two NOPs.
- Use CALL Rtrn as quad, 1 instruction NOP (where "Rtrn" is the exit label from existing subroutine).

Example 15-1

```
        NOP
        NOP             ;2 instructions, 2 cycles
```

```
        GOTO    $+1     ;1 instruction, 2 cycles
```

```
        CALL Rtrn       ;1 instruction, 4 cycles
        . . .
Rtrn    RETURN
```

MCUs are commonly used to interface with the "outside world" by means of a data bus, LEDs, buttons, latches, etc. Because the MCU runs at a fixed frequency, it will often need delay routines to meet setup/hold times of other devices, pause for a handshake or decrease the data rate for a shared bus.

Longer delays are well-suited for the DECFSZ and INCFSZ instructions where a variable is decremented or incremented until it reaches zero when a conditional jump is executed. For shorter delays of a few cycles, here a few ideas to decrease code size.

For a two-cycle delay, it is common to use two NOP instructions which uses two program memory locations. The same result can be achieved by using "goto $+1". The "$" represents the current program counter value in MPASM™ Assembler. When this instruction is encountered, the MCU will jump to the next memory location. This is what it would have done if two NOP's were used but since the GOTO instruction uses two instruction cycles to execute, a two-cycle delay was created. This created a two-cycle delay using only one location of program memory.

To create a four-cycle delay, add a label to an existing RETURN instruction in the code. In this example, the label "Rtrn" was added to the RETURN of subroutine that already existed somewhere in the code. When executing "CALL Rtrn", the MCU delays two instruction cycles to execute the CALL and two more to execute the RETURN. Instead of using four NOP instructions to create a four-cycle delay, the same result was achieved by adding a single CALL instruction.

CHAPTER 2
PIC® Microcontroller Low Power 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. The Flash-based PIC® microcontrollers (MCUs) are used in an wide range of everyday products, from smoke detectors, hospital ID tags and pet containment systems, to industrial, automotive and medical products.

PIC MCUs featuring nanoWatt technology implement a variety of important features which have become standard in PIC microcontrollers. Since the release of nanoWatt technology, changes in MCU process technology and improvements in performance have resulted in new requirements for lower power. PIC MCUs with nanoWatt eXtreme Low Power (nanoWatt XLP™) improve upon the original nanoWatt technology by dramatically reducing static power consumption and providing new flexibility for dynamic power management.

The following series of Tips n’ Tricks can be applied to many applications to make the most of PIC MCU nanoWatt and nanoWatt XLP devices.

GENERAL LOW POWER TIPS ‘N TRICKS

The following tips can be used with all PIC MCUs to reduce the power consumption of almost any application.

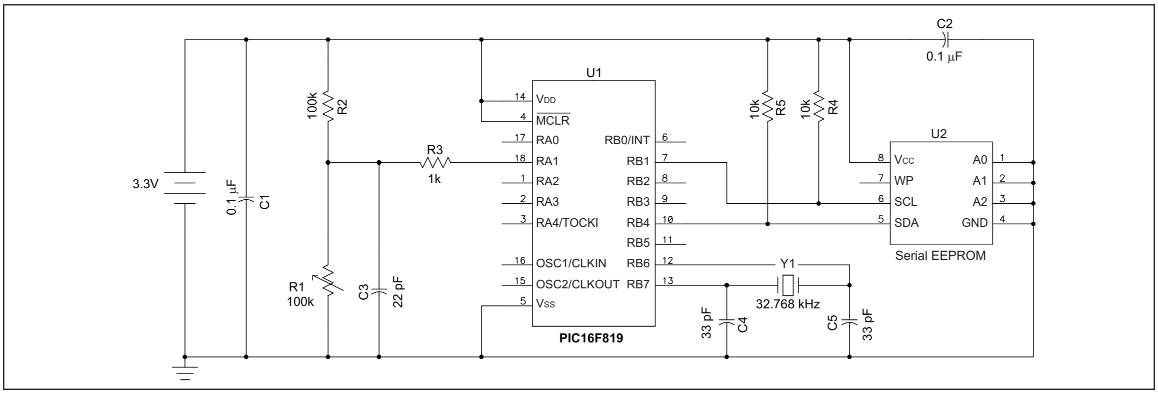
TIP #1 Switching Off External Circuits/Duty Cycle

All the low power modes in the world won't help your application if you are unable to control the power used by circuits external to the microprocessor. Lighting an LED is equivalent to running most PIC MCUs at 5V-20 MHz. When you are designing your circuitry, decide what physical modes or states are required and partition the electronics to shutdown unneeded circuitry.

Example:

The application is a long duration data recorder. It has a sensor, an EEPROM, a battery and a microprocessor. Every two seconds, it must take a sensor reading, scale the sensor data, store the scaled data in EEPROM and wait for the next sensor reading.

Figure: 1-1

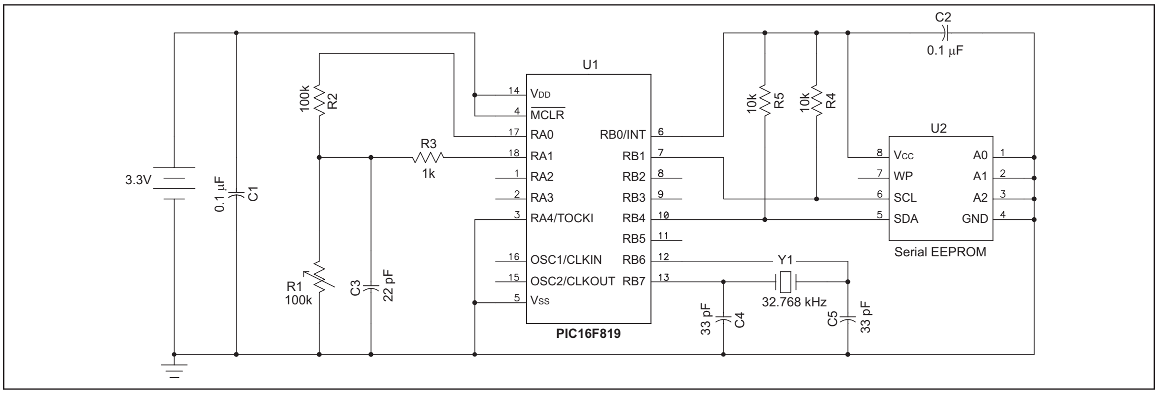


The system shown above is very simple and clearly has all the parts identified in the requirements. Unfortunately, it has a few problems in that the EEPROM, the sensor, and its bias circuit, are energized all the time. To get the minimum current draw for this design, it would be advantageous to shutdown these circuits when they are not required.

In Figure 1-2, I/O pins are used to power the EEPROM and the sensor. Many PIC MCU devices can source up to 20 mA of current from each I/O, so there is no need to provide additional components to switch the power.

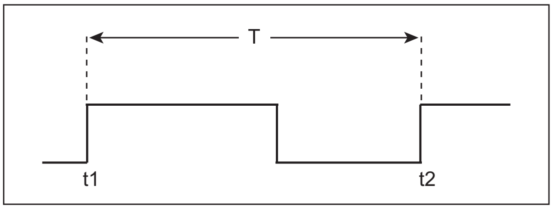
If more current than can be sourced by the PIC MCU is required, the PIC MCU can instead enable and disable a MOSFET to power the circuit. Refer to the data sheet for drive capabilities for a specific device.

Figure: 1-2



TIP #1 Measuring the Period of a Square Wave

Figure 1-1: Period



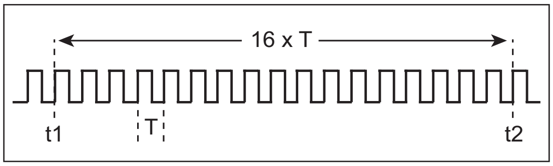
1. Configure control bits CCPxM3:CCPxM0 (CCPxCON<3:0>) to capture every rising edge of the waveform.
2. Configure the Timer1 prescaler so Timer1 with run TMAX(1) without overflowing.
3. Enable the CCP interrupt (CCPxIE bit).
4. When a CCP interrupt occurs:
   a) Subtract saved captured time (t1) from captured time (t2) and store (use Timer1 interrupt flag as overflow indicator).
   b) Save captured time (t2).
   c) Clear Timer1 flag if set.

The result obtained in step 4.a is the period (T).

Note 1: TMAX is the maximum pulse period that will occur.

TIP #2 Measuring the Period of a Square Wave with Averaging

Figure 2-1: Period Measurement



1. Configure control bits CCPxM3:CCPxM0 (CCPxCON<3:0>) to capture every 16th rising edge of the waveform.
2. Configure the Timer1 prescaler so Timer1 will run 16 TMAX(1) without overflowing.
3. Enable the CCP interrupt (CCPxIE bit).
4. When a CCP interrupt occurs:
   a) Subtract saved captured time (t1) from captured time (t2) and store (use Timer1 interrupt flag as overflow indicator).
   b) Save captured time (t2).
   c) Clear Timer1 flag if set.
   d) Shift value obtained in step 4.a right four times to divide by 16 – this result is the period (T).

Note 1: TMAX is the maximum pulse period that will occur.

The following are the advantages of this method as opposed to measuring the periods individually.

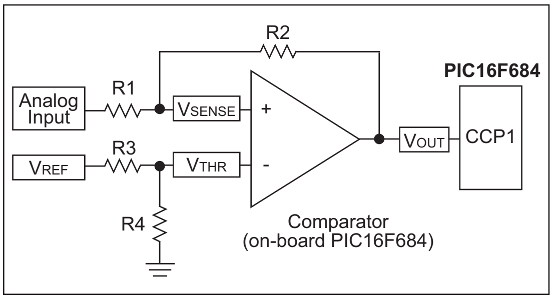
- Fewer CCP interrupts to disrupt program flow
- Averaging provides excellent noise immunity

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.

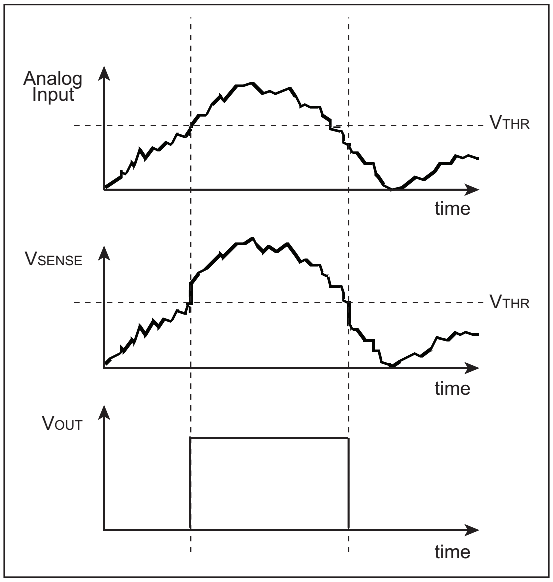
Figure 6-1 shows an example circuit using the peripherals of the PIC16F684.

Figure 6-1: Circuit



R3 and R4 set the threshold voltage for the comparator. When the analog input reaches the threshold voltage, V_{OUT} 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 V_{SENSE} does when the analog input reaches the threshold voltage.

Figure 6-2: Signal Comparison

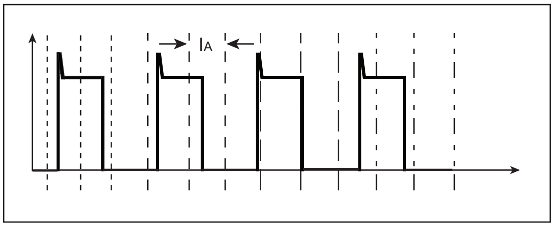


The CCP module, configured in Capture mode, can time the length between the rising edges of the comparator output (V_{OUT}.) This is the period of the analog input, provided the analog signal reaches V_{THR} during every period.

TIP #12 Repetitive Phase Shifted Sampling

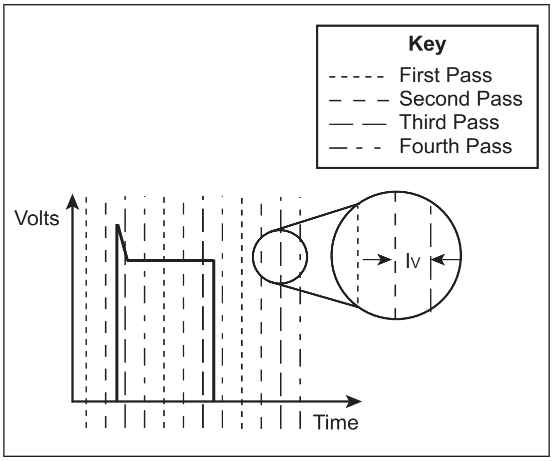
Repetitive phase shifted sampling is a technique to artificially increase the sampling rate of an A/D converter when sampling waveforms that are both periodic and constant from period to period. The technique works by capturing regularly spaced samples of the waveform from the start to finish of the waveform's period. Sampling of the next waveform is then performed in the same manner, except that the start of the sample sequence is delayed a percentage of the sampling period. Subsequent waveforms are also sampled, with each sample sequence slightly delayed from the last, until the delayed start of the sample sequence is equal to one sample period. Interleaving the sample sets then produces a sample set of the waveform at a higher sample rate. Figure 12-1 shows an example of a high frequency waveform.

Figure 12-1: High Frequency Periodic Waveform



As indicated in the key, the finely dotted lines show where the A/D readings are taken during the first period of the waveform. The medium sized dashed lines show when the A/D readings are taken during the second period, and so on. Figure 12-2 shows these readings transposed onto one period.

Figure 12-2: Transposed Waveform



The CCP module is configured in Compare Special Event Trigger mode to accomplish this task. The phase shift is implemented by picking values of CCPRxL and CCPRxH that are not synchronous with the period of the sampling waveform. For instance, if the period of a waveform is 100 μs, then sampling at a rate of once every 22 μs will give the following set of sample times over 11 periods (all values in μs).

1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
0	10	20	8	18	6	16	4	14	2	12
22	32	42	30	40	28	38	26	36	24	34
44	54	64	52	62	50	60	48	58	46	56
66	76	86	74	84	72	82	70	80	68	78
88	98		96		94		92		90	

When these numbers are placed in sequential order, they reveal a virtual sampling interval (I_V) of 2 μs from 0 μs to 100 μs, although the actual sampling interval (I_A) is 22 μs.

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

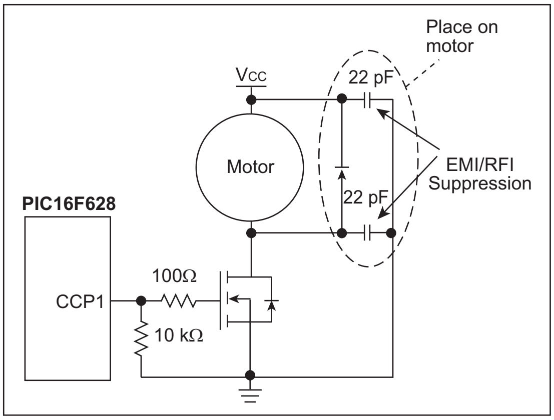


Figure 14-1 shows a unidirectional speed controller circuit for a brushed DC motor. Motor speed is proportional to the duty cycle of the PWM output on the CCP1 pin. The following steps show how to configure the PIC16F628 to generate a 20 kHz PWM with 50% duty cycle. The microcontroller is running on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

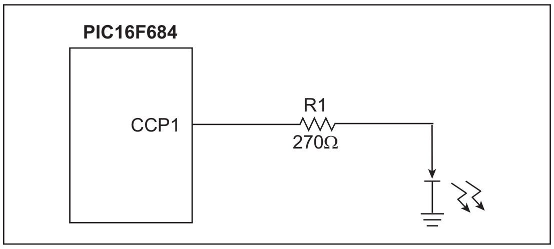
Step #4: Configure CCP1CON

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

TIP #18 Varying LED Intensity

The intensity of an LED can be varied by pulse-width modulating the voltage across the LED. A microcontroller typically drives an LED with the circuit shown in Figure 18-1. The purpose of R1 is to limit the LED current so that the LED runs in its specified current and voltage range, typically around 1.4 volts at 20 mA. Modulating the LED drive pin on the microcontroller will vary the average current seen by the LED and thus its intensity. As mentioned in Tip #13, LEDs and other light sources should be modulated at no less than 100 Hz in order to prevent noticeable flicker.

Figure 18-1: LED Drive

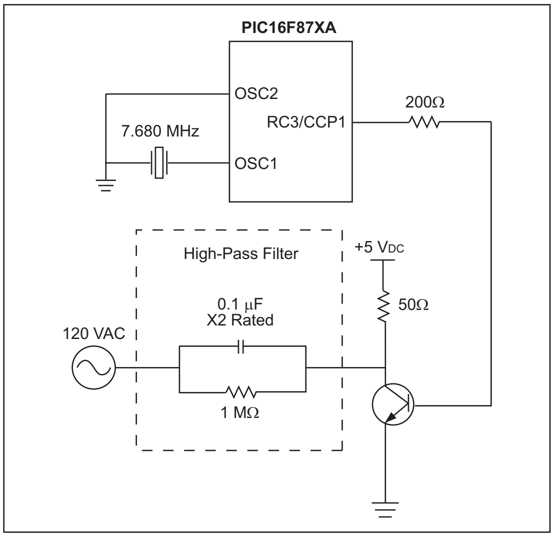


The CCP module, configured in PWM mode, is ideal for varying the intensity of an LED. Adjustments to the intensity of the LED are made by simply varying the duty cycle of the PWM signal driving the LED. This is accomplished by varying the CCPRxL register between 0 and 0xFF.

TIP #19 Generating X-10® Carrier Frequency

X-10 uses a piggybacked 120 kHz square wave (at 50% duty cycle) to transmit information over 60 Hz power lines. The CCP module, running in PWM mode, can accurately create the 120 kHz square wave, referred to as the carrier frequency. Figure 19-1 shows how the 120 kHz carrier frequency is piggybacked onto the sinusoidal 60 Hz power waveform.

Figure 19-1: Carrier Frequency With Sinusoidal Waveform

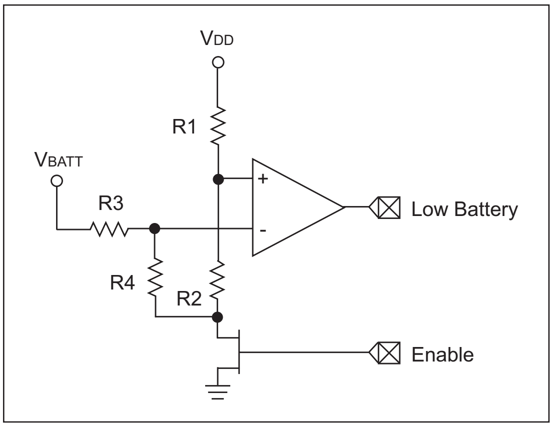


X-10 specifies the carrier frequency at 120 kHz (± 2 kHz). The system oscillator in Figure 18-1 is chosen to be 7.680 MHz, so that the CCP module can generate precisely 120 kHz. X-10 requires that the carrier frequency be turned on and off at different points on the 60 Hz power waveform. This is accomplished by configuring the TRIS register for the CCP1 pin as either an input (carrier frequency off) or an output (carrier frequency on). Refer to Application Note AN236 "*X-10 Home Automation Using the PIC16F877A*" for more details on X-10 and for source code for setting up the CCP module appropriately.

TIP #1 Low Battery Detection

When operating from a battery power supply, it is important for a circuit to be able to determine when the battery charge is insufficient for normal operation of the circuit. Typically, this is a comparator-based circuit similar to the Programmable Low Voltage Detect (PLVD) peripheral. If the PLVD peripheral is not available in the microcontroller, a similar circuit can be constructed using a comparator and a few external components (see Figure 1-1 and Figure 1-2). The circuit in Figure 1-1 assumes that the microcontroller is operating from a regulated supply voltage. The circuit in Figure 1-2 assumes that the microcontroller supply is unregulated.

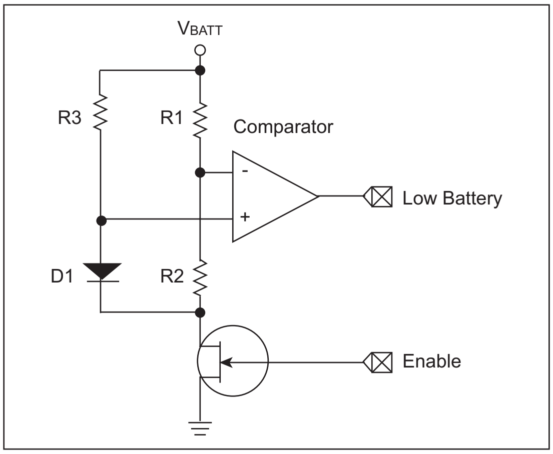
Figure 1-1: Regulated Supply



The comparator will trip when the battery voltage, V_{BATT} = 5.7V: R1 = 33k, R2 = 10k, R3 = 39k, R4 = 10k, V_{DD} = 5V.

In Figure 1-1, resistors R1 and R2 are chosen to place the voltage at the non-inverting input at approximately 25% of V_{DD}. R3 and R4 are chosen to set the inverting input voltage equal to the non-inverting input voltage when the battery voltage is equal to the minimum operating voltage for the system.

Figure 1-2: Unregulated Supply



Comparator will trip when V_{BATT} = 3V: R1 = 33k, R2 = 10k and R3 = 470Ω.

In Figure 1-2, resistor R3 is chosen to bias diode D1 above its forward voltage when V_{BATT} is equal to the minimum battery voltage for the system. Resistors R1 and R2 are chosen to set the inverting input voltage equal to the forward voltage of D1.

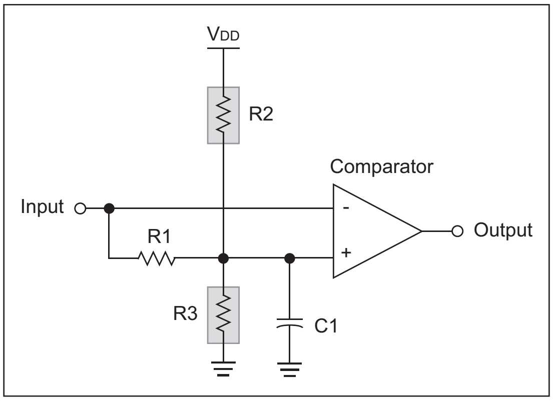
TIP #6 Data Slicer

In both wired and wireless data transmission, the data signal may be subject to DC offset shifts due to temperature shifts, ground currents or other factors in the system. When this happens, using a simple level comparison to recover the data is not possible because the DC offset may exceed the peak-to-peak amplitude of the signal. The circuit typically used to recover the signal in this situation is a data slicer.

The data slicer shown in Figure 6-1 operates by comparing the incoming signal with a sliding reference derived from the average DC value of the incoming signal. The DC average value is found using a simple RC low-pass filter (R1 and C1). The corner frequency of the RC filter should be high enough to ignore the shifts in the DC level while low enough to pass the data being transferred.

Resistors R2 and R3 are optional. They provide a slight bias to the reference, either high or low, to give a preference to the state of the output when no data is being received. R2 will bias the output low and R3 will bias the output high. Only one resistor should be used at a time, and its value should be at least 50 to 100 times larger than R1.

Figure 6-1: Data Slicer



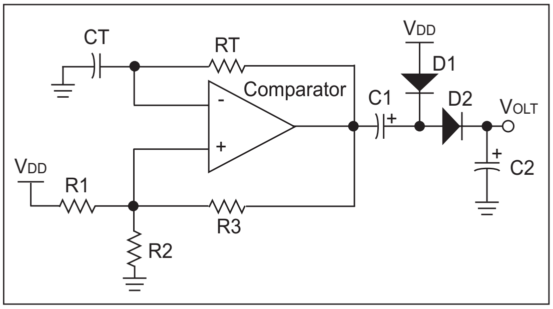
Example:

Data rate of 10 kbits/second. A low pass filter frequency of 500 Hz: R1 = 10k, C1 = 33 μF. R2 or R3 should be 500k to 1 MB.

TIP #10 Capacitive Voltage Doubler

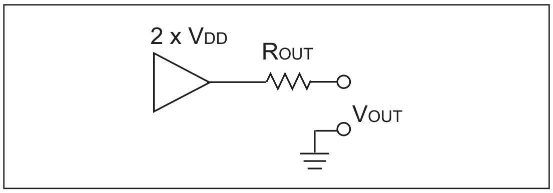
This tip takes the multi-vibrator described in Tip #8 and builds a capacitive voltage doubler around it (see Figure 10-1). The circuit works by alternately charging capacitor C1 through diode D1, and then charge balancing the energy in C1 with C2 through diode D2. At the start of the cycle, the output 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 #12 Making an Op Amp Out of a Comparator

When interfacing to a sensor, some gain is typically required to match the full range of the sensor to the full range of an ADC. Usually this is done with an operational amplifier, however, in cost sensitive applications, an additional active component may exceed the budget. This tip shows how an on-chip comparator can be used as an op amp like gain stage for slow sensor signals. Both an inverting and non-inverting topology are shown (see Figure 12-1 and Figure 12-2).

Figure 12-1: Non-Inverting Amplifier

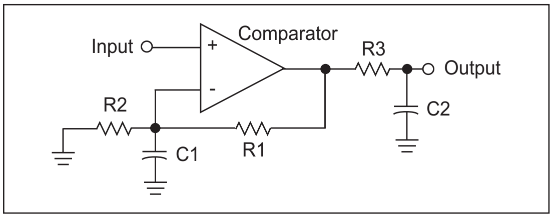
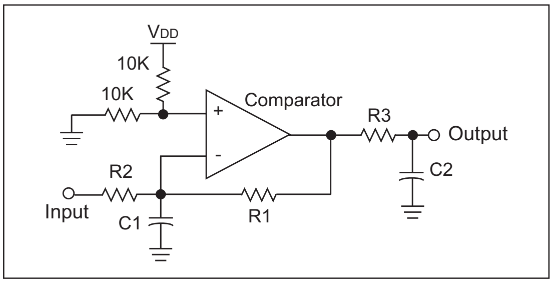


Figure 12-2: Inverting Amplifier



To design a non-inverting amplifier, choose resistors R1 and R2 using the Gain formula for an op amp non-inverting amplifier (see Equation 12-1).

Equation 12-1

$$\text{Gain} = \frac{R1 + R2}{R2}$$

Once the gain has been determined, values for R3 and C2 can be determined. R3 and C2 form a low-pass filter on the output of the amplifier. The corner frequency of the low pass should be 2 to 3 times the maximum frequency of the signal being amplified to prevent attenuation of the signal, and R3 should be kept small to minimize the output impedance of the amplifier. Equation 12-2 shows the relationship between R3, C2 and the corner frequency of the low pass filter.

Equation 12-2

$$F_{CORNER} = \frac{1}{2 * \quad * R3 * C2}$$

A value for C1 can then be determined using Equation 12-3. The corner frequency should be the same as Equation 12-3.

Equation 12-3

$$F_{CORNER} = \frac{1}{2 * \quad * (R1 \text{ II } R2) * C2}$$

To design an inverting amp, choose resistors R1 and R2 using the Gain formula for an op amp inverting amplifier (see Equation 12-4).

Equation 12-4

$$\text{Gain} = \frac{R_1}{R_2}$$

Then choose values for the resistor divider formed by R4 and R5. Finally choose C1 and C2 as shown in the non-inverting amplifier design.

Example:

- For C2 will set the corner F
- Gain = 6.156, R1 = R3 = 19.8k
- R2 = 3.84k, C1 = .047 µF, F_{CORNER} = 171 Hz
- C2 = .22 µF

CHAPTER 5
DC Motor Control Tips 'n Tricks

Table Of Contents

TIPS 'N TRICKS INTRODUCTION

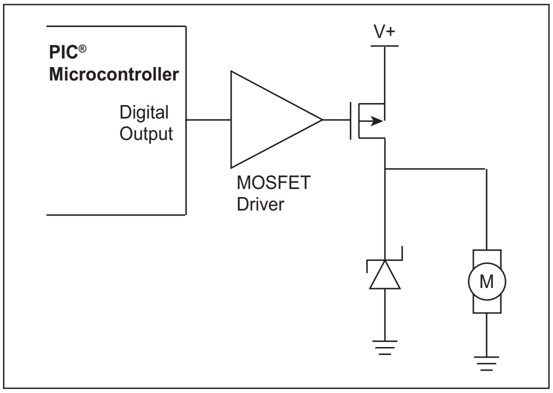
Every motor control circuit can be divided into the drive electronics and the controlling software. These two pieces can be fairly simple or extremely complicated depending upon the motor type, the system requirements and the hardware/software complexity trade-off. Generally, higher performance systems require more complicated hardware. This booklet describes many basic circuits and software building blocks commonly used to control motors. The booklet also provides references to Microchip application notes that describe many motor control concepts in more detail. The application notes can be found on the Microchip web site at www.microchip.com.

Additional motor control design information can be found at the Motor Control Design Center (www.microchip.com/motor).

TIP #1 Brushed DC Motor Drive Circuits

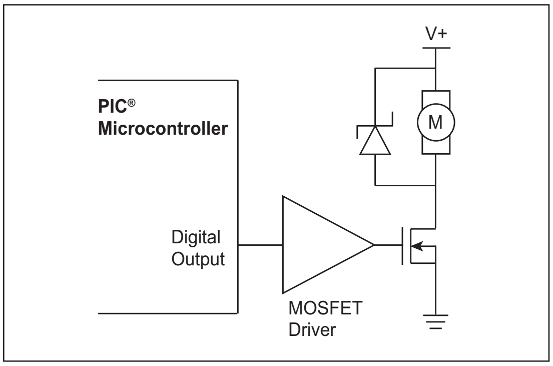
All motors require drive circuitry which controls the current flow through the motor windings. This includes the direction and magnitude of the current flow. The simplest type of motor, to drive, is the Brushed DC motor. Drive circuits for this type of motor are shown below.

Figure 1-1: High Side Drive



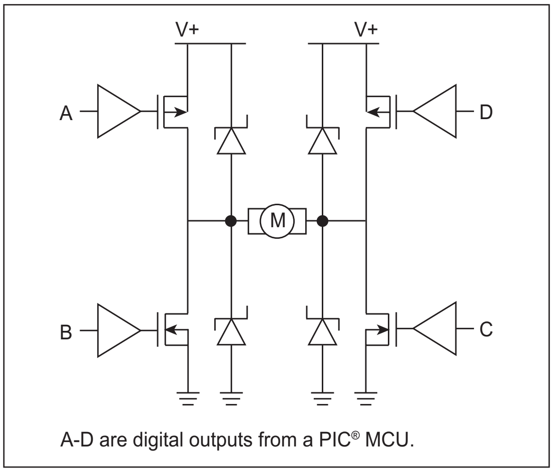
This drive can control a Brushed DC motor in one direction. This drive is often used in safety critical applications because a short circuit at the motor terminals cannot turn the motor on.

Figure 1-2: Low Side Drive



This is the lowest cost drive technique because of the MOSFET drive simplicity. Most applications can simply use an output pin from the PIC® microcontroller to turn the MOSFET on.

Figure 1-3: H-Bridge Drive



The H-Bridge derived its name from the common way the circuit is drawn. This is the only solid state way to operate a motor in both directions.

Application notes that drive Brushed DC motors are listed below and can be found on the Microchip web site at: www.microchip.com.

- AN847, “*RC Model Aircraft Motor Control*” (DS00847)
- AN893, “*Low-cost Bidirectional Brushed DC Motor Control Using the PIC16F684*” (DS00893)
- AN905, “*Brushed DC Motor Fundamentals*” (DS00905)

TIP #2 Brushless DC Motor Drive Circuits

A Brushless DC motor is a good example of simplified hardware increasing the control complexity. The motor cannot commutate the windings (switch the current flow), so the control circuit and software must control the current flow correctly to keep the motor turning smoothly. The circuit is a simple half-bridge on each of the three motor windings.

There are two basic commutation methods for Brushless DC motors; sensored and sensorless. Because it is critical to know the position of the motor so the correct winding can be energized, some method of detecting the rotor position is required. A motor with sensors will directly report the current position to the controller. Driving a sensored motor requires a look-up table. The current sensor position directly correlates to a commutation pattern for the bridge circuits.

Without sensors, another property of the motor must be sensed to find the position. A popular method for sensorless applications is to measure the back EMF voltage that is naturally generated by the motor magnets and windings. The induced voltage in the un-driven winding can be sensed and used to determine the current speed of the motor. Then, the next commutation pattern can be determined by a time delay from the previous pattern.

Sensorless motors are lower cost due to the lack of the sensors, but they are more complicated to drive. A sensorless motor performs very well in applications that don't require the motor to start and stop. A sensor motor would be a better choice in applications that must periodically stop the motor.

Figure 2-1: 3 Phase Brushless DC Motor Control

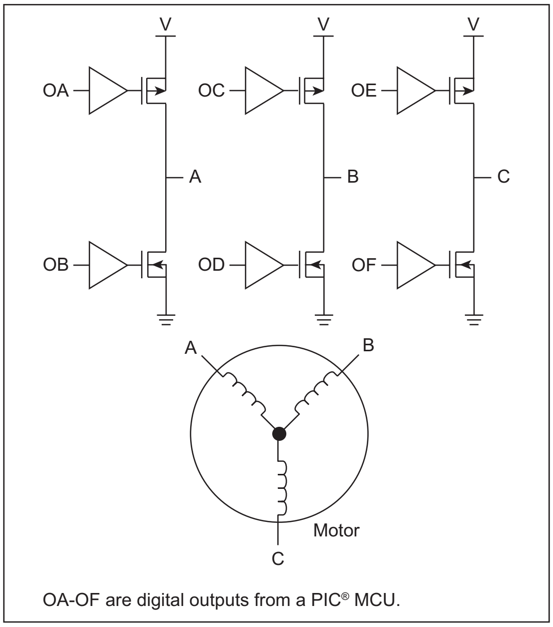


Figure 2-2: Back EMF Sensing (Sensorless Motor)

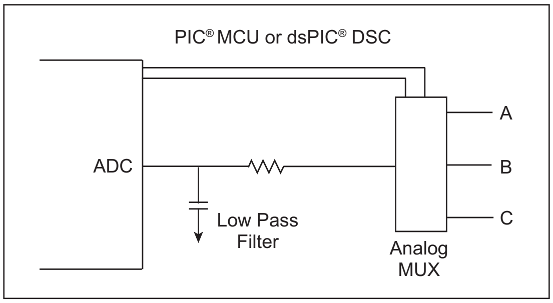
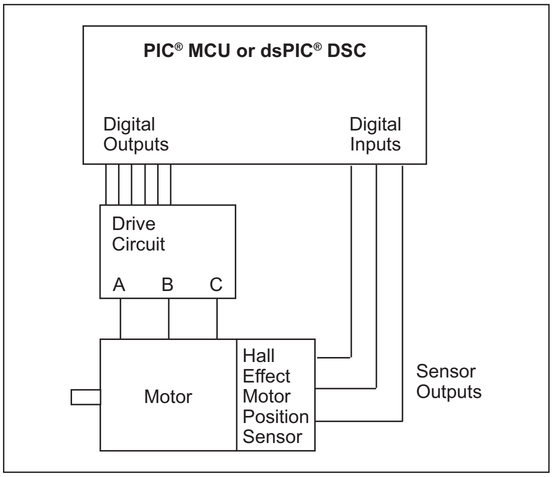


Figure 2-3: Quadrature Decoder (Sensor Motor)



Application notes describing Brushless DC Motor Control are listed below and can be found on the Microchip web site at: www.microchip.com.

- AN857, “*Brushless DC Motor Control Made Easy*” (DS00857)
- AN885, “*Brushless DC Motor Fundamentals*” (DS00885)
- AN899, “*Brushless DC Motor Control Using PIC18FXX31*” (DS00899)
- AN901, “*Using the dsPIC30F for Sensorless BLDC Control*” (DS00901)
- AN957, “*Sensored BLDC Motor Control Using dsPIC30F2010*” (DS00957)
- AN992, “*Sensorless BLDC Motor Control Using dsPIC30F2010*” (DS00992)
- AN1017, “*Sinusoidal Control of PMSM with dsPIC30F DSC*” (DS01017)
- GS005, “*Using the dsPIC30F Sensorless Motor Tuning Interface*” (DS93005)

TIP #3 Stepper Motor Drive Circuits

Stepper motors are similar to Brushless DC motors in that the control system must commutate the motor through the entire rotation cycle. Unlike the brushless motor, the position and speed of a stepping motor is predictable and does not require the use of sensors.

There are two basic types of stepper motors, although some motors are built to be used in either mode. The simplest stepper motor is the unipolar motor. This motor has four drive connections and one or two center tap wires that are tied to ground or Vsupply, depending on the implementation. Other motor types are the bipolar stepper and various combinations of unipolar and bipolar, as shown in Figure 3-1 and Figure 3-2. When each drive connection is energized, one coil is driven and the motor rotates one step. The process is repeated until all the windings have been energized. To increase the step rate, often the voltage is increased beyond the motors rated voltage. If the voltage is increased, some method of preventing an over current situation is required.

There are many ways to control the winding current, but the most popular is a chopper system that turns off current when it reaches an upper limit and enables the current flow a short time later. Current sensor systems are discussed in Tip #6. Some systems are built with a current chopper, but they do not detect the current, rather the system is designed to begin a fixed period chopping cycle after the motor has stepped to the next position. These are simpler systems to build, as they only require a change in the software.

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).

CHAPTER 7
Intelligent Power Supply Design 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.

While some designs such as Switch Mode Power Supplies (SMPS) are traditionally implemented using a purely analog control scheme, these designs can benefit from the configurability and intelligence that can only be realized by adding a microcontroller.

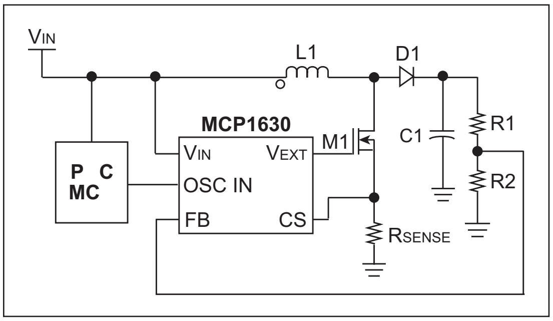
This document showcases several examples in which a PIC microcontroller may be used to increase the functionality of a design with a minimal increase in cost.

Several of the tips provide working software examples or reference other documents for more information. The software and referenced documents can be found on the Microchip web site at www.microchip.com/tipsntricks.

TIP #5 Using a PIC® Microcontroller as a Clock Source for a SMPS PWM Generator

A PIC MCU can be used as the clock source for a PWM generator, such as the MCP1630.

Figure 5-1: PIC MCU and MCP1630 Example Boost Application



The MCP1630 begins its cycle when its clock/oscillator source transitions from high-to-low, causing its PWM output to go high state. The PWM pulse can be terminated in any of three ways:

1. The sensed current in the magnetic device reaches 1/3 of the error amplifier output.
2. The voltage at the Feedback (FB) pin is higher than the reference voltage (V_{REF}).
3. The clock/oscillator source transitions from low-to-high.

The switching frequency of the MCP1630 can be adjusted by changing the frequency of the clock source. The maximum on-timer of the MCP1630 PWM can be adjusted by changing the duty cycle of the clock source.

The PIC MCU has several options for providing this clock source:

• The Fosc/4 pin can be enabled. This will produce a 50% duty cycle square wave that is 1/4th of the oscillator frequency. Tip #4 provides both example software and information on clock dithering using the FOSC/4 output.

• For PIC MCUs equipped with a Capture/Compare/PWM (CCP) or Enhanced CCP (ECCP) module, a variable frequency, variable duty cycle signal can be created with little software overhead. This PWM signal is entirely under software control and allows advanced features, such as soft-start, to be implemented using software.

• For smaller parts that do not have a CCP or ECCP module, a software PWM can be created. Tips #1 and #2 use software PWM for soft-start and provide software examples.

TIP #16 5V → 3.3V Active Analog Attenuator

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

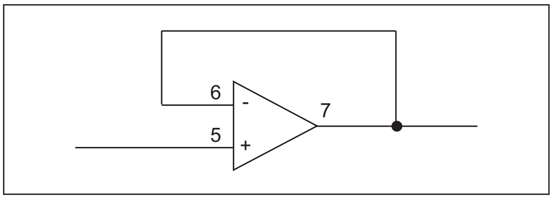
The simplest method of converting a 5V analog signal to a 3.3V analog signal is to use a resistor divider with a ratio R1:R2 of 1.7:3.3. However, there are a few problems with this.

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).

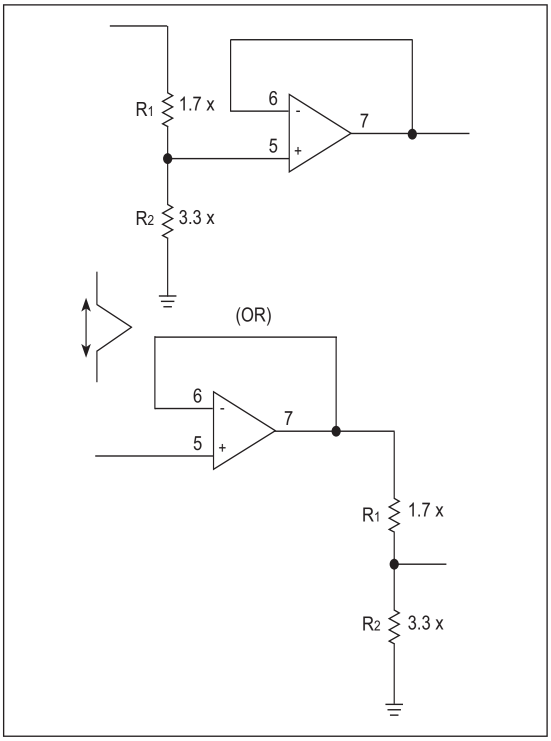
Figure 16-1: Unity Gain



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.