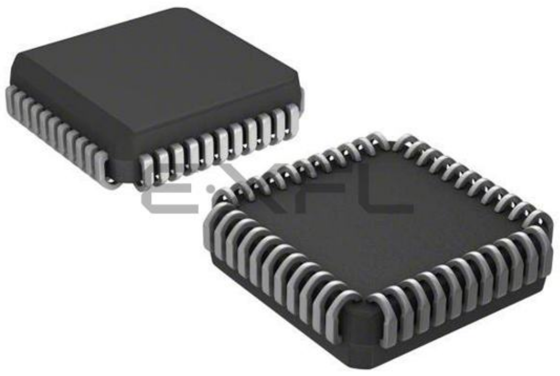


Welcome to E-XFL.COM

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	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-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c65b-20-l

Tips ‘n Tricks TABLE OF CONTENTS

TIP #5 Scanning Many Keys With One Input

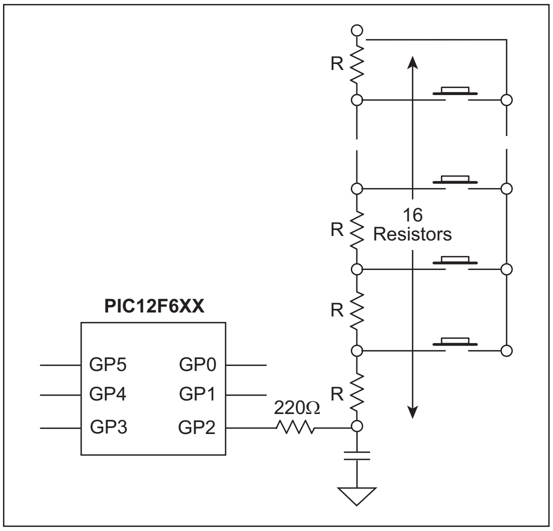
The time required to charge a capacitor depends on resistance between V_{DD} and capacitor. When a button is pressed, V_{DD} is supplied to a different point in the resistor ladder. The resistance between V_{DD} and the capacitor is reduced, which reduces the charge time of the capacitor. A timer is used with a comparator or changing digital input to measure the capacitor charge time. The charge time is used to determine which button is pressed.

Software sequence:

1. Configure GP2 to output a low voltage to discharge capacitor through I/O resistor.
2. Configure GP2 as one comparator input and CV_{REF} as the other.
3. Use a timer to measure when the comparator trips. If the time measured is greater than the maximum allowed time, then repeat; otherwise determine which button is pressed.

When a key is pressed, the voltage divider network changes the RC ramp rate.

Figure 5-1



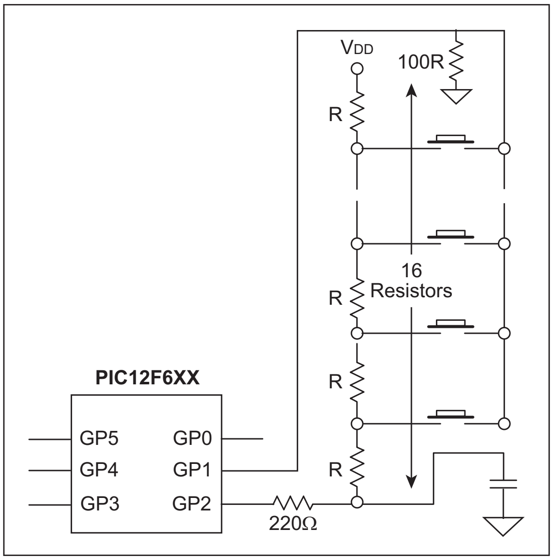
See AN512, “*Implementing Ohmmeter/ Temperature Sensor*” for code ideas.

TIP #6 Scanning Many Keys and Wake-up From Sleep

An additional I/O can be added to wake the part when a button is pressed. Prior to Sleep, configure GP1 as an input with interrupt-on-change enabled and GP2 to output high. The pull-down resistor holds GP1 low until a button is pressed. GP1 is then pulled high via GP2 and V_{DD} generating an interrupt. After wake-up, GP2 is configured to output low to discharge the capacitor through the 220Ω resistor. GP1 is set to output high and GP2 is set to an input to measure the capacitor charge time.

- GP1 pin connected to key common
- Enable wake-up on port change
- Set GP1 as input and GP2 high prior to Sleep
- If key is pressed the PIC MCU wakes up, GP2 must be set low to discharge capacitor
- Set GP1 high upon wake-up to scan keystroke

Figure 6-1



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 #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

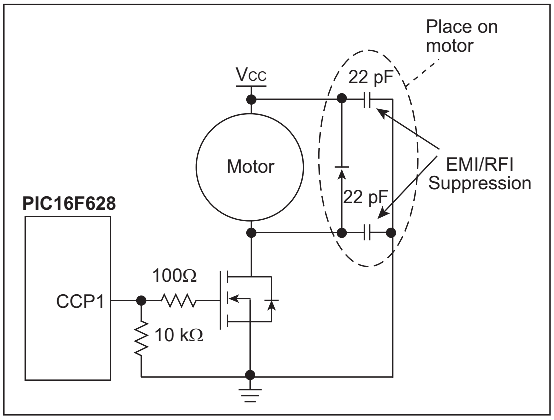


Figure 14-1 shows a unidirectional speed controller circuit for a brushed DC motor. Motor speed is proportional to the duty cycle of the PWM output on the CCP1 pin. The following steps show how to configure the PIC16F628 to generate a 20 kHz PWM with 50% duty cycle. The microcontroller is running on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

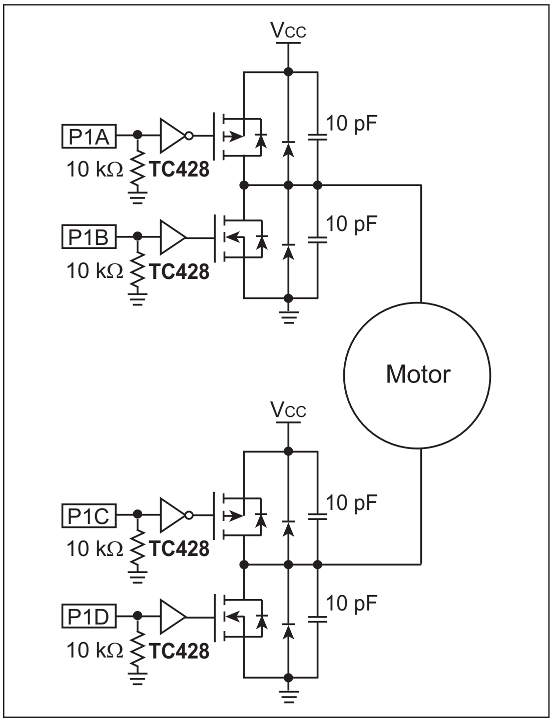
b) CCPR1L = 0x1FF >> 2 = 0x7F, CCP1CON<5:4> = 3

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 #15 Bidirectional Brushed DC Motor Control Using ECCP

Figure 15-1: Full-Bridge BDC Drive Circuit



The ECCP module has brushed DC motor control options built into it. Figure 15-1 shows how a full-bridge drive circuit is connected to a BDC motor. The connections P1A, P1B, P1C and P1D are all ECCP outputs when the module in configured in “Full-bridge Output Forward” or “Full-bridge Output Reverse” modes (CCP1CON<7:6>). For the circuit shown in Figure 15-1, the ECCP module should be configured in PWM mode: P1A, P1C active high; P1B, P1D active high (CCP1CON<3:1>). The reason for this is the MOSFET drivers (TC428) are configured so a high input will turn on the respective MOSFET.

The following table shows the relation between the states of operation, the states of the ECCP pins and the ECCP Configuration register.

State	P1A	P1B	P1C	P1D	CCP1CON
Forward	1	tri-state	tri-state	mod	‘b01xx1100’
Reverse	tri-state	mod	1	tri-state	‘b11xx1100’
Coast	tri-state	tri-state	tri-state	tri-state	N/A
Brake	tri-state	1	1	tri-state	N/A

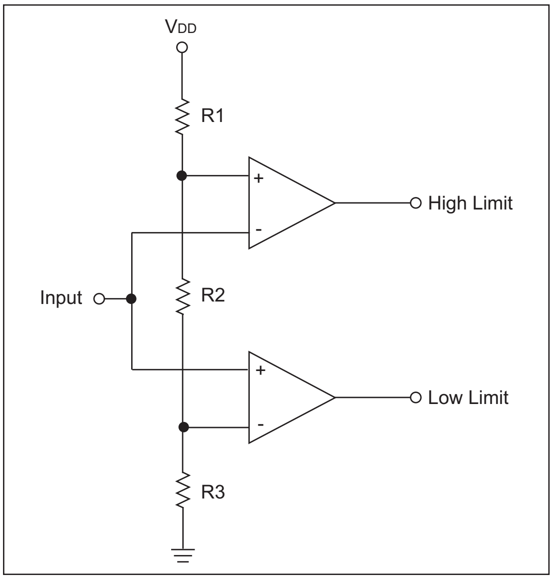
Legend: ‘1’ = high, ‘0’ = low, mod = modulated, tri-state = pin configured as input

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

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

Equation 5-2

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

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.

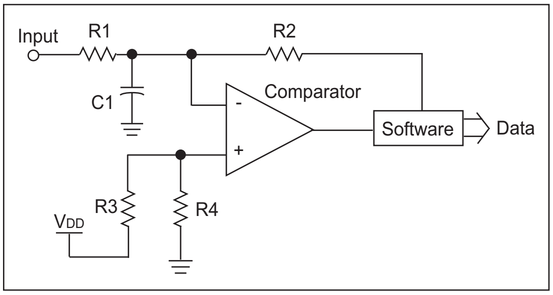
To add hysteresis to the LOW LIMIT comparator, choose a suitable value for Req, 1k to 10 k , and place it between the circuit input and the non-inverting input of the LOW LIMIT comparator. Then calculate the needed feedback resistor using Equation 3-4 and Equation 3-5.

TIP #14 Delta-Sigma ADC

This tip describes the creation of a hardware/software-based Delta-Sigma ADC. A Delta-Sigma ADC is based on a Delta-Sigma modulator composed of an integrator, a comparator, a clock sampler and a 1-bit DAC output. In this example, the integrator is formed by R1 and C1. The comparator is an on-chip voltage comparator. The clock sampler is implemented in software and the 1-bit DAC output is a single I/O pin. The DAC output feeds back into the integrator through R2.

Resistors R3 and R4 form a $V_{DD}/2$ reference for the circuit (see Figure 14-1).

Figure 14-1: Delta-Sigma Modulator



In operation, the feedback output from the software is a time sampled copy of the comparator output. In normal operation, the modulator output generates a PWM signal which is inversely proportional to the input voltage. As the input voltage increases, the PWM signal will drop in duty cycle to compensate. As the input decreases, the duty cycle rises.

To perform an A-to-D conversion, the duty cycle must be integrated over time, digitally, to integrate the duty cycle to a binary value. The software starts two counters. The first counts the total number of samples in the conversion and the second counts the number of samples that were low. The ratio of the two counts is equal to the ratio of the input voltage over V_{DD}.

Note: This assumes that R1 and R2 are equal and R3 is equal to R4. If R1 and R2 are not equal, then the input voltage is also scaled by the ratio of R2 over R1, and R3 must still be equal to R4.

For a more complete description of the operation of a Delta-Sigma ADC and example firmware, see Application Note AN700 “*Make A Delta-Sigma Converter Using a Microcontroller’s Analog Comparator Module*.”

Example:

- R3 = R4 = 10 kHz
- R1 = R2 = 5.1k
- C1 = 1000 pF

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

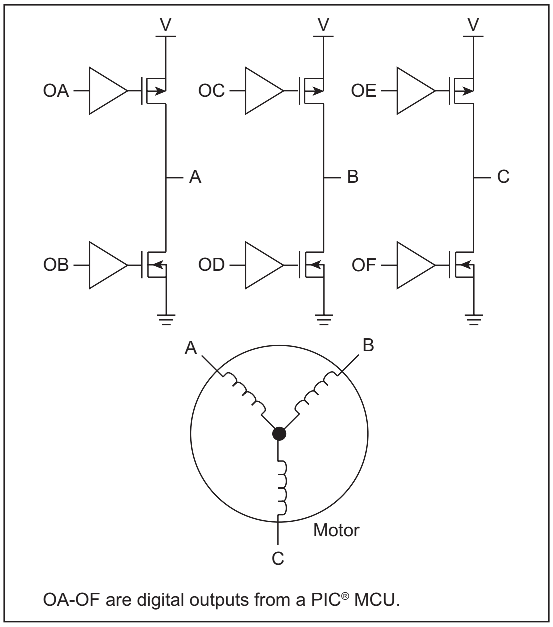
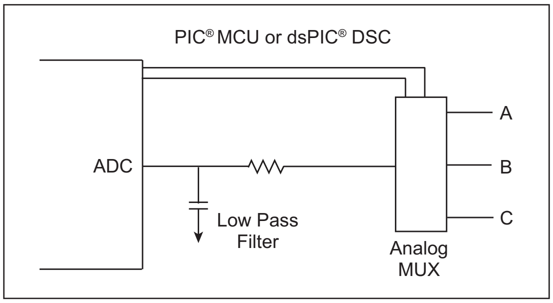


Figure 2-2: Back EMF Sensing (Sensorless Motor)



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
```

NOTES:

CHAPTER 6
LCD PIC® Microcontroller Tips ‘n Tricks

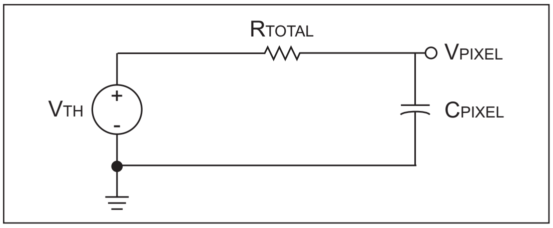
Table Of Contents

TIPS ‘N TRICKS INTRODUCTION

Using an LCD PIC® MCU for any embedded application can provide the benefits of system control and human interface via an LCD. Design practices for LCD applications can be further enhanced through the implementation of these suggested “Tips ‘n Tricks”.

This booklet describes many basic circuits and software building blocks commonly used for driving LCD displays. The booklet also provides references to Microchip application notes that describe many LCD concepts in more detail.

Figure 3-4: Voltage Change Across Pixel



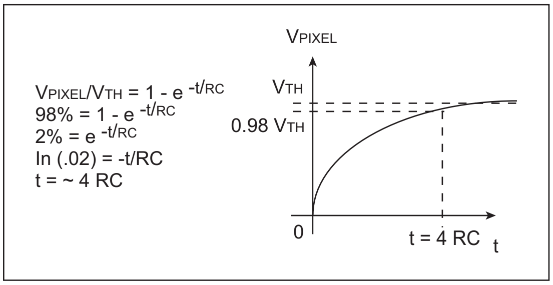
The step response of the voltage across a pixel is subject to the following equation:

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

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

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

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

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

Equation 3-5

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

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.

TIP #7 Driving Common Backlights

Any application that operates in a low light condition requires a backlight. Most low-cost applications use one of the following backlights:

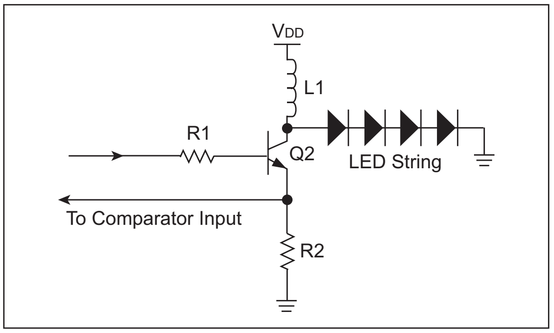
1) Electroluminescent (EL)
2) LEDs in series
3) LEDs in parallel

Other backlight technologies, such as CCFL, are more commonly used in high brightness graphical panels, such as those found in laptop computers. The use of white LEDs is also more common in color LCDs, where a white light source is required to generate the colors.

Driving an EL panel simply requires an AC signal. You may be able to generate this signal simply by using an unused segment on the LCD controller. The signal can also be generated by a CCP module or through software. The AC signal will need to pass through a transformer for voltage gain to generate the required voltage across the panel.

LEDs in series can be easily driven with a boost power supply. In the following diagram, a simple boost supply is shown. In this circuit, a pulse is applied to the transistor. The pulse duration is controlled by current through R2. When the pulse is turned off, the current stored in the inductor will be transferred to the LEDs. The voltage will rise to the level required to drive the current through the LEDs. The breakdown voltage of the transistor must be equal to the forward voltage of the LEDs multiplied by the number of LEDs. The comparator voltage reference can be adjusted in software to change the output level of the LEDs.

Figure 7-1: Simple Boost Supply



If the LEDs are in parallel, the drive is much simpler. In this case, a single transistor can be used to sink the current of many LEDs in parallel. The transistor can be modulated by PWM to achieve the desired output level. If V_{DD} is higher than the maximum forward voltage, a resistor can be added to control the current, or the transistor PWM duty cycle can be adjusted to assure the LEDs are operating within their specification.

Figure 7-2: LEDs in Parallel

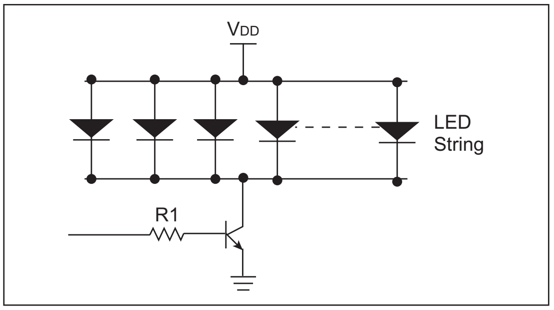
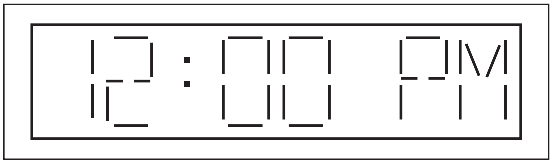
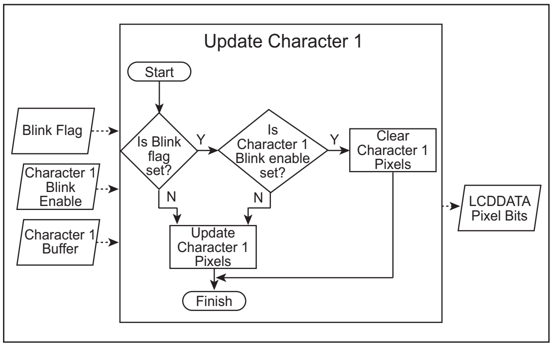


Figure 11-1: Common Clock Application



Fortunately, blinking is quite easy to implement. There are many ways to implement a blinking effect in software. Any regular event can be used to update a blink period counter. A blink flag can be toggled each time the blink period elapses. Each character or display element that you want to blink can be assigned a corresponding blink enable flag. The flowchart for updating the display would look like:

Figure 11-2: Updating Display Flowchart



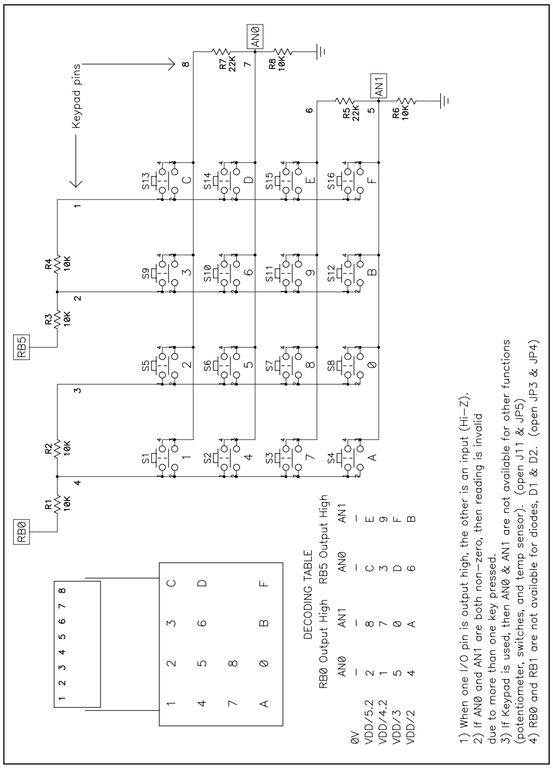
TIP #12 4 x 4 Keypad Interface that Conserves Pins for LCD Segment Drivers

A typical digital interface to a 4 x 4 keypad uses 8 digital I/O pins. But using eight pins as digital I/Os can take away from the number of segment driver pins available to interface to an LCD.

By using 2 digital I/O pins and 2 analog input pins, it is possible to add a 4 x 4 keypad to the PIC microcontroller without sacrificing any of its LCD segment driver pins.

The schematic for keypad hook-up is shown in Figure 12-1. This example uses the PIC18F8490, but the technique could be used on any of the LCD PIC MCUs.

Figure 12-1: Keypad Hook-up Schematic



DECODING TABLE

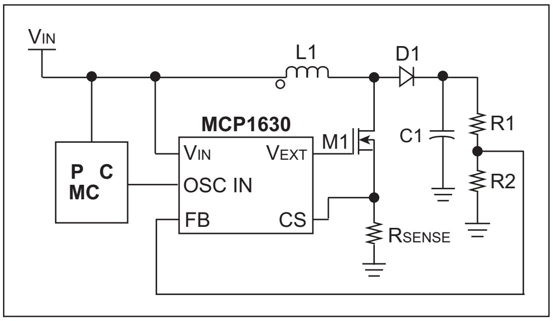
	RB0 Output High		RB5 Output High	
	AN0	AN1	AN0	AN1
0V	–	–	–	–
VDD/5.2	2	8	C	E
VDD/4.2	1	7	3	9
VDD/3	5	0	D	F
VDD/2	4	A	6	B

1) When one I/O pin is output high, the other is an input (Hi-Z).
2) If AN0 and AN1 are both non-zero, then reading is invalid due to more than one key pressed.
3) If Keypad is used, then AN0 & AN1 are not available for other functions (potentiometer, switches, and temp sensor). (open J11 & JP5)
4) RB0 and RB1 are not available for diodes, D1 & D2. (open JP3 & JP4)

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.

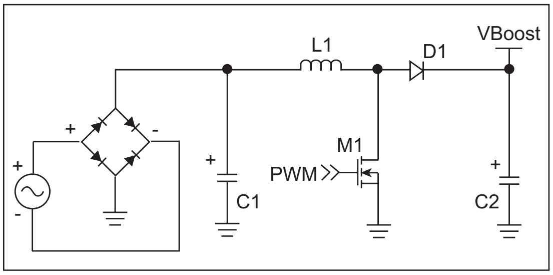
TIP #7 Using a PIC® Microcontroller for Power Factor Correction

In AC power systems, the term Power Factor (PF) is used to describe the fraction of power actually used by a load compared to the total apparent power supplied.

Power Factor Correction (PFC) is used to increase the efficiency of power delivery by maximizing the PF.

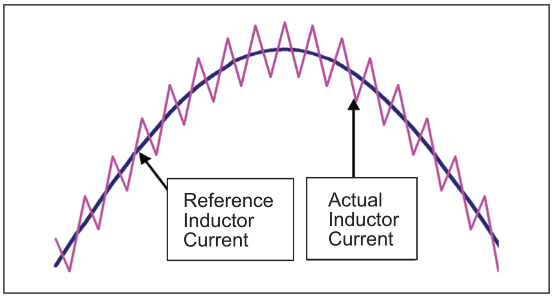
The basis for most Active PFC circuits is a boost circuit, shown in Figure 7-1.

Figure 7-1: Typical Power Factor Correction Boost Supply



The AC voltage is rectified and boosted to voltages as high as 400 VDC. The unique feature of the PFC circuit is that the inductor current is regulated to maintain a certain PF. A sine wave reference current is generated that is in phase with the line voltage. The magnitude of the sine wave is inversely proportional to the voltage at VBoost. Once the sine wave reference is established, the inductor current is regulated to follow it, as shown in Figure 7-2.

Figure 7-2: Desired and Actual Inductor Currents



A PIC MCU has several features that allow it to perform power factor correction.

- The PIC MCUs CCP module can be used to generate a PWM signal that, once filtered, can be used to generate the sine wave reference signal.
- The PIC Analog-to-Digital (A/D) converter can be used to sense VBoost and the reference sine wave can be adjusted in software.
- The interrupt-on-change feature of the PIC MCU input pins can be used to allow the PIC MCU to synchronize the sine wave reference to the line voltage by detecting the zero crossings.
- The on-chip comparators can be used for driving the boost MOSFET(s) using the PWM sine wave reference as one input and the actual inductor current as another.

TIP #8 Transformerless Power Supplies

When using a microcontroller in a line-powered application, such as the IR remote control actuated AC switch described in Tip #9, the cost of building a transformer-based AC/DC converter can be significant. However, there are transformerless alternatives which are described below.

Capacitive Transformerless Power Supply

Figure 8-1: Capacitive Power Supply

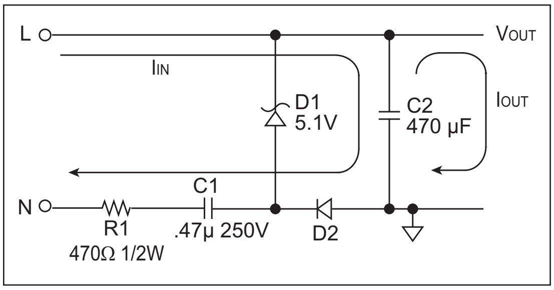


Figure 8-1 shows the basics for a capacitive power supply. The Zener diode is reverse-biased to create the desired voltage. The current drawn by the Zener is limited by R1 and the impedance of C1.

Advantages:

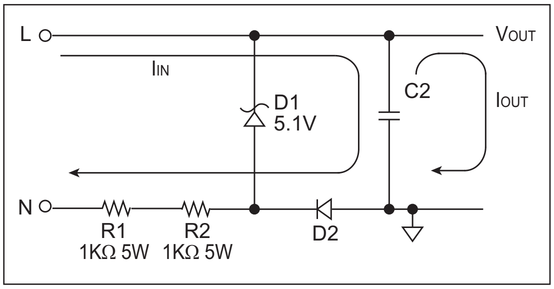
- Significantly smaller than a transformer-based power supply
- Lower cost than a transformer-based or switcher-based power supply
- Power supply is more efficient than a resistive transformerless power supply

Disadvantages:

- Not isolated from the AC line voltage which introduces safety issues
- Higher cost than a resistive power supply because X2 rated capacitors are required

Resistive Power Supply

Figure 8-2: Resistive Power Supply



The resistive power supply works in a similar manner to the capacitive power supply by using a reversed-biased Zener diode to produce the desired voltage. However, R1 is much larger and is the only current limiting element.

Advantages:

- Significantly smaller than a transformer-based power supply
- Lower cost than a transformer-based power supply
- Lower cost than a capacitive power supply

Disadvantages:

- Not isolated from the AC line voltage which introduces safety issues
- Power supply is less energy efficient than a capacitive power supply
- More energy is dissipated as heat in R1

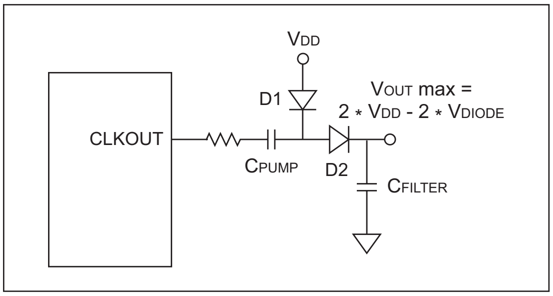
More information on either of these solutions, including equations used for calculating circuit parameters, can be found in AN954, “*Transformerless Power Supplies: Resistive and Capacitive*” (DS00954) or in TB008, “*Transformerless Power Supply*” (DS91008).

TIP #10 Driving High Side FETs

In applications where high side N channel FETs are to be driven, there are several means for generating an elevated driving voltage. One very simple method is to use a voltage doubling charge pump as shown in Figure 10-1.

Method 1

Figure 10-1: Typical Change Pump

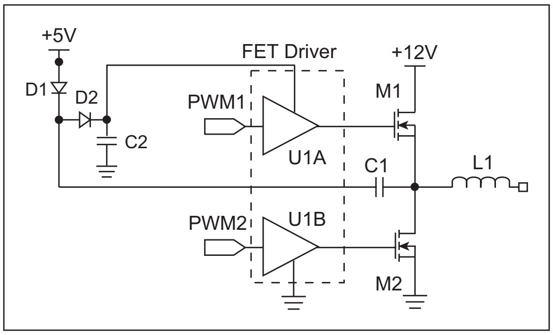


The PIC MCUs CLKOUT pin toggles at 1/4 of the oscillator frequency. When CLKOUT is low, D1 is forward biased and conducts current, thereby charging C_{PUMP}. After CLKOUT is high, D2 is forward biased, moving the charge to C_{FILTER}. The result is a voltage equal to twice the V_{DD} minus two diode drops. This can be used with a PWM or any other I/O pin that toggles.

In Figure 10-2, a standard FET driver is used to drive both the high and low side FETs by using the diode and capacitor arrangement.

Method 2

Figure 10-2: Schematic



The +5V is used for powering the microcontroller. Using this arrangement, the FET driver would have approximately 12 + (5 - V_{DIODE}) - V_{DIODE} volts as a supply and is able to drive both the high and low side FETs.

The circuit above works by charging C1 through D1 to (5V - V_{DIODE}) while M2 is on, effectively connecting C1 to ground. When M2 turns off and M1 turns on, one side of C1 is now at 12V and the other side is at 12V + (5V - V_{DIODE}). The D2 turns on and the voltage supplied to the FET driver is 12V + (5V - V_{DIODE}) - V_{DIODE}.

TIP #20 Compensating Sensors Digitally

Many sensors and references tend to drift with temperature. For example, the MCP9700 specification states that its typical is ±0.5°C and its max error is ±4°C.

Figure 20-1: MCP9700 Accuracy

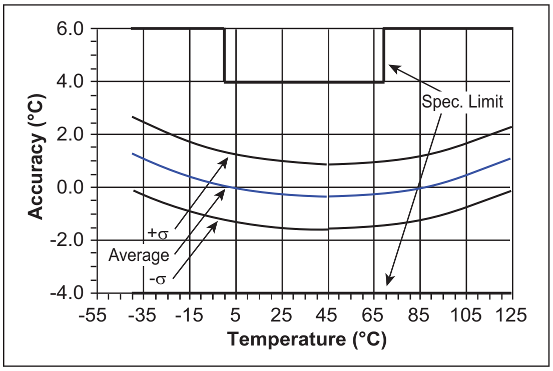


Figure 20-1 shows the accuracy of a 100 sample lot of MCP9700 temperature sensors. Despite the fact that the sensor’s error is nonlinear, a PIC microcontroller (MCU) can be used to compensate the sensor’s reading.

Polynomials can be fitted to the average error of the sensor. Each time a temperature reading is received, the PIC MCU can use the measured result and the error compensation polynomials to determine what the true temperature is.

Figure 20-2: MCP9700 Average Accuracy After Compensation

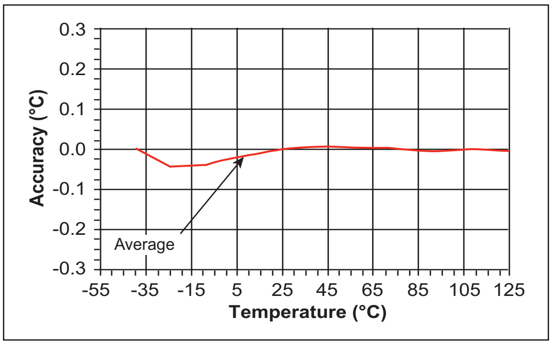


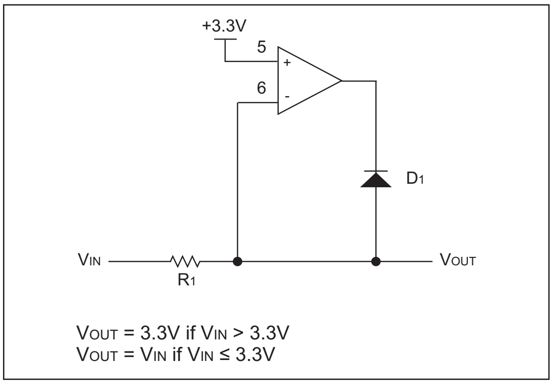
Figure 20-2 shows the average accuracy for the 100 sample lot of MCP9700 temperature sensors after compensation. The average error has been decreased over the full temperature range.

It is also possible to compensate for error from voltage references using this method.

For more information on compensating a temperature sensor digitally, refer to AN1001, “*IC Temperature Sensor Accuracy Compensation with a PIC Microcontroller*” (DS01001).

If a more precise overvoltage clamp is required that does not rely upon the supply, then an op amp can be employed to create a precision diode. In Figure 17-3, such a circuit is shown. The op amp compensates for the forward drop in the diode and causes the voltage to be clamped at exactly the voltage supplied on the non-inverting input to the op amp. The op amp can be powered from 3.3V if it is rail-to-rail.

Figure 17-3: Precision Diode Clamp



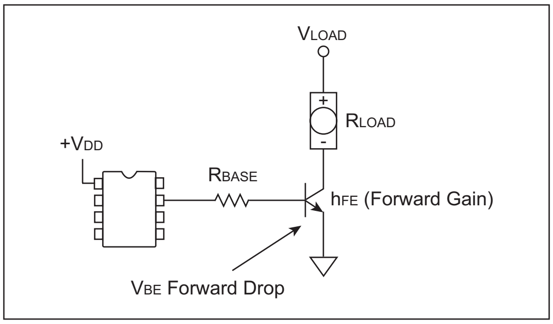
Because the clamping is performed by the op amp, there is no affect on the power supply. The impedance presented to the low voltage circuit is not improved by the op amp, it remains R1 in addition to the source circuit impedance.

TIP #18 Driving Bipolar Transistors

When driving bipolar transistors, the amount of base current “drive” and forward current gain

(B/h_{FE}) will determine how much current the transistor can sink. When driven by a microcontroller I/O port, the base drive current is calculated using the port voltage and the port current limit (typically 20 mA). When using 3.3V technology, smaller value base current limiting resistors should be used to ensure sufficient base drive to saturate the transistor.

Figure 18-1: Driving Bipolar Transistors Using Microcontroller I/O Port



The value of R_{BASE} will depend on the microcontroller supply voltage. Equation 18-1 describes how to calculate R_{BASE}.