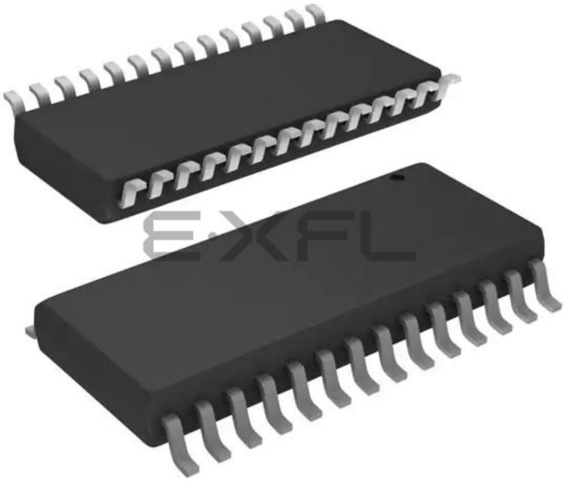


What is "Embedded - Microcontrollers"?

"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	22
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	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c63a-04-so

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

GENERAL LOW POWER TIPS ‘N TRICKS

DYNAMIC OPERATION TIPS ‘N TRICKS

STATIC POWER REDUCTION TIPS ‘N TRICKS

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.

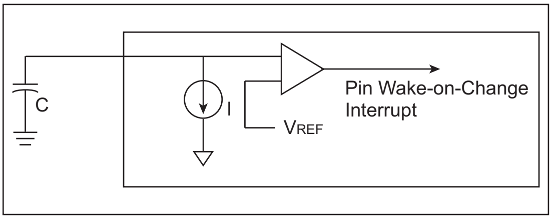
GENERAL LOW POWER TIPS ‘N TRICKS

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

TIP #22 Ultra Low-Power Wake-Up Peripheral

Newer devices have a modification to PORTA that creates an Ultra Low-Power Wake-Up (ULPWU) peripheral. A small current sink and a comparator have been added that allows an external capacitor to be used as a wake-up timer. This feature provides a low-power periodic wake-up source which is dependent on the discharge time of the external RC circuit.

Figure 22-1: Ultra Low-Power Wake-Up Peripheral



If the accuracy of the Watchdog Timer is not required, this peripheral can save a lot of current.

Visit the low power design center at: www.microchip.com/lowpower for additional design resources.

TIP #5 Measuring RPM Using an Encoder

Revolutions Per Minute (RPM), or how fast something turns, can be sensed in a variety of ways. Two of the most common sensors used to determine RPM are optical encoders and Hall effect sensors. Optical encoders detect the presence of light shining through a slotted wheel mounted to a turning shaft (see Figure 5-1.) As the shaft turns, the slots in the wheel pass by the eye of the optical encoder. Typically, an infrared source on the other side of the wheel emits light that is seen by the optical encoder through slots in the wheel. Hall effect sensors work by sensing the position of the magnets in an electric motor, or by sensing a permanent magnet mounted to a rotating object (see Figure 5-2). These sensors output one or more pulses per revolution (depending on the sensor).

Figure 5-1: Optical Encoder

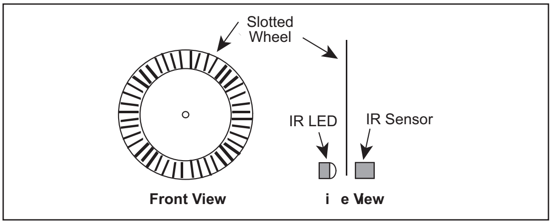
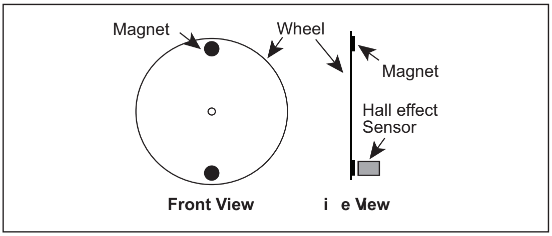


Figure 5-2: Hall Effect Sensor



In Figure 5-3 and Figure 5-4, the waveform is high when light is passing through a slot in the encoder wheel and shining on the optical sensor. In the case of a Hall effect sensor, the high corresponds to the time that the magnet is in front of the sensor. These figures show the difference in the waveforms for varying RPMs. Notice that as RPM increases, the period (T) and pulse width (W) becomes smaller. Both period and pulse width are proportional to RPM. However, since the period is the greater of the two intervals, it is good practice to measure the period so that the RPM reading from the sensor will have the best resolution. See Tip #1 for measuring period. The technique for measuring period with averaging described in Tip #2 is useful for measuring high RPMs.

Figure 5-3: Low RPM

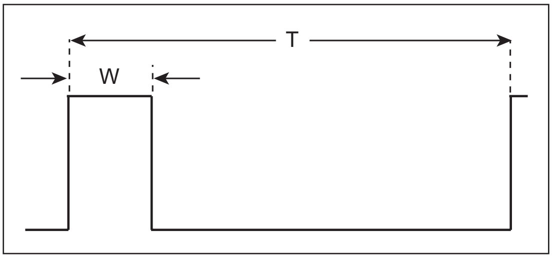
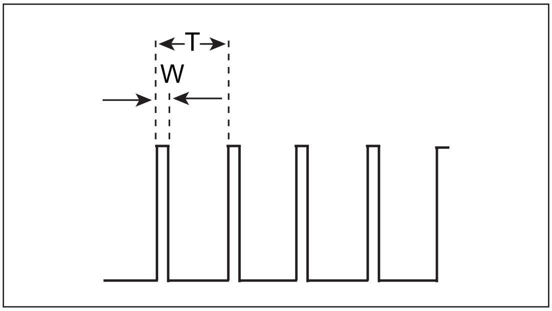


Figure 5-4: High RPM



TIP #8 Modulation Formats

The CCP module, configured in Compare mode, can be used to generate a variety of modulation formats. The following figures show four commonly used modulation formats:

Figure 8-1: Pulse-width Modulation

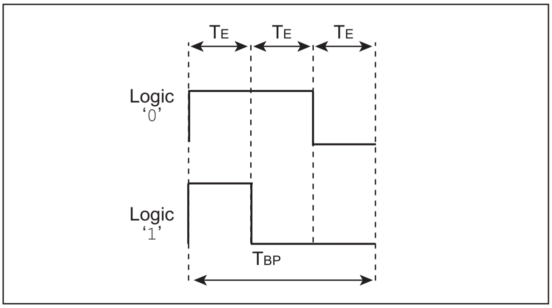


Figure 8-2: Manchester

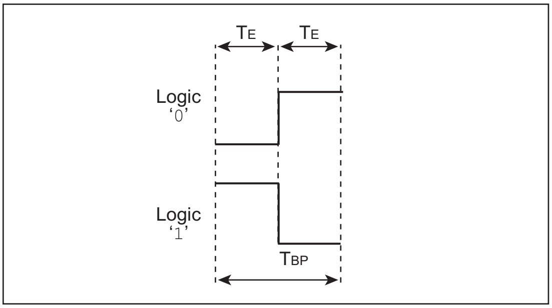


Figure 8-3: Pulse Position Modulation

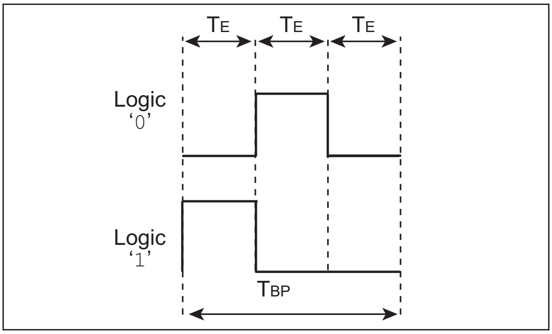
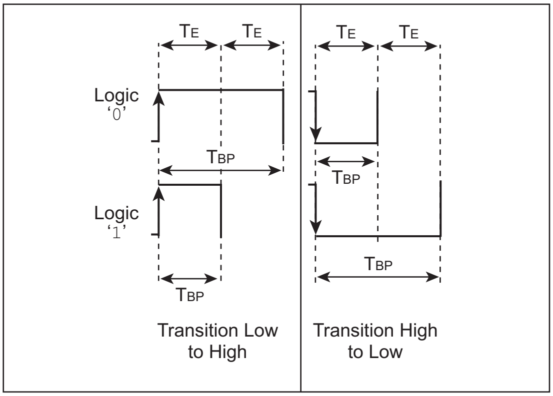


Figure 8-4: Variable Pulse-width Modulation



The figures show what a logic ‘0’ or a logic ‘1’ looks like for each modulation format. A transmission typically resembles an asynchronous serial transmission consisting of a Start bit, followed by 8 data bits, and a Stop bit.

T_E is the basic timing element in each modulation format and will vary based on the desired baud rate.

Trigger Special Event mode can be used to generate T_E, (the basic timing element). When the CCPx interrupt is generated, code in the ISR routine would implement the desired modulation format (additional modulation formats are also possible).

TIP #9 Generating the Time Tick for a RTOS

Real Time Operating Systems (RTOS) require a periodic interrupt to operate. This periodic interrupt, or "tick rate", is the basis for the scheduling system that RTOS's employ. For instance, if a 2 ms tick is used, the RTOS will schedule its tasks to be executed at multiples of the 2 ms. A RTOS also assigns a priority to each task, ensuring that the most critical tasks are executed first. Table 9-1 shows an example list of tasks, the priority of each task and the time interval that the tasks need to be executed.

Table 9-1: Tasks

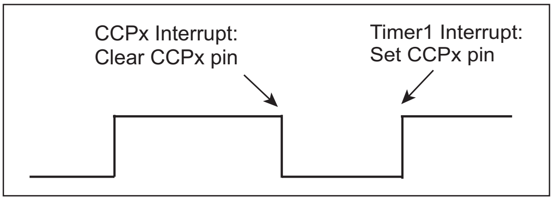
Task	Interval	Priority
Read ADC Input 1	20 ms	2
Read ADC Input 2	60 ms	1
Update LCD	24 ms	2
Update LED Array	36 ms	3
Read Switch	10 ms	1
Dump Data to Serial Port	240 ms	1

The techniques described in Tip #7 can be used to generate the 2 ms periodic interrupt using the CCP module configured in Compare mode.

Note: For more information on RTOSs and their use, see Application Note AN777 "*Multitasking on the PIC16F877 with the Salvo™ RTOS*".

TIP #10 16-Bit Resolution PWM

Figure 10-1: 16-Bit Resolution PWM



1. Configure CCPx to clear output (CCPx pin) on match in Compare mode (CCPxCON <CCPSM3:CCPxM0>).
2. Enable the Timer1 interrupt.
3. Set the period of the waveform via Timer1 prescaler (T1CON <5:4>).
4. Set the duty cycle of the waveform using CCPRxL and CCPRxH.
5. Set CCPx pin when servicing the Timer1 overflow interrupt(1).

Note 1: One hundred percent duty cycle is not achievable with this implementation due to the interrupt latency in servicing Timer1. The period is not affected because the interrupt latency will be the same from period to period as long as the Timer1 interrupt is serviced first in the ISR.

Timer1 has four configurable prescaler values. These are 1:1, 1:2, 1:4 and 1:8. The frequency possibilities of the PWM described above are determined by Equation 10-1.

Equation 10-1

$$F_{PWM} = F_{OSC}/(65536*4*prescaler)$$

For a microcontroller running on a 20 MHz oscillator (Fosc) this equates to frequencies of 76.3 Hz, 38.1 Hz, 19.1 Hz and 9.5 Hz for increasing prescaler values.

PWM TIPS ‘N TRICKS

The ECCP and CCP modules produce a 10-bit resolution Pulse-Width Modulated (PWM) waveform on the CCPx pin. The ECCP module is capable of transmitting a PWM signal on one of four pins, designated P1A through P1D. The PWM modes available on the ECCP module are:

- Single output (P1A only)
- Half-bridge output (P1A and P1B only)
- Full-bridge output forward
- Full-bridge output reverse

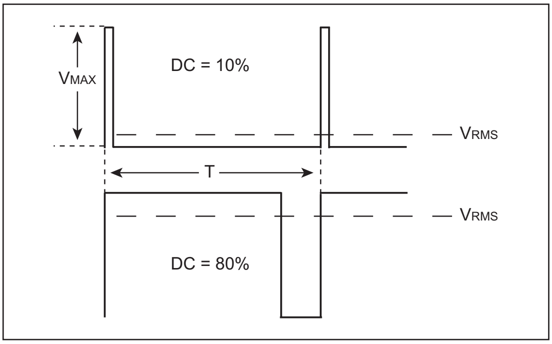
One of the following configurations must be chosen when using the ECCP module in PWM Full-bridge mode:

- P1A, P1C active-high; P1B, P1D active-high
- P1A, P1C active-high; P1B, P1D active-low
- P1A, P1C active-low; P1B, P1D active-high
- P1A, P1C active-low; P1B, P1D active-low

“Why Would I Use PWM Mode?”

As the next set of Tips ‘n Tricks demonstrate, Pulse-Width Modulation (PWM) can be used to accomplish a variety of tasks from dimming LEDs to controlling the speed of a brushed DC electric motor. All these applications are based on one basic principle of PWM signals – as the duty cycle of a PWM signal increases, the average voltage and power provided by the PWM increases. Not only does it increase with duty cycle, but it increases linearly. The following figure illustrates this point more clearly. Notice that the RMS and maximum voltage are functions of the duty cycle (DC) in the following Figure 12-3.

Figure 12-3: Duty Cycle Relation to V_{RMS}



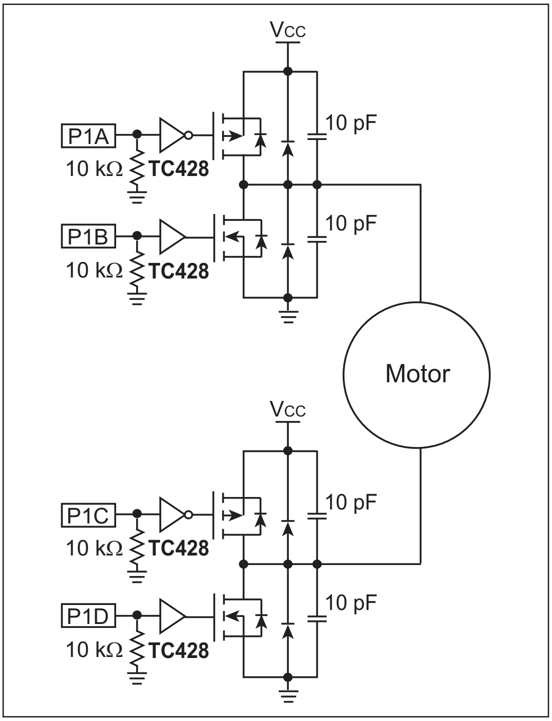
Equation 12-1 shows the relation between V_{RMS} and V_{MAX}.

Equation 12-1: Relation Between V_{RMS} and V_{MAX}

$$V_{RMS} = DC x V_{MAX}$$

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

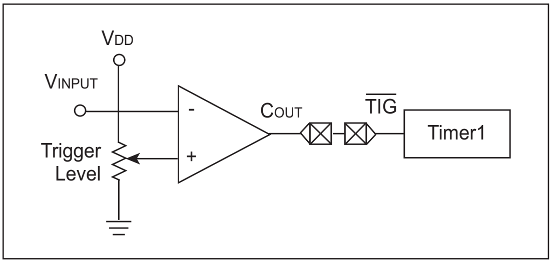
TIP #4 Pulse Width Measurement

To measure the high or low pulse width of an incoming analog signal, the comparator can be combined with Timer1 and the Timer1 Gate input option (see Figure 4-1). Timer1 Gate acts as a count enable for Timer1. If the input is low, Timer1 will count. If the $\overline{\text{T1G}}$ input is high, Timer1 does not count. Combining $\overline{\text{T1G}}$ with the comparator allows the designer to measure the time between a high-to-low output change and a low-to-high output change.

To make a measurement between a low-to-high and a high-to-low transition, the only change required is to set the CINV bit in the comparator CMCON register which inverts the comparator output.

Because the output of the comparator can change asynchronously with the Timer1 clock, only comparators with the ability to synchronize their output with the Timer1 clock should be used and their C2SYNC bits should be set.

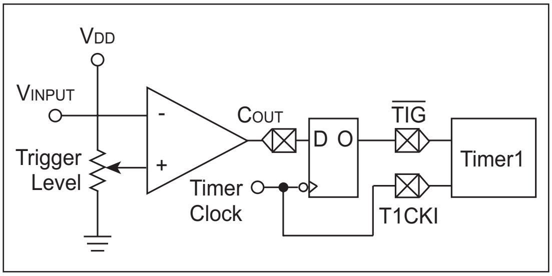
Figure 4-1: Comparator with Timer1 and $\overline{\text{T1G}}$



If the on-chip comparator does not have the ability to synchronize its output to the Timer1 clock, the output can be synchronized externally using a discrete D flip-flop (see Figure 4-2).

> **Note:** The flip-flop must be falling edge triggered to prevent a race condition.

Figure 4-2: Externally Synchronized Comparator

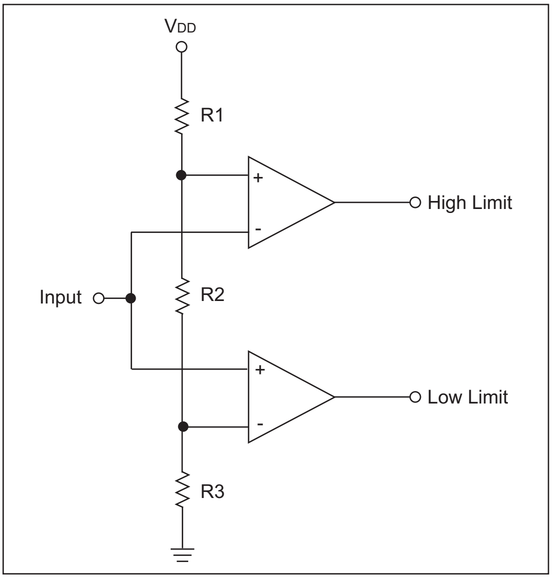


TIP #5 Window Comparison

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.

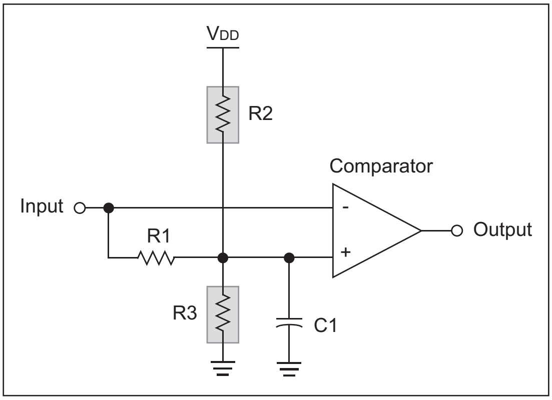
TIP #6 Data Slicer

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



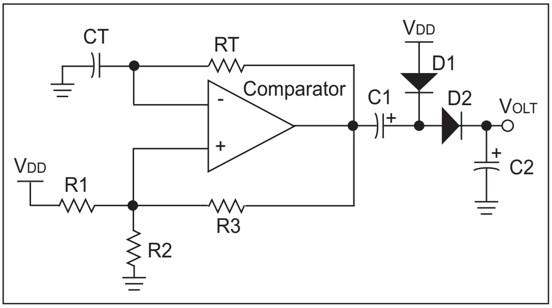
Example:

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

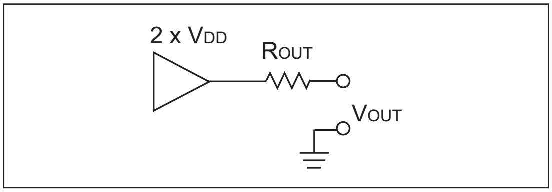
This tip takes the multi-vibrator described in Tip #8 and builds a capacitive voltage doubler around it (see Figure 10-1). The circuit works by alternately charging capacitor C1 through diode D1, and then charge balancing the energy in C1 with C2 through diode D2. At the start of the cycle, the output 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 #20 Logic: Set/Reset Flip Flop

This tip shows the use of the comparator to implement a Set/Reset Flip Flop.

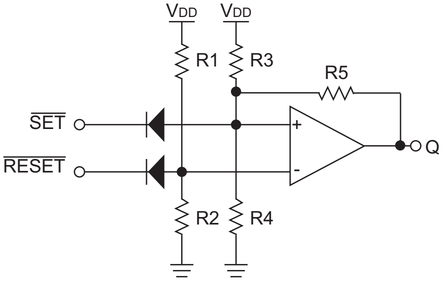
The inverting and non-inverting inputs are biases at VDD/2 by resistors R1 through R4. The non-inverting input also receives positive feedback from the output through R5. The common bias voltages and the positive feedback configure the comparator as a bistable latch. If the output Q is high, the non-inverting input is also pulled high, which reinforces the high output. If Q is low, the non-inverting input is also pulled low, which reinforces the low output. To change state, the appropriate input must be pulled low to overcome the positive feedback. The diodes prevent a positive state on either input from pulling the bias of either input above VDD/2.

> **Note:** Typical propagation delay for the circuit is 250-350 ns using the typical on-chip comparator peripheral of a microcontroller. Delay measurements were made with 10k resistance values.

While the circuit is fairly simple, there are a few requirements for correct operation:

1. The inputs Set and Reset must be driven near ground for the circuit to operate properly.
2. The combination of R1/R2 and R3/R4 will draw current constantly, so they must be kept large to minimize current draw.
3. R1 through R4 must be equal for a VDD/2 trip level.
4. R5 must be greater or equal to R3.
5. R1 through R4 will react with the input capacitance of the comparator, so larger values will limit the minimum input pulse width.

Figure 20-1: Set/Reset Flip Flop



Example:

- Diodes = 1N4148
- R1 = R2 = R3 = R4 = 10k
- R5 = 10k

Figure 3-1: 4 and 5 Wire Stepper Motors

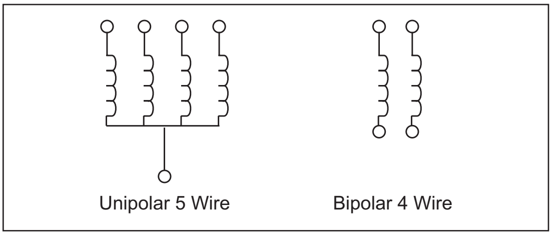


Figure 3-2: 6 and 8 Wire Stepper Motors

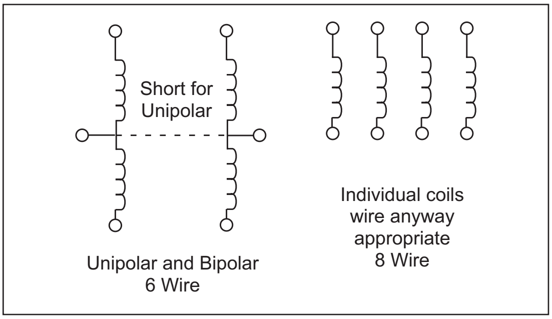
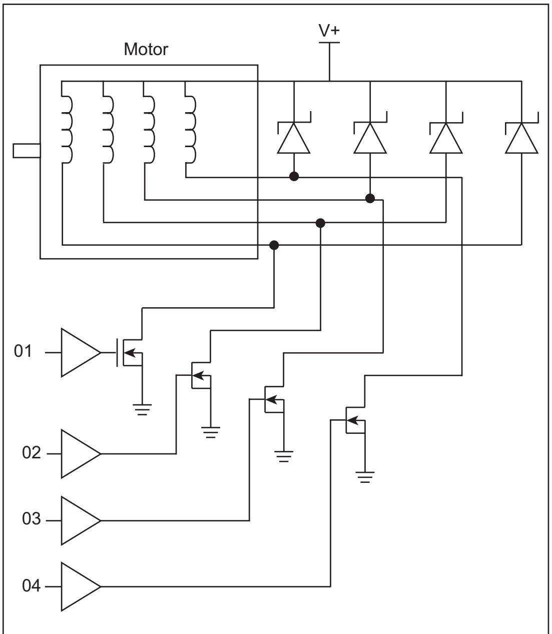
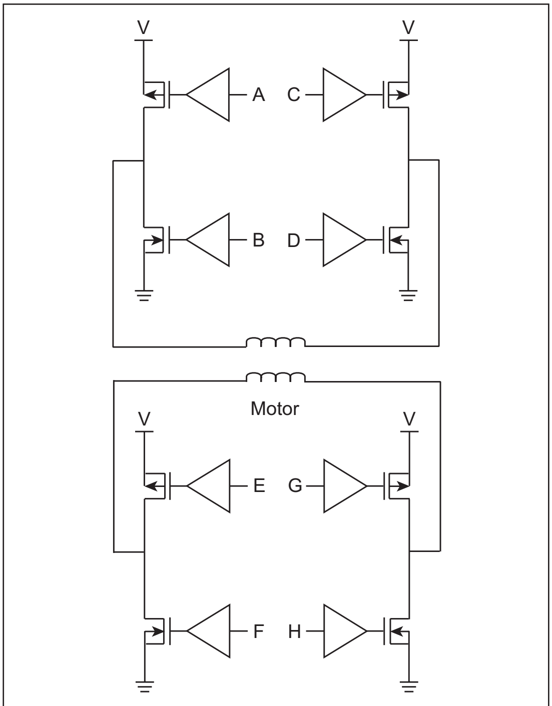


Figure 3-3: Unipolar Motor (4 Low Side Switches)



01-04 are outputs from a PIC® MCU or dsPIC® DSC.

Figure 3-4: Bipolar Motor (4 Half-Bridges)



A-H are digital outputs from a PIC® MCU or dsPIC® DSC.

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

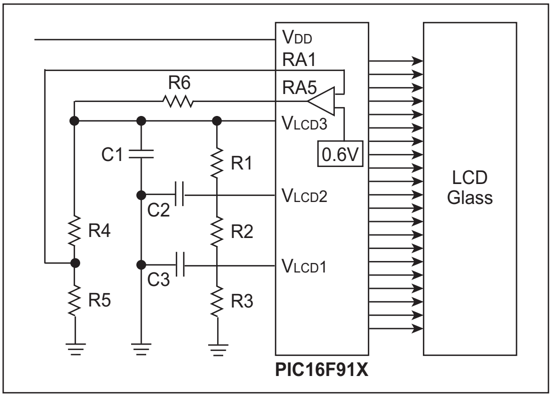
Multiplex Commons	Maximum Number of Segments/Pixels					Bias
	PIC16F913/ 916	PIC16F914/ 917	PIC16F946	PIC18F6X90 (PIC18F6XJ90)	PIC18F8X90 (PIC18F8XJ90)	
Static (COM0)	15	24	42	32/ (33)	48	Static
1/2 (COM1: COM0)	30	48	84	64/ (66)	96	1/2 or 1/3
1/3 (COM2: COM0)	45	72	126	96/ (99)	144	1/2 or 1/3
1/4 (COM3: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

TIP #4: Contrast Control with a Buck Regulator

Contrast control in any of the LCD PIC MCUs is accomplished by controlling the voltages applied to the V_{LCD} voltage inputs. The simplest contrast voltage generator is to place a resistor divider across the three pins. This circuit is shown in the data sheet. The resistor ladder method is good for many applications, but the resistor ladder does not work in an application where the contrast must remain constant over a range of V_{DD}s. The solution is to use a voltage regulator. The voltage regulator can be external to the device, or it can be built using a comparator internal to the LCD PIC microcontroller.

Figure 4-1: Voltage Generator with Resistor Divider



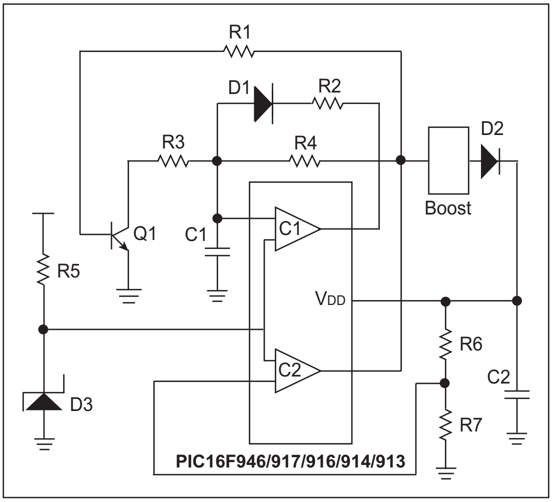
The PIC16F946/917/916/914/913 devices have a special Comparator mode that provides a fixed 0.6V reference. The circuit shown in Figure 4-1 makes use of this reference to provide a regulated contrast voltage. In this circuit, R1, R2 and R3 provide the contrast control voltages. The voltage on $V_{LCD}3$ is compared to the internal voltage reference by dividing the voltage at $V_{LCD}3$ at R4 and R5 and applying the reduced voltage to the internal comparator. When the voltage at $V_{LCD}3$ is close to the desired voltage, the output of the comparator will begin to oscillate. The oscillations are filtered into a DC voltage by R6 and C1. C2 and C3 are simply small bypass capacitors to ensure that the voltages at $V_{LCD}1$ and $V_{LCD}2$ are steady.

TIP #5: Contrast Control Using a Boost Regulator

In LCD Tip #4, a buck converter was created using a comparator. This circuit works great when V_{DD} is greater than the LCD voltage. The PIC microcontroller can operate all the way down to 2.0V, whereas most low-voltage LCD glass only operates down to 3V. In a battery application, it is important to stay operational as long as possible. Therefore, a boost converter is required to boost 2.0V up to 3.0V for the LCD.

The figure below shows one circuit for doing this.

Figure 5-1: Boost Converter

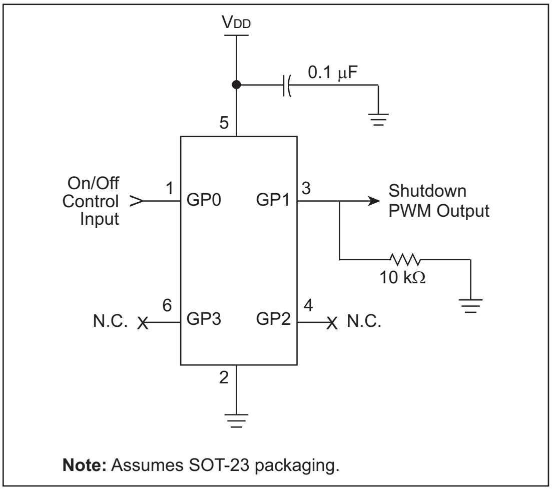


In this circuit, both comparators are used. The voltage setpoint is determined by the value of Zenier diode D3 and the voltage at R6:R7. The rest of the circuit creates a simple multivibrator to stimulate a boost circuit. The boost circuit can be inductor or capacitor-based. When the output voltage is too low, the multivibrator oscillates and causes charge to build up in C2. As the voltage at C2 increases, the multivibrator will begin to operate sporadically to maintain the desired voltage at C2.

TIP #1 Soft-Start Using a PIC10F200

Almost all power supply controllers are equipped with shutdown inputs that can be used to disable the MOSFET driver outputs. Using Pulse-Width Modulation (PWM), the amount of time the power supply is allowed to operate can be slowly incremented to allow the output voltage to slowly rise from 0% to 100%.

Figure 1-1: Soft-Start Circuit Schematic



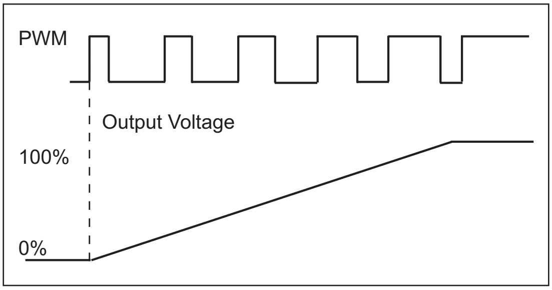
Note: Assumes SOT-23 packaging.

This technique is called soft-start and is used to prevent the large inrush currents that are associated with the start-up of a switching power supply.

GP0 on the PIC MCU is used to enable or disable the soft-start. Once enabled, the on-time of the PWM signal driving the shutdown output will increase each cycle until the power supply is fully on.

During the PIC MCU Power-on Reset, the PWM output (GP1) is initially in a high-impedance state. A pull-down resistor on the PWM output ensures the power supply will not unexpectedly begin operating.

Figure 1-2: Timing Diagram



It is important to note that this type of soft-start controller can only be used for switching regulators that respond very quickly to changes on their shutdown pins (such as those that do cycle-by-cycle limiting). Some linear regulators have active-low shutdown inputs, however, these regulators do not respond fast enough to changes on their shutdown pins in order to perform soft-start.

Example software is provided for the PIC10F200 which was taken from TB081. Please refer to TB081, “*Soft-Start Controller For Switching Power Supplies*” (DS91081) for more information.

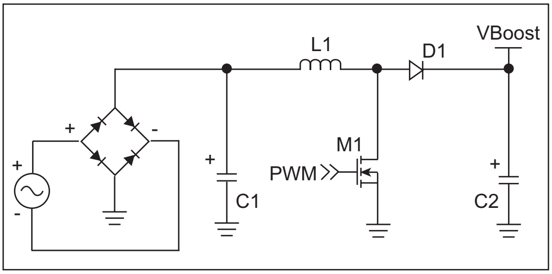
TIP #7 Using a PIC® Microcontroller for Power Factor Correction

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.

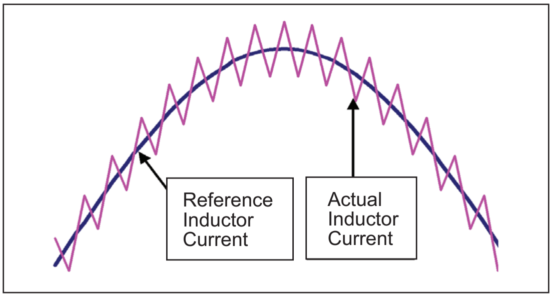
The basis for most Active PFC circuits is a boost circuit, shown in Figure 7-1.

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

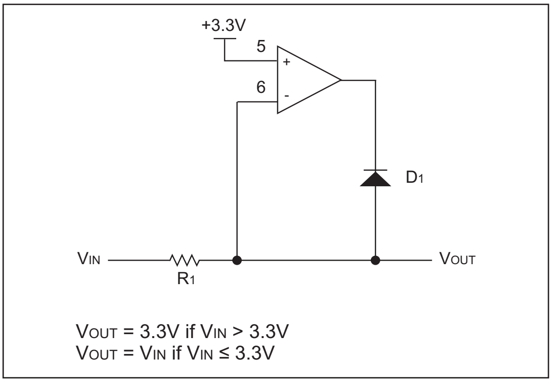


A PIC MCU has several features that allow it to perform power factor correction.

- 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.

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

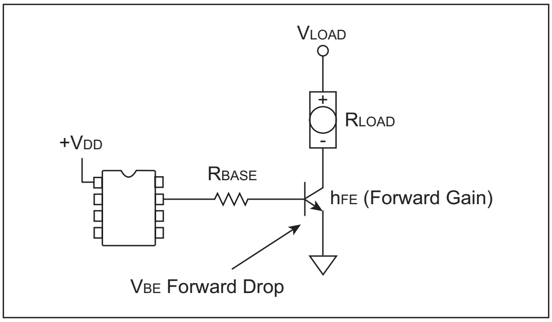


Because the clamping is performed by the op amp, there is no affect on the power supply. The impedance presented to the low voltage circuit is not improved by the op amp, it remains R1 in addition to the source circuit impedance.

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}.

Table 18-1: Bipolar Transistor DC Specifications

Characteristic	Sym	Min	Max	Unit	Test Condition
OFF CHARACTERISTICS					
Collector-Base Breakdown Voltage	$V_{(BR)CBO}$	60	–	V	I_C = 50 µA, I_E = 0
Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	50	–	V	I_C = 1.0 mA, I_B = 0
Emitter-Base Breakdown Voltage	$V_{(BR)EBO}$	7.0	–	V	I_E = 50 µA, I_C = 0
Collector Cutoff Current	I_{CBO}	–	100	nA	V_{CB} = 60V
Emitter Cutoff Current	I_{EBO}	–	100	nA	V_{EB} = 7.0V
ON CHARACTERISTICS					
DC Current Gain	h_{FE}	120 180 270	270 390 560	–	V_{CE} = 6.0V, I_C = 1.0 mA
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	–	0.4	V	I_C = 50 mA, I_B = 5.0 mA

When using bipolar transistors as switches to turn on and off loads controlled by the microcontroller I/O port pin, use the minimum h_{FE} specification and margin to ensure complete device saturation.

Equation 18-1: Calculating the Base Resistor Value

$$R_{BASE} = \frac{(V_{DD} - V_{BE}) \times h_{FE} \times R_{LOAD}}{V_{LOAD}}$$

3V Technology Example

V_{DD} = +3V, V_{LOAD} = +40V, R_{LOAD} = 400Ω,
h_{FE} min. = 180, V_{BE} = 0.7V

R_{BASE} = 4.14 kΩ, I/O port current = 556 µA

5V Technology Example

V_{DD} = +5V, V_{LOAD} = +40V, R_{LOAD} = 400Ω,
h_{FE} min. = 180, V_{BE} = 0.7V

R_{BASE} = 7.74 kΩ, I/O port current = 556 µA

For both examples, it is good practice to increase base current for margin. Driving the base with 1 mA to 2 mA would ensure saturation at the expense of increasing the input power consumption.