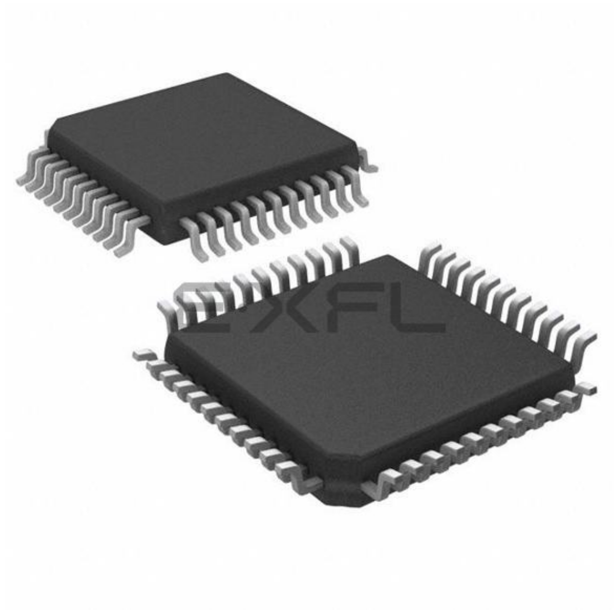
Microchip Technology - PIC16C65B-04/PQ Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

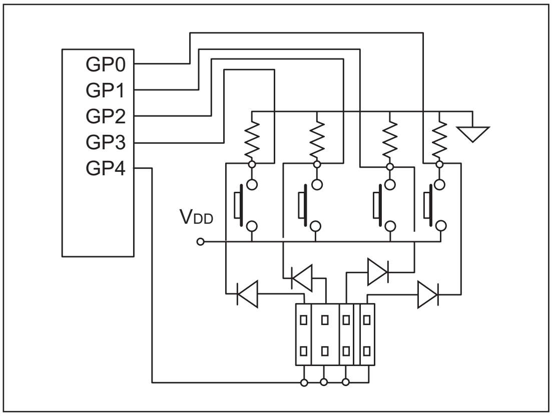
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c65b-04-pq

TIP #9 Decode Keys and ID Settings

Buttons and jumpers can share I/O’s by using another I/O to select which one is read. Both buttons and jumpers are tied to a shared pull-down resistor. Therefore, they will read as ‘0’ unless a button is pressed or a jumper is connected. Each input (GP3/2/1/0) shares a jumper and a button. To read the jumper settings, set GP4 to output high and each connected jumper will read as ‘1’ on its assigned I/O or ‘0’ if it’s not connected. With GP4 output low, a pressed button will be read as ‘1’ on its assigned I/O and ‘0’ otherwise.

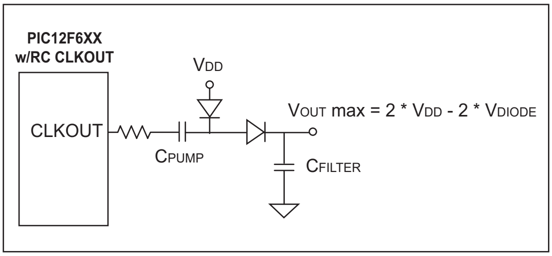
Figure 9-1



- When GP4 = 1 and no keys are pressed, read ID setting
- When GP4 = 0, read the switch buttons

TIP #10 Generating High Voltages

Figure 10-1



Voltages greater than V_{DD} can be generated using a toggling I/O. PIC MCUs CLKOUT/OSC2 pin toggles at one quarter the frequency of OSC1 when in external RC oscillator mode. When OSC2 is low, the V_{DD} diode is forward biased and conducts current, thereby charging C_{PUMP}. After OSC2 is high, the other diode is forward biased, moving the charge to C_{FILTER}. The result is a charge equal to twice the V_{DD} minus two diode drops. This can be used with a PWM, a toggling I/O or other toggling pin.

Tip #13.2 Reading a Sensor With Higher Accuracy – Charge Balancing Method

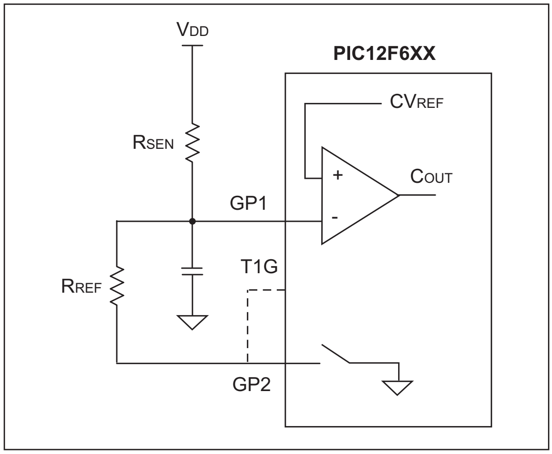
1. Sensor charges a capacitor
2. Reference resistor discharges the capacitor
3. Modulate reference resistor to maintain constant average charge in the capacitor
4. Use comparator to determine modulation

To improve resolution beyond 10 or 12 bits, a technique called “Charge Balancing” can be used. The basic concept is for the MCU to maintain a constant voltage on a capacitor by either allowing the charge to build through a sensor or discharge through a reference resistor. A timer is used to sample the capacitor voltage on regular intervals until a predetermined number of samples are counted. By counting the number of times the capacitor voltage is over an arbitrary threshold, the sensor voltage is determined.

The comparator and comparator voltage reference (CV_{REF}) on the PIC12F629/675 are ideal for this application.

1. GP1 average voltage = CV_{REF}
2. Time base as sampling rate
3. At the end of each time base period:
   - If GP1 > CV_{REF}, then GP2 Output Low
   - If GP1 < CV_{REF}, then GP2 Input mode
4. Accumulate the GP2 lows over many samples
5. Number of samples determines resolution
6. Number of GP2 lows determine effective duty cycle of R_{REF}

Figure 13-3



TIP #18 Swap File Register with W

Example 18-1

```
SWAPWF       MACRO  REG
             XORWF  REG,F
             XORWF  REG,W
             XORWF  REG,F
             ENDM
```

The following macro swaps the contents of W and REG without using a second register.

Needs: 0 TEMP registers
3 Instructions
3 TCY

An efficient way of swapping the contents of a register with the working register is to use three XORWF instructions. It requires no temporary registers and three instructions. Here's an example:

W	REG	Instruction
10101100	01011100	XORWF REG,F
10101100	11110000	XORWF REG,W
01011100	11110000	XORWF REG,F
01011100	10101100	Result

TIP #19 Bit Shifting Using Carry Bit

Rotate a byte through carry without using RAM variable for loop count:

- Easily adapted to serial interface transmit routines.
- Carry bit is cleared (except last cycle) and the cycle repeats until the zero bit sets indicating the end.

Example 19-1

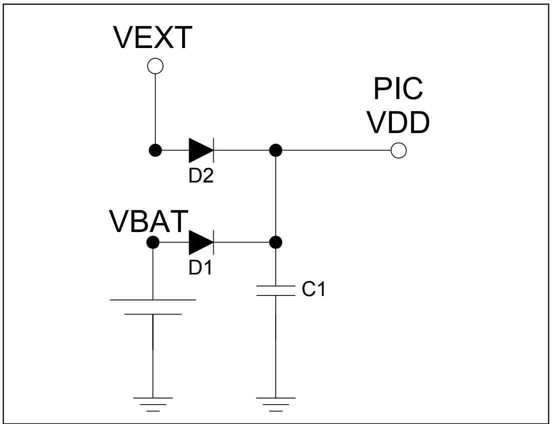
```
        LIST P=PIC12f629
        INCLUDE P12f629.INC
        buffer      equ   0x20

bsf     STATUS,C      ;Set 'end of loop' flag
rlf     buffer,f      ;Place first bit into C
bcf     GPIO,Dout     ;precondition output
btfsc   STATUS,C      ;Check data 0 or 1 ?
bsf     GPIO,Dout
bcf     STATUS,C      ;Clear data in C
rlf     buffer,f      ;Place next bit into C
movf    buffer,f      ;Force Z bit
btfss   STATUS,Z      ;Exit?
goto    Send_Loop
```


TIP #7 Battery Backup for PIC MCUs

For an application that can operate from either an external supply or a battery backup, it is necessary to be able to switch from one to the other without user intervention. This can be accomplished with battery backup ICs, but it is also possible to implement with a simple diode OR circuit, shown in Figure 7-1. Diode D1 prevents current from flowing into the battery from VEXT when the external power is supplied. D2 prevents current from flowing into any external components from the battery if VEXT is removed. As long as the external source is present and higher voltage than the battery, no current from the battery will be used. When VEXT is removed and the voltage drops below VBAT, the battery will start powering the MCU. Low forward voltage Schottky diodes can be used in order to minimize the voltage dropout from the diodes. Additionally, inputs can be referenced to VEXT and VBAT in order to monitor the voltage levels of the battery and the external supply. This allows the micro to enter lower power modes when the supply is removed or the battery is running low. In order to avoid glitches on V_{DD} caused by the diode turn-on delay when switching supplies, ensure enough decoupling capacitance is used on V_{DD} (C1).

Figure 7-1:



Dynamic Operation Tips n’ Tricks

The following tips and tricks apply to methods of improving the dynamic operating current consumption of an application. This allows an application to get processing done quicker which enables it to sleep more and will help reduce the current consumed while processing.

TIP #8 Enhanced PIC16 Mid-Range Core

The Enhanced PIC16 mid-range core has a few features to assist in low power. New instructions allow many applications to execute in less time. This allows the application to spend more time asleep and less time processing and can provide considerable power savings. It is important not to overlook these new instructions when designing with devices that contain the new core. The Timer1 oscillator and WDT have also been improved, now meeting nanoWatt XLP requirements and drawing much less current than in previous devices.

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

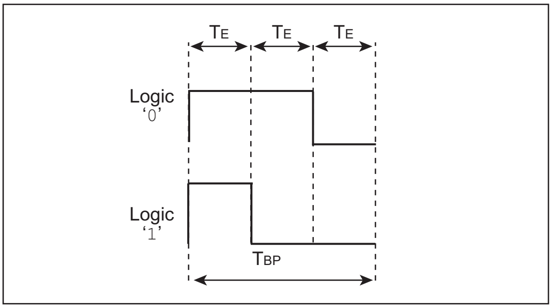


Figure 8-2: Manchester

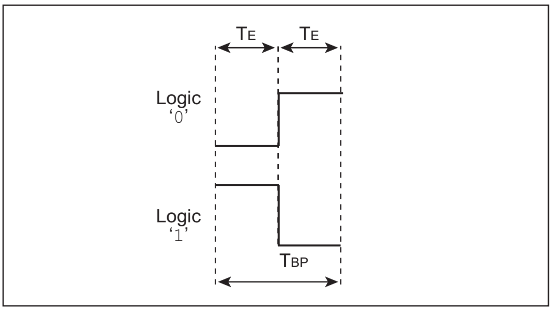


Figure 8-3: Pulse Position Modulation

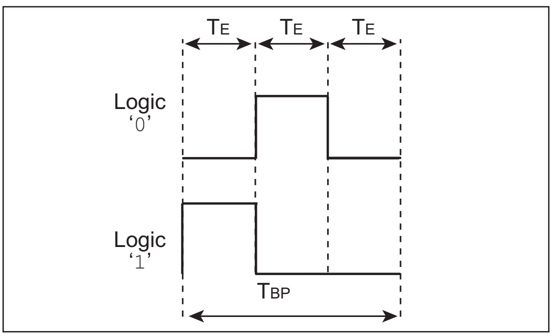
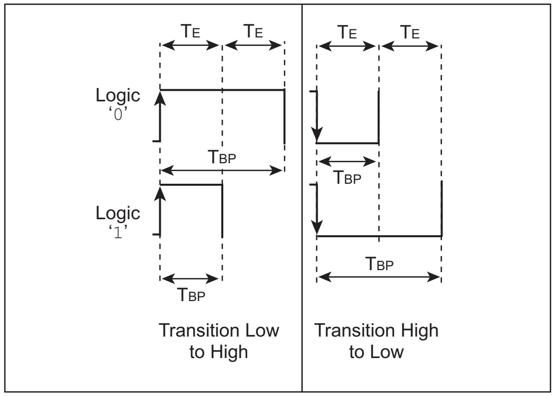


Figure 8-4: Variable Pulse-width Modulation



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.

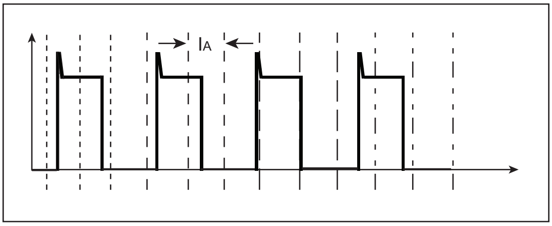
T_E is the basic timing element in each modulation format and will vary based on the desired baud rate.

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 #12 Repetitive Phase Shifted Sampling

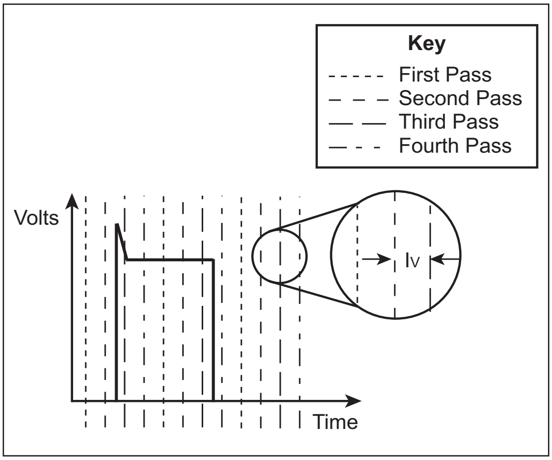
Repetitive phase shifted sampling is a technique to artificially increase the sampling rate of an A/D converter when sampling waveforms that are both periodic and constant from period to period. The technique works by capturing regularly spaced samples of the waveform from the start to finish of the 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



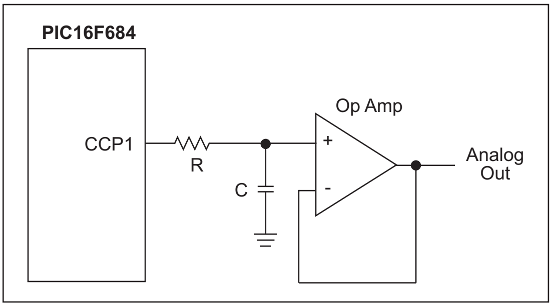
The CCP module is configured in Compare Special Event Trigger mode to accomplish this task. The phase shift is implemented by picking values of CCPRxL and CCPRxH that are not synchronous with the period of the sampling waveform. For instance, if the period of a waveform is 100 μs, then sampling at a 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 #16 Generating an Analog Output

Figure 16-1: Low-Pass Filter



Pulse-width modulated signals can be used to create Digital-to-Analog (D/A) converters with only a few external components. Conversion of PWM waveforms to analog signals involves the use of an analog low-pass filter. In order to eliminate unwanted harmonics caused by a PWM signal to the greatest degree possible, the frequency of the PWM signal (F_{PWM}) should be significantly higher than the bandwidth (F_{BW}) of the desired analog signal. Equation 16-1 shows this relation.

Equation 16-1

$$F_{PWM} = K*F_{BW}$$

Where harmonics decrease as K increases

R and C are chosen based on the following equation:

Equation 16-2

$$RC = 1/(2\pi F_{BW})$$

Pick a value of C arbitrarily and then calculate R. The attenuation of the PWM frequency for a given RC filter is:

Equation 16-3

$$Att(dB = – 10*log[1+(2\pi F_{PWM}RC)2]$$

If the attenuation calculated in Equation 16-3 is not sufficient, then K must be increased in Equation 16-1. See Application Note AN538 "*Using PWM to Generate Analog Output in PIC17C42*" for more details on using PWM to generate an analog output.

COMBINATION CAPTURE AND COMPARE TIPS

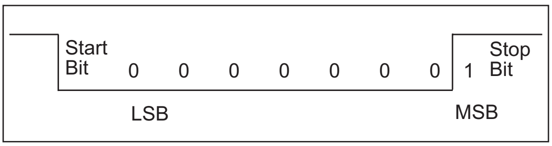
The CCP and ECCP modules can be configured on the fly. Therefore, these modules can perform different functions in the same application provided these functions operate exclusively (not at the same time). This section will provide examples of using a CCP module in different modes in the same application.

TIP #20 RS-232 Auto-baud

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

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

Figure 20-1: RS-232 Calibration Character



Auto-baud Routine Implementation:

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

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

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

NOTES:

TIP #19 Logic: XOR/XNOR Gate

This tip shows the use of the comparator to implement an XOR gate and its complement the XNOR gate.

The operation is best described in three sections:

1. Both A and B inputs are low
   With both inputs low, the inverting input is held at .7V and the non-inverting is held at ground. This combination results in a low output.
2. Both A and B inputs are high
   With both inputs high, the inverting input is pulled up to V_{DD} and the non-inverting input is equal to 2/3 V_{DD} (the average of V_{DD} inputs and GND). This combination also results in a low output.
3. Input A or B is high
   With one input high and one low, The inverting input is held at .7V and the non-inverting input is equal to 1/3 V_{DD} (the average of a V_{DD} input and GND). This combination results in a high output.

> **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 A and B must drive from ground to V_{DD} for the circuit to operate properly.
2. All resistances on the both inputs react with the input capacitance of the comparator, so the speed of the gate will be affected by the source resistance of A and B, as well as, the size of resistors R1, R2, R3 and R4.
3. Resistor R1, R2 and R3 must be equal.
4. Resistor R4 must be small enough to produce a 1.0V, or lower, voltage drop across D1 and D2.

Figure 19-1: XOR Gate

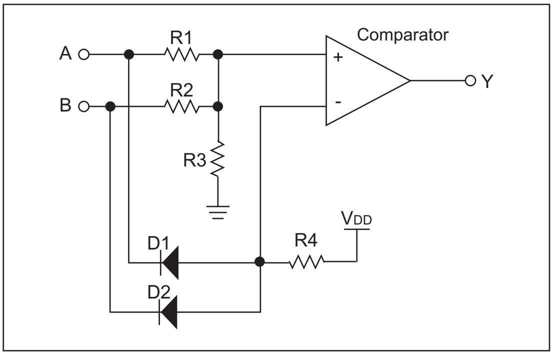
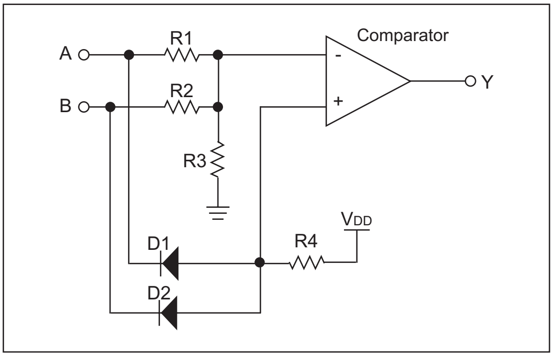


Figure 19-2: XNOR Gate



Example:

- D1 = D2, = 1N4148
- R4 = 10k, R1 = R2 = R3 = 5.1k

TIP #2 Brushless DC Motor Drive Circuits

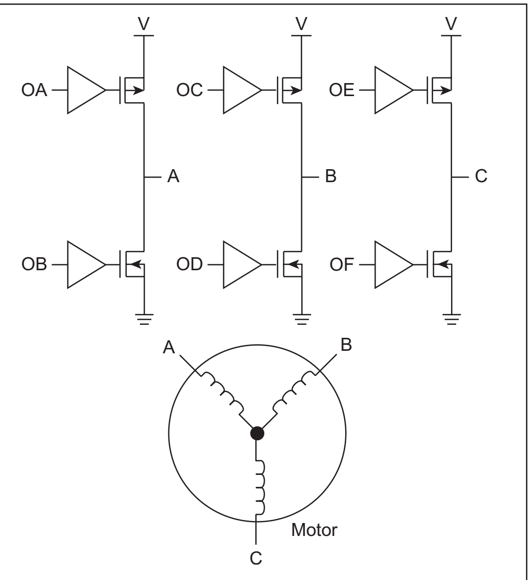
A Brushless DC motor is a good example of simplified hardware increasing the control complexity. The motor cannot commutate the windings (switch the current flow), so the control circuit and software must control the current flow correctly to keep the motor turning smoothly. The circuit is a simple half-bridge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



OA-OF are digital outputs from a PIC® MCU.

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

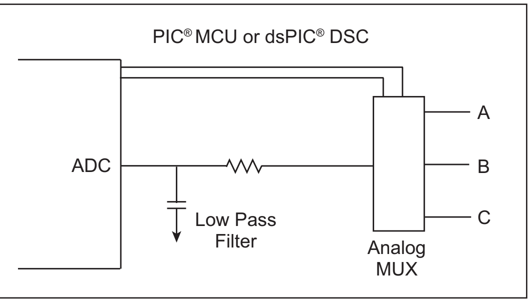
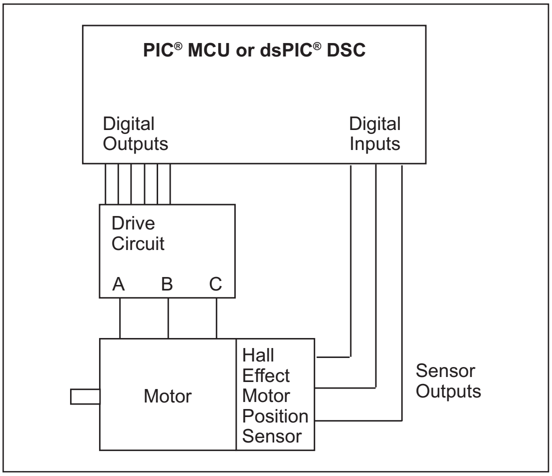


Figure 2-3: Quadrature Decoder (Sensor Motor)



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.

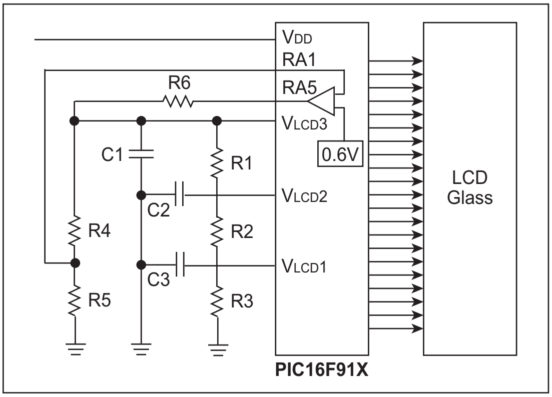
There are two basic types of stepper motors, although some motors are built to be used in either mode. The simplest stepper motor is the unipolar motor. This motor has four drive connections and one or two center tap wires that are tied to ground or Vsupply, depending on the implementation. Other motor 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 #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



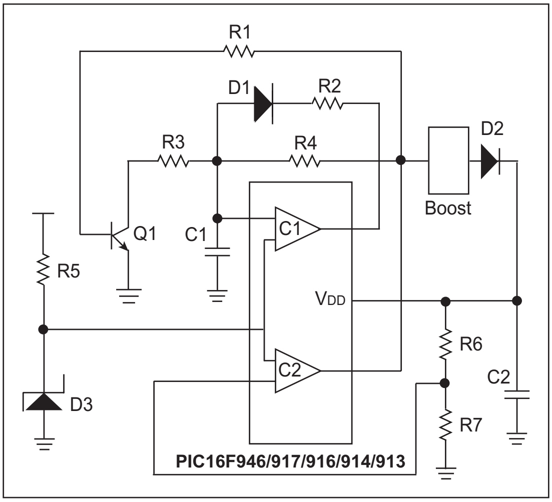
The PIC16F946/917/916/914/913 devices have a special Comparator mode that provides a fixed 0.6V reference. The circuit shown in Figure 4-1 makes use of this reference to provide a regulated contrast voltage. In this circuit, R1, R2 and R3 provide the contrast control voltages. The voltage on $V_{LCD}3$ 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 #10 How to Update LCD Data Through Firmware

To update the LCD, the content of the LCDDATA registers is modified to turn on, or off, each pixel on the LCD display. The application firmware will usually modify buffer variables that are created to correspond with elements on the display, such as character positions, bar graph, battery display, etc.

When the application calls for a display update, the values stored in the buffer variables must be converted to the correct setting of the pixel bits, located in the LCDDATA registers.

For Type-A waveforms, the LCD Data registers may be written any time without ill effect. However, for Type-B waveforms, the LCD Data registers can only be written every other LCD frame in order to ensure that the two frames of the Type-B waveform are compliments of one another. Otherwise, a DC bias can be presented to the LCD.

The LCD Data registers should only be written when a write is allowed, which is indicated by the WA bit in the LCDCON register being set.

On the PIC16C926 parts, there is no WA bit. The writing of the pixel data can be coordinated on an LCD interrupt. The LCD interrupt is only generated when a multiplexed (not static) Type-B waveform is selected.

TIP #11 Blinking LCD

Information can be displayed in more than one way with an LCD panel. For example, how can the user’s attention be drawn to a particular portion of the LCD panel? One way that does not require any additional segments is to create a blinking effect.

Look at a common clock application. The “:” between the hours and minutes is commonly made to blink once a second (on for half a second, off for half a second). This shows that the clock is counting in absence of the ticking sound or second hand that accompanies the usual analog face clock. It serves an important purpose of letting the user know that the clock is operating.

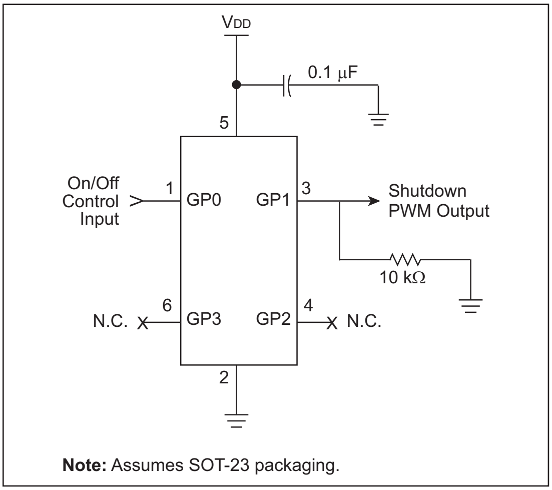
If there is a power outage, then it is common for the entire clock display to blink. This gives the user of the clock an immediate indication that the clock is no longer showing the correct time.

When the user sets the time, then blinking is commonly used to show that a new mode has been entered, such as blinking the hours to identify that the hours are being set, or blinking the minutes to show that the minutes are being set. In a simple clock, blinking is used for several different purposes. Without blinking effects, the common digital clock would not be nearly as user friendly.

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

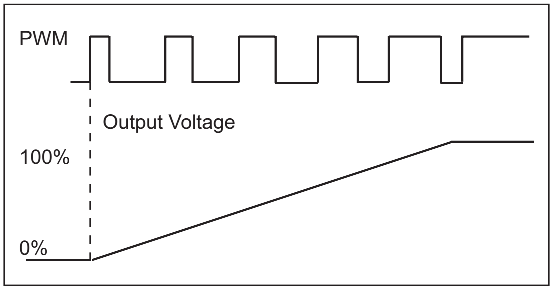


This technique is called soft-start and is used to prevent the large inrush currents that are associated with the start-up of a switching power supply.

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 #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

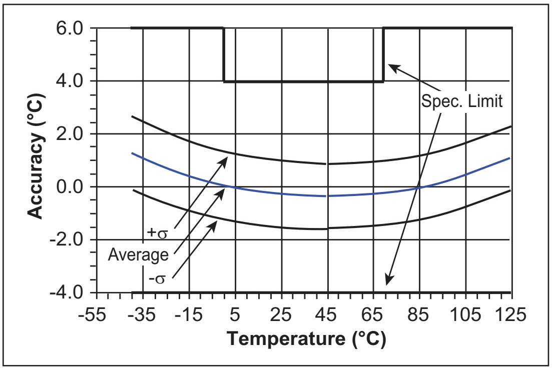


Figure 20-1 shows the accuracy of a 100 sample lot of MCP9700 temperature sensors. Despite the fact that the sensor's error is nonlinear, a PIC microcontroller (MCU) can be used to compensate the sensor's reading.

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

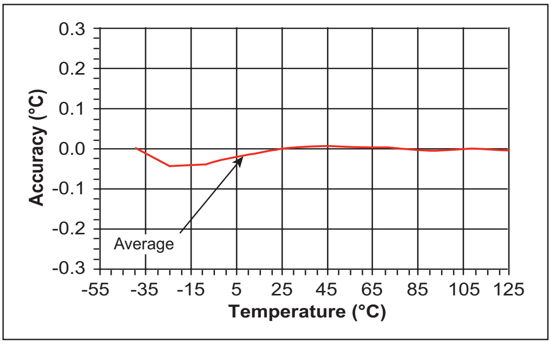


Figure 20-2 shows the average accuracy for the 100 sample lot of MCP9700 temperature sensors after compensation. The average error has been decreased over the full temperature range.

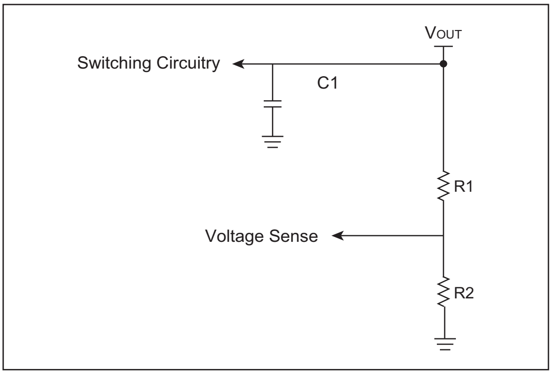
It is also possible to compensate for error from voltage references using this method.

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

TIP #21 Using Output Voltage Monitoring to Create a Self-Calibration Function

A PIC microcontroller can be used to create a switching power supply controlled by a PID loop (as described in Tip #16). This type of power supply senses its output voltage digitally, compares that voltage to the desired reference voltage and makes duty cycle changes accordingly. Without calibration, it is sensitive to component tolerances.

Figure 21-1: Typical Power Supply Output Stage



The output stage of many power supplies is similar to Figure 21-1. R1 and R2 are used to set the ratio of the voltage that is sensed and compared to the reference.

A simple means of calibrating this type of power supply is as follows:

1. Supply a known reference voltage to the output of the supply.
2. Place the supply in Calibration mode and allow it to sense that reference voltage.

By providing the supply with the output voltage that it is to produce, it can then sense the voltage across the resistor divider and store the sensed value. Regardless of resistor tolerances, the sensed value will always correspond to the proper output value for that particular supply.

Futhermore, this setup could be combined with Tip #20 to calibrate at several temperatures.

This setup could also be used to create a programmable power supply by changing the supplied reference and the resistor divider for voltage feedback.

TIP #7 3.3V → 5V Using a Diode Offset

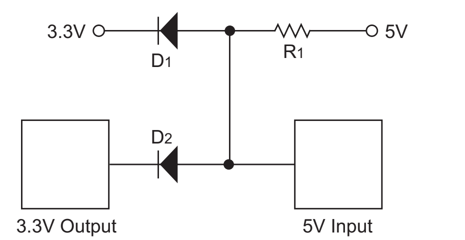
The inputs voltage thresholds for 5V CMOS and the output drive voltage for 3.3V LVTTL and LVCMOS are listed in Table 7-1.

Table 7-1: Input/Output Thresholds

	5V CMOS Input	3.3V LVTTL Output	3.3V LVCMOS Output
High Threshold	> 3.5V	> 2.4V	> 3.0V
Low Threshold	< 1.5V	< 0.4V	< 0.5V

Note that both the high and low threshold input voltages for the 5V CMOS inputs are about a volt higher than the 3.3V outputs. So, even if the output from the 3.3V system could be offset, there would be little or no margin for noise or component tolerance. What is needed is a circuit that offsets the outputs and increases the difference between the high and low output voltages.

Figure 7-1: Diode Offset



When output voltage specifications are determined, it is done assuming that the output is driving a load between the output and ground for the high output, and a load between 3.3V and the output for the low output. If the load for the high threshold is actually between the output and 3.3V, then the output voltage is actually much higher as the load resistor is the mechanism that is pulling the output up, instead of the output transistor.

If we create a diode offset circuit (see Figure 7-1), the output low voltage is increased by the forward voltage of the diode D1, typically 0.7V, creating a low voltage at the 5V CMOS input of 1.1V to 1.2V. This is well within the low threshold input voltage for the 5V CMOS input. The output high voltage is set by the pull-up resistor and diode D2, tied to the 3.3V supply. This puts the output high voltage at approximately 0.7V above the 3.3V supply, or 4.0 to 4.1V, which is well above the 3.5V threshold for the 5V CMOS input.

> **Note:** For the circuit to work properly, the pull-up resistor must be significantly smaller than the input resistance of the 5V CMOS input, to prevent a reduction in the output voltage due to a resistor divider effect at the input. The pull-up resistor must also be large enough to keep the output current loading on the 3.3V output within the specification of the device.

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	$I_C = 50\ \mu A$, $I_E = 0$
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	$I_E = 50\ \mu A$, $I_C = 0$
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

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

<u>$R_{BASE} = 4.14\ k\Omega$, I/O port current = 556 µA</u>

5V Technology Example

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

<u>$R_{BASE} = 7.74\ k\Omega$, I/O port current = 556 µA</u>

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.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4080

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

03/26/09