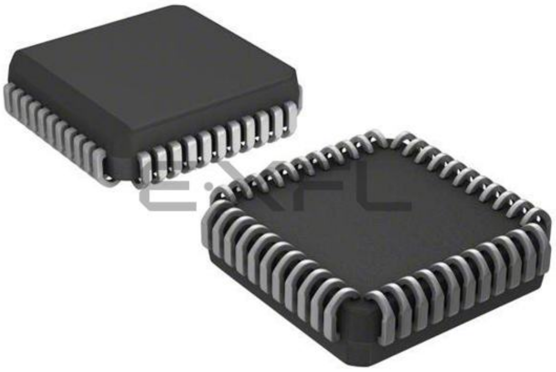
Microchip Technology - PIC16C65B-04I/L Datasheet



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	-40°C ~ 85°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-04i-l

CHAPTER 1
8-Pin Flash PIC® Microcontrollers Tips 'n Tricks

Table Of Contents

TIPS 'N TRICKS INTRODUCTION

Microchip continues to provide innovative products that are smaller, faster, easier to use and more reliable. The 8-pin Flash PIC® microcontrollers (MCU) are used in an wide range of everyday products, from toothbrushes, hair dryers and rice cookers to industrial, automotive and medical products.

The PIC12F629/675 MCUs merge all the advantages of the PIC MCU architecture and the flexibility of Flash program memory into an 8-pin package. They provide the features and intelligence not previously available due to cost and board space limitations. Features include a 14-bit instruction set, small footprint package, a wide operating voltage of 2.0 to 5.5 volts, an internal programmable 4 MHz oscillator, on-board EEPROM data memory, on-chip voltage reference and up to 4 channels of 10-bit A/D. The flexibility of Flash and an excellent development tool suite, including a low-cost In-Circuit Debugger, In-Circuit Serial Programming™ and MPLAB® ICE 2000 emulation, make these devices ideal for just about any embedded control application.

TIPS 'N TRICKS WITH HARDWARE

The following series of Tips 'n Tricks can be applied to a variety of applications to help make the most of the 8-pin dynamics.

TIP #3 Read Three States From One Pin

To check state Z:

- Drive output pin high
- Set to Input
- Read 1
- Drive output pin low
- Set to Input
- Read 0

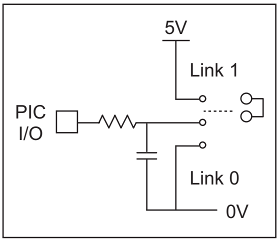
To check state 0:

- Read 0 on pin

To check state 1:

- Read 1 on pin

Figure 3-1



State	Link 0	Link 1
0	closed	open
1	open	closed
NC	open	open

Jumper has three possible states: not connected, Link 1 and Link 0. The capacitor will charge and discharge depending on the I/O output voltage allowing the “not connected” state. Software should check the “not connected” state first by driving I/O high, reading 1 and driving I/O low and reading 0. The “Link 1” and “Link 0” states are read directly.

TIP #4 Reading DIP Switches

The input of a timer can be used to test which switch(s) is closed. The input of Timer1 is held high with a pull-up resistor. Sequentially, each switch I/O is set to input and Timer1 is checked for an increment indicating the switch is closed.

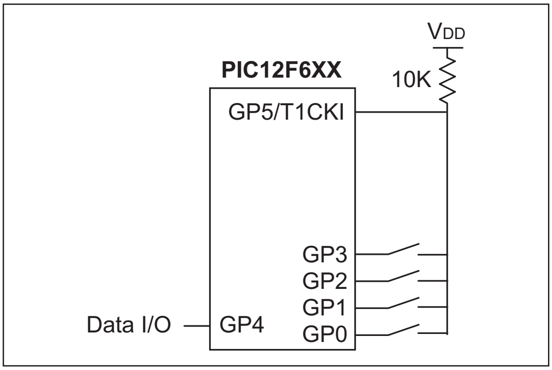
Example 4-1

```
        movlw    b'11111111'
        movwf    TRISIO
        movwf    DIP
        movlw    b'00000111'
        movwf    T1CON
        movlw    b'11111110'
        movwf    Mask
        clrf     GPIO
LOOP
        clrf     TMR1L
        movf     Mask,W
        movwf    TRISIO
        btfsc    TMR1L,0
        andwf    DIP,F
        bsf      STATUS,C
        rlf      Mask,F
        btfsc    Mask,4
        goto     Loop
        retlw    0
```

Each bit in the DP register represents its corresponding switch position. By setting Timer1 to FFFFh and enabling its interrupt, an increment will cause a rollover and generate an interrupt. This will simplify the software by eliminating the bit test on the TMR1L register.

Sequentially set each GPIO to an input and test for TMR1 increment (or 0 if standard I/O pin is used).

Figure 4-1



TIP #16 Optimizing Destinations

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

Example 16-1

Example: A + B → A

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

3 instructions

```
MOVF    B,W
ADDWF   A,F
```

2 instructions

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

TIP #17 Conditional Bit Set/Clear

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

Example 17-1

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

4 instructions

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

3 instructions

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

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

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

TIP #3 Configuring Port Pins

All PIC MCUs have bidirectional I/O pins. Some of these pins have analog input capabilities. It is very important to pay attention to the signals applied to these pins so the least amount of power will be consumed.

Unused Port Pins

If a port pin is unused, it may be left unconnected but configured as an output pin driving to either state (high or low), or it may be configured as an input with an external resistor (about 10 kΩ) pulling it to V_{DD} or V_{SS}. If configured as an input, only the pin input leakage current will be drawn through the pin (the same current would flow if the pin was connected directly to V_{DD} or V_{SS}). Both options allow the pin to be used later for either input or output without significant hardware modifications.

Digital Inputs

A digital input pin consumes the least amount of power when the input voltage is near V_{DD} or V_{SS}. If the input voltage is near the midpoint between V_{DD} and V_{SS}, the transistors inside the digital input buffer are biased in a linear region and they will consume a significant amount of current. If such a pin can be configured as an analog input, the digital buffer is turned off, reducing both the pin current as well as the total controller current.

Analog Inputs

Analog inputs have a very high-impedance so they consume very little current. They will consume less current than a digital input if the applied voltage would normally be centered between V_{DD} and V_{SS}. Sometimes it is appropriate and possible to configure digital inputs as analog inputs when the digital input must go to a low power state.

Digital Outputs

There is no additional current consumed by a digital output pin other than the current going through the pin to power the external circuit. Pay close attention to the external circuits to minimize their current consumption.

TIP #4 Use High-Value Pull-Up Resistors

It is more power efficient to use larger pull-up resistors on I/O pins such as MCLR, I^2C™ signals, switches and for resistor dividers. For example, a typical I^2C pull-up is 4.7k. However, when the I^2C is transmitting and pulling a line low, this consumes nearly 700 uA of current for each bus at 3.3V. By increasing the size of the I^2C pull-ups to 10k, this current can be halved. The tradeoff is a lower maximum I^2C bus speed, but this can be a worthwhile trade in for many low power applications. This technique is especially useful in cases where the pull-up can be increased to a very high resistance such as 100k or 1M.

TIP #5 Reduce Operating Voltage

Reducing the operating voltage of the device, V_{DD}, is a useful step to reduce the overall power consumption. When running, power consumption is mainly influenced by the clock speed. When sleeping, the most significant factor is leakage in the transistors. At lower voltages, less charge is required to switch the system clocks and transistors leak less current.

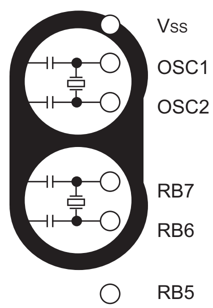
It is important to pay attention to how reducing the operating voltage reduces the maximum allowed operating frequency. Select the optimum voltage that allows the application to run at its maximum speed. Refer to the device data sheet for the maximum operating frequency of the device at the given voltage.

TIP #19 Low Power Timer1 Oscillator Layout

Applications requiring very low power Timer1/SOSC oscillators on nanoWatt and nanoWatt XLP devices must take PCB layout into consideration. The very low power Timer1/SOSC oscillators on nanoWatt and nanoWatt XLP devices consume very little current, and this sometimes makes the oscillator circuit sensitive to neighboring circuits. The oscillator circuit (crystal and capacitors) should be located as close as possible to the microcontroller.

No circuits should be passing through the oscillator circuit boundaries. If it is unavoidable to have high-speed circuits near the oscillator circuit, a guard ring should be placed around the oscillator circuit and microcontroller pins similar to the figure below. Placing a ground plane under the oscillator components also helps to prevent interaction with high speed circuits.

Figure 19-1: Guard Ring Around Oscillator Circuit and MCU Pins



TIP #20 Use LVD to Detect Low Battery

The Low Voltage Detect (LVD) interrupt present in many PIC MCUs is critical in battery based systems. It is necessary for two reasons. First, many devices cannot run full speed at the minimum operating voltage. In this case, the LVD interrupt indicates when the battery voltage is dropping so that the CPU clock can be slowed down to an appropriate speed, preventing code misexecution. Second, it allows the MCU to detect when the battery is nearing the end of its life, so that a low battery indication can be provided and a lower power state can be entered to maximize battery lifetime. The LVD allows these functions to be implemented without requiring the use of extra analog channels to measure the battery level.

TIP #21 Use Peripheral FIFO and DMA

Some devices have peripherals with DMA or FIFO buffers. These features are not just useful to improve performance; they can also be used to reduce power. Peripherals with just one buffer register require the CPU to stay operating in order to read from the buffer so it doesn’t overflow. However, with a FIFO or DMA, the CPU can go to sleep or idle until the FIFO fills or DMA transfer completes. This allows the device to consume a lot less average current over the life of the application.

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

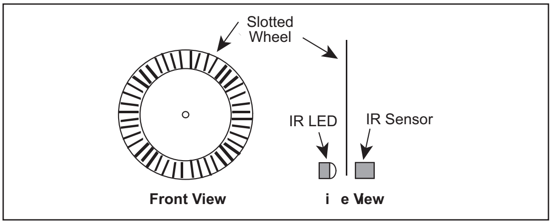
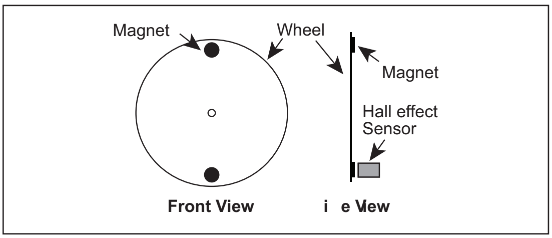


Figure 5-2: Hall Effect Sensor



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

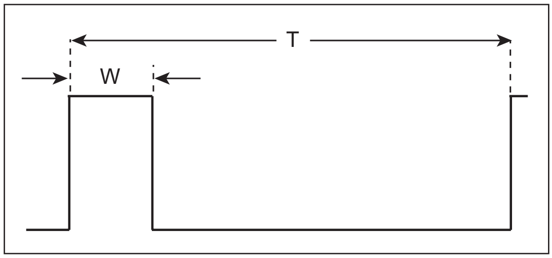
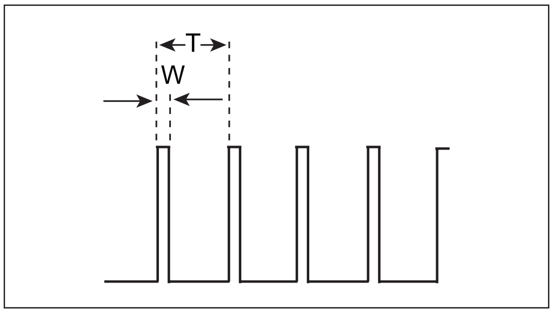


Figure 5-4: High RPM

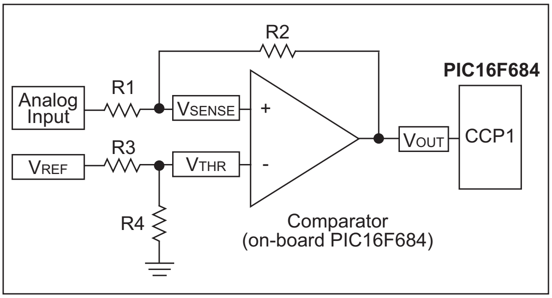


TIP #6 Measuring the Period of an Analog Signal

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

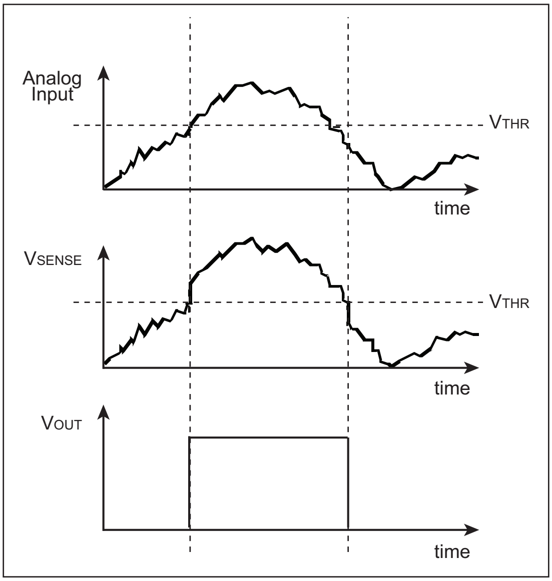
Figure 6-1 shows an example circuit using the peripherals of the PIC16F684.

Figure 6-1: Circuit



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

Figure 6-2: Signal Comparison



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

TIP #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

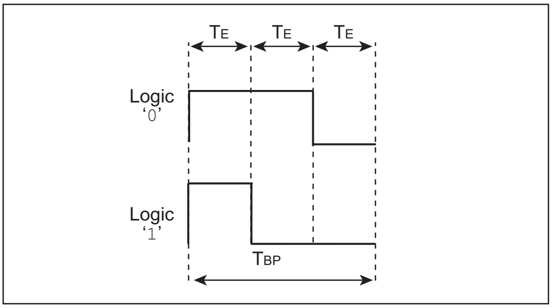


Figure 8-2: Manchester

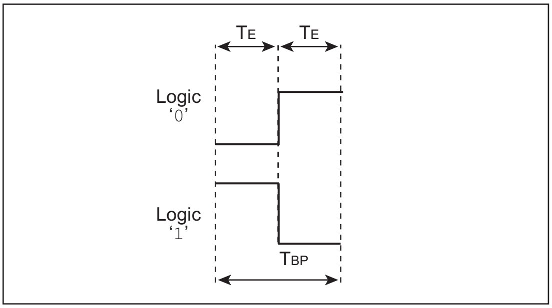


Figure 8-3: Pulse Position Modulation

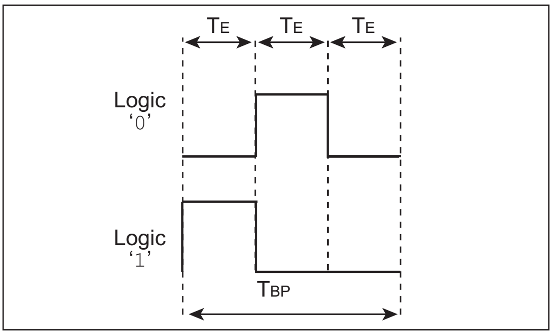
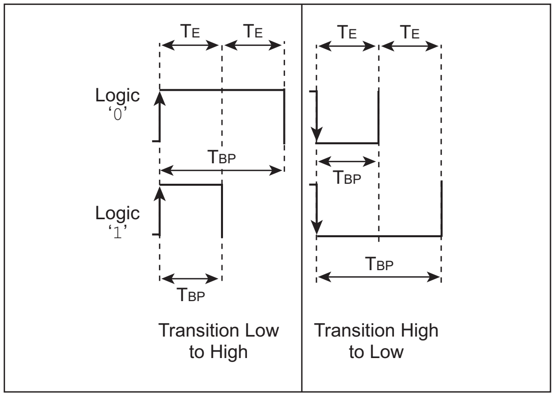


Figure 8-4: Variable Pulse-width Modulation



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).

Example 20-1: Transmit Routine

```
TxRountine
   MOVLW  8         ;preload bit counter
                    ;with 8
   MOVWF  counter
   BCF    TxLine    ;line initially high,
                    ;toggle low for START
                    ;bit
TxLoop
   CALL   DelayTb   ;wait Tb (bit period)
   RRF    RxByte,f  ;rotate LSB first into
                    ;the Carry flag
   BTFSS  STATUS,C  ;Tx line state equals
                    ;state of Carry flag
   BCF    TxLine
   BTFSC  STATUS,C
   BSF    TxLine
   DECFSZ Counter,f ;Repeat 8 times
   GOTO   TxLoop
   CALL   Delay Tb  ;Delay Tb before
                    ;sending STOP bit
   BSF    TxLine    ;send STOP bit
```

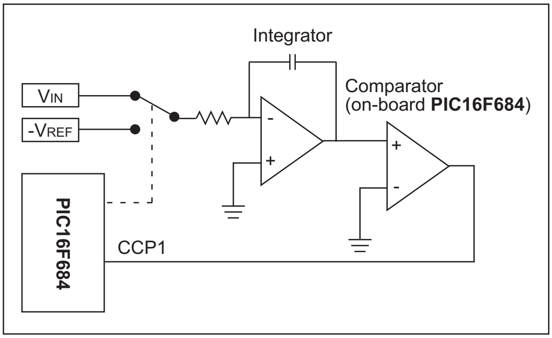
Example 20-2: Receive Routine

```
RxRoutine
   BTFSC  RxLine      ;wait for receive
                      ;line to go low
   GOTO   RxRoutine
   MOVLW  8           ;initialize bit
                      ;counter to 8
   MOVWF  Counter
   CALL   Delay1HalfTb;delay 1/2 Tb here
                      ;plus Tb in RxLoop
                      ;in order to sample
                      ;at the right time
RxLoop
   CALL   DelayTb     ;wait Tb (bit
                      ;period)
   BTFSS  RxLine      ;Carry flag state
                      ;equals Rx line
                      ;state
   BCF    STATUS,C
   BTFSC  RxLine
   BSF    STATUS,C
   BTFSC  RxLine
   BSF    STATUS,C
   RRF    RxByte,f    ;Rotate LSB first
                      ;into receive type
   DECFSZ Counter,f   ;Repeat 8 times
   GOTO   RxLoop
```


TIP #21 Dual-Slope Analog-to-Digital Converter

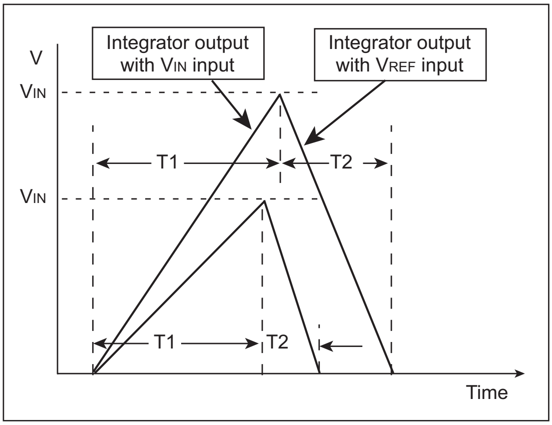
A circuit for performing dual-slope A/D conversion utilizing the CCP module is shown in Figure 21-1.

Figure 21-1: Dual-Slope Analog-to-Digital Converter



Dual-slope A/D conversion works by integrating the input signal (V_{IN}) for a fixed time (T1). The input is then switched to a negative reference ($-V_{REF}$) and integrated until the integrator output is zero (T2). V_{IN} is a function of V_{REF} and the ratio of T2 to T1.

Figure 21-2: V vs. Time



The components of this conversion type are the fixed time and the timing of the falling edge. The CCP module can accomplish both of these components via Compare mode and Capture mode respectively. Here’s how:

1. Configure the CCP module in Compare mode, Special Event Trigger.
2. Switch the analog input into the integrator from V_{REF} to V_{IN}.
3. Use the CCP module to wait T1 (T1 chosen based on capacitor value).
4. When the CCP interrupt occurs, switch the analog input into the regulator from V_{IN} to V_{REF} and reconfigure the module in Capture mode; wait for falling edge.
5. When the next CCP interrupt occurs, the time captured by the module is T2.
6. Calculate V_{IN} using Equation 21-1.

Equation 21-1

$$V_{IN} = V_{REF} \frac{T2}{T1}$$

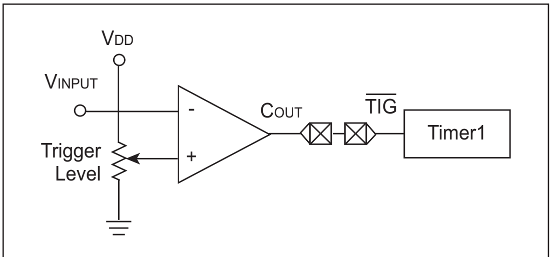
TIP #4 Pulse Width Measurement

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 $\overline{T1G}$ input is high, Timer1 does not count. Combining $\overline{T1G}$ 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.

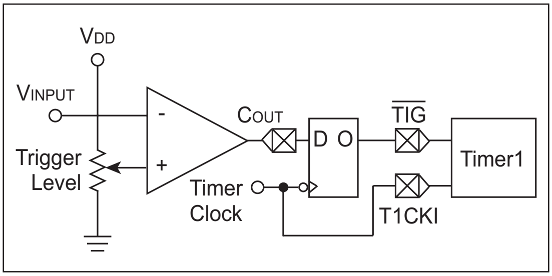
Figure 4-1: Comparator with Timer1 and $\overline{T1G}$



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 #17 Logic: AND/NAND Gate

This tip shows the use of the comparator to implement an AND gate and its complement the NAND gate (see Figure 17-2). Resistors R1 and R2 drive the non-inverting input with 2/3 the supply voltage. Resistors R3 and R4 average the voltage of input A and B at the inverting input. If either A or B is low, the average voltage will be one half V_{DD} and the output of the comparator remains low. The output will go high only if both inputs A and B are high, which raises the input to the inverting input above 2/3 V_{DD}.

The operation of the NAND gate is identical to the AND gate, except that the output is inverted due to the swap of the inverting and non-inverting inputs.

> **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. The combination of R1 and R2 will draw current constantly, so they must be kept large to minimize current draw.
3. All resistances on the inverting input 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 R3 and R4.
4. Resistor R2 must be 2 x R1.
5. Resistor R3 must be equal to R4.

Figure 17-1: AND Gate

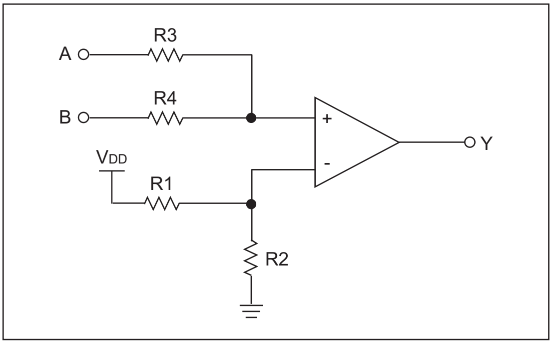
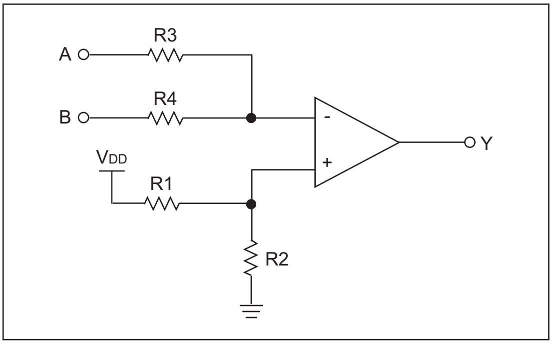


Figure 17-2: NAND Gate



Example:

- V_{DD} = 5V, R3 = R4 = 10k
- R1 = 5.1k, R2 = 10k

TIP #18 Logic: OR/NOR Gate

This tip shows the use of the comparator to implement an OR gate, and its complement, the NOR gate.

Resistors R1 and R2 drive the non-inverting input of the comparator with 1/3 V_{DD}. Resistors R3 and R4 average the voltages of the inputs A and B at the inverting input. If either A or B is high, the average voltage is 1/2 V_{DD} and the output of the comparator is high. Only if both A and B are low does the average voltage at the non-inverting input drop below 1/3 the supply voltage, causing the comparator output to go low. The operation of the NOR gate is identical to the OR gate, except the output is inverted due to the swap of the inverting and non-inverting inputs.

> **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. The combination of R1 and R2 will draw current constantly, so they must be kept large to minimize current draw.
3. All resistances on the inverting input 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 R3 and R4.
4. Resistor R1 must be 2 x R2.
5. Resistor R3 must be equal to R4.

Figure 18-1: OR Gate

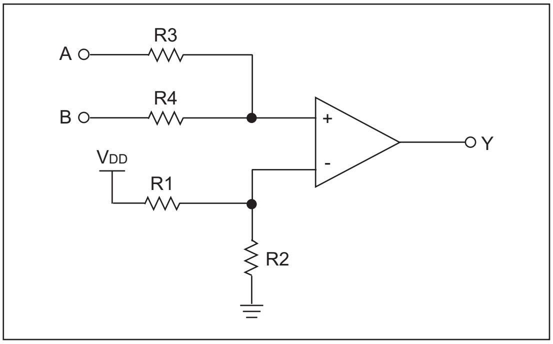
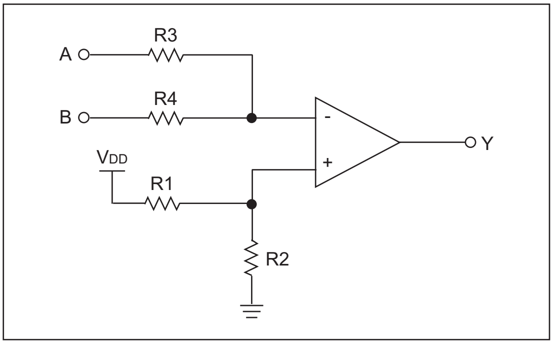


Figure 18-2: NOR Gate



Example:

- V_{DD} = 5V, R3 = R4 = 10k
- R1 = 10k, R2 = 5.1k

TIP #5 Writing a PWM Value to the the CCP Registers With a Mid-Range PIC® Microcontroller

The two PWM LSb's are located in the CCPCON register of the CCP. This can make changing the PWM period frustrating for a developer. Example 5-1 through Example 5-3 show three macros written for the mid-range product family that can be used to set the PWM period. The first macro takes a 16-bit value and uses the 10 MSb's to set the PWM period. The second macro takes a 16-bit value and uses the 10 LSb's to set the PWM period. The last macro takes 8 bits and sets the PWM period. This assumes that the CCP is configured for no more than 8 bits.

Example 5-1: Left Justified 16-Bit Macro

```
pwm_tmp    equ xxx  ;this variable must be
                    ;allocated someplace
setPeriod  macro a  ;a is 2 SFR's in
                    ;Low:High arrangement
                    ;the 10 MSb's are the
                    ;desired PWM value
 RRF       a,w      ;This macro will
                    ;change w
 MOVWF     pwm_tmp
 RRF       pwm_tmp,w
 ANDLW     0x30
 IORLW     0x0F
 MOVWF     CCP1CON
 MOVF      a+1,w
 MOVWF     CCPR1L
```

Example 5-2: Right Justified 16-Bit Macro

```
pwm_tmp    equ xxx  ;this variable must be
                    ;allocated someplace
setPeriod  macro a  ;a is 2 bytes in
                    ;Low:High arrangement
                    ;the 10 LSb's are the
                    ;desired PWM value
 SWAPF     a,w      ;This macro will
                    ;change w
 ANDLW     0x30
 IORLW     0x0F
 MOVWF     CCP1CON
 RLF       a,w
 IORLW     0x0F
 MOVWF     pwm_tmp
 RRF       pwm_tmp,f
 RRF       pwm_tmp,w
 MOVWF     CCPR1L
```

Example 5-3: 8-Bit Macro

```
pwm_tmp    equ xxx  ;this variable must be
                    ;allocated someplace
setPeriod  macro a  ;a is 1 SFR
 SWAPF     a,w      ;This macro will
                    ;change w
 ANDLW     0x30
 IORLW     0x0F
 MOVWF     CCP1CON
 RRF       a,w
 MOVWF     pwm_tmp
 RRF       pwm_tmp,w
 MOVWF     CCPR1L
```

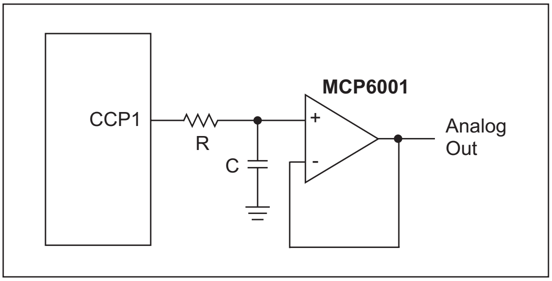
Application Note References

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

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

TIP #11 Generating a Reference Voltage with a PWM Output

Figure 11-1: Low-Pass Filter



A PWM signal can be used to create a Digital-to-Analog Converter (DAC) 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, the PWM frequency (F_{PWM}) should be significantly higher than the bandwidth (F_{BW}) of the desired analog signal. Equation 11-1 shows this relation.

Equation 11-1

$$F_{PWM} = K \cdot F_{BW}$$

Where harmonics decrease as K increases.

R and C are chosen based on the following equation:

Equation 11-2

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

Where harmonics decrease as K increases.

Choose the R value based on drive capability and then calculate the required C value. The attenuation of the PWM frequency for a given RC filter is shown in Equation 11-3.

Equation 11-3

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

If the attenuation calculated in Equation 11-3 is not sufficient, then K must be increased in Equation 11-1.

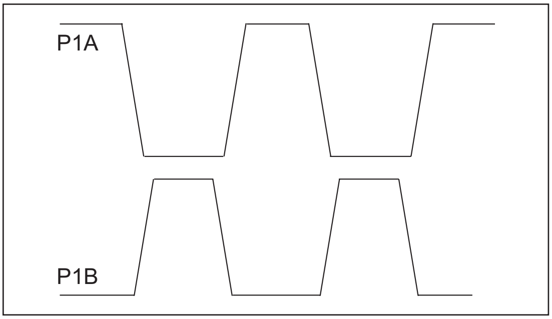
In order to sufficiently attenuate the harmonics, it may be necessary to use small capacitor values or large resistor values. Any current draw will effect the voltage across the capacitor. Adding an op amp allows the analog voltage to be buffered and, because of this, any current drawn will be supplied by the op amp and not the filter capacitor.

For more information on using a PWM signal to generate an analog output, refer to AN538, "*Using PWM to Generate Analog Output*" (DS00538).

TIP #13 Generating a Two-Phase Control Signal

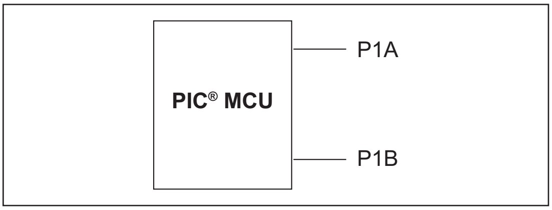
Power supplies using a push-pull topology or with multiple switching components require a two-phase control signal as shown in Figure 13-1.

Figure 13-1: Two-Phase Control Signal



It is possible to produce this type of control signal with two out-of-phase square waves using a PIC MCU with an ECCP module.

Figure 13-2: Two-Phase Control Signal Schematic



In order to configure the ECCP to produce this type of output:

1. Configure the ECCP in half H-bridge configuration PWM pulse with both outputs active-high.
2. Set the duty cycle register (CCPR1L) with the maximum duty cycle of 50%.
3. Change the programmable dead-time generator to reduce the pulse width to the desired value.

The programmable dead-time generator has a 7-bit resolution and, therefore, the resulting pulses will only have a 7-bit resolution. Each pulse will have a 50% duty cycle, less the dead time.

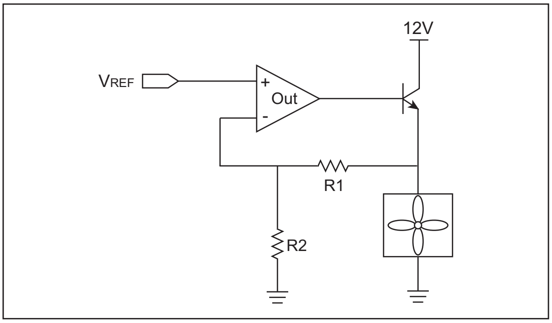
Using an internal 4 MHz clock produces 31 kHz output pulses, and using a 20 MHz crystal would produce 156 kHz output. The frequency of the output could be increased with a loss in resolution.

Example software is provided for the PIC16F684, but this tip is applicable to all PIC MCUs with ECCP modules.

Method 2 – Linear Control

When using PWM, the voltage will vary between a maximum and a minimum, however, is it also possible to use a linear method to control fan speed, as shown in Figure 14-4.

Figure 14-4: Linear Control Drive



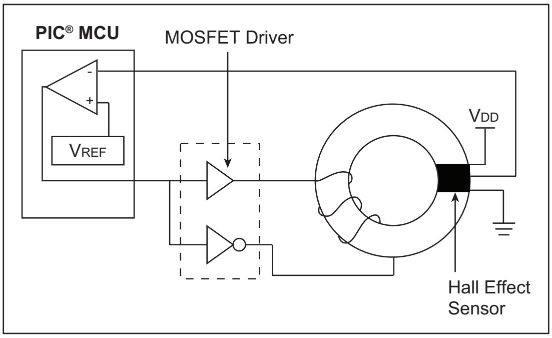
The voltage applied at the non-inverting terminal of the op amp is used to vary the voltage across the op amp. The non-inverting terminal voltage can be produced by a Digital-to-Analog Converter (DAC) or by the method shown in Tip #11.

When using this method, care must be taken to ensure that the fan voltage is not too low or the fan will stop spinning. One advantage this method has over PWM is that the tachometer output will function properly on 3-wire fans. The disadvantage, however, is that it often offers less speed control. For example, a 12V fan will not spin below 8V, so a range of only 4V is available for speed control. A 5V fan will not spin below 4V and so the control range is only 1V, which is often unacceptable. Another disadvantage is the power consumption of the circuit. The transistor will dissipate more power than the PWM method.

TIP #15 High Current Delta-Sigma Based Current Measurement Using a Slotted Ferrite and Hall Effect Device

Many current sensors rely on ferrite cores. Non-linearity in the ferrite can lead to inaccurate results, especially at high currents. One way to avoid the non-linearities is to keep the net flux in the ferrite near zero. Consider the circuit in Figure 15-1.

Figure 15-1: Hall Effect Current Measurement Schematic



The Hall effect sensor output is proportional to the current being measured. When I_{IN} = 0 amps, the output of the sensor will be $V_{DD}/2$. A current passing through the sensor in one direction will increase the output of the sensor, and a current in the other direction will decrease the output of the sensor.

The output of the comparator is used to drive a coil of wire wound around the ferrite core. This coil of wire will be used to create flux in the opposite direction as the flux imposed in the core.

Neglecting the affects of R_S and R_L, the formula for determining the values for R1 and R2 is given by Equation 12-1.

Equation 12-1: Divider Values

$$\frac{V_S}{R1 + R2} = \frac{V_L}{R2} \quad \text{; General relationship}$$

$$R1 = \frac{(V_S - V_L) \bullet R2}{V_L} \quad \text{; Solving for } R_1$$

$$R1 = 0.515 \bullet R2 \quad \text{; Substituting voltages}$$

The formula for determining the rise and fall times is given in Equation 12-2. For circuit analysis, the Thevenin equivalent is used to determine the applied voltage, V_A, and the series resistance, R. The Thevenin equivalent is defined as the open circuit voltage divided by the short circuit current. The Thevenin equivalent, R, is determined to be 0.66*R1 and the Thevenin equivalent, V_A, is determined to be 0.66*V_S for the circuit shown in Figure 12-2 according to the limitations imposed by Equation 12-2.

Equation 12-2: Rise/Fall Time

$$t = -\left[R \bullet C \bullet \ln\left(\frac{V_F - V_A}{V_I - V_A}\right)\right]$$

Where:

- t = Rise or Fall time
- R = 0.66*R_1
- C = C_S+C_L
- V_I = Initial voltage on C (V_L)
- V_F = Final voltage on C (V_L)
- V_A = Applied voltage (0.66*V_S)

As an example, suppose the following conditions exist:

- Stray capacitance = 30 pF
- Load capacitance = 5 pF
- Maximum rise time from 0.3V to 3V ≤ 1 µS
- Applied source voltage Vs = 5V

The calculation to determine the *maximum* resistances is shown in Equation 12-3.

Equation 12-3: Example Calculation

Solve Equation 12-2 for *R*:

$$R = -\left[\frac{t}{C \bullet \ln\left(\frac{V_F - V_A}{V_I - V_A}\right)}\right]$$

Substitute values:

$$R = -\left[\frac{10 \bullet 10^{-7}}{35 \bullet 10^{-12} \bullet \ln\left(\frac{3 - (0.66 \bullet 5)}{0.3 - (0.66 \bullet 5)}\right)}\right]$$

Thevenin equivalent maximum R:

$R = 12408$

Solve for maximum R1 and R2:

$R1 = 0.66 \bullet R$ $\qquad R2 = \frac{R1}{0.515}$

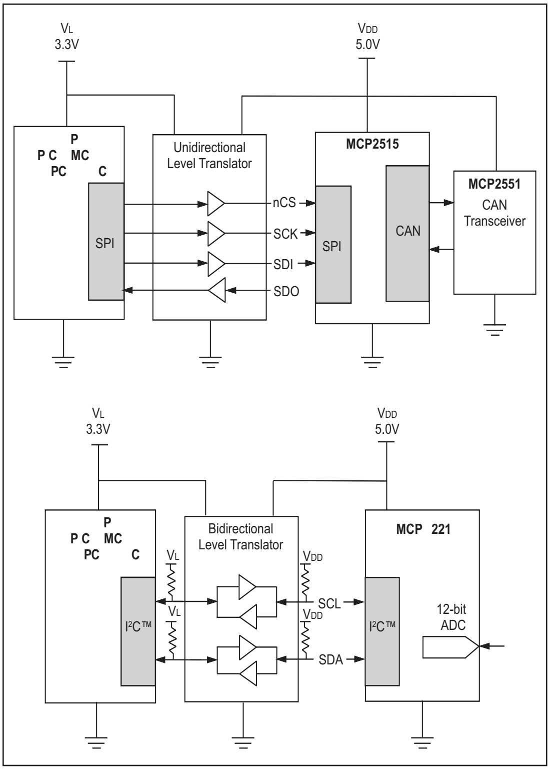
$R1 = 8190$ $\qquad R2 = 15902$

TIP #13 3.3V → 5V Level Translators

While level translation can be done discretely, it is often preferred to use an integrated solution. Level translators are available in a wide range of capabilities. There are unidirectional and bidirectional configurations, different voltage translations and different speeds, all giving the user the ability to select the best solution.

Board-level communication between devices (e.g., MCU to peripheral) is most often done by either SPI or I²C™. For SPI, it may be appropriate to use a unidirectional level translator and for I²C, it is necessary to use a bidirectional solution. Figure 13-1 illustrates both solutions.

Figure 13-1: Level Translator



Analog

The final 3.3V to 5V interface challenge is the translation of analog signals across the power supply barrier. While low level signals will probably not require external circuitry, signals moving between 3.3V and 5V systems will be affected by the change in supply. For example, a 1V peak analog signal converted by an ADC in a 3.3V system will have greater resolution than an ADC in a 5V system, simply because more of the ADCs range is used to convert the signal in the 3.3V ADC. Alternately, the relatively higher signal amplitude in a 3.3V system may have problems with the system’s lower common mode voltage limitations.

Therefore, some interface circuitry, to compensate for the differences, may be needed. This section will discuss interface circuitry to help alleviate these problems when the signal makes the transition between the different supply voltages.