MICROCHIP

PICmicro®
CCP and ECCP Tips ‘n Tricks

Table of Contents

© 2003 Microchip Technology Inc.

Combination Capture and Compare Tips

© 2003 Microchip Technology Inc.

TIPS 'N TRICKS INTRODUCTION

Microchip continues to provide innovative products that are smaller, faster, easier-to-use and more reliable. PICmicro® microcontrollers (MCUs) are used in a wide range of everyday products, from washing machines, garage door openers, and television remotes to industrial, automotive, and medical products.

The Capture, Compare, and PWM (CCP) modules that are found on many of Microchip's microcontrollers are used primarily for the measurement and control of time-based pulse signals. The Enhanced CCP (ECCP), available on some of Microchip's devices, differs from the regular CCP module in that it provides enhanced PWM functionality – namely, full-bridge and half-bridge support, programmable dead-band delay, and enhanced PWM auto-shutdown. The ECCP and CCP modules are capable of performing a wide variety of tasks. This document will describe some of the basic guidelines to follow when using these modules in each mode, as well as give suggestions for practical applications.

© 2003 Microchip Technology Inc.

ECCP/CCP Register Listing

	Capture Mode	Compare Mode	PWM Mode
CCPxCON	Select mode	Select mode	Select mode, LSB of duty cycle
CCPRxL	Timer1 capture (LSB)	Timer1 compare (LSB)	MSB of duty cycle
CCPRxH	Timer1 capture (MSB)	Timer1 compare (MSB)	N/A
TRISx	set CCPx pin to input	set CCPx pin to output	set CCPx pin(s) to output(s)
T1CON	Timer1 on, prescaler	Timer1 on, prescaler	N/A
T2CON	N/A	N/A	Timer2 on, prescaler
PR2	N/A	N/A	Timer2 period
PIE1	Timer1 interrupt enable	Timer1 interrupt enable	Timer2 interrupt enable
PIR1	Timer1 interrupt flag	Timer1 interrupt flag	Timer2 interrupt flag
INTCON	global/ peripheral interrupt enable	global/ peripheral interrupt enable	global/ peripheral interrupt enable
PWM1CON(1)	N/A	N/A	set dead band, auto-restart control
ECCPAS(1)	N/A	N/A	auto-shutdown control

Note 1: Only on ECCP module.

© 2003 Microchip Technology Inc.

CAPTURE TIPS ‘N TRICKS

In Capture mode, the 16-bit value of Timer1 is captured in CCPRxH:CCPRxL when an event occurs on pin CCPx. An event is defined as one of the following and is configured by CCPxCON<3:0>:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

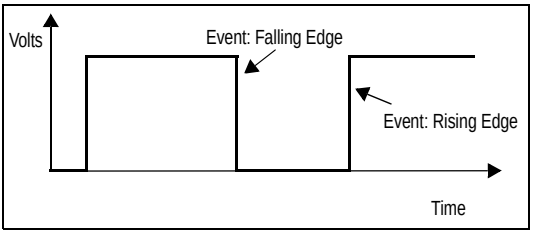
"When Would I Use Capture Mode?"

Capture mode is used to measure the length of time elapsed between two events. An event, in general, is either the rising or falling edge of a signal (see Figure A “Defining Events”).

An example of an application where Capture mode is useful is reading an accelerometer. Accelerometers typically vary the duty cycle of a square wave in proportion to the acceleration acting on a system. By configuring the CCP module in Capture mode, the PIC® microcontroller can measure the duty cycle of the accelerometer with little intervention on the part of the microcontroller firmware. TIP #4 goes into more detail about measuring duty cycle by configuring the CCP module in Capture mode.

© 2003 Microchip Technology Inc.

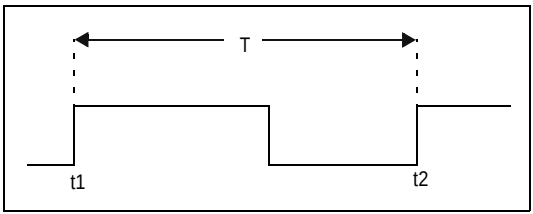
FIGURE A: DEFINING EVENTS



© 2003 Microchip Technology Inc.

TIP #1 Measuring the Period of a Square Wave

FIGURE 1-1: PERIOD



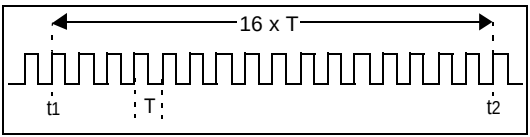
1. Configure control bits CCPxM3:CCPxM0 (CCPxCON<3:0>) to capture every rising edge of the waveform.
2. Configure the Timer1 prescaler so Timer1 with run TMAX**(1)** without overflowing.
3. Enable the CCP interrupt (CCPxIE bit).
4. When a CCP interrupt occurs:
   a) Subtract saved captured time (t1)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

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

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

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

FIGURE 2-1: PERIOD MEASUREMENT



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

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

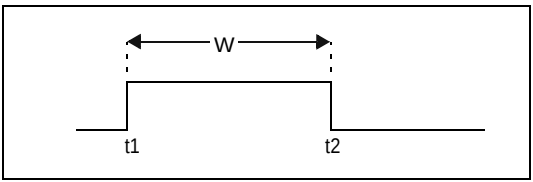
© 2003 Microchip Technology Inc.

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

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

TIP #3 Measuring Pulse Width

FIGURE 3-1: PULSE WIDTH

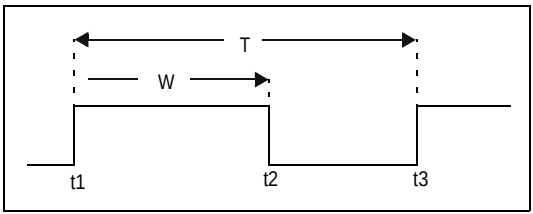


1. Configure control bits CCPxM3:CCPxM0 (CCPxCON<3:0>) to capture every rising edge of the waveform.
2. Configure Timer1 prescaler so that Timer1 will run WMAX without overflowing.
3. Enable the CCP interrupt (CCPxIE bit).
4. When CCP interrupt occurs, save the captured timer value (t1) and reconfigure control bits to capture every falling edge.
5. When CCP interrupt occurs again, subtract saved value (t1) from current captured value (t2) – this result is the pulse width (W).
6. Reconfigure control bits to capture the next rising edge and start process all over again (repeat steps 3 through 6).

© 2003 Microchip Technology Inc.

TIP #4 Measuring Duty Cycle

FIGURE 4-1: DUTY CYCLE



The duty cycle of a waveform is the ratio between the width of a pulse (W) and the period (T). Acceleration sensors, for example, vary the duty cycle of their outputs based on the acceleration acting on a system. The CCP module, configured in Capture mode, can be used to measure the duty cycle of these types of sensors. Here's how:

1. Configure control bits CCPxM3:CCPxM0 (CCPxCON<3:0>) to capture every rising edge of the waveform.
2. Configure Timer1 prescaler so that Timer1 will run TMAX**(1)** without overflowing.
3. Enable the CCP interrupt (CCPxIE bit).
4. When CCP interrupt occurs, save the captured timer value (t1) and reconfigure control bits to capture every falling edge.

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

5. When the CCP interrupt occurs again, subtract saved value (t1) from current captured value (t2) – this result is the pulse width (W).
6. Reconfigure control bits to capture the next rising edge.
7. When the CCP interrupts occurs, subtract saved value (t1) from the current captured value (t3) – this is the period (T) of the waveform.
8. Divide T by W – this result is the Duty Cycle.
9. Repeat steps 4 through 8.

© 2003 Microchip Technology Inc.

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

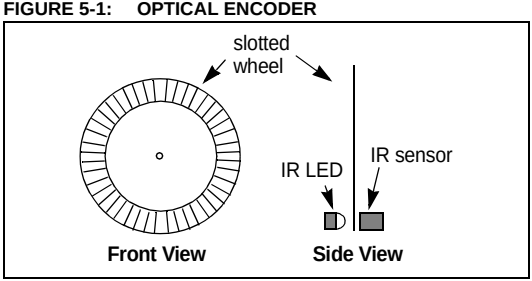
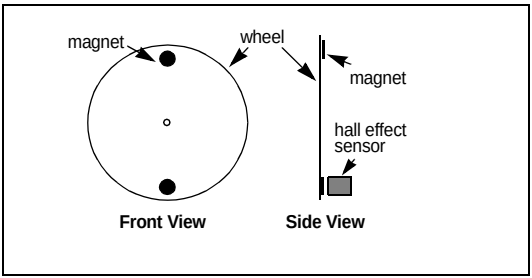


FIGURE 5-2: HALL EFFECT SENSOR



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.

© 2003 Microchip Technology Inc.

FIGURE 5-3: LOW RPM

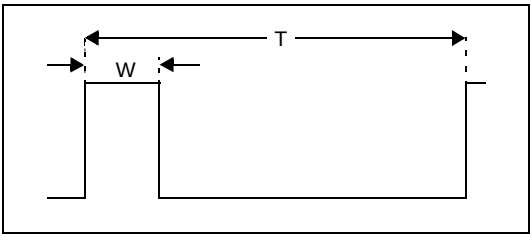
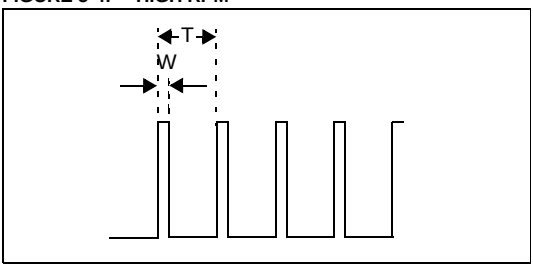


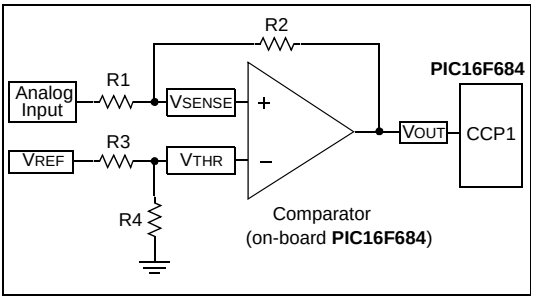
FIGURE 5-4: HIGH RPM



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.

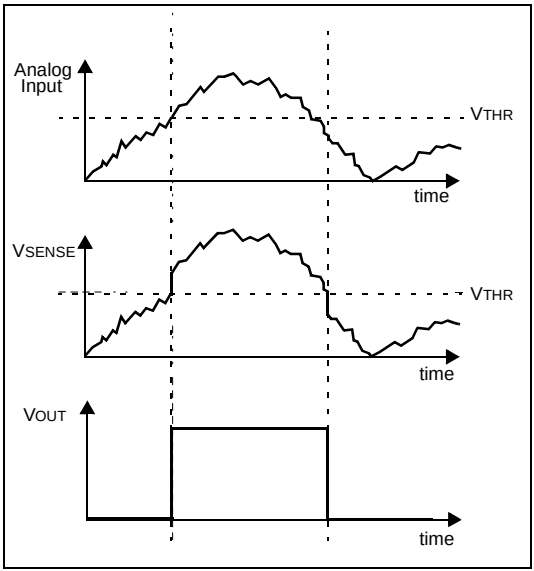
FIGURE 6-1: CIRCUIT



R3 and R4 set the threshold voltage for the comparator. When the analog input reaches the threshold voltage, VOUT 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 VSENSE does when the analog input reaches the threshold voltage.

© 2003 Microchip Technology Inc.

FIGURE 6-2: SIGNAL COMPARISON



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

COMPARE TIPS 'N TRICKS

In Compare mode, the 16-bit CCPRx register value is constantly compared against the TMR1 register pair values. When a match occurs, the CCPx pin is:

- Driven high
- Driven low
- Remains unchanged, or
- Toggles based on the module’s configuration

The action on the pin is determined by control bits CCPxM3:CCPxM0 (CCPxCON<3:0>). A CCP interrupt is generated when a match occurs.

Special Event Trigger

Timer1 is normally not cleared during a CCP interrupt when the CCP module is configured in Compare mode. The only exception to this is when the CCP module is configured in Special Event Trigger mode. In this mode, when Timer1 and CCPRx are equal, the CCPx interrupt is generated, Timer1 is cleared, and an A/D conversion is started (if the A/D module is enabled.)

© 2003 Microchip Technology Inc.

"Why Would I Use Compare Mode?"

Compare mode works much like the timer function on a stopwatch. In the case of a stopwatch, a predetermined time is loaded into the watch and it counts down from that time until zero is reached.

Compare works in the same way with one exception – it counts from zero to the predetermined time. This mode is useful for generating specific actions at precise intervals. A timer could be used to perform the same functionality, however, it would mean preloading the timer each time. Compare mode also has the added benefit of automatically altering the state of the CCPx pin based on the way the module is set up.

© 2003 Microchip Technology Inc.

TIP #7 Periodic Interrupts

Generating interrupts at periodic intervals is a useful technique implemented in many applications. This technique allows the main loop code to run continuously, and then, at periodic intervals, jump to the interrupt service routine to execute specific tasks (i.e., read the ADC). Normally, a timer overflow interrupt is adequate for generating the periodic interrupt. However, sometimes it is necessary to interrupt at intervals that can not be achieved with a timer overflow interrupt. The CCP configured in Compare mode makes this possible by shortening the full 16-bit time period.

Example Problem:

A PIC16F684 running on its 8 MHz internal oscillator needs to be configured so that it updates a LCD exactly 5 times every second.

Step #1: Determine a Timer1 prescaler that allows an overflow at greater than 0.2 seconds

a) Timer1 overflows at: Tosc*4*65536* prescaler

b) For a prescaler of 1:1, Timer1 overflows in 32.8 ms.

c) A prescaler of 8 will cause an overflow at a time greater than 0.2 seconds.
8 x 32.8 ms = 0.25s

© 2003 Microchip Technology Inc.

Step #2: Calculate CCPR1 (CCPR1L and CCPR1H) to shorten the time-out to exactly 0.2 seconds

a) CCPR1 = Interval Time/(Tosc*4*prescaler) = 0.2/(125 ns*4*8) = 5000 = 0xC350

b) Therefore, CCPR1L = 0x50, and CCPR1H = 0xC3

Step #3: Configuring CCP1CON

The CCP module should be configured in Trigger Special Event mode. This mode generates an interrupt when the Timer1 equals the value specified in CCPR1L and Timer1 is automatically cleared**(1)**. For this mode, CCP1CON = ‘b00001011’.

Note 1:	Trigger Special Event mode also starts an A/D conversion if the A/D module is enabled. If this functionality is not desired, the CCP module should be configured in "generate software interrupt-on-match only" mode (i.e., CCP1CON = b’00001010’.) Timer 1 must also be cleared manually during the CCP interrupt.

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

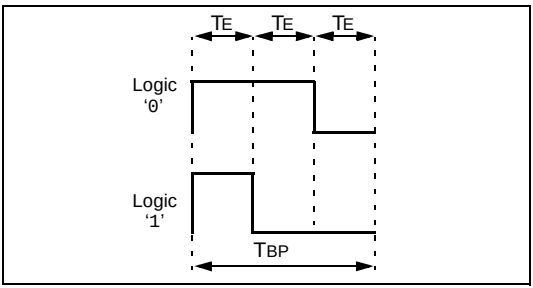
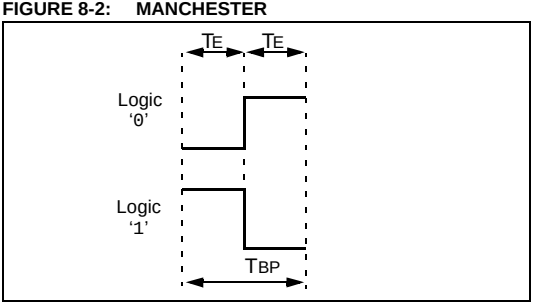


FIGURE 8-2: MANCHESTER



© 2003 Microchip Technology Inc.

FIGURE 8-3: PULSE POSITION MODULATION

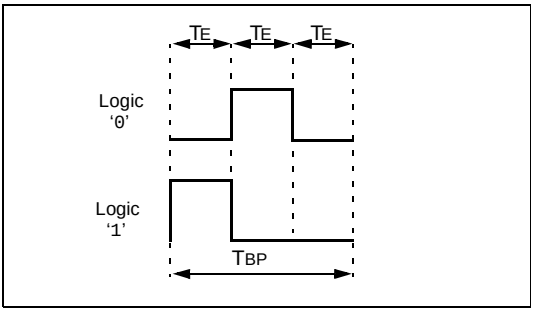
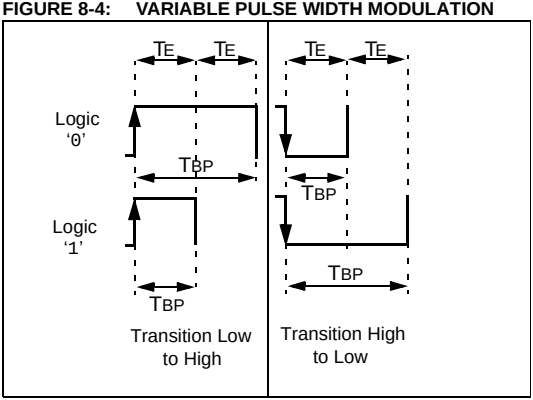


FIGURE 8-4: VARIABLE PULSE WIDTH MODULATION



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.

TE 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 TE, (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).

© 2003 Microchip Technology Inc.

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

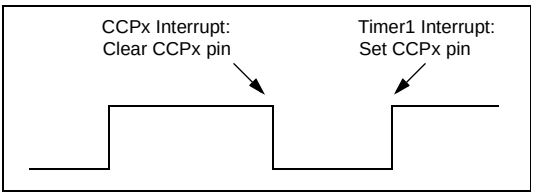
Task	Interval	Priority
Read ADC input 1	20 ms	2
Read ADC input 2	60 ms	1
Update LCD	24 ms	2
Update LED array	36 ms	3
Read Switch	10 ms	1
Dump Data to Serial Port	240 ms	1

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

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.

© 2003 Microchip Technology Inc.

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.

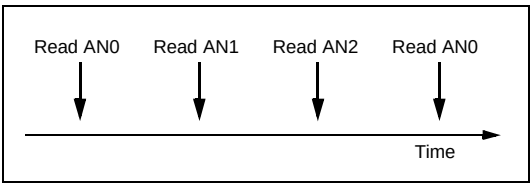
EQUATION 10-1:

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

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

TIP #11 Sequential ADC Reader

FIGURE 11-1: TIMELINE



Trigger Special Event mode (a sub-mode in Compare mode) generates a periodic interrupt in addition to automatically starting an A/D conversion when Timer1 matches CCPRxL and CCPRxH. The following example problem demonstrates how to sequentially read the A/D channels at a periodic interval.

Example

Given the PIC16F684 running on its 8 MHz internal oscillator, configure the microcontroller to sequentially read analog pins AN0, AN1 and AN2 at 30 ms intervals.

Step #1: Determine Timer1 prescaler

a) Timer1 overflows at: TOSC*4*65536* prescaler.

b) For a prescaler of 1:1, the Timer1 overflow occurs in 32.8 ms.

c) This is greater than 30 ms, so a prescaler of 1 is adequate.

© 2003 Microchip Technology Inc.

Step #2: Calculate CCPR1 (CCPR1L and CCPR1H)

a) CCPR1 = Interval Time/(TOSC*4*prescaler) = 0.030/(125 ns*4*1) = 6000 = 0xEA60

b) Therefore, CCPR1L = 0x60, and CCPR1H = 0xEA

Step #3: Configuring CCP1CON

The ECCP module should be configured in Trigger Special Event mode. This mode generates an interrupt when Timer1 equals the value specified in CCPR1. Timer1 is automatically cleared and the GO bit in ADCON0 is automatically set. For this mode, CCP1CON = ‘b00001011’.

Step #4: Add Interrupt Service Routine logic

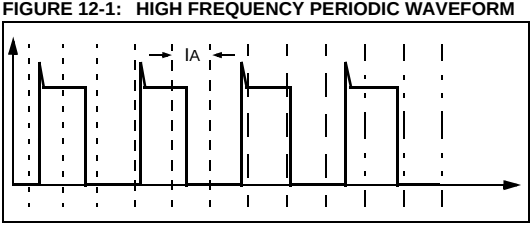
When the ECCP interrupt is generated, select the next A/D pin for reading by altering the ADCON0 register.

© 2003 Microchip Technology Inc.

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.

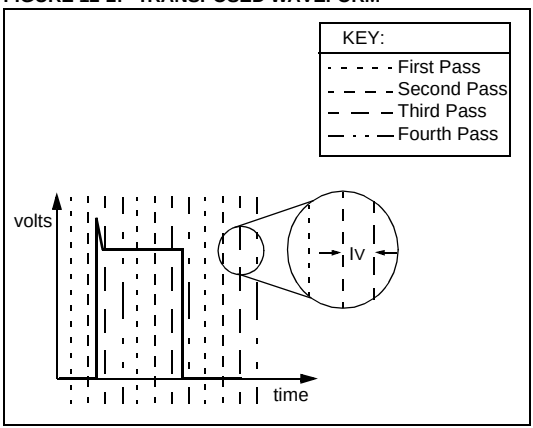
FIGURE 12-1: HIGH FREQUENCY PERIODIC WAVEFORM



© 2003 Microchip Technology Inc.

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 (IV) of 2 μs from 0 μs to 100 μs, although the actual sampling interval (IA) is 22 μs.

© 2003 Microchip Technology Inc.

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:

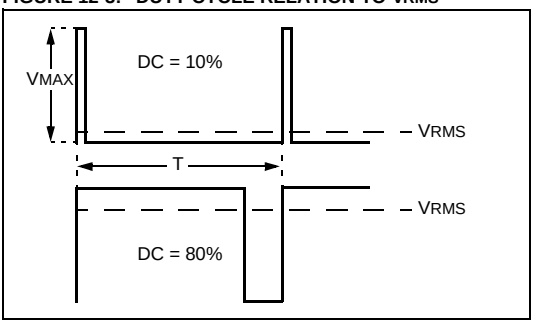
- P1A, P1C active-high; P1B, P1D active-high
- P1A, P1C active-high; P1B, P1D active-low
- P1A, P1C active-low; P1B, P1D active-high
- P1A, P1C active-low; P1B, P1D active-low

© 2003 Microchip Technology Inc.

"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 VRMS



© 2003 Microchip Technology Inc.

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

EQUATION 12-1:

$$V_{RMS} = DC \times V_{MAX}$$

TIP #13 Deciding on PWM Frequency

In general, PWM frequency is application dependent although two general rules-of-thumb hold regarding frequency in all applications. They are:

1. As frequency increases, so does current requirement due to switching losses.
2. Capacitance and inductance of the load tend to limit the frequency response of a circuit.

In low-power applications, it is a good idea to use the minimum frequency possible to accomplish a task in order to limit switching losses. In circuits where capacitance and/or inductance are a factor, the PWM frequency should be chosen based on an analysis of the circuit.

© 2003 Microchip Technology Inc.

Motor Control

PWM is used extensively in motor control due to the efficiency of switched drive systems as opposed to linear drives. An important consideration when choosing PWM frequency for a motor control application is the responsiveness of the motor to changes in PWM duty cycle. A motor will have a faster response to changes in duty cycle at higher frequencies. Another important consideration is the sound generated by the motor. Brushed DC motors will make an annoying whine when driven at frequencies within the audible frequency range (20 Hz – 4 kHz.) In order to eliminate this whine, drive brushed DC motors at frequencies greater than 4 kHz. (Humans can hear frequencies at upwards of 20 kHz, however, the mechanics of the motor winding will typically attenuate motor whine above 4 kHz).

LED and Light Bulbs

PWM is also used in LED and light dimmer applications. Flicker may be noticable with rates below 50 Hz. Therefore, it is generally a good rule to pulse width modulate LEDs and light bulbs at 100 Hz or higher.

TIP #14 Unidirectional Brushed DC Motor Control Using CCP

FIGURE 14-1: BRUSHED DC (BDC) MOTOR CONTROL CIRCUIT

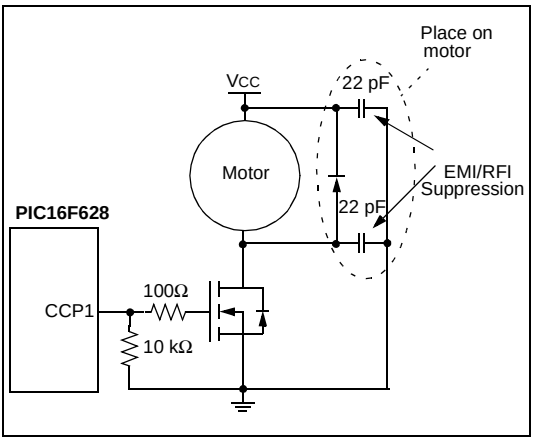


Figure 14-1 shows a unidirectional speed controller circuit for a brushed DC motor. Motor speed is proportional to the duty cycle of the PWM output on the CCP1 pin. The following steps show how to configure the PIC16F628 to generate a 20 kHz PWM with 50% duty cycle. The microcontroller is running on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

© 2003 Microchip Technology Inc.

Step #1: Choose Timer2 prescaler

a) $F_{PWM} = F_{OSC}/((PR2+1)*4*prescaler)$ = 19531 Hz for PR2 = 255 and prescaler of 1

b) This frequency is lower than 20 kHz, therefore a prescaler of 1 is adequate.

Step #2: Calculate PR2

$PR2 = F_{OSC}/(F_{PWM}*4*prescaler) - 1 = 249$

Step #3: Determine CCPR1L and CCP1CON<5:4>

a) CCPR1L:CCP1CON<5:4> = DutyCycle*0x3FF = 0x1FF

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

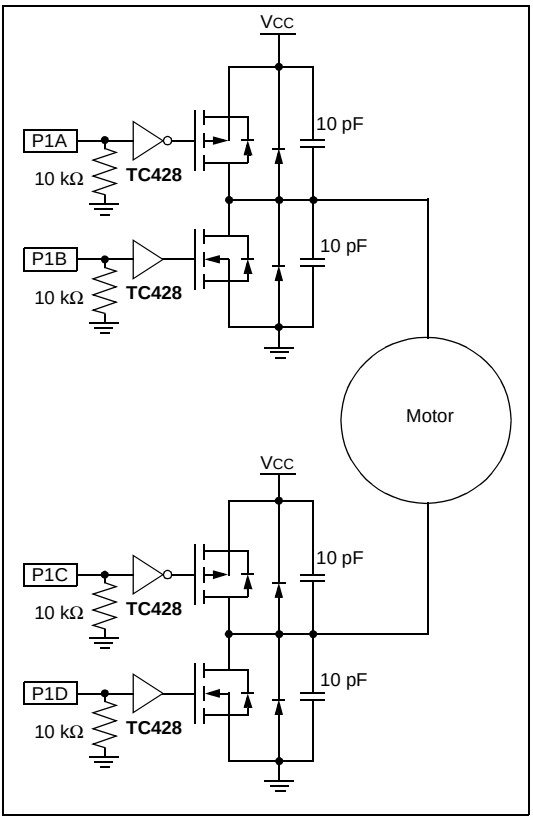
Step #4: Configure CCP1CON

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

© 2003 Microchip Technology Inc.

TIP #15 Bidirectional Brushed DC Motor Control Using ECCP

FIGURE 15-1: FULL-BRIDGE BDC DRIVE CIRCUIT



© 2003 Microchip Technology Inc.

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.

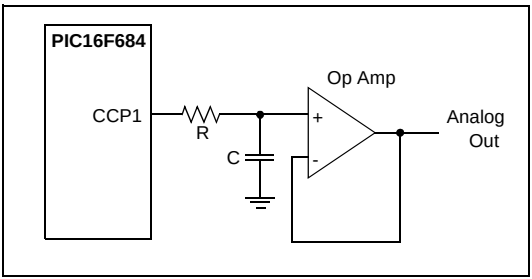
The following table shows the relation between the states of operation, the states of the ECCP pins, and the ECCP Configuration register.

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

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

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

© 2003 Microchip Technology Inc.

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:

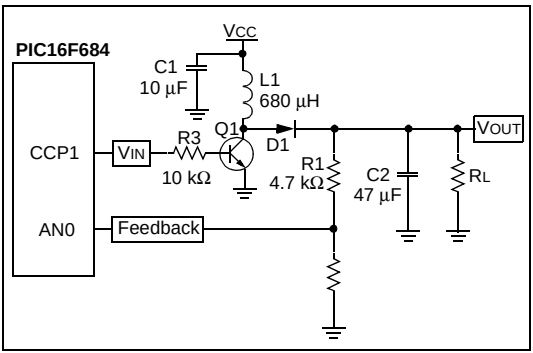
$$Att(dB = -10*\log[1+(2\pi F_{PWM}RC)2]$$

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.

© 2003 Microchip Technology Inc.

TIP #17 Boost Power Supply

FIGURE 17-1: BOOST POWER SUPPLY CIRCUIT



Hardware

Pulse width modulation plays a key role in boost power supply design. Figure 17-1 shows a typical boost circuit. The circuit works by Q1 grounding the inductor (L1) during the high phase of the PWM signal generated by CCP1. This causes an increasing current to flow through L1 while VCC is applied. During the low phase of the PWM signal, the energy stored in L1 flows through D1 to the storage capacitor (C2) and the load. VOUT is related to VIN by Equation 17-1.

Note: Technical Brief TB053 “*Generating High Voltage Using the PIC16C781/782*” provides details on boost power supply design.

© 2003 Microchip Technology Inc.

The first parameter to determine is the duty cycle based upon the input and output voltages. See Equation 17-1.

EQUATION 17-1:

$$\frac{V_{OUT}}{V_{IN}} = \frac{1}{1 - D}$$

Next, the value of the inductor is chosen based on the maximum current required by the load, the switching frequency, and the duty cycle. A function for inductance in terms of load current is given by Equation 17-2, where T is the PWM period, D is the duty cycle, and I_{OUT} is the maximum load current.

EQUATION 17-2:

$$L = \frac{V_{IN}\ (1 - D)\ DT}{2\ I_{OUT}}$$

The value for L is chosen arbitrarily to satisfy this equation given I_{OUT}, a maximum duty cycle of 75% and a PWM frequency in the 10 kHz to 100 kHz range.

© 2003 Microchip Technology Inc.

Using the value chosen for L, the ripple current is calculated using Equation 17-3.

EQUATION 17-3:

$$I_{RIPPLE} = \frac{V_{IN}\ DT}{L}$$

I_{RIPPLE} can not exceed the saturation current for the inductor. If the value for L does produce a ripple current greater than I_{SAT}, a bigger inductor is needed.

Note: All equations above assume a discontinuous current mode.

© 2003 Microchip Technology Inc.

Firmware

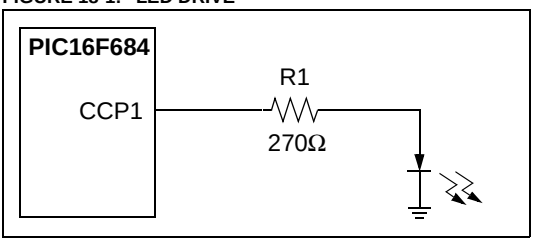
The PWM duty cycle is varied by the microcontroller in order to maintain a stable output voltage over fluctuating load conditions. A firmware implemented PID control loop is used to regulate the duty cycle. Feedback from the boost power supply circuit provides the input to the PID control.

Note: Application Note AN258 “*Low Cost USB Microcontroller Programmer*” provides details on firmware based PID control.

TIP #18 Varying LED Intensity

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

FIGURE 18-1: LED DRIVE



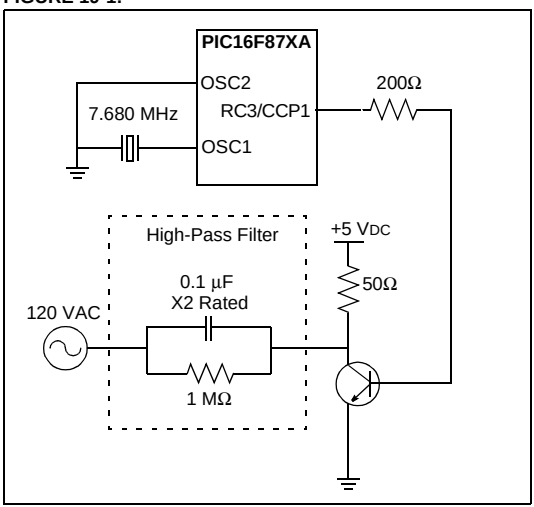
The CCP module, configured in PWM mode, is ideal for varying the intensity of an LED. Adjustments to the intensity of the LED are made by simply varying the duty cycle of the PWM signal driving the LED. This is accomplished by varying the CCPRxL register between 0 and 0xFF.

© 2003 Microchip Technology Inc.

TIP #19 Generating X-10® Carrier Frequency

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

FIGURE 19-1:



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

© 2003 Microchip Technology Inc.

COMBINATION CAPTURE AND COMPARE TIPS

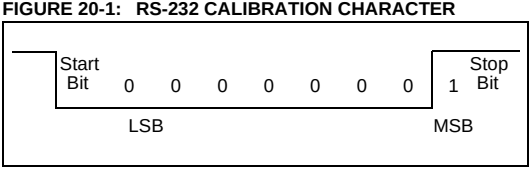
The CCP and ECCP modules can be configured on the fly. Therefore, these modules can perform different functions in the same application provided these functions operate exclusively (not at the same time.) This section will provide examples of using a CCP module in different modes in the same application.

TIP #20 RS-232 Autobaud

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. Autobaud 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 autobaud to work, a known calibration character must be transmitted initially from one device to another. One possible calibration character is show in Figure 19-1. Timing this known character provides the device with the baud rate for all subsequent communications.

FIGURE 20-1: RS-232 CALIBRATION CHARACTER



© 2003 Microchip Technology Inc.

Autobaud 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 (TB).
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 Autobaud for the PIC16C5X Devices*” for more details on autobaud.

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   ;Dleay Tb before
                    ;sending STOP bit
  BSF    TxLine     ;send STOP bit
```


© 2003 Microchip Technology Inc.

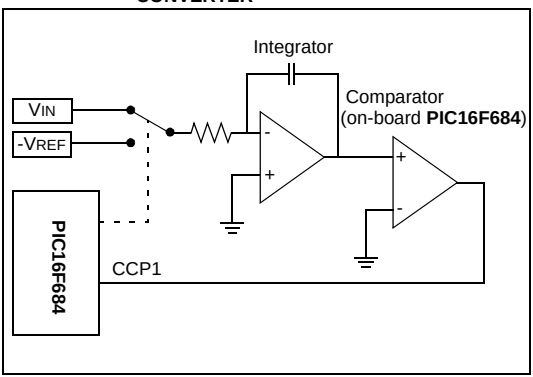
FIGURE 20-2: RECEIVE ROUTINE

```
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.

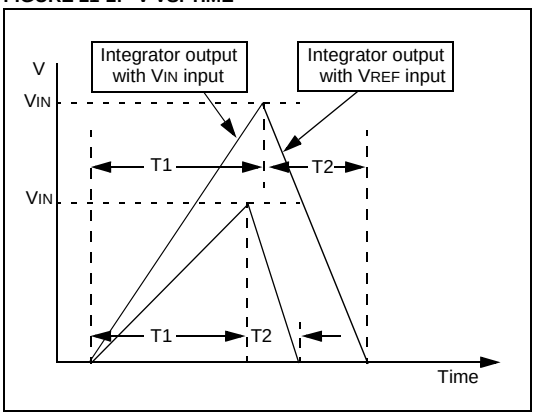
FIGURE 21-1: DUAL-SLOPE ANALOG-TO-DIGITAL CONVERTER



© 2003 Microchip Technology Inc.

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:

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

© 2003 Microchip Technology Inc.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

The graphics in this document are for illustration only. Microchip reserves the right to modify the contents of its development systems.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartShunt and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPIC, Select Mode, SmartSensor, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

Worldwide Sales and Service

AMERICAS

Corporate Office
Tel: 480-792-7200
Technical Support:
480-792-7627

Atlanta
Tel: 770-640-0034

Boston
Tel: 978-692-3848

Chicago
Tel: 630-285-0071

Dallas
Tel: 972-818-7423

Detroit
Tel: 248-538-2250

Kokomo
Tel: 765-864-8360

Los Angeles
Tel: 949-263-1888

Phoenix
Tel: 480-792-7966

San Jose
Tel: 650-215-1444

Toronto
Tel: 905-673-0699

ASIA/PACIFIC

Australia
Tel: 61-2-9868-6733

China-Beijing
Tel:86-10-85282100

China-Chengdu
Tel: 86-28-86766200

China-Fuzhou
Tel: 86-591-7503506

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

China-Shanghai
Tel: 86-21-6275-5700

China-Shenzhen
Tel: 86-755-82901380

China-Shunde
Tel: 86-765-8395507

China-Qingdao
Tel: 86-532-5027355

India
Tel: 91-80-2290061

Japan
Tel: 81-45-471- 6166

Korea
Tel: 82-2-554-7200

Singapore
Tel: 65-6334-8870

Taiwan
Kaohsiung
Tel: 886-7-536-4818

Taiwan
Tel: 886-2-2717-7175

EUROPE

Austria
Tel: 43-7242-2244-399

Denmark
Tel: 45-4420-9895

France
Tel: 33-1-69-53-63-20

Germany
Tel: 49-89-627-144-0

Italy
Tel: 39-0331-742611

United Kingdom
Tel: 44-118-921-5869

11/24/03

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

© 2003 Microchip Technology Inc.

Microchip Technology Inc.
2355 W. Chandler Blvd. • Chandler, AZ 85224 U.S.A.
Phone: 480-792-7200 • Fax: 480-792-9210
www.microchip.com

© 2003, Microchip Technology Inc., 12/03 DS41214A