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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 43x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP Exposed Pad
Supplier Device Package	48-TQFP-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f55k42-i-pt

4.3 Register Definitions: Stack Pointer

REGISTER 4-1: TOSU: TOP OF STACK UPPER BYTE

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TOS<20:16>				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **TOS<20:16>:** Top of Stack Location bits

REGISTER 4-2: TOSH: TOP OF STACK HIGH BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TOS<15:8>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **TOS<15:8>:** Top of Stack Location bits

REGISTER 4-3: TOSL: TOP OF STACK LOW BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TOS<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **TOS<7:0>:** Top of Stack Location bits

REGISTER 9-3: PIR0: PERIPHERAL INTERRUPT REQUEST REGISTER 0

R-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCIF(2)	CRCIF	SCANIF	NVMIF	CSWIF(3)	OSFIF	HLVDIF	SWIF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HS = Bit is set in hardware

bit 7 **IOCIF:** Interrupt-on-Change Interrupt Flag bit(2)
- 1 = Interrupt has occurred
- 0 = Interrupt event has not occurred

bit 6 **CRCIF:** CRC Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 5 **SCANIF:** Memory Scanner Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 4 **NVMIF:** NVM Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 3 **CSWIF:** Clock Switch Interrupt Flag bit(3)
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 2 **OSFIF:** Oscillator Fail Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 1 **HLVDIF:** HLVD Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

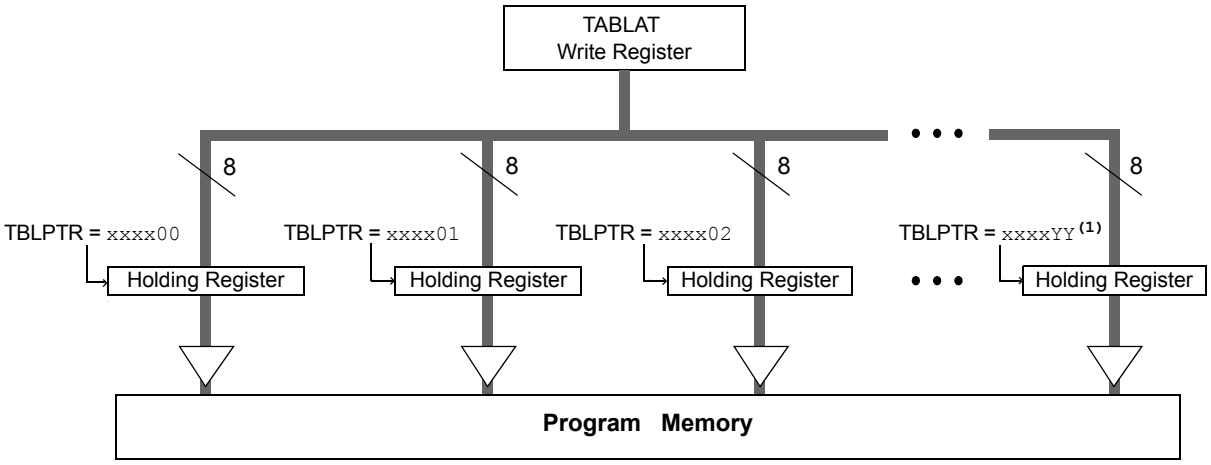
bit 0 **SWIF:** Software Interrupt Flag bit
- 1 = Software Interrupt Flag Enable
- 0 = Software Interrupt Flag Disable

Note 1: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

2: IOCIF is a read-only bit. To clear the interrupt condition, all bits in the IOCxF registers must be cleared.

3: The CSWIF interrupt will not wake the system from Sleep. The system will sleep until another interrupt causes the wake-up.

FIGURE 13-8: TABLE WRITES TO PROGRAM FLASH MEMORY



Note 1: Refer to Table 5-4 for number of holding registers (e.g., YY = 3F for 64 holding registers).

13.1.6.1 Program Flash Memory Write Sequence

The sequence of events for programming an internal program memory location should be:

1. Read appropriate number of bytes into RAM. Refer to Table 5-4 for Write latch size.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the block erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the n-byte block into the holding registers with auto-increment. Refer to Table 5-4 for Write latch size.
7. Set REG<1:0> bits to point to program memory.
8. Clear FREE bit and set WREN bit in NVMCON1 register.
9. Disable interrupts.
10. Execute the unlock sequence (see **Section 13.1.4 "NVM Unlock Sequence"**).
11. WR bit is set in NVMCON1 register.
12. The CPU will stall for the duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Verify the memory (table read).

This procedure will require about 6 ms to update each write block of memory. An example of the required code is given in Example 13-4.

Note:	Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the bytes in the holding registers.

14.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the program memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 14-1: CRC EXAMPLE

Rev. 10-000206A
1/8/2014

CRC-16-ANSI

$x^{16} + x^{15} + x^2 + 1$ (17 bits)

Standard 16-bit representation = 0x8005

```
CRCXORH = 0b10000000
CRCXORL = 0b0000010-
```
(1)

Data Sequence:
0x55, 0x66, 0x77, 0x88

```
DLEN = 0b0111
PLEN = 0b1111
```

Data entered into the CRC:
SHIFTM = 0:

```
01010101 01100110 01110111 10001000
```

SHIFTM = 1:

```
10101010 01100110 11101110 00010001
```

Check Value (ACCM = 1):

SHIFTM = 0: 0x32D6

```
CRCACCH = 0b00110010
CRCACCL = 0b11010110
```

SHIFTM = 1: 0x6BA2

```
CRCACCH = 0b01101011
CRCACCL = 0b10100010
```

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always '1' and will always be treated as a '1' by the CRC for calculating the CRC check value. This bit will be read in software as a '0'.

22.5 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the T2CON register are cleared).
- The diagrams illustrate any clock except FOSC/4 and show clock-sync delays of at least two full cycles for both ON and T2TMR_ers. When using FOSC/4, the clock-sync delay is at least one instruction period for T2TMR_ers; ON applies in the next instruction period.
- ON and T2TMR_ers are somewhat generalized, and clock-sync delays may produce results that are slightly different than illustrated.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in **Section 23.0 "Capture/ Compare/PWM Module"** and **Section 24.0 "Pulse-Width Modulation (PWM)"**. The signals are not a part of the T2TMR module.

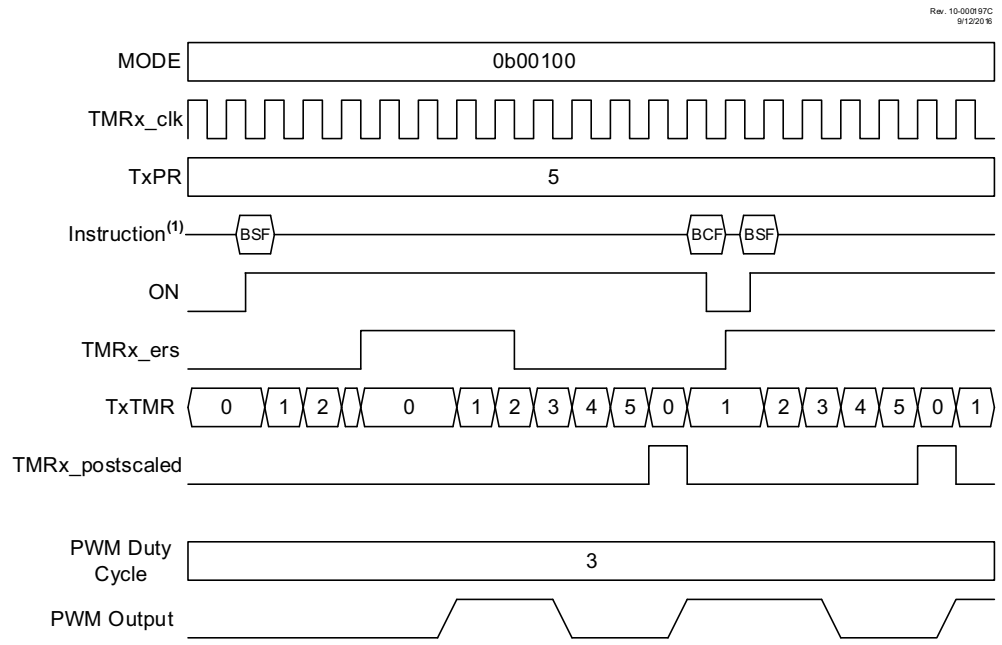
22.5.3 EDGE-TRIGGERED HARDWARE LIMIT MODE

In Hardware Limit mode the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE<4:0> = 00011)
- Reset on rising edge (MODE<4:0> = 0010)
- Reset on falling edge (MODE<4:0> = 00101)

When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two clock delay. Refer to Figure 22-6.

FIGURE 22-6: EDGE TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE=00100)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

REGISTER 22-2: TxRST: TIMER2 EXTERNAL RESET SIGNAL SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	RSEL<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RSEL<4:0>:** Timer2 External Reset Signal Source Selection bits

RSEL<4:0>	T2TMR	TMR4	TMR6
	Reset Source	Reset Source	Reset Source
11111-11001	Reserved	Reserved	Reserved
11000	UART2_tx_edge	UART2_tx_edge	UART2_tx_edge
10111	UART2_rx_edge	UART2_rx_edge	UART2_rx_edge
10110	UART1_tx_edge	UART1_tx_edge	UART1_tx_edge
10101	UART1_rx_edge	UART1_rx_edge	UART1_rx_edge
10100	CLC4_out	CLC4_out	CLC4_out
10011	CLC3_out	CLC3_out	CLC3_out
10010	CLC2_out	CLC2_out	CLC2_out
10001	CLC1_out	CLC1_out	CLC1_out
10000	ZCD_OUT	ZCD_OUT	ZCD_OUT
01111	CMP2OUT	CMP2OUT	CMP2OUT
01110	CMP1OUT	CMP1OUT	CMP1OUT
01101-01100	Reserved	Reserved	Reserved
01011	PWM8OUT	PWM8OUT	PWM8OUT
01010	PWM7OUT	PWM7OUT	PWM7OUT
01001	PWM6OUT	PWM6OUT	PWM6OUT
01000	PWM5OUT	PWM5OUT	PWM5OUT
00111	CCP4OUT	CCP4OUT	CCP4OUT
00110	CCP3OUT	CCP3OUT	CCP3OUT
00101	CCP2OUT	CCP2OUT	CCP2OUT
00100	CCP1OUT	CCP1OUT	CCP1OUT
00011	TMR6 postscaled	TMR6 postscaled	Reserved
00010	TMR4 postscaled	Reserved	TMR4 postscaled
00001	Reserved	T2TMR postscaled	T2TMR postscaled
00000	Pin selected by T2INPPS	Pin selected by T4INPPS	Pin selected by T6INPPS

23.0 CAPTURE/COMPARE/PWM MODULE

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate pulse-width modulated signals of varying frequency and duty cycle.

This family of devices contains four standard Capture/Compare/PWM modules (CCP1, CCP2, CCP3 and CCP4). Each individual CCP module can select the timer source that controls the module. Each module has an independent timer selection which can be accessed using the CxTSEL bits in the CCPTMRS register (Register 23-2). The default timer selection is TMR1 when using Capture/Compare mode and TMR2 when using PWM mode in the CCPx module.

Please note that the Capture/Compare mode operation is described with respect to TMR1 and the PWM mode operation is described with respect to TMR2 in the following sections.

The Capture and Compare functions are identical for all CCP modules.

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

23.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (CCPxCON), a capture input selection register (CCPxCAP) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte).

23.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers 1 through 6 that vary with the selected mode. Various timers are available to the CCP modules in Capture, Compare or PWM modes, as shown in Table 23-1.

TABLE 23-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1, Timer3 or Timer5
Compare	
PWM	Timer2, Timer4 or Timer6

The assignment of a particular timer to a module is determined by the timer to CCP enable bits in the CCPTMRS register (see Register 23-2) All of the modules may be active at once and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time.

23.1.2 OPEN-DRAIN OUTPUT OPTION

When operating in Output mode (the Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

23.2.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE Interrupt Priority bit of the respective PIE register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the respective PIR register following any change in Operating mode.

23.2.4 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1 is clocked by FOSC/4, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep as long as the clock source for Timer1 is active in Sleep.

23.3 Compare Mode

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMRxH:TMRxL register pair. When a match occurs, one of the following events can occur:

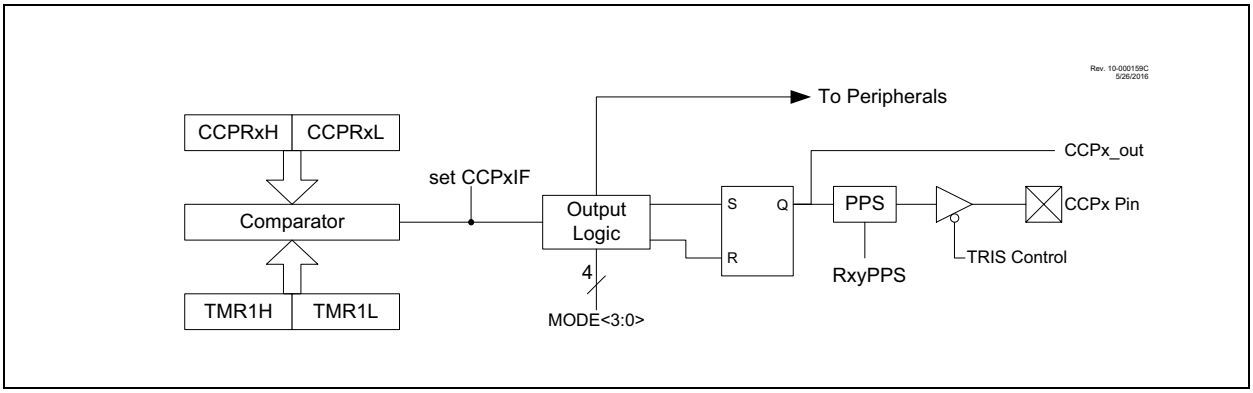
- Toggle the CCPx output, clear TMRx
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Pulse output
- Pulse output, clear TMRx

The action on the pin is based on the value of the MODE<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set, and an ADC conversion can be triggered, if selected.

All Compare modes can generate an interrupt and trigger an ADC conversion. When MODE = `0b0001` or `0b1011`, the CCP resets the TMR register pair.

Figure 23-2 shows a simplified diagram of the compare operation.

FIGURE 23-2: COMPARE MODE OPERATION BLOCK DIAGRAM



25.6.3 PERIOD AND DUTY CYCLE MODE

In Duty Cycle mode, either the duty cycle or period (depending on polarity) of the SMT1_signal can be acquired relative to the SMT clock. The CPW register is updated on a falling edge of the signal, and the CPR register is updated on a rising edge of the signal, along with the SMT1TMR resetting to 0x0001. In addition, the GO bit is reset on a rising edge when the SMT is in Single Acquisition mode. See Figure 25-6 and Figure 25-7.

25.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

25.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMT1CPW and SMT1CPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMT1CPW interrupt is controlled by SMT1PWAIF and SMT1PWAIE bits in the respective PIR and PIE registers. The SMT1CPR interrupt is controlled by the SMT1PRAIF and SMT1PRAIE bits, also located in the respective PIR and PIE registers.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMT1CLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

25.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in **Section 25.1.2 "Period Match interrupt"**, the SMT will also interrupt upon SMT1TMR, matching SMT1PR with its period match limit functionality described in **Section 25.3 "Halt Operation"**. The period match interrupt is controlled by SMT1IF and SMT1IE, located in the respective PIR and PIE registers.

29.4 ZCD Interrupts

An interrupt will be generated upon a change in the ZCD logic output when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in the ZCD for this purpose.

The ZCDIF bit of the respective PIR register will be set when either edge detector is triggered and its associated enable bit is set. The INTP enables rising edge interrupts and the INTN bit enables falling edge interrupts. Both are located in the ZCDCON register. Priority of the interrupt can be changed if the IPEN bit of the INTCON register is set. The ZCD interrupt can be made high or low priority by setting or clearing the ZCDIP bit of the respective IPR register.

To fully enable the interrupt, the following bits must be set:

- ZCDIE bit of the respective PIE register
- INTP bit of the ZCDCON register (for a rising edge detection)
- INTN bit of the ZCDCON register (for a falling edge detection)
- GIE bits of the INTCON0 register

Changing the POL bit can cause an interrupt, regardless of the level of the SEN bit.

The ZCDIF bit of the respective PIR register must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

29.5 Correcting for VCPINV offset

The actual voltage at which the ZCD switches is the reference voltage at the noninverting input of the ZCD op amp. For external voltage source waveforms other than square waves, this voltage offset from zero causes the zero-cross event to occur either too early or too late. When the waveform is varying relative to VSS, then the zero cross is detected too early as the waveform falls and too late as the waveform rises. When the waveform is varying relative to VDD, then the zero cross is detected too late as the waveform rises and too early as the waveform falls. The actual offset time can be determined for sinusoidal waveforms with the corresponding equations shown in Equation 29-2.

EQUATION 29-2: ZCD EVENT OFFSET

When External Voltage Source is relative to VSS:

$$T_{OFFSET} = \frac{\text{asin}\left(\frac{V_{CPINV}}{V_{PEAK}}\right)}{2\pi \bullet Freq}$$

When External Voltage Source is relative to VDD:

$$T_{OFFSET} = \frac{\text{asin}\left(\frac{V_{DD}-V_{CPINV}}{V_{PEAK}}\right)}{2\pi \bullet Freq}$$

This offset time can be compensated for by adding a pull-up or pull-down biasing resistor to the ZCD pin. A pull-up resistor is used when the external voltage source is varying relative to VSS. A pull-down resistor is used when the voltage is varying relative to VDD. The resistor adds a bias to the ZCD pin so that the target external voltage source must go to zero to pull the pin voltage to the VCPINV switching voltage. The pull-up or pull-down value can be determined with the equations shown in Equation 29-3 or Equation 29-4.

EQUATION 29-3: ZCD PULL-UP/DOWN

When External Signal is relative to Vss:

$$R_{PULLUP} = \frac{R_{SERIES}(V_{PULLUP} - V_{CPINV})}{V_{CPINV}}$$

When External Signal is relative to VDD:

$$R_{PULLDOWN} = \frac{R_{SERIES}(V_{CPINV})}{(V_{DD} - V_{CPINV})}$$

TABLE 30-1: MD1CARH/MD1CARL SELECTION MUX CONNECTIONS

MD1CARH			MD1CARL		
CH<4:0>		Connection	CL<4:0>		Connection
11111-10011	31-19	Reserved	11111-10011	31-19	Reserved
10010	18	CLC4OUT	10010	18	CLC4OUT
10001	17	CLC3OUT	10001	17	CLC3OUT
10000	16	CLC2OUT	10000	16	CLC2OUT
01111	15	CLC1OUT	01111	15	CLC1OUT
01110	14	NCO1OUT	01110	14	NCO1OUT
01101-01100	13-12	Reserved	01101-01100	13-12	Reserved
01011	11	PWM8 OUT	01011	11	PWM8 OUT
01010	10	PWM7 OUT	01010	10	PWM7 OUT
01001	9	PWM6 OUT	01001	9	PWM6 OUT
01000	8	PWM5 OUT	01000	8	PWM5 OUT
00111	7	CCP4 OUT	00111	7	CCP4 OUT
00110	6	CCP3 OUT	00110	6	CCP3 OUT
00101	5	CCP2 OUT	00101	5	CCP2 OUT
00100	4	CCP1 OUT	00100	4	CCP1 OUT
00011	3	CLKREF output	00011	3	CLKREF output
00010	2	HFINTOSC	00010	2	HFINTOSC
00001	1	FOSC (system clock)	00001	1	FOSC (system clock)
00000	0	Pin selected by MD1CARHPPS	00000	0	Pin selected by MD1CARLPPS

REGISTER 30-5: MD1SRC: MODULATION SOURCE CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	MS<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **MS<4:0>:** Modulator Source Selection bits(1)
See Table 30-2 for signal list

Note 1: Unused selections provide a zero as the input value.

31.3 Asynchronous Address Mode

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems.

When Asynchronous Address mode is enabled, all data is transmitted and received as 9-bit characters. The 9th bit determines whether the character is an address or data. When the 9th bit is set, the eight Least Significant bits are the address. When the 9th bit is clear, the Least Significant bits are data. In either case, the 9th bit is stored in PERIF when the byte is written to the receive FIFO. When PERIE is also set, the RXIF will be suppressed, thereby suspending DMA transfers allowing software to process the received address.

An address character will enable all receivers that match the address and disable all other receivers. Once a receiver is enabled, all non-address characters will be received until an address character is received that does not match.

31.3.1 ADDRESS MODE TRANSMIT

The UART transmitter is enabled for asynchronous address operation by configuring the following control bits:

- TXEN = 1
- MODE<3:0> = 0100
- UxBRGH:L = desired baud rate
- RxyPPS = code for desired output pin
- ON = 1

Addresses are sent by writing to the UxP1L register. This transmits the written byte with the 9th bit set, which indicates that the byte is an address.

Data is sent by writing to the UxTXB register. This transmits the written byte with the 9th bit cleared, which indicates that the byte is data.

To send data to a particular device on the transmission bus, first transmit the address of the intended device. All subsequent data will be accepted only by that device until an address of another device is transmitted.

Writes to UxP1L take precedence over writes to UxTXB. When both the UxP1L and UxTXB registers are written while the TSR is busy, the next byte to be transmitted will be from UxP1L.

To ensure that all data intended for one device is sent before the address is changed, wait until the TXMTIF bit is high before writing UxP1L with the new address.

31.3.2 ADDRESS MODE RECEIVE

The UART receiver is enabled for asynchronous address operation by configuring the following control bits:

- RXEN = 1
- MODE<3:0> = 0100
- UxBRGH:L = desired baud rate
- RXPPS = code for desired input pin
- Input pin ANSEL bit = 0
- UxP2L = receiver address
- UxP3L = address mask
- ON = 1

In Address mode, no data will be transferred to the input FIFO until a valid address is received. This is the default state. Any of the following conditions will cause the UART to revert to the default state:

- ON = 0
- RXEN = 0
- Received address does not match

When a character with the 9th bit set is received, the Least Significant eight bits of that character will be qualified by the values in the UxP2L and UxP3L registers.

The byte is XOR'd with UxP2L then AND'd with UxP3L. A match occurs when the result is 0h, in which case, the unaltered received character is stored in the receive FIFO, thereby setting the UxRXIF interrupt bit. The 9th bit is stored in the corresponding PERIF bit, identifying this byte as an address.

An address match also enables the receiver for all data such that all subsequent characters without the 9th bit set will be stored in the receive FIFO.

When the 9th bit is set and a match does not occur, the character is not stored in the receive FIFO and all subsequent data is ignored.

The UxP3L register mask allows a range of addresses to be accepted. Software can then determine the sub-address of the range by processing the received address character.

The Master process is started by writing the PID to the UxP1L register when UxP2 is '0' and the UART is idle. The UxTXIF will not be set in this case. Only the six Least Significant bits of UxP1L are used in the PID transmission.

The two Most Significant bits of the transmitted PID are PID parity bits. PID<6> is the exclusive-or of PID bits 0,1,2,and 4. PID<7> is the inverse of the exclusive-or of PID bits 1,3,4,and 5.

The UART calculates and inserts these bits in the serial stream.

Writing UxP1L automatically clears the UxTXCHK and UxRXCHK registers and generates the Break, delimiter bit, Sync character (55h), and PID transmission portion of the transaction. The data portion of the transaction that follows, if there is one, is a Slave process. See **Section 31.5.2 "LIN Slave Mode"** for more details of that process. The Master receives it's own PID when RXEN is set. Software performs the Slave process corresponding to the PID that was sent and received. Attempting to write UxP1L before an active master process is complete will not succeed. Instead, the TXWRE bit will be set.

31.5.2 LIN SLAVE MODE

LIN Slave mode is configured by the following settings:

- MODE<3:0> = 1011
- TXEN = 1
- RXEN = 1
- UxP2 = Number of data bytes to transmit
- UxP3 = Number of data bytes to receive
- UxBRGH:L = Value to achieve default baud rate
- TXPOL = 0 (for high Idle state)
- STP = desired Stop bits selection
- C0EN = desired checksum mode
- RxyPPS = TX pin selection code
- TX pin TRIS control = 0
- ON = 1

The Slave process starts upon detecting a Break on the RX pin. The Break clears the UxTXCHK, UxRXCHK, UxP2, and UxP3 registers. At the end of the Break, the auto-baud circuity is activated and the baud rate is automatically set using the Sync character following the Break. The character following the Sync character is received as the PID code and is saved in the receive FIFO. The UART computes the two PID parity bits from the six Least Significant bits of the PID. If either parity bit does not match the corresponding bit of the received PID code, the PERIF flag is set and saved at the same FIFO location as the PID code. The UxRXIF bit is set indicating that the PID is available.

Software retrieves the PID by reading the UxRXB register and determines the Slave process to execute from that. The checksum method, number of data bytes, and whether to send or receive data, is defined by software according to the PID code.

31.5.2.1 LIN Slave Receiver

When the Slave process is a receiver, the software performs the following tasks:

- UxP3 register is written with a value equal to the number of data bytes to receive.
- C0EN bit is set or cleared to select the appropriate checksum. This must be completed before the Start bit of the checksum byte is received.
- Each byte of the process response is read from UxRXB when UxRXIF is set.

The UART updates the checksum on each received byte. When the last data byte is received, the computed checksum total is stored in the UxRXCHK register. The next received byte is saved in the receive FIFO and added with the value in UxRXCHK. The result of this addition is not accessible. However, if the result is not all '1's, the CERIF bit in the UxERRIR is set. The CERIF flag persists until cleared by software. Software needs to read UxRXB to remove the checksum byte from the FIFO, but the byte can be discarded if not needed for any other purpose.

After the checksum is received, the UART ignores all activity on the RX pin until a Break starts the next transaction.

31.5.2.2 LIN Slave Transmitter

When the Slave process is a transmitter, software performs the following tasks in the order shown:

- UxP2 register is written with a value equal to the number of bytes to transmit. This will enable TXIF flag which is disabled when UxP2 is '0'.
- C0EN bit is set or cleared to select the appropriate checksum
- Inter-byte delay is performed
- Each byte of the process response is written to UxTXB when UxTXIF is set

The UART accumulates the checksum as each byte is written to UxTXB. After the last byte is written, the UART stores the calculated checksum in the UxTXCHK register and transmits the inverted result as the last byte in the response.

The TXIF flag is disabled when UxP2 bytes have been written. Any writes to UxTXB that exceed the UxP2 count will be ignored and set the TXWRE flag in the UxFIFO register.

32.9 Register definitions: SPI

REGISTER 32-1: SPIxINTF: SPI INTERRUPT FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	R/W/HS-0/0	R/W/HS-0/0	U-0
SRMTIF	TCZIF	SOSIF	EOSIF	—	RXOIF	TXUIF	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0' HS = Bit can be set by hardware

bit 7 **SRMTIF**: Shift Register Empty Interrupt Flag bit

Slave mode:

This bit is ignored

Master mode:

1 = The data transfer is complete

0 = Either no data transfers have occurred or a data transfer is in progress

bit 6 **TCZIF**: Transfer Counter is Zero Interrupt Flag bit

1 = The transfer counter (as defined by BMODE in Register 32-7, TCNTH/L, and TWIDTH) has decremented to zero

0= No interrupt pending

bit 5 **SOSIF**: Start of Slave Select Interrupt Flag bit

1 = SS(in) transitioned from false to true

0 = No interrupt pending

bit 4 **EOSIF**: End of Slave Select Interrupt Flag bit

1 = SS(in) transitioned from true to false

0 = No interrupt pending

bit 3 **Unimplemented**: Read as '0'

bit 2 **RXOIF**: Receiver Overflow Interrupt Flag bit

1 = Data transfer completed when RXBF = 1 (edge triggered) and RXR = 1

0 = No interrupt pending

bit 1 **TXUIF**: Transmitter Underflow Interrupt Flag bit

1 = Slave Data transfer started when TXBE = 1 and TXR = 1

0 = No interrupt pending

bit 0 **Unimplemented**: Read as '0'

36.4 ADC Charge Pump

The ADC module has a dedicated charge pump which can be controlled through the ADCP register (Register 36-36). The primary purpose of the charge pump is to supply a constant voltage to the gates of transistor devices in the A/D converter, signal and reference input pass-gates, to prevent degradation of transistor performance at low operating voltage.

The charge pump can be enabled by setting the CPON bit in the ADC register. Once enabled, the pump will undergo a start-up time to stabilize the charge pump output. Once the output stabilizes and is ready for use, the CPRDY bit of the ADCP register will be set.

36.5 Capacitive Voltage Divider (CVD) Features

The ADC module contains several features that allow the user to perform a relative capacitance measurement on any ADC channel using the internal ADC sample and hold capacitance as a reference. This relative capacitance measurement can be used to implement capacitive touch or proximity sensing applications. Figure 36-6 shows the basic block diagram of the CVD portion of the ADC module.

FIGURE 36-6: HARDWARE CAPACITIVE VOLTAGE DIVIDER BLOCK DIAGRAM

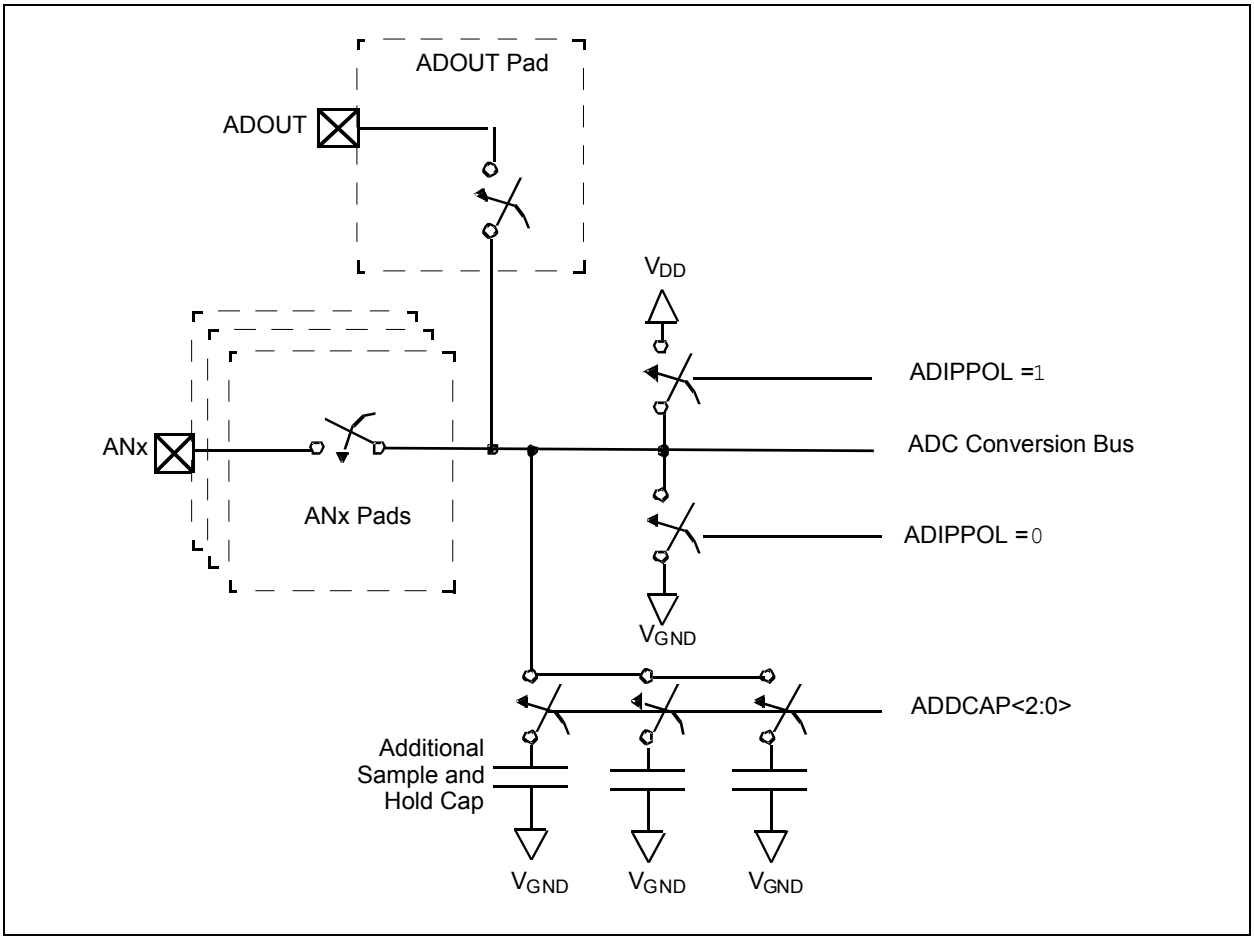


FIGURE 44-12: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

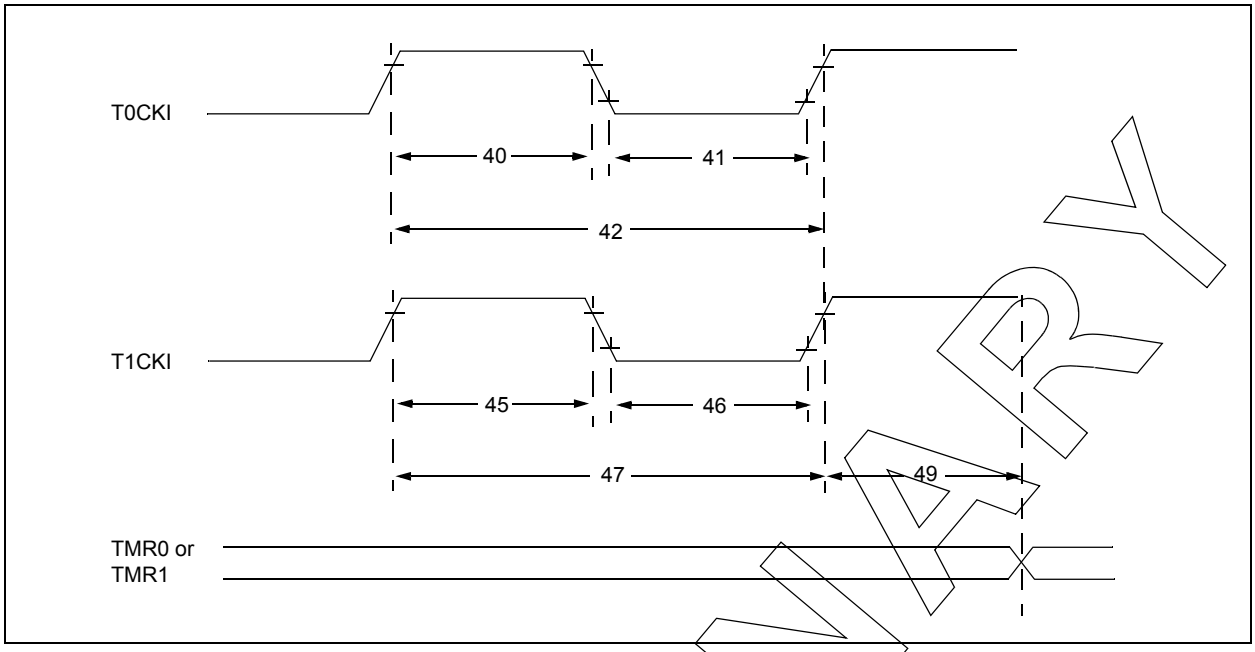


TABLE 44-21: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
40*	TT0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period		Greater of: 20 or TCY + 40 / N	—	—	ns	N = prescale value
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	Greater of: 30 or TCY + 40 / N	—	—	ns	N = prescale value
			Asynchronous	60	—	—	ns	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

* These parameters are characterized but not tested.
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 44-23: SPI MODE REQUIREMENTS (MASTER MODE)

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	TSCK	SCK Cycle Time (2x Prescaled)	61	—	—	ns	Transmit only mode
			—	16(1)	—	MHz	
			95	—	—	ns	Full duplex mode
			—	10(1)	—	MHz	
SP70*	TSSL2SCH, TSSL2SCL	SDO to SCK↓ or SCK↑ input	TSCK	—	—	ns	FST = 0
			0	—	—	ns	FST = 1
SP71*	TSCH	SCK output high time	0.5 TSCK - 12	—	0.5 TSCK + 12	ns	
SP72*	TSCL	SCK output low time	0.5 TSCK - 12	—	0.5 TSCK + 12	ns	
SP73*	TDIV2SCH, TDIV2SCL	Setup time of SDI data input to SCK edge	85	—	—	ns	
SP74*	TSCH2DIL, TSCL2DIL	Hold time of SDI data input to SCK edge	0	—	—	ns	
		Hold time of SDI data input to final SCK	0.5 TSCK	—	—	ns	CKE = 0, SMP = 1
SP75*	TDOR	SDO data output rise time	—	10	25	ns	CL = 50 pF
SP76*	TDOF	SDO data output fall time	—	10	25	ns	CL = 50 pF
SP78*	TSCR	SCK output rise time	—	10	25	ns	CL = 50 pF
SP79*	TSCF	SCK output fall time	—	10	25	ns	CL = 50 pF
SP80*	TSCH2DOV, TSCL2DOV	SDO data output valid after SCK edge	- 15	—	15	ns	CL = 20 pF
SP81*	TDOV2SCH, TDOV2SCL	SDO data output valid to first SCK edge	TSCK - 10	—	—	ns	CL = 20 pF CKE = 1
SP82*	TSSL2DOV	SDO data output valid after $\overline{SS}$↓ edge	—	—	50	ns	CL = 20 pF
SP83*	TSCH2SSH, TSCL2SSH	$\overline{SS}$ ↑ after last SCK edge	0.5 TSCK - 10	—	—	ns	
SP84*	TSSH2SSL	SS↑ to SS↓ edge	0.5 TSCK - 10	—	—	ns	

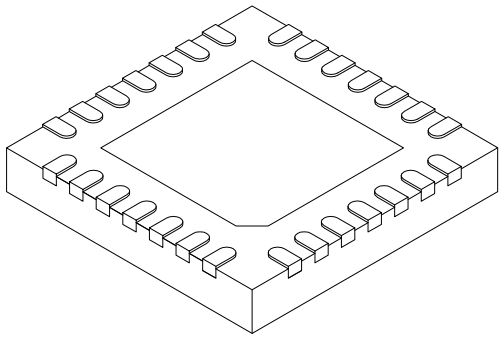
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: SPIxCON1.SMP bit must be set and the slew rate control must be disabled on the clock and data pins (clear the corresponding bits in SLRCONx register) for SPI to operate over 4 MHz.

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN]
With 0.55 mm Terminal Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105C Sheet 2 of 2