

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

-XF

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	15
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1847-e-mv

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

TABLE OF CONTENTS

1.0	Device Overview	9
2.0	Enhanced Mid-Range CPU	
3.0	Memory Organization	
4.0	Device Configuration	
5.0	Oscillator Module (With Fail-Safe Clock Monitor)	
6.0	Reference Clock Module	
7.0	Resets	
8.0	Interrupts	
9.0	Power-Down Mode (Sleep)	
10.0	Watchdog Timer	
11.0	Data EEPROM and Flash Program Memory Control	
12.0	I/O Ports	117
13.0	Interrupt-On-Change	
14.0	Fixed Voltage Reference (FVR)	
15.0	Temperature Indicator Module	
16.0	Analog-to-Digital Converter (ADC) Module	
17.0	Digital-to-Analog Converter (DAC) Module	
18.0	SR Latch	
19.0	Comparator Module	
20.0	Timer0 Module	
21.0	Timer1 Module with Gate Control	
22.0	Timer2/4/6 Modules	
23.0	Data Signal Modulator	
24.0	Capture/Compare/PWM Modules	
25.0	Master Synchronous Serial Port (MSSP1 and MSSP2) Module	
26.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	
27.0	Capacitive Sensing Module	
28.0	In-Circuit Serial Programming™ (ICSP™)	
29.0	Instruction Set Summary	
30.0	Electrical Specifications	
31.0	DC and AC Characteristics Graphs and Charts	
32.0	Development Support	
33.0	Packaging Information	
The M	Microchip Web Site	
Custo	omer Change Notification Service	
Custo	omer Support	
Produ	uct Identification System	

PIC16(L)F1847





4.0 DEVICE CONFIGURATION

Device Configuration consists of Configuration Word 1 and Configuration Word 2, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h and Configuration Word 2 at 8008h.

Note:	The DEBUG bit in Configuration Word is
	managed automatically by device
	development tools including debuggers
	and programmers. For normal device
	operation, this bit should be maintained as
	a '1'.

11.6 Write/Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM or program memory should be verified (see Example 11-6) to the desired value to be written. Example 11-6 shows how to verify a write to EEPROM.

EXAMPLE 11-6: EEPROM WRITE/VERIFY

BANKSEI	EEDATL		;
MOVF	EEDATL, W	I	;EEDATL not changed
			;from previous write
BSF	EECON1, R	RD	;YES, Read the
			;value written
XORWF	EEDATL, W	I	;
BTFSS	STATUS, Z		;Is data the same
GOTO	WRITE_ERR	ł.	;No, handle error
:			;Yes, continue

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	122
LATA	LATA7	LATA6	—	LATA4	LATA3	LATA2	LATA1	LATA0	121
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		175
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	120
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120
WPUA	_	—	WPUA5	_		_	_	—	121

TABLE 12-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Legend: — = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

TABLE 12-3: SUMMARY OF CONFIGURATION WORD ASSOCIATED WITH PORTA

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
	13:8	—	—	FCMEN	IESO	CLKOUTEN	BORE	N<1:0>	CPD	40
CONFIGT	7:0	CP	MCLRE	PWRTE	WDTE<1:0>			FOSC<2:0>		40

Legend: — = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

14.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference (FVR) is a stable voltage reference, independent of VDD, with a nominal output level (VFVR) of 1.024V. The output of the FVR can be configured to supply a reference voltage to the following:

- · ADC input channel
- · Comparator positive input
- · Comparator negative input

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

14.1 Independent Gain Amplifier

The output of the FVR supplied to the peripherals, (listed above), is routed through a programmable gain amplifier. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 16.0** "**Analog-to-Digital Converter** (**ADC**) **Module**" for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the comparator modules. Reference **Section 19.0 "Comparator Module**" for additional information.

To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the Buffer Gain Selection bits.

14.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 30.0** "**Electrical Specifications**" for the minimum delay requirement.

FIGURE 14-1: VOLTAGE REFERENCE BLOCK DIAGRAM



24.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

- 1. Disable the CCPx pin output driver by setting the associated TRIS bit.
- 2. Load the PRx register with the PWM period value.
- Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
- Load the CCPRxL register and the DCxBx bits of the CCPxCON register, with the PWM duty cycle value.
- 5. Configure and start Timer2/4/6:
 - Select the Timer2/4/6 resource to be used for PWM generation by setting the CxTSEL<1:0> bits in the CCPTMRS register.
 - Clear the TMRxIF interrupt flag bit of the PIRx register. See Note below.
 - Configure the TxCKPS bits of the TxCON register with the Timer prescale value.
 - Enable the Timer by setting the TMRxON bit of the TxCON register.
- 6. Enable PWM output pin:
 - Wait until the Timer overflows and the TMRxIF bit of the PIRx register is set. See Note below.
 - Enable the CCPx pin output driver by clearing the associated TRIS bit.
- **Note:** In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

24.3.3 TIMER2/4/6 TIMER RESOURCE

The PWM standard mode makes use of one of the 8-bit Timer2/4/6 timer resources to specify the PWM period.

Configuring the CxTSEL<1:0> bits in the CCPTMRS register selects which Timer2/4/6 timer is used.

24.3.4 PWM PERIOD

The PWM period is specified by the PRx register of Timer2/4/6. The PWM period can be calculated using the formula of Equation 24-1.

EQUATION 24-1: PWM PERIOD

 $PWM Period = [(PRx) + 1] \bullet 4 \bullet TOSC \bullet$ (TMRx Prescale Value)

Note 1: Tosc = 1/Fosc

When TMRx is equal to PRx, the following three events occur on the next increment cycle:

- TMRx is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

Note: The Timer postscaler (see Section 22.1 "Timer2/4/6 Operation") is not used in the determination of the PWM frequency.

24.3.5 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPRxL register and DCxB<1:0> bits of the CCPxCON register. The CCPRxL contains the eight MSbs and the DCxB<1:0> bits of the CCPxCON register contain the two LSbs. CCPRxL and DCxB<1:0> bits of the CCPxCON register can be written to at any time. The duty cycle value is not latched into CCPRxH until after the period completes (i.e., a match between PRx and TMRx registers occurs). While using the PWM, the CCPRxH register is read-only.

Equation 24-2 is used to calculate the PWM pulse width.

Equation 24-3 is used to calculate the PWM duty cycle ratio.

EQUATION 24-2: PULSE WIDTH

$$Pulse Width = (CCPRxL:CCPxCON < 5:4>) \bullet$$

TOSC • (TMRx Prescale Value)

EQUATION 24-3: DUTY CYCLE RATIO

 $Duty Cycle Ratio = \frac{(CCPRxL:CCPxCON < 5:4>)}{4(PRx + 1)}$

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMRx register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2/4/6 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH and 2-bit latch, then the CCPx pin is cleared (see Figure 24-4).

24.4.2 FULL-BRIDGE MODE

In Full-Bridge mode, all four pins are used as outputs. An example of Full-Bridge application is shown in Figure 24-10.

In the Forward mode, pin CCPx/PxA is driven to its active state, pin PxD is modulated, while PxB and PxC will be driven to their inactive state as shown in Figure 24-11.

In the Reverse mode, PxC is driven to its active state, pin PxB is modulated, while PxA and PxD will be driven to their inactive state as shown Figure 24-12.

PxA, PxB, PxC and PxD outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the PxA, PxB, PxC and PxD pins as outputs.

FIGURE 24-10: EXAMPLE OF FULL-BRIDGE APPLICATION



24.4.5 PROGRAMMABLE DEAD-BAND DELAY MODE

In Half-Bridge applications where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on, and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (*shoot-through current*) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Figure 24-16 for illustration. The lower seven bits of the associated PWMxCON register (Figure 24-4) sets the delay period in terms of microcontroller instruction cycles (TcY or 4 Tosc).

FIGURE 24-16: EXAMPLE OF HALF-BRIDGE PWM OUTPUT



FIGURE 24-17: EXAMPLE OF HALF-BRIDGE APPLICATIONS







25.2.1 SPI MODE REGISTERS

The MSSPx module has five registers for SPI mode operation. These are:

- MSSPx STATUS register (SSPxSTAT)
- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Control Register 3 (SSPxCON3)
- MSSPx Data Buffer register (SSPxBUF)
- MSSPx Address register (SSPxADD)
- MSSPx Shift register (SSPxSR) (Not directly accessible)

SSPxCON1 and SSPxSTAT are the control and STATUS registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

In SPI master mode, SSPxADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in Section 25.7 "Baud Rate Generator".

SSPxSR is the shift register used for shifting data in and out. SSPxBUF provides indirect access to the SSPxSR register. SSPxBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPxSR and SSPxBUF together create a buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

PIC16(L)F1847



25.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSPx module configured as an I^2C Slave in 10-bit Addressing mode.

Figure 25-20 is used as a visual reference for this description.

This is a step by step process of what must be done by slave software to accomplish I^2C communication.

- 1. Bus starts Idle.
- Master sends Start condition; S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- Master sends matching high address with R/W bit clear; UA bit of the SSPxSTAT register is set.
- 4. Slave sends ACK and SSPxIF is set.
- 5. Software clears the SSPxIF bit.
- 6. Software reads received address from SSPxBUF clearing the BF flag.
- 7. Slave loads low address into SSPxADD, releasing SCLx.
- 8. Master sends matching low address byte to the Slave; UA bit is set.

Note: Updates to the SSPxADD register are not allowed until after the ACK sequence.

9. Slave sends ACK and SSPxIF is set.

Note: If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

- 10. Slave clears SSPxIF.
- 11. Slave reads the received matching address from SSPxBUF clearing BF.
- 12. Slave loads high address into SSPxADD.
- 13. Master clocks a data <u>byte</u> to the slave and clocks out the slaves ACK on the 9th SCLx pulse; SSPxIF is set.
- 14. If SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
- 15. Slave clears SSPxIF.
- 16. Slave reads the received byte from SSPxBUF clearing BF.
- 17. If SEN is set the slave sets CKP to release the SCLx.
- 18. Steps 13-17 repeat for each received byte.
- 19. Master sends Stop to end the transmission.

25.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCLx line is held low are the same. Figure 25-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 25-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

- 26.1.2.8 Asynchronous Reception Setup:
- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 26.3 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- 4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. If 9-bit reception is desired, set the RX9 bit.
- 6. Enable reception by setting the CREN bit.
- 7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 8. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the 9th data bit.
- 9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.
- 10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

26.1.2.9 9-bit Address Detection Mode Setup

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 26.3 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- 3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. Enable 9-bit reception by setting the RX9 bit.
- 6. Enable address detection by setting the ADDEN bit.
- 7. Enable reception by setting the CREN bit.
- The RCIF interrupt flag bit will be set when a character with the 9th bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 9. Read the RCSTA register to get the error flags. The 9th data bit will always be set.
- 10. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
- 11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
- 12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.



FIGURE 26-5: ASYNCHRONOUS RECEPTION

26.4.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for Synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART.

26.4.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see Section 26.4.1.3 "Synchronous Master Transmission"), except in the case of the Sleep mode. If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- 1. The first character will immediately transfer to the TSR register and transmit.
- 2. The second word will remain in TXREG register.
- 3. The TXIF bit will not be set.
- After the first character has been shifted out of TSR, the TXREG register will transfer the second character to the TSR and the TXIF bit will now be set.
- 5. If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.
- 26.4.2.2 Synchronous Slave Transmission Setup:
- 1. Set the SYNC and SPEN bits and clear the CSRC bit.
- 2. Clear the ANSEL bit for the CK pin (if applicable).
- 3. Clear the CREN and SREN bits.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. If 9-bit transmission is desired, set the TX9 bit.
- 6. Enable transmission by setting the TXEN bit.
- 7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
- 8. Start transmission by writing the Least Significant eight bits to the TXREG register.

TABLE 26-9: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON0	RXDTSEL	SDO1SEL	SS1SEL	P2BSEL	CCP2SEL	P1DSEL	P1CSEL	CCP1SEL	118
APFCON1	—	_	_	_	_	_	_	TXCKSEL	118
BAUDCON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	298
INTCON	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	88
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	93
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	297
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	126
TXREG	EUSART Transmit Data Register								289*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	296

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Synchronous Slave Transmission. * Page provides register information.

27.5 Timer Resources

To measure the change in frequency of the capacitive sensing oscillator, a fixed time base is required. For the period of the fixed time base, the capacitive sensing oscillator is used to clock either Timer0 or Timer1. The frequency of the capacitive sensing oscillator is equal to the number of counts in the timer divided by the period of the fixed time base.

27.6 Fixed Time Base

To measure the frequency of the capacitive sensing oscillator, a fixed time base is required. Any timer resource or software loop can be used to establish the fixed time base. It is up to the end user to determine the method in which the fixed time base is generated.

Note:	The fixed time base can not be generated
	by the timer resource that the capacitive
	sensing oscillator is clocking.

27.6.1 TIMER0

To select Timer0 as the timer resource for the CPS module:

- Set the T0XCS bit of the CPSCON0 register.
- Clear the TMR0CS bit of the OPTION_REG register.

When Timer0 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer0. Refer to **Section 20.0 "Timer0 Module**" for additional information.

27.6.2 TIMER1

To select Timer1 as the timer resource for the CPS module, set the TMR1CS<1:0> of the T1CON register to '11'. When Timer1 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer1. Because the Timer1 module has a gate control, developing a time base for the frequency measurement can be simplified by using the Timer0 overflow flag.

It is recommended that the Timer0 overflow flag, in conjunction with the Toggle mode of the Timer1 Gate, be used to develop the fixed time base required by the software portion of the CPS module. Refer to **Section 21.12 "Timer1 Gate Control Register**" for additional information.

TABLE 27-2: TIMER1 ENABLE FUNCTION

TMR10N	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	On
1	1	Count Enabled by input

27.7 Software Control

The software portion of the CPS module is required to determine the change in frequency of the capacitive sensing oscillator. This is accomplished by the following:

- Setting a fixed time base to acquire counts on Timer0 or Timer1.
- Establishing the nominal frequency for the capacitive sensing oscillator.
- Establishing the reduced frequency for the capacitive sensing oscillator due to an additional capacitive load.
- Set the frequency threshold.

27.7.1 NOMINAL FREQUENCY (NO CAPACITIVE LOAD)

To determine the nominal frequency of the capacitive sensing oscillator:

- Remove any extra capacitive load on the selected CPSx pin.
- At the start of the fixed time base, clear the timer resource.
- At the end of the fixed time base save the value in the timer resource.

The value of the timer resource is the number of oscillations of the capacitive sensing oscillator for the given time base. The frequency of the capacitive sensing oscillator is equal to the number of counts on in the timer divided by the period of the fixed time base.

27.7.2 REDUCED FREQUENCY (ADDITIONAL CAPACITIVE LOAD)

The extra capacitive load will cause the frequency of the capacitive sensing oscillator to decrease. To determine the reduced frequency of the capacitive sensing oscillator:

- Add a typical capacitive load on the selected CPSx pin.
- Use the same fixed time base as the nominal frequency measurement.
- At the start of the fixed time base, clear the timer resource.
- At the end of the fixed time base save the value in the timer resource.

The value of the timer resource is the number of oscillations of the capacitive sensing oscillator with an additional capacitive load. The frequency of the capacitive sensing oscillator is equal to the number of counts on in the timer divided by the period of the fixed time base. This frequency should be less than the value obtained during the nominal frequency measurement.

27.7.3 FREQUENCY THRESHOLD

The frequency threshold should be placed midway between the value of nominal frequency and the reduced frequency of the capacitive sensing oscillator. Refer to Application Note AN1103, "*Software Handling for Capacitive Sensing*" (DS01103) for more detailed information on the software required for CPS module.

Note:	For more information on general capacitive sensing refer to Application Notes:	
	 AN1101, "Introduction to Capacitive Sensing" (DS01101) 	
 AN1102, "Layout and Physical Design Guidelines for Capacitive 		

7.9 Operation during Clean

Sensing" (DS01102)

27.8 Operation during Sleep

The capacitive sensing oscillator will continue to run as long as the module is enabled, independent of the part being in Sleep. In order for the software to determine if a frequency change has occurred, the part must be awake. However, the part does not have to be awake when the timer resource is acquiring counts.

Note: Timer0 does not operate when in Sleep, and therefore cannot be used for capacitive sense measurements in Sleep.

30.0 ELECTRICAL SPECIFICATIONS

30.1 Absolute Maximum Ratings^(†)

Ambient temperature under bias	40°C to +125°C
Voltage on pins with respect to Vss	05 C 10 + 150 C
on VDD pin	
PIC16F1847	-0.3V to +6.5V
PIC16LF1847	-0.3V to +4.0V
on MCLR pin	-0.3V to +9.0V
on all other pins	to (VDD + 0.3V)
Maximum current	
on Vss pin ⁽¹⁾	
$-40^{\circ}C \leq TA \leq +85^{\circ}C$	170 mA
$-40^{\circ}C \leq TA \leq +125^{\circ}C$	70 mA
on VDD pin ⁽¹⁾	
-40°C \leq Ta \leq +85°C	170 mA
$-40^{\circ}C \leq TA \leq +125^{\circ}C$	70 mA
on any I/O pin	±25 mA
Clamp current, Ik (VPIN < 0 or VPIN > VDD)	±20 mA

Note 1: Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characterizations, see Table 30-6: "Thermal Characteristics" to calculate device specifications.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

30.4 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

- 1. TppS2ppS
- 2. TppS

Т			
F	Frequency	Т	Time
Lowerc	ase letters (pp) and their meanings:		
рр			
сс	CCP1	OSC	OSC1
ck	CLKOUT	rd	RD
CS	CS	rw	RD or WR
di	SDIx	SC	SCKx
do	SDO	SS	SS
dt	Data in	t0	ТОСКІ
io	I/O PORT	t1	T1CKI
mc	MCLR	wr	WR
Upperc	ase letters and their meanings:		
S			
F	Fall	Р	Period
Н	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 30-4: LOAD CONDITIONS



32.0 DEVELOPMENT SUPPORT

The PIC[®] microcontrollers and dsPIC[®] digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
- MPLAB[®] IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB C Compiler for Various Device Families
 - HI-TECH C[®] for Various Device Families
 - MPASM[™] Assembler
 - MPLINK[™] Object Linker/ MPLIB[™] Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- · Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
 - MPLAB ICD 3
 - PICkit[™] 3 Debug Express
- Device Programmers
 - PICkit[™] 2 Programmer
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

32.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows[®] operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - In-Circuit Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- · A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- · High-level source code debugging
- · Mouse over variable inspection
- Drag and drop variables from source to watch windows
- · Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- · Debug using:
 - Source files (C or assembly)
 - Mixed C and assembly
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

33.2 Package Details

The following sections give the technical details of the packages.

18-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



Units		INCHES		
Dimension	n Limits	MIN	NOM	MAX
Number of Pins	N	18		
Pitch	е	.100 BSC		
Top to Seating Plane	А	-	-	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	_	-
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.880	.900	.920
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	С	.008	.010	.014
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	_	_	.430

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.

3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.

4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-007B