



Welcome to [E-XFL.COM](https://www.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.

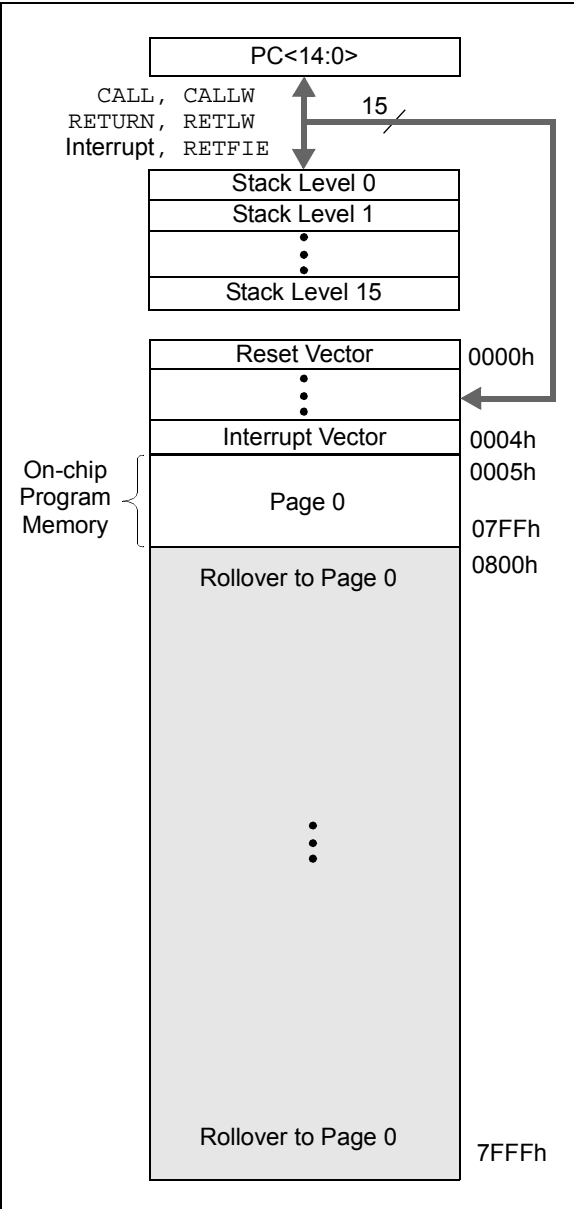
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	14-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	14-TSSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1823-i-st">https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1823-i-st</a>

# PIC12(L)F1822/16(L)F1823

**FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC12(L)F1822/16(L)F1823**



### 3.1.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

#### 3.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

#### EXAMPLE 3-1: RETLW INSTRUCTION

```
constants
    BRW          ;Add Index in W to
                  ;program counter to
                  ;select data

    RETLW DATA0  ;Index0 data
    RETLW DATA1  ;Index1 data
    RETLW DATA2
    RETLW DATA3

my_function
    ;... LOTS OF CODE...
    MOVLW        DATA_INDEX
    CALL constants
    ;... THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

# PIC12(L)F1822/16(L)F1823

## 8.5.1 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, interrupt-on-change and external INT pin interrupts.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 8-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF <sup>(1)</sup>
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      **GIE:** Global Interrupt Enable bit  
1 = Enables all active interrupts  
0 = Disables all interrupts
- bit 6      **PEIE:** Peripheral Interrupt Enable bit  
1 = Enables all active peripheral interrupts  
0 = Disables all peripheral interrupts
- bit 5      **TMR0IE:** Timer0 Overflow Interrupt Enable bit  
1 = Enables the Timer0 interrupt  
0 = Disables the Timer0 interrupt
- bit 4      **INTE:** INT External Interrupt Enable bit  
1 = Enables the INT external interrupt  
0 = Disables the INT external interrupt
- bit 3      **IOCF:** Interrupt-on-Change Enable bit  
1 = Enables the interrupt-on-change  
0 = Disables the interrupt-on-change
- bit 2      **TMR0IF:** Timer0 Overflow Interrupt Flag bit  
1 = TMR0 register has overflowed  
0 = TMR0 register did not overflow
- bit 1      **INTF:** INT External Interrupt Flag bit  
1 = The INT external interrupt occurred  
0 = The INT external interrupt did not occur
- bit 0      **IOCF:** Interrupt-on-Change Interrupt Flag bit<sup>(1)</sup>  
1 = When at least one of the interrupt-on-change pins changed state  
0 = None of the interrupt-on-change pins have changed state

**Note 1:** The IOCF Flag bit is read-only and cleared when all the Interrupt-on-Change flags in the IOCAF register have been cleared by software.

# PIC12(L)F1822/16(L)F1823

## 8.5.2 PIE1 REGISTER

The PIE1 register contains the interrupt enable bits, as shown in Register 8-2.

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

### REGISTER 8-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit  
1 = Enables the Timer1 Gate Acquisition interrupt  
0 = Disables the Timer1 Gate Acquisition interrupt
- bit 6 **ADIE:** A/D Converter (ADC) Interrupt Enable bit  
1 = Enables the ADC interrupt  
0 = Disables the ADC interrupt
- bit 5 **RCIE:** USART Receive Interrupt Enable bit  
1 = Enables the USART receive interrupt  
0 = Disables the USART receive interrupt
- bit 4 **TXIE:** USART Transmit Interrupt Enable bit  
1 = Enables the USART transmit interrupt  
0 = Disables the USART transmit interrupt
- bit 3 **SSP1IE:** Synchronous Serial Port (MSSP) Interrupt Enable bit  
1 = Enables the MSSP interrupt  
0 = Disables the MSSP interrupt
- bit 2 **CCP1IE:** CCP1 Interrupt Enable bit  
1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt
- bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit  
1 = Enables the Timer2 to PR2 match interrupt  
0 = Disables the Timer2 to PR2 match interrupt
- bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit  
1 = Enables the Timer1 overflow interrupt  
0 = Disables the Timer1 overflow interrupt

# PIC12(L)F1822/16(L)F1823

**TABLE 18-1: SRCLK FREQUENCY TABLE**

SRCLK	Divider	Fosc = 32 MHz	Fosc = 20 MHz	Fosc = 16 MHz	Fosc = 4 MHz	Fosc = 1 MHz
111	512	62.5 kHz	39.0 kHz	31.3 kHz	7.81 kHz	1.95 kHz
110	256	125 kHz	78.1 kHz	62.5 kHz	15.6 kHz	3.90 kHz
101	128	250 kHz	156 kHz	125 kHz	31.25 kHz	7.81 kHz
100	64	500 kHz	313 kHz	250 kHz	62.5 kHz	15.6 kHz
011	32	1 MHz	625 kHz	500 kHz	125 kHz	31.3 kHz
010	16	2 MHz	1.25 MHz	1 MHz	250 kHz	62.5 kHz
001	8	4 MHz	2.5 MHz	2 MHz	500 kHz	125 kHz
000	4	8 MHz	5 MHz	4 MHz	1 MHz	250 kHz

**REGISTER 18-1: SRCON0: SR LATCH CONTROL 0 REGISTER**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/S-0/0	R/S-0/0
SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR
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	S = Bit is set only

bit 7	<b>SRLEN:</b> SR Latch Enable bit 1 = SR latch is enabled 0 = SR latch is disabled
bit 6-4	<b>SRCLK&lt;2:0&gt;:</b> SR Latch Clock Divider bits 000 = Generates a 1 Fosc wide pulse every 4th Fosc cycle clock 001 = Generates a 1 Fosc wide pulse every 8th Fosc cycle clock 010 = Generates a 1 Fosc wide pulse every 16th Fosc cycle clock 011 = Generates a 1 Fosc wide pulse every 32nd Fosc cycle clock 100 = Generates a 1 Fosc wide pulse every 64th Fosc cycle clock 101 = Generates a 1 Fosc wide pulse every 128th Fosc cycle clock 110 = Generates a 1 Fosc wide pulse every 256th Fosc cycle clock 111 = Generates a 1 Fosc wide pulse every 512th Fosc cycle clock
bit 3	<b>SRQEN:</b> SR Latch Q Output Enable bit <b>If SRLEN = 1:</b> 1 = Q is present on the SRQ pin 0 = External Q output is disabled <b>If SRLEN = 0:</b> SR latch is disabled
bit 2	<b>SRNQEN:</b> SR Latch $\bar{Q}$ Output Enable bit <b>If SRLEN = 1:</b> 1 = $\bar{Q}$ is present on the SRnQ pin 0 = External $\bar{Q}$ output is disabled <b>If SRLEN = 0:</b> SR latch is disabled
bit 1	<b>SRPS:</b> Pulse Set Input of the SR Latch bit <sup>(1)</sup> 1 = Pulse set input for 1 Q-clock period 0 = No effect on set input.
bit 0	<b>SRPR:</b> Pulse Reset Input of the SR Latch bit <sup>(1)</sup> 1 = Pulse reset input for 1 Q-clock period 0 = No effect on reset input.

**Note 1:** Set only, always reads back '0'.

## 22.1 Timer2 Operation

The clock input to the Timer2 modules is the system instruction clock ( $F_{osc}/4$ ).

TMR2 increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 22.2 “Timer2 Interrupt”**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- MCLR Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

<b>Note:</b> TMR2 is not cleared when T2CON is written.
---

## 22.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (TMR2-to-PR2 match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE of the PIE1 register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS<3:0>, of the T2CON register.

## 22.3 Timer2 Output

The unscaled output of TMR2 is available primarily to the CCP1 module, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSP1 module operating in SPI mode. Additional information is provided in **Section 25.1 “Master SSP (MSSP1) Module Overview”**

## 22.4 Timer2 Operation During Sleep

The Timer2 timers cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

# PIC12(L)F1822/16(L)F1823

---

## 24.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral which 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 one Enhanced Capture/Compare/PWM module (ECCP1).

The Full-Bridge ECCP module has four available I/O pins, while the Half-Bridge ECCP module only has two. See Table 24-1.

**TABLE 24-1: PWM RESOURCES**

Device Name	ECCP1
PIC12(L)F1822	Enhanced PWM Half-Bridge
PIC16(L)F1823	Enhanced PWM Full-Bridge

# PIC12(L)F1822/16(L)F1823

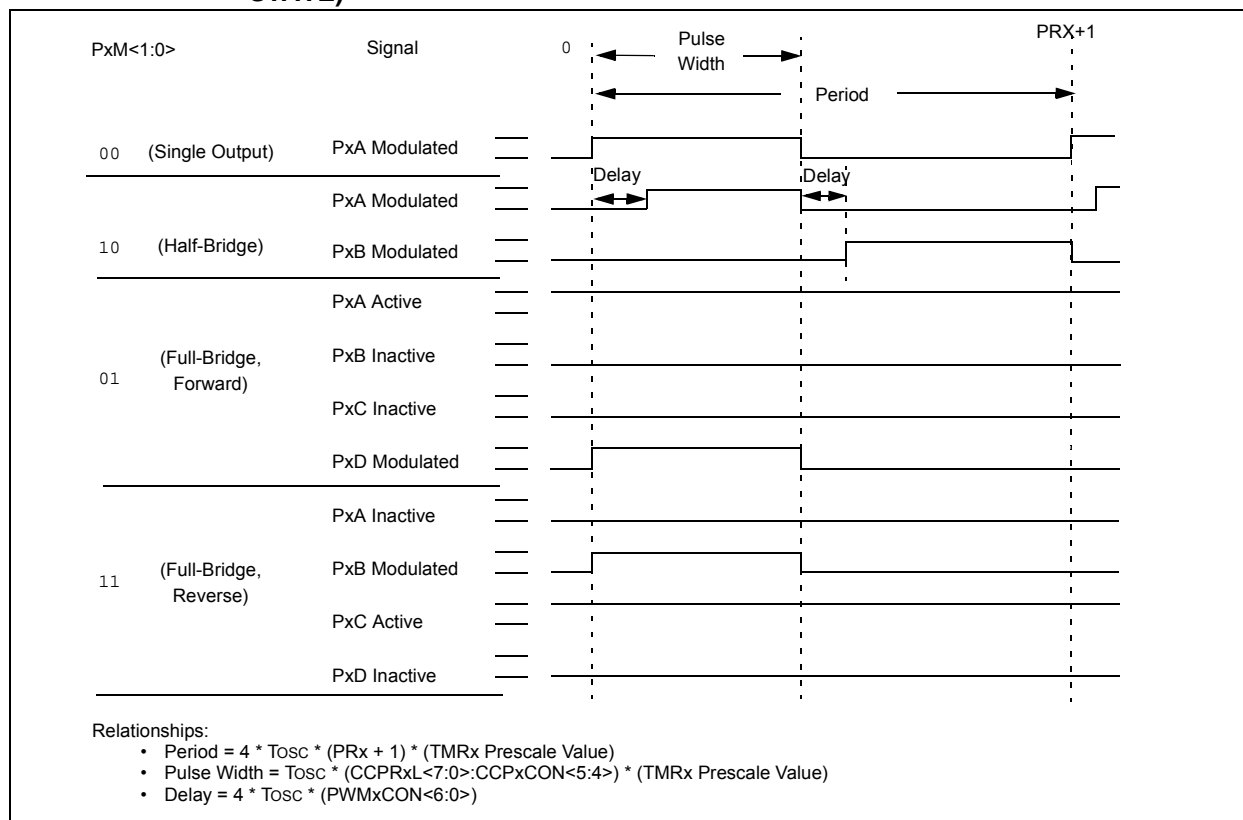
**TABLE 24-9: EXAMPLE PIN ASSIGNMENTS FOR VARIOUS PWM ENHANCED MODES**

ECCP Mode	P1M<1:0>	CCP1/P1A	P1B	P1C <sup>(2)</sup>	P1D <sup>(2)</sup>
Single	00	Yes <sup>(1)</sup>	Yes <sup>(1)</sup>	Yes <sup>(1)</sup>	Yes <sup>(1)</sup>
Half-Bridge	10	Yes	Yes	No	No
Full-Bridge, Forward <sup>(2)</sup>	01	Yes	Yes	Yes	Yes
Full-Bridge, Reverse <sup>(2)</sup>	11	Yes	Yes	Yes	Yes

**Note 1:** PWM Steering enables outputs in Single mode.

**2:** PIC16(L)F1823 only.

**FIGURE 24-6: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)**





---

- Figure 25-2 is a block diagram of the I<sup>2</sup>C interface module in Master mode. Figure 25-3 is a diagram of the I<sup>2</sup>C interface module in Slave mode.

---



# PIC12(L)F1822/16(L)F1823

---

## 25.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSP1CON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSP1IF interrupt is set.

Figure 25-19 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSP1STAT is set; SSP1IF is set if interrupt-on-Start detect is enabled.
3. Master sends matching address with  $\overline{R/W}$  bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSP1IF interrupt is generated.
4. Slave software clears SSP1IF.
5. Slave software reads ACKTIM bit of SSP1CON3 register, and  $\overline{R/W}$  and D/A of the SSP1STAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSP1BUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to  $\overline{ACK}$  or not  $\overline{ACK}$  and sets ACKDT bit of the SSP1CON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the  $\overline{ACK}$  value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSP1IF after the  $\overline{ACK}$  if the  $\overline{R/W}$  bit is set.
11. Slave software clears SSP1IF.
12. Slave loads value to transmit to the master into SSP1BUF setting the BF bit.

**Note:** SSP1BUF cannot be loaded until after the  $\overline{ACK}$ .

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an  $\overline{ACK}$  value on the 9th SCL pulse.
15. Slave hardware copies the  $\overline{ACK}$  value into the ACKSTAT bit of the SSP1CON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not  $\overline{ACK}$  the slave releases the bus, allowing the master to send a Stop and end the communication.

**Note:** Master must send a not  $\overline{ACK}$  on the last byte to ensure that the slave releases the SCL line to receive a Stop.

## 26.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH, SPBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Table 26-3 contains the formulas for determining the baud rate. Example 26-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 26-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

### EXAMPLE 26-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64(SPBRGH:SPBRGL + 1)}$$

Solving for SPBRGH:SPBRGL:

$$X = \frac{\frac{F_{OSC}}{\text{Desired Baud Rate}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

# PIC12(L)F1822/16(L)F1823

---

## 26.3.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRGL register pair. After the ABDOVF has been set, the counter continues to count until the fifth rising edge is detected on the RX pin. Upon detecting the fifth RX edge, the hardware will set the RCIF interrupt flag and clear the ABDEN bit of the BAUDCON register. The RCIF flag can be subsequently cleared by reading the RCREG register. The ABDOVF flag of the BAUDCON register can be cleared by software directly.

To terminate the auto-baud process before the RCIF flag is set, clear the ABDEN bit then clear the ABDOVF bit of the BAUDCON register. The ABDOVF bit will remain set if the ABDEN bit is not cleared first.

## 26.3.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 26-7), and asynchronously if the device is in Sleep mode (Figure 26-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

## 26.3.3.1 Special Considerations

### Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be 10 or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

### Oscillator Start-up Time

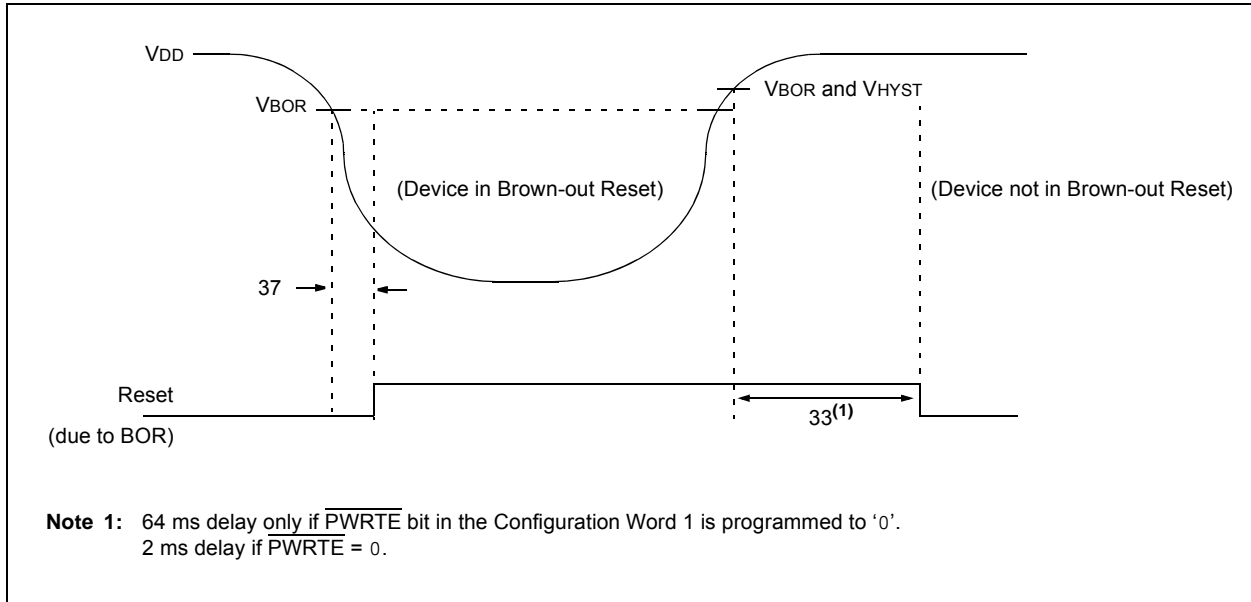
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

### WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

**FIGURE 30-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS**



# PIC12(L)F1822/16(L)F1823

**TABLE 30-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2	—	—	$\mu\text{s}$	
31	TWDTLP	Watchdog Timer Time-out Period	12	16	20	ms	$V_{DD} = 3.3\text{V}-5\text{V}$ , 1:16 Prescaler used
32	TOST	Oscillator Start-up Timer Period <sup>(1), (2)</sup>	—	1024	—	Tosc	
33*	TPWRT	Power-up Timer Period, $\overline{\text{PWRTE}} = 0$	40	65	140	ms	
34*	TIOZ	I/O high-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.0	$\mu\text{s}$	
35	VBOR	Brown-out Reset Voltage	2.55 1.80	2.7 1.9	2.85 2.11	V	BORV = 0 BORV = 1
36*	VHYS	Brown-out Reset Hysteresis	20	35	75	mV	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
37*	TBORDC	Brown-out Reset DC Response Time	1	3	35	$\mu\text{s}$	$V_{DD} \leq V_{BOR}$

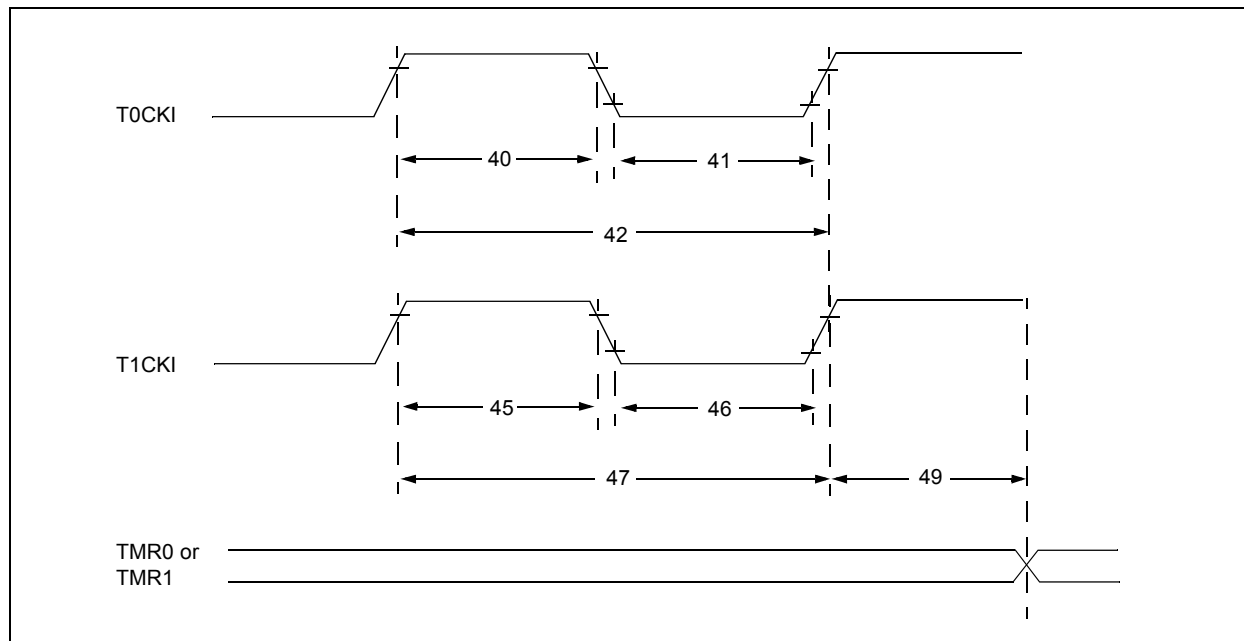
\* These parameters are characterized but not tested.

† 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:** By design, the oscillators start-up (OST) counts the first 1024 cycles, independent of frequency.

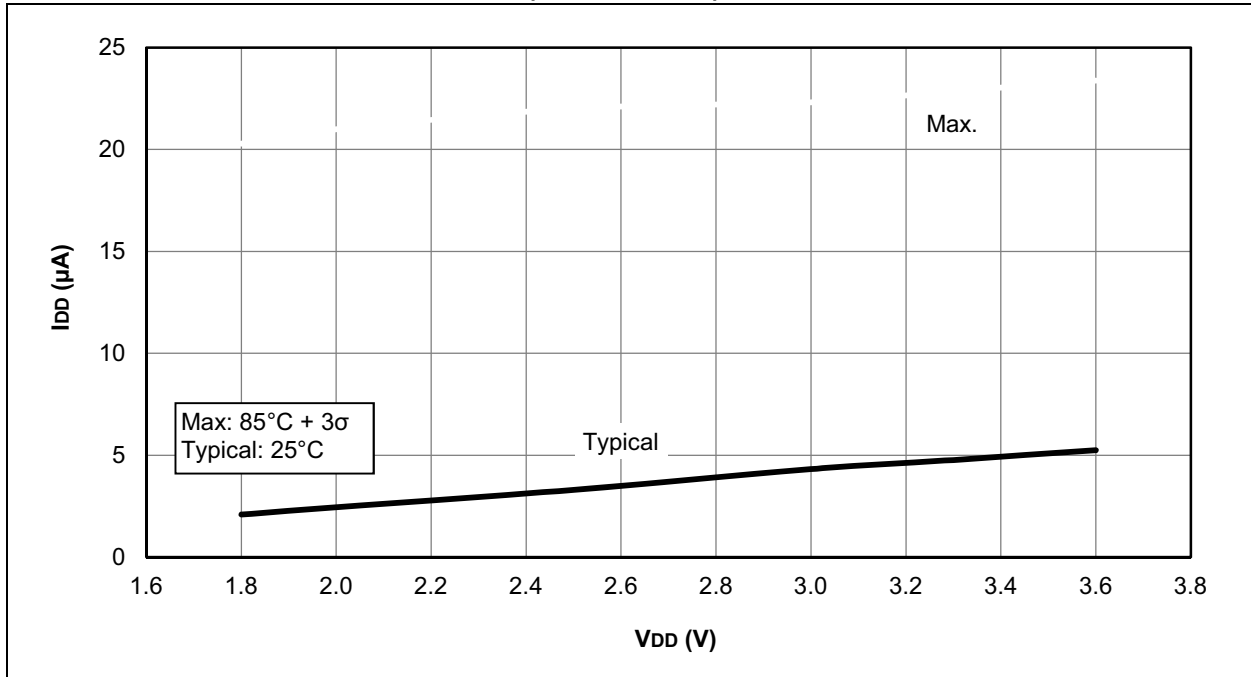
**2:** To ensure these voltage tolerances,  $V_{DD}$  and  $V_{SS}$  must be capacitively decoupled as close to the device as possible. 0.1  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.

**FIGURE 30-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

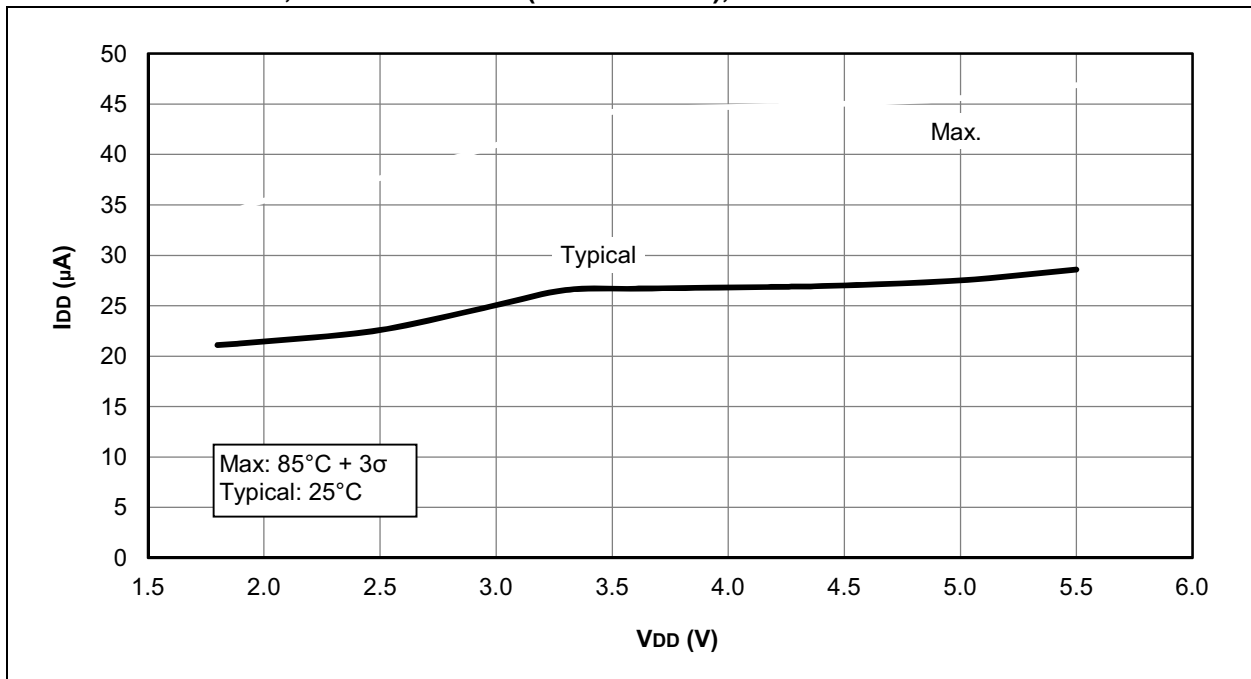


# PIC12(L)F1822/16(L)F1823

**FIGURE 31-19:  $I_{DD}$ , LFINTOSC MODE ( $F_{osc} = 32$  kHz), PIC12LF1822 AND PIC16LF1823 ONLY**

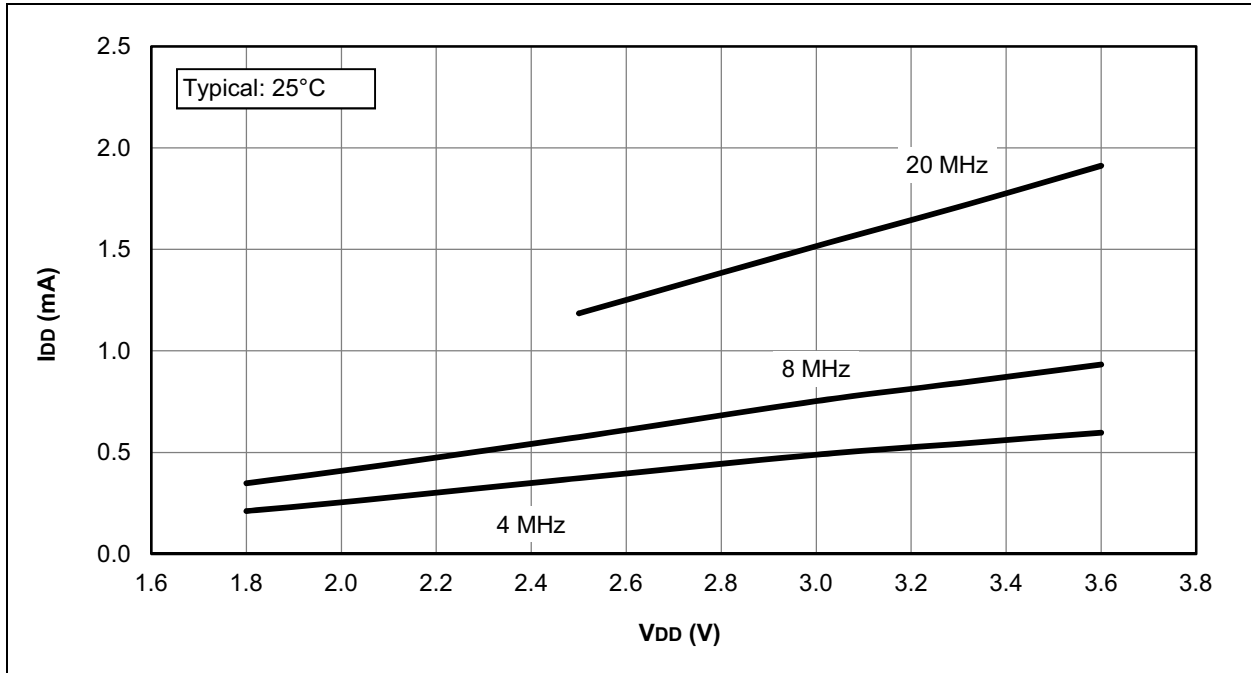


**FIGURE 31-20:  $I_{DD}$ , LFINTOSC MODE ( $F_{osc} = 32$  kHz), PIC12F1822 AND PIC16F1823 ONLY**

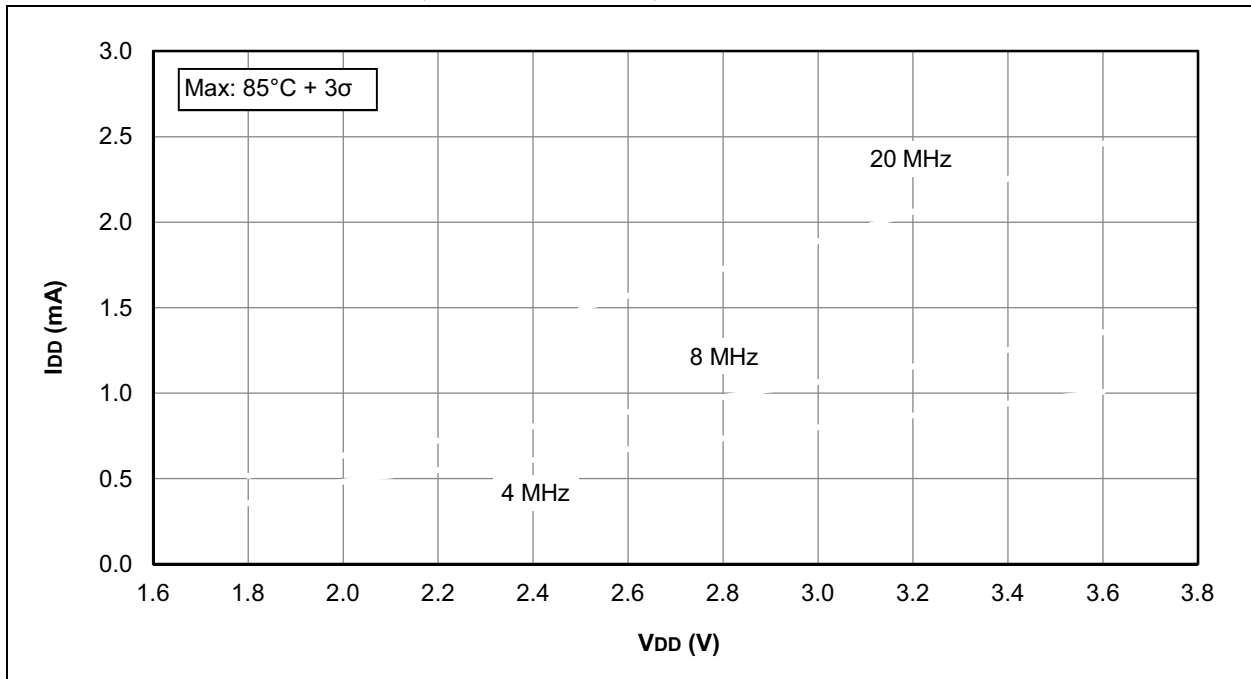


# PIC12(L)F1822/16(L)F1823

**FIGURE 31-27: I<sub>DD</sub> TYPICAL, HS OSCILLATOR, PIC12LF1822 AND PIC16LF1823 ONLY**



**FIGURE 31-28: I<sub>DD</sub> MAXIMUM, HS OSCILLATOR, PIC12LF1822 AND PIC16LF1823 ONLY**





# PIC12(L)F1822/16(L)F1823

FIGURE 31-29: I<sub>DD</sub> TYPICAL, HS OSCILLATOR, PIC12F1822 AND PIC16F1823 ONLY

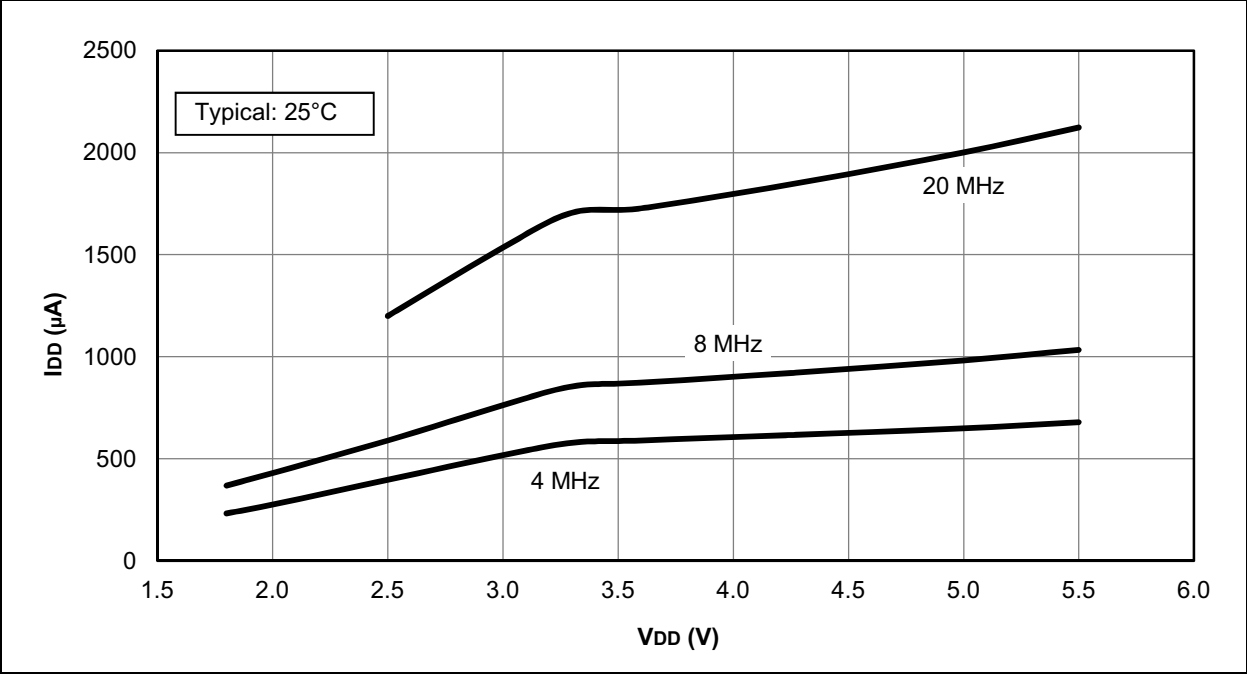
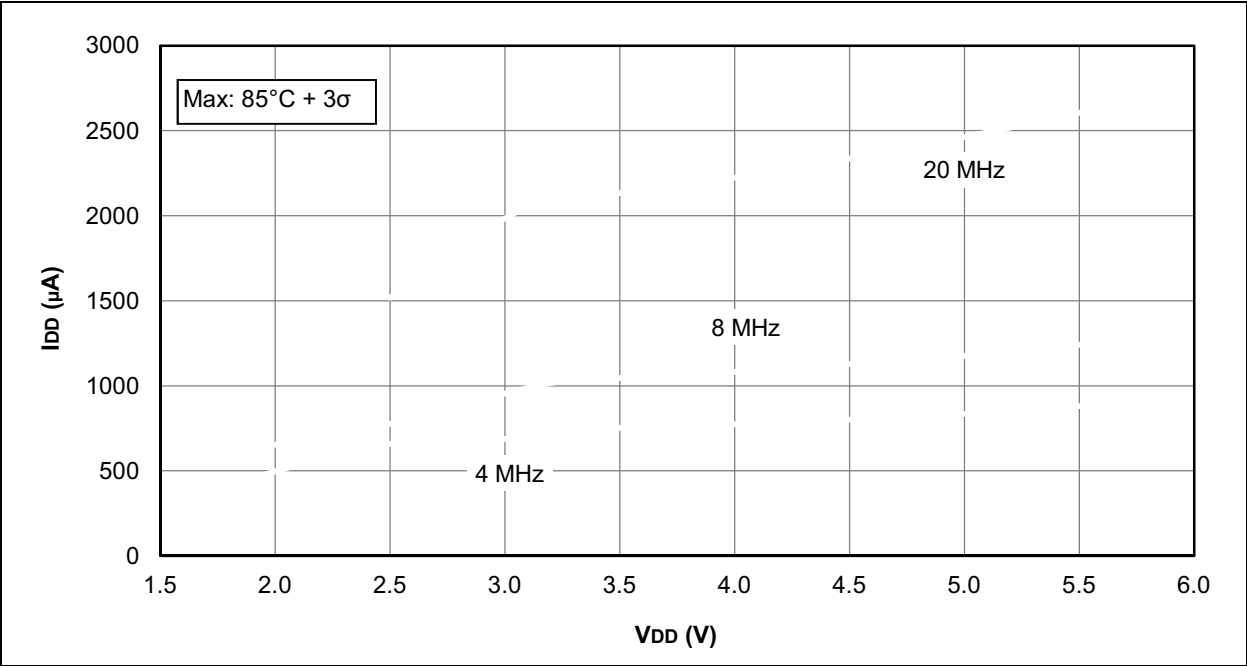


FIGURE 31-30: I<sub>DD</sub> MAXIMUM, HS OSCILLATOR, PIC12F1822 AND PIC16F1823 ONLY



## 32.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for  
Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICKit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,  
Evaluation Kits and Starter Kits
- Third-party development tools

## 32.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

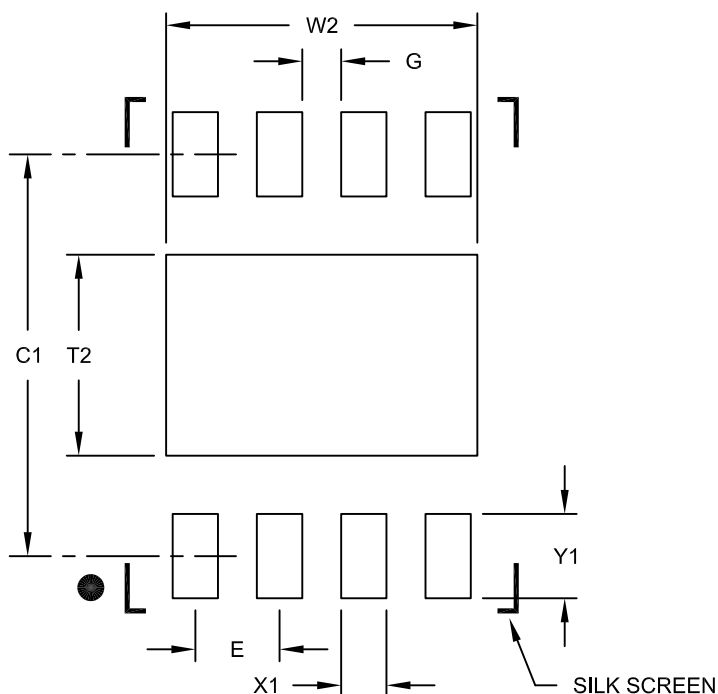
File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

# PIC12(L)F1822/16(L)F1823

## 8-Lead Plastic Dual Flat, No Lead Package (MF) - 3x3x0.9mm Body [DFN]

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



### RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			2.40
Optional Center Pad Length	T2			1.55
Contact Pad Spacing	C1		3.10	
Contact Pad Width (X8)	X1			0.35
Contact Pad Length (X8)	Y1			0.65
Distance Between Pads	G	0.30		

#### Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

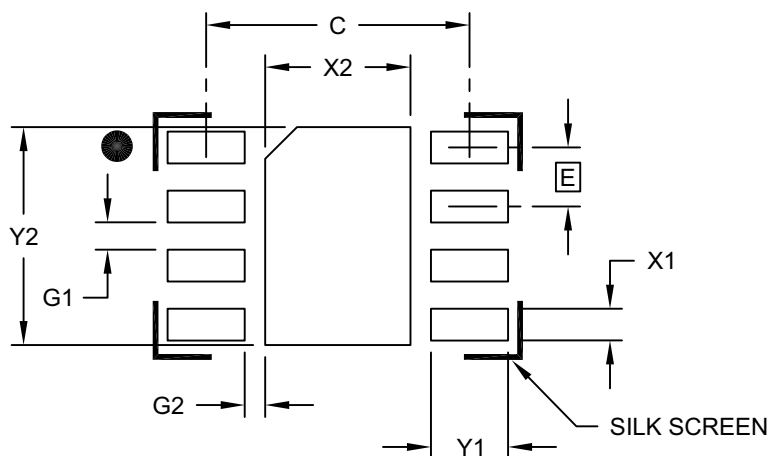
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2062B

# PIC12(L)F1822/16(L)F1823

## 8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (RF) - 3x3x0.50 mm Body [UDFN]

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



### RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	X2			1.60
Optional Center Pad Length	Y2			2.40
Contact Pad Spacing	C		2.90	
Contact Pad Width (X8)	X1			0.35
Contact Pad Length (X8)	Y1			0.85
Contact Pad to Contact Pad (X6)	G1	0.20		
Contact Pad to Center Pad (X8)	G2	0.30		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2254A

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

## THE MICROCHIP WEB SITE

Microchip provides online support via our web site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com). Under "Support", click on "Customer Change Notification" and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://www.microchip.com/support>**