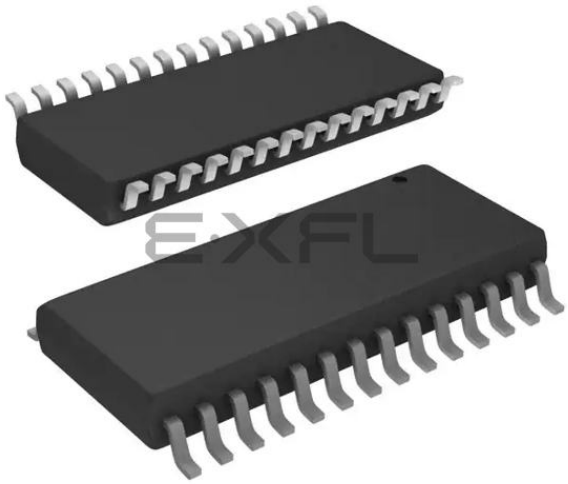


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

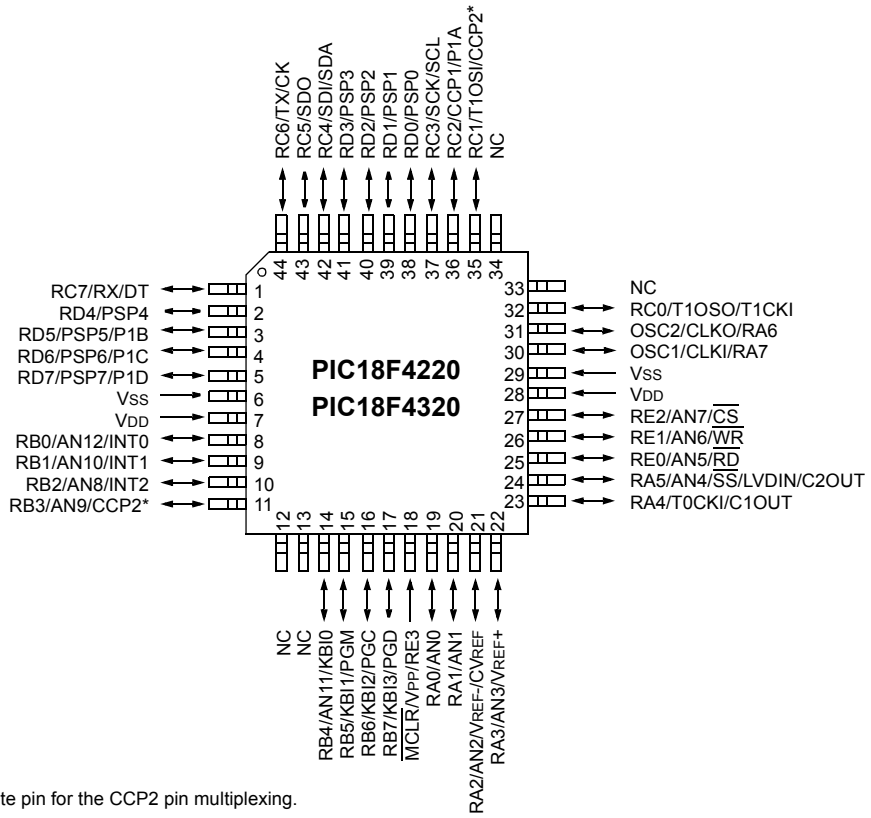
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf2220t-i-so

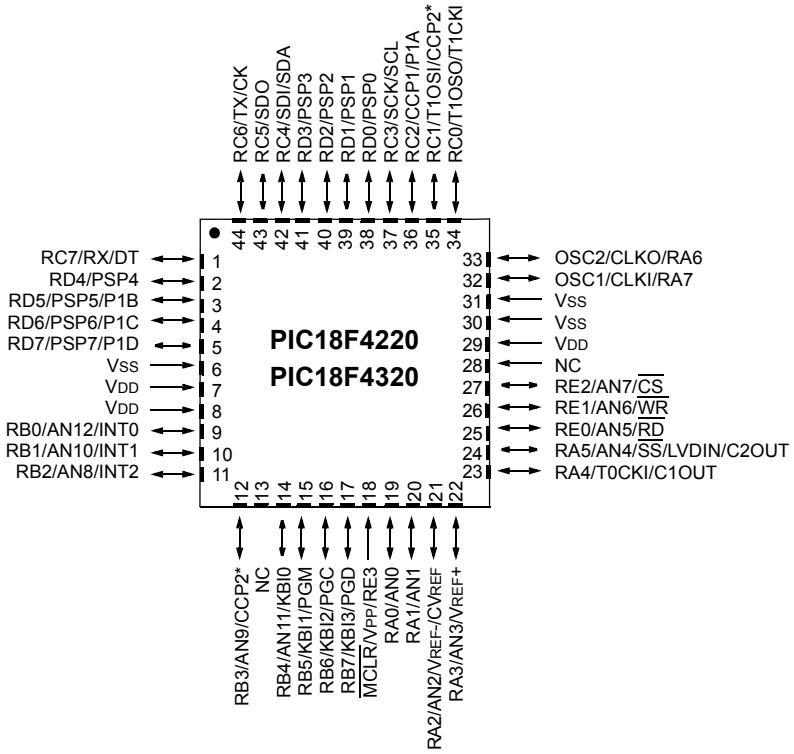
Pin Diagrams (Cont.'d)

44-Pin TQFP



* RB3 is the alternate pin for the CCP2 pin multiplexing.

44-Pin QFN



* RB3 is the alternate pin for the CCP2 pin multiplexing.

Table of Contents

TABLE 1-3: PIC18F4220/4320 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
						PORTD is a bidirectional I/O port or a Parallel Slave Port (PSP) for interfacing to a microprocessor port. These pins have TTL input buffers when PSP module is enabled.
RD0/PSP0	19	38	38			
RD0				I/O	ST	Digital I/O.
PSP0				I/O	TTL	Parallel Slave Port data.
RD1/PSP1	20	39	39			
RD1				I/O	ST	Digital I/O.
PSP1				I/O	TTL	Parallel Slave Port data.
RD2/PSP2	21	40	40			
RD2				I/O	ST	Digital I/O.
PSP2				I/O	TTL	Parallel Slave Port data.
RD3/PSP3	22	41	41			
RD3				I/O	ST	Digital I/O.
PSP3				I/O	TTL	Parallel Slave Port data.
RD4/PSP4	27	2	2			
RD4				I/O	ST	Digital I/O.
PSP4				I/O	TTL	Parallel Slave Port data.
RD5/PSP5/P1B	28	3	3			
RD5				I/O	ST	Digital I/O.
PSP5				I/O	TTL	Parallel Slave Port data.
P1B				O	—	Enhanced CCP1 output.
RD6/PSP6/P1C	29	4	4			
RD6				I/O	ST	Digital I/O.
PSP6				I/O	TTL	Parallel Slave Port data.
P1C				O	—	Enhanced CCP1 output.
RD7/PSP7/P1D	30	5	5			
RD7				I/O	ST	Digital I/O.
PSP7				I/O	TTL	Parallel Slave Port data.
P1D				O	—	Enhanced CCP1 output.

Legend: TTL = TTL compatible input
CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
I = Input
O = Output
P = Power
OD = Open-drain (no diode to V_{DD})

Note 1: Alternate assignment for CCP2 when CCP2MX is cleared.
2: Default assignment for CCP2 when CCP2MX (CONFIG3H<0>) is set.

2.6.3 OSCTUN2 REGISTER

The internal oscillator block is calibrated at the factory to produce an INTRC output frequency of approximately 31 kHz. (See parameters F20 and F21 in Table 26-8.)

The INTRC frequency can be adjusted two ways:

- If TUNSEL (OSCTUN2<7>) is clear – TUN5:TUN1 in OSCTUNE<5:1> adjusts the INTRC clock frequency and also can adjust the INTOSC clock frequency. (See Register 2-1, OSCTUNE.)
- If TUNSEL (OSCTUN2<7>) is set – TUN5:TUN1 in OSCTUN2<5:1> adjusts the INTRC clock frequency *without* affecting the INTOSC frequency. (See Register 2-2, OSCTUN2.)

In OSCTUN2, the OSCTUN2<0> bit has no effect, but is readable and writable, enabling changes of the INTRC frequency using two increment or decrement instructions.

When the OSCTUN2 register is modified, the INTRC frequency will begin shifting to the new frequency, and will stabilize at the new frequency within 100 μs. Code execution continues during this shift.

There is no indication when the shift occurs. Operation of features that depend on the INTRC clock source frequency also will be affected by the change in frequency. This includes the WDT, Fail-Safe Clock Monitor and peripherals.

REGISTER 2-2: OSCTUN2: INTRC OSCILLATOR TUNING REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TUNSEL	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

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

bit 7 **TUNSEL:** Enables tuning of INTRC using OCSTUN2<5:1>
1 = INTRC adjusted by OSCTUN2<5:1>
0 = INTRC adjusted by OSCTUNE<5:1>

bit 6 **Unimplemented:** Read as ‘0’

bit 5-1 **TUN<5:1>:** Frequency Tuning bits – Adjusts the frequency of INTRC when TUNSEL is set
011111 = Maximum frequency
• •
• •
000001
000000 = Center frequency. Oscillator module is running at the calibrated frequency.
111111
• •
• •
100000 = Minimum frequency

bit 0 **TUN<0>:** A placeholder with no effect on the INTRC frequency. Provided to facilitate incrementation and decrementation of the OSCTUN2 register and adjustment of the INTRC frequency.

FIGURE 5-8: INDIRECT ADDRESSING OPERATION

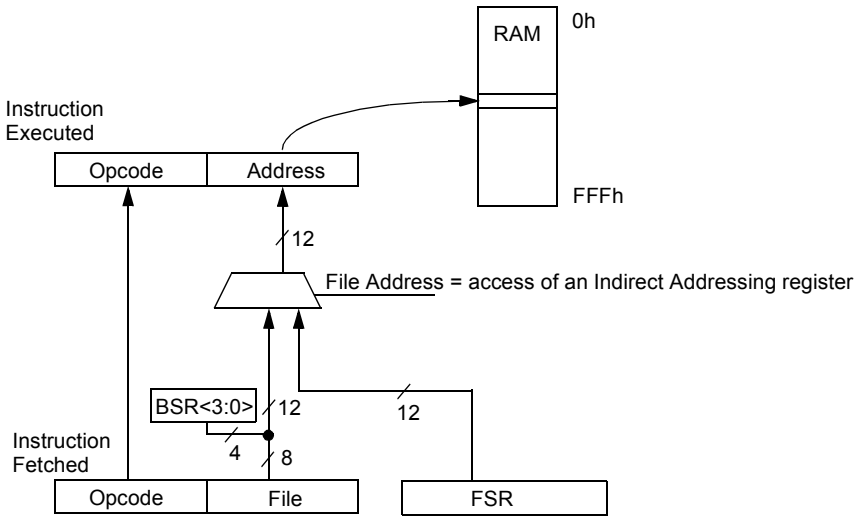
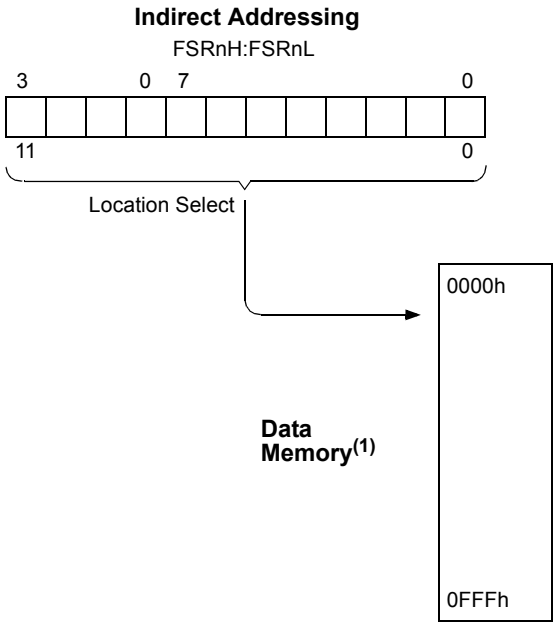


FIGURE 5-9: INDIRECT ADDRESSING



Note 1: For register file map detail, see Table 5-1.

NOTES:

11.0 TIMER0 MODULE

The Timer0 module has the following features:

- Software selectable as an 8-bit or 16-bit timer/counter
- Readable and writable
- Dedicated 8-bit software programmable prescaler
- Clock source selectable to be external or internal
- Interrupt-on-overflow from FFh to 00h in 8-bit mode and FFFFh to 0000h in 16-bit mode
- Edge select for external clock

Figure 11-1 shows a simplified block diagram of the Timer0 module in 8-bit mode and Figure 11-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

The T0CON register (Register 11-1) is a readable and writable register that controls all the aspects of Timer0, including the prescale selection.

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

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

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

12.4 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing Timer1 interrupt enable bit, TMR1IE (PIE1<0>).

12.5 Resetting Timer1 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a "Special Event Trigger" (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1 and start an A/D conversion if the A/D module is enabled (see **Section 15.4.4 "Special Event Trigger"** for more information).

Note:	The Special Event Triggers from the CCP1 module will not set interrupt flag bit, TMR1IF (PIR1<0>).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a Special Event Trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

12.6 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 12-2). When the RD16 control bit (T1CON<7>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 high byte buffer. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, is valid due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

12.7 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 12.2 "Timer1 Oscillator"** above), gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, `RTCisr`, shown in Example 12-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow, triggers the interrupt and calls the routine, which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflow.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it; the simplest method is to set the MSbit of TMR1H with a `BSF` instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

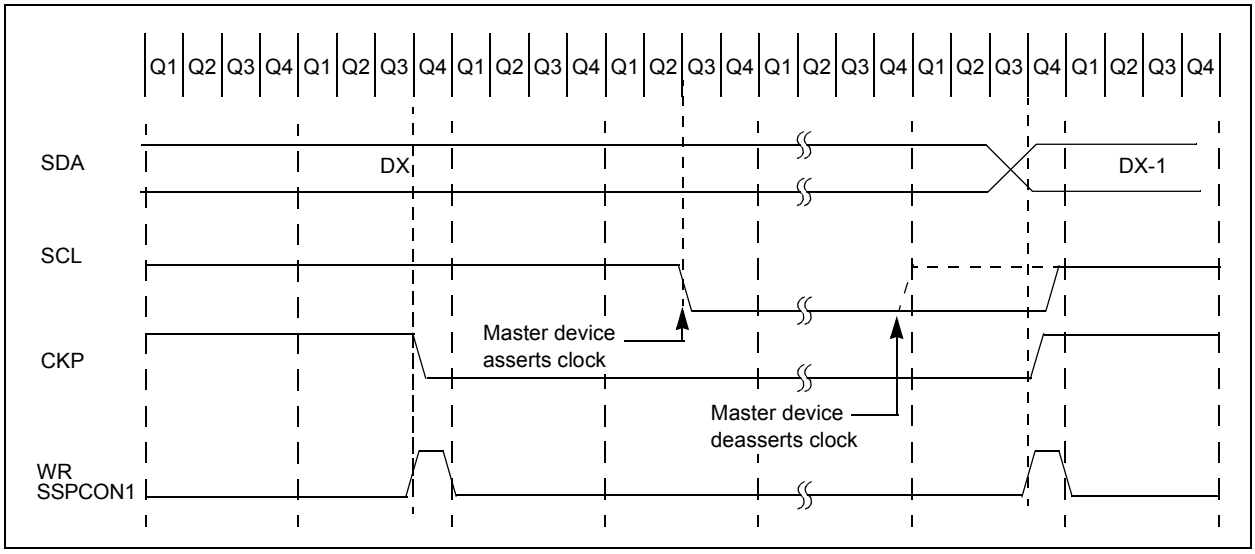
For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1) as shown in the routine, `RTCinit`. The Timer1 oscillator must also be enabled and running at all times.

17.4.4.5 Clock Synchronization and the CKP bit (SEN = 1)

The SEN bit is also used to synchronize writes to the CKP bit. If a user clears the CKP bit, the SCL output is forced to '0'. When the SEN bit is set to '1', setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. If the user attempts to drive SCL low, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 17-12).

Note:	If the SEN bit is '0', clearing the CKP bit will result in immediately driving the SCL output to '0' regardless of the current state.

FIGURE 17-12: CLOCK SYNCHRONIZATION TIMING



17.4.6.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 17.4.7 "Baud Rate"** for more detail.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start enable bit, SEN (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

18.3.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 18-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

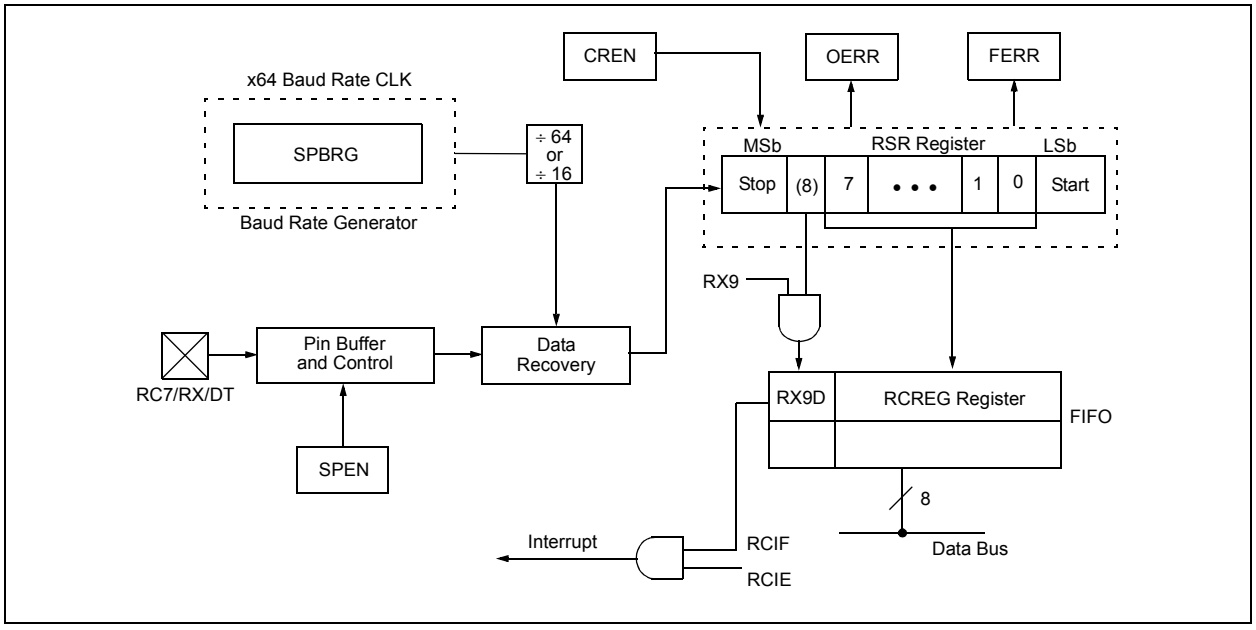
1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (**Section 18.2 "USART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RCIE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCIE, was set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

18.3.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with address detect enable:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is required, set the BRGH bit.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 18-4: USART RECEIVE BLOCK DIAGRAM



18.4 USART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTA<4>). In addition, enable bit, SPEN (RCSTA<7>), is set in order to configure the RC6/TX/CK and RC7/RX/DT I/O pins to CK (clock) and DT (data) lines, respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit, CSRC (TXSTA<7>).

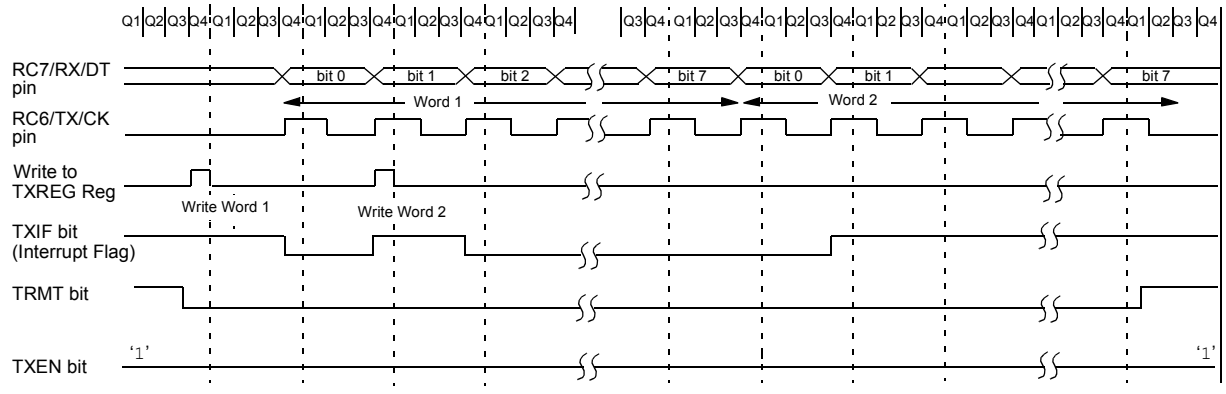
18.4.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCYCLE), the TXREG is empty and interrupt bit, TXIF (PIR1<4>), is set. The interrupt can be enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit, TXIF, will be set regardless of the state of enable bit, TXIE, and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit, TXIF, indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 18.2 "USART Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit, TXIE.
4. If 9-bit transmission is desired, set bit, TX9.
5. Enable the transmission by setting bit, TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 18-6: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SPBRG = 0; continuous transmission of two 8-bit words.

REGISTER 23-1: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
IESO	FSCM	—	—	FOSC3	FOSC2	FOSC1	FOSC0
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7 **IESO:** Internal/External Switchover bit
1 = Internal/External Switchover mode enabled
0 = Internal/External Switchover mode disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **FOSC3:FOSC0:** Oscillator Selection bits(1)
11xx = External RC oscillator, CLKO function on RA6
1001 = Internal oscillator block, CLKO function on RA6 and port function on RA7
1000 = Internal oscillator block, port function on RA6 and port function on RA7
0111 = External RC oscillator, port function on RA6
0110 = HS oscillator, PLL enabled (clock frequency = 4 x FOSC1)
0101 = EC oscillator, port function on RA6
0100 = EC oscillator, CLKO function on RA6
0010 = HS oscillator
0001 = XT oscillator
0000 = LP oscillator

REGISTER 23-4: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

R/P-1	U-0	U-0	U-0	U-0	U-0	R/P-1	R/P-1
MCLRE	—	—	—	—	—	PBAD	CCP2MX
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit
1 = $\overline{\text{MCLR}}$ pin enabled; RE3 input pin disabled
0 = $\overline{\text{MCLR}}$ disabled; RE3 input is enabled in 40-pin devices only (PIC18F4X20)

bit 6-2 **Unimplemented:** Read as '0'

bit 1 **PBAD:** PORTB A/D Enable bit (Affects ADCON1 Reset state. ADCON1 controls PORTB<4:0> pin configuration.)
1 = PORTB<4:0> pins are configured as analog input channels on Reset
0 = PORTB<4:0> pins are configured as digital I/O on Reset

bit 0 **CCP2MX:** CCP2 MUX bit
1 = CCP2 input/output is multiplexed with RC1
0 = CCP2 input/output is multiplexed with RB3

REGISTER 23-5: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVR
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins
0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug

bit 6-3 **Unimplemented:** Read as '0'

bit 2 **LVP:** Single-Supply ICSP™ Enable bit
1 = Single-Supply ICSP enabled
0 = Single-Supply ICSP disabled

bit 1 **Unimplemented:** Read as '0'

bit 0 **STVR:** Stack Full/Underflow Reset Enable bit
1 = Stack full/underflow will cause Reset
0 = Stack full/underflow will not cause Reset

24.2 Instruction Set

ADDLW **ADD Literal to W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: $(W) + k \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal 'k' and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: `ADDLW 0x15`

Before Instruction
W = 0x10

After Instruction
W = 0x25

ADDWF **ADD W to f**

Syntax: [*label*] ADDWF f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: $(W) + (f) \rightarrow$ dest

Status Affected: N, OV, C, DC, Z

Encoding:

0010	01da	ffff	ffff

Description: Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank will be selected. If 'a' is '1', the BSR is used.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `ADDWF REG, W`

Before Instruction
W = 0x17
REG = 0xC2

After Instruction
W = 0xD9
REG = 0xC2

26.3 DC Characteristics: PIC18F2220/2320/4220/4320 (Industrial) PIC18LF2220/2320/4220/4320 (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**				
		I/O Ports:				
D030		with TTL Buffer	VSS	0.15 VDD	V	VDD < 4.5V
D030A			—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D031		with Schmitt Trigger Buffer	VSS	0.2 VDD	V	
		RC3 and RC4	VSS	0.3 VDD	V	
D032		$\overline{MCLR}$	VSS	0.2 VDD	V	
D032A		OSC1 and T1OSI	VSS	0.2 VDD	V	LP, XT, HS, HSPLL modes(1)
D033		OSC1	VSS	0.2 VDD	V	EC mode(1)
	VIH	**Input High Voltage**				
		I/O Ports:				
D040		with TTL Buffer	0.25 VDD + 0.8V	VDD	V	VDD < 4.5V
D040A			2.0	VDD	V	4.5V ≤ VDD ≤ 5.5V
D041		with Schmitt Trigger Buffer	0.8 VDD	VDD	V	
		RC3 and RC4	0.7 VDD	VDD	V	
D042		$\overline{MCLR}$	0.8 VDD	VDD	V	
D042A		OSC1 and T1OSI	1.6	VDD	V	LP, XT, HS, HSPLL modes(1)
D043		OSC1	0.8 VDD	VDD	V	EC mode(1)
	IIL	**Input Leakage Current(2,3)**				
D060		I/O Ports	—	±0.2	µA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
D061		$\overline{MCLR}$, RA4	—	±1.0	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1	—	±1.0	µA	Vss ≤ VPIN ≤ VDD
	IPU	**Weak Pull-up Current**				
D070	IPURB	PORTB Weak Pull-up Current	50	400	µA	VDD = 5V, VPIN = VSS

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC® device be driven with an external clock while in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: Parameter is characterized but not tested.

TABLE 26-7: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 4.2V TO 5.5V)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
F10	F_{OSC}	Oscillator Frequency Range	4	—	10	MHz	HS mode only
F11	F_{SYS}	On-Chip VCO System Frequency	16	—	40	MHz	HS mode only
F12	t_{PLL}	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13	Δ_{CLK}	CLKO Stability (Jitter)	-2	—	+2	%	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 26-8: INTERNAL RC ACCURACY: PIC18F2220/2320/4220/4320 (Industrial)
PIC18LF2220/2320/4220/4320 (Industrial, Extended)

PIC18LF1220/1320 (Industrial)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial						
PIC18F1220/1320 (Industrial, Extended)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended						
Param No.	**Device**	**Min**	**Typ**	**Max**	**Units**	**Conditions**		
	INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz(1)							
F14	PIC18LF2220/2320/4220/4320	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V	
F15		-5	—	5	%	-10°C to +85°C	VDD = 2.7-3.3V	
F16		-10	—	10	%	-40°C to +85°C	VDD = 2.7-3.3V	
F17	PIC18F2220/2320/4220/4320	-2	+/-1	2	%	+25°C	VDD = 4.5-5.5V	
F18		-5	—	5	%	-10°C to +85°C	VDD = 4.5-5.5V	
F19		-10	—	10	%	-40°C to +85°C	VDD = 4.5-5.5V	
	INTRC Accuracy @ Freq = 31 kHz(2)							
F20	PIC18LF2220/2320/4220/4320	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V	
F21	PIC18F2220/2320/4220/4320	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 4.5-5.5V	

Legend: Shading of rows is to assist in readability of the table.

Note 1: Frequency calibrated at 25°C. OSCTUNE register can be used to compensate for temperature drift.

2: INTRC frequency after calibration.

3: Change of INTRC frequency as VDD changes.

FIGURE 27-7: MAXIMUM IDD vs. FOSC OVER VDD PRI_RUN, EC MODE, -40°C TO +125°C

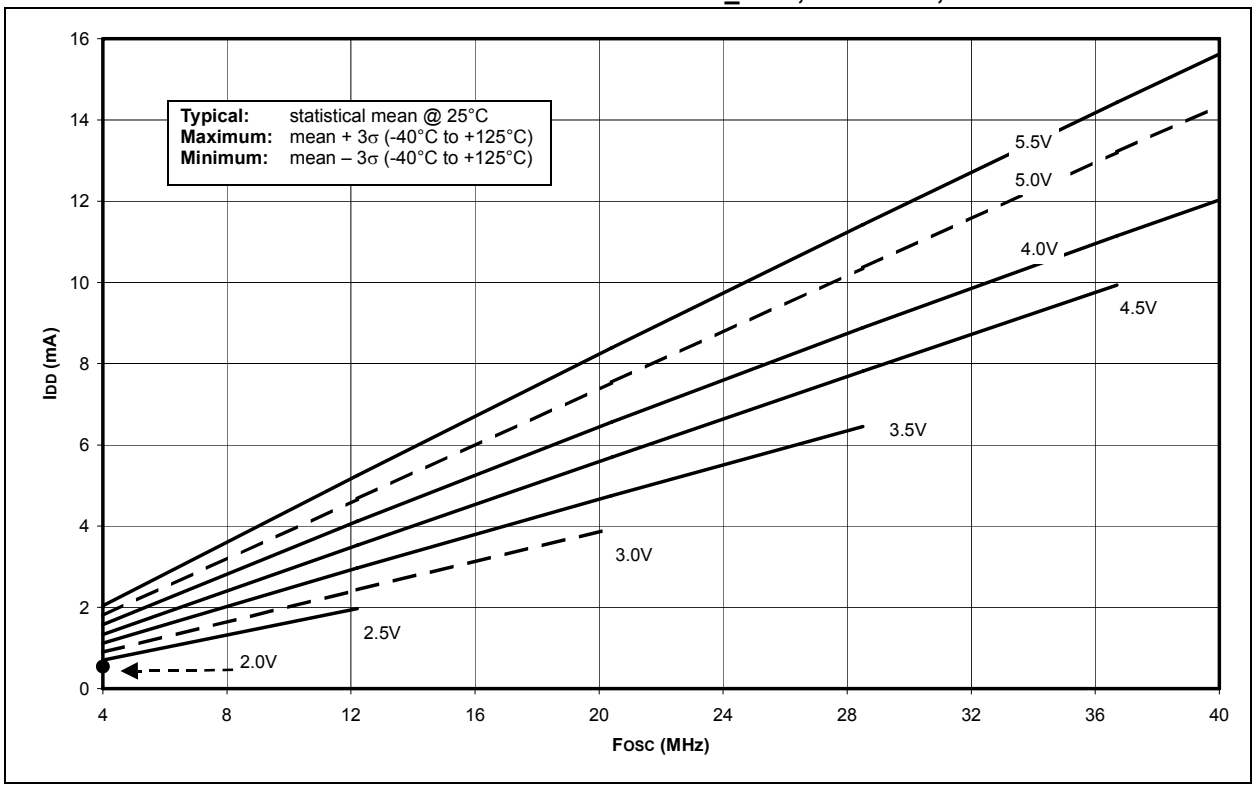


FIGURE 27-8: TYPICAL IDD vs. FOSC OVER VDD PRI_IDLE, EC MODE, +25°C

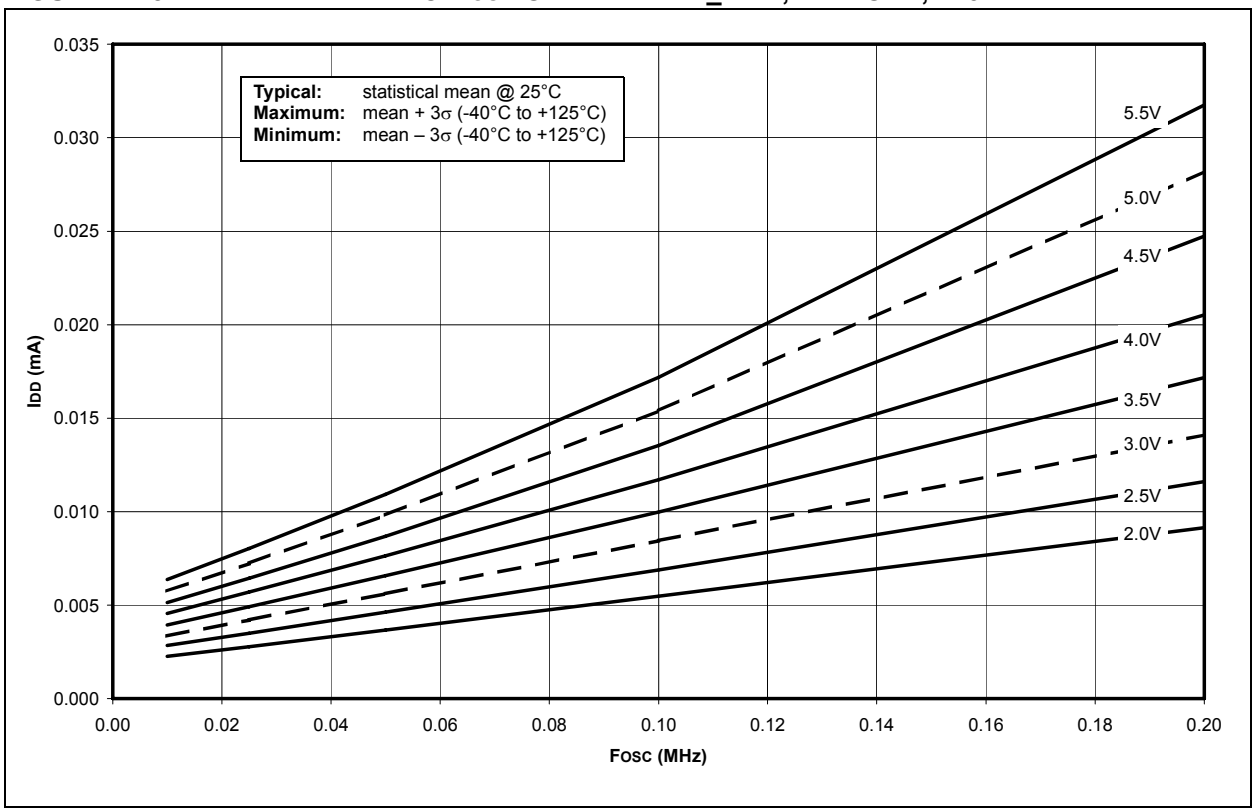


FIGURE 27-15: TYPICAL I_{PD} vs. V_{DD} (+25°C), 125 kHz TO 8 MHz RC_RUN MODE, ALL PERIPHERALS DISABLED

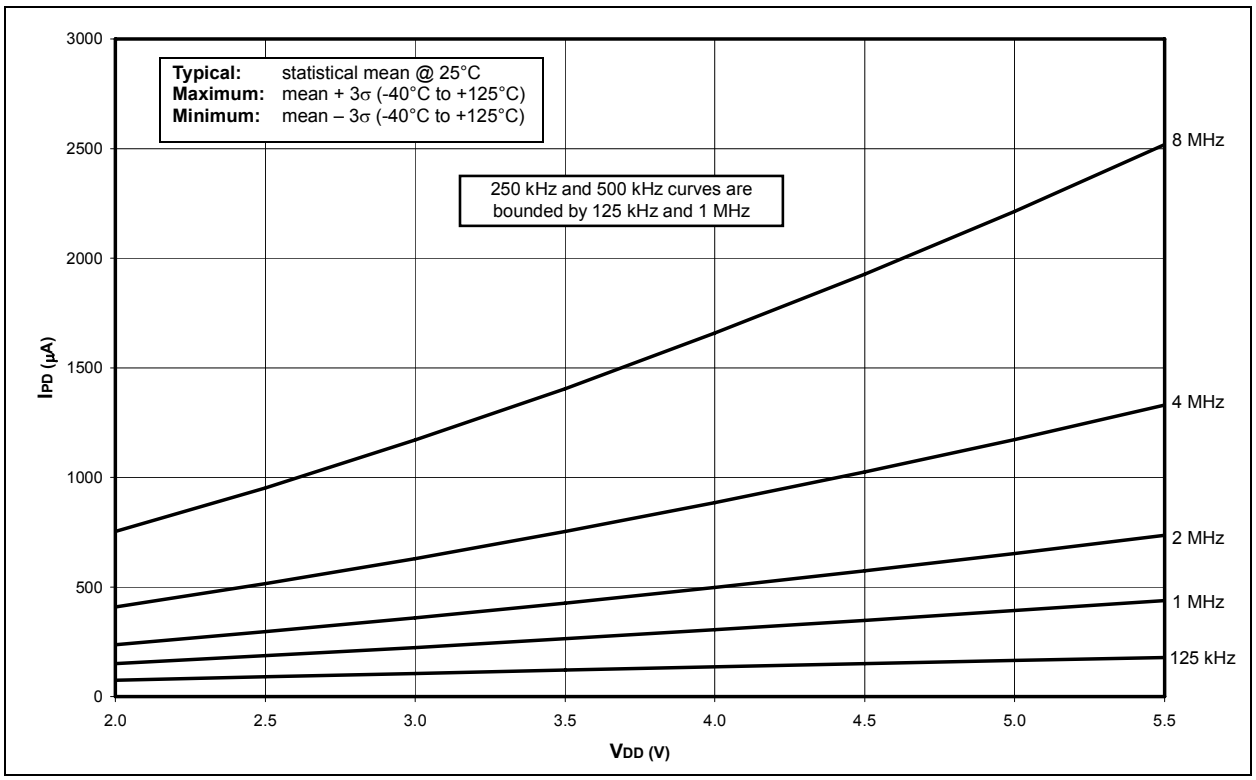
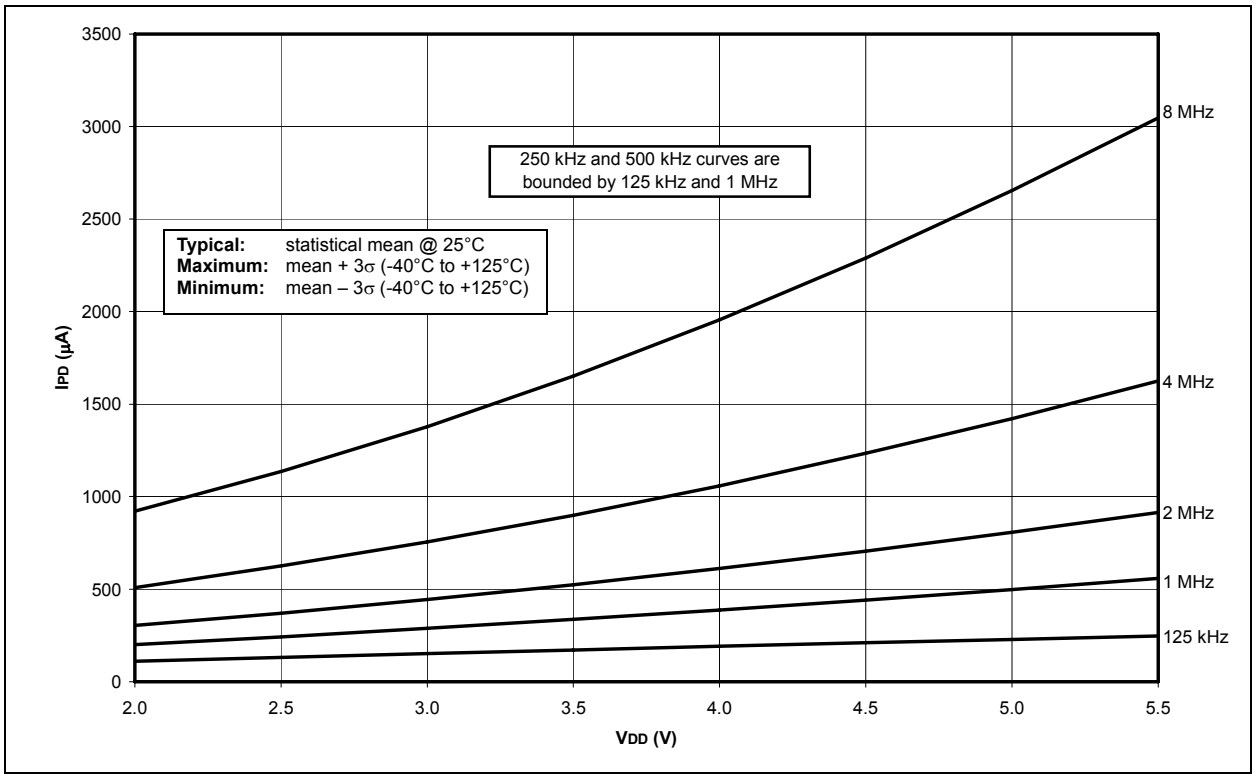


FIGURE 27-16: MAXIMUM I_{PD} vs. V_{DD} (-40°C TO +125°C), 125 kHz TO 8 MHz RC_RUN, ALL PERIPHERALS DISABLED



APPENDIX A: REVISION HISTORY

Revision A (June 2002)

Original data sheet for PIC18F2X20/4X20 devices.

Revision B (October 2002)

This revision includes major changes to **Section 2.0 "Oscillator Configurations"** and **Section 3.0 "Power-Managed Modes"**, updates to the Electrical Specifications in **Section 26.0 "Electrical Characteristics"** and minor corrections to the data sheet text.

Revision C (October 2003)

This revision includes updates to the Electrical Specifications in **Section 26.0 "Electrical Characteristics"** and to the DC Characteristics Graphs and Charts in **Section 27.0 "DC and AC Characteristics Graphs and Tables"** and minor corrections to the data sheet text.

Revision D (October 2006)

This revision includes updates to the packaging diagrams.

Revision E (January 2007)

This revision includes updates to the packaging diagrams.

Revision F (February 2007)

This revision includes updates to the packaging diagrams.

Revision G (December 2007)

- Modified OSCTUNE register data and added OSCTUN2 register data to **Section 2.6 "Internal Oscillator Block"** and Table 4-3 and Table 5-1.
- Changed Brown-out Voltage values in **Section 26.1 "DC Characteristics: Supply Voltage PIC18F2220/2320/4220/4320 (Industrial) PIC18LF2220/2320/4220/4320 (Industrial)"**.
- Updated low-voltage detect values in Table 26-4.
- Removed RE3 pin references for PIC18F2220/2320 devices in **Section 1.0 "Device Overview"**, **Section 5.0 "Memory Organization"**, **Section 10.0 "I/O Ports"**, **Section 19.0 "10-bit Analog-to-Digital Converter (A/D) Module"** and **Section 23.0 "Special Features of the CPU"**.
- Made minor changes to **Section 17.3.3 "Enabling SPI I/O"**; Table 1-2, Table 1-3, Table 3-3, Table 12-1 and Table 26-2; Figure 12-3 and Figure 16-1; Example 10-1 and Example 10-2; and Table 21-1 and Table 23-1.