



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](#)"

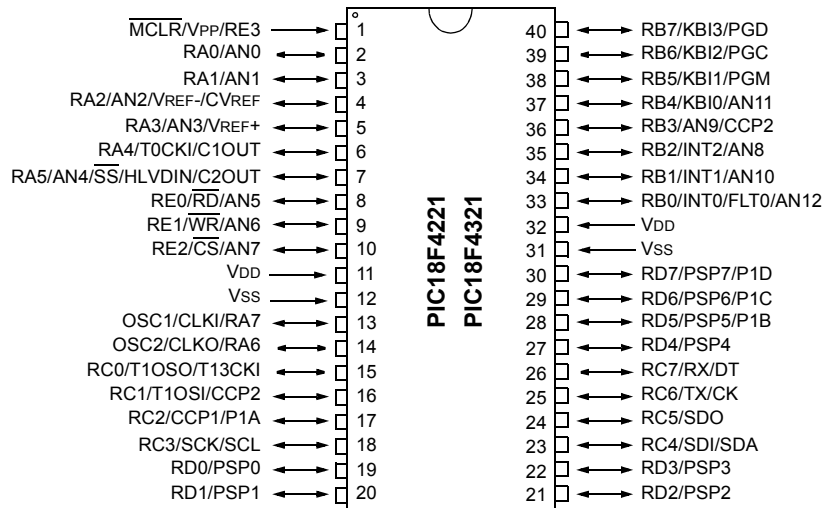
Details

Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2321t-i-ml

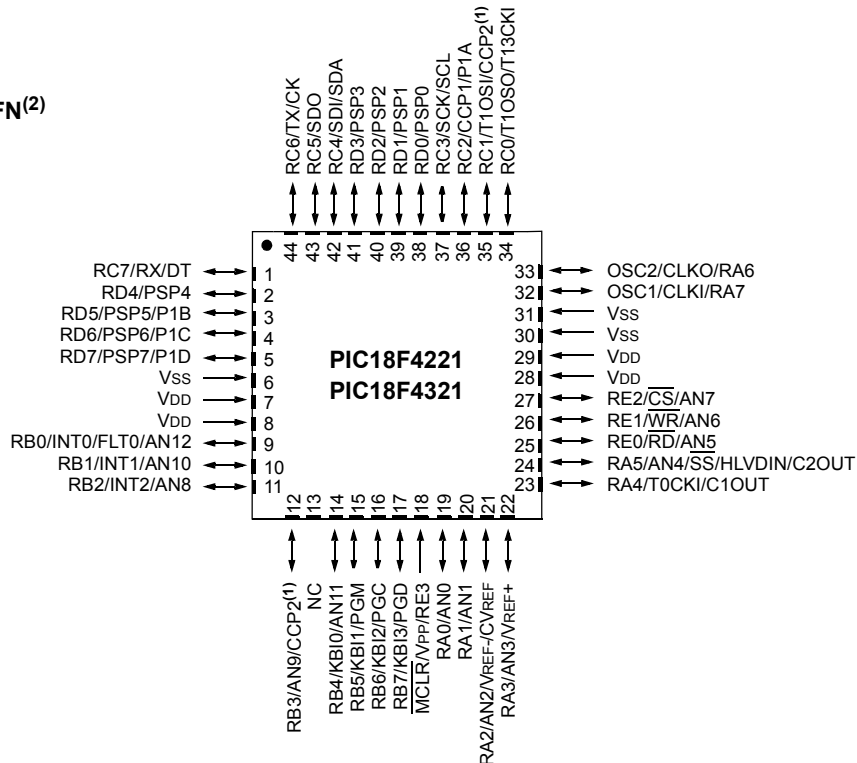
PIC18F2221/2321/4221/4321 FAMILY

Pin Diagrams (Continued)

40-Pin PDIP



44-Pin QFN⁽²⁾



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

Note 2: For the QFN package, it is recommended that the bottom pad be connected to VSS.

PIC18F2221/2321/4221/4321 FAMILY

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD} , V_{DD} RISE < T_{PWRT})

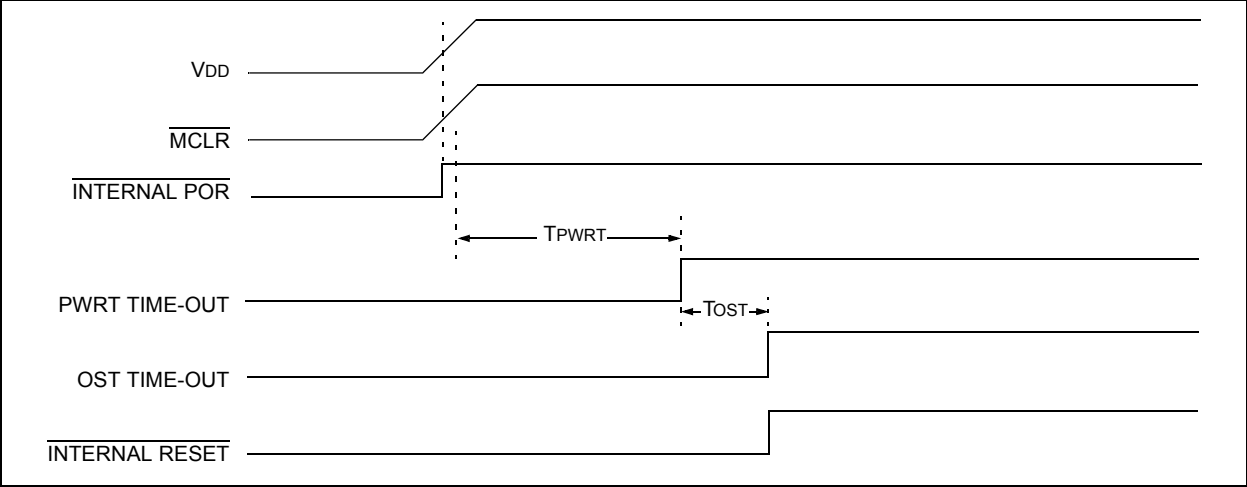


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

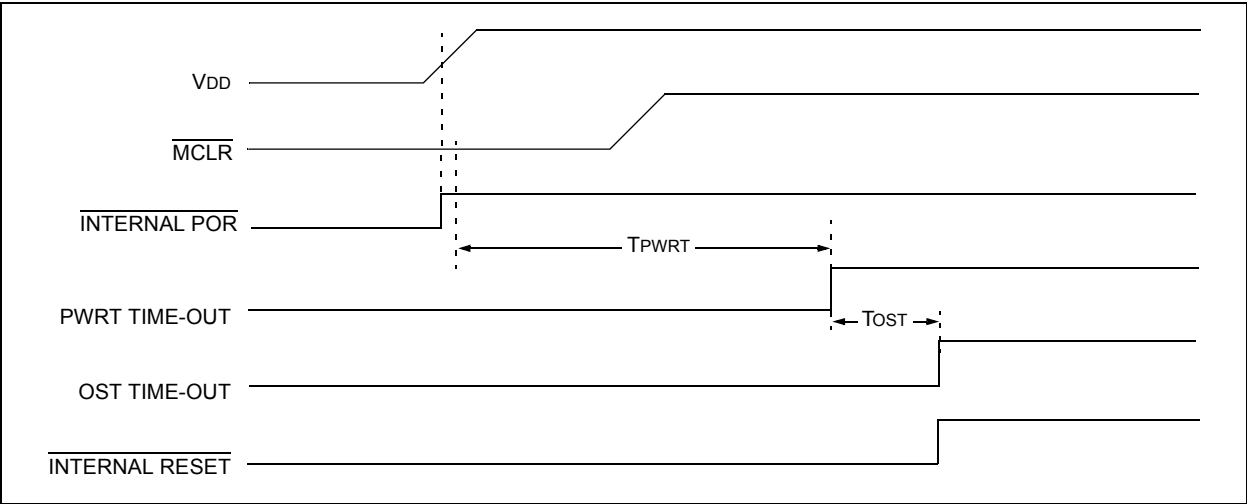
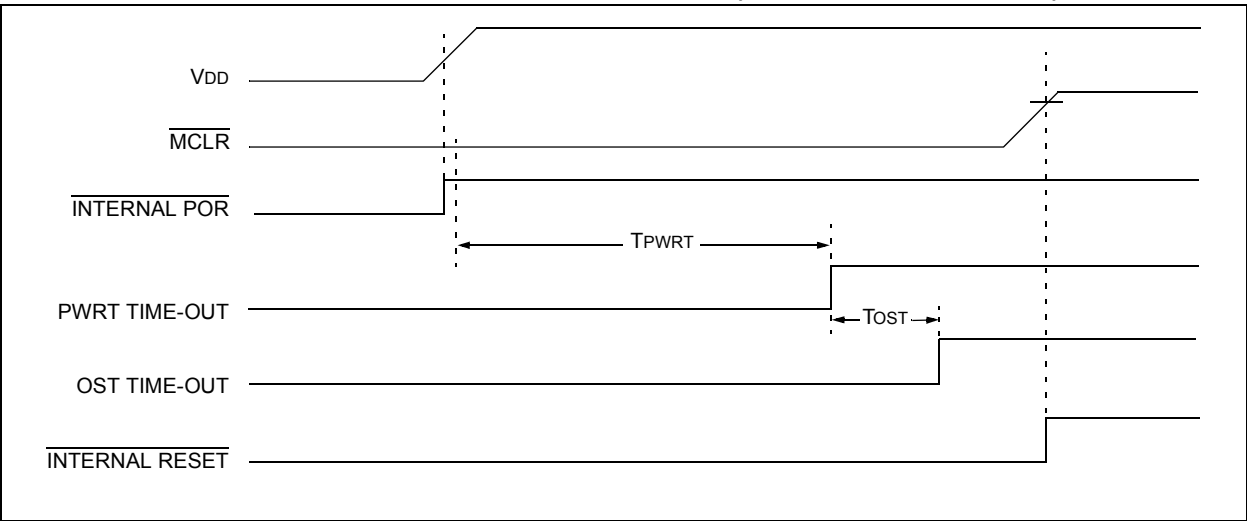


FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2



PIC18F2221/2321/4221/4321 FAMILY

EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

```

PROGRAM_MEMORY
    BCF     INTCON, GIE           ; disable interrupts
    MOVLW   55h                  ; required sequence
    MOVWF   EECON2               ; write 55h
    MOVLW   0AAh
    MOVWF   EECON2               ; write AAh
    BSF     EECON1, WR           ; start program (CPU stall)
    NOP
    BSF     INTCON, GIE           ; re-enable interrupts
    DECFSZ  COUNTER_HI           ; loop until done
    GOTO    PROGRAM_LOOP
    BCF     EECON1, WREN         ; disable write to memory
    
```

7.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

7.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a $\overline{\text{MCLR}}$ Reset or a WDT Time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

7.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See **Section 24.0 “Special Features of the CPU”** for more detail.

7.6 Flash Program Operation During Code Protection

See **Section 24.5 “Program Verification and Code Protection”** for details on code protection of Flash program memory.

TABLE 7-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					55
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								55
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								55
TABLAT	Program Memory Table Latch								55
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
EECON2	EEPROM Control Register 2 (not a physical register)								57
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	57
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	58
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	58

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used during Flash/EEPROM access.

PIC18F2221/2321/4221/4321 FAMILY

15.2 Timer3 16-Bit Read/Write Mode

Timer3 can be configured for 16-bit reads and writes (see Figure 15-2). When the RD16 control bit (T3CON<7>) is set, the address for TMR3H is mapped to a buffer register for the high byte of Timer3. A read from TMR3L will load the contents of the high byte of Timer3 into the Timer3 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer3 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer3 must also take place through the TMR3H Buffer register. The Timer3 high byte is updated with the contents of TMR3H when a write occurs to TMR3L. This allows a user to write all 16 bits to both the high and low bytes of Timer3 at once.

The high byte of Timer3 is not directly readable or writable in this mode. All reads and writes must take place through the Timer3 High Byte Buffer register.

Writes to TMR3H do not clear the Timer3 prescaler. The prescaler is only cleared on writes to TMR3L.

15.3 Using the Timer1 Oscillator as the Timer3 Clock Source

The Timer1 internal oscillator may be used as the clock source for Timer3. The Timer1 oscillator is enabled by setting the T1OSCEN (T1CON<3>) bit. To use it as the Timer3 clock source, the TMR3CS bit must also be set. As previously noted, this also configures Timer3 to increment on every rising edge of the oscillator source.

The Timer1 oscillator is described in **Section 13.0 “Timer1 Module”**.

15.4 Timer3 Interrupt

The TMR3 register pair (TMR3H:TMR3L) increments from 0000h to FFFFh and overflows to 0000h. The Timer3 interrupt, if enabled, is generated on overflow and is latched in interrupt flag bit, TMR3IF (PIR2<1>). This interrupt can be enabled or disabled by setting or clearing the Timer3 Interrupt Enable bit, TMR3IE (PIE2<1>).

15.5 Resetting Timer3 Using the CCP Special Event Trigger

If either of the CCP modules is configured to use Timer3 and to generate a Special Event Trigger in Compare mode (CCP1M<3:0> or CCP2M<3:0> = 1011), this signal will reset Timer3. It will also start an A/D conversion if the A/D module is enabled (see **Section 16.3.4 “Special Event Trigger”** for more information).

The module must be configured as either a timer or synchronous counter to take advantage of this feature. When used this way, the CCPR2H:CCPR2L register pair effectively becomes a period register for Timer3.

If Timer3 is running in Asynchronous Counter mode, the Reset operation may not work.

In the event that a write to Timer3 coincides with a Special Event Trigger from a CCP module, the write will take precedence.

Note: The Special Event Triggers from the CCP2 module will not set the TMR3IF interrupt flag bit (PIR2<1>).

TABLE 15-1: REGISTERS ASSOCIATED WITH TIMER3 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	58
TMR3L	Timer3 Register Low Byte								57
TMR3H	Timer3 Register High Byte								57
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \overline{C}	TMR1CS	TMR1ON	56
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYN \overline{C}	TMR3CS	TMR3ON	57

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer3 module.

PIC18F2221/2321/4221/4321 FAMILY

18.3.8 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full power mode. In the case of Sleep mode, all clocks are halted.

In Idle modes, a clock is provided to the peripherals. That clock should be from the primary clock source, the secondary clock (Timer1 oscillator at 32.768 kHz) or the INTOSC source. See **Section 3.7 “Clock Sources and Oscillator Switching”** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

18.3.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

18.3.10 BUS MODE COMPATIBILITY

Table 18-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 18-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit which controls when the data is sampled.

TABLE 18-2: REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58
TRISA	TRISA7 ⁽²⁾	TRISA6 ⁽²⁾	PORTA Data Direction Control Register						58
TRISC	PORTC Data Direction Control Register								58
SSPBUF	MSSP Receive Buffer/Transmit Register								56
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	56
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	56

Legend: Shaded cells are not used by the MSSP in SPI mode.

Note 1: These bits are unimplemented on 28-pin devices and read as ‘0’.

2: PORTA<7:6> and their direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as ‘0’.

PIC18F2221/2321/4221/4321 FAMILY

18.4.3.3 Reception

When the $\overline{R/W}$ bit of the address byte is clear and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low (\overline{ACK}).

When the address byte overflow condition exists, then the no Acknowledge (\overline{ACK}) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON1<6>) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit, SSPIF (PIR1<3>), must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPCON2<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPCON1<4>). See **Section 18.4.4 “Clock Stretching”** for more detail.

18.4.3.4 Transmission

When the $\overline{R/W}$ bit of the incoming address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The \overline{ACK} pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low regardless of SEN (see **Section 18.4.4 “Clock Stretching”** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit, CKP (SSPCON1<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 18-10).

The \overline{ACK} pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not \overline{ACK}), then the data transfer is complete. In this case, when the \overline{ACK} is latched by the slave, the slave logic is reset and the slave monitors for another occurrence of the Start bit. If the SDA line was low (\overline{ACK}), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

19.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

The EUSART can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-wake-up on Break signal
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of the Enhanced USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT as an EUSART:

- bit SPEN (RCSTA<7>) must be set (= 1)
- bit TRISC<7> must be set (= 1)
- bit TRISC<6> must be set (= 1)

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.
--

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 19-1, Register 19-2 and Register 19-3, respectively.

PIC18F2221/2321/4221/4321 FAMILY

19.2.4 AUTO-WAKE-UP ON SYNC BREAK CHARACTER

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

The auto-wake-up feature is enabled by setting the WUE bit (BAUDCON<1>). Once set, the typical 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/J2602 protocol.)

Following a wake-up event, the module generates an RCIF interrupt. The interrupt is generated synchronously to the Q clocks in normal operating modes (Figure 19-8) and asynchronously, if the device is in Sleep mode (Figure 19-9). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared once a low-to-high transition is observed on the RX line following the wake-up event. At this point, the EUSART module is in Idle mode and returns to normal operation. This signals to the user that the Sync Break event is over.

19.2.4.1 Special Considerations Using Auto-Wake-up

Since auto-wake-up functions by sensing rising edge transitions on RX/DT, information with any state changes before the Stop bit may signal a false end-of-character

and cause data or framing errors. To work properly, therefore, the initial character in the transmission must be all '0's. This can be 00h (8 bytes) for standard RS-232 devices or 000h (12 bits) for the LIN/J2602 bus.

Oscillator start-up time must also be considered, especially in applications using oscillators with longer start-up intervals (i.e., XT or HS 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.

19.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCIF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSART in an Idle mode. The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared after this when a rising edge is seen on RX/DT. The interrupt condition is then cleared by reading the RCREG register. Ordinarily, the data in RCREG will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set) and the RCIF flag is set should not be used as an indicator of the integrity of the data in RCREG. Users should consider implementing a parallel method in firmware to verify received data integrity.

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

FIGURE 19-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION

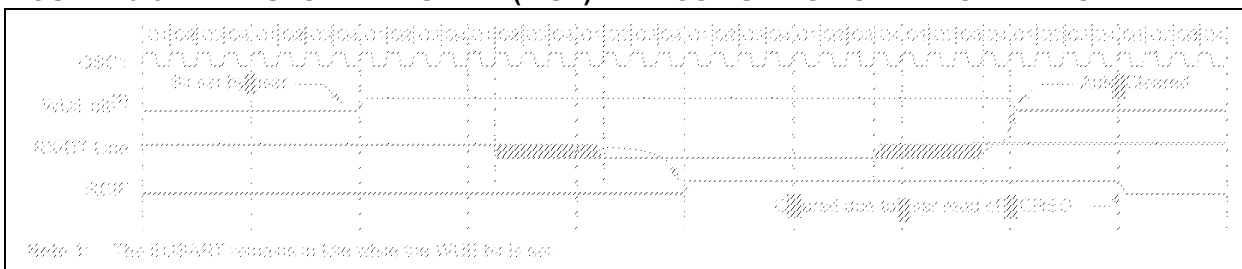
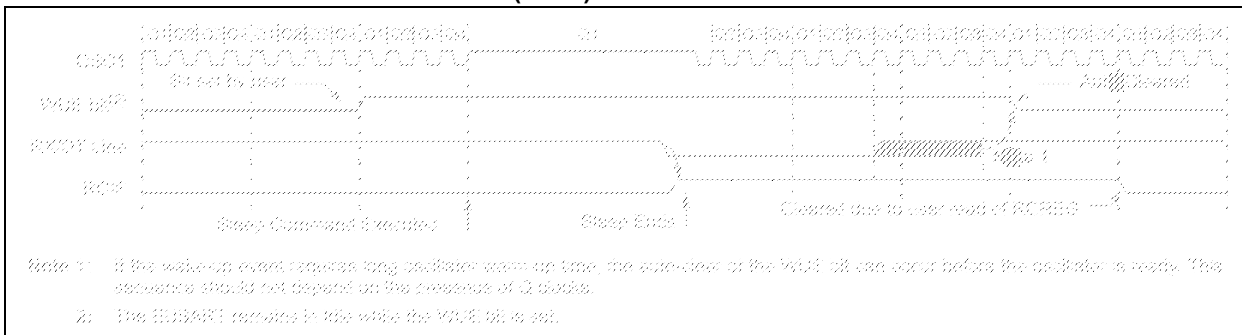


FIGURE 19-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



PIC18F2221/2321/4221/4321 FAMILY

19.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any Idle mode and bit SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREG register; if the RCIE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If the signal from the CK pin is to be inverted, set the TXCKP bit.
4. If 9-bit reception is desired, set bit, RX9.
5. To enable reception, set enable bit, CREN.
6. Flag bit, RCIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCIE, was set.
7. Read the RCSTA register to get the 9th 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 bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 19-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREG	EUSART Receive Register								57
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	57
SPBRGH	EUSART Baud Rate Generator Register High Byte								57
SPBRG	EUSART Baud Rate Generator Register Low Byte								57

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

Note 1: These bits are unimplemented on 28-pin devices and read as ‘0’.

PIC18F2221/2321/4221/4321 FAMILY

NOTES:

PIC18F2221/2321/4221/4321 FAMILY

REGISTER 24-2: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	BORV1 ⁽¹⁾	BORV0 ⁽¹⁾	BOREN1 ⁽²⁾	BOREN0 ⁽²⁾	PWRTEN ⁽²⁾
bit 7							bit 0

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

bit 4-3 **BORV<1:0>:** Brown-out Reset Voltage bits⁽¹⁾

11 = Minimum setting

.

.

.

00 = Maximum setting

bit 2-1 **BOREN<1:0>:** Brown-out Reset Enable bits⁽²⁾

11 = Brown-out Reset enabled in hardware only (SBOREN is disabled)

10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)

01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled)

00 = Brown-out Reset disabled in hardware and software

bit 0 **PWRTEN:** Power-up Timer Enable bit⁽²⁾

1 = PWRT disabled

0 = PWRT enabled

Note 1: See **Section 27.1 “DC Characteristics”** for the specifications.

2: The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18F2221/2321/4221/4321 FAMILY

REGISTER 24-10: CONFIG7L: CONFIGURATION REGISTER 7 LOW (BYTE ADDRESS 30000Ch)

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
—	—	—	—	—	—	EBTR1	EBTR0
bit 7						bit 0	

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **EBTR1:** Table Read Protection bit

1 = Block 1 not protected from table reads executed in other blocks⁽¹⁾

0 = Block 1 protected from table reads executed in other blocks⁽¹⁾

bit 0 **EBTR0:** Table Read Protection bit

1 = Block 0 not protected from table reads executed in other blocks⁽¹⁾

0 = Block 0 protected from table reads executed in other blocks⁽¹⁾

Note 1: See Figure 24-5 for variable block boundaries.

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

REGISTER 24-11: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7						bit 0	

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

bit 6 **EBTRB:** Boot Block Table Read Protection bit

1 = Boot block not protected from table reads executed in other blocks⁽¹⁾

0 = Boot block protected from table reads executed in other blocks⁽¹⁾

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

Note 1: See Figure 24-5 for variable block boundaries.

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18F2221/2321/4221/4321 FAMILY

REGISTER 24-14: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN ⁽¹⁾
bit 7							bit 0

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit⁽¹⁾

1 = Watchdog Timer is on

0 = Watchdog Timer is off

Note 1: This bit has no effect if the Configuration bit, WDTEN, is enabled.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

TABLE 24-2: SUMMARY OF WATCHDOG TIMER REGISTERS

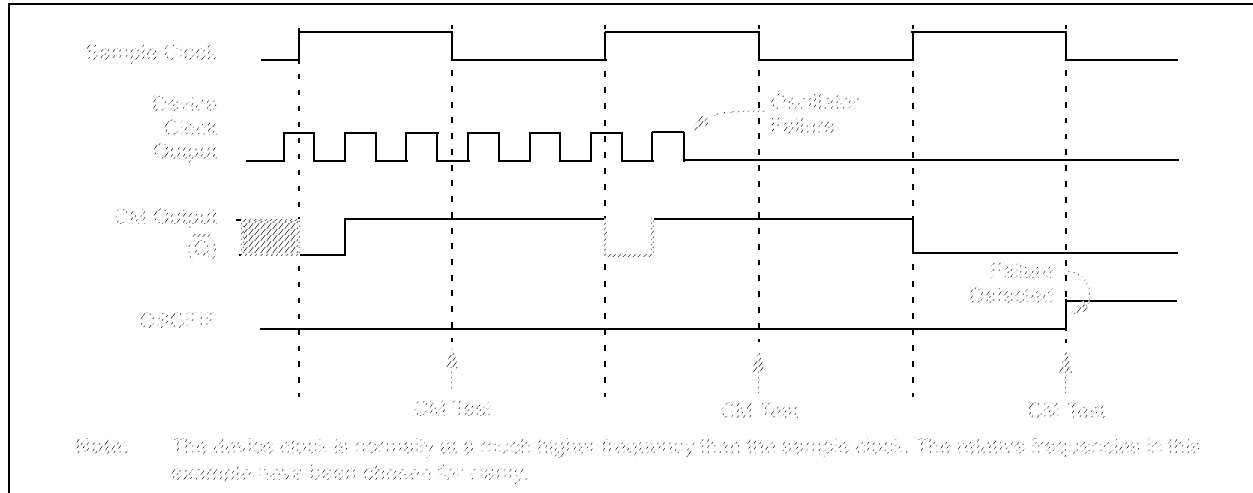
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
RCON	IPEN	SBOREN ⁽¹⁾	—	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}	56
WDTCON	—	—	—	—	—	—	—	SWDTEN	56

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

Note 1: The SBOREN bit is only available when the BOREN<1:0> Configuration bits = 01; otherwise, it is disabled and reads as '0'. See **Section 5.4 “Brown-out Reset (BOR)”**.

PIC18F2221/2321/4221/4321 FAMILY

FIGURE 24-4: FSCM TIMING DIAGRAM



24.4.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexer selects the clock source selected by the OSCCON register. Fail-Safe Monitoring of the power-managed clock source resumes in the power-managed mode.

If an oscillator failure occurs during power-managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTOSC multiplexer. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, subsequent interrupts while in Idle mode will cause the CPU to begin executing instructions while being clocked by the INTOSC source.

24.4.4 POR OR WAKE FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is EC, RC or INTRC modes, monitoring can begin immediately following these events.

For oscillator modes involving a crystal or resonator (HS, HSPLL, LP or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

Note: The same logic that prevents false oscillator failure interrupts on POR, or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 24.3.1 "Special Considerations for Using Two-Speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary clock to become stable. When the new power-managed mode is selected, the primary clock is disabled.

PIC18F2221/2321/4221/4321 FAMILY

BRA Unconditional Branch

Syntax:	BRA n				
Operands:	$-1024 \leq n \leq 1023$				
Operation:	$(PC) + 2 + 2n \rightarrow PC$				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>1101</td><td>0nnn</td><td>nnnn</td><td>nnnn</td></tr></table>	1101	0nnn	nnnn	nnnn
1101	0nnn	nnnn	nnnn		
Description:	Add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is a two-cycle instruction.				
Words:	1				
Cycles:	2				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

Example: HERE BRA Jump

Before Instruction
PC = address (HERE)
After Instruction
PC = address (Jump)

BSF Bit Set f

Syntax:	BSF f, b {,a}			
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$ $a \in [0, 1]$			
Operation:	$1 \rightarrow f$			
Status Affected:	None			
Encoding:	1000	bbba	ffff	ffff
Description:	<p>Bit 'b' in register 'f' is set.</p> <p>If 'a' is '0', the Access Bank is selected.</p> <p>If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>			
Words:	1			
Cycles:	1			

Q Cycle Activity:

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

Example: BSF FLAG_REG, 7, 1

Before Instruction
FLAG_REG = 0Ah
After Instruction
FLAG_REG = 8Ah

PIC18F2221/2321/4221/4321 FAMILY

NEGF

Negate f

Syntax: `NEGF f{,a}`

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

Operation: $(\bar{f}) + 1 \rightarrow f$

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

Encoding:

0110	110a	ffff	ffff
------	------	------	------

Description: Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'.
 If 'a' is '0', the Access Bank is selected.
 If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `NEGF REG, 1`

Before Instruction

REG = 0011 1010 [3Ah]

After Instruction

REG = 1100 0110 [C6h]

NOP

No Operation

Syntax: `NOP`

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000	0000
1111	xxxx	xxxx	xxxx

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

PIC18F2221/2321/4221/4321 FAMILY

RCALL

Relative Call

Syntax:	RCALL n				
Operands:	-1024 ≤ n ≤ 1023				
Operation:	(PC) + 2 → TOS, (PC) + 2 + 2n → PC				
Status Affected:	None				
Encoding:	<table><tr><td>1101</td><td>1nnn</td><td>nnnn</td><td>nnnn</td></tr></table>	1101	1nnn	nnnn	nnnn
1101	1nnn	nnnn	nnnn		
Description:	Subroutine call with a jump up to 1K from the current location. First, return address (PC + 2) is pushed onto the stack. Then, add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is a two-cycle instruction.				
Words:	1				
Cycles:	2				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'n' PUSH PC to stack	Process Data	Write to PC
No operation	No operation	No operation	No operation

Example: HERE RCALL Jump

Before Instruction

PC = Address (HERE)

After Instruction

PC = Address (Jump)

TOS = Address (HERE + 2)

RESET

Reset

Syntax:	RESET				
Operands:	None				
Operation:	Reset all registers and flags that are affected by a MCLR Reset.				
Status Affected:	All				
Encoding:	<table><tr><td>0000</td><td>0000</td><td>1111</td><td>1111</td></tr></table>	0000	0000	1111	1111
0000	0000	1111	1111		
Description:	This instruction provides a way to execute a MCLR Reset in software.				
Words:	1				
Cycles:	1				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Start Reset	No operation	No operation

Example: RESET

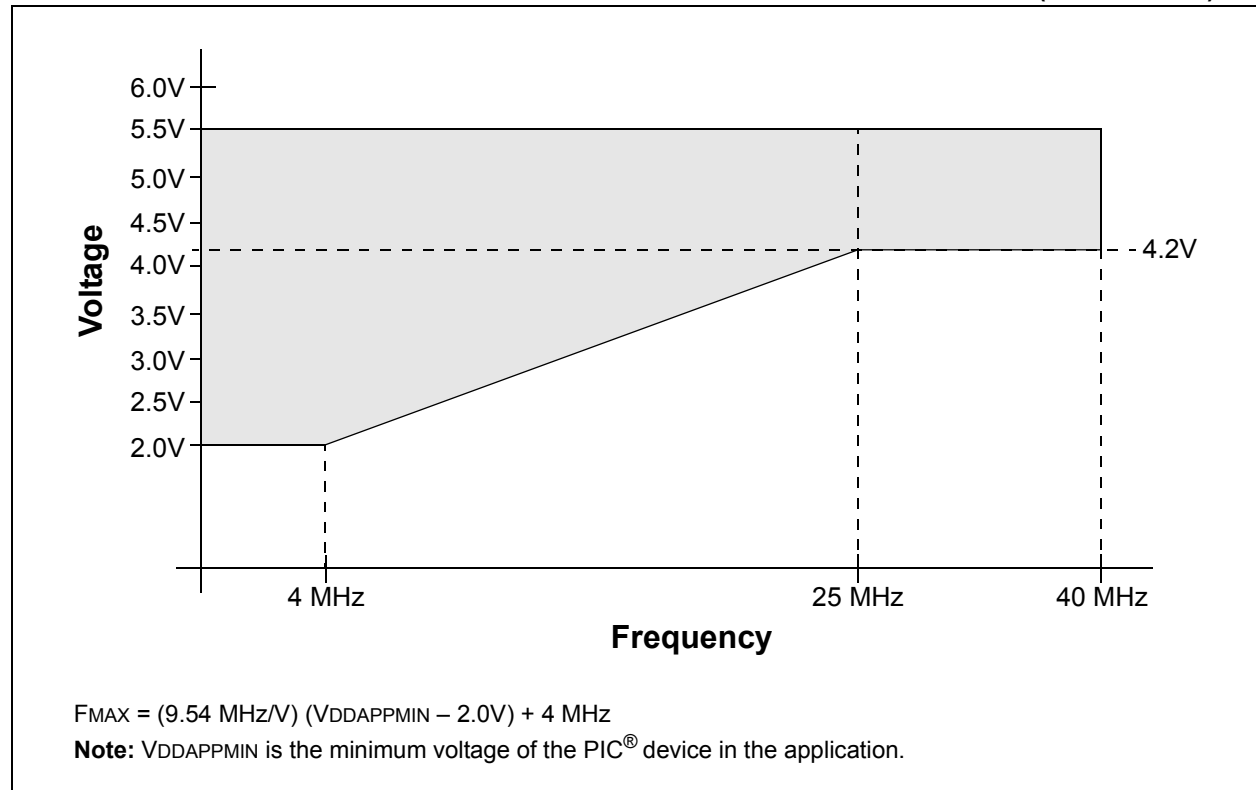
After Instruction

Registers = Reset Value

Flags* = Reset Value

PIC18F2221/2321/4221/4321 FAMILY

FIGURE 27-3: PIC18LF2221/2321/4221/4321 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



PIC18F2221/2321/4221/4321 FAMILY

27.4 AC (Timing) Characteristics

27.4.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created using one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

T		T	
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp		osc	OSC1
cc	CCP1	rd	\overline{RD}
ck	CLKO	rw	\overline{RD} or \overline{WR}
cs	\overline{CS}	sc	SCK
di	SDI	ss	\overline{SS}
do	SDO	t0	T0CKI
dt	Data in	t1	T13CKI
io	I/O port	wr	\overline{WR}
mc	\overline{MCLR}		

Uppercase letters and their meanings:

S		P	Period
F	Fall	R	Rise
H	High	V	Valid
I	Invalid (High-impedance)	Z	High-impedance
L	Low		
I ² C only		High	High
AA	output access	Low	Low
BUF	Bus free		

TCC:ST (I²C specifications only)

CC		SU	Setup
HD	Hold		
ST		STO	Stop condition
DAT	DATA input hold		
STA	Start condition		

PIC18F2221/2321/4221/4321 FAMILY

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at 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, 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
- Development Systems Information Line

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://support.microchip.com>