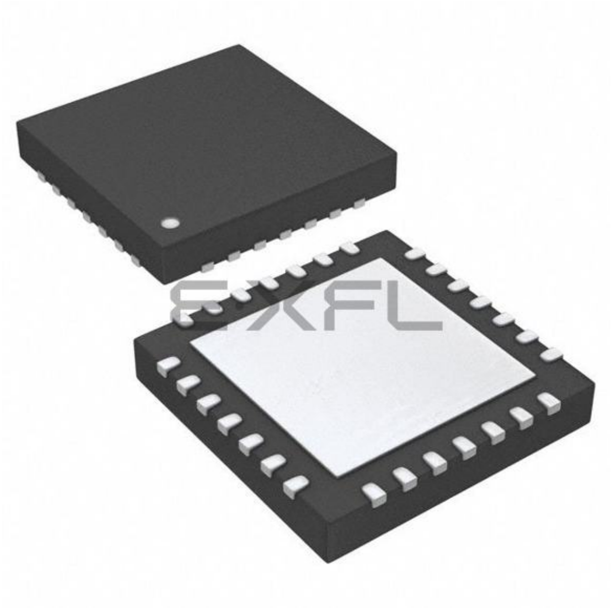


Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18f26k22-e-ml

TABLE 1-2: PIC18(L)F2XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Number		Pin Name	Pin Type	Buffer Type	Description
PDIP, SOIC	QFN, UQFN				
9	6	RA7/CLKI/OSC1			
		RA7	I/O	TTL	Digital I/O.
		CLKI	I	CMOS	External clock source input. Always associated with pin function OSC1.
		OSC1	I	ST	Oscillator crystal input or external clock source input ST buffer when configured in RC mode; CMOS otherwise.
21	18	RB0/INT0/CCP4/FLT0/SRI/$\overline{SS2}$/AN12			
		RB0	I/O	TTL	Digital I/O.
		INT0	I	ST	External interrupt 0.
		CCP4	I/O	ST	Capture 4 input/Compare 4 output/PWM 4 output.
		FLT0	I	ST	PWM Fault input for ECCP Auto-Shutdown.
		SRI	I	ST	SR latch input.
		$\overline{SS2}$	I	TTL	SPI slave select input (MSSP).
		AN12	I	Analog	Analog input 12.
22	19	RB1/INT1/P1C/SCK2/SCL2/C12IN3-/AN10			
		RB1	I/O	TTL	Digital I/O.
		INT1	I	ST	External interrupt 1.
		P1C	O	CMOS	Enhanced CCP1 PWM output.
		SCK2	I/O	ST	Synchronous serial clock input/output for SPI mode (MSSP).
		SCL2	I/O	ST	Synchronous serial clock input/output for I^2C mode (MSSP).
		C12IN3-	I	Analog	Comparators C1 and C2 inverting input.
		AN10	I	Analog	Analog input 10.
23	20	RB2/INT2/CTED1/P1B/SDI2/SDA2/AN8			
		RB2	I/O	TTL	Digital I/O.
		INT2	I	ST	External interrupt 2.
		CTED1	I	ST	CTMU Edge 1 input.
		P1B	O	CMOS	Enhanced CCP1 PWM output.
		SDI2	I	ST	SPI data in (MSSP).
		SDA2	I/O	ST	I^2C data I/O (MSSP).
		AN8	I	Analog	Analog input 8.
24	21	RB3/CTED2/P2A/CCP2/SDO2/C12IN2-/AN9			
		RB3	I/O	TTL	Digital I/O.
		CTED2	I	ST	CTMU Edge 2 input.
		P2A	O	CMOS	Enhanced CCP2 PWM output.
		CCP2**(2)**	I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
		SDO2	O	—	SPI data out (MSSP).
		C12IN2-	I	Analog	Comparators C1 and C2 inverting input.
		AN9	I	Analog	Analog input 9.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

Note 1: Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

2: Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

2.7.1 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a 31.25 kHz internal clock source. The LFINTOSC is not tunable, but is designed to be stable across temperature and voltage. See **Section 27.0 “Electrical Specifications”** for the LFINTOSC accuracy specifications.

The output of the LFINTOSC can be a clock source to the primary clock or the INTOSC clock (see Figure 2-1). The LFINTOSC is also the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

2.7.2 FREQUENCY SELECT BITS (IRCF)

The HFINTOSC (16 MHz) and MFINTOSC (500 MHz) outputs connect to a divide circuit that provides frequencies of 16 MHz to 31.25 kHz. These divide circuit frequencies, along with the 31.25 kHz LFINTOSC output, are multiplexed to provide a single INTOSC clock output (see Figure 2-1). The IRCF<2:0> bits of the OSCCON register, the MFIOSEL bit of the OSCCON2 register and the INTSRC bit of the OSCTUNE register, select the output frequency of the internal oscillators. One of eight frequencies can be selected via software:

- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz (default after Reset)
- 500 kHz (MFINTOSC or HFINTOSC)
- 250 kHz (MFINTOSC or HFINTOSC)
- 31 kHz (LFINTOSC, MFINTOSC or HFINTOSC)

2.7.3 INTOSC FREQUENCY DRIFT

The factory calibrates the internal oscillator block outputs (HFINTOSC/MFINTOSC) for 16 MHz/500 kHz. However, this frequency may drift as V_{DD} or temperature changes. It is possible to adjust the HFINTOSC/MFINTOSC frequency by modifying the value of the TUN<5:0> bits in the OSCTUNE register. This has no effect on the LFINTOSC clock source frequency.

Tuning the HFINTOSC/MFINTOSC source requires knowing when to make the adjustment, in which direction it should be made and, in some cases, how large a change is needed. Three possible compensation techniques are discussed in the following sections. However, other techniques may be used.

2.7.3.1 Compensating with the EUSART

An adjustment may be required when the EUSART begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the device clock frequency is too high; to adjust for this, decrement the value in OSCTUNE to reduce the clock frequency. On the other hand, errors in data may suggest that the clock speed is too low; to compensate, increment OSCTUNE to increase the clock frequency.

2.7.3.2 Compensating with the Timers

This technique compares device clock speed to some reference clock. Two timers may be used; one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator.

Both timers are cleared, but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

2.7.3.3 Compensating with the CCP Module in Capture Mode

A CCP module can use free running Timer1, Timer3 or Timer5 clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is much greater than the calculated time, the internal oscillator block is running too fast; to compensate, decrement the OSCTUNE register. If the measured time is much less than the calculated time, the internal oscillator block is running too slow; to compensate, increment the OSCTUNE register.

4.2 Register Definitions: Reset Control

REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0/0	R/W-q/u	U-0	R/W-1/q	R-1/q	R-1/q	R/W-q/u	R/W-0/q
IPEN	SBOREN(1)	—	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$(2)	$\overline{\text{BOR}}$
bit 7							bit 0

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

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **SBOREN:** BOR Software Enable bit(1)
If BOREN<1:0> = 01:
1 = BOR is enabled
0 = BOR is disabled
If BOREN<1:0> = 00, 10 or 11:
Bit is disabled and read as '0'.

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

bit 4 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
1 = The RESET instruction was not executed (set by firmware or Power-on Reset)
0 = The RESET instruction was executed causing a device Reset (must be set in firmware after a code-executed Reset occurs)

bit 3 **$\overline{\text{TO}}$:** Watchdog Time-out Flag bit
1 = Set by power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 2 **$\overline{\text{PD}}$:** Power-down Detection Flag bit
1 = Set by power-up or by the CLRWDT instruction
0 = Set by execution of the SLEEP instruction

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit(2)
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit(3)
1 = A Brown-out Reset has not occurred (set by firmware only)
0 = A Brown-out Reset occurred (must be set by firmware after a POR or Brown-out Reset occurs)

Note 1: When CONFIG2L[2:1] = 01, then the SBOREN Reset state is '1'; otherwise, it is '0'.
2: The actual Reset value of $\overline{\text{POR}}$ is determined by the type of device Reset. See the notes following this register and **Section 4.7 "Reset State of Registers"** for additional information.
3: See Table 4-1.

Note 1: Brown-out Reset is indicated when $\overline{\text{BOR}}$ is '0' and POR is '1' (assuming that both $\overline{\text{POR}}$ and $\overline{\text{BOR}}$ were set to '1' by firmware immediately after POR).
2: It is recommended that the $\overline{\text{POR}}$ bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
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 **Unimplemented:** Read as ‘0’.

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared by software)
0 = The A/D conversion is not complete or has not been started

bit 5 **RC1IF:** EUSART1 Receive Interrupt Flag bit
1 = The EUSART1 receive buffer, RCREG1, is full (cleared when RCREG1 is read)
0 = The EUSART1 receive buffer is empty

bit 4 **TX1IF:** EUSART1 Transmit Interrupt Flag bit
1 = The EUSART1 transmit buffer, TXREG1, is empty (cleared when TXREG1 is written)
0 = The EUSART1 transmit buffer is full

bit 3 **SSP1IF:** Master Synchronous Serial Port 1 Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared by software)
0 = Waiting to transmit/receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR register capture occurred (must be cleared by software)
0 = No TMR register capture occurred
Compare mode:
1 = A TMR register compare match occurred (must be cleared by software)
0 = No TMR register compare match occurred
PWM mode:
Unused in this mode

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared by software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared by software)
0 = TMR1 register did not overflow

Note 1: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE/GIEH of the INTCON register.

Note: User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

TABLE 10-6: REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	150
ECCP2AS	CCP2ASE	CCP2AS<2:0>			PSS2AC<1:0>		PSS2BD<1:0>		202
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				198
ECCP3AS	CCP3ASE	CCP3AS<2:0>			PSS3AC<1:0>		PSS3BD<1:0>		202
CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<3:0>				198
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	110
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	111
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	153
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	152
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	148
SLRCON	—	—	—	SLRE(1)	SLRD(1)	SLRC	SLRB	SLRA	153
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		167
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSCEN	$\overline{\text{T3SYNC}}$	T3RD16	TMR3ON	166
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		167
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	152

Legend: — = unimplemented locations, read as ‘0’. Shaded bits are not used for PORTB.

Note 1: Available on PIC18(L)F4XK22 devices.

TABLE 10-7: CONFIGURATION REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	348
CONFIG4L	$\overline{\text{DEBUG}}$	XINST	—	—	—	LVP(1)	—	STRVEN	349

Legend: — = unimplemented locations, read as ‘0’. Shaded bits are not used for PORTB.

Note 1: Can only be changed when in high voltage programming mode.

TABLE 10-14: PORTE I/O SUMMARY

Pin	Function	TRIS Setting	ANSEL Setting	Pin Type	Buffer Type	Description
RE0/P3A/CCP3/AN5	RE0	0	0	O	DIG	LATE<0> data output; not affected by analog input.
		1	0	I	ST	PORTE<0> data input; disabled when analog input enabled.
	P3A(1)	0	0	O	DIG	Enhanced CCP3 PWM output.
	CCP3(1)	0	0	O	DIG	Compare 3 output/PWM 3 output.
		1	0	I	ST	Capture 3 input.
	AN5	1	1	I	AN	Analog input 5.
RE1/P3B/AN6	RE1	0	0	O	DIG	LATE<1> data output; not affected by analog input.
		1	0	I	ST	PORTE<1> data input; disabled when analog input enabled.
	P3B	0	0	O	DIG	Enhanced CCP3 PWM output.
	AN6	1	1	I	AN	Analog input 6.
RE2/CCP5/AN7	RE2	0	0	O	DIG	LATE<2> data output; not affected by analog input.
		1	0	I	ST	PORTE<2> data input; disabled when analog input enabled.
	CCP5	0	0	O	DIG	Compare 5 output/PWM 5 output.
		1	0	I	ST	Capture 5 input.
	AN7	1	1	I	AN	Analog input 7.
RE3/VPP/$\overline{\text{MCLR}}$	RE3	—	—	I	ST	PORTE<3> data input; enabled when Configuration bit MCLRE = 0.
	VPP	—	—	P	AN	Programming voltage input; always available
	$\overline{\text{MCLR}}$	—	—	I	ST	Active-low Master Clear (device Reset) input; enabled when configuration bit MCLRE = 1.

Legend: AN = Analog input or output; TTL = TTL compatible input; HV = High Voltage; OD = Open Drain; XTAL = Crystal; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I^2C = Schmitt Trigger input with I^2C.

Note 1: Alternate pin assignment for P3A/CCP3 when Configuration bit CCP3MX is clear.

REGISTER 10-4: ANSELB – PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **ANSB<5:0>:** RB<5:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

REGISTER 10-5: ANSELC – PORTC ANALOG SELECT REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0
ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—
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 7-2 **ANSC<7:2>:** RC<7:2> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 10-6: ANSELD – PORTD ANALOG SELECT 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
ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
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 7-0 **ANSD<7:0>:** RD<7:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

TABLE 14-10: REGISTERS ASSOCIATED WITH STANDARD PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				198
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				198
CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<3:0>				198
CCP4CON	—	—	DC4B<1:0>		CCP4M<3:0>				198
CCP5CON	—	—	DC5B<1:0>		CCP5M<3:0>				198
CCPTMRS0	C3TSEL<1:0>		—	C2TSEL<1:0>		—	C1TSEL<1:0>		201
CCPTMRS1	—	—	—	—	C5TSEL<1:0>		C4TSEL<1:0>		201
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	122
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP	124
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	118
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE	120
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	113
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF	115
PMD0	UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	52
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
PR2	Timer2 Period Register								—
PR4	Timer4 Period Register								—
PR6	Timer6 Period Register								—
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		166
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		166
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		166
TMR2	Timer2 Register								—
TMR4	Timer4 Register								—
TMR6	Timer6 Register								—
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	151
TRISE	WPUE3	—	—	—	—	TRISE2(1)	TRISE1(1)	TRISE0(1)	151

Legend: — = Unimplemented location, read as '0'. Shaded bits are not used by Standard PWM mode.

Note 1: These registers/bits are available on PIC18(L)F4XK22 devices.

TABLE 14-11: CONFIGURATION REGISTERS ASSOCIATED WITH STANDARD PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	348

Legend: — = Unimplemented location, read as '0'. Shaded bits are not used by Standard PWM mode.

15.3 I^{2}C Mode Overview

The Inter-Integrated Circuit Bus (I^{2}C) is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

The I^{2}C bus specifies two signal connections:

- Serial Clock (SCLx)
- Serial Data (SDAx)

Figure 15-2 shows the block diagram of the MSSPx module when operating in I^{2}C mode.

Both the SCLx and SDAx connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 15-11 shows a typical connection between two processors configured as master and slave devices.

The I^{2}C bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

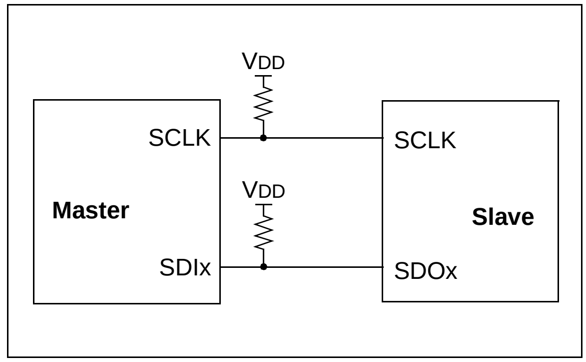
- Master Transmit mode
  (master is transmitting data to a slave)
- Master Receive mode
  (master is receiving data from a slave)
- Slave Transmit mode
  (slave is transmitting data to a master)
- Slave Receive mode
  (slave is receiving data from the master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an $\overline{\text{ACK}}$. The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDAx line while the SCLx line is held high. Address and data bytes are sent out, Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

FIGURE 15-11: I^{2}C MASTER/ SLAVE CONNECTION



The Acknowledge bit ($\overline{\text{ACK}}$) is an active-low signal, which holds the SDAx line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of data bits is always performed while the SCLx line is held low. Transitions that occur while the SCLx line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, then it repeatedly sends out a byte of data, with the slave responding after each byte with an $\overline{\text{ACK}}$ bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, then it repeatedly receives a byte of data from the slave, and responds after each byte with an $\overline{\text{ACK}}$ bit. In this example, the master device is in Master Receive mode and the slave is Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last $\overline{\text{ACK}}$ bit. A Stop bit is indicated by a low-to-high transition of the SDAx line while the SCLx line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last $\overline{\text{ACK}}$ bit when it is in receive mode.

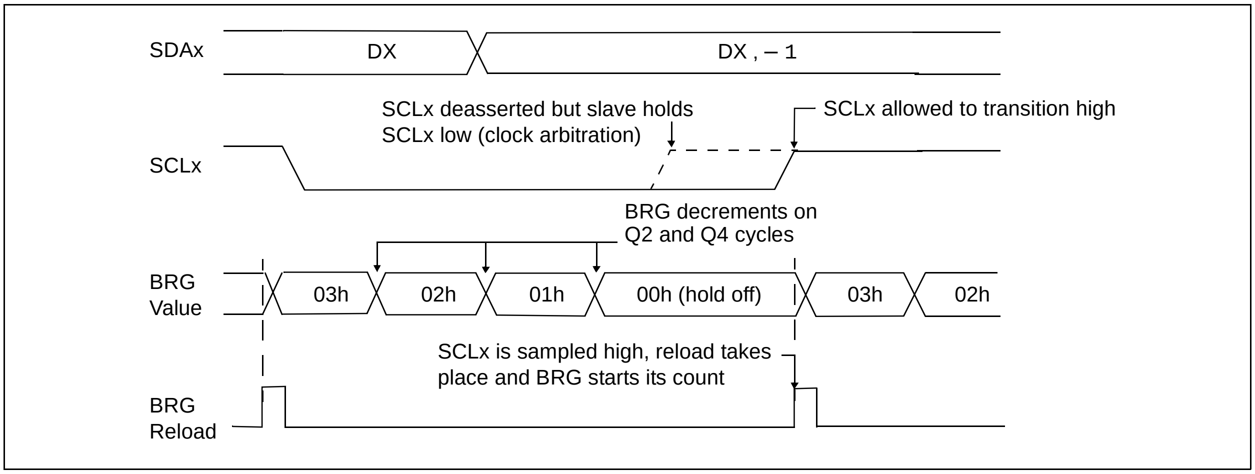
The I^{2}C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.

15.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCLx pin (SCLx allowed to float high). When the SCLx pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCLx pin is actually sampled high. When the SCLx pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCLx high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 15-25).

FIGURE 15-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



15.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not Idle.

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPxCON2 is disabled until the Start condition is complete.

16.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (HFINTOSC). However, the HFINTOSC frequency may drift as V_{DD} or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the HFINTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 2.6 “Internal Clock Modes”** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 16.4.1 “Auto-Baud Detect”**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

16.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 16.5.1.6 “Synchronous Master Reception”**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a “don't care” in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREGx register. If the RCxIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE/GIEH bit is also set, the program will branch to the interrupt vector.

16.5.2.4 Synchronous Slave Reception Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Set the RXx/DTx and TXx/CKx TRIS controls to ‘1’.
3. If using interrupts, ensure that the GIE/GIEH and PEIE/GIEL bits of the INTCON register are set and set the RCxIE bit.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RCxIF bit will be set when reception is complete. An interrupt will be generated if the RCxIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTAx register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREGx register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTAx register or by clearing the SPEN bit which resets the EUSART.

TABLE 16-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON1	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	271
BAUDCON2	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	271
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PMD0	UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	52
RCREG1	EUSART1 Receive Register								—
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	270
RCREG2	EUSART2 Receive Register								—
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	270
SPBRG1	EUSART1 Baud Rate Generator, Low Byte								—
SPBRGH1	EUSART1 Baud Rate Generator, High Byte								—
SPBRG2	EUSART2 Baud Rate Generator, Low Byte								—
SPBRGH2	EUSART2 Baud Rate Generator, High Byte								—
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	269
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	269

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

EXAMPLE 19-2: CURRENT CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 500                           //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027                           //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA
#define ADSCALE 1023                        //for unsigned conversion 10 sig bits
#define ADREF 3.3                           //Vdd connected to A/D Vr+

int main(void)
{
    int i;
    int j = 0;                              //index for loop
    unsigned int Vread = 0;
    double VTot = 0;
    float Vavg=0, Vcal=0, CTMUISrc = 0;     //float values stored for calcs

//assume CTMU and A/D have been set up correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONHbits.CTMUEN = 1;                    //Enable the CTMU
CTMUCONLbits.EDG1STAT = 0;                  // Set Edge status bits to zero
CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;           //drain charge on the circuit
        DELAY;                              //wait 125us
        CTMUCONHbits.IDISSEN = 0;           //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;          //Begin charging the circuit
                                            //using CTMU current source
        DELAY;                              //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;          //Stop charging circuit

        PIR1bits.ADIF = 0;                  //make sure A/D Int not set
        ADCON0bits.GO=1;                    //and begin A/D conv.
        while(!PIR1bits.ADIF);              //Wait for A/D convert complete

        Vread = ADRES;                      //Get the value from the A/D
        PIR1bits.ADIF = 0;                  //Clear A/D Interrupt Flag
        VTot += Vread;                      //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);            //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;                   //CTMUISrc is in 1/100ths of uA

}
```


19.4 Measuring Capacitance with the CTMU

There are two separate methods of measuring capacitance with the CTMU. The first is the absolute method, in which the actual capacitance value is desired. The second is the relative method, in which the actual capacitance is not needed, rather an indication of a change in capacitance is required.

19.4.1 ABSOLUTE CAPACITANCE MEASUREMENT

For absolute capacitance measurements, both the current and capacitance calibration steps found in **Section 19.3 "Calibrating the CTMU Module"** should be followed. Capacitance measurements are then performed using the following steps:

1. Initialize the A/D Converter.
2. Initialize the CTMU.
3. Set EDG1STAT.
4. Wait for a fixed delay, T.
5. Clear EDG1STAT.
6. Perform an A/D conversion.
7. Calculate the total capacitance, $C_{TOTAL} = (I * T)/V$, where I is known from the current source measurement step (see **Section 19.3.1 "Current Source Calibration"**), T is a fixed delay and V is measured by performing an A/D conversion.
8. Subtract the stray and A/D capacitance (C_{OFFSET} from **Section 19.3.2 "Capacitance Calibration"**) from C_{TOTAL} to determine the measured capacitance.

19.4.2 RELATIVE CHARGE MEASUREMENT

An application may not require precise capacitance measurements. For example, when detecting a valid press of a capacitance-based switch, detecting a relative change of capacitance is of interest. In this type of application, when the switch is open (or not touched), the total capacitance is the capacitance of the combination of the board traces, the A/D Converter, etc. A larger voltage will be measured by the A/D Converter. When the switch is closed (or is touched), the total capacitance is larger due to the addition of the capacitance of the human body to the above listed capacitances, and a smaller voltage will be measured by the A/D Converter.

Detecting capacitance changes is easily accomplished with the CTMU using these steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT.
3. Wait for a fixed delay.
4. Clear EDG1STAT.
5. Perform an A/D conversion.

The voltage measured by performing the A/D conversion is an indication of the relative capacitance. Note that in this case, no calibration of the current source or circuit capacitance measurement is needed. See Example 19-4 for a sample software routine for a capacitive touch switch.

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

27.4 DC Characteristics: RC Idle Supply Current, PIC18(L)F2X/4XK22

PIC18LF2X/4XK22		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +125°C					
PIC18F2X/4XK22		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Device Characteristics**	**Typ**	**Max**	**Units**	**Conditions**		
D045	Supply Current (IDD)(1),(2)	0.5	18	μA	-40°C	VDD = 1.8V	FOSC = 31 kHz (**RC_IDLE** mode, LFINTOSC source)
		0.6	18	μA	+25°C		
		0.7	—	μA	+60°C		
		0.75	20	μA	+85°C		
		2.3	22	μA	+125°C		
D046		1.1	20	μA	-40°C	VDD = 3.0V	
		1.2	20	μA	+25°C		
		1.3	—	μA	+60°C		
		1.4	22	μA	+85°C		
		3.2	25	μA	+125°C		
D047		17	30	μA	-40°C	VDD = 2.3V	FOSC = 31 kHz (**RC_IDLE** mode, LFINTOSC source)
		13	30	μA	+25°C		
		14	30	μA	+85°C		
		15	45	μA	+125°C		
D048		19	35	μA	-40°C	VDD = 3.0V	
		15	35	μA	+25°C		
		16	35	μA	+85°C		
		17	50	μA	+125°C		
D049		21	40	μA	-40°C	VDD = 5.0V	
		15	40	μA	+25°C		
		16	40	μA	+85°C		
		18	60	μA	+125°C		
D050		0.11	0.20	mA	-40°C to +125°C	VDD = 1.8V	FOSC = 500 kHz (**RC_IDLE** mode, MFINTOSC source)
D051		0.12	0.25	mA	-40°C to +125°C	VDD = 3.0V	
D052		0.14	0.21	mA	-40°C to +125°C	VDD = 2.3V	FOSC = 500 kHz (**RC_IDLE** mode, MFINTOSC source)
D053		0.15	0.25	mA	-40°C to +125°C	VDD = 3.0V	
D054		0.20	0.31	mA	-40°C to +125°C	VDD = 5.0V	

Note 1: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
Test condition: All Peripheral Module Control bits in PMD0, PMD1 and PMD2 set to '1'.

2: The test conditions for all IDD measurements in active operation mode are:
All I/O pins set as outputs driven to Vss;
MCLR = VDD;
OSC1 = external square wave, from rail-to-rail (PRI_RUN and PRI_IDLE only).

FIGURE 28-60: PIC18LF2X/4XK22 TYPICAL IDD: PRI_IDLE EC MEDIUM POWER

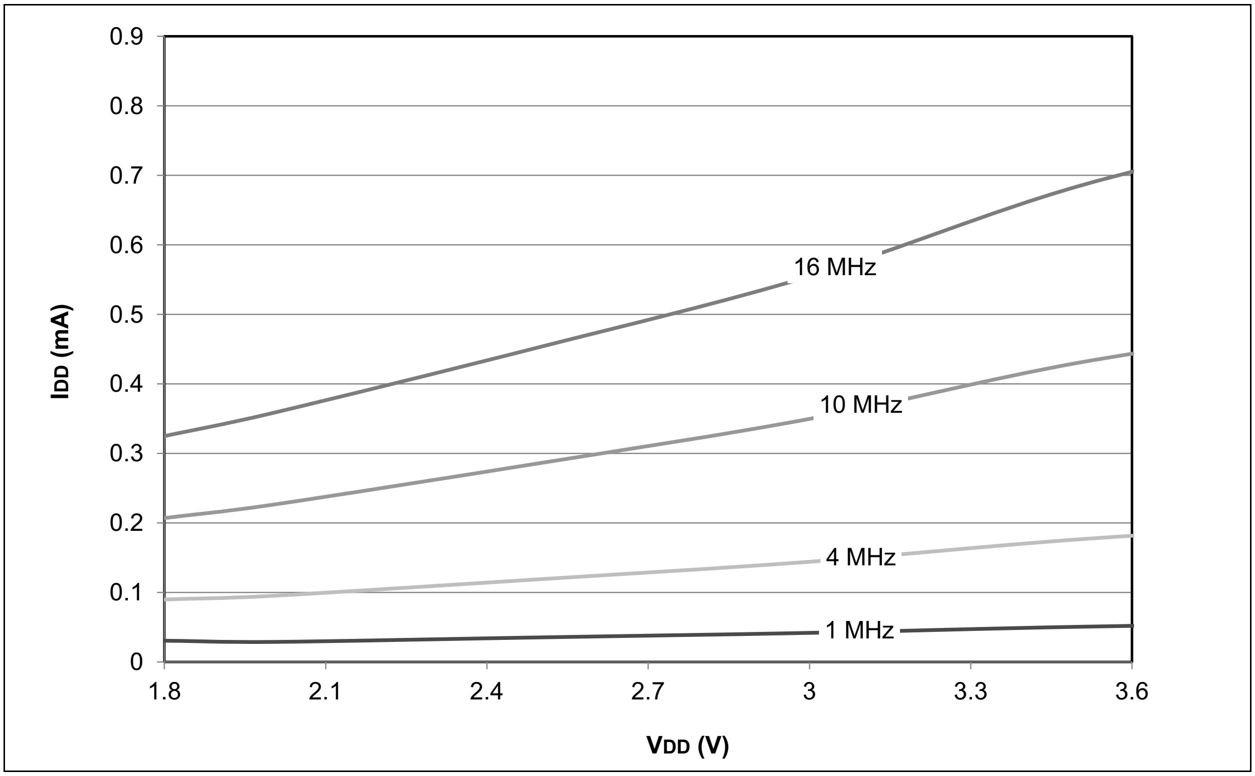


FIGURE 28-61: PIC18LF2X/4XK22 MAXIMUM IDD: PRI_IDLE EC MEDIUM POWER

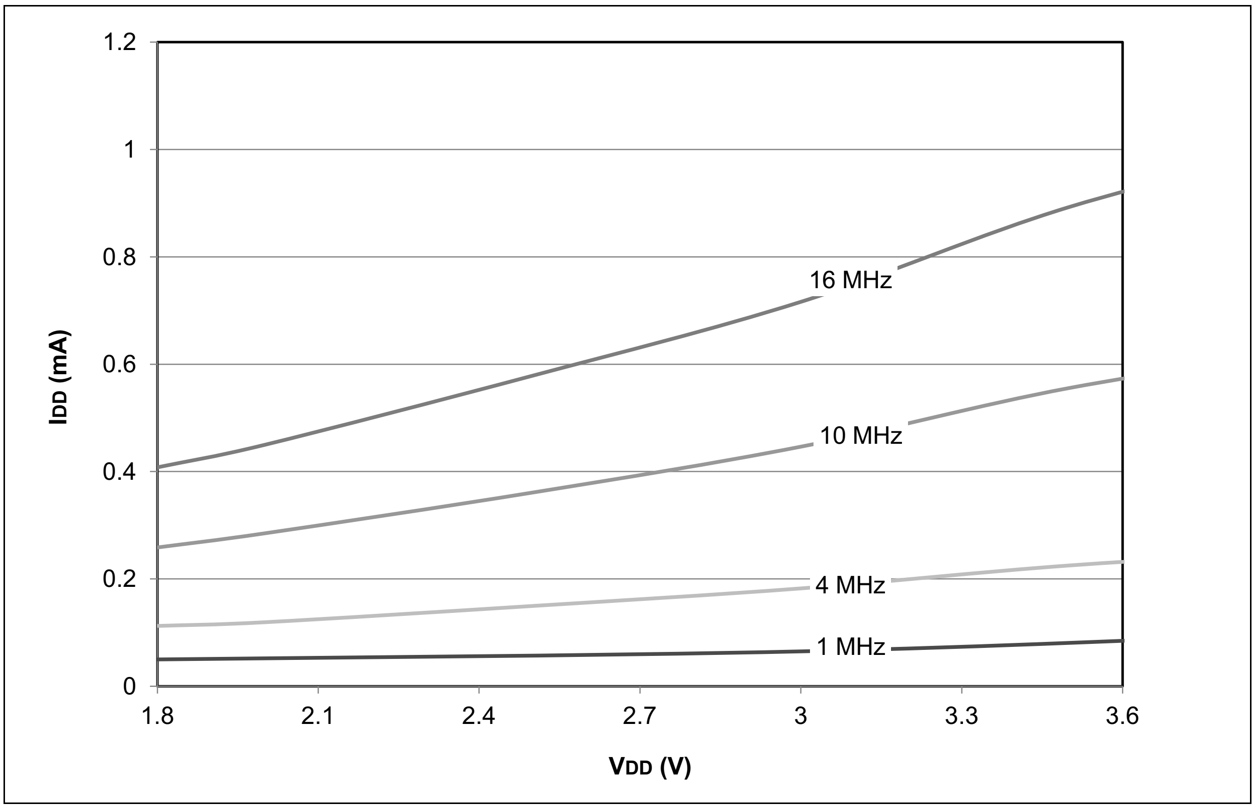


FIGURE 28-68: PIC18LF2X/4XK22 TYPICAL IDD: PRI_IDLE EC with PLL

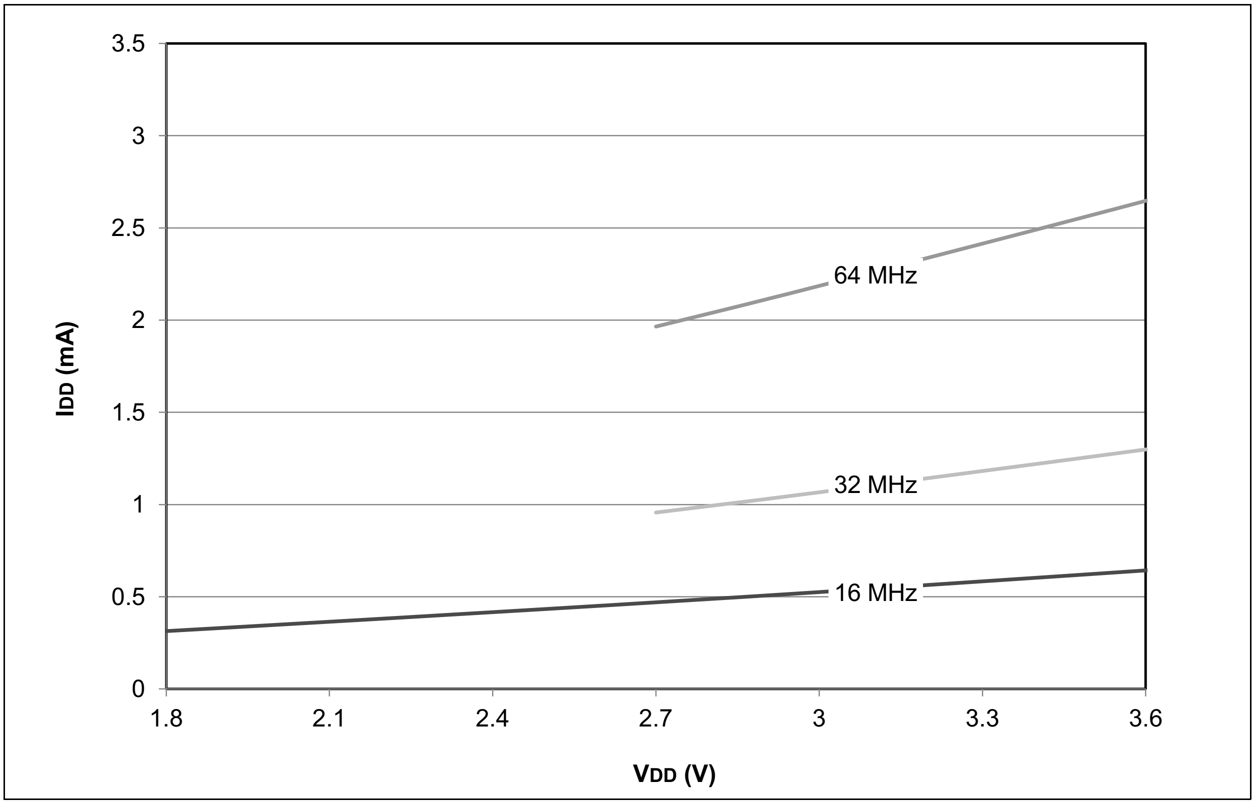


FIGURE 28-69: PIC18LF2X/4XK22 MAXIMUM IDD: PRI_IDLE EC with PLL

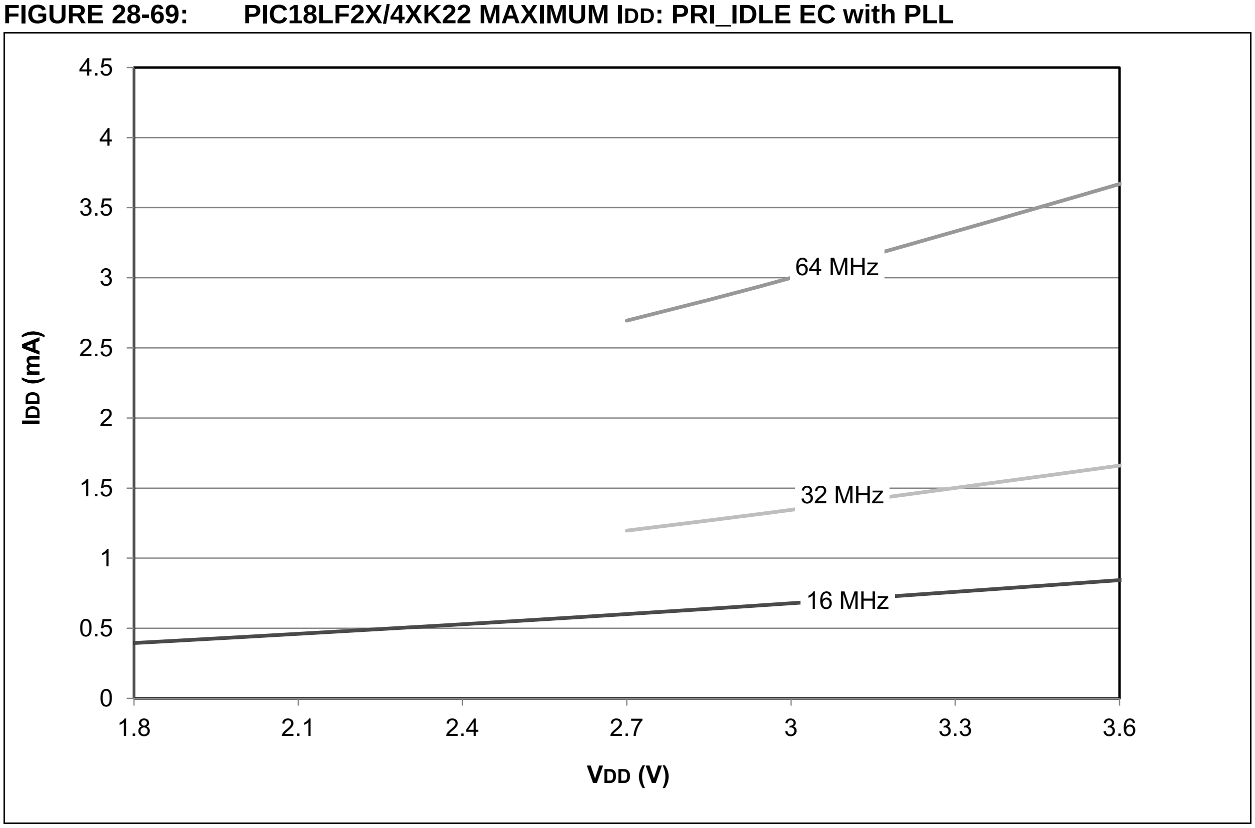
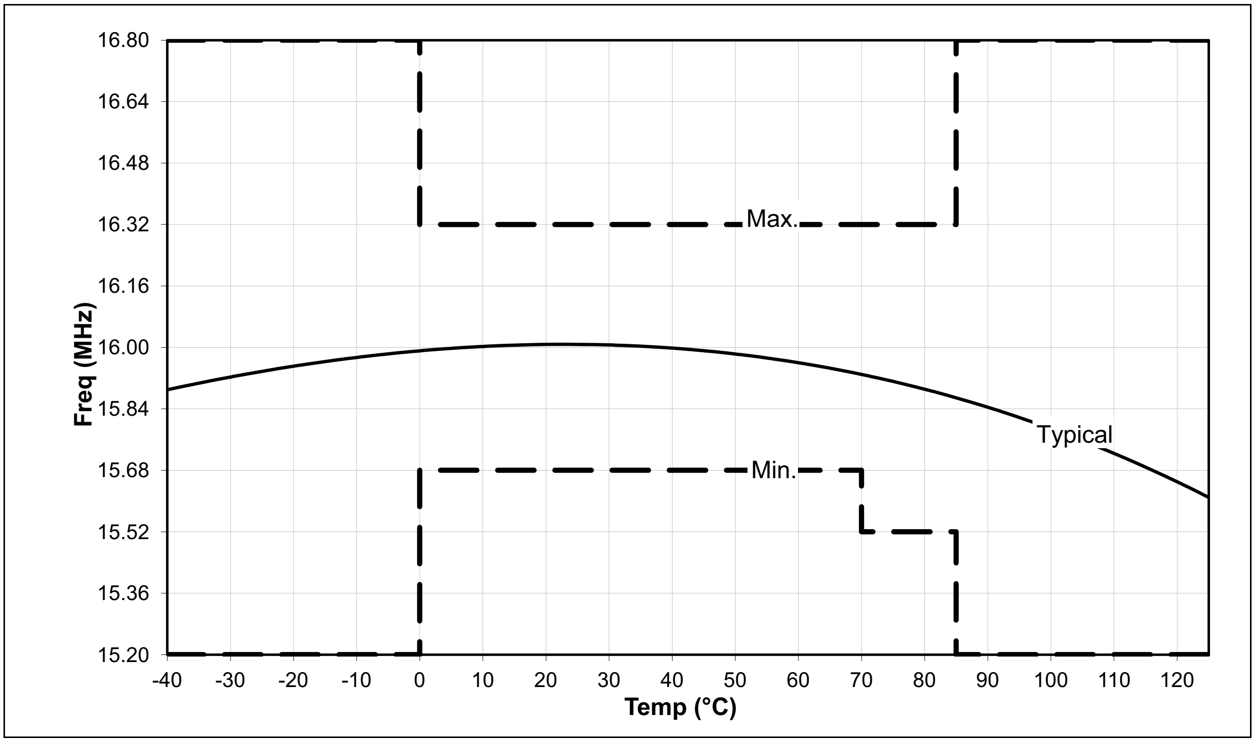
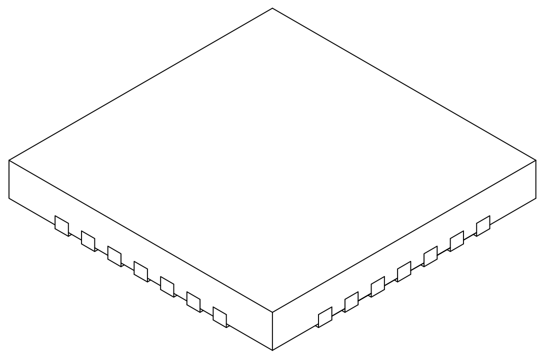


FIGURE 28-100: PIC18(L)F2X/4XK22 HF-INTOSC FREQUENCY vs. TEMPERATURE at 16 MHz
MIN / MAX: ± 2%, T = 0°C to +70°C
+2% / -3%, T = +70°C to +85°C
± 5%, T = -40°C to 0°C and +85°C to +125°C



28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

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

Microchip Technology Drawing C04-152A Sheet 2 of 2