



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

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

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
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)	2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4321-i-ml

PIC18F2221/2321/4221/4321 FAMILY

2.0 GUIDELINES FOR GETTING STARTED WITH PIC18F MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC18F2221/2321/4221/4321 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins (see **Section 2.2 “Power Supply Pins”**)
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used (see **Section 2.2 “Power Supply Pins”**)
- MCLR pin (see **Section 2.3 “Master Clear (MCLR) Pin”**)

These pins must also be connected if they are being used in the end application:

- PGC/PGD pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see **Section 2.4 “ICSP Pins”**)
- OSCI and OSCO pins when an external oscillator source is used (see **Section 2.5 “External Oscillator Pins”**)

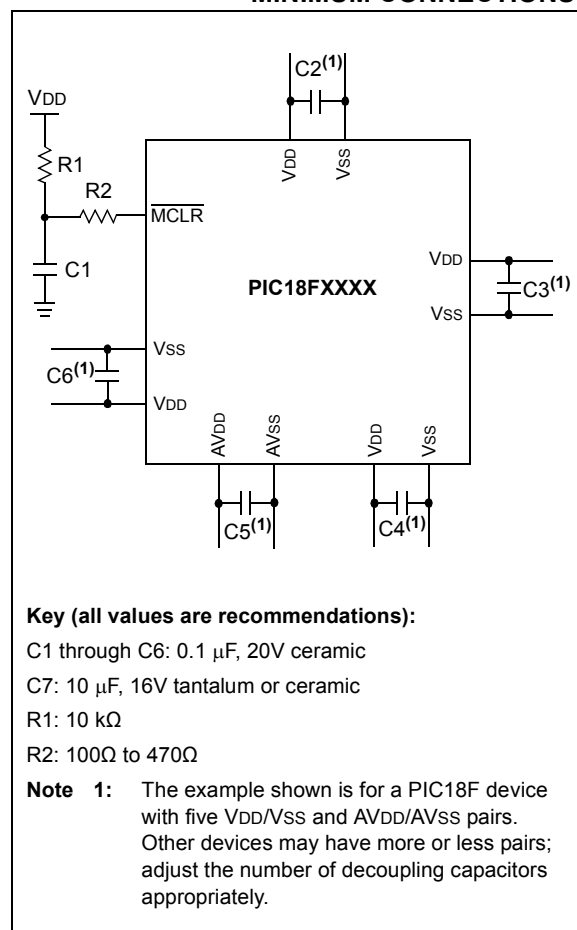
Additionally, the following pins may be required:

- VREF+/VREF- pins used when external voltage reference for analog modules is implemented

Note: The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



PIC18F2221/2321/4221/4321 FAMILY

3.4 RC Oscillator

For timing insensitive applications, the RC and RCIO Oscillator modes offer additional cost savings. The actual oscillator frequency is a function of several factors:

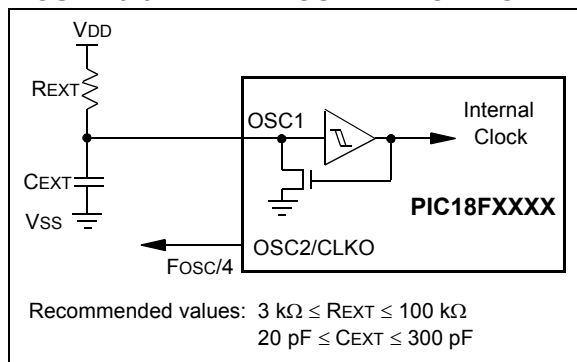
- supply voltage
- values of the external resistor (R_{EXT}) and capacitor (C_{EXT})
- operating temperature

Given the same device, operating voltage, temperature and component values, there will also be unit-to-unit frequency variations. These are due to factors such as:

- normal manufacturing variation
- difference in lead frame capacitance between package types (especially for low C_{EXT} values)
- variations within the tolerance of limits of R_{EXT} and C_{EXT}

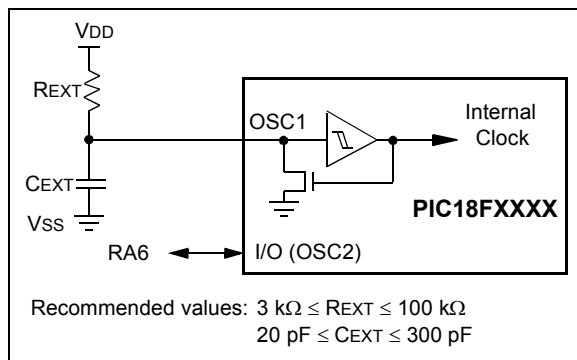
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 3-5 shows how the R/C combination is connected.

FIGURE 3-5: RC OSCILLATOR MODE



The RCIO Oscillator mode (Figure 3-6) functions like the RC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

FIGURE 3-6: RCIO OSCILLATOR MODE



3.5 PLL Frequency Multiplier

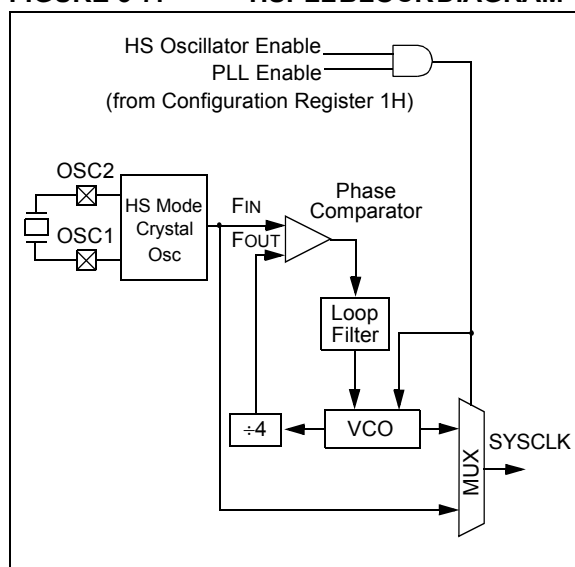
A Phase Locked Loop (PLL) circuit is provided as an option for users who wish to use a lower frequency oscillator circuit or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals or users who require higher clock speeds from an internal oscillator.

3.5.1 HSPLL OSCILLATOR MODE

The HSPLL mode makes use of the HS mode oscillator for frequencies up to 10 MHz. A PLL then multiplies the oscillator output frequency by 4 to produce an internal clock frequency up to 40 MHz. The PLEN bit is not available when this mode is configured as the primary clock source.

The PLL is only available to the crystal oscillator when the FOSC<3:0> Configuration bits are programmed for HSPLL mode (= 0110).

FIGURE 3-7: HSPLL BLOCK DIAGRAM



3.5.2 PLL AND INTOSC

The PLL is also available to the internal oscillator block when the internal oscillator block is configured as the primary clock source. In this configuration, the PLL is enabled in software and generates a clock output of up to 32 MHz. The operation of INTOSC with the PLL is described in **Section 3.6.4 “PLL in INTOSC Modes”**.

PIC18F2221/2321/4221/4321 FAMILY

6.4.3.1 FSR Registers and the INDF Operand

At the core of indirect addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. The four upper bits of the FSRnH register are not used so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as “virtual” registers: they are mapped in the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction’s target. The INDF operand is just a convenient way of using the pointer.

Because indirect addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

6.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

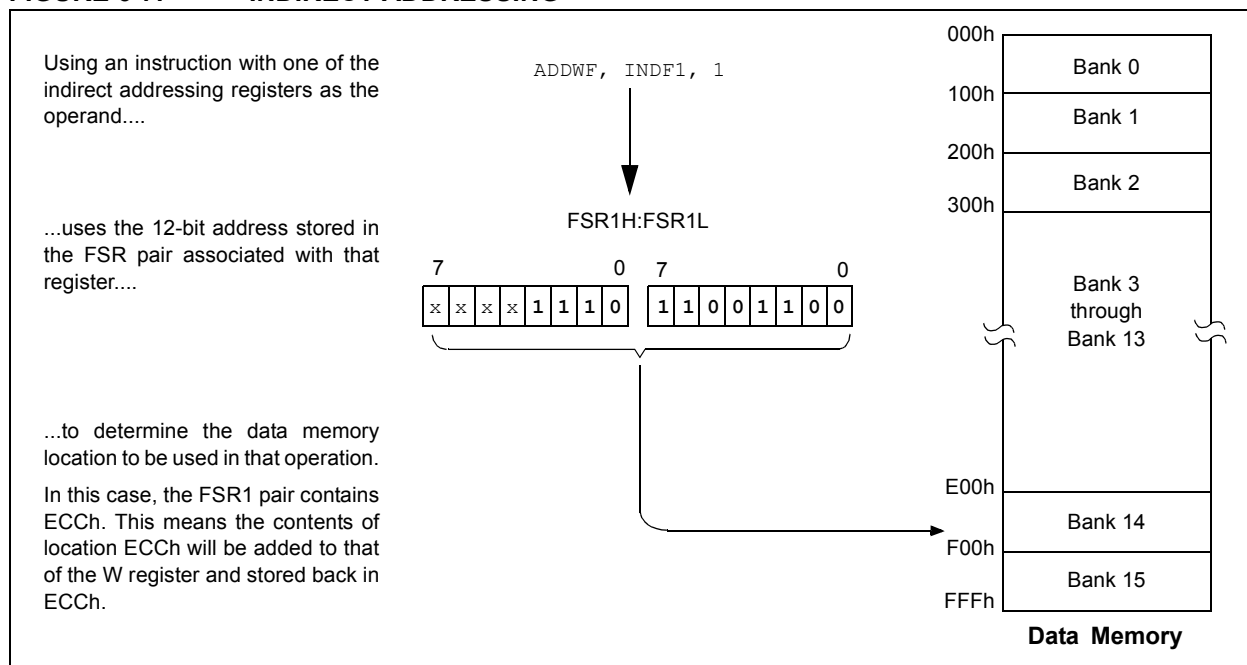
In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- POSTDEC: accesses the FSR value, then automatically decrements it by 1 afterwards
- POSTINC: accesses the FSR value, then automatically increments it by 1 afterwards
- PREINC: increments the FSR value by 1, then uses it in the operation
- PLUSW: adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation.

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by that in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, roll-overs of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

FIGURE 6-7: INDIRECT ADDRESSING



PIC18F2221/2321/4221/4321 FAMILY

7.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

7.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

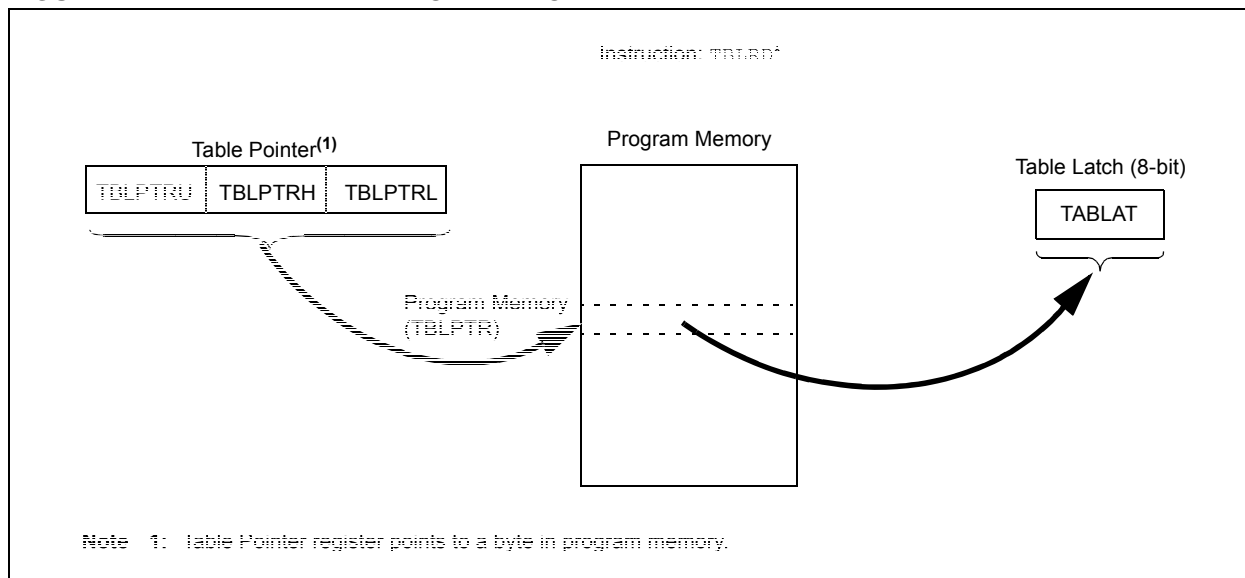
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and place it into the data RAM space. Figure 7-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 7.5 “Writing to Flash Program Memory”**. Figure 7-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word-aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned.

FIGURE 7-1: TABLE READ OPERATION



PIC18F2221/2321/4221/4321 FAMILY

NOTES:

PIC18F2221/2321/4221/4321 FAMILY

11.2 PORTB, TRISB and LATB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB.

EXAMPLE 11-2: INITIALIZING PORTB

```
CLRF    PORTB    ; Initialize PORTB by
                  ; clearing output
                  ; data latches
CLRF    LATB     ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   0Fh      ; Set RB<4:0> as
MOVWF   ADCON1   ; digital I/O pins
                  ; (required if config bit
                  ; PBDEN is set)
MOVLW   0CFh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISB    ; Set RB<3:0> as inputs
                  ; RB<5:4> as outputs
                  ; RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit, $\overline{\text{RBPU}}$ (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Note: On a Power-on Reset, RB<4:0> are configured as analog inputs by default and read as '0'; RB<7:5> are configured as digital inputs.

By clearing the Configuration bit, PBDEN, RB<4:0> will alternatively be configured as digital inputs on POR.

Four of the PORTB pins (RB<7:4>) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB<7:4>) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from Sleep mode or any of the Idle modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB (except with the `MOVFF (ANY), PORTB` instruction).
- 1 Tcy.
- Clear flag bit, RBIF.

A mismatch condition will continue to set flag bit, RBIF. Reading PORTB and waiting 1 Tcy will end the mismatch condition and allow flag bit, RBIF, to be cleared. Also, if the port pin returns to its original state, the mismatch condition will be cleared.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

RB3 can be configured by the Configuration bit, CCP2MX, as the alternate peripheral pin for the CCP2 module (CCP2MX = 0).

PIC18F2221/2321/4221/4321 FAMILY

NOTES:

PIC18F2221/2321/4221/4321 FAMILY

13.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Reset on CCP Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 13-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 13-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 13-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 13-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

- bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
 1 = Enables register read/write of Timer1 in one 16-bit operation
 0 = Enables register read/write of Timer1 in two 8-bit operations
- bit 6 **T1RUN:** Timer1 System Clock Status bit
 1 = Device clock is derived from Timer1 oscillator
 0 = Device clock is derived from another source
- bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
 11 = 1:8 Prescale value
 10 = 1:4 Prescale value
 01 = 1:2 Prescale value
 00 = 1:1 Prescale value
- bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
 1 = Timer1 oscillator is enabled
 0 = Timer1 oscillator is shut off
 The oscillator inverter and feedback resistor are turned off to eliminate power drain.
- bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input
When TMR1CS = 0:
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 **TMR1CS:** Timer1 Clock Source Select bit
 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
 0 = Internal clock (Fosc/4)
- bit 0 **TMR1ON:** Timer1 On bit
 1 = Enables Timer1
 0 = Stops Timer1

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

PIC18F2221/2321/4221/4321 FAMILY

18.4 I²C Mode

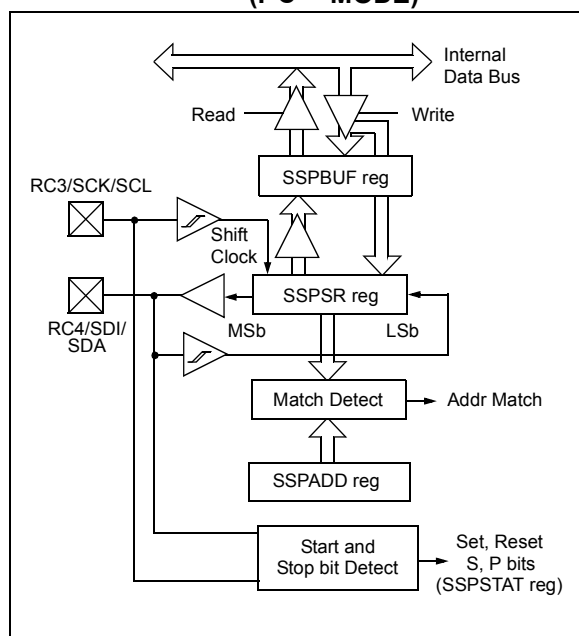
The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RC3/SCK/SCL
- Serial data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

FIGURE 18-7: MSSP BLOCK DIAGRAM (I²C™ MODE)



18.4.1 REGISTERS

The MSSP module has six registers for I²C operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON1, SSPCON2 and SSPSTAT are the control and status registers in I²C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the MSSP is configured in I²C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

PIC18F2221/2321/4221/4321 FAMILY

REGISTER 18-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C™ MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPBUF register was attempted while the I²C™ conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

In Slave Transmit mode:

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

In Receive mode (Master or Slave modes):

This is a “don’t care” bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)

0 = No overflow

In Transmit mode:

This is a “don’t care” bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit

1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, the SDA and SCL pins must be properly configured as inputs.

bit 4 **CKP:** SCK Release Control bit

In Slave mode:

1 = Release clock

0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:

Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits

1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled

1011 = I²C Firmware Controlled Master mode (slave Idle)

1000 = I²C Master mode, clock = Fosc/(4 * (SSPAD + 1))

0111 = I²C Slave mode, 10-bit address

0110 = I²C Slave mode, 7-bit address

Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

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

PIC18F2221/2321/4221/4321 FAMILY

18.4.10 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameter 106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification parameter 107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time if an address match occurred, or if data was received properly. The status of ACK is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 18-23).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

18.4.10.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

18.4.10.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL flag is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 Tcy after the SSPBUF write. If SSPBUF is rewritten within 2 Tcy, the WCOL bit is set and SSPBUF is updated. This may result in a corrupted transfer. The user should verify that the WCOL flag is clear after each write to SSPBUF to ensure the transfer is correct.

18.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge (ACK = 0) and is set when the slave does not Acknowledge (ACK = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

18.4.11 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

Note:	The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.
--------------	--

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

18.4.11.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

18.4.11.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

18.4.11.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

PIC18F2221/2321/4221/4321 FAMILY

TABLE 19-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

PIC18F2221/2321/4221/4321 FAMILY

RLNCF Rotate Left f (No Carry)

Syntax: RLNCF f {,d {,a}}

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

Operation: $(f \ll n) \rightarrow \text{dest} \ll n + 1$,
 $(f \ll 7) \rightarrow \text{dest} \ll 0$

Status Affected: N, Z

Encoding:

0100	01da	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). 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 to destination

Example: RLNCF REG, 1, 0

Before Instruction

REG = 1010 1011

After Instruction

REG = 0101 0111

RRCF Rotate Right f through Carry

Syntax: RRCF f {,d {,a}}

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

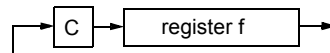
Operation: $(f \gg n) \rightarrow \text{dest} \gg n - 1$,
 $(f \gg 0) \rightarrow C$,
 $(C) \rightarrow \text{dest} \gg 7$

Status Affected: C, N, Z

Encoding:

0011	00da	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). 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 to destination

Example: RRCF REG, 0, 0

Before Instruction

REG = 1110 0110
 C = 0

After Instruction

REG = 1110 0110
 W = 0111 0011
 C = 0

PIC18F2221/2321/4221/4321 FAMILY

SUBLW	Subtract W from Literal				
Syntax:	SUBLW k				
Operands:	$0 \leq k \leq 255$				
Operation:	$k - (W) \rightarrow W$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table><tr><td>0000</td><td>1000</td><td>kkkk</td><td>kkkk</td></tr></table>	0000	1000	kkkk	kkkk
0000	1000	kkkk	kkkk		
Description	W is subtracted from the eight-bit literal 'k'. 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 1: SUBLW 02h

Before Instruction		
W	=	01h
C	=	?
After Instruction		
W	=	01h
C	=	1 ; result is positive
Z	=	0
N	=	0

Example 2: SUBLW 02h

Before Instruction		
W	=	02h
C	=	?
After Instruction		
W	=	00h
C	=	1 ; result is zero
Z	=	1
N	=	0

Example 3: SUBLW 02h

Before Instruction		
W	=	03h
C	=	?
After Instruction		
W	=	FFh ; (2's complement)
C	=	0 ; result is negative
Z	=	0
N	=	1

SUBWF	Subtract W from f				
Syntax:	SUBWF f {,d {,a}}				
Operands:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$				
Operation:	$(f) - (W) \rightarrow \text{dest}$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table><tr><td>0101</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>	0101	11da	ffff	ffff
0101	11da	ffff	ffff		
Description:	Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1' the				

Q Cycle Activity:

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

Example 1: SUBWF REG, 1, 0

Before Instruction		
REG	=	3
W	=	2
C	=	?
After Instruction		
REG	=	1
W	=	2
C	=	1 ; result is positive
Z	=	0
N	=	0

Example 2: SUBWF REG, 0, 0

Before Instruction		
REG	=	2
W	=	2
C	=	?
After Instruction		
REG	=	2
W	=	0
C	=	1 ; result is zero
Z	=	1
N	=	0

Example 3: SUBWF REG, 1, 0

Before Instruction		
REG	=	1
W	=	2
C	=	?
After Instruction		
REG	=	FFh ; (2's complement)
W	=	2
C	=	0 ; result is negative
Z	=	0
N	=	1

PIC18F2221/2321/4221/4321 FAMILY

CALLW Subroutine Call Using WREG

Syntax:	CALLW				
Operands:	None				
Operation:	(PC + 2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0001</td><td>0100</td></tr></table>	0000	0000	0001	0100
0000	0000	0001	0100		
Description	<p>First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched. Unlike CALL, there is no option to update W, STATUS or BSR.</p>				
Words:	1				
Cycles:	2				
Q Cycle Activity:					

Q1	Q2	Q3	Q4
Decode	Read WREG	PUSH PC to stack	No operation
No operation	No operation	No operation	No operation

Example: HERE CALLW

Before Instruction

PC = address (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h

After Instruction

PC = 001006h
TOS = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h

MOVSF Move Indexed to f

Syntax:	MOVSF [z _s], f _d			
Operands:	0 ≤ z _s ≤ 127 0 ≤ f _d ≤ 4095			
Operation:	((FSR2) + z _s) → f _d			
Status Affected:	None			
Encoding:				
1st word (source)	1110	1011	0zzz	zzzz _s
2nd word (destin.)	1111	ffff	ffff	ffff _d
Description:	<p>The contents of the source register are moved to destination register 'f_d'. The actual address of the source register is determined by adding the 7-bit literal offset 'z_s' in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f_d' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).</p> <p>The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.</p> <p>If the resultant source address points to an indirect addressing register, the value returned will be 00h.</p>			

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVSF [05h], REG2

Before Instruction

FSR2 = 80h
Contents =
of 85h = 33h
REG2 = 11h

After Instruction

FSR2 = 80h
Contents =
of 85h = 33h
REG2 = 33h

PIC18F2221/2321/4221/4321 FAMILY

27.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings^(†)

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to V _{SS} (except V _{DD} and $\overline{\text{MCLR}}$)	-0.3V to (V _{DD} + 0.3V)
Voltage on V _{DD} with respect to V _{SS}	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V _{SS} (Note 2)	0V to +13.25V
Total power dissipation (Note 1)	1.0W
Maximum current out of V _{SS} pin	300 mA
Maximum current into V _{DD} pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD})	±20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD})	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports	200 mA

Note 1: Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

- 2:** Voltage spikes below V_{SS} at the $\overline{\text{MCLR}}$ /V_{PP}/RE3 pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the $\overline{\text{MCLR}}$ /V_{PP}/RE3 pin, rather than pulling this pin directly to V_{SS}.

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

PIC18F2221/2321/4221/4321 FAMILY

27.2 DC Characteristics: Power-Down and Supply Current PIC18F2221/2321/4221/4321 (Industrial) PIC18LF2221/2321/4221/4321 (Industrial) (Continued)

PIC18LF2221/2321/4221/4321 (Industrial)		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature		-40°C ≤ TA ≤ +85°C for industrial			
PIC18F2221/2321/4221/4321 (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	Typ	Max	Units	Conditions		
	Supply Current (IDD) ⁽²⁾						
	PIC18LF2X21/4X21	51	75	μA	-40°C	VDD = 2.0V	Fosc = 1 MHz (PRI_IDLE mode, EC oscillator)
		54	75	μA	+25°C		
		60	75	μA	+85°C		
	PIC18LF2X21/4X21	83	123	μA	-40°C	VDD = 3.0V	
		88	123	μA	+25°C		
		93	123	μA	+85°C		
	All Devices	180	260	μA	-40°C	VDD = 5.0V	
		180	260	μA	+25°C		
		180	260	μA	+85°C		
	Extended Devices Only	190	260	μA	+125°C		
	PIC18LF2X21/4X21	210	290	μA	-40°C	VDD = 2.0V	Fosc = 4 MHz (PRI_IDLE mode, EC oscillator)
		220	290	μA	+25°C		
		230	290	μA	+85°C		
	PIC18LF2X21/4X21	350	480	μA	-40°C	VDD = 3.0V	
		360	480	μA	+25°C		
		370	480	μA	+85°C		
	All Devices	0.69	1	mA	-40°C	VDD = 5.0V	
		0.70	1	mA	+25°C		
		0.72	1	mA	+85°C		
	Extended Devices Only	0.74	1	mA	+125°C		
	Extended Devices Only	3.7	4.0	mA	+125°C	VDD = 4.2V	Fosc = 25 MHz (PRI_IDLE mode, EC oscillator)
		4.6	5.0	mA	+125°C	VDD = 5.0V	
	All Devices	6.0	7.3	mA	-40°C	VDD = 4.2V	Fosc = 40 MHz (PRI_IDLE mode, EC oscillator)
		6.2	7.3	mA	+25°C		
		6.6	7.3	mA	+85°C		
	All Devices	6.8	9.2	mA	-40°C	VDD = 5.0V	
		7.0	9.2	mA	+25°C		
7.1		9.2	mA	+85°C			

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

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} , and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: 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.

The test conditions for all I_{DD} measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} or V_{SS} ;

$\text{MCLR} = V_{DD}$; WDT enabled/disabled as specified.

3: Low-power, Timer1 oscillator is selected unless otherwise indicated, where $\text{LPT1OSC}(\text{CONFIG3H}<2>) = 1$.

4: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

5: When operation below -10°C is expected, use T1OSC High-Power mode, where $\text{LPT1OSC}(\text{CONFIG3H}<2>) = 0$.

PIC18F2221/2321/4221/4321 FAMILY

FIGURE 27-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)

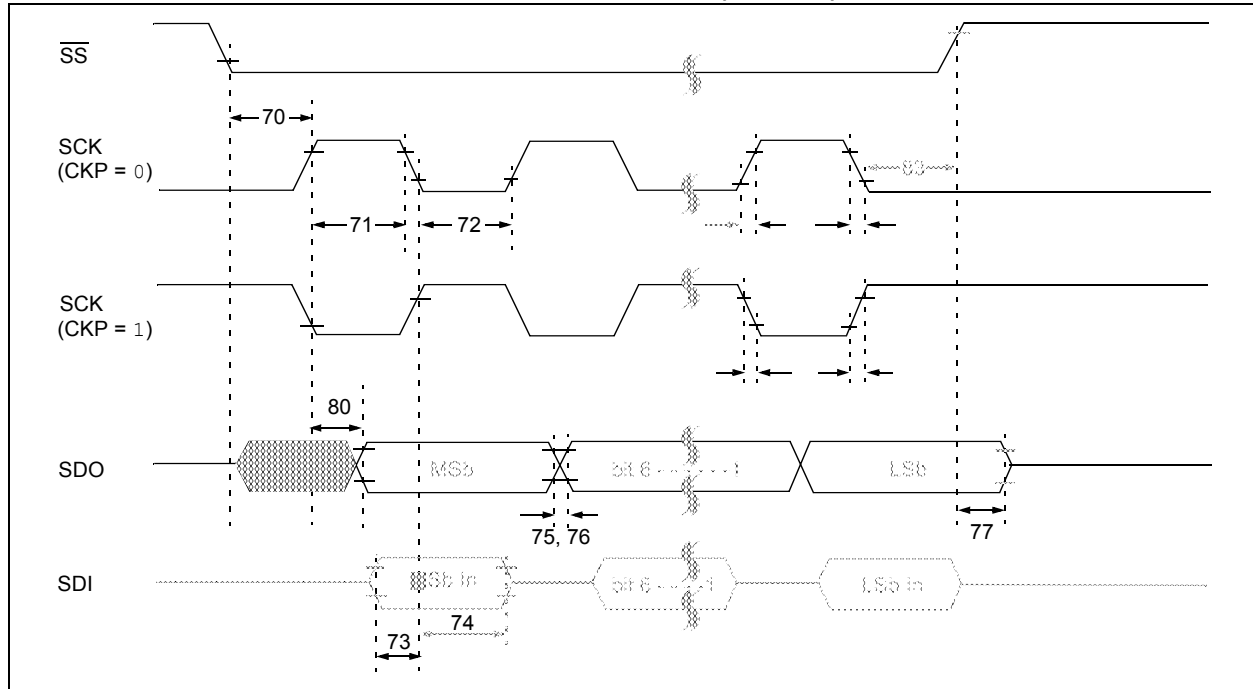


TABLE 27-16: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
70	Tssl2sch, Tssl2scl	SS ↓ to SCK ↓ or SCK ↑ Input	3 Tcy	—	ns	
71	Tsch	SCK Input High Time	Continuous	1.25 Tcy + 30	—	ns
71A		Single Byte	40	—	ns	(Note 1)
72	Tscl	SCK Input Low Time	Continuous	1.25 Tcy + 30	—	ns
72A		Single Byte	40	—	ns	(Note 1)
73	Tdiv2sch, Tdiv2scl	Setup Time of SDI Data Input to SCK Edge	20	—	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2	1.5 Tcy + 40	—	ns	(Note 2)
74	Tsch2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge	40	—	ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXXXX	—	25	ns
		PIC18LFXXXX		45	ns	VDD = 2.0V
76	TdoF	SDO Data Output Fall Time	—	25	ns	
77	TssH2doZ	SS ↑ to SDO Output High-Impedance	10	50	ns	
80	Tsch2doV, TscL2doV	SDO Data Output Valid after SCK Edge	PIC18FXXXX	—	50	ns
		PIC18LFXXXX		100	ns	VDD = 2.0V
83	Tsch2ssH, TscL2ssH	SS ↑ after SCK edge	1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.

PIC18F2221/2321/4221/4321 FAMILY

C

C Compilers	
MPLAB C18	330
MPLAB C30	330
CALL	294
CALLW	323
Capture (CCP Module)	147
Associated Registers	149
CCP Pin Configuration	147
CCPRxH:CCPRxL Registers	147
Prescaler	147
Software Interrupt	147
Timer1/Timer3 Mode Selection	147
Capture (ECCP Module)	154
Capture/Compare/PWM (CCP)	145
Capture Mode. See Capture.	
CCPRxH Register	146
CCPRxL Register	146
Compare Mode. See Compare.	
Interaction of Two CCP Modules	146
Module Configuration	146
Pin Assignment	146
Timer Resources	146
Clock Sources	35
Selecting the 31 kHz Source	36
Selection Using OSCCON Register	36
CLRF	295
CLRWDT	295
Code Examples	
16 x 16 Signed Multiply Routine	96
16 x 16 Unsigned Multiply Routine	96
8 x 8 Signed Multiply Routine	95
8 x 8 Unsigned Multiply Routine	95
Address Masking	182
Changing Between Capture Prescalers	147
Computed GOTO Using an Offset Value	62
Data EEPROM Read	91
Data EEPROM Refresh Routine	92
Data EEPROM Write	91
Erasing a Flash Program Memory Row	84
Fast Register Stack	62
How to Clear RAM (Bank 1) Using Indirect Addressing	73
Implementing a Real-Time Clock Using a Timer1 Interrupt Service	137
Initializing PORTA	111
Initializing PORTB	114
Initializing PORTC	117
Initializing PORTD	120
Initializing PORTE	123
Loading the SSPBUF (SSPSR) Register	170
Reading a Flash Program Memory Word	83
Saving STATUS, WREG and BSR Registers in RAM	109
Writing to Flash Program Memory	86–87
Code Protection	259, 274
Associated Registers	275
Configuration Register Protection	277
Data EEPROM	277
Program Memory	275
COMF	296
Comparator	243
Analog Input Connection Considerations	247
Associated Registers	247
Configuration	244

Effects of a Reset	246
Interrupts	246
Operation	245
Operation During Sleep	246
Outputs	245
Reference	245
External Signal	245
Internal Signal	245
Response Time	245
Comparator Specifications	350
Comparator Voltage Reference	249
Accuracy and Error	250
Associated Registers	251
Configuring	249
Connection Considerations	250
Effects of a Reset	250
Operation During Sleep	250
Compare (CCP Module)	148
CCPRx Register	148
Pin Configuration	148
Software Interrupt	148
Special Event Trigger	143, 148, 242
Timer1/Timer3 Mode Selection	148
Compare (ECCP Module)	154
Special Event Trigger	154
Computed GOTO	62
Configuration Bits	259
Context Saving During Interrupts	109
Conversion Considerations	387
CPFSEQ	296
CPFSGT	297
CPFSLT	297
Crystal Oscillator/Ceramic Resonator	29
Customer Change Notification Service	399
Customer Notification Service	399
Customer Support	399

D

Data Addressing Modes	73
Comparing Options with the Extended Instruction Set Enabled	76
Direct	73
Indexed Literal Offset	75
Instructions Affected	75
Indirect	73
Inherent and Literal	73
Data EEPROM Memory	89
Associated Registers	93
EEADR Register	89
EECON1 Register	89
EECON2 Register	89
EEDATA Register	89
Operation During Code-Protect	92
Protection Against Spurious Write	92
Reading	91
Using	92
Write Verify	91
Writing	91
Data Memory	65
Access Bank	67
and the Extended Instruction Set	75
Bank Select Register (BSR)	65
General Purpose Registers	67
Map for PIC18F2221/2321/4221/4321 Family	66
Special Function Registers	68

PIC18F2221/2321/4221/4321 FAMILY

DAW	298
DC Characteristics	347
Power-Down and Supply Current	337
Supply Voltage	336
DCFSNZ	299
DECF	298
DECFSZ	299
Development Support	329
Device Differences	386
Device Overview	9
Details on Individual Family Members	10
Features (table)	11
New Core Features	9
Other Special Features	10
Device Reset Timers	51
Oscillator Start-up Timer (OST)	51
PLL Lock Time-out	51
Power-up Timer (PWRT)	51
Time-out Sequence	51
Direct Addressing	74
E	
Effect on Standard PIC MCU Instructions	326
Effects of Power-Managed Modes on Various	
Clock Sources	38
Electrical Characteristics	333
Enhanced Capture/Compare/PWM (ECCP)	153
Associated Registers	166
Capture and Compare Modes	154
Capture Mode. <i>See</i> Capture (ECCP Module).	
Outputs and Configuration	154
Pin Configurations for ECCP1	154
PWM Mode. <i>See</i> PWM (ECCP Module).	
Standard PWM Mode	154
Timer Resources	154
Enhanced PWM Mode. <i>See</i> PWM (ECCP Module).	155
Enhanced Universal Synchronous Asynchronous Receiver	
Transmitter (EUSART). <i>See</i> EUSART.	
Equations	
A/D Acquisition Time	238
A/D Minimum Charging Time	238
Calculating the Minimum Required	
Acquisition Time	238
Errata	8
EUSART	
Asynchronous Mode	221
12-Bit Break Transmit and Receive	227
Associated Registers, Receive	225
Associated Registers, Transmit	223
Auto-Wake-up on Sync Break	226
Receiver	224
Setting up 9-Bit Mode with Address Detect	224
Transmitter	221
Baud Rate Generator	
Operation in Power-Managed Mode	215
Baud Rate Generator (BRG)	215
Associated Registers	216
Auto-Baud Rate Detect	219
Baud Rate Error, Calculating	216
Baud Rates, Asynchronous Modes	217
High Baud Rate Select (BRGH Bit)	215
Sampling	215

Synchronous Master Mode	228
Associated Registers, Receive	230
Associated Registers, Transmit	229
Reception	230
Transmission	228
Synchronous Slave Mode	231
Associated Registers, Receive	232
Associated Registers, Transmit	231
Reception	232
Transmission	231
Extended Instruction Set	
ADDFSR	322
ADDULNK	322
and Using MPLAB Tools	328
CALLW	323
Considerations for Use	326
MOVSF	323
MOVSS	324
PUSHL	324
SUBFSR	325
SUBULNK	325
Syntax	321
External Clock Input	30
F	
Fail-Safe Clock Monitor	259, 272
Exiting Operation	272
Interrupts in Power-Managed Modes	273
POR or Wake From Sleep	273
WDT During Oscillator Failure	272
Fast Register Stack	62
Firmware Instructions	279
Flash Program Memory	79
Associated Registers	87
Control Registers	80
EECON1 and EECON2	80
TABLAT (Table Latch) Register	82
TBLPTR (Table Pointer) Register	82
Erase Sequence	84
Erasing	84
Operation During Code-Protect	87
Reading	83
Table Pointer	
Boundaries	82
Boundaries Based on Operation	82
Operations with TBLRD and TBLWT (table)	82
Table Reads and Table Writes	79
Write Sequence	85
Writing	85
Protection Against Spurious Writes	87
Unexpected Termination	87
Write Verify	87
FSCM. <i>See</i> Fail-Safe Clock Monitor.	
G	
GOTO	300
H	
Hardware Multiplier	95
Introduction	95
Operation	95
Performance Comparison	95